From Motion Capture to Action Capture: A Review of Imitation Learning Techniques and their Application to VR-based Character Animation

Bernhard Jung, Heni Ben Amor, Guido Heumer, Matthias Weber
VR and Multimedia Group
TU Bergakademie Freiberg
Freiberg, Germany
jung|amor|guido.heumer|matthias.weber@informatik.tu-freiberg.de

ABSTRACT

We present a novel method for virtual character animation that we call *action capture*. In this approach, virtual characters learn to imitate the actions of Virtual Reality (VR) users by tracking not only the users' movements but also their interactions with scene objects. Action capture builds on conventional motion capture but differs from it in that higher-level action representations are transferred rather than low-level motion data. As an advantage, the learned actions can often be naturally applied to varying situations, thus avoiding retargetting problems of motion capture. The idea of action capture is inspired by human imitation learning; related methods have been investigated for a longer time in robotics. The paper reviews the relevant literature in these areas before framing the concept of action capture in the context of VR-based character animation. We also present an example in which the actions of a VR user are transferred to a virtual worker.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; I.3.6 [**Computer Graphics**]: Methodology and Techniques—*Interaction techniques*; I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*Virtual Reality*

General Terms

Algorithms

Keywords

Virtual Reality, Motion Capture, Imitation Learning, Character Animation

1. INTRODUCTION

Since its inception, motion capture has proved to be a powerful and natural way of producing complex animations for virtual characters. Animations recorded from a human actor through motion capture are natural, lifelike, and contain even subtle details of human movement. Today, motion capturing is used in a variety of applications, ranging from virtual reality environments to video games and movies. In recent years, however, the virtual worlds found in such applications have become more and more complex, thus making various limitations of motion capture visible. For example, there is a current trend in the computer graphics community towards increased interactivity: virtual characters have to be able to react appropriately to sensations from their environment or the user. For this, the animations have to be changed at runtime; a process which is very difficult to do with motion capture data. A partial solution to this problem, motion graphs, is proposed in [13]. Another recent trend is the use of physical models in order to enrich the virtual worlds with a higher amount of realism and responsiveness. Again, online modifications of recorded motions are needed, in order to have the animation of a virtual character change according to the physical forces acting on it. One approach for combining physical models with motion capture data has been proposed in [29]. Recent movie productions also have started to use large groups of autonomous characters, each of which is equipped with a basic set of motion captured skills. However, these motions are not flexible enough in order to be changed according to the context they are triggered in. For example, a recorded motion for attacking a normal sized opponent might be rendered obsolete, if the current opponent is only half as tall.

A more flexible synthesis of motions can be achieved by means of behavioral animation techniques. Here, the animation is represented through a model or controller, instead of a set of motion data. Applying the controller in each simulation step yields the desired animation. Unfortunately, creating good controllers is often a difficult process based on complex search and optimization techniques. Additionally, by relying on such optimization or planning processes, the user sacrifices a large amount of control over the resulting animation. As a consequence, the animator is often required to put significant effort into parameter tweaking in order to achieve the desired end result of naturally appearing animations.

In this paper, we introduce a new approach to animation synthesis that we call *action capture*. Action capture is a VR-based method that not only tracks the user's movements but also his or her interactions with scene objects.

VRST'06, November 1–3, 2006, Limassol, Cyprus.
Copyright 2006 ACM 1-59593-321-2/06/0011 ...$5.00.

The goal of action capture is to combine the power of motion capture for creating natural and complex animations with the flexibility and dynamic responses of behavioral animation. To this end, *imitation learning* techniques are used in order to build faithful, yet situationally adaptable models of recorded movements that preserve interactions with scene objects under varying environmental conditions. Imitation learning has been investigated for a longer time in the cognitive and neuro sciences as well as in the fields of robotics and Artificial Intelligence, where related methods have been applied as a means of programming robots by demonstration. We review and discuss some of the relevant publications in these areas from a computer animation point of view. Building upon the results of this discussion, we present a basic framework for action capture and suggest possible extensions. Finally, we present first results from an ongoing research project in which manipulation tasks performed in VR are used to generate the animation of virtual humans.

2. MOTIVATION

Extending conventional motion capture, action capture aims to take advantage of increasingly available, complete VR systems for the purpose of virtual character animation. Similar to motion capture, the user's movements are recorded by means of position trackers and data-gloves. However, the recorded movements are abstracted to higher-level action representations that also account for user interactions with scene objects. Captured actions are later reproduced by virtual characters using behavioral animation techniques. The resulting animations naturally include interactions with scene objects. As further advantage, actions can be reproduced by virtual characters of different sizes and body proportions as well as in situations where the task environment differs from the original recording situation.

To illustrate the above advantages over conventional motion capture, consider the following example of a VR system for evaluating the interior design of a car's virtual prototype. To test the prototype's design, the user may interactively simulate the handling of the steering wheel, stick shift, radio controls, and other instruments. For ergonomic analyzes, these procedures are later to be repeated by virtual humans of different sizes and body proportions. Using conventional motion capture, the recorded movements would need *retargetting* to each differently sized virtual human; although powerful methods have been proposed for this retargetting step, e.g. based on spacetime constraints [28], this process still requires an explicit modeling effort by the human animator [9]. To complicate – and add realism to – the example, the ergonomic analyzes might further reveal a flaw in the car's interior design resulting e.g. in the repositioning of some instrument in the car's prototype. Again, the goal of action capture is to reuse the learned actions; i.e. data captured of the VR user e.g. twisting a knob in its original position should still be valid for synthesizing the virtual characters' animations when the knob's position is slightly altered in the next version of the virtual prototype. In the case of motion capture, the knob's repositioning would again require retargetting of the movement data; in contrast, in the case of action capture, a reusable, abstract animation command such as *twist(knob-1)* would play an integral role when synthesizing the desired animation. A prerequisite for action capture is thus a certain degree of autonomy of the virtual characters. Such animation methods have in recent years become popular in movies and interactive games under the name of *behavioral* or *interactive* animation [22, 27].

Note that humans are highly capable of transferring learned actions to slightly different task environments as detailed in the above example. The following section therefore reviews empirical findings on human imitation learning.

3. IMITATION IN HUMANS AND ANIMALS

According to Thorndike [26] imitation is: "from an act witnessed learn to do an act"[1]. Imitation is a powerful ability of humans and higher animals, which enables them to acquire new skills by observing actions of others. Young children for example are able to learn a number of social behaviors by observing their parents. Sport students can learn how to do a complex tennis serve by watching a teacher repeating it a few times. Imitation enables us to quickly acquire new skills without going through a lengthy trial-and-error process. As a result humans are very flexible and adaptive to changes in their environment. When confronted with new situations, environments or cultures we can rapidly learn a set of skills which might be crucial for progress or even survival. For the purpose of introducing the term *action capture* in later sections, we will focus on the imitation of 'actions'. Following Arbib's equation 'action = movement + goal / expectation' [3], with 'action' we refer to a goal-directed intentional motor behavior.

Bakker and Kuniyoshi [4] identify three requirements for the process of imitation:

1. **Observation** The action of a teacher is observed and processed.
2. **Representation** The action is represented through an internal model.
3. **Reproduction** Based on the internal model, the situation, and environment, an appropriate variant of the action is reproduced.

The process of observation involves an abstraction step, which can happen at different levels of cognitive complexity. For example, it might involve dissecting the seen action into simpler components which are part of the imitator's repertoire of skills. At a higher level of complexity, it involves analyzing the relevant environmental information accompanying the action, such as manipulated tools. At the highest level of complexity, observation also includes an act of 'understanding'. Here, the imitator infers the goals and intentions behind the perceived action. The observed action is then represented through an internal model. For this, a mapping from the teacher's body onto the student's body has to be applied. Each body part of the teacher has to be put in relation to the student's own body part, such that the internal model can afterwards be used to replicate the observed action. In the literature, this is known as the 'correspondence problem' [18].

Piaget's studies regarding imitation processes in children led to the distinction between two forms of imitation ([20], see also [4]): *conservative* and *true imitation.* Conservative imitation means imitating an action through already available behaviors. True imitation acts on a higher level and

[1] Although there exist more recent definitions in the literature, many of them are conflicting or even confusing.

generates new behaviors to accurately imitate actions with a greater understanding of what such actions are about.

Meltzoff and colleagues (see [16] and [21]) studied the imitative learning abilities of infants and came up with a four stage progression of imitative abilities. In the first stage, a so-called *body babbling* phase, the infant explores its body and learns how specific muscle movements achieve elementary body configurations. The result is an internal model of the infant's own body. Adding to this, the infant learns a set of motor primitives which can afterwards be connected to achieve complex movements. After the body babbling phase, the imitative abilities progress as follows:

1. **Imitation of Body Movements** In this stage, the infant uses it's body parts to imitate observed body movements or facial acts. First they activate the corresponding body part, then they correct their imitative response until they converge on the accurate match.

2. **Imitation of Actions on Objects** In this stage, infants learn to imitate the manipulation of objects which are external to their body. This includes playing with toys in a variety of contexts.

3. **Inferring Intentions** This is the highest form of imitative learning. It requires inferring the goals and intentions of the demonstrator from his observed behavior. In such a case, even an unsuccessful act can be correctly imitated.

In the *imitation of body movements* phase, the child imitates observed movements of a person by mapping them to one (or a set) of it's own motor primitives. Such a behavior is often called 'mimicry'; the mere reproduction of movements without having the same intention or goal. In the *imitation of action on objects* stage, the infant is able to learn manipulation tasks. In this stage, the imitative behavior becomes more goal-oriented. Although the child might not infer the purpose of the manipulation task, it is able to understand the basic steps involved and reproduce the task under different conditions. Obviously, this needs a more sophisticated internal model, in which relations between actions and objects are also stored. The internal model must also be complex enough to include models of the physics of passive objects. For example, a child might have to represent that bigger objects are typically heavier than smaller ones. Finally, in the last phase infants are able to understand seen actions and infer goals and intentions behind them. In such a case, the internal representation might only include the ends but not the means to achieve them. Representing an action through such high-level models enables for high ability of generalization. If the goal is clear, an action can be imitated even in a different context such as an unseen situation.

From the above, it becomes obvious that action understanding plays a vital role in imitation. A recent hypothesis in the neurophysiological literature suggests, that action understanding and action execution are based on a shared neural substrate. This, so called 'direct-matching hypothesis' was advanced by Rizzolatti and colleagues [23] as a consequence of the accumulating empirical evidence for the existence of mirror systems in humans and monkeys. Mirror systems or more specifically mirror neurons were first discovered in a sector of the premotor cortex of monkeys. Interestingly, these neurons fired both when the monkey saw another (living) individual performing a particular action and when it performed the action itself. For example, when seeing someone grasping food, the monkey activated the same neuron that would in another situation make it perform a grasp for food itself. This established for the first time a direct connection between action observation and action execution. It was also shown that mirror neurons only fire for goal-directed movements with a target. They do not respond to seeing objects alone or a mimicked action in absence of a target. This suggests that "the difference between imitation and understanding is that, in the case of imitation, the observed act is not only internally represented, but must also be externally manifested"[23]. Although this view is still debated in the neuroscience literature, it surely highlights the tight bonds between action understanding and imitation.

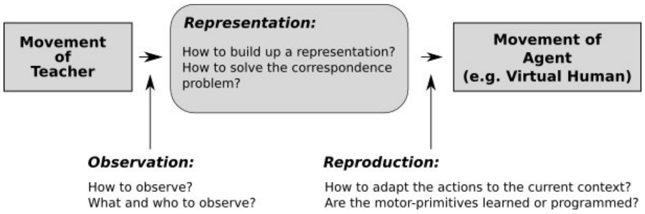
Movement of Teacher
Representation:
How to build up a representation?
How to solve the correspondence problem?
Movement of Agent (e.g. Virtual Human)
Observation:
How to observe?
What and who to observe?
Reproduction:
How to adapt the actions to the current context?
Are the motor-primitives learned or programmed?

Figure 1: The process of imitation

4. A REVIEW OF COMPUTATIONAL APPROACHES TO IMITATION LEARNING

Recent years have seen a growing interest in computational models of imitation learning, mainly in the field of robotics as a method of 'Programming by Demonstration (PbD)'; the edited collections of Dautenhahn & Nehaniv [7] and Billard & Siegwart [5] provide general overviews.

The following review of selected techniques for computational imitation learning is based on the work by Bakker and Kuniyoshi [4] and on our discussion in Section 3. The review is structured through raising the following questions: When observing, who and how should be observed? How are the observed actions represented? And how can the seen action be converted to the observer's actuators (correspondence problem)? When reproducing the actions, how to adapt the actions to the current context? And are the motor primitives used in actions learned or explicitly programmed? Figure 1 shows the process of imitation and the questions involved.

Further, w.r.t. the classes of imitative abilities identified by Meltzoff et al. [16], cf. Section 3: Does the observer imitate body movements (trajectories)? Do the imitated actions involve the manipulation of objects? And finally, does the observer infer the intentions and imitate the intentions behind an observed action?

In this section, we answer the above questions for a number of recent publications on imitation learning. The chosen publications reflect different approaches to the problem and can roughly be categorized in two groups: *biologically inspired approaches* and *engineering oriented approaches*.

4.1 Biologically Inspired Approaches

Mataric [14, 15] describes a biologically inspired, behavior-based approach for imitation learning in embodied agents,

Table 1: Comparison of different biologically inspired approaches to imitation.

	Mataric [14, 15]	**Rao et al. [21]**	**Oztop and Arbib [19]**
Observation			
-How?	2D vision, magnetic trackers, exoskeleton	synthetic vision	3D simulator, 2D vision
-What?	upper body postures of user	grid location of virtual teacher	hand state trajectory during grasping
Representation			
-How to represent?	sequences of visuo-motor primitives	forward/inverse models using state transition probabilities	neural network in core mirror system
-How to convert to observer's actuators?	classify as best matching basic behavior	computing state transition probabilities	matching in core mirror system
Reproduction			
-How to adapt to current context?	(not applicable)	Bayesian inference and probability maximization	neural networks reactive to visual input
-Learned or programmed motor primitives?	programmed / learned	programmed	programmed
Imitation of			
-Body Movements?	yes	yes	no
-Actions on Objects?	no	no	yes
-Inferred Intentions?	no	yes	no

which could be robots or virtual characters. The agents dispose of a pre-programmed – or alternatively, learned – set of basic behaviors called "perceptual-motor primitives" which serve to segment and continuously classify the movements of the human trainer as well as to generate the agent's movement. The primitives can further be parametrized with metric values such that they are sufficient, when properly combined, for generating a large range of complex behaviors. Imitation learning thus creates new skills as novel sequences and superpositions of classified primitives. As in motion capture, the goal of her imitation learning approach consists of the humanoid agents repeating the movements of the human trainer. An advantage of the behavior-based approach over motion capture is that it allows the robot a certain degree of freedom in the interpretation of the observed movement and thus naturally generalizes over varying body sizes. The reported setting is limited to the imitation of arm movements, tracked e.g. visually or using magnetic markers, and does not involve interactions with scene objects.

Rao and colleagues [21] propose a probabilistic framework in which they formalize Meltzoff and Moores's four-stage progression model of imitative abilities in infants [16] (see Section 3). They apply this model for learning how to solve a (simulated) maze task through imitation. In the body babbling phase, the imitating agent first wanders through the maze by performing random actions. The frequencies of the outcomes of each state-action pair are recorded in order to compute probabilities. These probabilities represent a *forward model* of the environment: a model predicting the next state of a system, given the current state and the action to be executed. Then, the imitator observes a sequence of states (grid positions) the teacher went through in order to solve the maze. From this sequence, a so called *inverse model* is learned: a model that tells what action to choose given the current state and the desired goal. Reproduction of the seen action is accomplished using probabilistic inference techniques. Using these techniques and the internal models (forward and inverse), the imitator is even able to infer the intent of the teacher. Due to the power of Bayesian inference in dealing with uncertainty, noise, and missing data, a high level of generalization can be achieved. However, due to the simplicity of the maze domain, imitation did not include complex interaction with scene objects. The primitive behaviors for moving in the maze such as `move north` were also pre-programmed.

Oztop and Arbib [19] present a detailed computational model of reaching and grasping that integrates many neurophysiological findings of the monkey's brain. In particular, they develop a computational account of the workings of the mirror neuron system, a brain region activated both when an action is observed and when the action is performed. Their computational model consists of three 'grand schemas': Schema 1, 'reach and grasp', takes input from the vision system, extracts the object position and object 'affordances' such as size; it further encapsulates motor programs to execute the grasp. Schema 2, 'visual analysis of hand state', also takes input from the vision system but is concerned with the extraction of 'hand state' information. The extracted hand state includes e.g. time based information about the distance and angle between hand and target object, wrist velocity, and several hand shape parameters such as aperture. Finally, schema 3, 'core mirror circuit', takes input from the two other schemas and is responsible for the generation of action code to control the grasping action. To test the model, they implemented a simple virtual environment where a stylized (monkey or human) arm simulates the grasping of an object. For further validation, they also describe experimental work with 2D vision input. Overall, the article focusses on the logic of the mirror neuron system rather than a more complete imitation system. Furthermore, actions other than grasping for which mirror system activity has been reported are not accounted for by the model.

Table 2: Comparison of different engineering oriented approaches to imitation.

	Aleotti et al.[1]	Buchsbaum&Blumberg [6]	Dillman et al. [8]
Observation			
-How?	data glove, magnetic trackers, 2D vision system	synthetic vision	data glove, force sensors, magnetic trackers, cameras
-What?	hand position and hand posture of user	positions of body parts of other virtual character	hand trajectory and hand posture of user
Representation			
-How to represent?	classes of pre-programmed tasks for movement synthesis	path through pose graph	tree structure of elementary actions
-How to convert to observer's actuators?	classify tasks	search for similar path in own graph	set of heuristic rules for sensor control
Reproduction			
-How to adapt to current context?	(not addressed)	(not addressed)	parametrization of pre-defined programs
-Learned or programmed motor primitives?	programmed	programmed	programmed
Imitation of			
-Body Movements?	no	yes	no
-Actions on Objects?	yes	yes	yes
-Inferred Intentions?	no	yes	no

4.2 Engineering Oriented Approaches

Aleotti et al. [1] use the Programming by Demonstration paradigm to instruct a robot in picking and placing objects in a scene. Instruction and first-time reproduction occur in a virtual environment, enriched with vibro-tactile feedback and visual fixtures. Observation of the user is achieved via a data glove and a magnetic position tracking system. The demonstrated action sequence is represented by instantiating a hierarchical structure of classes describing possible basic and high level tasks. Segmentation is achieved with a third-party software provided with the data glove. It can determine when grasps take place and when they stop. Using this representation they map the high level tasks to a robot and control it in a virtual environment. If the result is sufficient, the real robot repeats this action sequence. The pick-and-place tasks are pre-programmed. The mapping of actions from the human to the robot succeeds through high level representations that are independent of either agent's geometry. A prerequisite is of course that pre-programmed task models are available.

Dillman et al. [8] present an approach of PbD for humanoid service robots. The laying of a table is considered as example scenario, which involves mainly pick and place operations. User actions are tracked via a data glove, fitted with force sensors, and magnetic position trackers. Additionally an active trinocular camera head observes the user visually. Hand movements are segmented into elementary actions like moves, static and dynamic grasps, which are in turn chunked into semantically related groups, e.g. approach phase, grasp phase and release phase. Segmentation takes place in a two-phase process. In the first phase, the hand trajectory is segmented at times of contact between hand and object by analyzing the force values with a threshold based algorithm. In the second phase, where the actions during a grasp are segmented and analyzed, actions are classified into three classes of force value profiles by processing force sensor input: static grasps, external forces and dynamic grasps. The segmented elementary actions are then stored in a tree structure of operations. In turn to map this representation of operations to a robot, first the human grasp is mapped to robotic grippers (with less fingers) by "calculating an optimal group of coupled fingers", which exert a force in a common direction. To generate the arm trajectory, a set of logical rules for selecting sensor constraints, like force thresholds, etc. is used. The rules are based on the current context of operation like approach phase, grasp, retract phase, etc. The parameters generated by this mapping are then used to trigger and parametrize a pre-defined robot program, which, after a first step of testing the mapped trajectory, executes the movement in the real world. Overall this approach allows to instruct a robot to execute previously specified tasks in a specific way, as shown by the user. However, neither new movement or operation types can be learned, nor are inferences made about the intention of demonstrated tasks.

Buchsbaum and Blumberg apply imitation learning to a computer animation setting where one virtual character learns the actions of another one [6]. Two virtual articulated characters in the shape of mice observe each other through a simple form of synthetic vision. The characters perceive each other as a set of color-coded dots, which represent the position of the various body parts, like finger tips, nose, shoulders, etc. The motor system for animation of these figures is based on a multi-resolution variant of a standard motion-graph they call *posegraph.* This graph consists of a number of pre-defined poses as nodes with the edges defining valid transitions between poses. Basic movements are defined as paths through this graph. When one character observes the other's motion it perceives the global positions of several key body parts through the synthetic vision system. After transforming these absolute positions to body-root-relative positions the input pose is compared to the character's own known poses by use of a euclidean distance metrics. Movements are segmented at certain transitory poses, such as standing, which are assumed to always be taken between movements. Each observed movement is then compared to the character's own known movements by searching for a path through the posegraph with a minimal overall distance value. The movement represented by this path is then identified by the character as the observed movement. Reasoning about the intentions behind observed actions is enabled by a hierarchically organized action system, which is composed of individual behavior units, referred to as *action tuples*. Hereby an action consisting of one or more movements is annotated with trigger contexts, optional objects of attention and do-until context, that determine when the action finishes. Reasoning is again performed by searching a path through this hierarchy from a top-level motivation to the observed movements at the leaf nodes. Along the path the annotated constraints are attempted to be matched. This way a reasoning from observed movements towards the underlying motivations is possible. The downside of this model is, however, that everything from the behavior hierarchy to the identified poses is predefined by the developers. New movements could theoretically be identified and stored as new paths through the motion-graph. The basic poses the movement consists of, however, still have to be pre-defined. A further restriction lies in the restriction to characters with the same skeletal structure, with the correspondence of body parts of one character to another being hard-wired. It remains unclear how basic movements could be adapted to dynamic changes in the situation, e.g. the target object of a reach motion changing its position.

4.3 Related Work

Some other interesting work on imitation has been carried out by Ijspeert et al. [11]. Using a technique called *locally weighted regression* they approximate the trajectories of a human demonstrator. The approximated trajectory is stored as a set of nonlinear differential equations which form a control policy for a humanoid robot. The power of this approach was shown by having a humanoid robot with 30 degrees of freedom imitate a tennis forehand and backhand swing. The ALICE system proposed by Alissandrakis et al. [2] was also able to achieve imitation in robots. This system mainly focusses on solving the correspondence problem. To this end, a so called 'correspondence library', which maps actions, states and effects of a teacher to those of the imitator, was introduced. A generation mechanism proposes for each observed action a candidate corresponding action. Using a specified metrics, the system then decides whether to apply an action from the correspondence library or from the generation mechanism. In the case where the generation mechanism proposes a better corresponding action the library gets updated. In Nakanishi et al. [17] it was also demonstrated that imitation can be used to teach a biped robot complex locomotion and self stabilization skills. The learned locomotion controllers enabled the robot to walk over surfaces with different friction properties without loosing balance. In contrast to the mainly robotics centered research, the work of Kopp and Graeser [12] mainly focused on virtual embodied agents. They proposed a motor control framework which is based on a combination of coupled forward and inverse models, and graph-based representations. The framework enables the imitator to predict and execute the teacher's most probable next move by traversing a graph-based representation called 'motor command graph'.

4.4 Summary

In recent years an increasing body of research on imitation in artificial characters and robots has been published. Not all of the problems tackled in these publications are of interest for achieving *action capture*. Still, it becomes obvious from the above review that some common difficulties have been attacked and (partial) solutions proposed. Applications of these solutions on real-world robots show remarkable results. It also becomes obvious from the review, that different approaches to imitation exist. Often, however, the following assumptions can be found made in computational realizations of imitation learning: (1) The learning agent is equipped with a repertoire of primitive behaviors/motor skills/actions (2) Imitation involves finding new combinations or sequences of these primitive behaviors. (3) Imitated actions are represented through complex structures such as inverse models or plans. (4) The correspondence problem, i.e. the mapping of observed movements to the imitator's body is tackled through abstraction of movements to actions /behaviors to replicate the effects of actions rather than outer appearance of motion.

With *action capture* we aim at finding an approach to imitation which is particularly suitable for virtual reality and computer animation applications. With the hope of building on achievements of the community, we summarized in this section some of the influential papers on imitation. Tables 1 and 2 summarize in tabular form how each of the papers addressed the questions posed for the review.

5. ACTION CAPTURE

Based on the above discussion on previous work on imitation learning we are now in a position to frame an adaptation to virtual environments that we call *action capture*. One way to conceptualize action capture is as an extension of motion capture where the human trainer's performance is recorded not (only) at the lower level of movements but at the higher level of actions, particularly actions involving the manipulation of scene objects. Another way to conceptualize action capture is as a learning method for behavior-based virtual characters that empowers virtual characters to learn novel complex behaviors from basic behaviors by imitating a human trainer. Whichever conceptualization is chosen, a feature of action capture is that the learned actions / behaviors are inherently adaptable to situations that are similar but not necessarily identical to the learning situation.

In the following we will first present a basic framework for action capture. The basic framework can be seen as a translation of operational approaches to imitation learning in robotics to immersive VR; it is further restricted to actions on objects, i.e. manipulation tasks. We will then discuss possible choices of action representations and point out directions for extending the framework.

5.1 A Basic Framework for Action Capture

Action capture is a VR-based method for recording the actions of a human VR user and later reproducing these actions by virtual characters. In general, with 'action' we refer to any kind of intentional motor behavior. For the basic framework, we restrict 'actions' to manipulations of scene objects. Actions are decomposable into primitive actions which correspond to basic behaviors of the virtual characters. The *setting* for action capture thus consists of:

- Virtual environment: which supports its interactive manipulation by a human user. In particular, whenever the user manipulates a scene object, the virtual environment can detect this manipulation and generate a corresponding event, e.g., as simple cases, *pushed(button-1)* or *grasped(block-2)*.
- Human teacher: who performs an action or a sequence of actions in the virtual environment. The human teacher's actions are typically tracked using standard VR input devices such as position trackers and data gloves although in principle alternative methods e.g. based on visual input are also possible.
- Virtual character (learner): who observes the teacher's actions and learns to repeat them. The virtual character's body is assumed to be similar to the teacher's body, i.e. humanoid. This assumption ensures a more or less straightforward mapping of the teacher's body parts to the virtual character's body, thus simplifying the solution to the correspondence problem. The virtual character's body size and proportions may however differ from the human VR user. The virtual character further is equipped with a repertoire of *basic behaviors*, e.g. for pushing a button or grasping an object. These basic behaviors may be further parametrized, e.g. with a target position, a target object, or a hand shape to be assumed during a manipulation action.

The teacher's actions are tracked and abstracted to action representations that allow a later reproduction of the actions. The phases of action capture and reproduction are:

1. Action capture: during which the teacher's movements are tracked, segmented, classified as actions, and stored as high-level representations of the action or action sequence. The action representation should at least be expressed at the level of basic behaviors. Segmentation and classification of the teacher's movements can be informed by the events thrown by the VR system whenever interesting parts of a scene manipulation occur. Different choices of action representations are discussed below.
2. Action reproduction: where the action's representation is mapped to behaviors of the virtual character and the behaviors are executed.

5.2 Action Representations

The choice of the *action representation* in general depends on the task environment and expected degree of faithfulness of the animation; it is thus application-specific. However, the internal representation of the observed actions should at least be expressed at the level of basic behaviors, i.e. at a higher level than movement data. An action representation may however include movement data, e.g. default postures that parametrize actions/behaviors. The internal representation should also allow for sequences and hierarchical organization of actions/behaviors. Means for representing parallel actions are useful if e.g. two-handed manipulations are considered in the application scenario.

Concerning the parametrization of actions in the representation, varying degrees of abstraction can be imagined which will correspond to different generalization capabilities. For example, in a simple case, the virtual character may have to perform its actions in the same task environment, on identical objects, which are in a same or similar initial configuration as demonstrated before by the human VR user. Here, it may suffice to parametrize the action representation with an internal identifier of the manipulated object, e.g. *grasp(block-1)*. In more challenging task environments, the virtual character might be expected to repeat the demonstrated not necessarily on identical objects but on objects of the same type. Then the action representation might be parametrized with something like *grasp(block(green))* which will require some additional instantiation mechanism during action reproduction.

Another aspect in the choice of the action representation concerns a trade-off between generality and accuracy. At one end of the spectrum is *task-level imitation* where actions are just parametrized with their target objects. Task-level imitation is very general and relatively easy to adapt to novel situations but may result in movements which are unfaithful to the original movement. For example, the result of capturing a soccer player kicking the ball might be a very stiff and robot-like kick, when the action is applied on a virtual character. Important properties of the motion such as timing, grace, style, and realism might get lost when focusing on high generalization solely. Thus, when faithfulness of the animation is important, then it may be useful to parametrize the action representation further with movement and posture data. Movements and postures could be represented in a variety of ways, including joint angle values, trajectories, results from Principal Component Analyzes, contact points of hands with scene objects, symbolic descriptions of shape or movement, etc. Furthermore, one way to cope with the tradeoff between generalization and faithfulness of the animation could be the use of multi-level representations that specialize on different levels of detail. Representations at higher levels can achieve high generalization, while representations at lower levels can focus on including subtle features of the original movement into the action.

5.3 Extending the Basic Framework

In the basic framework of action capture, we restricted 'action' to the manipulation of scene objects from the more general view of 'action' as any kind of intentional motor behavior. The pragmatic reason for this limitation lay in a desired reduction in complexity and the fact that VR systems extend motion capture systems exactly with capabilities for processing interactions with scene objects. Using Meltzoff's et al. classification ([16], also see Sec. 3), other levels of imitation learning include the mimicking of body movements and imitation at the level of inferring intentions. To simply mimic the movements of a human teacher, standard motion capture equipment without immersion in an virtual environment suffices; however, to count as 'action capture', the recorded movements should be abstracted to the level of actions. One way of addressing the challenging problem of inferring the intentions behind seen actions could involve the integration of further task and domain knowledge into the action capture process.

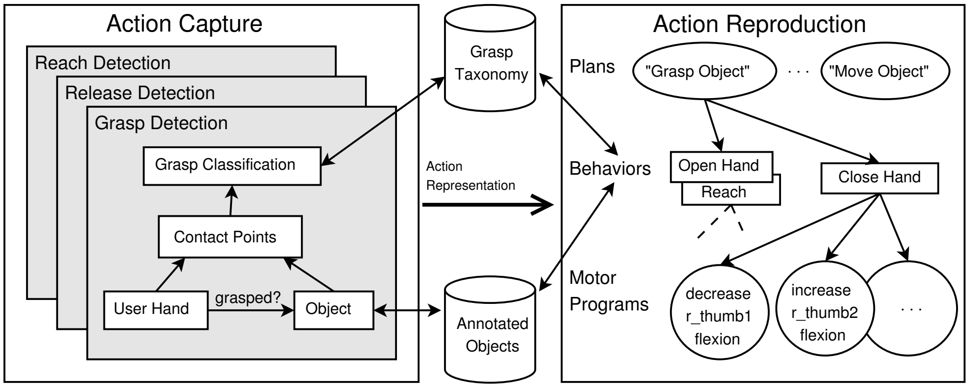

Figure 2: Illustration of Action Capture and Action Reproduction in a prototypical implementation

6. AN EXAMPLE: THE VIRTUAL WORKERS PROJECT

In the Virtual Workers project we aim at giving virtual humans the ability to learn and imitate object manipulations, particularly manipulations that involve different types of grasping. A major goal is to make the learned animations robust against dynamic scene changes such as repositioned or resized objects. In an example scenario, a virtual worker is located in a virtual environment that consists of various objects including a hammer, screwdrivers, etc. (see Figure 5). The user, who acts as teacher for the virtual worker, interacts with the environment on a wall-sized stereo projection. User movements are tracked by means of a 22 sensor data glove and an optical tracking system. For interaction with the virtual objects, a model of the user hand is projected into the virtual scene and checked for collisions with the virtual objects. The user can grasp objects and move them around or perform pre-defined functions on them, e.g. push a button, etc. These user actions are later reproduced by the virtual worker through means of action capture techniques.

Figure 2 illustrates the main components of the prototypical system architecture for action capture and reproduction. The action capture modules extract significant events from the continuous user interaction and generate higher level representations of the actions. Action reproduction is achieved through a multi-layer behavioral animation architecture. Action capture and reproduction modules as well as action representations may optionally refer to a grasp taxonomy which provides symbolic descriptions of hand shapes during grasping. Similarly, a database of object annotations may optionally be used that provides default information of typical grasp positions and orientations in an object-centered coordinate frame. These main components of the system architecture will now be described in more detail.

In the action capture phase, user interactions are first examined for relevant events. In the chosen scenario, these events correspond to grasp-related user actions and are generated e.g. when an object is grasped, released again, or when a reach motion is initiated. E.g. when a grasp is detected, first basic features of the user interaction are determined: tracking position and orientation values of the upper limbs, finger joint angle values, and collision points of the user's hands with the virtual objects. The grasp is then classified w.r.t. a grasp taxonomy to provide an abstract description of hand shape. In our current prototype, Schlesinger's grasp taxonomy [24] as summarized by Taylor and Schwarz [25] is used. In other work, we report how recognition of the Schlesinger grasps can be achieved reliably and in real-time even with uncalibrated data gloves [10]. Figure 3 gives an example of the various informations recorded in a grasp event. In the next step of motion analysis, grasp events are processed in order to dynamically generate an abstract action representation or *plan*. A plan is a specification of sequential or parallel actions that can be executed by the behavioral animation system used for action reproduction. In the current implementation, plans are generated via a simple template-based approach that maps

Figure 3: Example grasp event

```
<event-sequence>
  <event timestamp="3.895" type="grasp">
   <low-level>
    <joint-angle joint-id="r_index1">19.8306 -0.0865252
       0.678805 0.729203</joint-angle>
    <contact-point joint-id="sensor_r_index1">
      <object>Ball-1</object>
      <pos>0.00730081 -0.0734563 0.0135953</pos>
    </contact-point>
    <object-ids> Ball-1 </object-ids>
    <hand-transform>0.0205884 0.211408 -0.97718 0
                    0.0805939 0.973855 0.212386 0
                    0.996533 -0.0831275 0.00301189 0
                    -0.1502 -0.626599 0.917001 1
    </hand-transform>
    <hand>right</hand>
   </low-level>
   <high-level>
    <taxonomy>schlesinger</taxonomy>
    <category>spherical</category>
   </high-level>
  </event>
  ...
</event-sequence>
```

events to actions. The example plan in Figure 4 specifies the actions to replicate a complex grasping action. The plan describes the parallel execution of two behaviors for reaching and opening of the hand; then follows a hand closing behavior that results in a spherical grasp of the scene object.

Figure 4: Example of a dynamically generated plan: grasping an object

```
<plan type="captured">
  <parallel>
    <behavior>
      <type>Reach</type>
      <param name="object">Ball-1</param>
    </behavior>
    <behavior>
      <type>GraspOpen</type>
      <param name="preshape">spherical</param>
    </behavior>
  </parallel>
  <behavior>
    <type>GraspClose</type>
    <param name="grasp-type">spherical</param>
  </behavior>
</plan>
```

Reproduction of captured actions through the virtual worker is achieved via a multi-level behavioral animation approach. At the highest level, predefined plans describe the decomposition of complex animations into combinations of primitive actions. Predefined plans contain useful, often reoccurring action sequences, such as grasping an object, pick and place operations, etc. Primitive actions are executed by parametrizable mid-level behaviors, such as reaching a goal position, closing or opening the hand, etc. Behavior execution integrates collision sensor feedback, such that e.g. a hand closing behavior stops when fingers make contact with an object. Finally, low-level motor programs are responsible for controlling body movements.

The dynamically generated plans resulting from the action capture phase must be composed of actions which can be reproduced by the virtual worker. I.e. each action in a captured plan must correspond to either a behavior or a predefined plan. Captured plans can however combine the predefined plans and behaviors in novel ways. In this way, action capture enables the virtual agent to learn new tasks as novel action combinations from a built-in behavior repertoire.

Note that in the particular plan of Figure 4, the grasp action is specified in a comparatively abstract manner. Detailed information about hand position and orientation, joint angles, contact points, and timing available in the grasp events are deliberately missing in the plan. Through this, plan execution becomes robust against certain variations in the scene configuration such as changes to object location and size as well as differently sized virtual humans. Alternatively, the plan might have included such detailed information – at the cost of robustness against scene alterations but possibly gaining more accuracy w.r.t. the style of the original grasping action of the user. The deep exploration of this design space of action representation, and more generally, action capture is topic of on-going and future work.

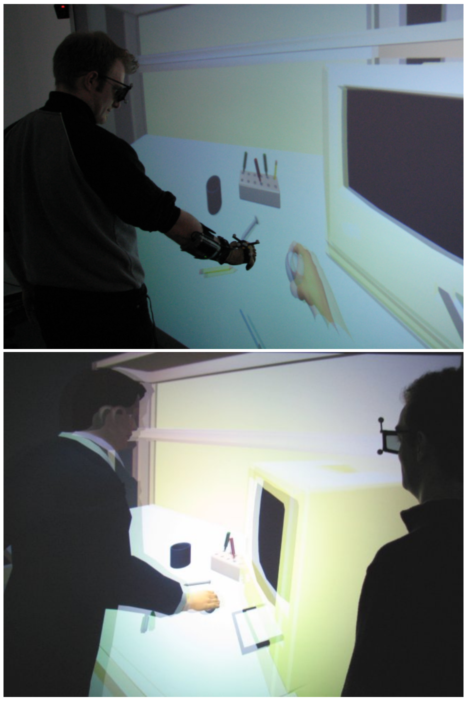

Figure 5: User demonstrating a grasp and the virtual human imitating it

7. CONCLUSION

In this paper, we have motivated and introduced an extension to motion capture, called *action capture*, which has its roots in imitation learning. Action capture aims at recording flexible, responsive, and adaptable animations suitable for interactive and physics-based virtual worlds. In particular, it extends motion capture systems with capabilities for processing interactions with scene objects.

We first presented previous work on imitation learning, which then helped to specify a basic framework of required features. We also described an extension to the basic framework, which includes additional, more complex features. Finally, we presented the ‘Virtual Workers’ project: an example application in which virtual agents learn manipulation tasks through action capture.

One of the main goals of this paper was to show that imitation can be a powerful tool for animating virtual characters. Although there exists already a variety of publications on imitation learning, much of the published work focuses on topics such as robotics or biological models. By introducing a special framework for action capture, we limited the research questions to the ones which are of interest to the VR community. We believe that action capture will prove particularly beneficial in virtual prototyping settings that require the automated generation of animations for many variants of prototypes and virtual humans.

8. ACKNOWLEDGMENTS

The research described in this contribution is partially supported by the DFG (Deutsche Forschungsgemeinschaft) in the Virtual Workers project.

9. REFERENCES

[1] J. Aleotti, S. Caselli, and M. Reggiani. Leveraging on a Virtual Environment for Robot Programming by Demonstration. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems IROS 2003, Workshop on Robot Programming by Demonstration, Las Vegas (USA)*, 2003.

[2] A. Alissandrakis, C. L. Nehaniv, and K. Dautenhahn. Imitation with ALICE: Learning to Imitate Corresponding Actions Across Dissimilar Embodiments. *IEEE Trans. Systems, Man and Cybernetics*, 32(4):482–296, 2002.

[3] M. A. Arbib. The mirror system, imitation, and the evolution of language. In Dautenhahn and Nehaniv [7].

[4] P. Bakker and Y. Kuniyoshi. Robot see, Robot do: An Overview of Robot Imitation. In *AISB96 Workshop: Learning in Robots and Animals*, pages 3–11, 1996.

[5] A. Billard and R. Siegwart, editors. *Special Issue on Robot Learning from Demonstration*, volume 47 of *Robotics and Autonomous Systems*, 2004.

[6] D. Buchsbaum and B. Blumberg. Imitation as a First Step to Social Learning In Synthetic Characters: A Graph-based Approach. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 9–18, New York, NY, USA, 2005. ACM Press.

[7] K. Dautenhahn and C. Nehaniv, editors. *Imitation in Animals and Artifacts*. MIT Press, 2002.

[8] R. Dillmann, M. Ehrenmann, P. Steinhaus, O. Rogalla, and R. Zöllner. Human Friendly Programming of Humanoid Robots: The German Collaborative Research Center. In *IARP 2002*, 2002.

[9] M. Gleicher. Retargetting Motion to New Characters. In *SIGGRAPH'98 Conference Proceedings*, Computer Graphics Annual Conference Series, pages 33–42. ACM, 1998.

[10] G. Heumer, H. Ben Amor, M. Weber, and B. Jung. Calibration-free Recognition of Grasp Types - A Comparison of Classification Methods. In *Proceedings Dritter Workshop Virtuelle und Erweiterte Realität der GI-Fachgruppe VR/AR*, 2006.

[11] A. Ijspeert, J. Nakanishi, and S. Schaal. Movement Imitation with Nonlinear Dynamical Systems in Humanoid Robots. In *IEEE International Conference on Robotics and Automation*, 2002.

[12] S. Kopp and O. Graeser. Imitation Learning and Response Facilitation in Embodied Agents. In *Intelligent Virtual Agents 2006*, LNAI, pages 28–41. Springer-Verlag, Berlin, 2006.

[13] L. Kovar, M. Gleicher, and F. Pighin. Motion Graphs. *ACM Transactions on Graphics*, 21(3), 2002.

[14] M. Mataric. Getting Humanoids to Move and Imitate. *IEEE Intelligent Systems*, pages 18–24, July/August 2000.

[15] M. Mataric. Sensory-motor primitives as a basis for learning by imitation: Linking perception to action and biology to robotics. In Dautenhahn and Nehaniv [7].

[16] A. N. Meltzoff. The Human Infant as Imitative Generalist: A 20-year Progress Report on Infant Imitation with Implications for Comparative Psychology. In *Social Learning in Animals: The Roots of Culture*, pages 347–370, 1996.

[17] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal, and M. Kawato. Learning from Demonstration and Adaptation of Biped Locomotion. *Robotics and Autonomous Systems*, 47(2-3):79–91, 2004.

[18] C. Nehaniv and K. Dautenhahn. The Correspondence Problem. In Dautenhahn and Nehaniv [7], pages 41–61.

[19] E. Oztop and M. Arbib. Schema design and implementation of the grasp-related mirror neuron system. *Biological Cybernetics*, 87:116–140, 2002.

[20] J. Piaget. *Play, Dreams and Imitation in Childhood*. New York: W. W. Norton, 1962.

[21] R. Rao, A. P. Shon, and A. N. Meltzoff. A Bayesian Model of Imitation in Infants and Robots. In *Imitation and Social Learning in Robots, Humans and Animals: Behavioural, Social and Communicative Dimensions*, 2004.

[22] C. Reynolds. Flocks, Herds and Schools: A Distributed Behavioural Model. *Computer Graphics*, 21(4):25–34, 1987.

[23] G. Rizzolatti, L. Fogassi, and V. Gallese. Neurophysiological Mechanisms Underlying the Understanding and Imitation of Action. *Nature Reviews Neuroscience*, pages 661–770, September 2001.

[24] G. Schlesinger. Der Mechanische Aufbau der Künstlichen Glieder. In M. Borchardt et al., editors, *Ersatzglieder und Arbeitshilfen für Kriegsbeschädigte und Unfallverletzte*, pages 321–661. Springer-Verlag: Berlin, Germany, 1919.

[25] C. Taylor and R. Schwarz. The Anatomy and Mechanics of the Human Hand. *Artificial Limbs*, 2:22–35, 1955.

[26] E. L. Thorndike. Animal Intelligence: An Experimental Study of the Associative Processes in Animals. *Psychological Review Monographs*, 8, 1898.

[27] B. Tomlinson. From Linear to Interactive Animation: How Autonomous Characters Change the Process and Product of Animating. *ACM Computers In Entertainment*, 3(1), 2005.

[28] A. Witkin and M. Kass. Spacetime Constraints. In *SIGGRAPH'88 Conference Proceedings*, volume 22, pages 159–168. ACM, 1988.

[29] V. B. Zordan, A. Majkowska, B. Chiu, and M. Fast. Dynamic Response for Motion Capture Animation. *ACM Transactions on Graphics*, 24(3):697–701, 2005.