Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2006)
M.-P. Cani, J. O'Brien (Editors)

Motion Templates for Automatic Classification and Retrieval of Motion Capture Data

Meinard Müller and Tido Röder

Department of Computer Science, University of Bonn, Germany

Abstract

This paper presents new methods for automatic classification and retrieval of motion capture data facilitating the identification of logically related motions scattered in some database. As the main ingredient, we introduce the concept of motion templates (MTs), by which the essence of an entire class of logically related motions can be captured in an explicit and semantically interpretable matrix representation. The 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 facilitates robust and efficient motion retrieval even in the presence of large spatio-temporal variations. Furthermore, we describe how to learn an MT for a specific motion class from a given set of training motions. In our extensive experiments, which are based on several hours of motion data, MTs proved to be a powerful concept for motion annotation and retrieval, yielding accurate results even for highly variable motion classes such as cartwheels, lying down, or throwing motions.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Animation

1. Introduction

The typical life cycle of a motion capture clip in the conventional production of computer-generated animations is very short: after some rehearsal, a motion clip is captured, incorporated in a single 3D scene, and then never used again. For reasons of flexibility, efficiency, and cost, much research on *motion reuse* for off-line and on-line synthesis of new motions from prerecorded motion data has been conducted. Here, the identification and extraction of logically related motions scattered within a large data set arises as a major problem. Such automatic methods for comparison, classification, and retrieval of motion data also play an important role in fields such as sports sciences, biometrics, medicine, and computer vision.

One major problem in content-based comparison of motion data is that logically similar motions need not be numerically similar, see [KG04]. In other words, there are certain aspects associated with a motion class that may show significant spatio-temporal variations between different executions of the motion, while other aspects are typically consistent. Like a fingerprint, these consistent aspects form the very essence of the motion class. In this paper, we propose a novel method for capturing the spatio-temporal characteristics of an entire motion class in a compact matrix representation called a *motion template* (MT). Given a set of training motions representing a motion class, a motion template that explicitly encodes the consistent and the variable aspects of the motion class can be learned. In addition, motion 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.

Based on our matching techniques, motion templates provide a fully automatic way of retrieving logically related motion segments from a large database and classifying or annotating segments of unknown motions. Here, a key contribution of our paper is to automatically exclude the variable aspects of a motion in the matching process while focusing on the consistent aspects—it is this idea that allows us to identify logically related motions even in the presence of large variations. This strategy can also be viewed as an automatic way of selecting appropriate features for the comparison in a locally adaptive fashion. In our experiments, we used qualitative boolean features that express suitable geometric relations between parts of the human body, as introduced by Müller et al. [MRC05]. As an important advantage,

our approach is generic in the following sense: by simply interchanging the feature set, one can use the same methods to characterize motions at different levels of granularity. For example, one could design one feature set that is specialized in full-body motions, and other feature sets that characterize aspects of arm or leg motions at a more detailed level.

The remainder of this paper is structured as follows. After discussing related work in Sect. 2, we briefly review the concept of relational motion features as introduced by Müller et al. [MRC05] in Sect. 3 and give an overview of the feature set used in our experiments in Appendix A. In Sect. 4, we describe how to use a set of logically related training motions to learn a *class MT* that captures the essence of the underlying motion class. Such a class MT can then be used as a matching operator for motion data streams that responds to motion segments belonging to the respective class, facilitating efficient and automatic motion annotation and retrieval, see Sect. 5. To prove the practicability of MT-based matching, we conducted extensive experiments based on a training set comprising roughly 1,500 short motion clips grouped into 64 classes as well as an unannotated test database consisting of 210 minutes of motion data. We compare MT-based matching to other methods, including the retrieval method by Müller et al. [MRC05] and several baseline methods using numerical similarity measures. To substantially speed up the annotation and retrieval process, we introduce an index-based preprocessing step to cut down the set of candidate motions by using suitable keyframes.

2. Related Work

The reuse of motion capture data via editing and morphing techniques has been a central topic in data-driven computer animation for a decade, starting with [BW95, WP95]. Since then, many different methods have been suggested to create new, realistic motions from prerecorded motions; see, for example, [GP00, PB02, AFO03, KG04, CH05, ZMCF05] and the references therein. Motion reuse requires efficient retrieval and browsing methods in order to fully exploit large motion databases. Due to possible spatio-temporal variations, the difficult task of identifying similar motion segments still bears open problems. 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 [WCYL03, KG04, KPZ*04, SKK04, FF05, HPP05]. One problem of such features is their sensitivity towards pose deformations, as may occur in logically related motions. To achieve more robustness, Liu et al. [LZWM05] transform motions into sequences of cluster centroids, which absorb spatio-temporal variations. Motion comparison is then performed on these sequences. The strategy of already absorbing spatio-temporal variations at the feature level is also pursued by Müller et al. [MRC05], who introduce relational features. It will turn out that such relational features become a powerful tool in combination with matching methods based on dynamic time warping (DTW). Originating from speech processing, DTW has become a well-established method to account for temporal variations in the comparison of related time series, see [RJ93]. In the context of motion retrieval, most of the approaches cited above rely on some variant of this technique—the crucial point being the choice of features and local cost measures. DTW is also used for motion alignment in blending applications such as the method by Kovar and Gleicher [KG03].

Automatic motion annotation and classification are closely related to the retrieval problem and constitute important tasks in view of motion reuse. Arikan and Forsyth [AFO03] propose a semi-automatic annotation procedure for motion data using SVM classifiers. Ramanan and Forsyth [RF03] apply this annotation technique for 3D motion data as a preprocessing step for the automatic annotation of 2D video recordings of human motion, using hidden Markov models (HMMs) to match the 2D data with the 3D data. Rose et al. [RCB98] group similar example motions into "verb" classes to synthesize new, user-controlled motions by suitable interpolation techniques. Several approaches to classification and recognition of motion patterns are based on HMMs, which are also a flexible tool to capture spatio-temporal variations, see, e. g., [BH00, GG04]. Opposed to HMM-based motion representations, where timing information is encoded in the form of transition probabilities, the motion representation developed in this paper encodes absolute and relative lengths of key events explicitly. Temporal segmentation of motion data can be viewed as another form of annotation, where consecutive, logically related frames are organized into groups, see, e. g., [FMJ02, BSP*04].

3. Relational Motion Features

In the following, a motion capture data stream D is regarded as a sequence of poses, where each pose consists of a full set of 3D coordinates describing the joint positions of a skeletal kinematic chain for a fixed point in time; see the lower right part of Table 6. In order to grasp characteristic aspects of motions, we adopt the concept of *relational motion features*, which describe (boolean) geometric relations between specified points of a pose, see [MRC05]. As an example of such a feature, consider a fixed pose for which we test whether the right foot lies in front of (feature value zero) or behind (feature value one) the plane spanned by the center of the hip (the root), the left hip joint, and the left foot, cf. Table 6 (a). Applying a set of f relational motion features to a motion data stream D of length K in a pose-wise fashion yields a *feature matrix* $X \in \{0,1\}^{f \times K}$, see Fig. 2 for an example. The k^{th} column of X then contains the feature values of frame k and will be denoted by $X(k)$, $k \in [1:K] := \{1,2,\ldots,K\}$. The main point is that even though relational features discard a lot of detail contained in the raw motion data, important information regarding the overall configuration of a pose is

retained. Moreover, relational motion features are invariant under global orientation and position, the size of the skeleton, and local spatial deformations of a pose, cf. [MRC05].

In this paper, we use the feature set described in Appendix A, which comprises $f = 39$ relational features and is very similar to the features used in [MRC05]. Note that our feature set has been specifically designed to focus on full-body motions. However, the methods described in this paper are generic, and the proposed test feature set may be replaced as appropriate for the respective application.

4. Motion Templates

Generalizing boolean feature matrices, we introduce in this section the concept of *motion templates* (MTs), which is suited to express the essence of an entire class of motions. A motion template of *dimension* f and *length* K is a real-valued matrix $X \in [0,1]^{f \times K}$. Each row of an MT corresponds to one relational feature, and time (in frames) runs along the columns, see Fig. 3 for an example. For the rest of the paper, we assume that all MTs under consideration have the same fixed dimension f. Intuitively, an MT can be thought of as a "fuzzified" version of a feature matrix; for the proper interpretation of the matrix entries, we refer to Sect. 4.1, where we describe how to learn an MT from training motions by a combination of time warping and averaging operations.

During the learning procedure, a *weight vector* $\alpha \in \mathbb{R}^K_{>0}$ is associated with an MT, where the total weight $\bar{\alpha} := \sum_{k=1}^{K} \alpha(k)$ is at least one. We say that the k^{th} column $X(k)$ of X has weight $\alpha(k)$. These weights are used to keep track of the time warping operations: initially, each column of an MT corresponds to the real time duration of one frame, which we express by setting all weights $\alpha(k)$ to one. Subsequent time warping may change the amount of time that is allotted to an MT column. The respective weights are then modified so as to reflect the new time duration. Hence, the weights allow us to unwarp an MT back to real time, similar to the strategy used in [HPP05].

4.1. Learning MTs from Example Motions

Given a set of N example motion clips for a specific motion class, such as the four cartwheels shown in Fig. 1, our goal is to automatically learn a meaningful MT that grasps the essence of the class. We start by computing the feature matrices for a fixed set of features, as shown in Fig. 2, where, for the sake of clarity, we only display a subset comprising ten features from our test feature set. From this point forward, we will consider feature matrices as a special case of MTs. Weight vectors α are attached to each of the MTs and initialized to $\alpha(k) = 1$ for all k.

Now, the goal is to compute a semantically meaningful average over the N input MTs, which would simply be the arithmetic mean of the feature matrices if all of the motions agreed in length and temporal structure. However, our

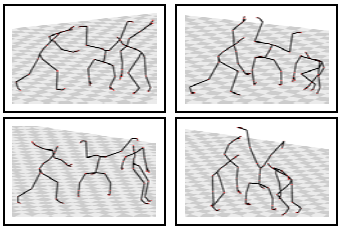

Figure 1: *Selected frames from four different cartwheel motions.*

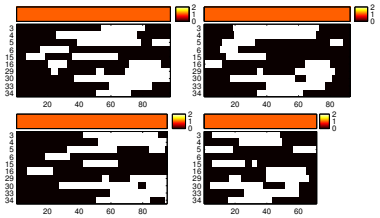

Figure 2: *Relational feature matrices resulting from the motions in Fig. 1 for selected features with associated weight vectors. The columns represent time in frames, whereas the rows correspond to boolean features encoded as black (0) and white (1), and are numbered in accordance with the relational features defined in Table 6. The weight vectors are displayed as color-coded horizontal bars.*

MTs 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. We do this by choosing one of the input MTs as the *reference MT*, R, and applying dynamic time warping (DTW) [RJ93] to compute optimal alignments of the remaining MTs with R (we measure local distances of feature vectors by the Manhattan distance, which coincides with the Hamming distance for boolean feature vectors.) According to these optimal alignments, the MTs are locally stretched and contracted, where time stretching is simulated by duplicating MT columns, while time contractions are resolved by forming a weighted average of the columns in question. As indicated above, the weights α associated with an MT X must now be adapted accordingly: in case a column $X(\ell)$ was matched to n columns $R(k), \ldots, R(k+n-1)$ of the reference, the new weights $\alpha'(k+i)$ are set to $\frac{1}{n}\alpha(\ell)$ for $i = 0, \ldots, n-1$, i. e., the weight $\alpha(\ell)$ is equally distributed among the n matching columns. In case column $R(k)$ of the reference was matched to multiple columns of X, the new weight $\alpha'(k)$ is the sum of the weights of the matching columns in X.

Now that all MTs and associated weight vectors have the same length as the reference MT, we compute the weighted average over all MTs in a column-wise fashion as well as the arithmetic mean α of all weight vectors. Note that the total weight, $\bar{\alpha}$, equals the average length of the input motions. Fig. 3 (a) shows the results for our cartwheel example, where the top left MT in Fig. 2 acted as the reference. Finally, we unwarp the average MT according to the weight vector: column ranges with $\alpha(k) < 1$ are unwarped by contracting the respective MT columns into one average col-

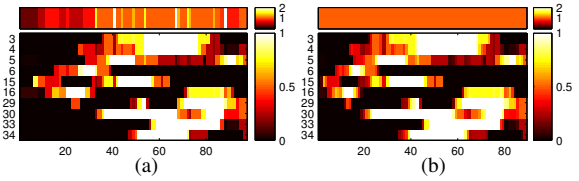

Figure 3: ***(a)*** *Average MT and average weights computed from the MTs in Fig. 2 after all four MTs have been aligned with the top left MT, which acted as the reference. The MTs are coded in white (1), black (0), and shades of gray for intermediary values (red and yellow in the color version).* ***(b)*** *Unwarped version of the MT in (a).*

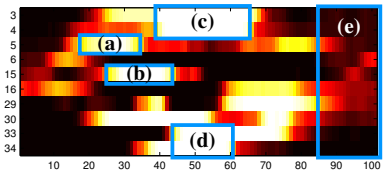

Figure 4: *Class MT for 'CartwheelLeft' based on $N = 11$ training motions. The boxes are exemplarily discussed in Sect. 4.3.*

Motion Class $\mathcal{C}$	Comment	Size	N	$\bar{\alpha}$	M	$t(\mathcal{C})$
CartwheelLeft	left hand first on floor	21	11	105.3	6	17.0
ElbowToKnee	start: relbow/lknee	27	14	36.6	5	4.9
JumpingJack	1 repetition	52	26	35.5	6	19.3
KickFrontRFoot	1 kick	30	15	53.3	5	9.4
KickSideRFoot	1 kick	30	15	48.9	6	10.1
LieDownFloor	start: standing pose	20	10	165.0	5	25.6
RotateRArmBwd3	3 times	16	8	80.8	4	3.8
RotateRArmFwd3	3 times	16	8	83.6	4	3.9
Squat	1 repetition	52	26	47.3	5	24.6
WalkSideRight3	3 steps	16	8	123.0	3	5.5

Table 1: *$\mathcal{D}^{\mathrm{MC}}$ contains 10 to 50 different variations for each of its 64 motion classes. This table shows ten of the motion classes, along with their respective size, the size N of the training subset, the average length $\bar{\alpha}$ in frames, as well as the number M of iterations and the running time $t(\mathcal{C})$ in seconds required to compute $X_{\mathcal{C}}$.*

umn (e. g., $k = 6, \ldots, 10$ in Fig. 3 (a)), while columns with $\alpha(k) > 1$ are unwarped by duplicating the respective column (e. g., $k = 42$). Since in general, columns will not have integer or reciprocal integer weights, we additionally perform suitable partial averaging between adjacent columns such that all weights but the last are one in the resulting unwarped MT, see Fig. 3 (b). Note that the total weight, $\bar{\alpha}$, is preserved by the unwarping procedure. The average MT now constitutes a combined representation of all the input motions, but it is still biased by the influence of the reference MT, to which all of the other MTs have been aligned. Our experiments show that it is possible to eliminate this bias by the following strategy: we let each of the original MTs act as the reference and perform for each reference the entire averaging and unwarping procedure as described above. This yields N averaged MTs corresponding to the different references. Then, we use these N MTs as the input to a second pass of mutual warping, averaging, and unwarping, and so on. The procedure is iterated until no major changes occur. Fig. 4 shows the output for $N = 11$ training motions.

Interpretation of MTs: An MT learned from training motions belonging to a specific motion class $\mathcal{C}$ is referred to as the *class template* $X_{\mathcal{C}}$ for $\mathcal{C}$. Note that the weight vector does not play a role any longer. Black/white regions in a class MT, see Fig. 4, indicate periods in time (horizontal axis) where certain features (vertical axis) consistently assume the same values zero/one in all training motions, respectively. By contrast, gray regions (red to yellow in the color version of this paper) indicate inconsistencies mainly resulting from variations in the training motions (and partly from inappropriate alignments). Some illustrative examples will be discussed in Sect. 4.3.

4.2. Experimental Results

For our experiments, we systematically recorded several hours of motion capture data containing a number of well-specified motion sequences, which were executed several times and performed by five different actors. Using this data, we built up the database $\mathcal{D}_{210}$ consisting of roughly 210 minutes of motion data. Then, we manually cut out suitable motion clips from $\mathcal{D}_{210}$ and arranged them into 64 different classes. Each such motion class (MC) contained 10 to 50 different realizations of the same type of motion, covering a broad spectrum of semantically meaningful variations. For example, the motion class 'CartwheelLeft' contained 21 variations of a cartwheel motion, all starting with the left hand. The resulting *motion class database* $\mathcal{D}^{\mathrm{MC}}$ contains 1,457 motion clips, amounting to 50 minutes of motion data.

Table 1 gives an overview of some of the motion classes contained in $\mathcal{D}^{\mathrm{MC}}$. We split up $\mathcal{D}^{\mathrm{MC}}$ into two disjoint databases $\mathcal{D}^{\mathrm{MCT}}$ and $\mathcal{D}^{\mathrm{MCE}}$, each consisting of roughly half the motions of each motion class. The database $\mathcal{D}^{\mathrm{MCT}}$ served as the *training database* to derive the motion templates, whereas $\mathcal{D}^{\mathrm{MCE}}$ was used as a training-independent *evaluation database*. All databases were preprocessed by computing and storing the feature matrices. Here, we used a sampling rate of 30 Hz, which turned out to be sufficient in view of MT quality. The duration of the training motion clips ranged from half a second up to ten seconds, leading to MT lengths between 15 and 300. The number of training motions used for each class ranged from 7 to 26. Using 3 to 7 iterations, it took on average 7.5 s to compute a class MT on a 3.6 GHz Pentium 4 with 1 GB of main memory, see Table 1. For example, for the class 'RotateRArmFwd3', the total computation time was $t(\mathcal{C}) = 3.9$ s with $\bar{\alpha} = 83.6$, $N = 8$, and $M = 4$, whereas for the class 'CartwheelLeft', it took $t(\mathcal{C}) = 17.0$ s with $\bar{\alpha} = 105.3$, $N = 11$, and $M = 6$.

4.3. Examples

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 'CartwheelLeft' as a representative example. Fig. 4 shows the cartwheel MT learned from $N = 11$ example motions, which form a superset of the motions shown in Fig. 1. Recall

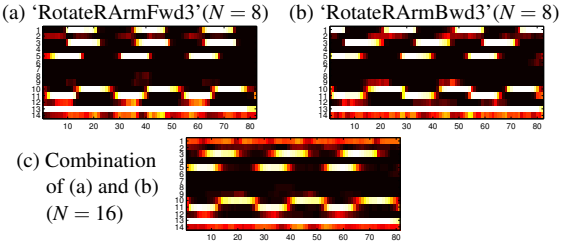

Figure 5: *Class MTs (only shown for the upper feature set, see Table 6) for arm rotation motions, three repetitions each.*

that black/white regions in a class MT correspond to consistent aspects of the training motions, while gray/colored regions correspond to variable aspects. The following observations illustrate that the essence of the cartwheel motion is captured by our class MT. Considering the regions marked by boxes in Fig. 4, 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 6). 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 Fig. 1. 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 Fig. 1. 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 Fig. 1.

The motion classes 'RotateRArmFwd3' and 'RotateRArmBwd3' stand for three repetitions of forward and backward rotation of the right arm, respectively. They are closely related, even though we do not consider them as logically similar. The respective class MTs are shown for the upper feature set in Fig. 5 (a) and (b). Even though the two class MTs exhibit a similar zero-one distribution, there is one characteristic difference: in the forward rotation, the right arm moves forwards (F_1) exactly when it is raised above the shoulder (F_3 is one). By contrast, in the backward rotation, the right arm moves forwards (F_1) exactly when it is below the shoulder (F_3 is zero). Using the training motions of both classes, it is possible to learn a single, combined MT, see Fig. 5 (c). Indeed, the resulting MT very well reflects the common characteristics as well as the disagreements of the two involved classes.

5. MT-based Matching for 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. Then, the key idea in designing a distance measure for comparing a class MT with unknown motion data is to mask out the inconsistent aspects such that related motions can be identified even in the presence of large variations. In Sect. 5.2, we define such a distance measure, which is based on DTW. Our experiments on MT-based annotation and retrieval are then described in Sect. 5.3–5.5.

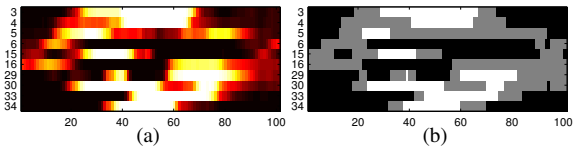

Figure 6: ***(a)*** *Class MT for 'CartwheelLeft' from Fig. 4.* ***(b)*** *The corresponding quantized MT.*

5.1. Classical DTW

To fix the notation, we summarize some basic facts on classical DTW. Let $X = (X(1), X(2), \ldots, X(K))$ and $Y = (Y(1), Y(2), \ldots, Y(L))$ be two feature sequences with $X(k), Y(\ell) \in \mathcal{F}$, $k \in [1:K]$, $\ell \in [1:L]$, where $\mathcal{F}$ denotes a feature space. In our case, we will have $\mathcal{F} = [0,1]^f$. Furthermore, let $c : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$ denote a *local cost measure* on $\mathcal{F}$. In the later discussion, c will denote the Manhattan distance. DTW is a standard technique to align X and Y with respect to the cost measure c. Recall that a *warping path* is a sequence $p = (p_1, \ldots, p_M)$ with $p_m = (k_m, \ell_m) \in [1:K] \times [1:L]$ for $m \in [1:M]$ satisfying the following conditions:

(i) Boundary condition: $p_1 = (1,1)$ and $p_M = (K,L)$.
(ii) Monotonicity condition: $1 \leq k_1 \leq k_2 \leq \ldots \leq k_M = K$ and $1 \leq \ell_1 \leq \ell_2 \leq \ldots \leq \ell_M = L$.
(iii) Step size condition: $p_{m+1} - p_m \in \{(1,0), (0,1), (1,1)\}$.

The total cost of p is defined as $\sum_{m=1}^{M} c(X(k_m), Y(\ell_m))$. Now, let p^* denote a warping path having minimal total cost among all possible warping paths. Then, the DTW distance $\mathrm{DTW}(X,Y)$ between X and Y is defined to be the total cost of p^*. It is well-known that p^* and $\mathrm{DTW}(X,Y)$ can be computed in $O(KL)$ using dynamic programming, see [RJ93].

5.2. MT-based Matching

In order to compare a class MT with the feature matrix resulting from an unknown motion data stream, we use a subsequence variant of DTW. The crucial point of our matching strategy is the local cost measure, which disregards the inconsistencies encoded in the class MT. To this end, we introduce a *quantized MT*, which has an entry 0.5 at all positions where the class MT indicates inconsistencies between different executions of a training motion within the same class. More precisely, let δ, $0 \leq \delta < 0.5$, be a suitable threshold. Then for an MT $X \in [0,1]^{f \times K}$, we define the quantized MT by replacing each entry of X 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 $\delta = 0.1$. See Fig. 6 for an example of a quantized MT.

Now, let D be a motion data stream. The goal is to identify subsegments of D that are similar to a given motion class $\mathcal{C}$. Let X be a quantized class MT of length K and Y the feature matrix of D of length L. We define for $k \in [1:K]$ and $\ell \in [1:L]$ a local cost measure $c^{\mathrm{Q}}(k,\ell)$ between the vectors $X(k)$ and $Y(\ell)$. 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

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

otherwise we set $c^{\mathrm{Q}}(k,\ell) = 0$. In other words, $c^{\mathrm{Q}}(k,\ell)$ only accounts for the consistent entries of X with $X(k)_i \in \{0,1\}$ and leaves the other entries unconsidered. Furthermore, to avoid degenerations in the DTW alignment, we use the modified step size condition $p_{m+1} - p_m \in \{(2,1),(1,2),(1,1)\}$, cf. (iii) of Sect. 5.1. This forces the slope of the warping path to assume values between $\frac{1}{2}$ and 2. Then, the distance function $\Delta_{\mathcal{C}} : [1:L] \to \mathbb{R} \cup \{\infty\}$ is defined by

$$\Delta_{\mathcal{C}}(\ell) := \frac{1}{K} \min_{a \in [1:\ell]} \Big(\mathrm{DTW}\big((X,\alpha), Y(a:\ell)\big) \Big), \qquad (2)$$

where $Y(a:\ell)$ denotes the submatrix of Y consisting of columns a through $\ell \in [1:L]$. (Due to the modified step size condition, some of the DTW distances in (2) may not exist, which are then set to ∞.) Note that the function $\Delta_{\mathcal{C}}$ can be computed by a standard subsequence DTW, see [RJ93]. Furthermore, one can derive from the resulting DTW matrix for each $\ell \in [1:L]$ the index $a_\ell \in [1:\ell]$ that minimizes (2). The interpretation of $\Delta_{\mathcal{C}}$ is as follows: a small value $\Delta_{\mathcal{C}}(\ell)$ for some $\ell \in [1:L]$ indicates that the motion subsegment of D starting at frame a_ℓ and ending at frame ℓ is similar to the motions of the class $\mathcal{C}$. Note that using the local cost function c^{Q} of (1) based on the quantized MT (instead of simply using the Manhattan distance c) is of crucial importance, as illustrated by Fig. 7. Further examples are discussed in Sect. 5.3.

5.3. MT-based Annotation

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 Fig. 8. For $\mathcal{C}_1$, there are four local minima between frames 100 and 300, which match the template with a cost of nearly zero and exactly correspond to the four jumping jacks contained in D, see Fig. 8 (a). Note that the remaining portion of D is clearly separated by Δ_1, yielding a value far above 0.1. Analogously, the two minima in Fig. 8 (b) and the four minima in Fig. 8 (c) correspond to the two repetitions of the elbow-to-knee exercise and the four squats, respectively. The choice of suitable quality thresholds for Δ_p as well as an evaluation of our experiments will be discussed in the next sections.

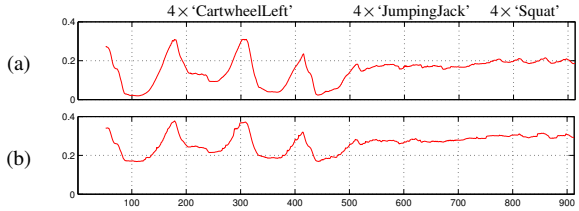

Figure 7: *(a) Distance function $\Delta_{\mathcal{C}}$ based on c^{Q} of (1) for the quantized class MT 'CartwheelLeft' and a motion sequence D consisting of four cartwheel (reflected by the four local minima close to zero), four jumping jacks, and four squats. (b) Corresponding distance function based on the Manhattan distance without MT quantization, leading to a much poorer result.*

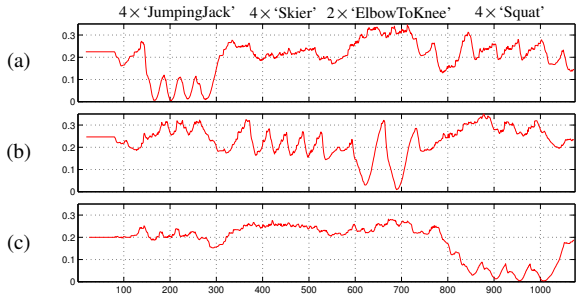

Figure 8: *Resulting distance functions for a 35-second gymnastics sequence consisting of four jumping jacks, four repetitions of a skiing coordination exercise, two repetitions of an alternating elbow-to-knee motion, and four squats with respect to the class MTs for (a) 'JumpingJack', (b) 'ElbowToKnee', and (c) 'Squat'.*

5.4. MT-based Retrieval

The goal of content-based motion retrieval is to automatically extract all logically similar motions of some specified type scattered in a motion capture database $\mathcal{D}$. By concatenating all documents of $\mathcal{D}$, we may assume that the database is represented by one single motion data stream D. To retrieve all motions represented by a class $\mathcal{C}$, we compute the distance function $\Delta_{\mathcal{C}}$ with respect to the precomputed class MT. Then, each local minimum of $\Delta_{\mathcal{C}}$ below some quality threshold $\tau > 0$ indicates a hit. To determine a suitable threshold τ and to measure the retrieval quality, we conducted extensive experiments based on several databases. We start with the evaluation database $\mathcal{D}^{\mathrm{MCE}}$, which consists of 718 motion clips corresponding to 24 minutes of motion data, see Sect. 4.2. Recall that $\mathcal{D}^{\mathrm{MCE}}$ is disjoint to the training database $\mathcal{D}^{\mathrm{MCT}}$, from which the class MTs were derived.

Fixing a quality threshold τ, we computed a set H_τ of hits for each of the 64 class MTs in a fully automated batch mode. Based on a manually generated annotation of $\mathcal{D}^{\mathrm{MCE}}$ used as the ground truth, we then determined the subset $H_\tau^+ \subseteq H_\tau$ of relevant hits corresponding to motion clips of the respective class. Table 2 shows some representative retrieval results for six different choices of τ. For example, for the motion class 'ClapHandsAboveHead' and the quality threshold $\tau = 0.02$, all of the 7 resulting hits are relevant—only one clapping motion is missing. Increasing the quality threshold to $\tau = 0.04$, one obtains 16 hits containing all of the 8 relevant hits. However, one also obtains 8 false positives, mainly coming from the jumping jack class, which contains a similar arm movement. The precision and recall values are very good for whole-body motions such as 'JumpingJack', 'Cartwheel', or 'LieDownFloor'—even in the presence of large variations within a class. Short motions with few characteristic aspects such as the class 'GrabHighRHand' are more problematic. For $\tau = 0.04$, one obtains 49 hits containing 12 of the 14 relevant movements. Confusion arises mainly with similar classes such as 'DepositHighRHand' or 'GrabMiddleRHand' and with subsegments of more complex motions containing a grabbing-like component such as the beginning of a cartwheel. Even from a semantical point of view, it is hard to distinguish such motions. Similar confusion arises with increasing values of τ for the kicking, walking/jogging, rotation, or sitting classes. However, most of the relevant hits could be found among the top ranked hits in all cases. For the classes 'RotateRArmFwd1' and 'RotateRArmBwd1', see Fig. 5 (a) and (b), all relevant movements could be correctly identified. Using a combined MT, as indicated by Fig. 5 (c), the two classes could not be distinguished any longer—the characteristics that had separated the two classes were now regarded as mere variations and therefore masked out in the retrieval process.

The above experiments imply that the quality threshold $\tau = 0.06$ constitutes a good trade-off between precision and recall. Since the distance function $\Delta_\mathcal{C}$ yields a ranking of the retrieved hits in a natural way, our strategy is to allow for some false positives rather than to have too many false negatives. Furthermore, note that the running time of MT-based retrieval depends linearly on the size of the database, where the bottleneck is the computation of the distance function $\Delta_\mathcal{C}$. For example, in case of the 24-minute database $\mathcal{D}^{\mathrm{MCE}}$ it took 4–28 seconds to process one query—depending on the respective MT length and the number of hits, see Table 2.

To speed up the retrieval on large databases, we introduce a keyframe-based preselection step to cut down the set of candidate motions prior to computing $\Delta_\mathcal{C}$. More precisely, for each class MT a small number of characteristic columns is labeled as keyframes. In our experiments, this labeling was done automatically using a simple heuristic: we basically picked two to five columns from the quantized MT that had many "white" entries (i. e., entries close to one, indicating some consistent action) and few "gray" entries

Motion Class $\mathcal{C}$	N	$\lvert H_\tau\rvert$ / $\lvert H_\tau^+\rvert$						K	$t(\Delta_\mathcal{C})$
CartwheelLeft	10	1 1	4 4	6 6	8 8	9 9	10 10	106	12.97
ClapHandsAboveHead	8	5 5	7 7	16 8	39 8	61 8	81 8	25	4.14
ElbowToKnee	13	8 8	11 11	12 12	13 13	13 13	22 13	36	4.19
GrabFloorRHand	8	6 5	8 7	11 7	20 8	41 8	75 8	61	8.36
GrabHighRHand	14	15 7	22 9	49 12	58 13	115 14	201 14	68	11.39
HopBothLegs	18	13 13	19 17	22 17	32 18	126 18	334 18	24	6.56
HopRLeg	21	17 17	19 19	22 21	35 21	66 21	107 21	18	3.08
JogRightCircleRFootStart4	8	6 4	13 8	37 8	41 8	53 8	74 8	59	7.73
JumpingJack	26	25 25	26 26	26 26	26 26	33 26	40 26	34	4.11
KickFrontRFoot	15	3 3	6 5	26 12	90 12	239 13	385 14	54	13.70
KickSideRFoot	15	6 5	13 9	27 14	48 14	163 15	359 15	51	12.88
LieDownFloor	10	4 4	6 6	8 8	8 8	9 9	11 9	172	28.05
RotateRArmBwd1	8	6 6	6 6	7 7	34 8	70 8	151 8	27	5.16
RotateRArmFwd1	8	6 6	6 6	7 7	39 8	77 8	186 8	28	6.13
RotateRArm(Bwd&Fwd)1	16	12 12	12 12	39 13	101 15	235 16	453 16	26	9.95
SitDownChair	10	4 4	9 8	17 10	29 10	53 10	70 10	83	12.92
SitDownFloor	10	9 4	15 6	25 9	34 10	48 10	61 10	106	15.78
SkierLeftFootStart	15	12 12	13 13	15 15	16 15	25 15	56 15	36	4.80
Squat	26	23 23	24 24	26 26	26 26	26 26	27 26	48	5.69
WalkFwdRFootStart4	8	17 7	21 7	25 8	44 8	69 8	131 8	82	11.42
WalkBwdRFootStart4	7	6 6	7 7	7 7	24 7	72 7	101 7	97	13.41
WalkSideRight3	8	7 7	8 8	11 8	14 8	28 8	49 8	123	16.08

Table 2: *Representative retrieval results for the evaluation database $\mathcal{D}^{\mathrm{MCE}}$ for various class MTs. Note that $\mathcal{D}^{\mathrm{MCE}}$ is disjoint to the training database $\mathcal{D}^{\mathrm{MCT}}$, from which the class MTs were derived. N denotes the number of relevant motions contained in $\mathcal{D}^{\mathrm{MCE}}$. $|H_\tau|$ (first rows) denotes the number of hits and $|H_\tau^+|$ (second rows) the number of relevant hits with respect to the quality thresholds $\tau = 0.01$, 0.02, 0.04, 0.06, 0.08, and 0.1. Finally, K denotes the length of the class MT and $t(\Delta_\mathcal{C})$ the running time in seconds required to compute the respective distance function $\Delta_\mathcal{C}$.*

(i. e., entries indicating inconsistencies). Then, in the preprocessing step, we extract the motion segments that contain the specified keyframes in the correct order within suitable time bounds. This preselection can be done efficiently using standard indexing techniques with inverted lists as described in [MRC05]. The computation of the distance function $\Delta_\mathcal{C}$ is then performed only on the preselected motion segments. We applied this strategy to our 210-minute database $\mathcal{D}_{210}$, which was introduced in Sect. 4.2. Some retrieval results as well as running times are summarized in Table 3 (upper). To assess retrieval quality, we manually inspected the set $H_{0.06}$ of hits as well as the database $\mathcal{D}_{210}$ for each class to determine the set $H_{0.06}^+$ of relevant hits. For example, the database $\mathcal{D}_{210}$ contains 24 left cartwheels. Using two automatically determined keyframes, it took 20 milliseconds to reduce the data from 210 to 2.8 minutes—1.3% of the original data. Then, MT retrieval was performed on the preselected 2.8 minutes of motion, which resulted in 21 hits and took 0.83 seconds. These hits contained 20 of the 24 cartwheels.

Even though keyframes are a powerful tool to significantly cut down the search space, there is also an attached risk: one single inappropriate keyframe may suffice to produce a large number of false negatives. For exam-

Motion Class $\mathcal{C}$	#(kf)	sel (m)	sel (%)	t(kf)	N	$\lvert H_{0.06}\rvert$	$\lvert H^{+}_{0.06}\rvert$	$t(\Delta_{\mathcal{C}})$
CartwheelLeft	2	2.8	1.3%	0.02	24	21	20	0.83
ElbowToKnee	2	0.8	0.4%	0.03	29	16	16	0.13
GrabHighRHand	2	8.9	4.2%	0.14	30	128	30	2.77
JumpingJack	2	1.5	0.7%	0.09	52	50	50	0.19
LieDownFloor	2	15.3	7.2%	0.06	20	24	16	4.42
RotateRArmFwd1	2	0.5	0.2%	0.48	66	6	5	0.17
SitDownChair	2	16.2	7.6%	0.11	20	27	4	3.00
Squat	2	2.2	1.1%	0.08	56	55	55	0.33

Motion Class $\mathcal{C}$	#(kf)	sel (m)	sel (%)	t(kf)	N	$\lvert H_{0.06}\rvert$	$\lvert H^{+}_{0.06}\rvert$	$t(\Delta_{\mathcal{C}})$
GrabHighRHand	3	3.2	1.5%	0.16	30	59	30	1.08
RotateRArmFwd1	3	1.0	0.5%	0.33	66	32	32	0.63
SitDownChair	3	3.8	1.8%	0.17	20	34	16	1.28

Table 3: ***Upper:*** *Retrieval results for the database $\mathcal{D}_{210}$ and $\tau = 0.06$ based on automatic keyframe selection. The second to fourth columns indicate the number of keyframes, the size of the preselected data set in minutes and percent as well as the running time for the preprocessing step. N is the number of relevant motions in $\mathcal{D}_{210}$. $\lvert H_{0.06}\rvert$ and $\lvert H^{+}_{0.06}\rvert$ denote the number of hits and the number of relevant hits, respectively. $t(\Delta_{\mathcal{C}})$ indicates the running time in seconds required to compute $\Delta_{\mathcal{C}}$ on the preselected motions.* ***Lower:*** *Retrieval results for manually selected keyframes.*

ple, this happened for the classes listed in Table 3 (lower). For these classes, using more appropriate, manually selected keyframes led to a significant improvement. A further benefit of the keyframe approach is that the large number of false positives, as typical for short and unspecific motions, can be easily cut down by adding a single keyframe. See, for example, the motion class 'GrabHighRHand' in Table 3 (upper). For future work, we plan to improve our ad-hoc method of keyframe selection. To this end, we have conducted first experiments to automatically learn characteristic keyframes from positive and negative motion examples employing a strategy based on genetic algorithms. It would also be possible to use similar methods as described in [ACCO05]

As a further test, we used the 180-minute database $\mathcal{D}^{\text{CMU}}_{180}$ containing motion capture material from the CMU database [CMU03]. Similar results and problems can be reported as for $\mathcal{D}_{210}$. Interestingly, our class MT X for 'CartwheelLeft' yielded no hits at all—as it turned out, all cartwheels in $\mathcal{D}^{\text{CMU}}_{180}$ are right cartwheels. We modified X by simply interchanging the rows corresponding to feature pairs pertaining to the right/left part of the body, see Table 6. Using the resulting *mirrored* MT, four out of the known five cartwheels in $\mathcal{D}^{\text{CMU}}_{180}$ appeared as the only hits. Due to their semantic meaning, class MTs can easily be modified in an intuitive way without any additional training data. Even designing a class MT from scratch (without resorting to any training motions) proved to be feasible. For example, to identify 'sweeping with a hand brush' in $\mathcal{D}^{\text{CMU}}_{180}$, we defined an MT of length 50, setting all matrix entries to 0.5 except for the rows corresponding to F_{13} (right hand fast), F_{32} (spine horizontal), and F_{33} (right hand lowered), which were set to one. Eight out of ten hits in $\mathcal{D}^{\text{CMU}}_{180}$ were relevant.

5.5. Comparison to Other Retrieval Methods

We compared our MT-based retrieval system to several baseline methods using subsequence DTW on raw motion capture data with suitable local distance measures. It turned out that such baseline methods show little or no generalization capability. The database (3.8 minutes, or 6,750 frames sampled at 30 Hz) consisted of 100 motion clips: ten different realizations for each of ten different motion classes. For each of the ten motion classes, we performed motion retrieval in four different ways:

(MT) retrieval using a quantized class MT,
(RF) DTW using the relational feature matrix of a single example motion and Manhattan distance,
(Q) DTW using unit quaternions and spherical geodesic distance,
(3D) DTW using 3D joint coordinates (normalized w. r. t. root rotation and size) and Euclidean distance.

Motion Class	$r^{\text{MT}}_{5/10/20}$	s^{MT}	$r^{\text{RF}}_{5/10/20}$	s^{RF}	$r^{\text{Q}}_{5/10/20}$	s^{Q}	$r^{\text{3D}}_{5/10/20}$	s^{3D}
CartwheelLeft	5/10/10	12.83	5/10/10	1.62	4/6/7	1.63	1/1/2	2.38
Squat	5/10/10	259.5	5/10/10	16.1	5/7/9	2.79	4/6/7	2.52
LieDownFloor	5/9/10	11.65	5/9/10	2.10	4/7/9	1.69	2/3/7	1.29
SitDownFloor	4/6/10	19.33	3/4/8	1.60	2/5/7	2.13	3/5/8	1.56
GrabHighRHand	5/7/9	33.93	5/8/8	9.72	3/5/8	3.39	1/3/4	2.22

Table 4: *Recall values (r) in the top 5/10/20 ranked hits and separation quotients (s) for different DTW-based retrieval methods: motion templates (MT), relational feature matrices (RF), quaternions (Q), and relative 3D coordinates (3D).*

For each strategy, we computed a Δ curve as in Fig. 7 and derived the top 5, top 10, and top 20 hits. Table 4 shows the resulting recall values (note that there are exactly 10 correct hits for each class) for five representative queries. As a further important quality measure of a strategy, we computed the *separation quotient*, denoted by s, which is defined as the median of Δ divided by the median of the cost of the correct hits among the top 10 hits. The larger the value of s, the better the correct hits are separated from the false positives, enabling the usage of simple thresholding strategies on Δ for the retrieval. Only for our MT-based strategy, the separation is good enough. These observations indicate that MT-based retrieval outperforms the other methods.

We also compared MT-based retrieval to the method by Müller et al. [MRC05]. Their system is based on fuzzy queries, and the performance heavily depends on the query formulation, which involves manual specification of a query-dependent feature selection. For each query, we carefully selected a suitable subset of features, which proved to be a time-consuming process. The resulting precision/recall values on $\mathcal{D}^{\text{MC}}$ are very good and reflect what seems to be achievable by their technique, see Table 5. For MT-based retrieval, we quote precision/recall values for two quality thresholds, $\tau = 0.02$ and $\tau = 0.06$. Our experiments show that the retrieval quality of our fully automatic MT-based approach is in most cases as good and in many cases even better than that obtained by Müller et al. [MRC05], even after hand-tweaking their parameters. Hence, our MT-based approach enables us to replace manual, global feature selection by fully automatic, local feature selection without loss of retrieval quality.

Motion Class	$r^{\mathrm{MT}}_{0.02}$	$p^{\mathrm{MT}}_{0.02}$	$r^{\mathrm{MT}}_{0.06}$	$p^{\mathrm{MT}}_{0.06}$	r^{Fuz}	p^{Fuz}
CartwheelLeft	0.57	1.00	0.90	1.00	1.00	1.00
HopBothLegs	0.94	0.89	1.00	0.52	0.58	0.62
HopRLeg	0.95	1.00	1.00	0.57	0.79	0.18
LieDownFloor	0.70	1.00	0.85	0.81	0.95	0.90
WalkBwdRFootStart4	1.00	1.00	1.00	0.26	0.53	0.36

Table 5: *Precision (p) and recall (r) values for representative queries, comparing fuzzy retrieval (Fuz) vs. MT-based retrieval. The subscript indices for MT-based retrieval indicate the value of* τ.

6. Conclusions and Future Work

In this paper, we introduced the concept of a motion template, which encodes the characteristic and the variable aspects of a motion class. We proposed an automatic procedure to learn a class MT from example motions. As a further contribution, we applied class templates to motion annotation and retrieval. By automatically masking out the variable aspects of a motion class in the retrieval process, related motions can be identified even in the presence of large variations and without any user intervention. Extensive experimental results show that our methods work with high precision and recall for whole-body motions and for longer motions of at least a second. More problematic are very short and unspecific motion fragments. Here, the use of suitably defined keyframes is a promising concept to not only speed up the retrieval process, but also to eliminate false positives. For future work, we plan to continue our experiments with genetic algorithms to extract characteristic keyframes based on our template representation. In collaboration with the HDM school of media sciences (Stuttgart), we investigate how motion templates may be used as a tool for specifying animations—replacing a keyframe-based by a template-based animation concept—similar to approaches such as [RCB98, AFO03].

A. Test Feature Set for Full-body Motion

In this paper, we rely on $f = 39$ relational features, see Table 6. Our features are derived from a small number of *generic* relational features, which encode certain joint constellations in 3D space and time. Specifically, we used the following generic features: $F_{\mathrm{plane}} = F^{(j_1,j_2,j_3;j_4)}_{\theta,\mathrm{plane}}$, $F_{\mathrm{nplane}} = F^{(j_1,j_2,j_3;j_4)}_{\theta,\mathrm{nplane}}$, $F_{\mathrm{angle}} = F^{(j_1,j_2;j_3,j_4)}_{\theta,\mathrm{angle}}$, $F_{\mathrm{move}} = F^{(j_1,j_2;j_3)}_{\theta,\mathrm{move}}$, $F_{\mathrm{nmove}} = F^{(j_1,j_2,j_3;j_4)}_{\theta,\mathrm{nmove}}$, and $F_{\mathrm{fast}} = F^{(j_1)}_{\theta,\mathrm{fast}}$. Each of these features maps a given pose to the set $\{0,1\}$ and depends on a set of joints, denoted by $j_1, j_2, \ldots$, as well as on a threshold value or threshold range θ. For the time being, we identify θ with θ_1, as specified in the eighth column of Table 6—the meaning of θ_2 will be explained below.

The first generic feature F_{plane} 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. For example, setting j_1 ='root', j_2 ='lhip', j_3 ='ltoes', j_4 ='rankle', one obtains the feature F_{15}, see Table 6 (a). A similar test is performed by $F^{(j_1,j_2,j_3;j_4)}_{\theta,\mathrm{nplane}}$, but here we define the plane in terms of a normal vector (given by j_1 and j_2), and fix it at j_3, see Table 6 (d). The generic feature $F^{(j_1,j_2;j_3,j_4)}_{\theta,\mathrm{angle}}$ 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 $\theta \subset \mathbb{R}$, as indicated by Table 6 (b). The remaining three generic features operate on velocity data that is approximated from the 3D joint trajectories of the input motion: $F^{(j_1,j_2;j_3)}_{\theta,\mathrm{move}}$ 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, see, for example, Table 6 (c). The generic feature $F^{(j_1,j_2,j_3;j_4)}_{\theta,\mathrm{nmove}}$ 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. The generic feature $F^{(j_1)}_{\theta,\mathrm{fast}}$ assumes the value one iff joint j_1 has an absolute velocity above θ.

The simple quantization scheme using only the threshold θ as described for the generic features is prone to strong output fluctuations if the input value fluctuates slightly around the threshold. To alleviate this problem, we employ a robust quantization strategy using two thresholds, θ_1 and θ_2, together with a hysteresis-like thresholding scheme, which effectively suppresses most unwanted zero-one fluctuations.

Acknowledgements: We thank Bernd Eberhardt from HDM Stuttgart for providing us with the mocap data, Michael Clausen for constructive and valuable comments, and the CMU motion lab for sharing their skeleton files. Some of the data was obtained from [CMU03], which was created with funding from NSF EIA-0196217. Tido Röder is supported by the German National Academic Foundation.

References

[ACCO05] ASSA J., CASPI Y., COHEN-OR D.: Action synopsis: pose selection and illustration. *ACM TOG 24*, 3 (2005), 667–676.

[AFO03] ARIKAN O., FORSYTH D. A., O'BRIEN J. F.: Motion synthesis from annotations. *ACM TOG 22*, 3 (2003), 402–408.

[BH00] BRAND M., HERTZMANN A.: Style machines. In *Proc. ACM SIGGRAPH* (2000), pp. 183–192.

[BSP*04] BARBIC J., SAFONOVA A., PAN J.-Y., FALOUTSOS C., HODGINS J., POLLARD N.: Segmenting motion capture data into distinct behaviors. In *GI '04: Proc. Graphics interface* (2004), Canadian Human-Comp. Comm. Soc., pp. 185–194.

[BW95] BRUDERLIN A., WILLIAMS L.: Motion signal processing. In *Proc. ACM SIGGRAPH* (1995), pp. 97–104.

[CH05] CHAI J., HODGINS J.: Performance animation from low-dimensional control signals. *ACM TOG 24*, 3 (2005), 686–696.

[CMU03] CMU: `http://mocap.cs.cmu.edu`, 2003.

[FF05] FORBES K., FIUME E.: An efficient search algorithm for motion data using weighted PCA. In *Proc. ACM SIGGRAPH/Eurographics SCA* (2005), pp. 67–76.

[FMJ02] FOD A., MATARIC M. J., JENKINS O. C.: Automated derivation of primitives for movement classification. *Auton. Robots 12*, 1 (2002), 39–54.

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}	$(0,0,0)^\top$	$(0,1,0)^\top$	$(0,Y_{\text{min}},0)^\top$	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$\rightarrow$lhip. Test: rankle closer to Π than lankle?						feet crossed over
F_{23}	ℓ		Consider velocity v of rankle relative to lankle in rankle$\rightarrow$lankle direction. Test: projection of v onto rhip$\rightarrow$lhip line large?						feet moving towards each other, sideways
F_{24}	ℓ		Same as above, but use lankle$\rightarrow$rankle instead of rankle$\rightarrow$lankle direction.						feet moving apart, sideways
F_{25}/F_{26}	ℓ	F_{fast}	rankle				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	$(0,0,0)^\top$	$(0,1,0)^\top$	$[70^\circ, 110^\circ]$	$[60^\circ, 120^\circ]$	spine horizontal
F_{33}/F_{34}	m	F_{nplane}	$(0,0,0)^\top$	$(0,-1,0)^\top$	$(0,Y_{\text{min}},0)^\top$	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

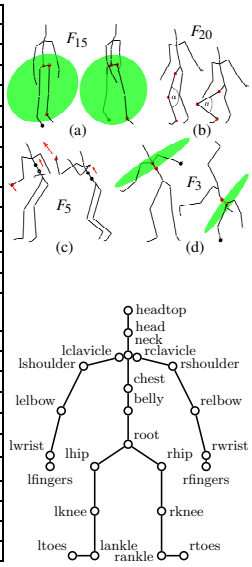

Table 6: ***Left:*** *the 39 relational features used in our experiments, divided into the sets "upper", "lower", and "mix", which are abbreviated as u, ℓ and m, respectively. Features with two entries in the ID column exist in two versions pertaining to the right/left half of the body but are only described 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 $(X,Y,Z)^\top$ world system in which the Y axis points upwards. The symbols $Y_{\text{min}}/Y_{\text{max}}$ 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.* ***Right, lower:*** *skeletal kinematic chain model consisting of rigid* body segments *flexibly connected by* joints*, which are highlighted by circular markers and labeled with joint names.* ***Right, upper:*** *illustration of selected relational features. The relevant joints are indicated by enlarged markers.*

[GG04] GREEN R., GUAN L.: Quantifying and recognizing human movement patterns from monocular video images: Part I. *IEEE Transactions on Circuits and Systems for Video Technology 14*, 2 (February 2004), 179–190.

[GP00] GIESE M., POGGIO T.: Morphable models for the analysis and synthesis of complex motion patterns. *IJCV 38*, 1 (2000), 59–73.

[HPP05] HSU E., PULLI K., POPOVIĆ J.: Style translation for human motion. *ACM TOG 24*, 3 (2005), 1082–1089.

[KG03] KOVAR L., GLEICHER M.: Flexible automatic motion blending with registration curves. In *Proc. ACM SIGGRAPH/Eurographics SCA* (2003), pp. 214–224.

[KG04] KOVAR L., GLEICHER M.: Automated extraction and parameterization of motions in large data sets. *ACM TOG 23*, 3 (2004), 559–568.

[KPZ*04] KEOGH E. J., PALPANAS T., ZORDAN V. B., GUNOPULOS D., CARDLE M.: Indexing large human-motion databases. In *Proc. 30th VLDB Conf.* (2004), pp. 780–791.

[LZWM05] LIU G., ZHANG J., WANG W., MCMILLAN L.: A system for analyzing and indexing human-motion databases. In *Proc. ACM SIGMOD* (2005), pp. 924–926.

[MRC05] MÜLLER M., RÖDER T., CLAUSEN M.: Efficient content-based retrieval of motion capture data. *ACM TOG 24*, 3 (2005), 677–685.

[PB02] PULLEN K., BREGLER C.: Motion capture assisted animation: Texturing and synthesis. In *Proc. ACM SIGGRAPH* (2002), pp. 501–508.

[RCB98] ROSE C., COHEN M. F., BODENHEIMER B.: Verbs and adverbs: Multidimensional motion interpolation. *IEEE Comp. Graph. Appl. 18*, 5 (1998), 32–40.

[RF03] RAMANAN D., FORSYTH D. A.: Automatic annotation of everyday movements. In *NIPS 16* (2003).

[RJ93] RABINER L. R., JUANG B. H.: *Fundamentals of Speech Recognition*. Prentice Hall Signal Processing Series, 1993.

[SKK04] SAKAMOTO Y., KURIYAMA S., KANEKO T.: Motion map: image-based retrieval and segmentation of motion data. In *Proc. ACM SIGGRAPH/Eurographics SCA* (2004), pp. 259–266.

[WCYL03] WU M.-Y., CHAO S., YANG S., LIN H.: Content-based retrieval for human motion data. In *16th IPPR Conf. on Comp. Vision, Graph., and Image Proc.* (2003), pp. 605–612.

[WP95] WITKIN A., POPOVIĆ Z.: Motion warping. In *Proc. ACM SIGGRAPH* (1995), pp. 105–108.

[ZMCF05] ZORDAN V. B., MAJKOWSKA A., CHIU B., FAST M.: Dynamic response for motion capture animation. *ACM TOG 24*, 3 (2005), 697–701.