PAPER

Content-Based Retrieval of Motion Capture Data Using Short-Term Feature Extraction

Jianfeng XU[†a)], ***Nonmember***, **Haruhisa KATO**[†b)], ***and*** **Akio YONEYAMA**[†c)], ***Members***

SUMMARY This paper presents a content-based retrieval algorithm for motion capture data, which is required to re-use a large-scale database that has many variations in the same category of motions. The most challenging problem is that logically similar motions may not be numerically similar due to the motion variations in a category. Our algorithm can effectively retrieve logically similar motions to a query, where a distance metric between our novel short-term features is defined properly as a fundamental component in our system. We extract the features based on short-term analysis of joint velocities after dividing an entire motion capture sequence into many small overlapped clips. In each clip, we select not only the magnitude but also the dynamic pattern of the joint velocities as our features, which can discard the motion variations while keeping the significant motion information in a category. Simultaneously, the amount of data is reduced, alleviating the computational cost. Using the extracted features, we define a novel distance metric between two motion clips. By dynamic time warping, a motion dissimilarity measure is calculated between two motion capture sequences. Then, given a query, we rank all the motions in our dataset according to their motion dissimilarity measures. Our experiments, which are performed on a test dataset consisting of more than 190 motions, demonstrate that our algorithm greatly improves the performance compared to two conventional methods according to a popular evaluation measure $P(N_R)$.

key words: *motion capture, content-based retrieval, short-term feature, distance metric, motion similarity, dynamic time warping*

1. Introduction

Motion capturing has become an increasingly popular way of creating motions for such applications as computer animation, video games, and movies [1]. Large databases are created in entertainment companies and are even available on the Internet (e.g. CMU Motion Capture (*MoCap*) Database [2]). Thus, management and re-use of MoCap data are becoming increasingly important [3], where retrieval of MoCap data is a basic function for exploiting many variations of the same category of motions in the database. Effective and efficient retrieval of MoCap data can provide users with flexibility in selecting appropriate motions to generate rich contents with subtle variations.

In this paper, a motion retrieval algorithm is proposed to search a database for all similar motions to a query. Basically, logically similar motions are preferred to numerically similar motions. Namely, we search the motions within the same category as a query no matter how these motions may vary. As described in [4] and [5], the major challenge in content-based retrieval of MoCap data is that MoCap sequences with logically similar motions may not be numerically similar. In other words, motion signals differ greatly in joint trajectories, joint angles, joint velocities, and so forth due to the motion variations that are caused by personality, motion pace, and motion style. This may lead to some similar motions being dismissed and some dissimilar motions being retrieved. In this paper, we explore the logical similarity by defining a proper distance metric using our short-term features.

As MoCap data are a time series, we need to define a proper unit in MoCap data to analyze the logical similarity in the temporal domain. We observe that a stable structure of dynamic patterns exists in joint velocities for a category of motions as discussed in detail in Sect. 3. To reflect the dynamic pattern structure effectively and discard the motion variations (see Fig. 3), it is more effective and efficient to use a motion clip (a group of frames) than a single frame. Therefore, we divide an entire MoCap sequence into small motion clips with a fixed length in an overlapped manner and regard a motion clip as a basic unit to be processed. A dynamic pattern is extracted from joint velocities in each motion clip, which is formed as our short-term features together with an average value of joint velocities, see Fig. 4. The extracted features are robust to motion variations and also reduce the data size. A distance metric of features is defined to balance the two parts in our features. Such a distance metric is very useful in analyzing the MoCap sequence and plays a key role in motion retrieval, motion clustering, motion graphs, and motion blending, see [3]. Furthermore, the motion dissimilarity measure is calculated based on dynamic time warping due to the warping of the time axis in MoCap data. Dynamic time warping is a popular method for dealing with time warping signals such as MoCap data [6], [7]. A well-known problem of dynamic time warping is its quadratic computational complexity. Benefiting from our short-term features, we can greatly reduce the computational complexity, see Sect. 5.3.

Although symbolized representation of human motion has been studied before (e.g., Muller et al. [5]), our approach is different in that we consider temporal correlation of human motion instead of Muller's spatial relationship. Moreover, not only symbolized representation (dynamic pattern in our paper) but also continuous features (average speed) are extracted in our short-term features. As far as we know,

Manuscript received February 2, 2009.
Manuscript revised May 18, 2009.

[†]The authors are with KDDI R&D Laboratories Inc., Fujimino-shi, 356–8502 Japan.
a) E-mail: ji-xu@kddilabs.jp
b) E-mail: hkato@kddilabs.jp
c) E-mail: yoneyama@kddilabs.jp

DOI: 10.1587/transinf.E92.D.1657

this is the first time that a short-term feature has been employed in human motion and both discrete features and continuous features are extracted in human motion simultaneously.

We compare the proposed technique to two conventional methods in a test dataset from CMU MoCap Database [2]. One is the prevailing method [3], called "baseline" in this paper. The other is the most recently proposed technique, called "FMDistance" [8]. Our experimental results demonstrate that our algorithm improves the performance greatly compared to that of conventional methods. Namely, $P(N_R)$ (see explanation in Sect. 5.3 or see [9]) can reach 90.8% on average in our method, while it is 68.1% in the baseline and 77.0% in FMDistance.

The remainder of this paper is organized as follows. In Sect. 2, related works are briefly introduced focusing on the main components of a typical framework. In Sect. 3, we firstly describe MoCap data and the pre-processing to prepare the feature extraction. Then, we explain how to extract the short-term features in a MoCap sequence. In Sect. 4, we define a distance metric based on the short-term features and calculate motion dissimilarity measure using dynamic time warping. In Sect. 5, experimental results are reported and analyzed. Lastly, we present our conclusions and describe future work.

2. Related Work

Content-based retrieval is a powerful tool in multimedia databases including image, video, and audio databases, see the latest special issue [10]. The retrieval of MoCap data is attracting research interest due to increasing demand, see [3] and references therein.

In this section, we introduce related works following a typical framework for content-based retrieval of MoCap data, see Fig. 1. The main purpose of pre-processing is to prepare a suitable variable (e.g. quaternion in [11] and joint position in [5]), filter the variable (e.g. down sample in [12]),

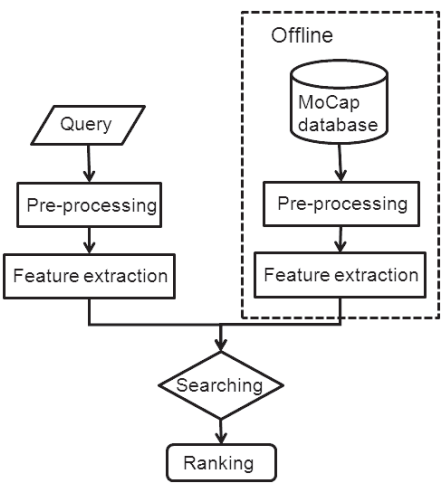

Fig. 1 A typical framework for content-based retrieval of MoCap data, which is also employed in this paper.

and reduce the independence among the high dimensions (e.g. PCA in [7]). In feature extraction, frequency has been popular in motion editing from an early stage [6] and in motion compression [13]. However, motion signals have rather complex frequency characteristics that limit the application of frequency methods. Muller et al. [5], [14], [15] propose a set of Boolean features expressing geometric relations between certain body points of a pose, which are robust to motion variations in motion retrieval. However, it requires considerable experience to choose the combination of Boolean features, restricting its usage in experts. It is quite intuitive that motion similarity should be measured by a combination of several frames as performed by some pioneering researchers, e.g., Kover et al. [16] propose "point clouds" in construction of their motion graphs and Arikan et al. [13] compress MoCap data regarding a motion clip as his unit. The idea of using short-term analysis such as motion clips has been intensively investigated in speech processing, see [17].

A motion dissimilarity measure is crucial for retrieval of MoCap data, which requires the necessary warping of the time axis. Although some new techniques have been developed in alignment of MoCap sequences such as uniform time scaling by Keogh et al. [18], dynamic time warping is commonly employed in MoCap data [4]–[7]. A recent approach "FMDistance", proposed by Onuma et al. [8], is based on the total kinetic energy in the MoCap sequence, which is compared with our algorithm in Sect. 5. It is very fast but is not able to analyze the subsequence. This restricts the use of "FMDistance" in applications such as motion structure analysis and makes it impossible to search subsequence in a motion. Guan et al. [19] compare several popular motion similarities in their technical report. Kovar et al. [4] present a smart multi-step search strategy to solve the motion variation problem, where those closer motions to the query are used as new queries to find more distant motions. However, a great deal of pre-processing time is required for sizable databases as pointed out in [3]. Sakamoto et al. [20] present an interesting interface for motion retrieval, where the user identifies key postures within a map of poses obtained from a self-organizing map algorithm. In this paper, we mainly focus on improving the accuracy of motion retrieval, removing the motion variations by our novel short-term features.

3. Symbolized and Continuous Short-Term Features

In this section, we describe how to extract short-term features after pre-processing MoCap data. Effective features are required to reflect the invariant information in a motion category while discarding its motion variations. For this purpose, we propose a method to extract short-term features from the joint velocities.

3.1 Focusing on Joint Velocities

MoCap data are a sequence of human poses (called *frames*

in this paper). Each frame is commonly modeled using a kinematic chain defining a stick figure of human pose. In most cases, a kinematic chain consists of root positions and orientation, and joint angles in local coordinate systems from a neutral pose. As shown in Fig. 2, there are many joints connected in a tree structure to shape a stick figure (only the root joint is marked with a white sphere and eight other joints are marked with gray spheres, which are used in our algorithm because arm and leg motions are the most perceptible). This data representation is suitable for rendering, controlling, and editing MoCap data but loses intuition. Assuming we have N frames, a MoCap sequence is denoted as $\{\mathbf{F}_t : t \in [0 : N-1]\}$, where $\mathbf{F}_t$ denotes the t-th frame in the sequence.

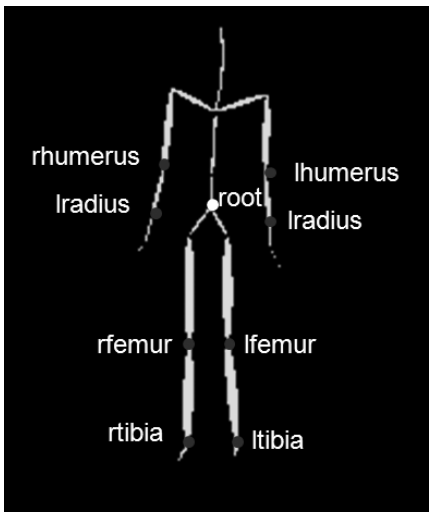

Fig. 2 Articulated model used in CMU MoCap data. There are 31 joints in total including end effectors and root, and 62 degrees of freedom in the stick figure, only eight joints are marked with gray spheres and the root joint is marked with a white sphere. For compatibility, we use the same names for joints as the CMU database does for bones.

The extracted features should reflect the motion essence to the extent possible with as little data as possible. This requires a perceptually meaningful variable and it should be invariant to coordinate transformation. Johansson's famous experiment on dots of light reveals the psychological fact that joint positions and velocities are essential for human beings to perceive a motion [21]. Therefore, instead of using joint angles directly, we employ joint velocities to extract motion features. The joint velocities are calculated by the first derivatives of joint positions, which can be transformed from a neutral pose and motion information including joint angles and root positions and orientation by forward kinematics [22]. To become invariant to coordinate transformation, relative velocities to the root joint are calculated. To reduce the data size further, the absolute value of relative velocity is used in feature extraction. Meanwhile, the direction of velocity is regarded as a kind of motion variants. For example, a motion punching to front direction is regarded to be similar to a motion punching to other direction. Therefore, just using the magnitude of velocity is useful to remove such motion variants. In the remainder of this paper, we refer to the absolute value of relative velocity for the k-th joint at the t-th frame as *joint speed*, denoted by $\{v_t^k : k \in [1 : K-1], t \in [1 : N-1], v_t^k \in \mathcal{R}^+\}$, where K denotes the number of the joints and $\mathcal{R}^+$ means the set of non-negative real numbers.

Despite the careful generation of MoCap data, there is still some noise in the raw data, which is enlarged by the differential in computing the joint speed. Therefore, we adopt a 5-tap median filter and then a 5-tap low pass filter to smooth the joint speeds in our implementation.

3.2 Short-Term Feature Extraction

Human motion is very complicated. As an example, Fig. 3

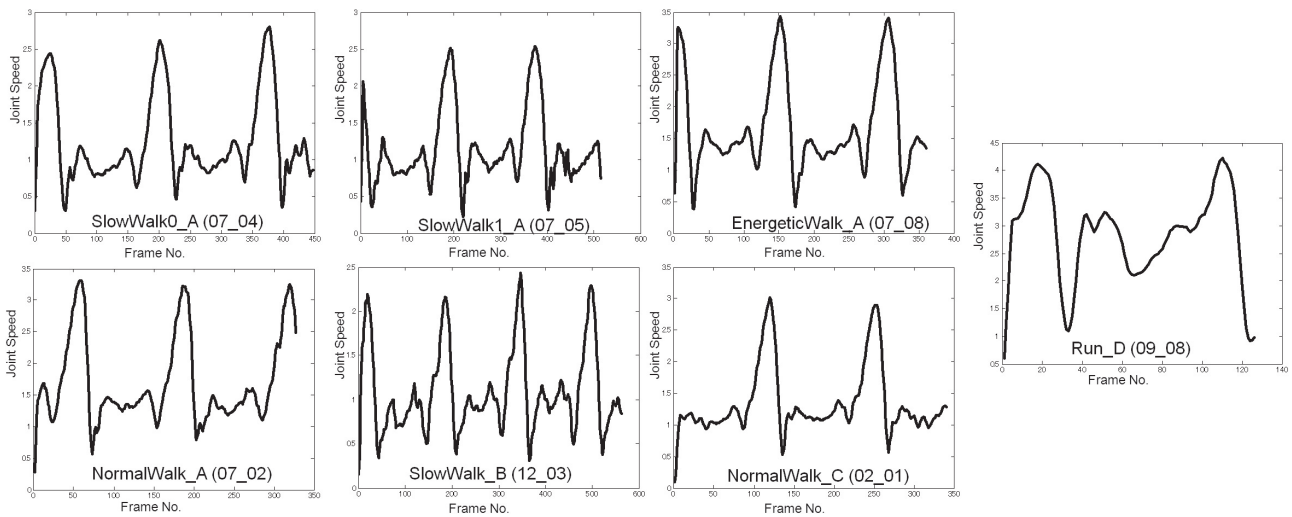

Fig. 3 Joint speeds of the left tibia in sample sequences, a stable dynamic pattern structure is observed in all the walking sequences although a huge motion variation exists due to the personality, motion pace, style, etc.. In the legend, the rule of the motion descriptor connected with the actor descriptor is used to name the curves, and actor IDs and motion IDs are connected with underline in the parentheses.

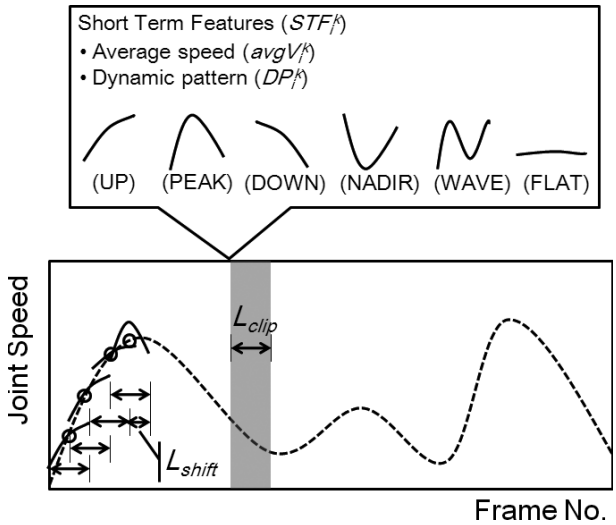

Fig. 4 A MoCap sequence is divided into small overlapped motion clips with a fixed length, and short-term features are extracted in each motion clip composing the average speed and dynamic pattern that includes six pre-defined templates.

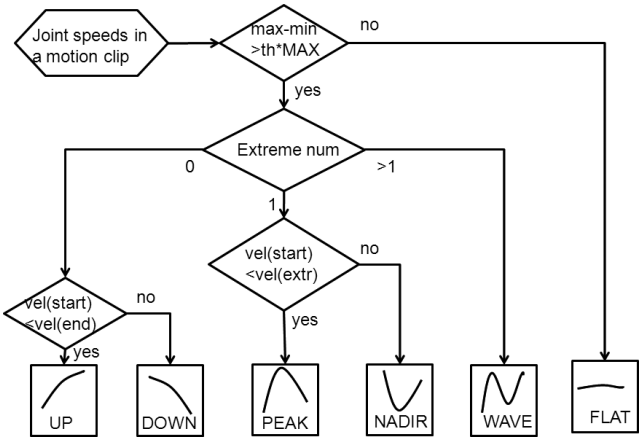

Fig. 5 A classifier of the dynamic pattern for a joint in a motion clip, where a motion clip is classified into one of the six patterns according to its joint speeds.

shows the joint speeds of the left tibia for six walking motions and a running motion. Different actors may walk very differently among the motion pace, style, and personality. And the same actor may walk in different styles such as normal walking, slow walking, and energetic walking. Even the different cycles of walking still have subtle variations, which make MoCap data very realistic. However, the dynamic pattern structure of joint speeds in a category of motions is rather stable. In Fig. 3, all walking motions have an invariant structure of peak and valley in joint speeds of the left tibia although the magnitudes of joint speeds are different and the places and durations of peak and valley vary greatly. This observation serves as a basic fact of short-term features that are robust to motion variations.

Inspired by the short-term processing of speech [17], we divide a MoCap sequence into many overlapped motion clips with a fixed number of frames, see Fig. 4. Namely, $L_{shift} = \lfloor L_{clip}/2 \rfloor$, where L_{shift} denotes the frame number of shifting one motion clip to the next, and L_{clip} denotes the length of the motion clip. Although using an adaptive length seems attractive, it is still difficult to obtain meaningful segmentations in MoCap data and thus limits its usage in feature extraction. On the other hand, a fixed length is widely applied in signal processing such as in audio analysis [17].

A joint's speeds in a motion clip are classified into six patterns by a decision tree: UP, PEAK, DOWN, NADIR, WAVE, and FLAT, see details in Fig. 5. In this paper, those six pre-defined templates are called *dynamic patterns*. It is clear that our dynamic pattern includes temporal information and is invariant to the magnitude of joint speeds in a motion clip. In Fig. 5, *max*/*min* denotes the maximum/minimum value of joint speeds in the motion clip, *th* is a threshold (set as 0.05 in our implementation), *MAX* is the maximum value in the joint speeds of the entire sequence, *Extreme num* means the number of extreme values in joint speeds in the motion clip, *vel*(*extr*) denotes the extreme value of joint speed when *Extreme num* is 1, and *vel*(*start*)/*vel*(*end*) denotes the joint speed in the first/last frame of the motion clip. Basically, we firstly decide whether the dynamic pattern is FLAT or not by the joint speed range. Then, according to the number of extreme values in the motion clip, we decide the dynamic pattern is WAVE (with more than one extreme) or PEAK/NADIR (only one extreme) or UP/DOWN (no extreme). Lastly, using the values of joint speeds, we can separate PEAK from NADIR or UP from DOWN. Note that as the dynamic pattern is dependent on the location of motion clip (e.g. PEAK may be divided into UP and DOWN if the location of the motion clip is changed), we propose an overlapped manner in dividing the MoCap sequence to alleviate the above ambiguity.

The average value of the joint speeds in the motion clip forms the second part of the short-term feature. This average value is very useful for distinguishing the different stages in the same dynamic pattern. Note that it has no problem in adding others such as deviation and energy to our feature set. However, the average value is sufficient in our experiments. After extracting the features for all the joints in all the motion clips, we can obtain our feature set.

$$STF_i^k = \{(DP_i^k, avgV_i^k) : k \in [1:K-1], i \in [0:I-1]\} \quad (1)$$

where STF_i^k denotes the short-term feature for the k-th joint in the i-th motion clip, DP_i^k means the dynamic pattern, $avgV_i^k$ denotes the average joint speed in the motion clip, and I is the number of motion clips (Obviously, $I = \lfloor (N - L_{clip})/L_{shift} + 1 \rfloor$). Note that the motion clip length L_{clip} is a parameter in our feature extraction.

4. Motion Dissimilarity Measure and Retrieval

In this section, we describe how to calculate the motion dissimilarity measure between two MoCap sequences based on

the extracted short-term features. As shown in Fig. 1, feature extraction in the dataset is an off-line process. Other processes are on-line including extracting the features of query sequence, and calculating the motion dissimilarity measures between the query and those MoCap data in the dataset.

4.1 Distance Metric of Features

Since our short-term feature consists of two parts, our distance metric consists of two corresponding parts. The first part is based on the dynamic patterns. If the two dynamic patterns DP_i^k and DP_j^k are exactly the same, their distance is set as 0. If they are half the same, their distance is set as 0.5. Otherwise, their distance is set as 1.0.

$$d(DP_i^k, DP_j^k) = \begin{cases} 0.0 & \text{if } DP_i^k = DP_j^k \\ 0.5 & \text{if } (DP_i^k, DP_j^k) \in \Pi \\ 1.0 & \text{others} \end{cases} \quad (2)$$

where the set Π is shown in Fig. 6 by those dynamic patterns with connections, which lists all 12 possible pairs. For example, (UP, PEAK) $\in \Pi$ and similarly (PEAK, UP) $\in \Pi$. Note that Π is symmetric. As mentioned in Sect. 3, WAVE can be regarded as a distortion of PEAK or NADIR in many cases.

The second part is based on the average value of the joint speeds, which is defined as Eq. (3).

$$d(avgV_i^k, avgV_j^k) = 1.0 - \frac{\min(avgV_i^k, avgV_j^k)}{\max(avgV_i^k, avgV_j^k)} \quad (3)$$

Then, we combine directly the two parts as the distance metric of the short-term features noticing that both $d(DP_i^k, DP_j^k)$ and $d(avgV_i^k, avgV_j^k)$ belong to $[0, 1]$.

$$D^k(STF_i^k, STF_j^k) = d(DP_i^k, DP_j^k) + d(avgV_i^k, avgV_j^k) \quad (4)$$

where $D^k(STF_i^k, STF_j^k)$ denotes the distance metric of short-term features for the k-th joint. It is clear that our definition is a balance between the dynamic pattern and average speed. The dynamic pattern distance $d(DP_i^k, DP_j^k)$, which is robust to motion variations, can distinguish a period with increasing speeds from that with decreasing speeds even though they have similar average speeds. On the other hand, $d(avgV_i^k, avgV_j^k)$ plays the main role if their dynamic patterns are the same. Therefore, different stages are separated for a dynamic pattern. For example, peaks with very different average speeds may fall in different stages of a motion (e.g. walking), which can be distinguished by $d(avgV_i^k, avgV_j^k)$. We consider not only the magnitude but also the dynamic pattern of joint speed in a motion clip within the short term.

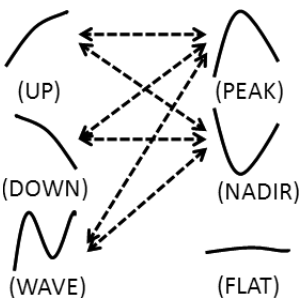

Fig. 6 Half the same dynamic pattern pairs (12 pairs in total), those dynamic patterns with connections are in the set Π of Eq. (2).

By averaging the distance metrics of all joints, we can define the distance metric of two motion clips.

$$D(STF_i, STF_j) = \sum_{k=1}^{K-1} w(k) D^k(STF_i^k, STF_j^k) \quad (5)$$

where $D(STF_i, STF_j)$ denotes the distance metric between two motion clips, $w(k)$ is a non-negative weight for the k-th joint. It is not difficult to demonstrate that the proposed distance metric satisfies the four properties of a metric, see the Appendix.

4.2 Motion Dissimilarity Measure

To determine the dissimilarity measure between two motions, we have to solve the time warping problem due to the different motion paces, different phases, and so forth. A common solution is called *dynamic time warping* algorithm [23], [24], which aligns two series of short-term features $\mathcal{F} = \{STF_i^k : k \in [1 : K-1], i \in [1 : I-1]\}$ and $\mathcal{G} = \{STF_j^k : k \in [1 : K-1], j \in [1 : J-1]\}$ with our proposed distance metric as Eq. (5). The motion dissimilarity measure $\mathcal{D}(\mathcal{F}, \mathcal{G})$ is defined as the average of local distance metrics $D(STF_i, STF_j)$ in the optimal path, which is obtained by dynamic time warping.

$$\mathcal{D}(\mathcal{F}, \mathcal{G}) = \frac{1}{L^{opt}} \sum_{(i,j) \in \mathbf{P}^{opt}} D(STF_i, STF_j) \quad (6)$$

where L^{opt} is the length of the optimal path $\mathbf{P}^{opt}$. Similar to [4], we use the average value rather than the total for independence of the path length. Obviously, the larger $\mathcal{D}(\mathcal{F}, \mathcal{G})$ is, the more dissimilar the two motions are.

5. Experimental Results

In this section, we report our experimental results on a test dataset of more than 190 MoCap sequences with a comparison of two conventional methods including a baseline and FMDistance. The baseline is the current prevailing method as pointed out in [3] (thus the de facto method). And FMDistance [8] is the most recently proposed technique where high performance is reported. For the baseline, the only difference from the proposal is that the baseline adopts the frame distance definition in [11] instead of extracting short-term features. The distance metric between the two frames $\mathbf{F}_{t1}$ and $\mathbf{F}_{t2}$ in [11] is the sum of the weighted difference of the joint orientations and the weighted distance of the joint velocities, see Fig. 7 (b) as an example. Those weights are optimized by [11]. Then, dynamic time warping is performed to align the corresponding frames among

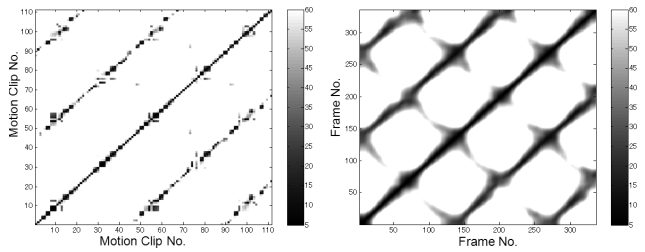

Fig. 7 Comparison of distance metrics of all pairs in a walking sequence (actor and motion ID:02_01), which includes more than two walking cycles. (a) proposed distance metrics among motion clips (8 frames/clip), (b) conventional distances among frames (details in Sect. 5 Baseline). Note that the corresponding values in FMDistance are not available.

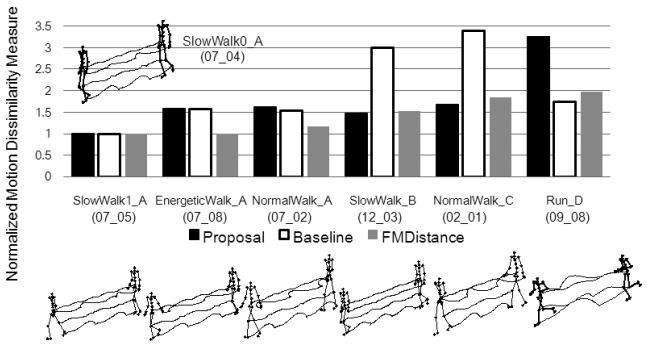

Fig. 8 Comparison of motion dissimilarity measures by the proposal (8 frames/clip), the baseline, and FMDistance, calculated between SlowWalk0_A and other sequences in Fig. 3, normalized by the motion dissimilarity measure between SlowWalk0_A and SlowWalk1_A.

the two sequences. Thus, motion dissimilarity measure is calculated in a similar way to the proposal. For FMDistance, we employ the best parameter setting reported in [8], where a feature vector is extracted by calculating the average kinetic energy of each joint angle in logarithms, followed by the Euclidean distance between feature vectors to obtain the motion dissimilarity measure. Figure 8 gives the motion dissimilarity measures of some examples for the baseline and FMDistance.

5.1 Dataset Setup

Currently, the CMU MoCap database is popular in the academic community [5], [8], [12], [25], [26]. Therefore, we decide to set up a test dataset from the CMU MoCap database, consisting of four categories of motions. In the CMU MoCap data, there are 31 joints in total and 62 degrees of freedom (DOF), where some DOFs are manually generated instead of being captured. The data are captured in 120 frames per second. Many sequences have multiple categories of motions or last a long time. To remove the ambiguity and retain simplicity in our test dataset, we only use 192 sequences within six categories of motions including swinging, punching, jumping, running, kicking, and walking, which are very common in our daily life and very basic for many applications. Each sequence has a single category of motion with no more than 1,032 frames. The actor and motion IDs used in this paper are identical with those in the CMU database. After manually removing the no motion parts in the motions from jumping and kicking, there are 66,445 frames in total in the test dataset, which lasts 553.7 seconds or is about 1/60 to 1/70 of the whole CMU database. Table 1 shows detailed information on our test dataset.

Table 1 Information of test dataset.

	Swing	Punch	Jump	Run	Kick	Walk
MoCap Num.	10	8	17	45	6	106
Actor Num.	1	2	2	4	2	20
Avg. Frames	446	438	262	174	333	417
Min. Frames	363	150	179	126	289	242
Max. Frames	561	900	369	760	382	1032

5.2 Preliminary Investigation

As a crucial component, we investigate the effectiveness of our distance metric in Eq. (5) as shown in Fig. 7 (a), which indicates that our distance metric clearly reflects the walking cycles. Compared to the baseline in Fig. 7 (b), our distance metric removes the half cycle much more effectively. This experimental result implies that our distance metric is more accurate than the baseline in reflecting the similarity of motions.

The motion dissimilarity measure should be robust to motion variations. Therefore, as an example, we test motion dissimilarity measures between SlowWalk0_A and other sequences in Fig. 3. Motion variations are observed in Fig. 3 from the personality, motion style, and motion pace. As shown in Fig. 8, the motion dissimilarity measures in the same category are similarly small by the proposal no matter how variant the motion pairs are. Meanwhile, it is very large between different categories of motions such as SlowWalk0_A and Run_D, which shows the great discrimination of our proposal. This desired characteristic is essential to reduce both false negative and false positive. On the other hand, the conventional methods fail to distinguish the differences derived from different categories of motions and motion variations (e.g. personality) in the same category. For example, in the baseline, the motion dissimilarity measure between two slow walking motions by two actors is rather large although the two motions are similar. Moreover, it is too small between SlowWalk0_A and the running motion Run_D to remove Run_D from the retrieval set. This is because the poses in walking and running may become similar although their dynamic characteristics are rather different, which demonstrates that it is insufficient to consider only pose difference in motion retrieval. In FMDistance, although they can separate two motion categories, the discrimination is rather weak.

5.3 Performance Analysis

We calculate the motion dissimilarity measures between the query and all the MoCap sequences in our test dataset. Then,

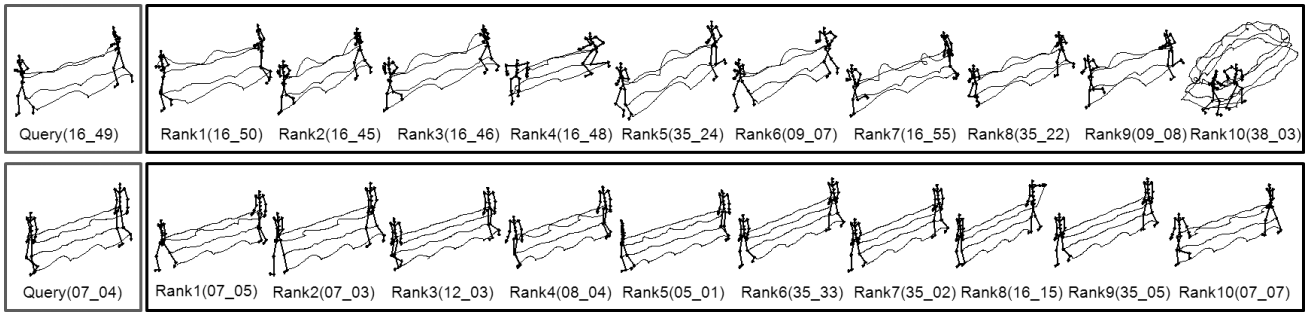

Fig. 9 The first ten motions retrieved by the proposal using a query of running motion (top) and a query of walking motion (bottom) respectively. Two stick figures in the first and last frames are rendered for each motion. Four trajectories are shown for root, left hand, left femur, and left tibia, respectively. The numbers in parentheses are actor and motion IDs in our test dataset, which is a subset of the CMU database. Benefiting from short-term features, our proposal can retrieve properly similar motions with large motion variations from the personality, motion pace, motion style, etc.

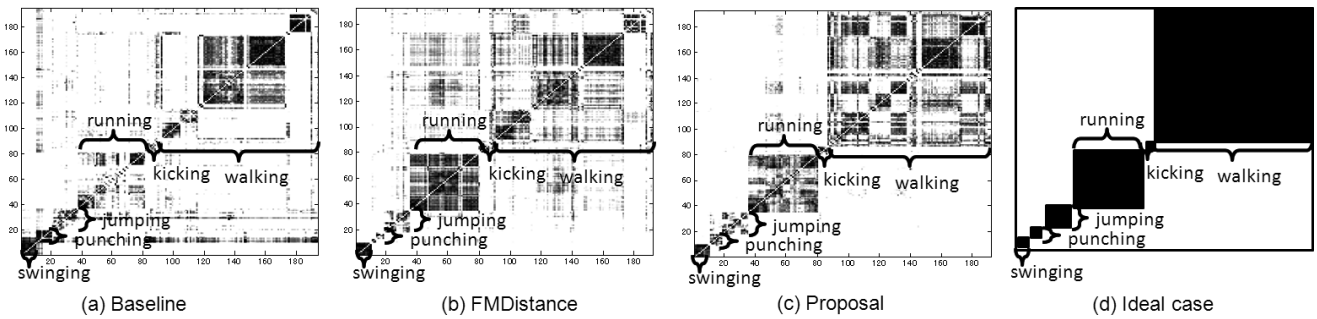

Fig. 10 Comparison of motion dissimilarity measures among all the MoCap data in our test dataset that consists of four categories of motions, (a) results from the baseline method, (b) results from the FMDsitance method, (c) results from the proposed method with 8 frames/clip, (d) ideal case. Note that the zeros in the diagonals are removed.

we rank the sequences in dataset by their motion dissimilarity measures. This procedure is the same in all three methods including the baseline, FMDistance, and the proposal. As an example, Fig. 9 shows the top ten retrieved motions by our proposed method using a running query and a walking query, respectively, where similar motions from different actors and motion styles are retrieved properly. Notice that all retrieved motions are in the same categories as the queries in both trials. And four actors are listed in the top ten motions when querying the running motion, which are all actors in 45 motions of the running category, see Table 1. In querying the walking motion, we retrieve 30% of actors by less than 10% motions in the top ten motions, see Fig. 9. This indicates that our proposal is robust to motion variations.

We calculate the motion dissimilarity measures among all the MoCap sequences in our test dataset in Fig. 10. Ideally, any motion dissimilarity measure between a query and a motion from another category should be larger than that between the query and a motion within the same category (see Fig. 10 (d)). Compared to the ideal case, the baseline (see Fig. 10 (a)) has more difficulty in distinguishing motions belonging to the same category or different categories. On the other hand, FMDistance (see Fig. 10 (b)) has greater discrimination (e.g. in the running category) than the baseline although their motion dissimilarity measures are still far from the ideal case. In contrast, the proposed method (see Fig. 10 (c)) approaches the ideal case best and thus has two advantages: one is the motion dissimilarity measures among a category of motions are smaller, which demonstrates that our algorithm is robust to motion variations; and the other is the motion dissimilarity measures in different categories are larger, which dismisses the dissimilar motions correctly.

We compare the proposal and two conventional methods by $P(N_R)$, a popular evaluation approach in content-based retrieval [9]. $P(N_R)$ denotes precision after N_R motions are retrieved, where N_R is the number of motions in the same category as the query. Figure 11 shows the results of all motions in our test dataset and Fig. 12 is a box plot of Fig. 11. From the both figures, our proposal achieves the best performance and the baseline has the worst performance in most of cases. Furthermore, assuming those values are mutually independent with the same continuous distribution, we perform the Friedman test among the proposal, the baseline, and FMDistance with the null hypothesis that there is no difference among the three methods. We get a p-value of 0, which discards the null hypothesis at a significant level of 0.01 and suggests the significant difference exists among the three methods. In Fig. 13, we average $P(N_R)$ in each category and the entire dataset. On average, $P(N_R)$ in our method is 90.8% for 8 frames/clip, which is about

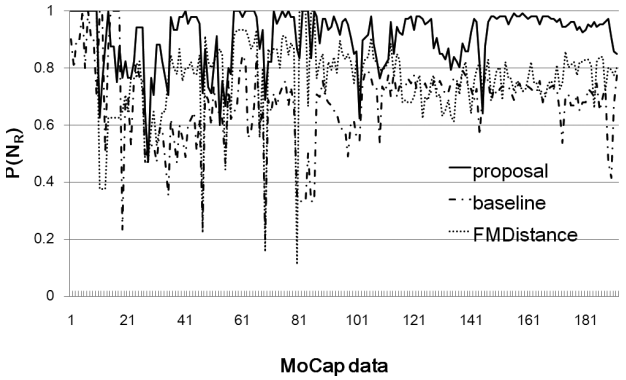

Fig. 11 Comparison of $P(N_R)$ for all the motions in our test dataset, where 8 frames/clip are used in the proposal.

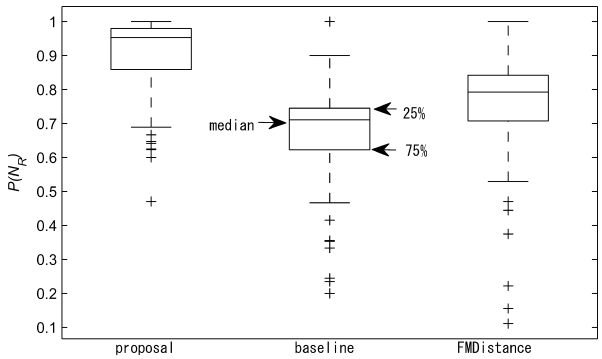

Fig. 12 A box plot of Fig. 11. On each box, the central mark is the median, the edges of the box are the 25th and 75th percentiles.

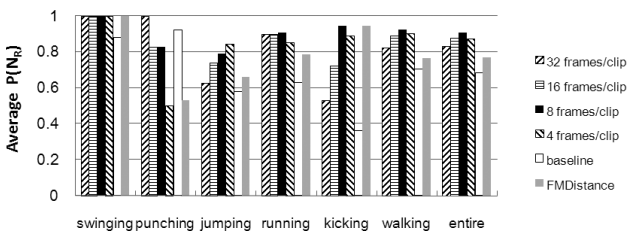

Fig. 13 Comparison of average $P(N_R)$ for six categories and the entire dataset, where four motion clip lengths are tested in the proposal.

23% higher than the baseline and about 14% higher than the FMDistance. Our technique also improves the performance greatly in most of the six categories, compared to the two conventional methods.

However, as shown in Fig. 10 (c), we have not reached the ideal state for the following two facts. One reason is that our feature extraction discards some information such as the root motion, which may cause errors. Unfortunately, information loss is very difficult if not impossible to completely avoid in feature extraction. The other reason is that middle motions exist between the two categories such as jogging from walking and running. In our test dataset, some walking motions are confused with running motions. It is understandable that the confusion will increase among energetic walking, jogging and slow running while the errors

Table 2 Joint weights used in our experiments (non-listed joints are set as zeros, see Fig. 2 for joint names).

	l/r-femur	l/r-tibia	l/r-humerus	l/r-radius
Setting 1	1.0	1.0	0.5	0.5
Setting 2	1.0	1.0	0.0	0.0
Setting 3	0.5	1.0	0.5	1.0
Setting 4	1.0	0.5	1.0	0.5
Setting 5	1.0	1.0	1.0	1.0
Setting 6	0.0	0.0	1.0	1.0
Setting 7	above 8 joints, shoulders, and head (totally 11 joints) are set as 1.0			
Setting 8	all 31 joints are set as 1.0			

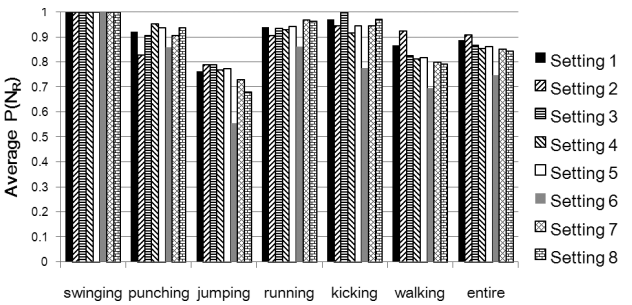

Fig. 14 Comparison of average $P(N_R)$ for four categories and the entire dataset, where eight combinations of joint weights are tested using 8 frames/clip in the proposal.

will decrease if querying a slow walking motion or an energetic running motion. Essentially, this challenging problem is due to the semantic gap between high-level human perception and low-level features.

Several motion clip lengths are tested including 4 frames/clip, 8 frames/clip, 16 frames/clip, and 32 frames/clip, which is regarded as a stable range of human motion. Basically, the longer the motion clip is, the less the discrimination is. This is because a motion clip that is too long may cover two or more dynamic patterns. Figure 13 shows the average $P(N_R)$ in the motion clip lengths. The best $P(N_R)$ (90.8%) is achieved in the case of 8 frames/clip or 0.067 seconds/clip.

As pointed out in [11], the joint weights in Eq. (5) should be optimized to achieve better performance. Here, we investigate how many joints are needed, which joints should be selected, and what are the best weights for those selected joints. We test some combinations of joint weights as listed in Table 2 using 8 frames/clip. From the results, as shown in Fig. 14, we can learn the following facts. 1. it cannot improve the performance just by increasing joint number (see Setting 7 and 8). 2. it works well by intuitively selecting the eight joints in legs and arms as shown in Fig. 2. However, in this paper, we remain the problem of optimizing joint weights as our future work.

The computational cost is evaluated by the running time of our programs. Because feature extraction is performed as an off-line process, we exclude its running time. Using the same laptop computer (Intel® Core2® Duo 2.0 GHz CPU and 1.5 GB main memory) and compiler (Microsoft® Visual Studio®), our algorithm is about

Table 3 Comparison of total running time and speedup to baseline.

	baseline	FMDistance	proposal			
			4 frame/clip	8 frame/clip	16 frame/clip	32 frame/clip
time (min.)	1115.1	0.4	81.0	23.8	9.1	4.3
speedup	1	2788	14	47	123	259

123 times faster than the baseline method in the case of 16 frames/clip. Note that FMDistance defines a feature vector considering all frames in a motion to avoid dynamic time warping, making it very fast to compute a motion dissimilarity measure with the cost of losing the ability of subsequence retrieval. Table 3 shows the total running time for calculating the motion dissimilarity measures among all the sequences in our test dataset. Note that the retrieval time depends on the lengths of query and motions in the dataset.

6. Conclusions and Future Work

This paper presents a content-based retrieval algorithm for MoCap data with promising experimental results in our test dataset. Our main contributions are as follows:

1. **Short-Term Features:** An effective feature extraction method has been proposed considering the short-term characteristics of joint velocities. It is our basic idea that short-term analysis is essential for the similarity of motions. Most previous works only consider a single frame [27] or a simple combination of several frames [13], [16]. In this paper, our features are extracted from both the dynamic pattern and magnitude of joint velocities in the short term, which are robust to motion variations. Moreover, it can quickly extract the short-term features with little requirement for user's expertise, leading to great potential in many applications for our short-term features. The idea of short-term analysis in MoCap is inspired by its successful usage in audio analysis [17], which is suitable for non-stationarity processes such as audio and motion. This is the first time a short-term feature in MoCap data has been proposed.
2. **Distance Metric of Features:** A distance metric is carefully defined for our short-term features, which is a crucial ingredient of the motion retrieval technique. Moreover, as pointed out by [3], it is fundamental for many other techniques such as segmentation, clustering, and motion blending. A trade-off is achieved to balance between the dynamic pattern and average speed in our short-term features. Our experiments suggest that the distance metric is effective.
3. **Motion Dissimilarity Measure and Retrieval:** Based on the distance metric, our motion dissimilarity measure of two MoCap sequences is calculated using dynamic time warping. We rank the MoCap sequences in the dataset according to their motion dissimilarity measures from the query. Generally speaking, high performance both in effectiveness and in efficiency is required, which is achieved in our algorithm.

Future work: Although our experimental results demonstrate the great potential of applying our technique, many improvements are possible. For example, it is necessary to further accelerate our algorithm. The main bottleneck in our system is the dynamic time warping algorithm, which is quadratic in the number of frames. We are interested in developing such a substitute as uniform time scaling [18].

As mentioned in Sect. 5.3, it is very interesting to optimize joint weights in Eq. (5) as [11]. Moreover, different motion categories may require different joint weights. Therefore, it is better to use adaptive weights. Currently, our static approach cannot be always optimal for any motion category.

We are also planning to employ the short-term features in other applications such as motion structure analysis [28] and motion blending [29]. Motion structure analysis is useful in obtaining a single category of motions from a complicated sequence. Therefore, it may be served as a preparation step of a MoCap retrieval system, e.g., see [5]. Without motion structure analysis, our test dataset has to consist of simple and short sequences. Moreover, the computational cost of retrieval may be reduced by segmenting the long sequence into single motions by techniques such as [26], [30]. Many motion blending techniques require the registration of two motions properly before blending [29], where our short-term features are obviously employable.

Acknowledgements

The data used in this project were obtained from mocap.cs.cmu.edu [2]. The database was created with funding from NSF EIA–0196217.

References

[1] C. Bregler, "Motion capture technology for entertainment," IEEE Signal Process. Mag., vol.24, no.6, pp.156–160. However, the three–page paper was published on Page #160, #156, #158 in order., Nov. 2007.
[2] CMU MoCap Database, http://mocap.cs.cmu.edu/, 2003.
[3] C. Faloutsos, J. Hodgins, and N. Pollard, "Database techniques with motion capture," ACM SIGGRAPH 2007 Course #21 Notes, Aug. 2007.
[4] L. Kovar and M. Gleicher, "Automated extraction and parameterization of motions in large data sets," ACM Trans. Graphics, vol.23, no.3, pp.559–568, Aug. 2004.
[5] M. Muller, T. Roder, and M. Clausen, "Efficient content-based retrieval of motion capture data," ACM Trans. Graphics, vol.24, no.3, pp.677–685, July 2005.
[6] A. Bruderlin and L. Williams, "Motion signal processing," SIGGRAPH '95: Proc. 22nd Annual Conference on Computer Graphics and Interactive Techniques, pp.97–104, 1995.

[7] K. Forbes and E. Fiume, "An efficient search algorithm for motion data using weighted pca," ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp.67–76, 2005.
[8] K. Onuma, C. Faloutsos, and J.K. Hodgins, "FMDistance: A fast and effective distance function for motion capture data," Proc. EUROGRAPHICS 2008, Short Papers, 2008.
[9] H. Muller, W. Muller, D.M. Squire, S. Marchand-Maillet, and T. Pun, "Performance evaluation in content-based image retrieval: Overview and proposals," Pattern Recognit. Lett., vol.22, no.5, pp.593–601, April 2001.
[10] A. Hanjalic, R. Lienhart, W.Y. Ma, and J.R. Smith, "The holy grail of multimedia information retrieval: So close or yet so far away?," Proc. IEEE, vol.96, no.4, pp.541–547, April 2008.
[11] J. Wang and B. Bodenheimer, "An evaluation of a cost metric for selecting transitions between motion segments," Proc. 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp.232–238, 2003.
[12] 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. Graphics, vol.24, no.3, pp.1090–1097, July 2005.
[13] O. Arikan, "Compression of motion capture databases," ACM Trans. Graphics, vol.25, no.3, pp.890–897, July 2006.
[14] M. Muller and T. Roder, "Motion templates for automatic classification and retrieval of motion capture data," ACM SIGGRAPH/ Eurographics Symposium on Computer Animation, pp.137–146, 2006.
[15] B. Demuth, M. Muller, and B. Eberhardt, "An information retrieval system for motion capture data," Proc. 28th European Conference on Information Retrieval (ECIR), pp.373–384, 2006.
[16] L. Kovar, M. Gleicher, and F. Pighin, "Motion graphs," ACM Trans. Graphics, vol.21, no.3, pp.473–482, July 2002.
[17] J.R. Deller, J.H. Hansen, and J.G. Proakis, Discrete-Time Processing of Speech Signals, IEEE Press, 2000.
[18] E. Keogh, T. Palpanas, V.B. Zordan, D. Gunopulos, and M. Cardle, "Indexing large human-motion databases," Proc. 30th VLDB Conference, pp.780–791, 2004.
[19] T. Guan and Y.H. Yang, Motion Similarity Analysis and Evaluation of Motion Capture Data. Technical Report 05–11 in University of Alberta, available at www.cs.ualberta.ca/TechReports/2005/TR05-11/TR05-11.pdf, 2005.
[20] Y. Sakamoto, S. Kuriyama, and T. Kaneko, "Motion map: Image-based retrieval and segmentation of motion data," ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp.259–266, 2004.
[21] G. Johansson, "Visual perception of biological motion and a model for its analysis," Perception & Psychophysics, vol.14, no.2, pp.201–211, 1973.
[22] A. Watt and F. Policarpo, Advanced Game Development with Programmable Graphics Hardware, A.K. Peters, 2005.
[23] C.S. Myers and L.R. Rabiner, "A comparative study of several dynamic time-warping algorithms for connected word recognition," Bell Syst. Tech. J., vol.60, no.7, pp.1389–1409, Sept. 1981.
[24] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," Intelligent Data Analysis, vol.11, no.5, pp.561–580, Sept. 2007.
[25] P.S.A. Reitsma and N.S. Pollard, "Evaluating motion graphs for character navigation," SCA '04: Proc. 2004 ACM SIGGRAPH/ Eurographics Symposium on Computer Animation, pp.89–98, Aire-la-Ville, Switzerland, Switzerland, Eurographics Association, 2004.
[26] J. Barbič, A. Safonova, J.Y. Pan, C. Faloutsos, J.K. Hodgins, and N.S. Pollard, "Segmenting motion capture data into distinct behaviors," GI '04: Proc. Graphics Interface 2004, pp.185–194, 2004.
[27] J. Lee, J. Chai, P.S.A. Reitsma, J.K. Hodgins, and N.S. Pollard, "Interactive control of avatars animated with human motion data," ACM Trans. Graphics, vol.21, no.3, pp.491–500, July 2002.
[28] T.H. Kim, S.I. Park, and S.Y. Shin, "Rhythmic-motion synthesis based on motion-beat analysis," ACM Trans. Graphics, vol.22, no.3, pp.392–401, July 2003.
[29] L. Kovar and M. Gleicher, "Flexible automatic motion blending with registration curves," ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp.214–224, 2003.
[30] T. Nakata, "Temporal segmentation and recognition of body motion data based on inter-limb correlation analysis," Intelligent Robots and Systems, 2007, IROS 2007, IEEE/RSJ International Conference on, pp.1383–1388, Nov. 2007.

Appendix

Here is a brief proof that our distance function $D(STF_i, STF_j)$ in Eq. (5) is a metric.

Definition: A *metric* on a set X is a function $d : X * X \rightarrow \mathcal{R}$. For all x, y, z in X, this function is required to satisfy the following conditions:

1. $d(x, y) \geq 0$ (non-negativity)
2. $d(x, y) = 0$ iff $x = y$ (identity of indiscernibles)
3. $d(x, y) = d(y, x)$ (symmetry)
4. $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality)

As a first step, we prove $d(DP_i^k, DP_j^k)$ in Eq. (2) is a metric.

It is obvious $d(DP_i^k, DP_j^k)$ satisfies the first three conditions by its definition. Note that the set Π in Eq. (2) is symmetric.

To prove $d(DP_i^k, DP_j^k) \leq d(DP_i^k, DP_x^k) + d(DP_x^k, DP_j^k)$, consider the following cases:

- If $DP_i^k = DP_j^k$, $d(DP_i^k, DP_j^k) = 0$. Thus, triangle inequality is satisfied.
- If $(DP_i^k, DP_j^k) \in \Pi$, $d(DP_i^k, DP_j^k) = 0.5$. The condition that triangle inequality is not held is that both $d(DP_i^k, DP_x^k)$ and $d(DP_x^k, DP_j^k)$ are 0. That means $DP_i^k = DP_x^k = DP_j^k$, which is conflict with $(DP_i^k, DP_j^k) \in \Pi$.
- If $d(DP_i^k, DP_j^k) = 1.0$, the condition that triangle inequality is not held is that $d(DP_i^k, DP_x^k)$ or $d(DP_x^k, DP_j^k)$ is 0. Suppose $d(DP_i^k, DP_x^k) = 0$, i.e., $DP_i^k = DP_x^k$. That is to say, $d(DP_i^k, DP_j^k) = d(DP_x^k, DP_j^k)$. So triangle inequality is satisfied.

Therefore, $d(DP_i^k, DP_j^k)$ is a metric.

Next, we prove $d(avgV_i^k, avgV_j^k)$ in Eq. (3) is a metric.

It is obvious $d(avgV_i^k, avgV_j^k)$ satisfies the first three conditions by its definition.

To prove triangle inequality, suppose $avgV_i^k \leq avgV_j^k$ and discuss the following cases:

- $avgV_x^k \leq avgV_i^k \leq avgV_j^k$
- $avgV_i^k < avgV_x^k \leq avgV_j^k$
- $avgV_i^k \leq avgV_j^k < avgV_x^k$

It is trivial to prove the triangle inequality in the above three cases respectively. Therefore, $d(avgV_i^k, avgV_j^k)$ is a metric.

The following lemma can be obviously proved.

Lemma: If $d1(x1, y1)$ and $d2(x2, y2)$ are two metrics

on the set $X1$ and $X2$ separately, then $d(x, y)$ is also a metric on the set $X1 \otimes X2$, where $d(x, y) = w1 \cdot d1(x1, y1) + w2 \cdot d2(x2, y2)$, $w1 \geq 0, w2 \geq 0$.

Therefore, it is straightforward to prove that $D(STF_i^k, STF_j^k)$ and $D(STF_i, STF_j)$ are metrics by the lemma.

Jianfeng Xu received the B.S. (with honor) and the M.S. degrees from Tsinghua University, China, in 2001 and 2004 respectively and the Ph.D. degree from The University of Tokyo, Japan, in 2007. Currently, he works as an associate research engineer in Media Solutions Laboratory, KDDI R&D Laboratories Inc., Japan. His research interests include motion capture data analysis and re-use such as content-based motion retrieval and synchronization with music.

Haruhisa Kato received the B. E. and M. E. degrees in electrical and electronics engineering from Kobe University in 1997 and 1999, respectively. In 1999, he joined KDD Co. Ltd. and he is currently a research engineer of the User Interface Laboratory at KDDI R&D Laboratories Inc. He has been working on audio/visual compression, compressed video processing and audio/vidual retrieval.

Akio Yoneyama received the B.E. and the M.E. degrees in electrical engineering from Keio University, Japan in 1992 and 1994, respectively, and the Ph.D degree in Information Processing from Tokyo Institute of Technology, Japan, in 2007. He joined KDD in 1994. Since 1996, he has been with KDDI R&D Laboratories Inc. From 2002 to 2003, he was a visiting researcher at University of Southern California. His current research interests include visual media processing.