On the Representation, Learning and Transfer of Spatio-Temporal Movement Characteristics

Winfried Ilg*, Gökhan. H. Bakır‡, Johannes Mezger♮, and Martin A. Giese*

*Laboratory for Action Representation and Learning
Dept. for Cognitive Neurology, University Clinic Tübingen, Germany
‡Max Planck Institute for Biological Cybernetics
Tübingen, Germany
♮ Graphical-Interactive Systems, Wilhelm Schickard Institute for Computer Science
University Tübingen, Germany
{winfried.ilg,gb,martin.giese}@tuebingen.mpg.de
jmezger@gris.uni-tuebingen.de

Abstract. In this paper we present a learning-based approach for the modeling of complex movement sequences. Based on the method of Spatio-Temporal Morphable Models (STMMs) [12] we derive a hierarchical algorithm that, in a first step, identifies automatically movement elements in movement sequences based on a coarse spatio-temporal description, and in a second step models these movement primitives by approximation through linear combinations of learned example movement trajectories. We describe the different steps of the algorithm and show how it can be applied for modeling and synthesis of complex sequences of human movements that contain movement elements with variable style. The proposed method is demonstrated on different applications of movement representation relevant for imitation learning of movement styles in humanoid robotics.

1 Introduction

The development of efficient representations of complex movements is an important problem in many technical disciplines, such as computer vision, computer graphics, robotics, sports, and medical diagnosis. For several applications it is crucial to learn such representations based on small amounts of training data. This requires learning techniques, that approximate whole classes of movement sequences with a small set of example sequences. One method that fulfills this requirement and seems which is suitable for both, the synthesis and analysis of movements with different spatio-temporal characteristics is the linear combination of movements. Such linear combinations can be defined on the basis of spatio-temporal correspondence. The technique of Spatio-Temporal Morphable Models (STMMs) defines linear combinations by weighted summation of spatial and temporal displacement fields that morph the combined prototypical movement into a reference pattern [12]. Interpolation based on spatio-temporal correspondence has been successfully applied for motion morphing in computer graphics [36, 6, 38] , and for the recognition and synthesis of periodic gait patterns in computer vision [12]. Most existing interpolation approaches (see also

[13, 28]) are restricted to individual short movements (e.g. one gait cycle, or a single arm movement), and require previous segmentation of the action stream. It seems desirable to define linear combinations for much more complex trajectories in order to model e.g. sequences of movements in sports, or an action sequence of a humanoid robot with different styles.

In this paper we present a hierarchical algorithm that makes STMM applicable to complex motion sequences by introducing a second hierarchy level that represents motion primitives. Each movement primitive is modeled by a STMM. In this way a generative model of complex sequences of acyclic movements with variable styles can be learned from few example trajectories. This representation is suitable for analysis and synthesis, and can thus be applied for imitation learning o complex movement sequences.

In the following we first describe the algorithm that consists of three steps: 1) the automatic identification of movement primitives, and 2) their approximation by STMM, and (3) the automatic concatenation of the modeled movement elements into a smooth trajectory. We then show three applications of this algorithm. In the first example we show the capability of the method to synthesize a parameterized spectrum of human walking styles based on a few recorded movement prototypes. Second, we describe the automatic synthesis of movement sequences with complex spatio-temporal structure modeling technique sequences (katas) from martial arts. Finally, we demonstrate the application of the method in the context of imitation learning in robotics: the imitation of different styles of human writing movements on a 7-DOF robot arm.

2 Hierarchical Spatio-Temporal Morphable Models (HSTMM)

The three steps of the algorithm for establishing spatio-temporal correspondence between complex movement sequences are illustrated in figure 1. More details are given in the following sections.

2.1 Representation of Key Features for Movement Primitives

The decomposition of complex movement sequences into movement primitives has been discussed as a basic principle of perception and action in a number of different fields, including neuroscience [24], [32], [8], [30], rehabilitation [27] and imitation learning in robotics [2],[21],[31].

For the selection of adequate movement primitives in technical applications it is important to consider the perception as well as the generation of movements. For perception the most important criterion is the robust identification of these primitives. For the generation of movement sequences movement primitives should define generic building blocks that encode a range of similar stereotypical movements. For our method a movement primitive is defined by the property that the style of the movement stays constant within the primitive.

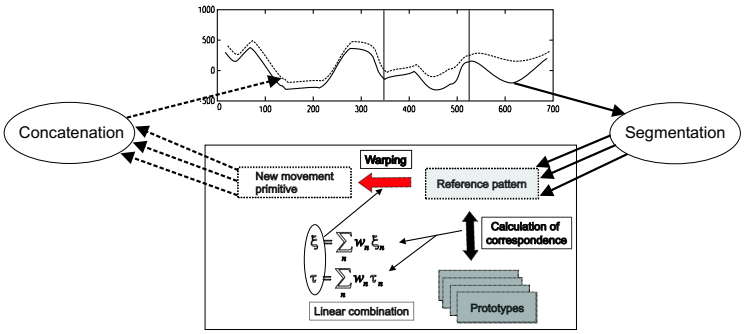

Fig. 1. Schematic description of the algorithm for analyzing and synthesis of complex movement sequences. In the first step the sequence is decomposed into movement primitives. These movement primitives can be analyzed and changed in style by approximating them by linear combinations of prototypes with different linear weight combinations. Afterward, the individual movement primitives are concatenated again into one movement sequence. The result is a new motion sequence comprising movement elements with different styles.

The identification of movement primitives within movement sequences must be based on characteristic features that are suitable for a robust and fast segmentation. Different elementary spatio-temporal and kinematic features have been discussed in the literature, like angular velocity [10][23] [25], or curvature and torsion of the 3D trajectories [7]. The key features of our algorithm are zeros of the velocity in a few "characteristic coordinates" of the trajectory $\boldsymbol{\zeta}(t)$. These features provide a coarse description of the spatio-temporal characteristics of trajectory segments. To localize movement primitives in sequences these characteristics are matched to previously stored templates of prototypical movement primitives. This matching is accomplished by dynamic programming (section 2.2). Each feature corresponds to a discrete event. Let m be the number of the motion primitive and r the number of characteristic coordinates of the trajectory. Let $\boldsymbol{\kappa}(t)$ be the "reduced trajectory" including only the characteristic coordinates that has the values $\boldsymbol{\kappa}_i^m$ at the velocity zeros[1]. The movement primitive is then characterized by the vector differences $\Delta\boldsymbol{\kappa}_i^m = \boldsymbol{\kappa}_i^m - \boldsymbol{\kappa}_{i-1}^m$ between subsequent velocity zeros (figure 2).

2.2 Identification of Movement Primitives

A robust identification of movement primitives in noisy data, allowing for additional or missing zero-velocity points κ_i^s, can be accomplished by dynamic programming. The result is an optimal sequence alignment between the key features of the prototypical movement primitive $\kappa_0^m \ldots \kappa_q^m$ and the key features of the search window $\kappa_0^s \ldots \kappa_p^s$ (see figure 2b). Dynamic programming is used to minimize a cost function that is given by the sum of $||\Delta\kappa_i^s - \Delta\kappa_j^m||$ over all matched key features. Robustness against additional and missing key features is

[1] Zero-velocity is defined by a zero of the velocity in at least one coordinate of the reduced trajectory.

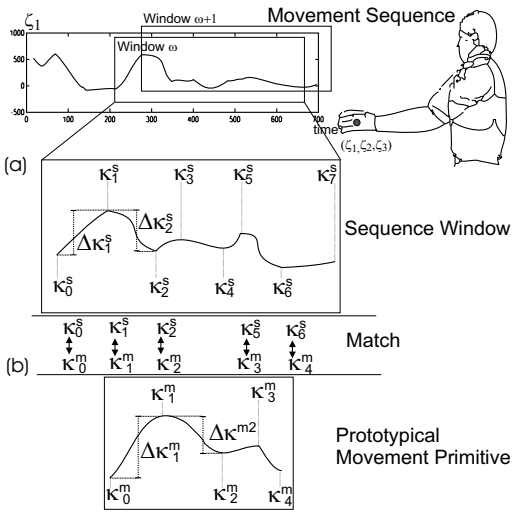

Fig. 2. The method for the automatic identification of movement primitives: (a) In a first step all key features $\boldsymbol{\kappa}_i^s$ are determined. (b) Sequences of key features from the sequences (s) are matched with sequences of key features from the prototypical movement primitives (m) using dynamic programming. A search window is moved over the sequence. The window contains two times the number of key features of the learned motor primitive. The best matching trajectory segment is defined by the sequence of feature vectors that minimizes $\sum_j ||\Delta\boldsymbol{\kappa}_i^s - \Delta\boldsymbol{\kappa}_j^m||$ over all matched key features. The method establishes a spatio-temporal correspondence at a coarse level.

achieved by appropriately constraining the set of admissible transitions for the dynamic programming as follows:

For the match of two successive key features κ_i^m and κ_{i+1}^m it is possible to skip up to two key features κ_{i+1}^s and κ_{i+2}^s in the sequence. In this case, i.e. the match $\kappa_i^s \rightarrow \kappa_j^m$ and $\kappa_{i+3}^s \rightarrow \kappa_{j+1}^m$ would be realized. Furthermore, it is possible to skip one key feature κ_j^m in the movement primitive. This implies that the match $\kappa_i^s \rightarrow \kappa_{j-1}^m$ and $\kappa_{i+1}^s \rightarrow \kappa_{j+1}^m$ is valid. The cost function that is minimized in order to find an optimal match between $\kappa_0^s \ldots \kappa_i^s$ and $\kappa_0^m \ldots \kappa_j^m$ under the given constraints can be written recursively:

$$\begin{aligned} D(i,j) = min(&D(i-1,j-1) + ||\Delta\boldsymbol{\kappa}_i^s - \Delta\boldsymbol{\kappa}_j^m||, \\ &D(i-2,j-1) + ||\Delta\boldsymbol{\kappa}_{[i-1,i]}^s - \Delta\boldsymbol{\kappa}_j^m||, \\ &D(i-3,j-1) + ||\Delta\boldsymbol{\kappa}_{[i-2,i]}^s - \Delta\boldsymbol{\kappa}_j^m||, \\ &D(i-1,j-2) + ||\Delta\boldsymbol{\kappa}_i^s - \Delta\boldsymbol{\kappa}_{[j-1,j]}^m||, \\ &D(i-2,j-2) + ||\Delta\boldsymbol{\kappa}_{[i-1,i]}^s - \Delta\boldsymbol{\kappa}_{[j-1,j]}^m||, \\ &D(i-3,j-2) + ||\Delta\boldsymbol{\kappa}_{[i-2,i]}^s - \Delta\boldsymbol{\kappa}_{[j-1,j]}^m||) \end{aligned} \tag{1}$$

where i and j denote the indexes for the key feature κ_i^s respectively κ_j^m. The starting value $D(1,1)$ is given by

$$D(1,1) = ||\Delta\boldsymbol{\kappa}_1^s - \Delta\boldsymbol{\kappa}_1^m|| \tag{2}$$

where $\Delta\boldsymbol{\kappa}_1^s$ is the first difference vector of the sequence window and $\Delta\boldsymbol{\kappa}_1^m$ is the first difference vector of the movement primitive m. If one or more key

features are skipped for the match, the difference vector $\Delta\kappa_i^s$ between successive key features must be adapted. An example is described in eq. (3) for the case that two key features κ_{i-2}^s and κ_{i-1}^s are skipped. The resulting difference vector between the two successive key features κ_{i-3}^s and κ_i^s is determined by

$$\Delta\kappa_{[i-2,i]}^s = \Delta\kappa_{i-2}^s + \Delta\kappa_{i-1}^s + \Delta\kappa_i^s \tag{3}$$

Let be p the number of key features in window ω, and q the number of key features in moment primitive m. To determine the best match between movement primitive m and the sequence window ω one has to find the key feature κ_k^s, $0 \leq k \leq p$, for which the cost function for matching the sequences is minimal. The minimum cost δ of movement primitive m for window ω is given by

$$\delta(m_\omega) = \min_k(D(k,q)). \tag{4}$$

A concrete example of the application of the segmentation algorithm is discussed in section 4.

2.3 Morphable Models as Movement Primitives

The technique of *Spatio-Temporal Morphable Models* [11, 12] is based on linearly combining prototypical movement trajectories. Linear combinations of movement patterns are defined on the basis of spatio-temporal correspondences that are computed by dynamic time warping using dynamic programming [6]. Complex movement patterns can be characterized by trajectories of feature points. The trajectories of the prototypical movement pattern p can be characterized by the time-dependent vector $\boldsymbol{\zeta}_p(t)$. The correspondence field between two trajectories $\boldsymbol{\zeta}_1$ and $\boldsymbol{\zeta}_2$ is defined by the spatial shifts $\boldsymbol{\xi}(t)$ and the temporal shifts $\tau(t)$ that transform the first trajectory into the second. This warping transformation is specified mathematically by the equation:

$$\boldsymbol{\zeta}_2(t) = \boldsymbol{\zeta}_1(t + \tau(t)) + \boldsymbol{\xi}(t) \tag{5}$$

By linear combination of spatial and temporal shifts it is possible to to interpolate smoothly between classes of motion patterns with significantly different spatial structure, but also between patterns that differ with respect to their timing.

The correspondence algorithm determines the temporal and spatial shifts by minimizing the weighted sum of the quadratic spatial and temporal displacements over the whole movement sequence. In the time-continuous case, this error is given by the integral:

$$E_c[\boldsymbol{\xi}, \tau] = \int \left[|\boldsymbol{\xi}(t)|^2 + \lambda\,\tau(t)^2\right] \mathrm{d}t \tag{6}$$

The error has to be minimized under the additional constraint that the mapping between the time variable t and the modified time $t' = t + \tau(t)$ for the trajectory $\zeta_1(t')$ must be continuous, one-to-one, and monotonically increasing, in order to define unique temporal warping of the sequence ζ_1. Our correspondence algorithm consists of two steps. The first step solves a discrete optimization problem on the temporally sub-sampled trajectory by dynamic programming. In

the second step, the obtained solution is refined by solving a continuous optimization problem that is derived by linear interpolation between the sampling points, resulting in a set of quasi-continuous spatial and temporal shifts. For further details we refer to [11][12].

Signifying the spatial and temporal shifts between prototype p and a reference trajectory[2] $\zeta_0(t')$ by $\boldsymbol{\xi}_p(t)$ and $\tau_p(t)$, linearly combined spatial and temporal shifts can be defined by the two equations:

$$\boldsymbol{\xi}(t) = \sum_{p=1}^{P} w_p \, \boldsymbol{\xi}_p(t) \qquad \tau(t) = \sum_{p=1}^{P} w_p \, \tau_p(t) \tag{7}$$

The weights w_p define the contributions of the individual prototypes to the linear combination. We always assume convex combinations with $0 \leq w_p \leq 1$ and $\sum_p w_p = 1$. After linearly combining the spatial and temporal shifts the trajectories of the morphed pattern can be recovered by morphing the reference trajectory $\zeta_0(t')$ in space-time using the spatial and temporal shifts $\boldsymbol{\xi}(t)$ and $\tau(t)$. The space-time morph is defined by equation (5) where $\boldsymbol{\zeta}_1$ has to be identified with the reference trajectory and $\boldsymbol{\zeta}_2$ with the resulting space-time morph.

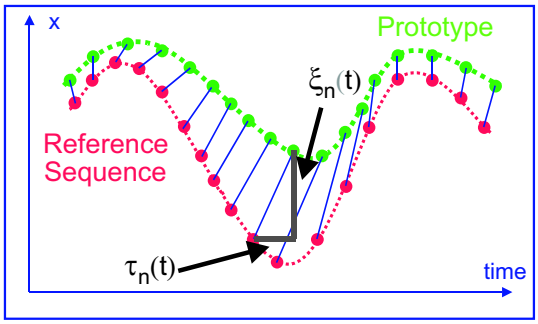

Fig. 3. Illustration of the established spatio-temporal correspondence between a prototypical trajectory and a reference sequence with the correspondence vector fields $\tau(t)$ and $\xi(t)$.

2.4 Linear Combination of Movement Trajectories with Multiple Movement Elements

The method presented in this paper allows to model movement sequences that consist of multiple movement elements with different styles. Figure 4 shows an example with 10 movement elements. The solid lines show one coordinate of two prototypical trajectories. The dashed lines illustrate two different linear combinations. One linear combination (LC 1) is obtained by combining the movement elements of the two prototypes using the same linear weights 0.5 for all movement elements. The second linear combination (LC 2) combines movement elements

[2] The reference trajectory is typically the average of the time-normalized training trajectories.

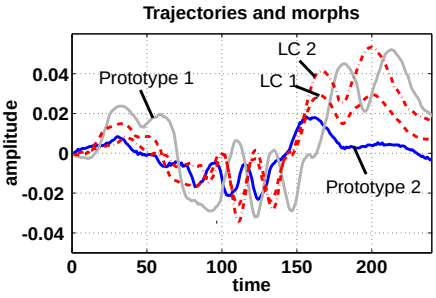

Fig. 4. Prototypical trajectories and linear combination.

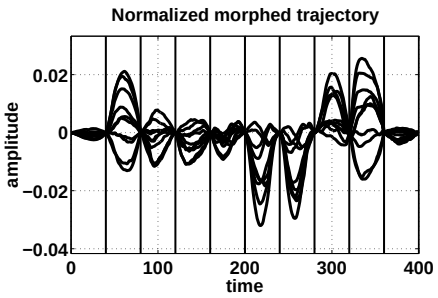

Fig. 5. Normalized linearly combined trajectory.

with different styles. The first 5 movement elements follow the style of the second prototype, corresponding to a weight vector $\boldsymbol{\omega} = [0, 1]$. The second 5 elements follow the first prototype with weight vector $\boldsymbol{\omega} = [1, 0]$.

In order to synthesize such complex motion sequences automatically the movement elements, modeled by STMM, must be concatenated into continuous longer sequences, avoiding discontinuities and artifacts at the boundaries between the elements. We have developed an algorithm that accomplishes this goal with the following four steps:

1. **Decomposition of the trajectories in movement elements:** The result of the segmentation step (section 2.2) are L starting points $T^s_{p,l}$, and end points $T^e_{p,l}$ of the individual movement elements with $1 \leq l \leq L$.
2. **Normalization of the movement elements:** After separation into trajectory segments $\boldsymbol{\zeta}_{p,l}(t)$ with boundaries that are given by the start and end times $T^{s,e}_{p,l}$ each segment is re-sampled with a fixed number of time steps in an interval $[0, T_{\text{seg}}]$. In addition, a linear function in time is subtracted from the original trajectory segments resulting in a modified trajectory that is always zero at the transition points between subsequent movement elements. This makes it possible to concatenate the segments with different styles without inducing discontinuities in the trajectories. With the start and endpoints $\boldsymbol{\zeta}^{\text{s}}_{p,l} = \boldsymbol{\zeta}_{p,l}(T^{\text{s}}_{p,l})$ and $\boldsymbol{\zeta}^{\text{e}}_{p,l} = \boldsymbol{\zeta}_{p,l}(T^{\text{e}}_{p,l})$ the normalized trajectory segments can be written after re-sampling:
$$\tilde{\boldsymbol{\zeta}}_{p,l}(t) = \boldsymbol{\zeta}_{p,l}(t) - \boldsymbol{\zeta}^{\text{s}}_{p,l} - (t/T_{\text{seg}})(\boldsymbol{\zeta}^{\text{e}}_{p,l} - \boldsymbol{\zeta}^{\text{s}}_{p,l}) \tag{8}$$
3. **Linear combination of the elements:** For each trajectory segment the movements are linearly combined separately. The normalized trajectory segments $\tilde{\boldsymbol{\zeta}}_{p,l}(t)$ are combined using STMMs in the way described in section 2.3 using the linear weights $\omega_{p,l}$. The result are the linearly combined normalized trajectory segments $\tilde{\boldsymbol{\zeta}}_l(t)$. The start and end points are also linearly combined, as well as the total durations of the individual trajectory segments, which are given by $D_{p,l} = T^{\text{e}}_{p,l} - T^{\text{s}}_{p,l}$:
$$\boldsymbol{\zeta}^{\text{s,e}}_l = \sum_{p=1}^{P} \omega_p \, \boldsymbol{\zeta}^{\text{s,e}}_{p,l} \qquad D_l = \sum_{p=1}^{P} \omega_p \, D_{p,l} \tag{9}$$

4. **Re-warping and concatenation of the movement elements:** The linearly combined trajectories $\tilde{\zeta}_l(t)$ of the movement elements are un-normalized and concatenated to obtain the final composite sequence. Un-normalization is achieved by applying equation (8) in order to obtain the trajectory segments $\zeta_l(t)$. If the linear weight vectors of subsequent trajectory segments are different the condition $\zeta_l^e = \zeta_{l+1}^s$ that ensures the continuity of the trajectories after concatenation might be violated. To ensure continuity, start and endpoint pairs that violate these conditions are replaced by the average $(\zeta_l^e + \zeta_{l+1}^s)/2$ before un-normalization. Figure 5 illustrates the normalized linearly combined trajectory segments. Ten coordinates of the normalized trajectory after concatenation of the movement elements are shown. The black vertical lines illustrate the boundaries between the movement elements.

In the following, we discuss three applications of our algorithm with different focus. In the first example we show the capability of the method to synthesize different action styles based on a few recorded movement prototypes. Second, we describe the synthesis of complex movements with highly complex spatio-temporal characteristics by modeling sequences of techniques from martial arts. The last example shows an application of the method for imitation learning of writing movements for a robot arm. In this section we also discuss how the synthesized trajectories can be transferred to a robotics hardware that introduces additional kinematic and dynamic constraints.

3 Synthesis of Walking and Gesture Movements with Different Styles

The first application of our algorithm is the synthesis of a sequence of walking and gesture movements with different emotional affects. Using a commercial motion capture system (VICON 612) with 6 cameras we recorded subjects executing a movement sequence with different emotional affects. The movement sequence comprises several steps of straight walking, waving, turning around 180°, and straight walking again.

The motion sequences were recorded with 41 markers distributed over the whole body. Figure 6 (top level) shows six snapshots illustrating the marker positions connected by lines taken from the graphical user interface of the motion capture system. For modeling the movement sequences they were automatically decomposed into the movement primitives (straight walking, waving, turning and straight walking). Figure 6 (row 2-6) show the recorded movements (with the affects sad, neutral, and happy), and morphs between these emotional expressions[3]. The morphed sequences illustrate that our method is capable of synthesizing sequences of acyclic movements with different emotional affects that look

[3] Movies of these animations and the robot arm described in section 5 can be retrieved from the web site: http://www.uni-tuebingen.de/uni/knv/arl/index.html

quite natural. The linear weights define a metric Euclidean space of emotional affects that is spanned using a small number of training trajectories[4].

4 Synthesis of Complex Movements from Martial Arts

The second application of our method demonstrates that it can be applied for modeling highly complex human body movements. STMM were used to model sequences of techniques (called "katas") from karate. We have shown elsewhere [16] that the same algorithms can also be applied for an automatic estimation of the skill levels of karate fighters from the recorded movements. Here we focus on the synthesis of different karate styles and of technique sequences with different skill levels.

In a first experiment we have captured several movement sequences from a kata from two actors (figure 7). The first actor was a third degree black belt in Jujitsu, and the second actor had the 1. Kyu degree in karate (Shotokan). Both actors executed the same movement sequence, but due to differences of the techniques between different schools of martial arts with different styles. Three sequences of actor 1 have been segmented manually resulting in six movement primitives, which served as prototypes to define the morphable models of the first actor (see figure 9). Based on the 6 morphable models prototypical representations with key features for the automatic identification of the movement primitives were generated in the way described in section 2.3. The "reduced trajectories" $\boldsymbol{\kappa}(t)$ consist of the coordinates of the markers on both hands[5]. Figure 8 shows the results from the automatic segmentation from sequence of actor 2. The automatic segmentation was successful for all 16 sequences recorded from both actors. Figure 9 shows a morph that was created based on these automatically identified primitives.

Morphing between different actors Based on the movement primitives identified by automatic segmentation morphs between the movements of two different actors were realized. The individual movement primitives were morphed and afterward concatenated into a longer sequence. Figure 9 shows snapshots from a morphed motion sequence, which corresponds to the "average" of the two original sequences ($w_1 = w_2 = 0.5$ in eq. 7). This sequence looks natural and shows no artifacts at the margins between the individual movement primitives. In cases, where the styles of both actors are different, the morph generates a realistic movement that interpolates between the styles of the two actors original movements.

[4] In previous work [20] emotional expressions of real humanoid robots have been modeled by carefully designing the features of individual emotions by manipulation of selected joint trajectories

[5] The hand trajectories are computed relative to the shoulder markers. All marker trajectories were filtered using a Savitzky-Golay polynomial least-squares filter

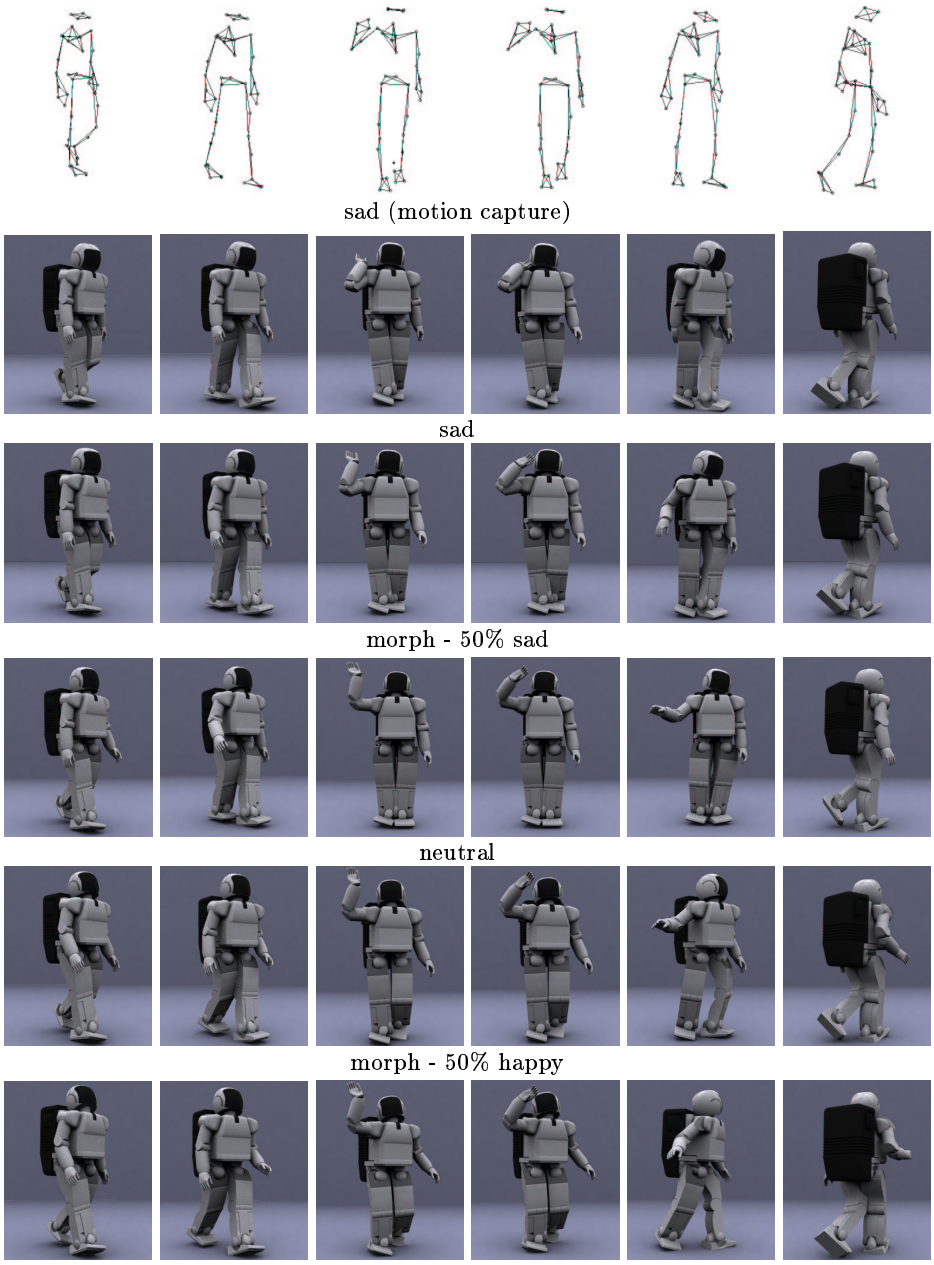

Fig. 6. Animations of synthesized movements with different emotional expressions. The top row shows snapshots of the marker positions obtained with the motion capture system. Rows 2, 4 and 6 show original captured trajectories with the affects sad, neutral and happy used for animating a computer graphics model of a humanoid robot. Rows 3 and 5 show the averages obtained with our method between sad and neutral, and between happy and neutral. Features that vary between the different emotional expressions are e.g. head posture, arm swing, step width, velocity and the body posture in turning motion.

Fig. 7. Recording movements from the karate kata "Heian Shodan" using a VICON motion capture system.

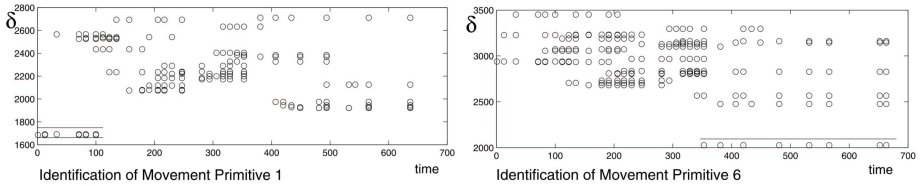

Fig. 8. Results of the automatic segmentation of one movement sequence of actor 2 based on the prototypical movement primitives of actor 1. As an example, the identification of the primitives 1 and 6 is shown. The diagrams show the distance measure δ (eq. 4) for different positions of the matching window of the corresponding movement primitive over the whole sequence. The circles mark the times of the matched key features κ_i^m of the sequence. Each match of a whole movement primitive is illustrated by a row of circles with the same δ. The number of circles corresponds to the number of key features of the movement primitive. The optimum match is given by the circles with minimal value of δ (indicated by the horizontal lines).

Morphing between different skill levels In a second larger experiment we captured the movements of 7 actors performing the karate kata "Heian Shodan". The actors had different belt levels (Kyu degrees) in karate (Shotokan). The kata was decomposed into 20 movement primitives (karate techniques). The total duration of the whole sequences was between 25 and 35 s. In this experiment we linearly combined the movements of karatekas with different skill levels in order to synthesize natural looking artificial karatekas with different belt levels. Furthermore it was possible to generate movement sequences that combine techniques with different skill levels within the same sequence. For example, the artificial karateka can start at beginner level and improve his performance gradually with each movement primitive[6].

[6] See [16] for further details as well as *http://www.uni-tuebingen.de/uni/knv/arl/index.html* for animations.

Fig. 9. Snapshots from a sequence of karate movements executed by two actors and a motion morph. The pictures show the initial posture at the beginning and the end postures of the movement primitives 1-5. The end posture is similar to the initial posture. The morphed sequence looks natural and contains no artifacts at the transitions between the 6 movement primitives. Especially interesting is the comparison between the different karate styles of the actors, i.e. for the third movement primitive (4th column). Actor 1 is doing a small side step with the left foot for turning. Instead of this, actor 2 turns without sidestep. The morph executes a realistic movement that interpolates between the two actors.

5 Imitation Learning of Writing Movements

The goal of imitation learning is to teach robots by observation of movement sequences[7]. Imitation learning has to address two fundamental problems. (1) The movement characteristics of observed movements have to be transferred from the perceptual level to the level of generated actions [31] [21]. (2) Continuous spaces of movements with variable styles have to be approximated based on a limited number of learned example sequences. This implies that the robot should be able to synthesize new movements based on the learned examples.

The proposed method was applied for synthesis and imitation of human writing movements. An overview of the overall algorithm is shown in figure 10. The

[7] In this paper we focus on the imitation of movement styles (see also [18], [25]). Our focus is not the imitation of event sequences, which are important for example in manipulation tasks like pick and place tasks.

steps of the method are briefly described in the following, focusing in particular on the transfer of the movements onto the robot arm.

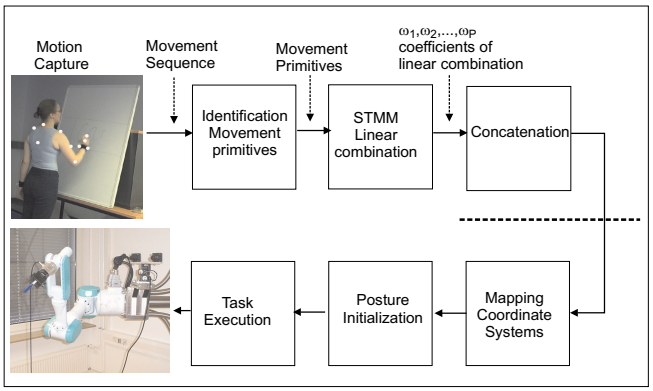

Fig. 10. Schematic description of the algorithm for synthesizing and imitation of writing movements. For modeling of the movement trajectories we used the proposed method. Movement primitives correspond to individual letters. The style of the individual primitives can be modified by choosing different linear weight combinations. The mapping of the movement sequences onto the robot arm is accomplished in three steps: mapping of coordinates, posture initialization, and task execution.

5.1 Synthesis of Writing Movements

We recorded writing movements of two human actors who wrote the word "ICAR" (figure 11). We used 10 markers that included the shoulders, 2 front and one rear torso, upper arm, elbow, front arm, hand and index finger of the writing arm.

Individual letters are defined as movement primitives. The automatic segmentation of the movement primitives was based on the index finger trajectories. The segmentation algorithm was trained with one example for each movement primitive that was obtained by manual segmentation of the trajectory of one of the actors (see [14] for details).

Continuous movement spaces for individual movement primitives are defined by the linear combinations of the prototypical movement primitives. The synthesized primitives are then automatically concatenated into longer sequences that can include multiple movement styles. Figure 12 shows the synthesized pen trajectories of the writing movements. The method allows to morph continuously between the writing sequences of the two actors (left panel). In addition, we can synthesize caricatures of the specific writing styles of each actor (right panel, EX A and EX B). The individual movement primitives can be reassembled in a different sequential order, e.g. in order to write the word "'IACR" (middle row). All these movement sequences were synthesized based on only two prototypical example trajectories.

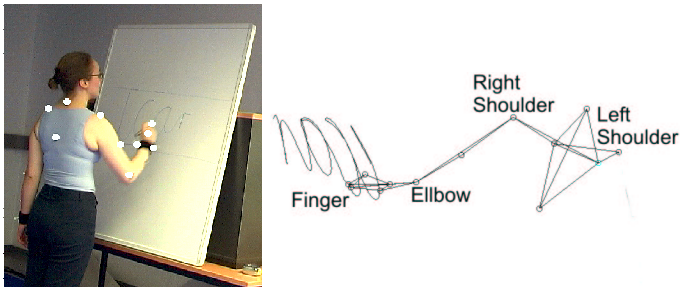

Fig. 11. Left panel: Motion capturing of writing movements on a board. White dots indicate the positions of the recorded markers. Right panel: Illustration of the marker set and the trace of the finger marker during the writing of the word "ICAR".

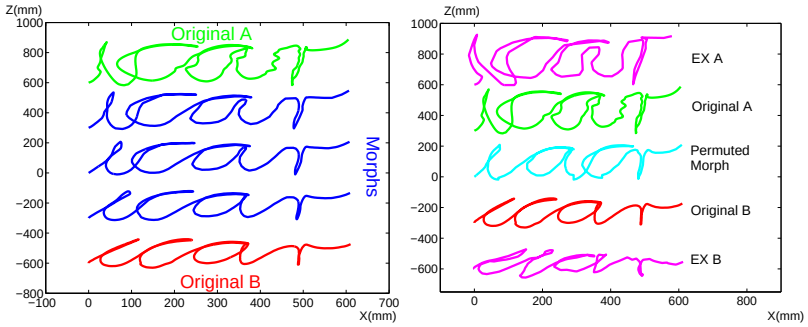

Fig. 12. Left panel: Recorded pen trajectories and morphs between the original writing movements. The morphs interpolate continuously in space-time between the prototypes. Right panel: Original pen trajectories and exaggerations of the writing styles of the two actors. The middle row shows synthesis of a new word "IACR" by reassembling the movement primitives in a different sequential order.

5.2 Transferring Human-Like Movements to a Robot Arm

The transfer of the trajectories to the robot is performed in three stages: 1) The HSTMM synthesizes trajectories in the same space as the prototype trajectories. Therefore, one has to transform synthesized trajectories from the prototype space into the task space of the robot. Also the trajectory must be scaled appropriately. 2) The second stage initializes the robot posture to a specific recorded (and appropriately transformed) initial human arm posture. 3) The task execution is performed by reproducing the exact end-effector trajectory and by approximating the human arm posture, as far as this is possible without violating kinematic constraints.

Mapping of the coordinate systems For the investigated task of writing movements the end effector trajectories are approximately planar. The drawing area of the synthesized writing movements has to be transformed into a drawing area in task space. The drawing plane is defined by two vectors $\mathbf{u}$ and $\mathbf{v}$, which

define a task orientation frame that is given by the matrix

$$\mathbf{T}_t = [\mathbf{u}\ \mathbf{v}\ \mathbf{u} \times \mathbf{v}]. \tag{10}$$

The starting point of the movement is given by the position vector $\mathbf{p}$. Since the task space is planar, we can use the first two principal components $\mathbf{e}_1, \mathbf{e}_2$ of the HSTMM output sequence $\zeta(t)$, to define an orientation frame of the trajectory as

$$\mathbf{T}_d = [\mathbf{e}_1\ \mathbf{e}_2\ \mathbf{e}_1 \times \mathbf{e}_2]. \tag{11}$$

Note that $\mathbf{e}_1, \mathbf{e}_2$ span the whole task space for our application. The trajectory $\zeta(t)$ is then first centered

$$\hat{\zeta}(t_i) = \zeta(t_i) - \frac{1}{N}\sum_{k=1}^{N}\zeta(t_k), \tag{12}$$

where we assume that the trajectory is given in a discretized form $\zeta(t_1), \ldots, \zeta(t_N)$ with $t_1 = 0$. The centered trajectory $\hat{\zeta}(t_i)$ can be scaled to avoid violation of task space constraints. The final target trajectory $\boldsymbol{\zeta}^\star(t)$ is given by

$$\boldsymbol{\zeta}^\star(t) = \mathbf{p} + \mathbf{T}_t\mathbf{T}_d^{-1}\left(\hat{\zeta}(t) - \hat{\zeta}(0)\right). \tag{13}$$

Initialization of robot posture The kinematic structure of humans and robots are usually different. Therefore, marker positions can usually not be transferred to the robot directly. Only if the robot is humanoid and has an equivalent kinematic structure the marker positions can be used directly [26], mechanisms to transfer motion capture data can be found for instance in [35]. We propose a way to transfer more the style of movements than explicit transfer joint angles. For this we to define "posture specifiers" that are applicable to humans as well as to robots. Imitation of posture is achieved by transferring these posture specifiers from the human to the robot.

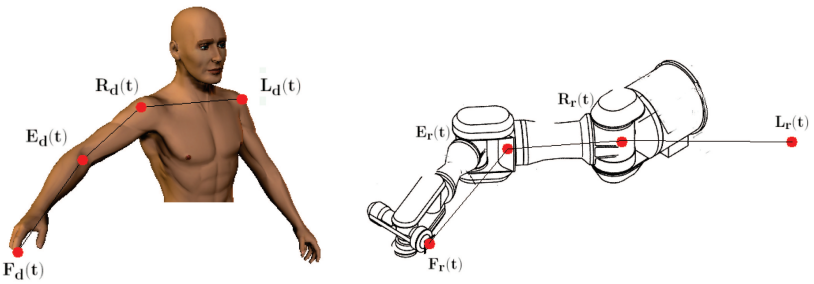

Fig. 13. Correspondence of human and robot posture.

Let $\mathbf{L}_d, \mathbf{R}_d, \mathbf{E}_d$ and $\mathbf{F}_d$ denote the positions of the left shoulder, right shoulder, elbow and the finger marker in the transformed prototype space (figure 13). As posture specifiers we choose orientation normals of two planes. The normal vector of the first plane is defined as

$$\mathbf{e}_d = \frac{(\mathbf{L}_d - \zeta^\star(t_1)) \times (\mathbf{E}_d - \zeta^\star(t_1))}{\| (\mathbf{L}_d - \zeta^\star(t_1)) \times (\mathbf{E}_d - \zeta^\star(t_1)) \|}. \tag{14}$$

This plane is spanned by the left shoulder, the elbow and an arbitrary reference point. In our case we chose the starting point $\zeta^\star(t_1)$ of the trajectory $\zeta^\star(t)$. Equivalently let

$$\mathbf{f}_d = \frac{(\mathbf{R}_d - \zeta^\star(t_1)) \times (\mathbf{F}_d - \zeta^\star(t_1))}{\| (\mathbf{R}_d - \zeta^\star(t_1)) \times (\mathbf{F}_d - \zeta^\star(t_1)) \|} \tag{15}$$

be the normal of the second plane which is spanned by finger, right shoulder and $\zeta^\star(t_1)$. Let $\mathbf{q} = [\mathbf{q}_1, \mathbf{q}_2]$ be the joint values of the robot, where $\mathbf{q}_1$ influences the elbow position and $\mathbf{q}_2$ does not. The corresponding plane normals $\mathbf{e}_r(\mathbf{q}_1)$, $\mathbf{f}_r(\mathbf{q}_2)$ of the robot are calculated in an equivalent way (see figure 14). For this purpose we use the a-priori specified position vector $\mathbf{p}$ from 5.2 instead of $\zeta^\star(t_1)$[8]. In addition a virtual left shoulder position has to be specified to determine the relative orientation of robot arm to the robot basis.

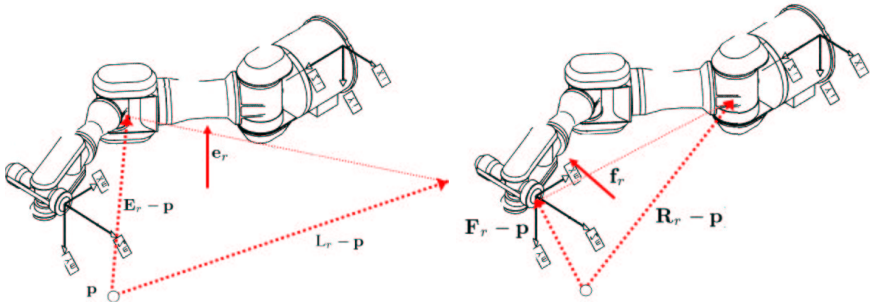

Fig. 14. Illustration of the plane normals $\mathbf{e}_r$ and $\mathbf{f}_r$. A virtual left shoulder $\mathbf{L}_r$ position of the robot is defined a-priori.

The initial posture of the robot is adjusted to the initial human posture by first minimizing

$$\min_{\mathbf{q}_1} \ \|\mathbf{e}_d - \mathbf{e}_r(\mathbf{q}_1)\|. \tag{16}$$

over the joints $\mathbf{q}_1$, and subsequently minimizing

$$\min_{\mathbf{q}_2} \ \|\mathbf{f}_d - \mathbf{f}_r(\mathbf{q}_2)\| \tag{17}$$

over $\mathbf{q}_2$. The solution minimizes the angles between $\mathbf{e}_r, \mathbf{e}_d$ and $\mathbf{f}_r, \mathbf{f}_d$ respectively.

Task Execution Starting from its initial posture, the trajectory of the robot is planned by solving the following optimization problem that depends on the discretely sampled joint variables $\mathbf{q}(t_i)$:

$$\min_{\mathbf{q}(t_i)} \ \rho(\mathbf{q}(t_i)) = \|\mathbf{e}_d - \mathbf{e}_r\|^2 + \alpha\|\mathbf{q}(t_i) - \mathbf{q}(t_{i-1})\|^2 \tag{18}$$

subject to

$$\mathbf{P}_r(\mathbf{q}(t_i)) - \zeta^\star(t_i) = 0 \tag{19}$$

[8] The reference point $\zeta(t_1)$ must ensure that $\mathbf{e}_d \neq f_d \forall t$. Otherwise another reference point has to be chosen.

where $\mathbf{P}_r(\mathbf{q}(t_i))$ describes the end-effector position. This problem is solved for each time step t_i of the trajectory separately. The objective function $\rho(\mathbf{q}(t_i))$ measures the euclidean distance between the normals $\mathbf{e}_d$ and $\mathbf{e}_r$. An additional regularization term is added to penalize high joint velocities. This term depends on the difference between the new joint configuration $\mathbf{q}(t_i)$ and the previous configuration $\mathbf{q}(t_{i-1})$. The scalar α determines the trade-off between smoothness of obtained joint trajectories and the quality of imitation. As a starting point, we use the joint values obtained by classical inverse kinematics. The joint trajectories were computed off-line[9].

The synthesized movements were executed using a Mitsubishi PA-10 7-DOF robot arm (figure 15). Optimization has been performed for different values of α (eq. 18). Figure 16 illustrates that for small values of α a better imitation (measured by the difference $\|\mathbf{e}_d - \mathbf{e}_d\|$) is achieved but discontinuous joint trajectories can arise. These discontinuities disappear for large values of α at the cost of worse imitation quality. Further analysis of the trajectories generated by imitation learning referring to robotic optimality measures can be found in [1].

Fig. 15. Left panel: The Mitsubishi PA–10 robot arm used to execute the writing movements. Right panel: Writing examples of the Originals A and B and the average morph in between (compare figure 12).

The proposed method for transferring the synthesized trajectories to the robot combines an exact control of the end effector position with a more "soft" control geometric variables that characterize the style of the executed arm movements. Another approach to include kinematic and end point constrains for the transfer of motion captured data can be found for instance in [29]. The proposed method can be generalized in a straightforward way to other tasks and movement classes, and is not restricted to the imitation of writing, and robot arms.

[9] A computational faster implementation to solve eq. (18) is obtained by using explicit information about the null space of the manipulator Jacobian (see [33]).

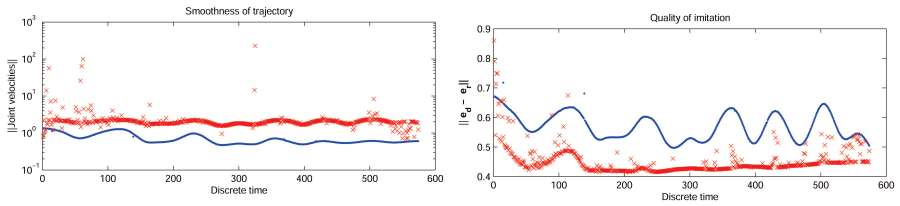

Fig. 16. Joint velocities (left panel) $\|\mathbf{q}(t_i) - \mathbf{q}(t_{i-1})\|$ and elbow norm difference (right panel) $\|\mathbf{e}_d - \mathbf{e}_r\|$ as a function of time for $\alpha = 10^{-1}$ (dots) and $\alpha = 10^{-2}$ (crosses). One obtains continuous joint trajectories for larger α.

6 Discussion

We have presented a method for the representation of complex movements that is based on linear combination of small sets of prototypical example movement sequences. The proposed algorithm decomposes long trajectories automatically into movement primitives, and models these primitives by linear combination of prototypical trajectories. Various methods for the parameterization of movement styles have been proposed in computer graphics and computer vision, e.g. based on Hidden Markov Models [4][37], principal component analysis [39] [3] [5], or Fourier components [36].

Different studies on imitation learning have investigated methods for describing the spatio-temporal characteristics of movements using principal component analysis [9] and spatio-temporal isomaps [18]. In [28] a verb-adverb approach was proposed that applies a combination of radial basis functions and low-order polynomials for defining parameterized interpolations between example movements. For this approach specific key times (e.g. the foot contact with the ground) must be specified by hand. Time Warping is defined by linear interpolation between these key times. In [19], [34] and [22], this interpolation is realized with splines.

The method of HSTMM has the advantage that it works with very small sets of training data [12][17][15]. Many popular methods for the representation of trajectories, e.g. HMMs or unsupervised learning of manifolds [18][4] typically require substantial amounts of training data. Another advantage of HSTMMs is the rather intuitive interpretation of the weights of the linear combinations that specify the style characteristics of the individual prototypes.

The presented application in robotics is a first demonstration of the application of HSTMMs in imitation learning. Future work has to apply and to extend the proposed algorithms for more complex robot systems, and for more complex tasks that include additional constraints, e.g. obstacle avoidance. The successful application of HSTMMs for the synthesis and analysis of complex whole body movements in computer graphics [12][15] and sports [16] suggests that the same algorithms might also perform well in imitation learning for humanoid robots.

Acknowledgments

This work is supported by the Deutsche Volkswagenstiftung. We thank also M.O. Franz, B. Eberhardt, H.P. Thier, B. Schölkopf, H.H. Bülthoff, W. Strasser, B. Knappmeyer, M. Hein, C. Röther, J. Jastorff, W. Jainek for further support.

References

1. G.H. Bakir, W. Ilg, M.O. Franz, and M.A. Giese. Constraints measures and reproduction of style in robot imitation learning. In *Beiträge zur 6. Tübinger Wahrnehmungskonferenz*, 2003.
2. A. Billard and M. Mataric. Learning human arm movements by imitation: Evaluation of a biologically-inspired connectionist architecture. *Robotics and Autonomous Systems*, 41(9):1–16, 2001.
3. A. F. Bobick and J. Davis. An appearance-based representation of action. In *Proceedings of the IEEE Conference on Pattern Recognition*, pages 307–312, 1996.
4. M. Brand. Style machines. In *SIGGRAPH*, 2000.
5. C. Bregler, L. Loeb, E. Chuang, and H. Deshpande. Turning to the masters: Motion capturing cartoons. In *SIGGRAPH*, 2002.
6. A. Bruderlin and L. Williams. Motion signal processing. In *SIGGRAPH*, pages 97–104, 1995.
7. T. Caelli, A. McCabe, and G. Binsted. On learning the shape of complex actions. In *International Workshop on Visual Form*, pages 24–39, 2001.
8. T. Flash and H. Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of Neuroscience*, 5:1688–1703, 1985.
9. A. Fod, M. J. Mataric, and O. C. Jenkins. Automated derivation of primitives for movement classification. *Autonomous Robots*, 12(1):39–54, 2002.
10. A. Galata, N. Johnson, and D. Hogg. Learning variable lenth markov models of behavior. *Journal of Computer Vision and Image Understanding*, 81:398–413, 2001.
11. M. A. Giese and T. Poggio. Synthesis and recognition of biological motion pattern based on linear superposition of prototypical motion sequences. In *Proceedings of IEEE MVIEW 99 Symposium at CVPR, Fort Collins*, pages 73–80, 1999.
12. M.A. Giese and T. Poggio. Morphable models for the analysis and synthesis of complex motion patterns. *International Journal of Computer Vision*, 38(1):59–73, 2000.
13. M. Gleicher. Retargeting motion to new characters. In *SIGGRAPH*, pages 33–42, 1998.
14. W. Ilg, G.H. Bakir, M.O. Franz, and M.A. Giese. Hierarchical spatio-temporal morphable models for representation of complex movements for imitation learning. In *IEEE International Conference on Advanced Robotics*, 2003.
15. W. Ilg and M.A. Giese. Modeling of movement sequences based on hierarchical spatial-temporal correspondence of movement primitives. In *Workshop on Biologically Motivated Computer Vision*, pages 528–537, 2002.
16. W. Ilg and M.A. Giese. Estimation of skill level in sports based on hierarchical spatio-temporal correspondences. 2003. submitted.
17. W. Ilg, M.A. Giese, H. Golla, and H.P. Thier. Quantitative movement analysis based on hierarchical spatial temporal correspondence of movement primitives. In *11th Annual Meeting of the European Society for Movement Analysis in Adults and Children*, 2002.
18. O.C. Jenkins and M. J. Mataric. Deriving action and behavior primitives from human motion data. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2551–2556, 2002.

19. J. Lee and S.Y. Shin. A hierarchical approach to interactive motion editing for human-like figures. In *SIGGRAPH*, 1999.
20. H.-o. Lim, A. Ishii, and A Takanishi. Emotion expression of a biped personal robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2000.
21. M. J. Mataric. Visuo-motor primitives as a basis for learning by imitation : Linking perception to action and biology to robotics. In K. Dautenhahn and C. Nehaniv, editors, *Imitation in Animals and Artifacts*, pages 392–422. MIT Press, 2002.
22. H. Miyamoto and M. Kawato. A tennis serve and upswing learning based on bi-directional theory. *Neural Networks*, 11:1331–1344, 1998.
23. T. Mori and K. Uehara. Extraction of primitive motion and discovery of association rules from motion data. In *Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication*, pages 200–206, 2001.
24. F. A. Mussa-Ivaldi, S. Gizter, and E. Bizzi. Linear combinations of primitives in vertebrate motor control. *Proccedings of the National Academy of Sciences*, 91:7534–7538, 1994.
25. A. Nakazawa, S. Nakaoka, K. Ikeuchi, and K. Yokoi. Imitating human dance motions through motion structure analysis. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2539–2544, 2002.
26. N. S. Pollard, J. Hodgins, M. J. Riley, and C. Atkeson. Adapting human motion for the control of a humanoid robot. In *IEEE International Conference on Robotics and Automation*, 2002.
27. B. Rohrer, S. Fasoli, H.I. Krebs, R. Hughes, B. Volpe, W. Frontera, J. Stein, and N. Hogan. Movement smoothness changes during stroke recovery. *Journal of Neuroscience*, 18:8297–8304, 2002.
28. C. Rose, M. F. Cohen, and B. Bodenheimer. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics and Applications*, 18(5):32–40, 1998.
29. A. Safonova, N.S. Pollard, and J.K. Hodgins. Optimizing human motion for the control of a humanoid robot. In *Proceedings of the 2nd International Symposium on Adaptive Motion of Animals and Machines*, march 2003.
30. T. D. Sanger. Human arm movements described by a low-dimensional superpostion of principal components. *Journal of Neuroscience*, 20(3):1066–1072, 2000.
31. S. Schaal. Is imitation learning a route to humanoid robots. *Trends in Cognitive Science*, 3:233–242, 1999.
32. S. Schaal. Dynamic movement primitives – a framework for motor control in humans and humanoid robots. In *Proceedings of the 2nd International Symposium on Adaptive Motion of Animals and Machines*, march 2003.
33. G. Schreiber, C. Ott, and G. Hirzinger. Interactive redundant robotics: Control of the inverted pendulum with nullspace motion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001.
34. A. Ude, C. G. Atkenson, and M. Riley. Planning of joint trajectories for humanoid robots using b-spline wavelets. In *IEEE International Conference on Robotics and Automation*, 2000.
35. A. Ude, C. Man, M. Riley, and C. G. Atkenson. Automatic generation of kinematic models for the conversion of human motion capture data into humanoid robot motion. In *First IEEE-RAS International Conference on Humanoid Robots, Cambridge, MA*, 2000. CD-Proceedings.
36. M. Unuma, K. Anjyo, and R. Takeuchi. Fourier principles for emotion-based human figure animation. In *SIGGRAPH*, pages 91–96, 1995.
37. A. D. Wilson and A. F. Bobick. Parametric hidden markov models for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):884–900, 1999.
38. A. Witkin and Z. Popovic. Motion warping. In *SIGGRAPH*, pages 105–108, 1995.
39. Y. Yacoob and M. J. Black. Parameterized modeling and recognition of activities. *Journal of Computer Vision and Image Understanding*, 73(2):398–413, 1999.