A Relational Approach to Content-based Analysis of Motion Capture Data

Meinard Müller and Tido Röder

Universität Bonn, Institut für Informatik III
Römerstr. 164, 53117 Bonn, Germany
{meinard, roedert}@cs.uni-bonn.de

Abstract. Motion capture or mocap systems allow for tracking and recording of human motions at high spatial and temporal resolutions. The resulting 3D mocap data is used for motion analysis in fields such as sports sciences, biomechanics, or computer vision, and in particular for motion synthesis in data-driven computer animation. In view of a rapidly growing corpus of motion data, automatic retrieval, annotation, and classification of such data has become an important research field. Since logically similar motions may exhibit significant spatio-temporal variations, the notion of similarity is of crucial importance in comparing motion data streams. After reviewing various aspects of motion similarity, we discuss as the main contribution of this paper a relational approach to content-based motion analysis, which exploits the existence of an explicitly given kinematic model underlying the 3D mocap data. Considering suitable combinations of boolean relations between specified body points allows for capturing the motion content while disregarding motion details. Finally, we sketch how such relational features can be used for automatic and efficient segmentation, indexing, retrieval, classification, and annotation of mocap data.

1 Introduction

Historically, the idea of motion capturing originates from the field of gait analysis, where locomotion patterns of humans and animals were investigated using arrays of analog photographic cameras, see Chapter **??**. With technological progress, motion capture data or simply *mocap data* became popular in computer animation to create realistic motions for both films and video games. Here, the motions are performed by live actors, captured by a digital mocap system, and finally mapped to an animated character. However, the lifecycle of a motion clip in the production of animations is very short. Typically, a motion clip is captured, incorporated in a single 3D scene, and then never used again. For efficiency and cost reasons, the reuse of mocap data as well as methods for modifying and adapting existing motion clips are gaining in importance. Applying editing, morphing, and blending techniques for the creation of new, realistic motions from prerecorded motion clips has become an active field of research [3, 13, 17, 18, 30, 39]. Such techniques depend on motion capture databases covering a

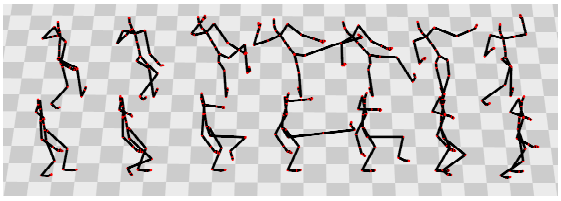

Fig. 1. Top: seven poses from a side kick sequence. **Bottom:** corresponding poses for a frontal kick. Even though the two kicking motions are similar in some logical sense, they exhibit significant spatial and temporal differences.

broad spectrum of motions in various characteristics. Larger collections of motion material such as [7] have become publicly available in the last few years. However, prior to reusing and processing motion capture material, one has to solve the fundamental problem of identifying and extracting logically related motions scattered in a given database. In this context, automatic and efficient methods for *content-based* motion analysis, comparison, classification, and retrieval are required that only access the raw mocap data itself and do not rely on manually generated annotations. Such methods also play an important role in fields such as sports sciences, biomechanics, and computer vision, see, e. g., Chapters **??**, **??**, and **??**.

One crucial point in content-based motion analysis is the notion of *similarity* that is used to compare different motions. Intuitively, two motions may be regarded as similar if they represent variations of the same action or sequence of actions [18]. Typically, these variations may concern the spatial as well as the temporal domain. For example, the kick sequences shown in Figure 1 describe a similar kind of motion even though they differ considerably with respect to motion speed as well as the direction, the height, and the style of the kick. How can a kicking motion be characterized irrespective of style? Or, conversely, how can motion style, the actor's individual characteristics, or emotional expressiveness be measured? Such questions are at the heart of motion analysis and synthesis. We will see that retrieval applications often aim at identifying related motions irrespective of certain motion details, whereas synthesis applications are often interested in just those motion details. Among other aspects of motion similarity, our discussion in Section 3 addresses the issue of separating motion details from motion content.

The difficult task of identifying similar motions in the presence of spatio-temporal variations still bears open problems. In this chapter, we will discuss analysis techniques that focus on the rough course of a motion while disregarding motion details. Most of the previous approaches to motion comparison are based on features that are semantically close to the raw data, using 3D positions, 3D point clouds, joint angle representations, or PCA-reduced versions thereof, see [12, 15, 16, 18, 34, 41]. One problem of such features is their sensitivity towards pose deformations, as may occur in logically related motions. Instead of using

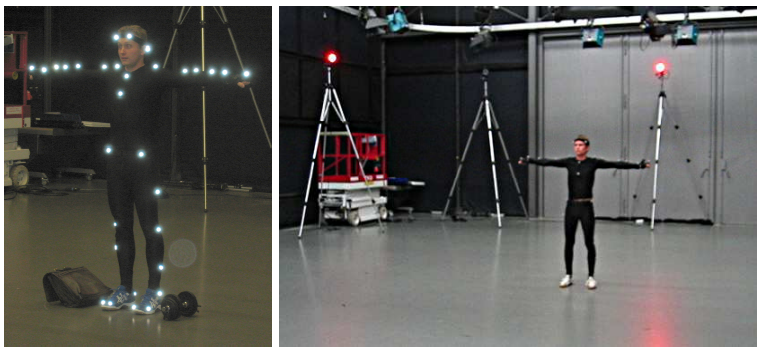

Fig. 2. Optical motion capture system based on retro-reflective markers attached to the actor's body. The markers are tracked by an array of six to twelve calibrated high-resolution cameras, typically arranged in a circle.

numerical, *quantitative* features, we suggest to use relational, *qualitative* features as introduced in [25]. Here, the following observation is of fundamental importance: opposed to other data formats such as images or video, 3D motion capture data is explicitly based on a kinematic chain that models the human skeleton. This underlying model can be exploited by looking for boolean relations between specified body points, where the relations possess explicit semantics. For example, even though there may be large variations between different kicking motions as illustrated by Figure 1, all such motions share some common characteristics: first the right knee is stretched, then bent, and finally stretched again, while the right foot is raised during this process. Afterwards, the right knee is once again bent and then stretched, while the right foot drops back to the floor. In other words, by only considering the temporal evolution of the two simple boolean relations "right knee bent or not" and "right foot raised or not", one can capture important characteristics of a kicking motion, which, in retrieval applications, allows for cutting down the search space very efficiently. In Section 4, we discuss in detail the concept and design of relational motion features. Then, in Section 5, we sketch several applications of relational features, including automatic and efficient motion segmentation, indexing, retrieval, annotation, and classification. In Section 2, for the sake of clarity, we summarize some basic facts about 3D motion capture data as used in this chapter, while describing the data model and introducing some notation. Further references to related work are given in the respective sections.

2 Motion Capture Data

There are many ways to generate motion capture data using, e. g., mechanical, magnetic, or optical systems, each technology having its own strengths and weaknesses. For an overview and a discussion of the pros and cons of such systems we

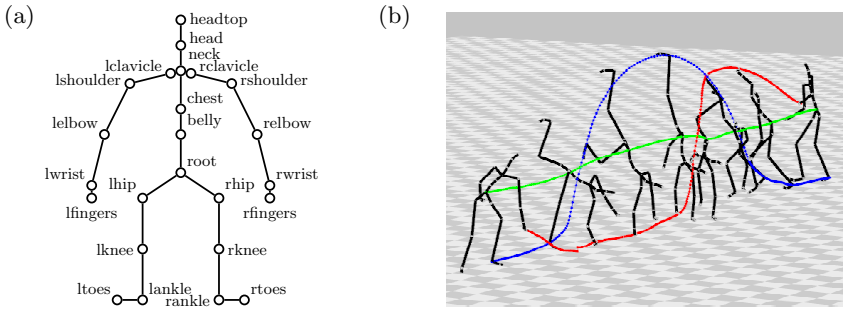

Fig. 3. (a) Skeletal kinematic chain model consisting of rigid *bones* that are flexibly connected by *joints*, which are highlighted by circular markers and labeled with joint names. **(b)** Motion capture data stream of a cartwheel represented as a sequence of poses. The figure shows the 3D trajectories of the joints 'root', 'rfingers', and 'lankle'.

refer to [38]. We exemplarily discuss an optical marker-based technology, which yields very clean and detailed motion capture data. Here, the actor is equipped with a set of 40–50 retro-reflective markers attached to a suit. These markers are tracked by an array of six to twelve calibrated high-resolution cameras at a frame rate of up to 240 Hz, see Figure 2. From the recorded 2D images of the marker positions, the system can then reconstruct the 3D marker positions with high precision (present systems have a resolution of less than a millimeter). Then, the data is cleaned with the aid of semi-automatic gap filling algorithms exploiting kinematic constraints. Cleaning is necessary to account for missing and defective data, where the defects are due to marker occlusions and tracking errors. For many applications, the 3D marker positions are then converted to a skeletal kinematic chain representation using appropriate fitting algorithms [9, 29]. Such an abstract model has the advantage that it does not depend on the specific number and the positions of the markers used for the recording. However, the mapping process from the marker data onto the abstract model can introduce significant artifacts that are not due to the marker data itself. Here, one major problem is that skeletal models are only approximations of the human body that often do not account for biomechanical issues, see [42].

In this chapter, we assume that the mocap data is modeled using a *kinematic chain*, which may be thought of as a simplified copy of the human skeleton. A kinematic chain consists of *body segments* (the bones) that are connected by *joints* of various types, see Figure 3 (a). Let J denote the set of joints, where each joint is referenced by an intuitive term such as 'root', 'lankle' (for 'left ankle'), 'rankle' (for 'right ankle'), 'lknee' (for 'left knee'), and so on. For simplicity, end effectors such as toes or fingers are also regarded as joints. In the following, a *motion capture data stream* is thought of as a sequence of *frames*, each frame specifying the 3D coordinates of the joints at a certain point in time. Moving from the technical background to an abstract geometric context, we also speak

of a *pose* instead of a frame. Mathematically, a pose can be regarded as a matrix $P \in \mathbb{R}^{3\times|J|}$, where $|J|$ denotes the number of joints. The j-th column of P, denoted by P^j, corresponds to the 3D coordinates of joint $j \in J$. A *motion capture data stream* (in information retrieval terminology also referred to as a *document*) can be modeled as a function

$$D : [1 : T] \to \mathcal{P} \subset \mathbb{R}^{3\times|J|}, \tag{1.1}$$

where $T \in \mathbb{N}$ denotes the number of poses, $[1 : T] := \{1, 2, \ldots, T\}$ corresponds to the time axis (for a fixed sampling rate), and $\mathcal{P}$ denotes the set of poses. A subsequence of consecutive frames is also referred to as a *motion clip*. Finally, the curve described by the 3D coordinates of a single body joint is termed *3D trajectory*. This definition is illustrated by Figure 3 (b).

3 Similarity Aspects

One central task in motion analysis is the design of suitable similarity measures to compare two given motion sequences in a semantically meaningful way. The notion of similarity, however, is an ill-defined term that depends on the respective application or on a person's perception. For example, a user may be interested only in the rough course of the motion, disregarding motion style or other motion details such as the facial expression. In other situations, a user may be particularly interested in certain nuances of motion patterns, which allows him to distinguish, e. g., between a front kick and a side kick, see Figure 1. In the following, we discuss some similarity aspects that play an important role in the design of suitable similarity measures or distance functions.

Typically, two motions are regarded as similar if they only differ by certain *global transformations* as illustrated by Figure 4 (a). For example, one may leave the absolute position in time and space out of consideration by using a similarity measure that is invariant under temporal and spatial translations. Often, two motions are identified when they differ with respect to a global rotation about the vertical axis or with respect to a global reflection. Furthermore, the size of the skeleton or the overall speed of the motions may not be of interest—in such a case, the similarity measure should be invariant to spatial or temporal scalings. More complex are variations that are due to different motion styles, see Figure 4 (b). For example, walking motions may differ by performance (e. g., limping, tiptoeing, or marching), by emotional expression or mood (e. g., "cheerful walking", "furious walking", "shy walking"), and by the complex individual characteristics determined by the motion's performer. The abstract concept of *motion style* appears in the literature in various forms and is usually contrasted by some notion of *motion content*, which is related to the semantics of the motion. In the following, we give an overview of how motion style and motion content are treated in the literature.

In the context of gait recognition, Lee and Elgammal, see [21] and Chapter **??**, define motion style as the time-invariant, personalized aspects of gait, whereas they view motion content as a time-dependent aspect representing different body

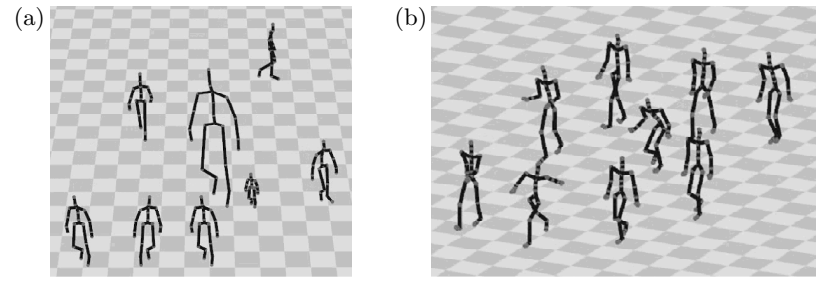

Fig. 4. (a) Different global transformations applied to a walking motion. **(b)** Different styles of walking motions.

poses during the gait cycle. Similarly, Davis and Gao [8] view motions as depending on style, pose, and time. In their experiments, they use PCA on expert-labeled training data to derive those factors (essentially linear combinations of joint trajectories) that best explain differences in style. Rose et al. [32] group several example motions that only differ by style into *verb* classes, each of which corresponds to a certain motion content. They synthesize new motions from these verb classes by suitable interpolation techniques, where the user can control interpolation parameters for each verb. These parameters are referred to as *adverbs* controlling the style of the verbs. To synthesize motions in different styles, Brand and Hertzmann [1] use example motions to train so-called *style machines* that are based on hidden Markov models (HMMs). Here, motion style is captured in certain parameters of the style machine such as average state dwell times and emission probability distributions for each state. On the other hand, motion content is encoded as the most likely state sequence of the style machine. Hsu et al. [15] propose a system for *style translation* that is capable of changing motions performed in a specific input style into new motions with the same content but a different output style. The characteristics of the input and output styles are learned from example data and are abstractly encoded in a linear dynamic system. A physically-based approach to grasping the stylistic characteristics of a motion performance is proposed by Liu et al. [23]. They use a complex physical model of the human body including bones, muscles, and tendons, the biomechanical properties of which (elasticity, stiffness, muscle activation preferences) can be learned from training data to achieve different motion styles in a synthesis step. Troje [36] trains linear PCA classifiers to recognize the gender of a person from recorded gait sequences, where the "gender" attribute seems to be located in the first three principal components of a suitable motion representation. Using a Fourier expansion of 3D locomotion data, Unuma et al. [37] identify certain *emotional* or *mood* aspects of locomotion style (for instance, "tired", "brisk", "normal") as gain factors for certain frequency bands. Pullen and Bregler [30] also use a frequency decomposition of motion data, but their aim is not to pinpoint certain parameters that describe specific styles. In-

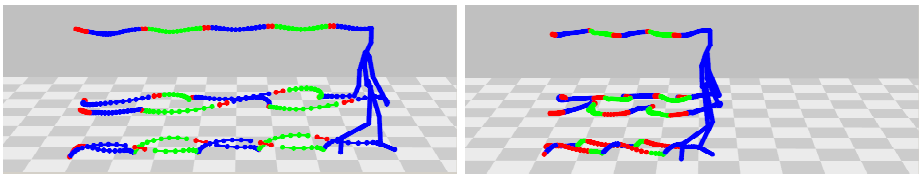

Fig. 5. Two walking motions performed in different speeds and styles. The figure shows the 3D trajectories for 'headtop', 'rfingers', 'lfingers', 'rankle', and 'lankle'. Logically corresponding segments in the two motions are indicated by the same colors.

stead, they try to extract those details of the data that account for the natural look of captured motion by means of multiresolution analysis (MRA) on mocap data [3]. These details are found in certain high-frequency bands of the MRA hierarchy and are referred to as *motion texture* in analogy to the texture concept in computer graphics, where photorealistic surfaces are rendered with texture mapping. The term "motion texture" is also used by Li et al. [22] in the context of motion synthesis, but their concept is in no way related to the signal processing approach of Pullen and Bregler [30]. In their parlance, motion textures are generative statistical models describing an entire class of motion clips. Similar to style machines [1], these models consist of a set of *motion textons* together with transition probabilities encoding typical orders in which the motion textons can be traversed. Each motion texton is a linear dynamic system (see also Hsu et al. [15]) that specializes in generating certain subclips of the modeled motion. Parameter tuning at the texton level then allows for manipulating stylistic details.

Inspired by the performing arts literature, Neff and Fiume [27, 28] explore the aspect of *expressiveness* in synthesized motions, see Chapter **??**. Their system enables the user to describe motion content in a high-level scripting language. The content can be modified globally and locally by applying procedural *character sketches* and *properties*, which implement expressive aspects such as "energetic", "dejected", or "old man".

Returning to the walking example of Figure 4 (b), we are faced with the question of how a walking motion can be characterized and recognized irrespective of motion style or motion texture. Video-based motion recognition systems such as [2, 14] tackle this problem by using hierarchical HMMs to model the motion content. The lower levels of the hierarchy comprise certain HMM building blocks representing fundamental components of full-body human motion such as "turning" or "raising an arm". In analogy to *phonemes* in speech recognition, these basic units are called *dynemes* by Green and Guan, see [14] and Chapter **??**, or *movemes* by Bregler [2]. Dynemes/movemes and higher-level aggregations of these building blocks are capable of absorbing some of the motion variations that distinguish different executions of a motion.

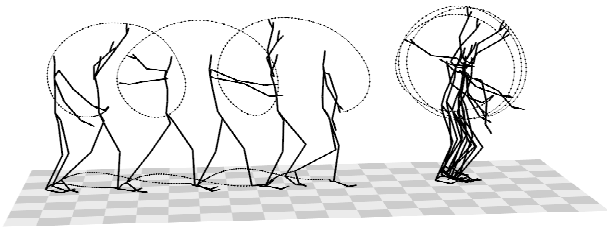

Fig. 6. Three repetitions of "rotating both arms forwards". The character on the left is walking while rotating the arms (2.7 seconds), whereas the character on the right is standing on one spot while rotating the arms (2.3 seconds). The trajectories of the joints 'rankle', 'lankle', and 'lfingers' are shown.

The focus of this chapter is the automatic analysis of motion content. How can one grasp the gist of a motion? How can logically similar motions be identified even in the presence of significant spatial and temporal variations? How can one determine and encode characteristic aspects that are common to all motions contained in some given motion class? As was mentioned earlier, the main problem in motion comparison is that logically related motions need not be numerically similar as was illustrated by the two kicking motions of Figure 1. As another example, the two walking motions shown in Figure 5 can be regarded as similar from a logical point of view even though they differ considerably in speed and style. Here, using techniques such as dynamic time warping, one may compensate for spatio-temporal deformations between related motions by suitably warping the time axis to establish frame correspondences, see [18]. Most features and local similarity measures used in this context, however, are based on numerical comparison of spatial or angular coordinates and cannot deal with qualitative variations. Besides spatio-temporal deformations, differences between logical and numerical similarity can also be due to *partial similarity*. For example, the two instances of "rotating both arms forwards" as shown in Figure 6 are almost identical as far as the arm movement is concerned, but differ with respect to the movement of the legs. Numerically, the resulting trajectories are very different—compare, for example, the cycloidal and the circular trajectories of the hands. Logically, the two motions could be considered as similar.

Even worse, numerical similarity does not necessarily imply logical similarity. For example, the two actions of picking up an object and placing an object on a shelf are very hard to distinguish numerically, even for a human [18]. Here, the context of the motion or information about interaction with objects would be required, see also [19]. Often, only minor nuances or partial aspects of a motion account for logical differences. Think of the motions "standing on a spot" compared to "standing accompanied by weak waving with one hand": such inconspicuous, but decisive details are difficult for a full-body similarity measure to pick up unless the focus of the similarity measure is primarily on the motion of the hands. As a further example, consider the difference between

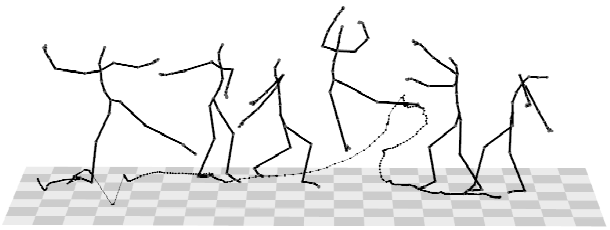

Fig. 7. A 500-frame ballet motion sampled at 120 Hz, adopted from the CMU mocap database [7]. The motion comprises two 180° right turns, the second of which is jumped. The trajectory of the joint 'ltoes' is shown.

walking and running. These motions may of course be distinguished by their absolute speed. Yet, the overall shape of most joints' trajectories is very similar in both motions. A better indicator would be the occurrence of simultaneous air phases for both feet, which is a discriminative feature of running motions.
Last but not least, noise is a further factor that may interfere with a similarity measure for motion clips. Mocap data may contain significant high-frequency noise components as well as undesirable artifacts such as sudden "flips" of a joint or systematic distortions due to wobbling mass or skin shift [20]. For example, consider the toe trajectory shown in the ballet motion of Figure 7, where the noise shows as extremely irregular sample spacing. Such noise is usually due to adverse recording conditions, occlusions, improper setup or calibration, or data conversion faults. On the left hand side of the figure, there is a discontinuity in the trajectory, which results from a 3-frame flip of the hip joint. Such flips are either due to confusions of trajectories in the underlying marker data or due to the fitting process. Ren et al. [31] have developed automatic methods for detecting "unnatural" movements in order to find noisy clips or clips containing artifacts within a mocap database. Noise and artifacts are also a problem in markerless, video-based mocap systems, see, e. g., [33] as well as Chapters **??**, **??**, and **??**. In view of such scenarios, it is important to design noise-tolerant similarity measures for the comparison of mocap data.

4 Relational Features

Applications of motion retrieval and classification typically aim at identifying related motions by content irrespective of motion style. To cope with significant numerical differences in 3D positions or joint angle configurations that may distinguish logically corresponding poses, we suggest to use qualitative features that are invariant to local deformations and allow for masking out irrelevant or inconsistent motion aspects. Note that mocap data, which is based on an explicit kinematic model, has a much richer semantic content than, for example, pure video data of a motion, since the position and the meaning of all joints is known for every pose. This fact can be exploited by considering features that describe

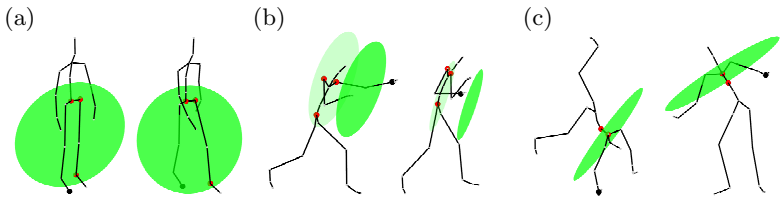

Fig. 8. Relational features describing geometric relations between the body points of a pose that are indicated by circular markers. The respective features express whether **(a)** the right foot lies in front of or behind the body, **(b)** the left hand is reaching out to the front of the body or not, **(c)** the left hand is raised above neck height or not.

boolean relations between specified points of a pose or short sequences of poses. Summarizing and extending the results of [25], we will introduce in this section several classes of boolean relational features that encode spatial, velocity-based, as well as directional information. The idea of considering relational instead of numerical features is not new and has already been applied by, e.g., Carlsson et al. [4, 5, 35] in other domains such as visual object recognition in 2D and 3D, or action recognition and tracking.

4.1 A Basic Example

As a basic example, we consider a relational feature that expresses whether the right foot lies in front of (feature value one) or behind (feature value zero) the plane spanned by the center of the hip (the root), the left hip joint, and the left foot for a fixed pose, cf. Figure 8 (a). More generally, let $p_i \in \mathbb{R}^3$, $1 \leq i \leq 4$, be four 3D points, the first three of which are in general position. Let $\langle p_1, p_2, p_3 \rangle$ denote the oriented plane spanned by the first three points, where the orientation is determined by point order. Then define

$$B(p_1, p_2, p_3; p_4) := \begin{cases} 1, \text{ if } p_4 \text{ lies in front of or on } \langle p_1, p_2, p_3 \rangle, \\ 0, \text{ if } p_4 \text{ lies behind } \langle p_1, p_2, p_3 \rangle. \end{cases} \tag{1.2}$$

From this we obtain a feature function $F_{\text{plane}}^{(j_1,j_2,j_3;j_4)} : \mathcal{P} \to \{0, 1\}$ for any four distinct joints $j_i \in J$, $1 \leq i \leq 4$, by defining

$$F_{\text{plane}}^{(j_1,j_2,j_3;j_4)}(P) := B(P^{j_1}, P^{j_2}, P^{j_3}; P^{j_4}). \tag{1.3}$$

The concept of such relational features is simple but powerful, as we will illustrate by continuing the above example. Setting j_1='root', j_2='lankle', j_3='lhip', and j_4='rtoes', we denote the resulting feature by $F^r := F_{\text{plane}}^{(j_1,j_2,j_3;j_4)}$. The plane determined by j_1, j_2, and j_3 is indicated in Figure 8 (a) as a green disc. Obviously, the feature $F^r(P)$ is 1 for a pose P corresponding to a person standing upright. It assumes the value 0 when the right foot moves to the back or the left foot to the front, which is typical for locomotion such as walking or running.

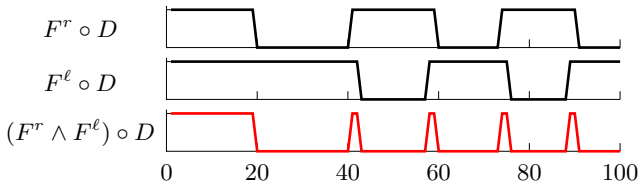

Fig. 9. Boolean features F^r, F^ℓ, and the conjunction $F^r \wedge F^\ell$ applied to the 100-frame walking motion $D = D_{\text{walk}}$ of Figure 15.

Interchanging corresponding left and right joints in the definition of F^r and flipping the orientation of the resulting plane, we obtain another feature function denoted by F^ℓ. Let us have a closer look at the feature function $F := F^r \wedge F^\ell$, which is 1 if and only if both, the right as well as the left toes, are in front of the respective planes. It turns out that F is very well suited to characterize any kind of walking or running movement. If a data stream $D : [1 : T] \to \mathcal{P}$ describes such a locomotion, then $F \circ D$ exhibits exactly two peaks for any locomotion cycle, from which one can easily read off the speed of the motion (see Figure 9). On the other hand, the feature F is invariant under global orientation and position, the size of the skeleton, and various local spatial deviations such as sideways and vertical movements of the legs. Furthermore, F leaves any upper body movements unconsidered.
In the following, we will define feature functions purely in terms of geometric entities that are expressible by joint coordinates. Such relational features are invariant under global transforms (Euclidean motions, scalings) and are very coarse in the sense that they express only a single boolean aspect, masking out all other aspects of the respective pose. This makes relational features robust to variations in the motion capture data stream that are not correlated with the aspect of interest. Using suitable boolean expressions and combinations of several relational features then allows to focus on or to mask out certain aspects of the respective motion.

4.2 Generic Features

The four joints in $F_{\text{plane}}^{(j_1,j_2,j_3;j_4)}$ can be picked in various meaningful ways. For example, in the case j_1='root', j_2='lshoulder', j_3='rshoulder', and j_4='lwrist', the feature expresses whether the left hand is in front of or behind the body. Introducing a suitable offset, one can change the semantics of a feature. For the previous example, one can move the plane $\langle P^{j_1}, P^{j_2}, P^{j_3} \rangle$ to the front by one length of the skeleton's humerus. The resulting feature can then distinguish between a pose with a hand reaching out to the front and a pose with a hand kept close to the body, see Figure 8 (b).
Generally, in the construction of relational features, one can start with some *generic relational feature* that encodes information about relative position, velocity, or direction of certain joints in 3D space. Such a generic feature de-

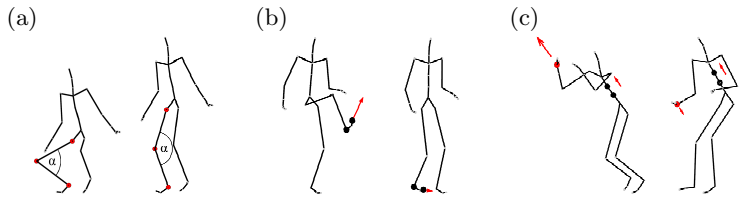

Fig. 10. Relational features that express whether **(a)** the right leg is bent or stretched, **(b)** the right foot is fast or not, **(c)** the right hand is moving upwards in the direction of the spine or not.

pends on a set of joint variables, denoted by $j_1, j_2, \ldots$, as well as on a variable θ for a threshold value or threshold range. For example, the generic feature $F_{\text{plane}} = F_{\theta,\text{plane}}^{(j_1,j_2,j_3;j_4)}$ assumes the value one iff joint j_4 has a signed distance greater than $\theta \in \mathbb{R}$ from the oriented plane spanned by the joints j_1, j_2 and j_3. Then each assignment to the joints $j_1, j_2, \ldots$ and the threshold θ leads to a boolean function $F : \mathcal{P} \to \{0, 1\}$. For example, by setting j_1='root', j_2='lhip', j_3='ltoes', j_4='rankle', and $\theta = 0$ one obtains the (boolean) relational feature indicated by Figure 8 (a).

Similarly, we obtain a generic relational feature $F_{\text{nplane}} = F_{\theta,\text{nplane}}^{(j_1,j_2,j_3;j_4)}$, where we define the plane in terms of a normal vector (given by j_1 and j_2) and fix it at j_3. For example, using the plane that is normal to the vector from the joint j_1='chest' to the joint j_2='neck' fixed at j_3='neck' with threshold $\theta = 0$, one obtains a feature that expresses whether a hand is raised above neck height or not, cf. Figure 8 (c).

Using another type of relational feature, one may check whether certain parts of the body such as the arms, the legs, or the torso are bent or stretched. To this end, we introduce the generic feature $F_{\text{angle}} = F_{\theta,\text{angle}}^{(j_1,j_2;j_3,j_4)}$, which assumes the value one iff the angle between the directed segments determined by (j_1, j_2) and (j_3, j_4) is within the threshold range $\theta \subset \mathbb{R}$. For example, by setting j_1='rknee', j_2='rankle', j_3='rknee', j_4='rhip', and $\theta = [0, 120]$, one obtains a feature that checks whether the right leg is bent (angle of the knee is below 120 degrees) or stretched (angle is above 120 degrees), see Figure 10 (a).

Other generic features may operate on velocity data that is approximated from the 3D joint trajectories of the input motion. An easy example is the generic feature $F_{\text{fast}} = F_{\theta,\text{fast}}^{(j_1)}$, which assumes the value one iff joint j_1 has an absolute velocity above θ. Figure 10 (b) illustrates the derived feature $F^{\text{rfootfast}} := F^{\text{rtoes}} \wedge F^{\text{rankle}}$, which is a movement detector for the right foot. $F^{\text{rfootfast}}$ checks whether the absolute velocity of both the right ankle (feature: F^{rankle}) and the right toes (feature: F^{rtoes}) exceeds a certain velocity threshold, θ_{fast}. If so, the feature assumes the value one, otherwise zero, see Figure 11. This feature is well-suited to detect kinematic constraints such as footplants. The reason why we require both the ankle and the toes to be sufficiently fast is that we only want to consider the

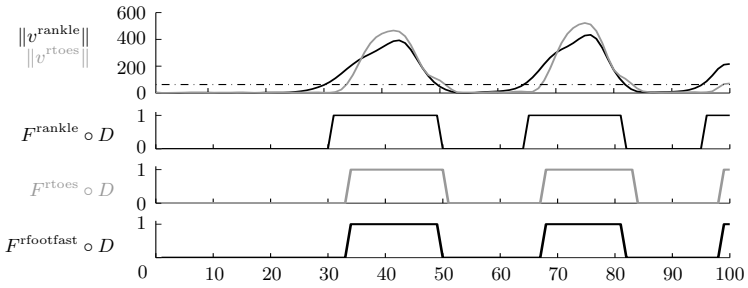

Fig. 11. Top: Absolute velocities in cm/s of the joints 'rankle' ($\|v^{\text{rankle}}\|$, black) and 'rtoes' ($\|v^{\text{rtoes}}\|$, gray) in the walking motion $D = D_{\text{walk}}$ of Figure 15. The dashed line at $\theta_{\text{fast}} = 63$ cm/s indicates the velocity threshold. **Middle**: Thresholded velocity signals for 'rankle' and 'rtoes'. **Bottom**: Feature values for $F^{\text{rfootfast}} = F^{\text{rtoes}} \wedge F^{\text{rankle}}$.

foot as being fast if all parts of the foot are moving. For example, during a typical walking motion, there are phases when the ankle is fast while the heel lifts off the ground, but the toes are firmly planted on the ground. Similarly, during heel strike, the ankle has zero velocity, while the toes are still rotating downwards with nonzero velocity. This feature illustrates one of our design principles for relational features: we construct and tune features so as to explicitly grasp the semantics of typical situations such as the occurrence of a footplant, yielding intuitive semantics for our relational features. However, while a footplant always leads to a feature value of zero for $F^{\text{rfootfast}}$, there is a large variety of other motions yielding the feature value zero (think of keeping the right leg lifted without moving). Here, the combination with other relational features is required to further classify the respective motions. In general, suitable combinations of our relational features prove to be very descriptive for full-body motions.

Another velocity-based generic feature is denoted by $F_{\text{move}} = F^{(j_1,j_2;j_3)}_{\theta,\text{move}}$. This feature considers the velocity of joint j_3 relative to joint j_1 and assumes the value one iff the component of this velocity in the direction determined by (j_1, j_2) is above θ. For example, setting j_1='belly', j_2='chest', j_3='rwrist', one obtains a feature that tests whether the right hand is moving upwards or not, see Figure 10 (c). The generic feature $F^{(j_1,j_2,j_3;j_4)}_{\theta,\text{nmove}}$ has similar semantics, but the direction is given by the normal vector of the oriented plane spanned by j_1, j_2, and j_3.

As a final example, we introduce generic features that check whether two joints, two body segments, or a joint and a body segment are within a θ-distance of each other or not. Here one may think of situations such as two hands touching each other, or a hand touching the head or a leg, see Figure 12. This leads to a generic feature $F^{(j_1,j_2,\theta)}_{\text{touch}}$, which checks whether the θ-neighborhoods of the joints j_1 and j_2 intersect or not. Similarly, one defines generic touch features for body segments.

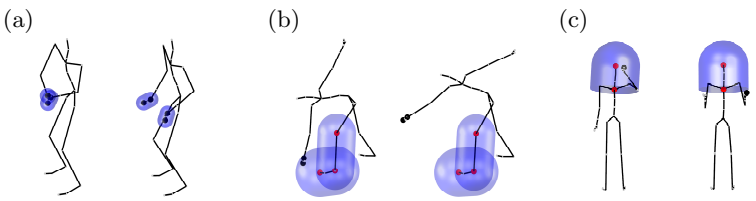

Fig. 12. Relational "touch" features that express whether **(a)** the two hands are close together or not, **(b)** the left hand is close to the leg or not, **(c)** the left hand is close to the head or not.

4.3 Threshold Selection

Besides selecting appropriate generic features and suitable combinations of joints, the crucial point in designing relational features is to choose the respective threshold parameter θ in a semantically meaningful way. This is a delicate issue, since the specific choice of a threshold has a strong influence on the semantics of the resulting relational feature. For example, choosing $\theta = 0$ for the feature indicated by Figure 8 (b) results in a boolean function that checks whether the left hand is in front of or behind the body. By increasing θ, the resulting feature checks whether the left hand is reaching out to the front of the body. Similarly, a small threshold in a velocity-based feature such as $F^{(j_1)}_{\theta,\text{fast}}$ leads to sensitive features that assume the value 1 even for small movements. Increasing θ results in features that only react for brisk movements. In general, there is no "correct" choice for the threshold θ—the specific choice will depend on the application in mind and is left to the designer of the desired feature set. In Section 4.4, we will specify a feature set that is suitable to compare the overall course of a full-body motion disregarding motion details.

To obtain a semantically meaningful value for the threshold θ in some automatic fashion, one can also apply supervised learning strategies. One possible strategy for this task is to use a training set $\mathcal{A}$ of "positive" motions that should yield the feature value one for most of its frames and a training set $\mathcal{B}$ of "negative" motions that should yield the feature value zero for most of its frames. The threshold θ can then be determined by a one-dimensional optimization algorithm, which iteratively maximizes the occurrences of the output one for the set $\mathcal{A}$ while maximizing the occurrences of the output zero for the set $\mathcal{B}$.

To make the relational features invariant under global scalings, the threshold parameter θ is specified relative to the respective skeleton size. For example, the value of θ may by given in terms of the length of the humerus, which scales quite well with the size of the skeleton. Such a choice handles differences in absolute skeleton sizes that are exhibited by different actors but may also result from different file formats for motion capture data.

Another problem arises from the simple quantization strategy based on the threshold θ to produce boolean features from the generic features. Such a strategy is prone to strong output fluctuations if the input value fluctuates slightly around

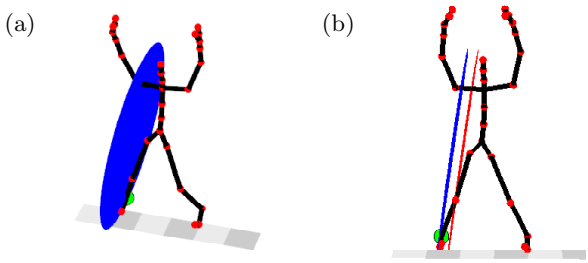

Fig. 13. Relational feature that expresses whether the right leg is stretched sideways or not. **(a)** The feature values may randomly fluctuate if the right ankle is located on the decision boundary (dark disc). **(b)** Introducing a second "weaker" decision boundary prevents the feature from fluctuations.

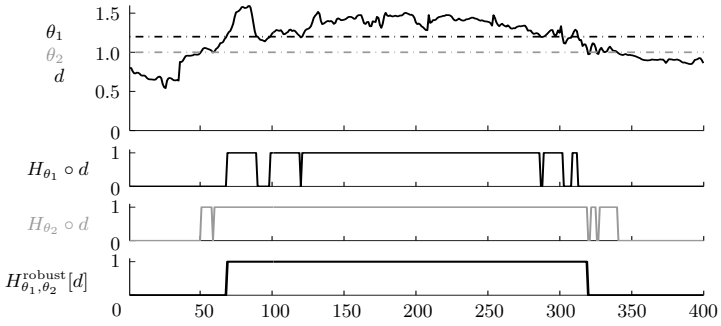

Fig. 14. Top: Distance d of the joint 'rankle' to the plane that is parallel to the plane shown in Figure 13 (a) but passes through the joint 'rhip', expressed in the relative length unit "hip width" (hw). The underlying motion is a Tai Chi move in which the actor is standing with slightly spread legs. The dashed horizontal lines at $\theta_2 = 1$ hw and $\theta_1 = 1.2$ hw, respectively, indicate the two thresholds, corresponding to the two planes of Figure 13 (b). **Middle**: Thresholded distance signals using the Heaviside thresholding function, H_θ; black: stronger threshold, θ_1; gray: weaker threshold, θ_2. **Bottom**: Thresholded distance signal using the robust thresholding operator $H^{\text{robust}}_{\theta_1,\theta_2}$.

the threshold. To alleviate this problem, we employ a robust quantization strategy using two thresholds: a stronger threshold θ_1 and a weaker threshold θ_2. As an example, consider a feature F^{sw} that checks whether the right leg is stretched sideways, see Figure 13. Such a feature can be obtained from the generic feature $F^{(j_1,j_2,j_3;j_4)}_{\theta,\text{nplane}}$, where the plane is given by the normal vector through j_1='lhip' and j_2='rhip' and is fixed at j_3='rhip'. Then the feature assumes the value one iff joint j_4='rankle' has a signed distance greater than θ from the oriented plane with a threshold $\theta = \theta_1 = 1.2$ measured in multiples of the hip width. As illustrated by Figure 13 (a), the feature values may randomly fluctuate, switching

ID	set	type	j_1	j_2	j_3	j_4	θ_1	θ_2	description
F_1 / F_2	u	F_{nmove}	neck	rhip	lhip	rwrist	1.8 hl/s	1.3 hl/s	rhand moving forwards
F_3 / F_4	u	F_{nplane}	chest	neck	neck	rwrist	0.2 hl	0 hl	rhand above neck
F_5 / F_6	u	F_{move}	belly	chest	chest	rwrist	1.8 hl/s	1.3 hl/s	rhand moving upwards
F_7 / F_8	u	F_{angle}	relbow	rshoulder	relbow	rwrist	$[0^\circ, 110^\circ]$	$[0^\circ, 120^\circ]$	relbow bent
F_9	u	F_{nplane}	lshoulder	rshoulder	lwrist	rwrist	2.5 sw	2 sw	hands far apart, sideways
F_{10}	u	F_{move}	lwrist	rwrist	rwrist	lwrist	1.4 hl/s	1.2 hl/s	hands approaching each other
F_{11} / F_{12}	u	F_{move}	rwrist	root	lwrist	root	1.4 hl/s	1.2 hl/s	rhand moving away from root
F_{13} / F_{14}	u	F_{fast}	rwrist				2.5 hl/s	2 hl/s	rhand fast
F_{15} / F_{16}	ℓ	F_{plane}	root	lhip	ltoes	rankle	0.38 hl	0 hl	rfoot behind lleg
F_{17} / F_{18}	ℓ	F_{nplane}	$(0, 0, 0)$	$(0, 1, 0)$	$(0, Y_{\text{min}}, 0)$	rankle	1.2 hl	1 hl	rfoot raised
F_{19}	ℓ	F_{nplane}	lhip	rhip	lankle	rankle	2.1 hw	1.8 hw	feet far apart, sideways
F_{20} / F_{21}	ℓ	F_{angle}	rknee	rhip	rknee	rankle	$[0^\circ, 110^\circ]$	$[0^\circ, 120^\circ]$	rknee bent
F_{22}	ℓ		Plane Π fixed at lhip, normal rhip→lhip. Test: rankle closer to Π than lankle?						feet crossed over
F_{23}	ℓ		Consider velocity v of rankle relative to lankle in rankle→lankle direction. Test: projection of v onto rhip→lhip line large?						feet moving towards each other, sideways
F_{24}	ℓ		Same as above, but use lankle→rankle instead of rankle→lankle direction.						feet moving apart, sideways
F_{25} / F_{26}	ℓ		$F^{\text{rfootfast}}$				2.5 hl/s	2 hl/s	rfoot fast
F_{27} / F_{28}	m	F_{angle}	neck	root	rshoulder	relbow	$[25^\circ, 180^\circ]$	$[20^\circ, 180^\circ]$	rhumerus abducted
F_{29} / F_{30}	m	F_{angle}	neck	root	rhip	rknee	$[50^\circ, 180^\circ]$	$[45^\circ, 180^\circ]$	rfemur abducted
F_{31}	m	F_{plane}	rankle	neck	lankle	root	0.5 hl	0.35 hl	root behind frontal plane
F_{32}	m	F_{angle}	neck	root	$(0, 0, 0)$	$(0, 1, 0)$	$[70^\circ, 110^\circ]$	$[60^\circ, 120^\circ]$	spine horizontal
F_{33} / F_{34}	m	F_{nplane}	$(0, 0, 0)$	$(0, -1, 0)$	$(0, Y_{\text{min}}, 0)$	rwrist	-1.2 hl	-1.4 hl	rhand lowered
F_{35} / F_{36}	m		Plane Π through rhip, lhip, neck. Test: rshoulder closer to Π than lshoulder?						shoulders rotated right
F_{37}	m		Test: Y_{min} and Y_{max} close together?						Y-extents of body small
F_{38}	m		Project all joints onto XZ-plane. Test: diameter of projected point set large?						XZ-extents of body large
F_{39}	m	F_{fast}	root				2.3 hl/s	2 hl/s	root fast

Table 1. A feature set consisting of 39 relational features.

between the numbers one and zero, if the right ankle is located on the decision boundary indicated by the dark disc. We therefore introduce a second decision boundary determined by a second, weaker, threshold $\theta_2 = 1.0$ indicated by the brighter disc in Figure 13 (b). We then define a robust version $F^{\text{sw}}_{\text{robust}}$ of F^{sw} that assumes the value one as soon as the right ankle moves to the right of the stronger decision boundary (as before). But we only let $F^{\text{sw}}_{\text{robust}}$ return to the output value zero if the right ankle moves to the left of the weaker decision boundary. It turns out that this heuristic of *hysteresis thresholding* [11, Chapter 4] suppresses undesirable zero-one fluctuations in relational feature values very effectively, see Figure 14.

4.4 Example of a Feature Set

Exemplarily, we describe a feature set that comprises $f = 39$ relational features. Note that this feature set has been specifically designed to focus on full-body

motions. However, the proposed feature set may be replaced as appropriate for the respective application.
The 39 relational features, given by Table 1, are divided into the three sets "upper", "lower", and "mix", which are abbreviated as u, ℓ and m, respectively. The features in the upper set express properties of the upper part of the body, mainly of the arms. Similarly, the features in the lower set express properties of the lower part of the body, mainly of the legs. Finally, the features in the mixed set express interactions of the upper and lower part or refer to the overall position of the body.
Features with two entries in the ID column exist in two versions pertaining to the right/left half of the body but are only described for the right half— the features for the left half can be easily derived by symmetry. The abbreviations "hl", "sw" and "hw" denote the relative length units "humerus length", "shoulder width", and "hip width", respectively, which are used to handle differences in absolute skeleton sizes. Absolute coordinates, as used in the definition of features such as F_{17}, F_{32}, or F_{33}, stand for virtual joints at constant 3D positions w.r.t. an (X, Y, Z) world system in which the Y axis points upwards. The symbols $Y_{\min}/Y_{\max}$ denote the minimum/maximum Y coordinates assumed by the joints of a pose that are not tested. Features such as F_{22} do not follow the same derivation scheme as the other features and are therefore described in words.

5 Applications

In this section, we show how relational features can be used for efficient motion retrieval, classification, and annotation. Fixing a set of boolean relational features, one can label each pose by its resulting feature vector. Such boolean vectors are ideally suited for indexing the mocap data according to these labels. Furthermore, a motion data stream can be segmented simply by grouping adjacent frames with identical labels. Motion comparison can then be performed at the segment level, which accounts for temporal variations, and efficient retrieval is possible by using inverted lists. As a further application, we introduce the concept of motion templates, by which the essence of an entire class of logically related motions can be captured. Such templates, which can be learned from training data, are suited for automatic classification and annotation of unknown mocap data.

5.1 Temporal Segmentation

We have seen that relational features exhibit a high degree of invariance against local spatial deformations. In this section, we show how to achieve invariance against local temporal deformations by means of a suitable feature-dependent temporal segmentation. To this end, we fix a list of, say, $f \in \mathbb{N}$ boolean relational features, which define the components of a boolean function $F : \mathcal{P} \to \{0, 1\}^f$. From this point forward, F will be referred to as a *feature function* and the vector $F(P)$ as a *feature vector* or simply a *feature* of the pose $P \in \mathcal{P}$. Any

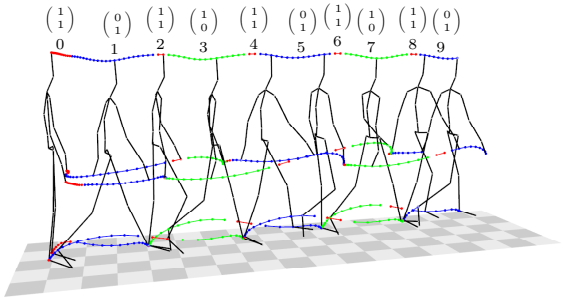

Fig. 15. F^2-segmentation of D_{walk}, where F^2-equivalent poses are indicated by uniformly colored trajectory segments. The trajectories of the joints 'headtop','rankle', 'rfingers' and 'lfingers' are shown.

feature function can be applied to a motion capture data stream $D : [1 : T] \to \mathcal{P}$ in a pose-wise fashion, which is expressed by the composition $F \circ D$. We say that two poses $P_1, P_2 \in \mathcal{P}$ are *F-equivalent* if the corresponding feature vectors $F(P_1)$ and $F(P_2)$ coincide, i. e., $F(P_1) = F(P_2)$. Then, an *F-run* of D is defined to be a subsequence of D consisting of consecutive F-equivalent poses, and the *F-segments* of D are defined to be the F-runs of maximal length.

We illustrate these definitions by continuing the example from Section 4.1. Let $F^2 := (F^r, F^\ell) : \mathcal{P} \to \{0,1\}^2$ be the combined feature formed by F^r and F^ℓ so that the pose set $\mathcal{P}$ is partitioned into four F^2-equivalence classes. Applying F^2 to the walking motion D_{walk} results in the segmentation shown in Figure 15, where the trajectories of selected joints have been plotted. F^2-equivalent poses are indicated by the same trajectory color: the color *red* represents the feature vector $(1,1)$, *blue* the vector $(1,0)$, and *green* the vector $(0,1)$. Note that no pose with feature vector $(0,0)$ appears in D_{walk}. Altogether, there are ten runs of maximal length constituting the F^2-segmentation of D_{walk}.

It is this feature-dependent segmentation that accounts for the postulated invariance under temporal deformations. To be more precise, let us start with the sequence of F-segments of a motion capture data stream D. Since each segment corresponds to a unique feature vector, the segments induce a sequence of feature vectors, which we simply refer to as the *F-feature sequence* of D and denote by $F[D]$. If M is the number of F-segments of D and if $D(t_m)$ for $t_m \in [1 : T]$, $0 \leq m < M$, is a pose of the m-th segment, then $F[D] = (F(D(t_0)), F(D(t_1)), \ldots, F(D(t_{M-1})))$. For example, for the data stream D_{walk} and the feature function F^2 from Figure 15, we obtain

$$F^2[D_{\text{walk}}] = \left(\binom{1}{1}, \binom{0}{1}, \binom{1}{1}, \binom{1}{0}, \binom{1}{1}, \binom{0}{1}, \binom{1}{1}, \binom{1}{0}, \binom{1}{1}, \binom{0}{1}\right). \tag{1.4}$$

Obviously, any two adjacent vectors of the sequence $F[D]$ are distinct. The crucial point is that time invariance is incorporated into the F-segments: two motions that differ by some deformation of the time axis will yield the same

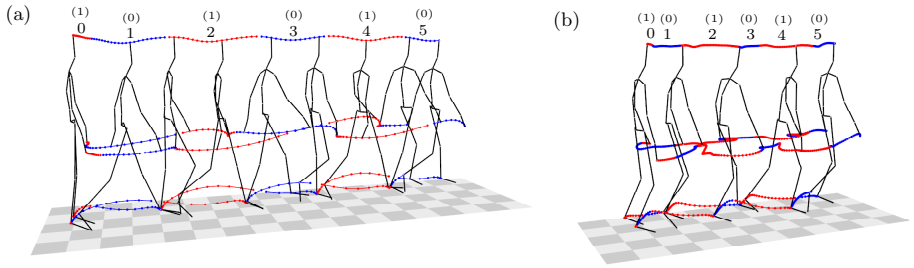

Fig. 16. (a) Restricting $F^2 = (F^r, F^\ell)$ to its first component results in an F^r-segmentation, which is coarser than the F^2-segmentation shown in Figure 15. **(b)** Five steps of a slow walking motion performed by an elderly person resulting in exactly the same F^r-feature sequence as the much faster motion of (a).

F-feature sequences. This fact is illustrated by Figure 16. Another property is that the segmentation automatically adapts to the selected features, as a comparison of Figure 15 and Figure 16 (a) shows. In general, fine features, i. e., feature functions with many components, induce segmentations with many short segments, whereas coarse features lead to a smaller number of long segments. The main idea is that two motion capture data streams D_1 and D_2 can now be compared via their F-feature sequences $F[D_1]$ and $F[D_2]$ instead of comparing the data streams on a frame-to-frame basis. This has several advantages:

1. One can decide which aspects of the motions to focus on by picking a suitable feature function F.
2. Since spatial and temporal invariance are already incorporated in the features and segments, one can use efficient methods from (fault-tolerant) string matching to compare the data streams instead of applying cost-intensive techniques such as dynamic time warping at the frame level.
3. In general, the number M of segments is much smaller than the number T of frames, which accounts for efficient computations.

Next, we will explain how our concept leads to an efficient way of indexing and searching motion capture data in a semantically meaningful way.

5.2 Indexing and Retrieval

In the retrieval context, the *query-by-example* paradigm has attracted a large amount of attention: given a query in form of a short motion clip, the task is to automatically retrieve all motion clips from the database that are logically similar to the query. The retrieved motion clips are also referred to as *hits* with respect to the query. Several general questions arise at this point:

1. How should the data, the database as well as the query, be modeled?
2. How does a user specify a query?

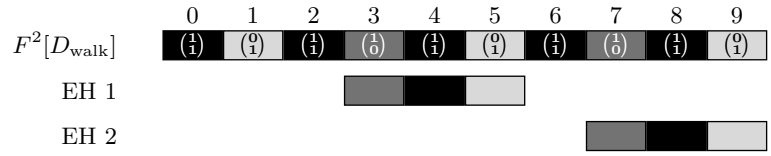

Fig. 17. Upper row: feature sequence $F^2[D_{\text{walk}}]$. **Below:** two exact hits (EH) for $\vec{v}_{\text{walk},1}$ in $F^2[D_{\text{walk}}]$, indicated by copies of $\vec{v}_{\text{walk},1}$ that are horizontally aligned with $F^2[D_{\text{walk}}]$ at the matching positions.

3. What is the precise definition of a hit?
4. How should the data be organized to afford efficient retrieval of all hits with respect to a given query?

In Section 5.1, we gave an answer to the first question by introducing the concept of feature sequences, which represent motion capture data streams as coarse sequences of binary vectors. For the moment, we assume that a query is given in form of a short motion clip Q. Furthermore, we assume that the database consists of a collection $\mathcal{D} = (D_1, D_2, \ldots, D_I)$ of mocap data streams or documents D_i, $i \in [1 : I]$. By concatenating the documents $D_1, \ldots, D_I$ while keeping track of document boundaries in a supplemental data structure, we may think of the database $\mathcal{D}$ as consisting of one large document D. Fixing a feature function $F : \mathcal{P} \to \{0,1\}^f$, we use the notation $F[D] = \vec{w} = (w_0, w_1, \ldots, w_M)$ and $F[Q] = \vec{v} = (v_0, v_1, \ldots, v_N)$ to denote the resulting F-feature sequences of D and Q, respectively. We then simply speak of the database $\vec{w}$ and the query $\vec{v}$.

Now, the trick is that by incorporating robustness against spatio-temporal variations into the relational features and adaptive segments, we are able to employ standard information retrieval techniques using an index of inverted lists [40]. For each feature vector $v \in \{0,1\}^f$ one stores the *inverted list* $L(v)$ consisting of the indices $m \in [0 : M]$ of the sequence $\vec{w} = (w_0, w_1, \ldots, w_M)$ with $v = w_m$. $L(v)$ tells us which of the F-segments of D exhibit the feature vector v. As an example, let us consider the feature function $F^2 = (F^r, F^\ell)$ from Figure 9 applied to a walking motion D as indicated by Figure 15. From the resulting feature sequence, one obtains the inverted lists $L\left(\binom{1}{1}\right) = \{0, 2, 4, 6, 8\}$, $L\left(\binom{0}{1}\right) = \{1, 5, 9\}$, $L\left(\binom{1}{0}\right) = \{3, 7\}$, and $L\left(\binom{0}{0}\right) = \emptyset$. The elements of the inverted lists can then be stored in ascending order, accounting for efficient union and intersection operations in the subsequent query stage. In a preprocessing step, we construct an index $I_F^{\mathcal{D}}$ consisting of the 2^f inverted lists $L(v)$, $v \in \{0,1\}^f$. Since we store segment positions of the F-segmentation rather than individual frame positions in the inverted lists, and since each segment position appears in exactly one inverted list, the index size is proportional to the number M of segments of D. In particular, the time and space required to build and store our index structure is *linear*, opposed to the *quadratic* complexity of strategies based on dynamic time warping, see [18].

Recall that two motion clips are considered as similar (with respect to the selected feature function) if they exhibit the same feature sequence. Adapting concepts from [6], we introduce the following notions. An *exact hit* is an element $k \in [0 : M]$ such that $\vec{v}$ is a subsequence of consecutive feature vectors in $\vec{w}$ starting from index k. Using the notation $\vec{v} \sqsubset_k \vec{w}$ for this case, one obtains

$$\vec{v} \sqsubset_k \vec{w} \;\; :\Leftrightarrow \;\; \forall i \in [0 : N] : \; v_i = w_{k+i}. \tag{1.5}$$

The set of all exact hits in the database $\mathcal{D}$ is then given by

$$H_{\mathcal{D}}(\vec{v}) := \{k \in [0 : M] \mid \vec{v} \sqsubset_k \vec{w}\}. \tag{1.6}$$

It is easy to see that $H_{\mathcal{D}}(\vec{v})$ can be evaluated very efficiently by intersecting suitably shifted inverted lists:

$$H_{\mathcal{D}}(\vec{v}) = \bigcap_{n \in [0:N]} (L(v_n) - n), \tag{1.7}$$

where the substraction of a list and a number is understood component-wise for every element in the list. As an example, we consider $D = D_{\text{walk}}$ and $F = F^2$ and the query sequence $\vec{v} = \left(\binom{1}{0}, \binom{1}{1}, \binom{0}{1}\right)$. Then

$$H_{\mathcal{D}}(\vec{v}) = \{3, 7\} \cap \{-1, 1, 3, 5, 7\} \cap \{-1, 3, 7\} = \{3, 7\} \tag{1.8}$$

resulting in two hits starting with the segments 3 and 7, respectively. See also Figure 17 for an illustration.

In many situations, the user may be unsure about certain parts of the query and wants to leave certain parts of the query unspecified. Or, the user may want to mask out some of the f components of the feature function F to obtain a less restrictive search leading to more hits. To handle such situations, one can employ the concept of *fuzzy search*. This technique admits at each position in the query sequence a whole set of possible, alternative feature vectors instead of a single one, see [6]. Here, a key idea is that the concept of temporal segmentation can be extended in such a way that segment lengths within a match not only adapt to the granularity of the feature function, but also to the fuzziness of the query. The resulting *adaptive fuzzy hits* can be computed very efficiently using the same index structure as for the case of exact hits. For further details on this strategy we refer to [25, 26].

We now describe how these techniques can be employed in an efficient motion retrieval system based on the query-by-example paradigm, which allows for intuitive and interactive browsing in a purely content-based fashion without relying on textual annotations, see Figure 18 for an overview. In the preprocessing step, a global feature function F has to be designed that covers all possible query requirements and provides the user with an extensive set of semantically rich features. In other words, it is not imposed upon the user to construct such features (even though this is also possible). Having fixed a feature function F, an index $I_F^{\mathcal{D}}$ is constructed for a given database $\mathcal{D}$ and stored on disk. (In practice,

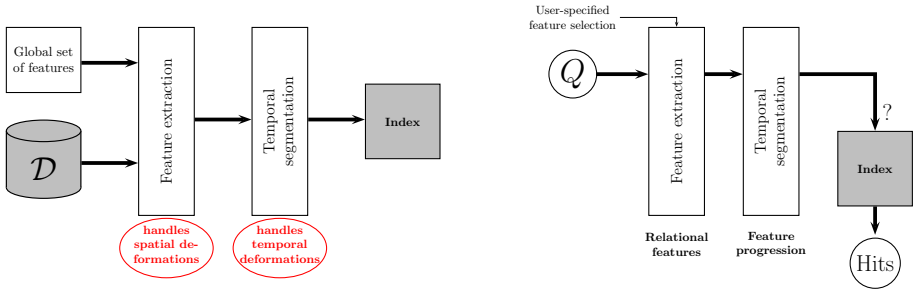

Fig. 18. Left: The preprocessing stage. **Right:** The query stage.

we split up the index into several smaller indices to reduce the number of inverted lists, see [25].) As an example, one may use the feature set comprising 39 relational features as described in Section 4.4. Note that this feature set has been specifically designed to focus on full-body motions. However, the described indexing and retrieval methods are generic, and the proposed test feature set may be replaced as appropriate for the respective application. Various query mechanisms of such a content-based retrieval system can be useful in practice, ranging from isolated pose-based queries, over query-by-example based on entire motion clips, up to manually specified geometric progressions. Here, we only consider the case that the input consists of a short query motion clip. Furthermore, the user should be able to incorporate additional knowledge about the query, e. g., by selecting or masking out certain body parts in the query. This is important to find, for example, all instances of "clapping one's hands" irrespective of any concurrent locomotion (recall the problem of partial similarity from Section 3.) To this end, the user selects relevant features from the given global feature set (i. e., components of F), where each feature expresses a certain relational aspect and refers to specific parts of the body. The query-dependent specification of motion aspects then determines the desired notion of similarity. In addition, parameters such as fault tolerance and the choice of a ranking or post-processing strategy can be adjusted. In the retrieval procedure, the query motion is translated into a feature sequence, which can be thought of as a progression of geometric constellations. The user-specified feature selection has to be encoded by a suitable fuzzy query, where the irrelevant features correspond to alternatives in the corresponding feature values. In the next step, the adaptive fuzzy hits are efficiently computed using the index. Finally, the hits may be post-processed by means of suitable ranking strategies. For further details we refer to [10, 25].

We implemented our indexing and retrieval algorithms in Matlab 6 and tested them on a database comprising roughly 180 minutes of motion data drawn from the CMU database [7]. The indexing time for $f = 31$ features (similar to the one of Table 1) was roughly 6 minutes. The storage requirement was reduced from 370 MB (for the entire database) to 7.5 MB (for the index). The running time to process a query very much depends on the query length (the number

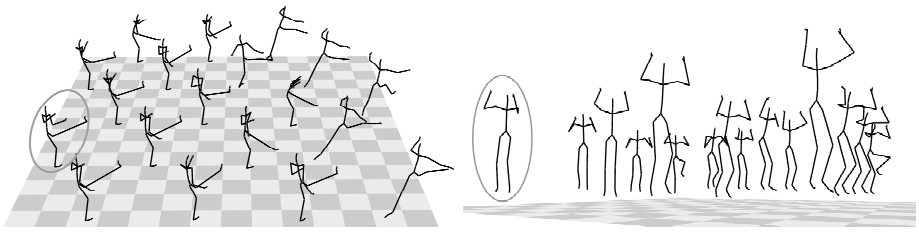

Fig. 19. Left: Selected frames from 19 adaptive fuzzy hits for a right foot kick. The query clip is highlighted. Query features: F_{17}, F_{18}, F_{20}, and F_{21}.; see Table 1. **Right:** Selected frames from 15 adaptive fuzzy hits for a jump query. Query features: F_3, F_4, F_{25}, and F_{26}.

of segments), the respective index, as well as the number of resulting hits. For example, Figure 19 (left) shows 19 adaptive fuzzy hits for a "kicking" motion (retrieval time: 5 ms), 13 of which are actual martial arts kicks. The remaining six motions (right hand side) are ballet moves containing a kicking component. A manual inspection of the database showed that there were no more than the 13 reported kicks in the database. Similarly, Figure 19 (right) shows the top 15 out of 133 hits for a very coarse adaptive fuzzy "jumping" query, which basically required the arms to move up above the shoulders and back down, while forcing the feet to lift off. The hits were ranked according to a simple strategy based on a comparison of segment lengths. This example demonstrates how such coarse queries can be applied to efficiently reduce the search space while retaining a superset of the desired hits.

One major limitation of this retrieval approach is that using all features at the same time in the retrieval process is far too restrictive—even in combination with fault tolerance strategies such as fuzzy or mismatch search—possibly leading to a large number of false negatives. Therefore, the user has to specify for each query a small subset of suitable features that reflect the characteristic properties of the respective query motion. Not only can this be a tedious manual process, but it also prohibits batch processing as needed in morphing and blending applications, where it may be required to identify similarities in a large database for many different motion clips without manual intervention. In the following, we introduce methods for automatic motion classification, annotation, and retrieval that overcome this limitation—however, at the expense of efficiency.

5.3 Motion Templates (MTs)

We now introduce a method for capturing the spatio-temporal characteristics of an entire motion class of logically related motions in a compact matrix representation called a *motion template* (MT). Given a set of training motions representing a motion class, we describe how to learn a motion template that explicitly encodes the consistent and the variable aspects of this class. Motion

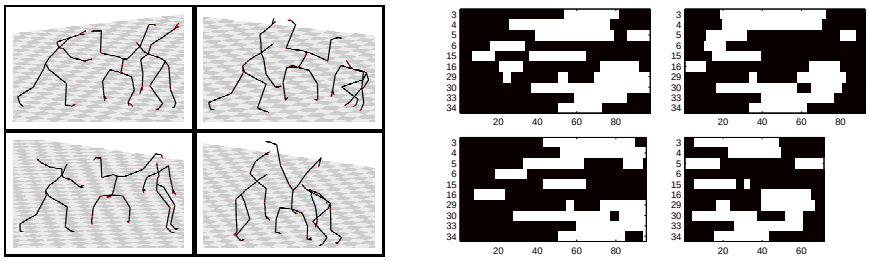

Fig. 20. Left: Selected frames from four different cartwheel motions. **Right:** Corresponding relational feature matrices for selected features. The columns represent time in frames, whereas the rows correspond to boolean features encoded as black (0) and white (1). They are numbered in accordance with the features defined in Table 1.

templates have a direct, semantic interpretation: an MT can easily be edited, manually constructed from scratch, combined with other MTs, extended, and restricted, thus providing a great deal of flexibility. One key property of MTs is that the variable aspects of a motion class can be automatically masked out in the comparison with unknown motion data. This strategy can also be viewed as an automatic way of selecting appropriate features for the comparison in a locally adaptive fashion.

In the following, we explain the main idea of motion templates and refer to [24] for details. Given a set of $\gamma \in \mathbb{N}$ example motion clips for a specific motion class, such as the four cartwheels shown in Figure 20, the goal is to automatically learn an MT representation that grasps the essence of the class. Based on a fixed set of f relational features, we start by computing the relational feature vectors for each of the γ motions. Denoting the length of a given motion by K, we think of the resulting sequence of feature vectors as a *feature matrix* $X \in \{0,1\}^{f \times K}$ as shown in Figure 20, where, for the sake of clarity, we only display a subset comprising ten features from the feature set of Table 1. Now, we want to compute a semantically meaningful average over the γ feature matrices, which would simply be their arithmetic mean if all of the motions agreed in length and temporal structure. However, our matrices typically differ in length and reflect the temporal variations that were present in the original motions. This fact necessitates some kind of temporal alignment prior to averaging, which is done by an iterative, reference-based time warping procedure, see [24] for details. Once the matrices have the same length, their average is computed, yielding as output a matrix with f rows, referred to as a *motion template*. The matrix entries are real values between zero and one. Figure 21 shows a motion template obtained from $\gamma = 11$ cartwheel motions (including the four cartwheels indicated by Figure 20), which constitutes a combined representation of all 11 input motions. An MT learned from training motions belonging to a specific motion class $\mathcal{C}$ is referred to as the *class template* for $\mathcal{C}$. Black/white regions in a class MT, see Figure 21, indicate periods in time (horizontal axis) where

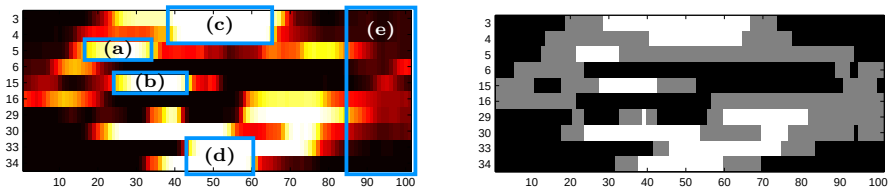

Fig. 21. Left: Class MT for 'CartwheelLeft' based on $\gamma = 11$ training motions. The framed regions are discussed in Section 5.3. **Right:** Corresponding quantized class MT.

certain features (vertical axis) consistently assume the same values zero/one in all training motions, respectively. By contrast, different shades of gray indicate inconsistencies mainly resulting from variations in the training motions (and partly from inappropriate alignments).

To illustrate the power of the MT concept, which grasps the essence of a specific type of motion even in the presence of large variations, we discuss the class template for the class 'CartwheelLeft', which consists of cartwheel motions starting with the left hand, see Figure 21. Considering the regions marked by boxes in Figure 21, the white region (a) reflects that during the initial phase of a cartwheel, the right hand moves to the top (feature F_5 in Table 1). Furthermore, region (b) shows that the right foot moves behind the left leg (F_{15}). This can also be observed in the first poses of Figure 20. Then, both hands are above the shoulders (F_3, F_4), as indicated by region (c), and the actor's body is upside down (F_{33}, F_{34}), see region (d) and the second poses in Figure 20. The landing phase, encoded in region (e), exhibits large variations between different realizations, leading to the gray/colored regions. Note that some actors lost their balance in this phase, resulting in rather chaotic movements, compare the third poses in Figure 20.

5.4 MT-based Motion Annotation and Retrieval

Given a class $\mathcal{C}$ of logically related motions, we have derived a class MT $X_{\mathcal{C}}$ that captures the consistent as well as the inconsistent aspects of all motions in $\mathcal{C}$. Our application of MTs to automatic annotation and retrieval are based on the following interpretation: the consistent aspects represent the class characteristics that are shared by all motions, whereas the inconsistent aspects represent the class variations that are due to different realizations. For a given class MT $X_{\mathcal{C}}$, we introduce a *quantized MT* by replacing each entry of $X_{\mathcal{C}}$ that is below δ by zero, each entry that is above $1 - \delta$ by one, and all remaining entries by 0.5. (In our experiments, we used the threshold $\delta = 0.1$.) Figure 21 (right) shows the quantized MT for the cartwheel class.

Now, let D be an unknown motion data stream. The goal is to identify subsegments of D that are similar to motions of a given class $\mathcal{C}$. Let $X \in \{0, 1, 0.5\}^{f \times K}$ be a quantized class MT of length K and $Y \in \{0, 1\}^{f \times L}$ the feature matrix of

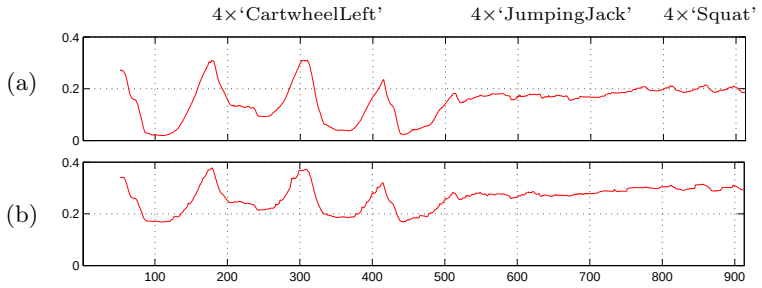

Fig. 22. (a) Distance function $\Delta_{\mathcal{C}}$ based on c^Q of (1.9) for the quantized class MT 'CartwheelLeft' and a motion sequence D consisting of four cartwheels (reflected by the four local minima close to zero), four jumping jacks, and four squats. The sampling rate is 30 Hz. **(b)** Corresponding distance function based on the Manhattan distance without MT quantization, leading to a much poorer result.

D of length L. We define for $k \in [1:K]$ and $\ell \in [1:L]$ a local cost measure $c^Q(k,\ell)$ between the k-th column $X(k)$ of X and the ℓ-th column $Y(\ell)$ of Y. Let $I(k) := \{i \in [1:f] \mid X(k)_i \neq 0.5\}$, where $X(k)_i$ denotes a matrix entry of X for $k \in [1:K]$, $i \in [1:f]$. Then, if $|I(k)| > 0$, we set

$$c^Q(k,\ell) = \frac{1}{|I(k)|} \sum_{i \in I(k)} |X(k)_i - Y(\ell)_i|, \tag{1.9}$$

otherwise we set $c^Q(k,\ell) = 0$. In other words, $c^Q(k,\ell)$ only accounts for the consistent entries of X with $X(k)_i \in \{0,1\}$ and leaves the other entries unconsidered. Based on this local distance measure and a subsequence variant of dynamic time warping, one obtains a distance function $\Delta_{\mathcal{C}} : [1:L] \to \mathbb{R} \cup \{\infty\}$ as described in [24] with the following interpretation: a small value $\Delta_{\mathcal{C}}(\ell)$ for some $\ell \in [1:L]$ indicates the presence of a motion subsegment of D starting at a suitable frame $a_\ell < \ell$ and ending at frame ℓ that is similar to the motions in $\mathcal{C}$. Note that using the local cost function c^Q of (1.9) based on the quantized MT (instead of simply using the Manhattan distance) is of crucial importance, as illustrated by Figure 22.

In the annotation scenario, we are given an unknown motion data stream D for which the presence of certain motion classes $\mathcal{C}_1, \ldots, \mathcal{C}_P$ at certain times is to be detected. These motion classes are identified with their respective class MTs $X_1, \ldots, X_P$, which are assumed to have been precomputed from suitable training data. Now, the idea is to match the input motion D with each of the X_p, $p = 1, \ldots, P$, yielding the distance functions $\Delta_p := \Delta_{\mathcal{C}_p}$. Then, every local minimum of Δ_p close to zero indicates a motion subsegment of D that is similar to the motions in $\mathcal{C}_p$. As an example, we consider the distance functions for a 35-second gymnastics motion sequence with respect to the motion classes $\mathcal{C}_1$='JumpingJack', $\mathcal{C}_2$='ElbowToKnee', and $\mathcal{C}_3$='Squat', see Figure 23. For $\mathcal{C}_1$, there are four local minima with a cost of nearly zero between frames 100 and

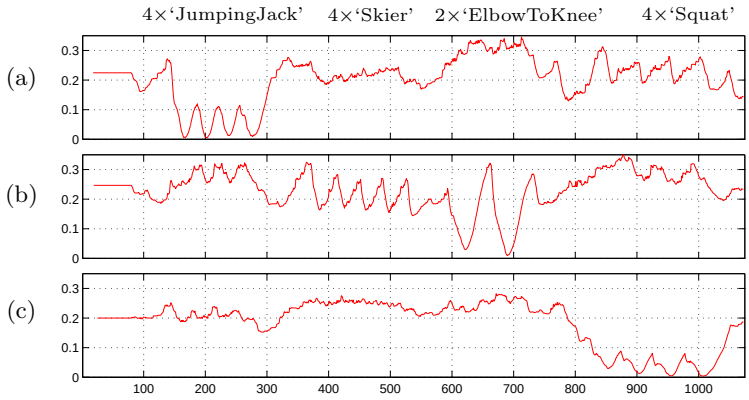

Fig. 23. Resulting distance functions for a 35-second gymnastics sequence (30 Hz) consisting of four jumping jacks, four repetitions of a skiing coordination exercise, two repetitions of an alternating elbow-to-knee motion, and four squats with respect to the quantized class MTs for **(a)** 'JumpingJack', **(b)** 'ElbowToKnee', and **(c)** 'Squat'.

300, which exactly correspond to the four jumping jacks contained in D, see Figure 23 (a). Note that the remaining portion of D is clearly separated by Δ_1, yielding a value far above 0.1. Analogously, the two local minima in Figure 23 (b) correspond to the two repetitions of the elbow-to-knee exercise and the four local minima in Figure 23 (c) correspond to the four squats.
Similarly, motion templates can be used for content-based motion retrieval, where the goal is to automatically extract all motion clips from a database that belong to a specified motion class $\mathcal{C}$. To this end, we compute a distance function $\Delta_{\mathcal{C}}$ with respect to the precomputed quantized class MT and the database documents. Then, each local minimum of $\Delta_{\mathcal{C}}$ below some quality threshold $\tau > 0$ indicates a hit, see [24] for details. As it turns out, the MT-based retrieval strategy works with high precision and recall for complex motions (such as a cartwheel) even in the presence of significant variations, whereas for short motions with few characteristic aspects it may produce a large number of false positives. Another drawback of the proposed MT-based retrieval strategy is its computational complexity, which is linear in the size of the database. For the future, we plan to combine the MT-based retrieval strategy with index-based retrieval techniques as proposed in Section 5.2. First experiments have shown that the use of suitably defined keyframes is a promising concept to cut down the set of candidate motions in an index-based preprocessing step. Such a preselection may also be suitable to eliminate a large number of false positives.

6 Conclusion and Future Work

In this chapter, various similarity aspects of 3D motion capture data have been discussed and reviewed. We then introduced the concept of relational features

that are particularly suited for the analysis of motion content and that facilitate logical (in contrast to numerical) comparison of motions. Once the features have been specified, they can be used for motion segmentation, efficient indexing, and fast content-based retrieval. As a further application, we introduced the concept of a motion template, which encodes the characteristic and the variable aspects of an entire motion class. By automatically masking out the variable aspects of a motion class in the annotation and retrieval process, logically related motions can be identified even in the presence of large variations and without any user intervention. We will investigate how to automatically learn characteristic keyframes in our template representation, which can then be used to cut down the search space efficiently. As a further promising application in the field of computer vision, we plan to use motion templates and related motion representations as a-priori knowledge to stabilize and control markerless tracking of human motions in video data, see also Chapters **??**, **??**, and **??**.

Acknowledgement: We would like to thank Bernd Eberhardt from HDM school of media sciences (Stuttgart) for providing us with extensive motion capture material. Furthermore, we thank Michael Clausen for constructive and valuable comments.

References

1. M. Brand and A. Hertzmann. Style machines. In *Proc. ACM SIGGRAPH 2000*, Computer Graphics Proc., pages 183–192. ACM Press, 2000.
2. C. Bregler. Learning and recognizing human dynamics in video sequences. In *Proc. CVPR 1997*, page 568, Washington, DC, USA, 1997. IEEE Computer Society.
3. A. Bruderlin and L. Williams. Motion signal processing. In *Proc. ACM SIGGRAPH 1995*, Computer Graphics Proc., pages 97–104. ACM Press, 1995.
4. S. Carlsson. Combinatorial geometry for shape representation and indexing. In *Object Representation in Computer Vision*, pages 53–78, 1996.
5. S. Carlsson. Order structure, correspondence, and shape based categories. In *Shape, Contour and Grouping in Computer Vision*, pages 58–71. Springer, 1999.
6. M. Clausen and F. Kurth. A unified approach to content-based and fault tolerant music recognition. *IEEE Trans. Multimedia*, 6(5):717–731, 2004.
7. CMU. Carnegie-Mellon Mocap Database. `http://mocap.cs.cmu.edu`, March, 2007.
8. J. W. Davis and H. Gao. An expressive three-mode principal components model of human action style. *Image Vision Comput.*, 21(11):1001–1016, 2003.
9. E. de Aguiar, C. Theobalt, and H.-P. Seidel. Automatic learning of articulated skeletons from 3D marker trajectories. In *Proc. Intl. Symposium on Visual Computing (ISVC 2006), to appear*, 2006.
10. B. Demuth, T. Röder, M. Müller, and B. Eberhardt. An information retrieval system for motion capture data. In *Proc. 28th European Conference on Information Retrieval (ECIR 2006)*, volume 3936 of *LNCS*, pages 373–384. Springer, 2006.
11. O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*, chapter 9, pages 341–400. MIT Press, Cambridge, MA, 1993.

12. K. Forbes and E. Fiume. An efficient search algorithm for motion data using weighted PCA. In *Proc. 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 67–76. ACM Press, 2005.
13. M. Giese and T. Poggio. Morphable models for the analysis and synthesis of complex motion patterns. *IJCV*, 38(1):59–73, 2000.
14. R. D. Green and L. Guan. Quantifying and recognizing human movement patterns from monocular video images: Part I. *IEEE Trans. Circuits and Systems for Video Technology*, 14(2):179–190, February 2004.
15. E. Hsu, K. Pulli, and J. Popović. Style translation for human motion. *ACM Trans. Graph.*, 24(3):1082–1089, 2005.
16. E. J. Keogh, T. Palpanas, V. B. Zordan, D. Gunopulos, and M. Cardle. Indexing large human-motion databases. In *Proc. 30th VLDB Conf., Toronto*, pages 780–791, 2004.
17. L. Kovar and M. Gleicher. Flexible automatic motion blending with registration curves. In *Proc. 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 214–224. Eurographics Association, 2003.
18. L. Kovar and M. Gleicher. Automated extraction and parameterization of motions in large data sets. *ACM Trans. Graph.*, 23(3):559–568, 2004.
19. P. G. Kry and D. K. Pai. Interaction capture and synthesis. *ACM Trans. Graph.*, 25(3):872–880, 2006.
20. M. A. Lafortune, C. Lambert, and M. Lake. Skin marker displacement at the knee joint. In *Proc. 2nd North American Congress on Biomechanics*, Chicago, 1992.
21. C.-S. Lee and A. Elgammal. Gait style and gait content: Bilinear models for gait recognition using gait re-sampling. In *Proc. IEEE Intl. Conf. Automatic Face and Gesture Recognition (FGR 2004)*, pages 147–152. IEEE Computer Society, 2004.
22. Y. Li, T. Wang, and H.-Y. Shum. Motion texture: a two-level statistical model for character motion synthesis. In *Proc. ACM SIGGRAPH 2002*, pages 465–472. ACM Press, 2002.
23. C. K. Liu, A. Hertzmann, and Z. Popović. Learning physics-based motion style with nonlinear inverse optimization. *ACM Trans. Graph.*, 24(3):1071–1081, 2005.
24. M. Müller and T. Röder. Motion templates for automatic classification and retrieval of motion capture data. In *Proc. 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA 2006)*. Eurographics Association, 2006.
25. M. Müller, T. Röder, and M. Clausen. Efficient content-based retrieval of motion capture data. *ACM Trans. Graph.*, 24(3):677–685, 2005.
26. M. Müller, T. Röder, and M. Clausen. Efficient indexing and retrieval of motion capture data based on adaptive segmentation. In *Proc. Fourth International Workshop on Content-Based Multimedia Indexing (CBMI)*, 2005.
27. M. Neff and E. Fiume. Methods for exploring expressive stance. In *Proc. 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA 2004)*, pages 49–58. ACM Press, 2004.
28. M. Neff and E. Fiume. AER: aesthetic exploration and refinement for expressive character animation. In *Proc. 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA 2005)*, pages 161–170. ACM Press, 2005.
29. J. F. O'Brien, R. Bodenheimer, G. Brostow, and J. K. Hodgins. Automatic joint parameter estimation from magnetic motion capture data. In *Graphics Interface*, pages 53–60, 2000.
30. K. Pullen and C. Bregler. Motion capture assisted animation: texturing and synthesis. In *Proc. SIGGRAPH 2002*, pages 501–508. ACM Press, 2002.

31. L. Ren, A. Patrick, A. A. Efros, J. K. Hodgins, and J. M. Rehg. A data-driven approach to quantifying natural human motion. *ACM Trans. Graph.*, 24(3):1090–1097, 2005.
32. C. Rose, M. F. Cohen, and B. Bodenheimer. Verbs and adverbs: multidimensional motion interpolation. *IEEE Comput. Graph. Appl.*, 18(5):32–40, 1998.
33. B. Rosenhahn, U. G. Kersting, A. W. Smith, J. K. Gurney, T. Brox, and R. Klette. A system for marker-less human motion estimation. In *DAGM-Symposium*, pages 230–237, 2005.
34. Y. Sakamoto, S. Kuriyama, and T. Kaneko. Motion map: image-based retrieval and segmentation of motion data. In *Proc. 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 259–266. ACM Press, 2004.
35. J. Sullivan and S. Carlsson. Recognizing and tracking human action. In *Proc. ECCV '02, Part I*, pages 629–644. Springer, 2002.
36. N. F. Troje. Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. *J. Vis.*, 2(5):371–387, 9 2002.
37. M. Unuma, K. Anjyo, and R. Takeuchi. Fourier principles for emotion-based human figure animation. In *Proc. ACM SIGGRAPH 1995*, pages 91–96. ACM Press, 1995.
38. Wikipedia. `http://en.wikipedia.org/wiki/Motion_capture`, March, 2007.
39. A. Witkin and Z. Popović. Motion warping. In *Proc. ACM SIGGRAPH 95*, Computer Graphics Proc., pages 105–108. ACM Press/ACM SIGGRAPH, 1995.
40. I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes*. Morgan Kaufmann Publishers, 1999.
41. M.-Y. Wu, S.-P. Chao, S.-N. Yang, and H.-C. Lin. Content-based retrieval for human motion data. In *16th IPPR Conf. on Computer Vision, Graphics, and Image Processing*, pages 605–612, 2003.
42. V. M. Zatsiorsky. *Kinematics of Human Motion*. Human Kinetics, 1998.

Index