Künstl Intell (2010) 24: 295–303
DOI 10.1007/s13218-010-0060-0

FACHBEITRAG

Advances in Robot Programming by Demonstration

Rüdiger Dillmann · Tamim Asfour · Martin Do · Rainer Jäkel · Alexander Kasper · Pedram Azad · Aleš Ude · Sven R. Schmidt-Rohr · Martin Lösch

Received: 26 April 2010 / Accepted: 28 July 2010 / Published online: 19 August 2010

Abstract Robot Programming by Demonstration (PbD) has been dealt with in the literature as a promising way to teach robots new skills in an intuitive way. In this paper we describe our current work in the field toward the implementation of PbD system which allows robots to learn continuously from human observation, build generalized representations of human demonstration and apply such representations to new situations.

1 Introduction and Related Work

Learning of skills and behaviours that can be applied to solve a given task regardless of the current configuration of the external world is a difficult problem because the search space that needs to be explored is potentially huge [1]. The size of the search space depends both on the number of degrees of freedom of the robot and on the objects involved in the action. Furthermore, external objects can affect the search space indirectly. To overcome problems arising from high dimensional and continuous perception-action spaces, it is necessary to guide the search process. One of the most successful paradigms that can be used for this purpose is imitation learning or robot programming by demonstration [1, 2].

Several imitation learning systems and architectures based on the perception and analysis of human demonstrations have been proposed (see [2–7]). In most of the proposed approaches, the imitation process proceeds through three stages: (1) perception and analysis of human demonstration, (2) representation of the demonstration, and (3) reproduction of the demonstrated task on the robot. Known approaches in the literature can be divided between two trends regarding the way demonstrations are represented, and the way such representations are generated: trajectory-level representations in the form of non-linear mappings between sensory and motor information [8–14], and symbolic-level representations that decompose demonstrations into sequences of more abstract perception-action units [15–20]. While trajectory-level representations allow different types of motions to be encoded, they do not allow high-level tasks to be generated. On the other hand, symbolic-level representations allow action hierarchies and rules to be learned, however they require pre-definded sets of motor controllers for reproduction.

A key issue in all these approaches is to find a generic action representation which (1) express actions as a combination of meaningful elements called motor primitives, (2) learn such motor primitives, and (3) use them to recognize and synthesize actions. In other words, such representations should allow action planning, action recognition, and action synthesis. Several representations have been proposed in the past; among the most successful are non-linear dynamic systems [11] and hidden Markov models [6, 13, 21], which have been demonstrated to enable both action recognition and action synthesis. Several approaches have dealt with the extraction of motor primitives from observed human motion, the classification of demonstrated activities and well as the learning and sequencing of the underlying motor primitives [22–26]. In addition, action description languages have been also proposed to express human activities [27–29].

R. Dillmann · T. Asfour (✉) · M. Do · R. Jäkel · A. Kasper · P. Azad · S.R. Schmidt-Rohr · M. Lösch
Adenauerring 2, 76131 Karlsruhe, Germany
e-mail: asfour@kit.edu

A. Ude
Jamova cesta 39, 1000 Ljubljana, Slovenia

2 Overview

In Programming by Demonstration, two different lines of research can be identified based on the representation of manipulation knowledge: subsymbolic and symbolic. In this work, a subsymbolic approach, see Sect. 3, based on learning Dynamic Movement Primitives (DMP) [11], and a partially symbolic approach, see Sect. 4, based on learning the parameterization of predefined movement primitives, are presented. In the first approach, markerless human motion capture and a predefined, general mapping interface, the Master Motor Map (MMM), are used to map a human demonstration to a sequence of DMPs. Each DMP represents an attractor landscape described by a second order dynamical system, which is a single, abstract, subsymbolic representation of a set of demonstrations. In the second approach, human demonstrations are mapped to sequences of predefined motion primitives, i.e. *grasp*, *ungrasp* and *move*, which are implemented using constrained motion planning. In this context, symbolic pre- and post-conditions can be learned automatically [16]. The identification of object features, which are relevant for a given task, is a prerequisite for learning and generalization of manipulation knowledge. In Sect. 5, the required attributes to model interaction tasks of a service robot, e.g. in a kitchen environment, are analyzed. The derived basic actions contain movement primitives, like *grasp*, *ungrasp* and *move*, and perception primitives, which can't be learned in our current system. The object attributes relevant to each basic action are derived, which is the basis for assigning symbolic information, e.g. labels like heavy or light, to movement primitives and learn the dependence of subsymbolic information on object properties.

The connection of both research lines, i.e. between high-level, symbolic information and low-level, subsymbolic information, is challenging. In the first approach, the learned DMPs are enriched manually with symbolic information and a symbolic planner can be used to generated sequences of DMPs to solve a problem on the task level. The subsymbolic information, which can be efficiently adapted to changes in the start and goal, is used to generate robot motions online. In the second approach, the parameterization of the constrained motion planner, which is used to implement *grasp*, *ungrasp* and *move*, is automatically learned. The learned parameterization, which is called manipulation strategy, is a flexible, constraint-based representation of the search space.

The main advantage of the first approach is the fast, online adaptation to perturbations in the environment and learning of complex robot motions. The second approach offers generalization based on symbolic information, e.g. object properties, and planning of robot motions in the full configuration space of the robot. In contrast to the first approach, global and self collision avoidance can be easily integrated but online adaptation to fast changes in the environment is not possible since planning time dominates the execution time. The advantages and disadvantages of both approaches are complementary. Current research focuses on the connection of subsymbolic and symbolic approaches. Future directions and current limitations will be discussed in Sect. 6.

3 Learning Imitation Strategies

3.1 Markerless Human Motion Capture

Markerless human motion capture is a prerequisite for learning from human observation in a natural way. The sensor system to be used for this perceptual capability is the wide angle camera system built-in in the head of ARMAR-III. The two main problems are to capture real 3D motion despite the small baseline of 90 mm as well as to meet the real-time requirements for online imitation learning. As probabilistic framework, a particle filter is used. In our earlier work [30], we have introduced the integration of a 3D head/hand tracker as an extra cue into the particle filter. This additional cue allows to reduce the effective search space by dragging the probability distribution into a relevant subspace of the whole search space. In our more recent work [31, 32] we have focused on improving the accuracy and smoothness of the acquired trajectories by analyzing and solving the typical problems with markerless human motion capture using particle filters. In order to achieve this goal, a prioritized fusion method, adaptive shoulder positions, and adaptive noise in particle sampling have been introduced. Furthermore, the redundant inverse kinematics of the arm, given a hand and a shoulder position, were integrated into particle sampling, in order increase robustness to unexpected movements, to allow immediate recovery from mistrackings, and to support application of the system at lower frame rates. After sampling a subset of the particles according to the redundant inverse kinematics, several runs of an Annealed Particle Filter [33] are performed to refine the probability distribution.

3.2 Master Motor Map

To allow the reproduction of human motion acquired from different human motion capture systems on different robot embodiments as well as to allow the development and evaluation of action recognition systems independent from the data source, we have specified the so-called Master Motor Map (MMM) as an interface for exchanging motor knowledge between different embodiment, and as a framework for decoupling data from various sources accounting for perception, visualization, reproduction, analysis, and recognition of human motion. The MMM is defined as a three-dimensional reference kinematic model of the human body enriched with body segment properties. The strategy with respect to the kinematic model is to define the maximum number of degrees of freedom (DoF) that might be used by any applied module [34].

3.3 Action Representations

To generate graping and manipulation tasks through imitation, motor knowledge learned from human observation must be represented in a way, which allows the adaptation of learned actions to new situations. For this purpose, we investigated and applied Dynamic Motor Primitives (DMP) as proposed in [11]. A DMP provides a representation of a movement segment by shaping an attractor landscape described by a second order dynamical system. Using this formulation, discrete and periodic movements can be described. In [35], we applied a motion representation based on dynamic movement primitives (DMPs), which has the advantage that perturbations can be directly handled by the dynamics of the system. Starting from the observation of a human performing a specific task, motion data is obtained, which is segmented automatically regarding the velocity and position changes of the hand or the object motion. After mapping automatically these motion segments onto the robot using the MMM Interface, DMPs are learned and stored in a motion library. Semantic information is added manually to the movement primitives such that they can code object-oriented actions. To reproduce these actions on a robot, a sequence of DMPs is selected and chained, either manually or through a symbolic planner, to achieve the task goal. The imitation of a pick-and-place action consists e.g. of learned DMPs for the different movement segments: *approach, place* and *retreat*. At the moment, human input is restricted to generating learning examples, defining the MMM Interface, which is valid for all imitation tasks, and to add semantic information to learned movement primitives.

3.4 Action Reproduction

The proposed framework was used to implement grasp scenario and a shell game scenario. For the grasp scenario, three primitive classes of human movements have to be captured and added to the library of DMPs: approach, pick and place, and retreat. Concerning the approach and retreat movement, each class includes two DMPs assuming that, e.g. an approach movement is targeting an object in front of the robot, while the object position may vary along the vertical axis. For the pick and place class, we generate four DMPs, which enable placing of objects from back to the front, from left to right and vice versa (see Figs. 1 and 2). The shell game scenario features a higher complexity, hence, in addition to the existing DMPs which were applied on grasping, sliding movements were demonstrated to the robot. For this purpose, the human user was asked to move the object along a figure eight trajectory. Segmentation led to four distinct movements, which can be distinguished whether the cup was moved from left to right, away or towards the robot. Adding the four sliding DMPs to the library, a set of movement primitives is obtained which cover the motion needed for performing the shell game. As depicted in Fig. 3, the shell game could be reproduced successfully by the humanoid robot ARMAR-III. No failures were encountered in both examples. The similarity of the generated robot motions to the human demonstrations using the DMP framework is described in [35].

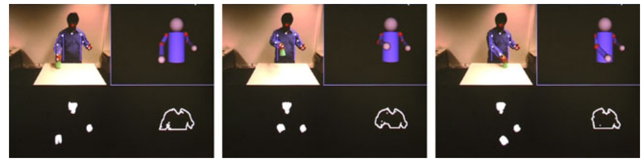


Fig. 1 Image samples of demonstrated of human motions

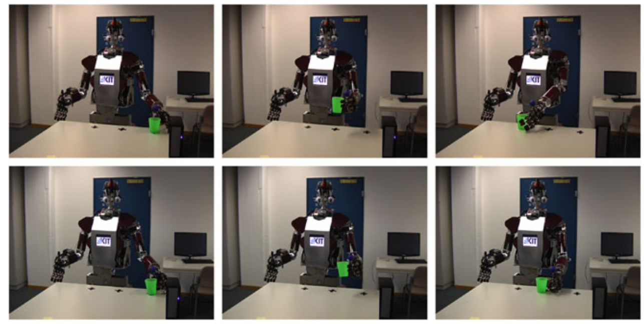


Fig. 2 Image samples of the online imitation of human motion by the humanoid ARMAR-IIIb

4 Learning Manipulation Strategies

The PbD process is summarized in Fig. 4. A human operator demonstrates the task on real objects in an environment [36] being observed by multiple sensor systems including a 6D motion tracking device for the wrist position and orientation and two datagloves measuring 22 degrees of freedom of the human hands. The sensor data is filtered, segmented and mapped to the symbolic predefined movement primitives: *grasp*, *ungrasp* and *move*. Each movement primitive is implemented using constrained motion planning. In general, the search space for grasp and move differ in the dimensionality of the problem and two different approaches are investigated. In the case of grasping, heuristics for the constrained motion planning are learned from the human operator, which is regarded as a complex parameter of the grasp movement primitive. In the case of moving, the search space itself depends on the task to learn and is automatically learned based on the human demonstration. The search

Fig. 3 The humanoid robot ARMAR-IIIb playing shell game through learning from human observation

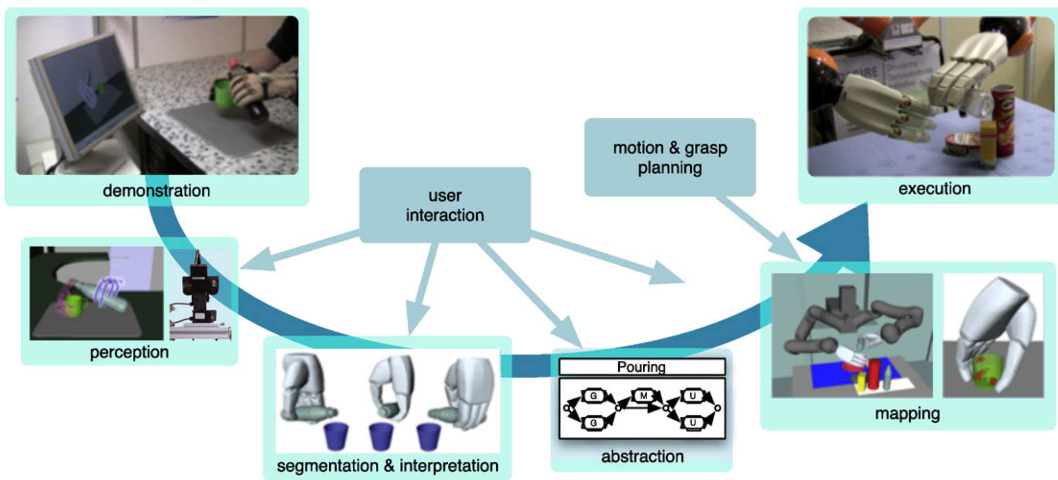


Fig. 4 Programming by demonstration: process overview, [38]

space is represented as a complex network of temporal and domain constraints, which is regarded as the parameter of the move movement primitive.

For each grasp operator, the example-trajectory is obtained by storing the wrist and fingertip trajectories relative to the grasped object. The demonstrated trajectories are mapped to the robot hand by using virtual fingers [37], which abstract a group of real fingers applying similar forces to the object to a single virtual finger, and by locally solving an optimization problem, which minimizes the distance of the robot finger tips to the tips of the virtual fingers and the distance of the robot wrist to the human wrist. Based on this initial mapping a probabilistic model based on factor graphs is learned, which explicitly models the optimization and modeling errors. This learned *variation model* represents a time dependent sampling distribution of the robot configuration space, that is used in a probabilistic motion planer to generate valid solutions for grasps of similar objects in new environments. By explicitly modeling the outcome of the transformation process as a stochastic process, an automatic weighting between exploitation of the knowledge demonstrated by the human operator and the fast exploration of the configuration space is achieved. The complete process is shown in Fig. 5.

In the execution environment, probabilistic motion planning is used to generate grasping motions for the robot system based on the learned *variation model*. The *variation model* is efficiently evaluated by sampling from the product distribution of the marginals, which are calculated using loopy belief propagation. This non-uniform sampling distribution is used in the probabilistic motion planner as a heuristic to guide the search process, allowing for the automatic weighting between exploitation of the learned task-dependent knowledge and the exploration of the search

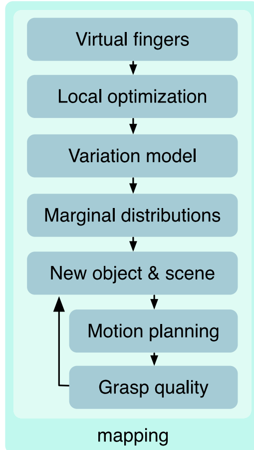


Fig. 5 Mapping of grasping strategies: overview, [38]

space. The incorporation of variations into the strategy representation allows for the flexible application of the learned strategy to different objects and environments. The advantages of using motion planning are generalization to environments with different obstacles and self collision avoidance. The generality of the approach has been demonstrated on two different experiments on a real anthropomorphic robot system with seven different objects. Details are given in [38].

For task knowledge representation we developed a partially symbolic representation of manipulation strategies that explicitly describes the search space for trajectories consistent with the strategy by a complex network of temporal and domain constraints. Based on the structure of constraints, manipulation strategies can be efficiently learned using the PbD paradigm and generalized to different robots, objects and environments on a symbolic level. Recent advances in the field of constraint motion planning are incorporated to plan robot trajectories based on a given manipulation strategy.

A manipulation motion is defined as an unconditioned motion of the robot system. The most common representation is a trajectory in the configuration space of the robot, that can be learned by playback programming and directly executed on the robot system. In general, generalization to different domains, e.g. with different start, target and object positions is not possible. In order to improve generalization, allowed variations of the trajectories can be learned based on multiple demonstrations. In [39], Gaussian mixture regression is used to determine a more flexible trajectory representation based on Gaussians. The main advantages of purely subsymbolic representations are fast learning times and low effort for the transformation to the robot system. Generalization to new objects and environments is complicated, due to the lack of understanding of what the goals are and why the solution is structured in a specific way. On the other hand, background knowledge can be easily integrated into a symbolic, e.g. STRIPS-like, representation. The symbolic description allows for the generalization based on symbolic properties, leading to a high degree of reusability, i.e. actions can be applied to objects with equal properties. Due to the complexity of robot manipulation, purely symbolic descriptions are insufficient to represent manipulation motions as an input for motion planning techniques. Consider the pour-in task, which could be described by the target relation *isFilled*(*Glass*, *Water*) and runtime constraint !*isWet*(*Table*). This simple symbolic description demands a powerful planning system taking the water dynamics into account. By mapping the relation !*isWet*(*Table*) to a subsymbolic constraint, that restricts the orientation of the bottle to be "upright", the problem complexity can be heavily reduced. Based on this observation, a representation capturing symbolic and subsymbolic properties of manipulation motions has been developed. In general, manipulation motions are heavily constrained, e.g. pushing a button requires the robot to stay in contact with a small part of an object. Instead of viewing constraints as a way to restrict motions, we regard constraints as the atomic element of manipulation motions and strategies, which can be combined in sequence and in parallel to describe the space of trajectories consistent with the manipulation motion. By introducing object depended constraints, e.g. staying on the table top, we derive a representation, that can be easily transformed to new environments based on its symbolic properties and easily executed based on the subsymbolic information provided by domain constraints.

The novel representation of manipulation strategies is based on atomic constraints. For each constraint, a formulation known from motion planning has been employed, which restricts the set of valid configurations by testing if a given point stays within a given region. Learning of new manipulation motions is reduced to learning the parameterization of a specific region type, which optimally covers the trajectory of a predefined point. The set of predefined points contains a.o. the position, orientation (and its derivatives) of object features and the robot manipulators. The set of region types considered in the learning process consists of cones, spheres, cubes and cylinders. For each class, the smallest representative containing all values of a point on a given trajectory can be efficiently calculated using a search algorithm. The result of the supervised learning process is a manipulation motion, that can be visualized as a strategy graph. By assigning regions to certain object features, the representation is (partially) symbolic, which can be efficiently exploited to reproduce the manipulation motion on different robot systems in different environments. In the pour-in experiment, the environment of the robot system contained additional objects and the robot had grasped a different type of cup. The learned manipulation motion was automatically transformed into this environment by using the predefined cylindrical region of the new cup and the developed constraint motion planning algorithm to incorporate collision avoidance. Consistent trajectories were planned, that differ fundamentally from the demonstrated trajectory, indicating that the relevant features of the manipulation had been learned. Further details are given in [40].

5 Learning Object Models

The goal of our research is to create an object model hierarchy for robots. This hierarchy should be created by a human user with little programming knowledge, supported by the modeling system. As an application example, we use a typical kitchen environment. The next two sections describe first the internal structure of the object models in more detail and then lead to the deduction of the different modeling tasks that need to be solved in such environments by intuitive user interaction.

5.1 The Object Model Hierarchy

In order to represent all the different objects, an object model needs to be very flexible and versatile. To achieve this, we developed a model consisting of four main parts [41]: object classes, object instances, features and attributes. In this concept, object classes consist of features (and potentially additional attributes), whereas features in turn consist of one or more attributes. Attributes are low-level descriptions of object properties, such as geometry, weight, texture, etc. Features describe higher level properties of objects which combine different attributes, e.g. the feature *is container* which implies attributes like *filling state*, *content type*, etc. On the top end of the hierarchy, object classes represent complete families of objects, such as cups, plates, forks or chairs. Objects of the same object classes share characteristic features and attributes. By setting special default values for the attributes of the object classes, sub-classes like e.g. *wooden chair* can be specified. Finally, object instances represent objects in a real world scene by instancing the appropriate object class and thus, setting situation and object specific values for its attributes. A more detailed explanation of this approach can be found in [42]. Based on this object model concept, two questions need to be answered to create a model for a real world scene: first, which attributes need to be modeled to describe the objects in the scene properly? And second, how can appropriate default values for these attributes be set by the human user in an intuitive way? The next section answers the first question whereas the second part of this paper describes a possible answer to the second question for two exemplary attributes.

5.2 From Tasks to Object Attributes

To create meaningful object classes to represent real world objects, the core attributes that are common to all objects of the domain in question need to be identified and modeled. The identification of these attributes was achieved through a careful analysis of possible interactions with the observed objects. This analysis consisted of three steps: the identification of the potential interaction tasks, the separation of the tasks into basic actions and finally the derivation of the attributes which are necessary to execute these basic actions.

The setting of service robots in a kitchen was taken as exemplary domain in our investigation. The analysis therefore concentrates on interaction tasks which service robots will be able to carry out in the near future. In the following two subsections, the derivation of basic actions from these tasks and of required object attributes from actions are presented.

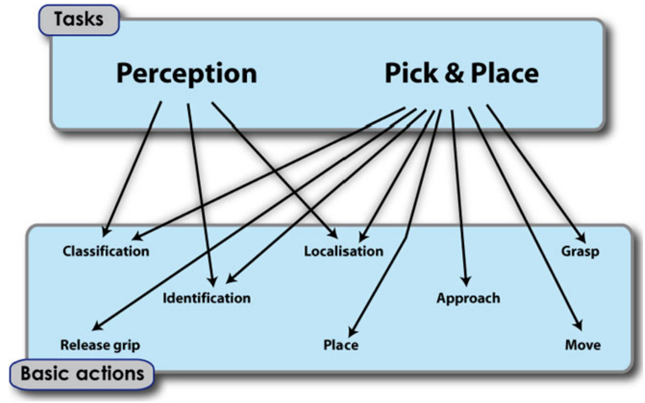


Fig. 6 Exemplary tasks and corresponding basic actions

5.2.1 Deriving Basic Actions

For the chosen domain, we identified several important tasks: For the fundamental recognition and localization of objects, the encompassing task is perception. As Fig. 6 exemplifies, perception tasks can be subdivided into three basic actions: classification, identification and localization. Figure 6 also shows that other tasks like pick & place, too, partly employ the same basic actions, but rely on additional actions like approach, grasp, move etc. In the same way, the remaining tasks of open/close, fill/empty and utilize were broken down into several basic actions. The analysis revealed that many of the basic actions are part of more than one task. The aforementioned classification, identification and localization, for example, are integral parts of each of the analyzed tasks.

5.2.2 Deriving Required Attributes

Now that the basic actions are known, the useful attributes required to perform these actions can be derived. Figure 7 shows this process at the example of moving an object. In

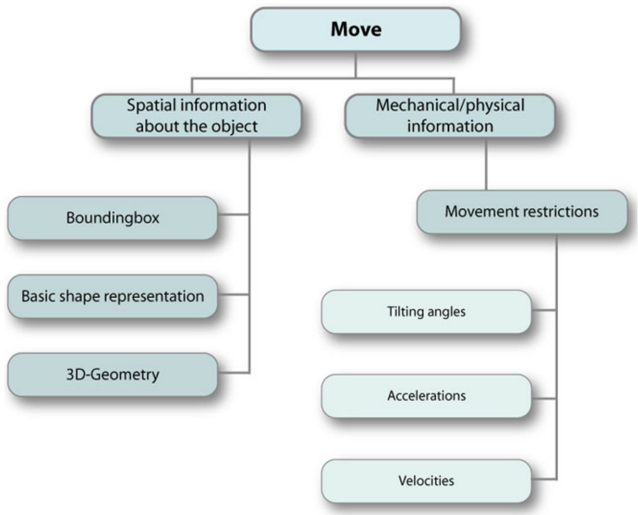


Fig. 7 The process of action analysis (here: movement of an object)

this case spatial information is needed e.g. to avoid collisions. This can be represented for example as a bounding box, in form of a basic shape representation or through highly detailed 3D geometry. When moving the object, mechanical and physical information is also crucial. Most important here are the movement restrictions, i.e. tilting angles, maximum accelerations and maximum velocities. In this fashion all of the basic actions, derived from the set of possible interactions in the environment, were analyzed and thus necessary object attributes extracted.

The most important resulting attributes for this domain are: movement restrictions, basic shapes, weight, bounding box, main axes, stable positions, grasp forces, grasp contact points, deformability, environmental conditions, risk potential, texture, colour graph, 3D geometry, type of locking mechanism, closability, graph of potential usages, container type and filling capacity. These attributes require different approaches to achieve intuitive, fast and exact object modeling. Two such attributes, namely stable positions and movement restrictions, and the way of their modeling in an interactive way are described in [41].

6 Conclusion and Future Work

In Programming by Demonstration, the connection between high-level, symbolic information and low-level, subsymbolic information is challenging and remains unsolved. At the symbolic level, generalization to different environments and objects is possible based on properties and the relation of objects in the scene. Subsymbolic information is missing to generate robot trajectories consistent with the symbolic goals and runtime conditions. At the subsymbolic level, information to adapt trajectories to perturbations in the start, goal and scene is present but generalization to different objects and environments is limited. In our current work, these two different lines of research are represented by learning imitation strategies and learning manipulation strategies. In both approaches, current work deals with the connection of the symbolic and subsymbolic levels. In learning imitation strategies, the subsymbolic information represented by the Dynamic Movement Primitives is enriched manually by symbolic pre- and post-conditions. In future work, the human teacher will be taken out of the learning loop by automatically attaching symbolic information to the learned DMPs. In learning manipulation strategies, the symbolic pre- and post-conditions of predefined motion primitives are automatically learned. In this line of research, current work focuses on learning the complex parameterization of the predefined motion primitives, closing the gap between the symbolic and subsymbolic level. Generalization of learned manipulation knowledge to different objects and obstacles is a prerequisite for the development of an autonomous robot, which acts in a large domain, e.g. the human environment. In this work, relevant object attributes were identified and future work will concentrate on learning the connection between object attributes and subsymbolic manipulation knowledge, which is the basis for automatic generalization to different objects.

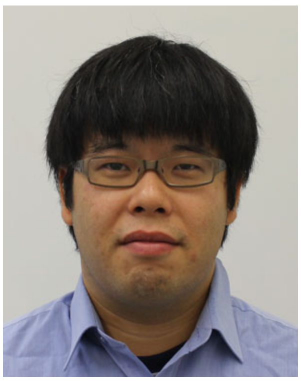
References

1. Schaal S (1999) Is imitation learning the route to humanoid robots? Trends Cogn Sci 3(6):233–242
2. Dillmann R (2004) Teaching and learning of robot tasks via observation of human performance. Robot Auton Syst 47(2–3):109–116
3. Kuniyoshi Y, Inaba M, Inoue H (1994) Learning by watching: extracting reusable task knowledge from visual observation of human performance. IEEE Trans Robot Autom 10:799–822
4. Breazeal C, Scassellati B (2002) Robots that imitate humans. Trends Cogn Sci 6(11):481–487
5. Schaal S, Ijspeert A, Billard A (2003) Computational approaches to motor learning by imitation. Philos Trans R Soc Lond B, Biol Sci 358(1431):537–547
6. Billard A, Siegwart R (2004) Robot learning from demonstration. Robot Auton Syst 47(2–3):65–67
7. Billard A, Calinon S, Dillmann R, Schaal S (2008) Robot programming by demonstration. In: Siciliano B, Khatib O (eds) Handbook of robotics. Springer, Secaucus, pp 1371–1394
8. Ude A (1993) Trajectory generation from noisy positions of object features for teaching robot paths. Robot Auton Syst 11(2):113–127
9. Yang J, Xu Y, Chen CS (1997) Human action learning via hidden Markov model. IEEE Trans Syst Man Cybern, Part A, Syst Hum 27(1):34–44
10. Yamane K, Nakamura Y (2003) Dynamics filter—concept and implementation of online motion generator for human figures. IEEE Trans Robot Autom 19(3):421–432

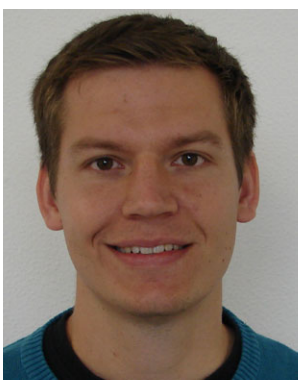
11. Ijspeert A, Nakanishi J, Schaal S (2002) Movement imitation with nonlinear dynamical systems in humanoid robots. In: IEEE international conference on robotics and automation, ICRA2002, pp 1398–1403
12. Ude A, Atkeson CG, Riley M (2004) Programming full-body movements for humanoid robots by observation. Robot Auton Syst 47:93–108
13. Asfour T, Azad P, Gyarfas F, Dillmann R (2008) Imitation learning of dual-arm manipulation tasks in humanoid robots. Int J Humanoid Robot 5(2):183–202
14. Calinon S, Guenter F, Billard A (2007) On learning, representing and generalizing a task in a humanoid robot. IEEE Trans Syst Man Cybern, Part B 37(2):286–298. Special issue on robot learning by observation, demonstration and imitation
15. Muench S, Kreuziger J, Kaiser M, Dillmann R (1994) Robot programming be demonstration (RPD)—using machine learning and user interaction methods for the development of easy and comfortable robot programming systems. In: Proceedings of international symposium on industrial robots, ISIR, pp 685–693
16. Friedrich H, Münch S, Dillmann R, Bocionek S, Sassin M (1996) Robot programming by demonstration (RPD): supporting the induction by human interaction. Mach Learn 23(2–3):163–189
17. Nicolescu M, Mataric M (2003) Natural methods for robot task learning: instructive demonstrations, generalization and practice. In: Proceedings of the second international joint conference on autonomous agents and multiagent systems, AAMAS, New York, NY, USA. ACM, New York, pp 241–248
18. Ekvall S, Kragic D (2006) Learning task models from multiple human demonstrations. In: IEEE international symposium on robot and human interactive communication, ROMAN, pp 358–363
19. Pardowitz M, Knoop S, Dillmann R, Zollner R (2007) Incremental learning of tasks from user demonstrations, past experiences, and vocal comments. IEEE Trans Syst Man Cybern, Part B, Cybern 37(2):322–332
20. Alissandrakis A, Nehaniv CL, Dautenhahn K (2007) Correspondence mapping induced state and action metrics for robotic imitation. IEEE Trans Syst Man Cybern, Part B, Cybern 37(2):299–307
21. Inamura T, Toshima I, Tanie H, Nakamura Y (2004) Embodied symbol emergence based on mimesis theory. Int J Robot Res 23(4–5):363–377
22. Jenkins OC, Mataric MJ (2004) Performance-derived behavior vocabularies: data-driven acquisition of skills from motion. Int J Humanoid Robot 1(2):237–288
23. Jenkins OC, Matarić MJ (2004) A spatio-temporal extension to Isomap nonlinear dimension reduction. In: Proceedings of the twenty-first international conference on machine learning, ICML, New York, NY, USA. ACM, New York, p 56
24. Lee D, Nakamura Y (2006) Stochastic model of imitating a new observed motion based on the acquired motion primitives. In: IROS, pp 4994–5000
25. Kadone H, Nakamura Y (2006) Segmentation, memorization, recognition and abstraction of humanoid motions based on correlations and associative memory. In: 6th IEEE-RAS international conference on Humanoid robots, 2006, pp 1–6
26. Kulic D, Takano W, Nakamura Y (2009) Online segmentation and clustering from continuous observation of whole body motions. IEEE Trans Robot 25(5):1158–1166
27. Aloimonos Y, Guerra G, Ogale A (2009) The language of action: a new tool for human-centric interfaces. In: Aghajan H, López-Cózar Delgado R, Augusto JC (eds) Human-centric interfaces for ambient intelligence. Elsevier, Amsterdam. ISBN 978-0-12-374708-2
28. Aloimonos Y (2009) Sensory grammars for sensor networks. J Ambient Intell Smart Environ 1(1):15–21. ISSN 1876-1364
29. POETICON. The poetics of everyday life: Grounding resources and mechanisms for artificial agents. http://www.poeticon.eu
30. Azad P, Ude A, Asfour T, Dillmann R (2007) Stereo-based markerless human motion capture for humanoid robot systems. In: IEEE international conference on robotics and automation, ICRA, Rome, Italy, April 2007, pp 3951–3956
31. Azad P, Asfour T, Dillmann R (2008) Robust real-time stereo-based markerless human motion capture. In: IEEE/RAS international conference on humanoid robots, Humanoids, Daejeon, Korea, December 2008, pp 700–707
32. Azad P (2008) Visual perception for manipulation and imitation in humanoid robots. PhD thesis, Universität Karlsruhe, TH, Karlsruhe, Germany
33. Deutscher J, Blake A, Reid I (2000) Articulated body motion capture by annealed particle filtering. In: IEEE computer society conference on computer vision and pattern recognition, CVPR, Hilton Head, USA, pp 2126–2133
34. Azad P, Ude A, Asfour T, Dillmann R (2007) Toward an unified representation for imitation of human motion on humanoids. In: IEEE international conference on robotics and automation, ICRA, Rome, Italy, April 2007, pp 2558–2563
35. Pastor P, Hoffmann H, Asfour T, Schaal S (2009) Learning and generalization of motor skills by learning from demonstration. In: Proceedings of the IEEE international conference on robotics and automation, Kobe, Japan
36. Rogalla O (2002) Abbildung von Benutzerdemonstrationen auf variable Roboterkonfigurationen. PhD thesis, Universität Karlsruhe
37. Arbib MA, Iberall T, Lyons D (1985) Coordinated control programs for movements of the hand. Exp Brain Res 10:111–129
38. Jaekel R, Schmidt-Rohr SR, Xue Z, Loesch M, Dillmann R (2010) Learning of probabilistic grasping strategies using programming by demonstration. In: IEEE international conference on robotics and automation, ICRA'10, May 2010
39. Calinon S, Billard A (2008) A probabilistic programming by demonstration framework handling constraints in joint space and task space. In: IEEE/RSJ intl conf. on intelligent robots and systems, IROS
40. Jaekel R, Schmidt-Rohr SR, Loesch M, Dillmann R (2010) Representation and constrained planning of manipulation strategies in the context of programming by demonstration. In: IEEE international conference on robotics and automation, ICRA'10, May 2010
41. Kasper A, Becher R, Steinhaus P, Dillmann R (2007) Developing and analyzing intuitive modes for interactive object modeling. In: ICMI'07: Proceedings of the nineth international conference on multimodal interfaces, New York, NY, USA. ACM, New York, pp 74–81
42. Becher R, Steinhaus P, Zöllner R, Dillmann R (2006) Design and implementation of an interactive object modelling system. In: Robotik/ISR, München, Germany, May 2006

Rüdiger Dillmann is full Professor at the Karlsruhe Institute of Technology, Institute for Anthropomatics, Karlsruhe, Germany. His research interests include robot programming by demonstration, machine learning, robot vision, humanoid robotics, cognitive cars, and medical applications of informatics.

Tamim Asfour is a senior researcher at the Karlsruhe Institute of Technology (KIT), Institute for Anthropomatics, Karlsruhe, Germany, where is leading the Humanoids Research Group. His main research interest is humanoid robotics.

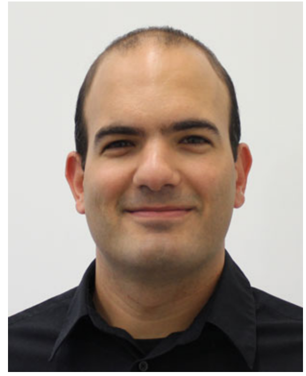


Martin Do is a research scientist at Karlsruhe Institute of Technology, Institute for Anthropomatics, Karlsruhe, Germany. His research focuses on imitation learning and perception of human manipulation activities.



Rainer Jäkel is a research scientist at Karlsruhe Institute of Technology, Institute for Anthropomatics, Karlsruhe, Germany. His research interests are robot programming by demonstration, constrained motion planning and probabilistic inference.

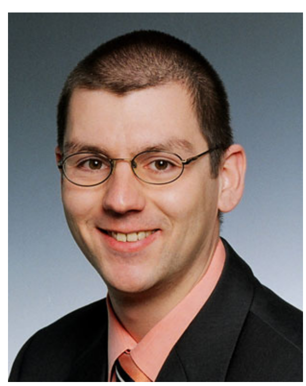
Alexander Kasper is a research scientist at Karlsruhe Institute of Technology, Institute for Anthropomatics, Karlsruhe, Germany. His research focuses on object modelling, knowledge representations for robots and scene visualisation.



Pedram Azad is a postdoctoral researcher at Karlsruhe Institute of Technology, Institute for Anthropomatics, Karlsruhe, Germany. His research focuses on robot vision, in particular visual perception for manipulation and imitation.

Aleš Ude is a senior research associate at the Department of Automatics, Biocybernetics, and Robotics, Jozef Stefan Institute, Ljubljana, Slovenia. His research focuses on imitation learning, perception of human activity, humanoid robot vision, and humanoid cognition.

Sven R. Schmidt-Rohr is a research assistant at Karlsruhe Institute of Technology, Institute for Anthropomatics, Karlsruhe, Germany. His research interests include robot programming by demonstration, machine learning and decision making for multi-modal service robots.



Martin Lösch is a research scientist Karlsruhe Institute of Technology, Institute for Anthropomatics, Karlsruhe, Germany. His research interests include recognition of human activity, machine learning and human robot interaction.