Action Sequence Reproduction based on Automatic Segmentation and Object-Action Complexes

Mirko Wächter[1], Sebastian Schulz[1], Tamim Asfour[1],
Eren Aksoy[2], Florentin Wörgötter[2] and Rüdiger Dillmann[1]
[1]Institute for Anthropomatics, Karlsruhe Institute of Technology
Adenauerring 2, 76131 Karlsruhe, Germany
[2]Bernstein Center for Computational Neuroscience, University of Göttingen
III. Physikalisches Institut, Friedrich-Hund Platz 1, 37077 Göttingen, Germany
{waechter,s.schulz,asfour,dillmann}@kit.edu, {eaksoye,worgott}@physik3.gwdg.de

***Abstract*— Teaching robots object manipulation skills is a complex task that involves multimodal perception and knowledge about processing the sensor data. In this paper, we show a concept for humanoid robots in household environments with a variety of related objects and actions. Following the paradigms of Programming by Demonstration (PbD), we provide a flexible approach that enables a robot to adaptively reproduce an action sequence demonstrated by a human. The obtained human motion data with involved objects is segmented into semantic conclusive sub-actions by the detection of relations between the objects and the human actor. Matching actions are chosen from a library of Object-Action Complexes (OACs) using the preconditions and effects of each sub-action. The resulting sequence of OACs is parameterized for the execution on a humanoid robot depending on the observed action sequence and on the state of the environment during execution. The feasibility of this approach is shown in an exemplary kitchen scenario, where the robot has to prepare a dough.**

I. Introduction and Related Work

Robots already are versatile helpers in structured industrial applications, and in the future, will they increasingly be supposed to work also in human centered environments. If robots are expected to interact in an unstructured household instead of working in a well-known factory environment, problems become more complex: In order to fulfill requirements of everyday activities, a wide variety of complex questions have to be solved. The robot has to cope with unfamiliar situations and unknown options for interaction so that behavior and actions have to be highly adaptive. Consequently, common methods of robot programming are not directly applicable in the above mentioned scenario.

To address these problems, the concept of Programming by Demonstration (PbD) has become a common approach. It has developed rapidly since its origins in the mid-1980s. The work of Halbert [1], shows how to program a software system by example. During the 90ies, similar approaches transferred into the robotics domain were presented [2], [3], [4]. PbD is a technique that enables teaching a robot new skills and behaviors by demonstrating actions on concrete examples. It enables the robot to learn continuously from human observation in scenes of everyday life. Using parameterizable representations of the observed data allows applying the demonstration to new situations. Ultimately, after setting up the PbD System, advanced programming skills will be not any longer needed because also untrained users are able to teach new skills to the robot by demonstration.

The development of suitable action representations and algorithms for PbD has been a key research topic for the last decades. Neuroscientists, computer scientists and engineers alike have been working on relevant problems concerning this issue. Schaal et al. [5] discussed imitation learning as methodology for PbD from a computational point of view.

Ijspeert et al. suggested nonlinear dynamical systems for the representation of a demonstrated motion [6], called *dynamic movement primitives (DMP)*. Following this approach Gams et al. used a two-layered dynamical system that allows to extract both the frequency and the waveform of the demonstration signal to learn periodic tasks on a humanoid [7]. Ernesti et al. enrich these DMP formulation by extending the canonical system by one dimension using a two dimensional oscillator, which unifies the representation of a periodic movement and its transients [8]. Regarding various perturbations while execution, the basic formulation can be modified in additional ways. Two exemplary works in this direction are an approach that enables the generalization of DMPs to new situations using the available training movements and the goal of the task [9] and another approach using nonlinear dynamical systems with gaussian mixture models, which can respond immediately to perturbations encountered during the motion [10].

In contrast Ude et al. [11] suggested using b-spline wavelets as the representation of whole-body motion. Several approaches make use of Hidden Markov Models to learn and reproduce demonstrated actions [12], [13], [14].

Besides kinematic representations, further problems have to be addressed, in order to build a system capable of interacting in unstructured environments. Among them, the representation for the manipulation of objects in the environment is a crucial factor. In order create trajectories for executing these tasks, position and orientation of the manipulated object must be determined [15], [16].

Another key question is how to decompose the observed task into a set of sub-actions. Such a subdivision can then be utilized to formulate motion primitives for the execution

on the robot [18], [19], [20], [21], or as shown by Kulic et al. [22] even for incremental learning new templates in real-time.

An important principle, known as affordance and elementary for the proposed approach, is the relationship between objects and particular actions. Each object of a certain type, has a quality, which allows performing only specific actions using these objects. A framework for the action-centered representation of these correlations at different levels of hierarchy is presented by the formulation of the Object-Action Complex (OAC) concept [17]. OACs describe how a robot has to perform an action with an object to achieve a given goal. They take several sensor channels on different levels into account, ranging from sensorimotor- to semantic level. Examples of the sensorimotor level are joint angles or forces acting on the tool center point and, on the semantic level, the label of an object. Further, all OACs have preconditions and a prediction function associated with them, that encodes the belief how actions will impact on the world changes. This prediction is called the effect of the OAC.

Another preliminary method for our work, called "semantic event chain" (SEC) [23], [24], employs the spatial relation between objects for the subdivision of the demonstrated task. This approach makes use of the visual perception, more precisely, stereo and optical flow information. Hence, it is limited to demonstrations which can be reliable captured by image processing, in particular fulfill the conditions of image segmentation. The basic idea of the SEC is to extract the change of contact relations, i.e. all moments (*keyframes*) when any object comes into contact or loses contact with another object. Analogous to OACs, the method is based on the affordance principle, in particular the linking of objects and object relations to actions. Therefore, the linking of both methods is evident.

II. Overview

In this paper, a novel concept for the automatic adaptive reproduction of human demonstration on a robot is presented. The goal is to enable the reproduction of beforehand completely unknown complex tasks with multiple object interaction. Knowledge about these tasks is acquired by observation of the human demonstration and the involved objects. This observation is further processed to determine distinct sub-actions and object relations. These sub-actions are associated with Object-Action Complexes (OACs) [17], which are organized in a prior known library. These OACs represent basic object manipulation and interaction skills. The association is done by utilizing the observed world states to select OACs with matching preconditions and effects. Using these associations the robot can reproduce the before unknown action sequence.

III. Our Approach

In this section, we will present in detail the components of the proposed system for automatic adaptive action sequence reproduction. The proposed system mainly consists of three components (see Fig. 1): demonstration, representation and execution. The demonstration component is responsible for the acquisition and segmentation of motion data. The representation component contains the object-action complex library and the association of the segmentation with specific object-action complexes. The execution component provides the adaptive reproduction of the observed action sequence on a robot.

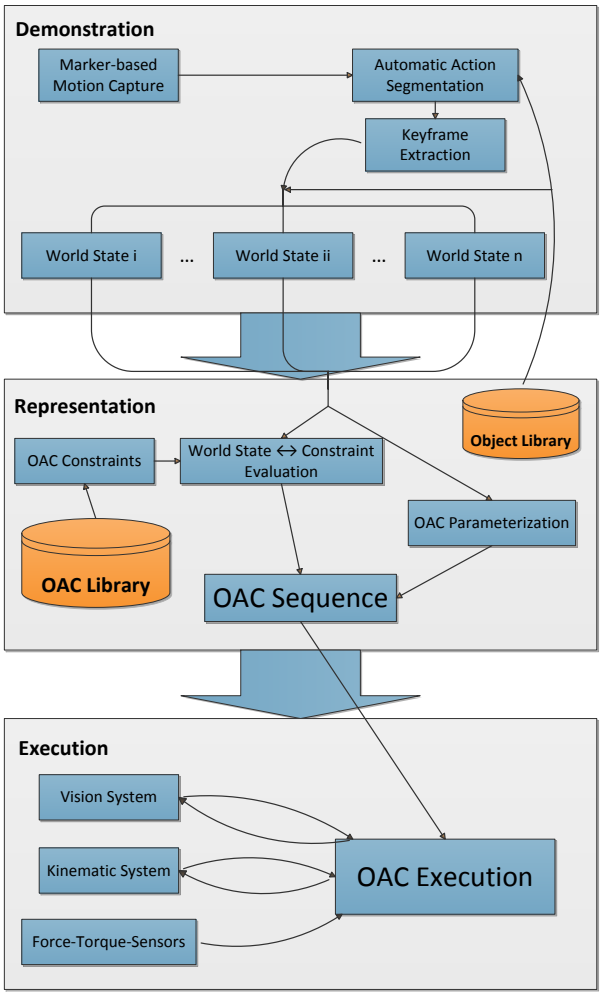

Fig. 1. System overview: The system consists of 3 components: demonstration, representation and execution.

A. Acquisition and segmentation of motion data

First, the human demonstration needs to be captured. The demonstrations are usually complex tasks like *preparing a dough*, which consist of several sub-actions. The trajectories of all components of the demonstrations need to be recorded. There are several ways to capture this data like inertial motion capture [25], marker-based motion capture with several cameras [26] or even 3D-based markerless motion capture [27]. Since this is a intensively researched field itself and outside the scope of this paper, a robust and precise marker-based motion capture system has been chosen in our experiments.

In order to extract the required trajectories of all components with this motion capture system, the agent and all involved objects have markers attached to them (see Fig. 2). To distinguish between the markers in the post-processing all markers are grouped by the object they are attached to. Further, the groups and the markers themselves are labeled. The labeling is important for the selection and parameterization of the object-action complexes, which are discussed in section III-B.

To segment the recorded action sequence we employ a method similar to the method presented in [23] by Aksoy et al. However, the main strategy remains the same. Instead of using a 3D vision system the demonstrated task is observed by an marker-based motion capture system. Consequently, the calculation of the object relations is not based on color segmentation but on the 3D Euclidian distance of objects over time. In this work, we call the resulting system for task segmentation *Automatic Action Segmentation* (AAS).

The environment is represented at any time for our approach as object relations. All object relations at a keyframe are considered as the *world state*. Contact changes between objects lead to a different world state. When the world state changes, there has been a change on the object relations, which in turn means that an action with an specific effect has happened. Thus, we are detecting actions by their effects on the environment. This approach is therefore model free in case of the actions and requires a simple spatial representation and a label for later processing for the objects. However, semantic information is not necessary.

The stated object relation is only of one kind at the moment: an object touches another object. One object can touch a set of other object, including the empty set. In contrast to the related work of Aksoy et al., we utilize the object distances to determine the keyframes instead of an exact graph-matching algorithm that extracts the main graphs of a sequence of graphs.

Further, Aksoy et al. use visual color segmentation and overlapping of color regions to detect changes of the object touching-relations. While this approach is flexible and does not require any prior knowledge about the objects, it lacks in robustness and precision. Therefore, we use marker trajectories from motion capture data to detect object touching-relations. All touching-relations at one frame are called the *world state*. The marker positions are the basis for all calculations. These markers are placed on the objects such that they are visible at all time. However, they do not represent the shape of the object. Thus, the distance calculations on the markers only may not detect all touching-relations between objects.

For calculation of the keyframes we use the markers $m_{G\in M,k=\{1\ldots|G|\},i=\{1\ldots l\}} \in \mathbb{R}^3$ and the following constraints, where M is the set of marker group sets and l is the length of the trajectory:

Whenever

i) a marker $m_{G_1,k,i}$ is sufficiently close enough to a marker of another object $m_{G_2,j,i}$,

ii) and the change of distance $|m_{G_1,k,i} - m_{G_2,j,i}|'$ of two markers is sufficiently small for a minimum number of frames n,

the transition *non-touching* $\rightarrow$ *touching* is made:

$$\begin{array}{ll|l} & |m_{G_1,k,i} - m_{G_2,j,i}| < d & \exists n_0 \in \{1 \ldots N\} \\ \wedge & |m_{G_1,k,i} - m_{G_2,j,i}|' < v & \forall i \in \{n_0 \ldots n_0 + n\} \end{array} , \quad (1)$$

where n_0 is the frame with the *non-touching* $\rightarrow$ *touching* transition, constant d is the distance threshold, constant v is the distance-change threshold

The two markers are labeled as "touching" from frame n_0 on until one of the previous conditions is false in the following n frames:

$$\begin{array}{ll|l} & |m_{G_1,k,i} - m_{G_2,j,i}| > d & \exists n_1 \in \{1 \ldots N\} \\ \vee & |m_{G_1,k,i} - m_{G_2,j,i}|' > v & \forall i \in \{n_1 \ldots n_1 + n\} \end{array} , \quad (2)$$

where n_1 marks the frame of the end of the touching-relation.

This way, passing other markers does not lead to touching-relations and noise in the data does not break up touching relations for a short time (see Fig. 3). At this point, for every change of the world state a new keyframe will be

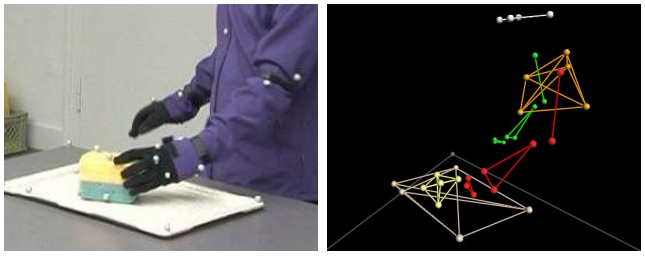

Fig. 2. Left: Human demonstrator with markers attached to him and all objects while wiping a tray. Right: The same frame of the demonstration as the 3D view of the reconstructed marker groups (different colors).

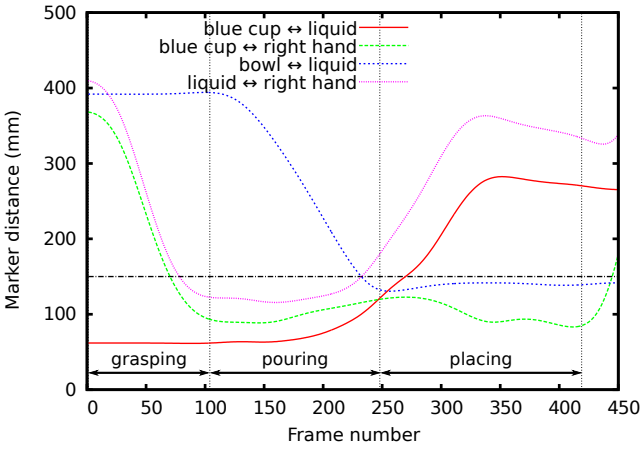

Fig. 3. Simple example of the AAS: depiction of the used measures (marker distance and change of distance) for the AAS. The black horizontal dotted line shows the distance threshold (d=150mm). The black vertical dotted lines show the resulting keyframes for the actions: grasping, pouring and placing.

inserted. However, this leads to oversegmentation of the action sequence with misleading keyframes. Some of these keyframes are separated by a relatively small number of frames, e.g. if an object is dropped onto another object. Since the demonstrated sub-actions in the scope of demonstrations for a robot always have a certain duration, these keyframes do not represent the desired segmentation. This can be solved by merging keyframes with only a few frames between them into one single keyframe. To achieve the merge the last keyframe of this group of keyframes and its relations are used. The previous keyframes of the group are discarded. A keyframe belongs to a group if the time difference to any other keyframe of the group is below a threshold. Proper chosing of this threshold is crucial for a correct segmentation. Too high and too low thresholds can both lead to additional false segments or missing segments depending on whether a touching relation started or ended.

With this segmentation method, the keyframes for the complete trajectory are calculated. In every keyframe, at least one object relation changes from *touching* to *non-touching* or vice versa. The corresponding object relations are stored for every keyframe and represent the current world state. In Fig. 3 the method's results are demonstrated in a simple example.

B. Object-Action Complex (OAC) Library

After applying the automatic action segmentation (see III-A), the demonstrated action sequence is subdivided into several sub-actions. However, the robot has no knowledge from the observation about how to reproduce the action sequence or any of the sub-actions. Although the trajectories of the markers and the involved objects are known, simply imitating the human demonstrator does not work. This is because the kinematics of the human and the robot usually are not interchangeable and because the objects are represented only by their attached markers. The robot has no knowledge about the shape of the objects and how to interact with them from motion capture data. Additionally, the used motion capture data does not contain any information about other perception channels, like the force applied during the demonstration. To enable the robot to reproduce the observed tasks a manually designed library of OACs is used. Though, only basic actions are stored in the library and complex tasks are observed.

The next step is to merge the gained information from the automatic action segmentation with the OAC library. The segmentation divides the action sequence in sub-actions, though, there is no association between the sub-actions and the OACs yet. Fortunately, the segmentation provides naturally for each sub-action a keyframe at the beginning and the end of the sub-action. At these keyframes, the current world state is stored. An OAC usually does not depend on the entire world state. Thus, a subset of the world states of the two keyframes is used as the preconditions and the effects of a sub-action.

There is a difference between the preconditions and effects of the OACs and two consecutive keyframes. The keyframes always contain instances of an object in the object relations, while on the other hand the OACs may contain variable terms in some or all the preconditions and effects, depending on the selected OAC. For instance, the grasping OAC has the following preconditions and effects:

$$\begin{array}{ll} Pre: & hand \leftrightarrow nothing \\ Effect: & hand \leftrightarrow object \in Graspable\ Objects \end{array}, \quad (3)$$

where $\leftrightarrow$ denotes a *touching* relation. Thus, to find the matching OAC, the object instances of the keyframes have to be validated against the compatibility with the OAC in question.

The world states of the segmented sub-action are utilized to perform a search within the OAC library to find a matching OAC by compairing the preconditions and effects of all OACs in the library with the previous and next world state of the segmented sub-action. It is important to notice, that OACs usually only depend on a small part of the world state and the remainder object relations are irrelevant. Hence, the object classes of the preconditions and effects of the OACs are checked against the specific object instances of the world states of the segmented sub-action for compatibility. If all preconditions and effects of an OAC are part of the segmented sub-action's world state, a link between the sub-action and the OAC is created and stored.

With the sequence of OACs it is possible to generate a new OAC that contains this sequence. The needed preconditions and effects of this new OAC can be calculated from the OAC preconditions and effects in the sequence. The complete method for this is shown in algorithm 1. The algorithm is divided into two parts: the calculation of the preconditions and the effects. The preconditions of sub-OACs are preconditions of the new OAC, if they are not effects of previous sub-OACs. The effects of the new OAC are the changes of the world state before the new OAC to the world state after it.

IV. Experiments

In this section, we will explain the experimental setup for the complete system and present an exemplary scenario for the application of the system.

A. Experimental Setup

The system for reproduction of the action sequence consist of two major hardware components. In this section, first the motion capture system and the second the humanoid robot, will be explained

1) Marker-based motion capture system: For motion capturing, we are using a multi-camera system with 10 cameras equipped with infrared lights and infrared filters. The human demonstrator has reflective markers attached to the torso, the arms and hands, and the head. On every object,at least three markers are asymmetrically placed to correct identify the pose and avoid instabilities in the assignment of markers.

2) Humanoid robot Armar-III: For the reproduction of the action sequence, we are using the humanoid robot ARMAR-III [28]. The kinematic chain of the robot consists of the following subsystems: As a base, it has a holonomic platform with three omniwheels. On this platform, a torso with three

oacs := list of (p:preconditions, e:effects)
Input: w_s := world state before new OAC
w_e := world state after new OAC
Result: list of preconditions and list of effects for new OAC
foreach *oac* $o_{current}$ *in sequence* **do**
  $o \leftarrow o_{current}$
  $preconditionRequired \leftarrow true$
  foreach *oac* $o_{current}$ *in sequence before* o **do**
    $o_b \leftarrow o_{current}$
    if $o_b.e \cap o.p \neq \{\emptyset\}$ **then**
      $preconditionRequired \leftarrow false$
    end
  end
  if $preconditionRequired = true$ **then**
    add $o.p$ to p_{new}
  end
end
foreach *object* o_s *in* w_s **do**
  foreach *object* o_e *in* w_e **do**
    if $o_s = o_e$ **then**
      add $(o_e.relations \setminus o_s.relations)$ to e_{new}
    end
  end
end
return p_{new} and e_{new}

Algorithm 1: Calculation of preconditions and effects of new OAC

degrees of freedom (DOF) is placed. Like a human, it has two arms. Each of them consists of seven joints and has a five finger pneumatic hand attached to it. The head kinematics is divided into the neck joints with 3 DOF and the two eyes with a common tilt joint and independent pan joints, resulting in 10 DoF in total. The visual perception of the robot is accomplished with a foveal and a peripheral stereo vision system.

The robot is equipped with a 6D-force-torque-sensor in both wrists to measure the force applied to the hand. This sensor is used in most OACs as a state trigger, trajectory modifier to reduce applied forces or merely as a trigger for aborting the OAC as a safety precaution.

3) Environment: The experiment was conducted in a kitchen environment. The robot is standing at a table with several objects on it: two cups of different color, one mixing bowl and a mixer. The cups are in the demonstration filled with a big marker to symbolize the liquid. For the reproduction a robot friendly liquid replacement, i.e. small balls, are used.

B. Exemplary Scenario

Analogously to the environment, we chose a task that belongs to the kitchen scenario: preparing a dough. This a complex task consisting of several OACs with multiple objects involved. It requires and shows all the components of our approach.

The execution of the task by an human demonstrator while being observed by the motion capture system is realized by the following sequence of commands:

1) Pour the *liquid one* into the *orange bowl*
2) Pour the *liquid two* into the *orange bowl*
3) Use the *electric mixer* for mixing the *dough*

This is one possible description for the OAC sequence that probably would be sufficient for most humans. It is written in a way that some important data for the execution is not explicitly described. A human infers the missing data from the context. However, a robot cannot execute this plan since it has a different view on the actions *pouring* and *mixing*. The OACs used with this system focus on moments when objects touch each other or stop touching. Thus, the AAS extracts the segmentation that is shown in Fig. 4. The figure shows the whole process of reproduction of action sequence of the described exemplary scenario. The left column shows the demonstration by a human at the keyframes that are extracted by the AAS. The graphs in the middle column represent the world state at each keyframe. Each node of the graph depicts an object and the connections illustrate the touching-relations between objects. Black connection lines depict already existing relations, while red solid lines stand for a new touch-relation, and dotted lines for fading touching-relations. The grey boxes in the right column show the selected OAC with the matching constraints. The wildcards in the OAC constraints are filled with the specific instances for this action sequence, which result from the previous and next world state as illustrated with the black arrows. The pictures on the far right show the robot while executing the action sequence.

During the reproduction of the dough preparation the robot needed four different OACs, some of them multiple times: Grasping, pouring, placing and mixing. This led to a new OAC with the following preconditions:

- $Left\ hand \leftrightarrow Nothing$
- $Right\ hand \leftrightarrow Nothing$
- $Liquid\ One \leftrightarrow Red\ cup$
- $Liquid\ Two \leftrightarrow Green\ cup$

and the resulting effects:

- $Right\ hand \leftrightarrow Mixer$
- $Liquid\ One \leftrightarrow Orange\ bowl$
- $Liquid\ Two \leftrightarrow Orange\ bowl$
- $Mixer \leftrightarrow Orange\ bowl$

The new OAC is inserted into the OAC library and available for future executions.

V. Conclusion

In this paper, we presented a system that first enables robots to observe human interaction with objects in unstructured environments. It then decomposes demonstrated tasks into sub-actions that can be mapped onto the entries of an action library. Finally, action sequences are parameterized for the current situation and can be reproduced on a robot. The feasibility was shown in an exemplary scenario for preparing a dough.

It can be summarized that our approach performs well in the chosen scenario, which covers frequent actions in the kitchen domain. The following desirable features that are circumvented or just not supported by other approaches are natively supported by our approach: Due to the fact that AAS relies on world states instead of commonly used agent poses we achieved time-invariance and pose-invariance. Furthermore, it is invariant to the kinematics of the demonstrator.

These features empower our approach to reproduce tasks that were previously completely unknown to the robot by automatically splitting them up in known sub-actions. However, these sub-actions need to be known beforehand and are the backbone of this approach. Learning these sub-actions only from observation is, even for a human, a challenging task. Humans usually need to evaluate them before achieving complete comprehension and take at least the haptic perception into account as well. To teach the robot these sub-actions, more specialized approaches might be required, for instance a multimodal approach integrating several sensor channels.

Future work could concentrate on integrating an interactive sequence completion for unknown sub-actions, where the robot signals that he could not comprehend a demonstrated sub-action and asks for user interaction to help him understanding it. The segmentation algorithm, in particular the keyframe merging, could be extended through automatic adaptation of the hyperparameters to reduce the dependency on correct parameterization.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement Nº 270273 (Xperience).

REFERENCES

[1] D. C. Halbert, "Programming by example," Ph.D. dissertation, University of California, Berkeley, 1984.

[2] Y. Kuniyoshi, M. Inaba, and H. Inoue, "Learning by watching: Extracting reusable task knowledge from visual observation of human performance," *IEEE Transactions on Robotics and Automation*, vol. 10, pp. 799–822, 1994.

[3] S. Muench, J. Kreuziger, M. Kaiser, and R. Dillmann, "Robot programming be demonstration (rpd) – using machine learning and user interaction methods for the development of easy and comfortable robot programming systems," in *Proc. International Symposium on Industrial Robots (ISIR)*, 1994, pp. 685–693.

[4] H. Friedrich, S. Münch, R. Dillmann, S. Bocionek, and M. Sassin, "Robot programming by demonstration (rpd): supporting the induction by human interaction," *Mach. Learn.*, vol. 23, no. 2-3, pp. 163–189, 1996.

[5] S. Schaal, A. Ijspeert, and A. Billard, "Computational approaches to motor learning by imitation," *Philosophical Transactions of the Royal Society of London: Series B, Biological Science*, vol. 358, no. 1431, pp. 537–547, 2003.

[6] A. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *In IEEE International Conference on Robotics and Automation (ICRA2002)*, 2002, pp. 1398–1403.

[7] A. Gams, M. Do, A. Ude, T. Asfour, and R. Dillmann, "On-Line periodic movement and force-profile learning for adaptation to new surfaces," Nashville, USA, December 2010.

[8] J. Ernesti, L. Righetti, M. Do, T. Asfour, and S. Schaal, "Encoding of periodic and their transient motions by a single dynamic movement primitive," in *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2012, pp. 57–64.

[9] A. Ude, A. Gams, T. Asfour, and J. Morimoto, "Task-specific generalization of discrete and periodic dynamic movement primitives," *Robotics, IEEE Transactions on*, vol. 26, no. 5, pp. 800–815, 2010.

[10] S. M. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with gaussian mixture models," *Robotics, IEEE Transactions on*, vol. 27, no. 5, pp. 943–957, 2011.

[11] A. Ude, C. G. Atkeson, and M. Riley, "Programming full-body movements for humanoid robots by observation," *Robotics and Autonomous Systems*, vol. 47, pp. 93–108, 2004.

[12] S. Calinon, F. Guenter, and A. Billard, "Goal-directed imitation in a humanoid robot," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 299–304.

[13] S. Calinon and A. Billard, "Recognition and reproduction of gestures using a probabilistic framework combining pca, ica and hmm," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 105–112.

[14] T. Asfour, P. Azad, F. Gyarfas, and R. Dillmann, "Imitation learning of dual-arm manipulation tasks in humanoid robots," *International Journal of Humanoid Robotics*, vol. 5, no. 2, pp. 183–202, December 2008.

[15] A. Ude, "Trajectory generation from noisy positions of object features for teaching robot paths," *Robotics and Autonomous Systems*, vol. 11, no. 2, pp. 113–127, 1993.

[16] A. Ude, D. Omrčen, and G. Cheng, "Making object learning and recognition an active process," *International Journal of Humanoid Robotics*, vol. 5, no. 2, pp. 267–286, 2008.

[17] N. Krüger, C. Geib, J. Piater, R. Petrick, M. Steedman, F. Wörgötter, A. Ude, T. Asfour, D. Kraft, D. Omrcene, A. Agostinig, and R. Dillmann, "Object-action complexes: Grounded abstractions of sensorimotor processes," *Robotics and Autonomous Systems*, 2011.

[18] J. Barbic, A. Safonova, J. Y. Pan, C. Faloutsos, J. K. Hodgins, and N. S. Pollard, "Segmenting motion capture data into distinct behaviors," in *GI '04: Proceedings of Graphics Interface 2004*. School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada: Canadian Human-Computer Communications Society, 2004, pp. 185–194.

[19] D. Kulic and Y. Nakamura, "Incremental learning of human behaviors using hierarchical hidden markov models," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 4649–4655.

[20] K. Yamane, M. Revfi, and T. Asfour, "Planning object receiving motions of humanoid robots with human motion database," in *ICRA*, 2013.

[21] M. Denisa and A. Ude, "Discovering new motor primitives in transition graphs," in *Intelligent Autonomous Systems 12*, ser. Advances in Intelligent Systems and Computing, S. Lee, H. Cho, K.-J. Yoon, and J. Lee, Eds. Springer Berlin Heidelberg, 2013, vol. 193, pp. 219–230.

[22] D. Kulic, W. Takano, and Y. Nakamura, "Online segmentation and clustering from continuous observation of whole body motions," *Robotics, IEEE Transactions on*, vol. 25, no. 5, pp. 1158–1166, 2009.

[23] E. E. Aksoy, A. Abramov, J. Dörr, N. Kejun, B. Dellen, and F. Wörgötter, "Learning the semantics of object-action relations by observation," *The International Journal of Robotics Research (IJRR)*, 2011.

[24] E. E. Aksoy, A. Abramov, F. Wörgötter, and B. Dellen, "Categorizing object-action relations from semantic scene graphs," in *ICRA*, 2010, pp. 398–405.

[25] D. Vlasic, R. Adelsberger, G. Vannucci, J. Barnwell, M. Gross, W. Matusik, and J. Popović, "Practical motion capture in everyday surroundings," *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, p. 35, 2007.

[26] J. Lee, J. Chai, P. S. Reitsma, J. K. Hodgins, and N. S. Pollard, "Interactive control of avatars animated with human motion data," in *ACM Transactions on Graphics (TOG)*, vol. 21, no. 3. ACM, 2002, pp. 491–500.

[27] P. Azad, *Visual Perception for Manipulation and Imitation in Humanoid Robots*. Springer, 2009, vol. 4.

[28] T. Asfour, K. Regenstein, P. Azad, J. Schröder, and R. Dillmann, "ARMAR-III: a humanoid platform for perception-action integration," in *Proc., International Workshop on Human-Centered Robotic Systems (HCRS), Munich*, 2006, pp. 51–56.

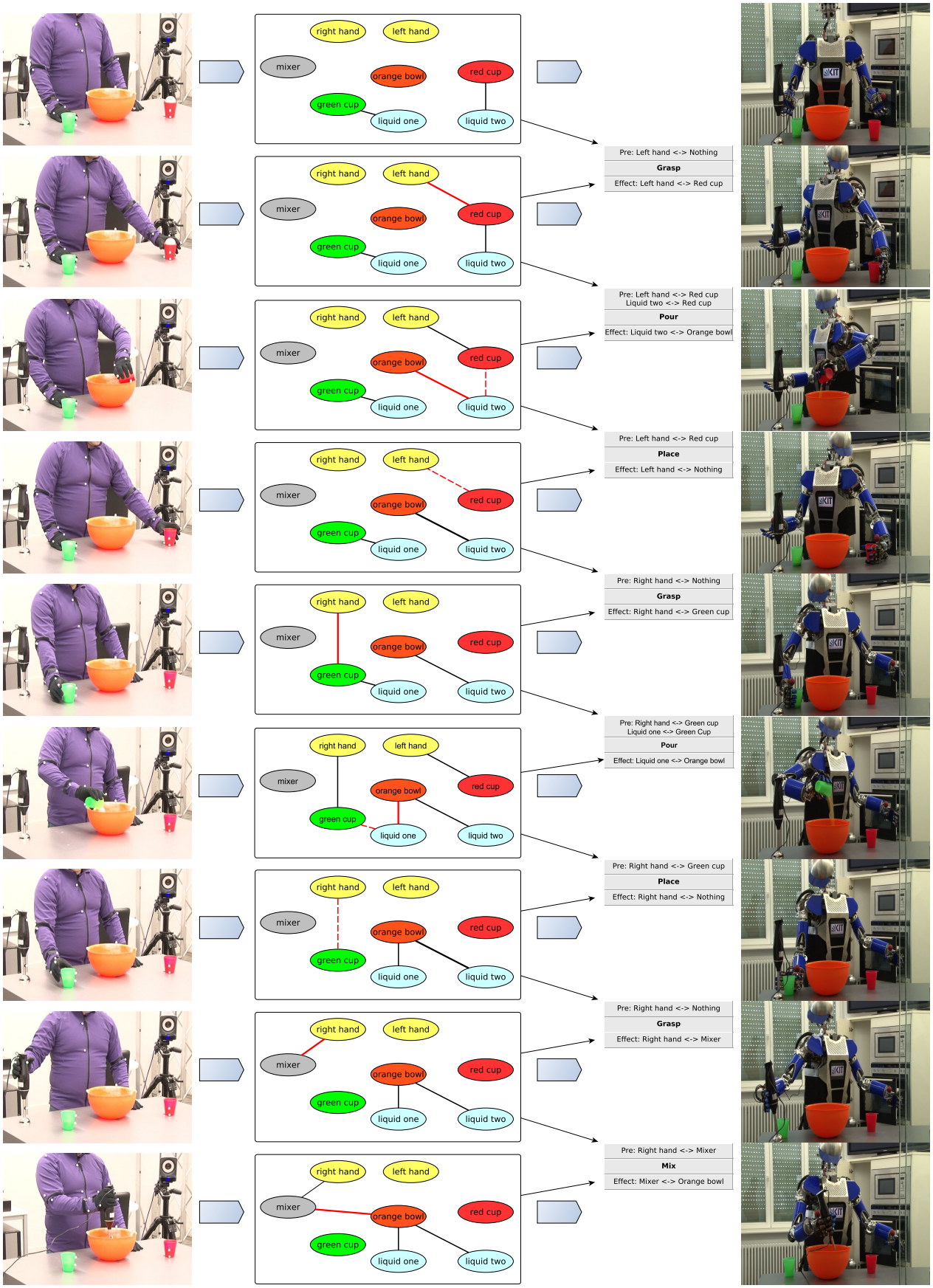

Fig. 4. Demonstrated action sequence at the detected keyframes (left column), the extracted world state at each keyframe (middle column), and the selected OACs from the two corresponding world states, which are executed on the robot (right column).