

Adaptive Motion Data Representation with Repeated Motion Analysis

I-Chen Lin, *Member, IEEE*, Jen-Yu Peng, Chao-Chih Lin, and Ming-Han Tsai

Abstract—In this paper, we present a representation method for motion capture data by exploiting the nearly repeated characteristics and spatio-temporal coherence in human motion. We extract similar motion clips of variable lengths or speeds across the database. Since the coding costs between these matched clips are small, we propose the repeated motion analysis to extract the referred and repeated clip pairs with maximum compression gains. For further utilization of motion coherence, we approximate the subspace-projected clip motions or residuals by interpolated functions with range-aware adaptive quantization. Our experiments demonstrate that the proposed feature-aware method is of high computational efficiency. Furthermore, it also provides substantial compression gains with comparable reconstruction and perceptual errors.

Index Terms—Three-dimensional graphics and realism-animation, compression (coding)-approximate methods.



1 INTRODUCTION

MOTION capture techniques (mocap), which estimate subjects' motions through marker tracking, are popularly used in various applications. In entertainment industry, the tracked motions are applied to character animation or further data-driven motion synthesis. They are also applied to gait analysis in biomechanics or clinical medicine.

Mocap data consist of a hierarchical skeletal structure and trajectories of degrees of freedom (DOFs) of joints, e.g. Euler angles or 3D positions. To grasp the large variety of human motions, we usually need to record 45 to 120 DOFs at 24 to 120 frames per second (fps). As the motion data increases, it becomes a troublesome problem, especially in large database, or real-time and telepresence applications, where the memory, bandwidth, and storages are limited.

An immediate solution is to represent the original motion data in a compressed form. In multimedia processing, from MPEG series to AC3, video and audio compression techniques have been well developed for decades. However, the characteristics of human motion are different from video or audio ones.

First, nearly repeated motions frequently occur. For instances, walking, running and hand waving are typical motions and can have variable periods. Besides, due to biomechanical characteristics, human motion has high temporal coherence and behavior-dependent correlations between joints. For instance, the left hand and right leg are highly correlated in walking.

Recently, researchers notice the rapidly growing problem. Barbic et al. [1] proposed segmenting motion sequences according to the accuracy variation of probabilis-

tic principal component analysis (PPCA). Liu and McMillan [2] then approximated the motion segments by key-frame interpolations. This segmentation is reliable for distinct behaviors, e.g. from walking to sitting, but cannot retrieve repeated clips, e.g. steps in walking. In seminal literature of O. Arikan [3], he first fit motion DOFs with curves and then applied principal component analysis (PCA) to extract joint correlations in short-term clips.

The goal of this paper is to efficiently represent motion capture data and retain few reconstruction errors as well. We propose a novel segmentation and indexing method, called repeated motion analysis (RMA).

The RMA is inspired by "video textures" [4], where similar frames in an image sequence were evaluated. By transiting between these frames, endless video could be synthesized. This concept initiated the motion transition techniques [5], [6], [7]. Later, Kovar and Gleicher [8] extended this idea and presented a match-web system to query similar motions for parameterized motion blending.

On the other hand, the proposed RMA retrieve and index nearly repeated motions of least compression costs within a sequence or across database. Since only slight differences need to be encoded, these repeated clips can be compressed more compactly. While compressing clips with PCA, we exploit adaptive curves to approximate the variations of projected coefficients. Instead of fixed quantization, we further make use of the variable variance characteristic in PCA coefficients and present adaptive-bit quantization for the subspace data.

Our experiments demonstrate that the proposed method can achieve a high compression ratio for general motion data with walking, running, dancing, exercising, etc. (108:1 for a data set from CMU mocap library [9]). The user evaluation shows that even with such considerable data compaction, the perceptual errors are also comparable to related research.

The rest of this paper is organized as follows. The Section 2 introduces related research, followed by an over-

• The authors are with the CAIG Lab, Department of Computer Science, National Chiao Tung University, 1001 Ta-Hsueh Road, Hsinchu City, 300 Taiwan. E-mail: ichenlin@cs.nctu.edu.tw, alvin@caig.cs.nctu.edu.tw, gme94529@cis.nctu.edu.tw, and ParkerTsai@caig.cs.nctu.edu.tw.

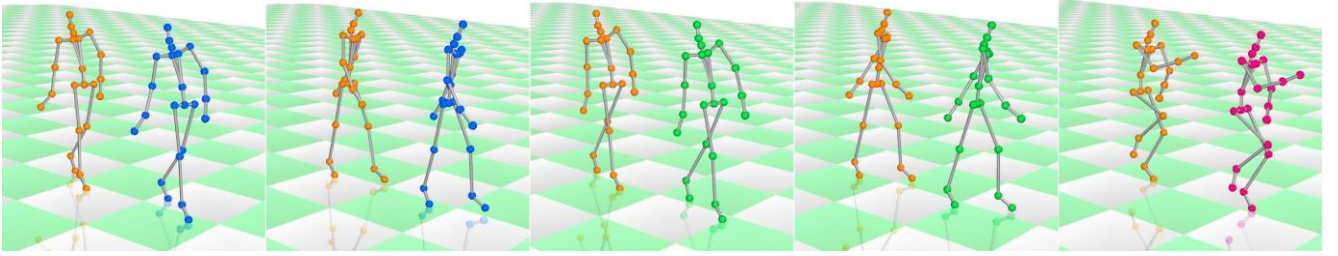


Fig. 1. The proposed RMA extracts similar motion clips across motion database for efficient compression. The original motion is rendered in orange; the referred motion, called primary clip, is in blue; the repeated clip is in green; the unique motion is rendered in pink.

view of the proposed system. In Section 4, we introduce how to extract repeated clips according to expected compression gains. Section 5 presents the clip approximation by curves and quantization. Experimental results and comparisons are presented in Section 6. The final section concludes this paper.

2 RELATED WORK

In computer graphics, numerous researches focus on geometry compression, e.g. the famous progressive mesh [10]. Alexa and Muller [11] projected entire animation sequences into a lower dimensional space by PCA for a more compact form.

On the other hand, character animation is usually represented by skeletal motions, which have high repetition and joint correlations. Liu and McMillan [2] proposed a segment-based approach to compress joint position trajectories. After dividing motion data into several behavior segments by PPCA, they applied PCA to project segmented motion to lower subspace and fit them by cubic splines. Their method used dynamic sampling but did not address the repetition characteristics.

Arikan [3] proposed compressing hierarchical joint angles through virtual markers in Euclidean space. After dividing database into a large amount of fixed-length clips and performing Bezier curve fitting, he then clustered these parameterized clips into groups and performed PCA to lower the dimensions for later entropy-based compression. User-specified high-frequency motions, like environment contacts, were compressed separately. Since he applied PCA in terms of clustered clips instead of frames, this approach achieved an impressive ratio on dimensional reduction but had large storage overhead on principal component (PC) axes.

In addition to general-purpose motion compression, Chattopadhyay et al. [12] proposed an indexing method for motion data encoding. This method is mainly for low-resolution encoding with limited power consumption. Tournier et al. [13] proposed using an Inverse Kinematics (IK) approach, called Principal Geodesics Analysis (PGA) IK. They only stored the motions of end effectors and roots, and PGA axes. They used optimization to reconstruct the full-body posture for decompression. This method provides a significant compression gain, but required considerable computation time. Besides, it suffered the ambiguous posture problem at non-constrained joints as other IK-based systems.

To retrieve similar clips across motions, it is necessary to perform segmentation before clip compression. Li et al. [14] exploited linear dynamic systems to extract motion textons. For high-level behaviors, Barbic et al. [1] proposed three different segmentation techniques based on statistical properties. In addition to large-scale motion segmentation, research about short- to mid- term periodic or cyclic sequences has also been studied in video processing. Cutler and Davis [15] tracked the object of interest and computed its self-similarity for each frame to extract the periodic motions in 2D image sequences. Gu et al. [16] proposed a motion indexing method. They directly separated motions by its hierarchical skeletal structure and chopped each partial motion sequence according to user-specified smoothness. Motion patterns can then be retrieved by k-means clustering.

The goal of extracting motion pattern is similar to our repeated motion analysis (RMA). Instead of separating joint motions, we embrace the joint correlations by PCA. Besides, the proposed RMA can extract both smooth and acute repeated clips with variable lengths or speeds according to the expected coding gain. The segmentation is according to repetition and depends less on indirect parameters, like smoothness. Fig. 1 shows an example of our reconstructed motion with different colors to indicate their repetition.

3 OVERVIEW

The proposed motion representation can be divided into three phases: preprocessing, clip extraction, and clip approximation.

After loading motion data files onto our system, we concatenate them as a long motion sequence. Character motions are composed of orientation and translation of the root and skeletal postures. For better analysis on repeated skeletal motion, the first step is to align the root orientation and translation. When aligning raw point set data, we apply Kovar and Gleicher's method for y -orientation [7] or B.Horn's quaternion-based alignment [17]. For hierarchical skeletal motion data, e.g. BVH (Biovision hierarchical data) files, we simply take the orientations and translations of the root joint.

Our analysis showed that encoding the full root motion ($R_x, R_y, R_z, T_x, T_y, T_z$) sequences in PCA subspace is inefficient, where only one or even none of six principal components can be omitted. Since most of root movements result from locomotion, in most cases, we only

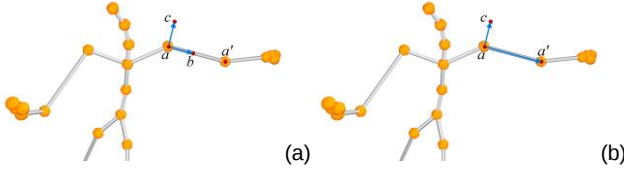


Fig. 2. Representing joint angles through virtual markers. (a)The virtual marker set in [3]. (b)The compact virtual marker set.

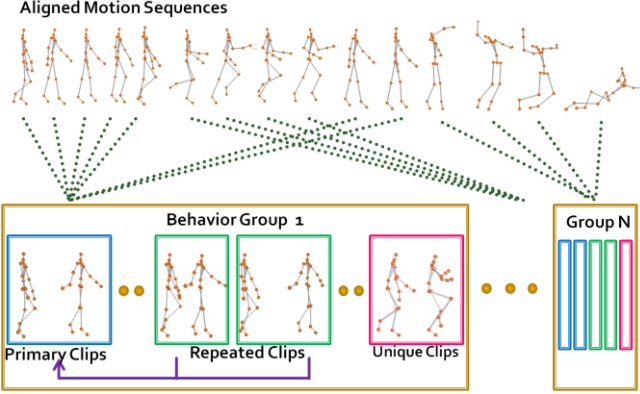


Fig. 3. The conceptual diagram of clip extraction. The aligned motion sequences are first segmented and grouped according to their behavior; each group is then divided into primary, repeated and unique clips. A repeated clip can be represented by a sequence of reference indices and motion differences to its referred primary clip.

align the major y orientation and x, y, z translation (R_y, T_x, T_y, T_z). For motions with large R_x, R_z variations, such as somersault and breakdance, we align all the six degrees of freedom. These root alignment data are separately fitted by Catmull-Rom splines. The residual root and joint movements can then be represented by the following clip approximation.

As suggested in [3], compressing joints at a Euclidean coordinate is more linear and adequate to PCA. We also transfer hierarchical joint angles to position space through virtual markers. As shown in Fig.2, the virtual marker set of a joint in [3] are a, b, c , but the markers a, b and its child marker a' are collinear along the bone vector. We can use only one additional marker c with original joint position a

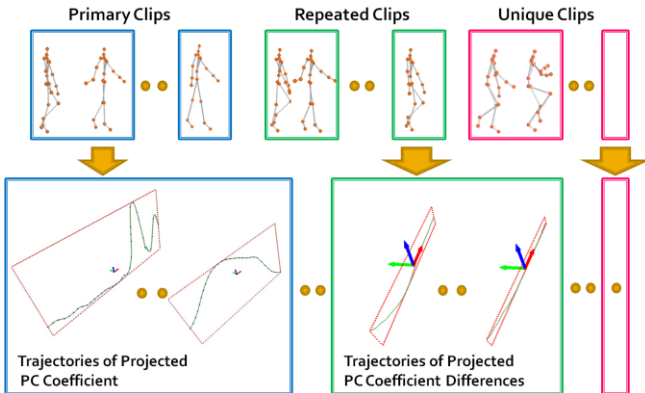


Fig. 4. The conceptual diagram of clip compression. For each clip, its projected PC coefficient trajectories are approximated by interpolation functions.

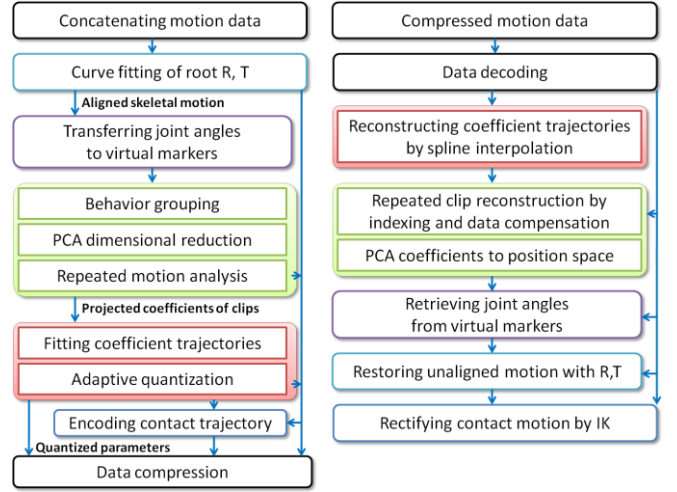


Fig. 5. The flow charts of the proposed motion representation method (Left) the encoding process; (right) the decoding process.

and a' to reconstruct the joint angles. These two virtual marker configurations result in an identical PCA subspace when we normalize the vector aa' , but using one fewer virtual markers per joint can spare one-third of storage for principal component axes. For joint without sufficient bone length, we still allocate marker b .

The second phase is to extract the compression unit blocks, called clips. As the conceptual diagram Fig.3, the aligned motion sequence is first divided into several behavior groups. For each group, we apply frame-based PCA to lower the dimension, e.g. from DOFs of 73 positions to 5 to 15 principal components. Repeated motion analysis (RMA) is then applied to these more compact data and further divides a behavior group into three types of clips. The first type is *primary clip*, including the most influential motions, and is referred by a set of repeated clips. The second type is *repeated clip*, including motions similar to those in the referred primary clip. The third type is *unique clip* where no repetition can be extracted.

After the PCA dimensional reduction, our motion data become a set of time-varying projected coefficients along PC axes. Variation of each coefficient can be regarded as a coefficient trajectory. Hence, the third phase, clip approximation, focuses on compact representation of these trajectories by curves. Fig. 4 introduces the concept of clip approximation.

Since the magnitude of primary and repeated motions are different and the projected coefficient variances of PC axes are decreasing, we propose using range-aware quantization to further compact the data and keep comparable accuracy. The flow charts of the proposed compression method and corresponding decompression are presented in Fig. 5.

4 REPEATED CLIP EXTRACTION

In this section, we describe how to extract primary, repeated and unique clips from an aligned motion sequence. To avoid intensive evaluation of all possible repetitions in

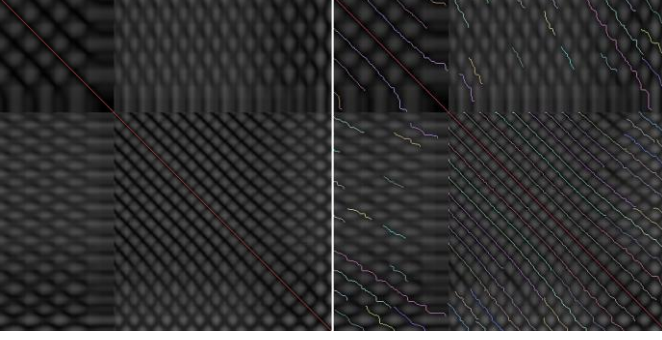


Fig. 6. The self-distance plot and corresponding similar posture paths. This plot contains two segments of running motions from different sequences, where the running speeds are various. (Left) the self-distance plot, where the intensity of a pixel(i, j) represents the L2 norm distance between frame i and j ; (right) the extracted similar posture paths are represented in different colors, respectively.

a whole motion sequence, we first group motion segments according to their behaviors as described in subsection 4.1. Section 4.2 introduces how we extract possible repetitions through self-distance plots and evaluate their effectiveness.

4.1 Behavior Segmentation and Grouping

The goal of behavior grouping is to collect motion segments with similar behaviors across the whole motion sequence. For instance, we would like to collect scattered dribbling motions from a basketball playing sequence, composed of dribbling, shooting, and passing.

We employ PPCA-based motion segmentation proposed by J. Barbic et al. [1] for the first step. Here, we briefly describe the concept. Given a short initial motion segment, we estimate its major probabilistic principal components. If the following frames can also be precisely represented by these PCs, we can merge them into the segment; otherwise, we make a cut and generate a new segment. In order to avoid over-fitted PCs caused by chaotic or complex motion in the initial segment, we perform behavior segmentation in both forward and backward directions and unify these two cut sets.

After data segmentation, we further make use of the estimated major PCs and mean postures as distance metrics. Normalized-cut (Ncut) method [18] is applied to group behavior segments according to the distance function:

$$SegDist(i, j) = \alpha \sum_{p=1}^K \|w(p)PC_p^i \cdot PC_p^j\| + (1 - \alpha)(MP^i \cdot MP^j) \quad (1)$$

, where PC_p^i is the p^{th} principal component (axis) of segment i , MP^i is the normalized vector of mean posture at segment i , $w(p)$ is a decreasing weight to enhance the effects of prior PCs, and $\alpha = 0.6$, $K = 5$ in our case.

For each group, we concatenate all its motion segments, and perform PCA for dimensional reduction. The original motion trajectories are then projected to valid principal components (axes) of corresponding groups. In the following subsections, grouped motion data become a set of projected coefficient trajectories at PCA subspace.

4.2 Repeated Motion Analysis

Distance plots, based on dynamic time alignment, have

proven to be a useful tool for matching two time-varying data. Kovar and Gleicher [7], [8] used distance plots to extract the optimal transition point and acquired similar motion segments between two motion sequences. While matching a sequence with itself, we can get a self-distance plot. Fig. 6 is an example of a running motion group. The intensity of pixel(i, j) represents the L2-norm distance of projected coefficients between frame i and j . The darker intensity means that two frames are much similar. For a periodic motion sequence, there are nearly diamond-shape or oblique-stripe patterns in the plot. Cutler and Davis [15] utilized this property to detect periodic motion in video, but the motion speed and length have to be nearly regular.

Similar-posture paths

From the viewpoint of compression or data abstraction, we prefer retrieving similar motion with variable speeds or lengths. Instead of regular pattern retrieval, we propose using similar-posture paths for clip extraction. The procedures are as follows:

1. Suppress dissimilar pixels by thresholding.
2. Label pixels which are 1D local minima (in x or y axes), or 2D minima (both directions).
3. Select a 2D minimum pixel; explore a minimum-cost path toward both upper-left and lower-right direction. (For efficiency, we only explore 1D and 2D minimum pixels)
4. Connect the explored pixels as a path.
5. Repeat Step 3 until all the labeled 2D minimum pixels belong to a path.
6. For each path, check whether it can merge other paths with a valid short bridge in lower-right direction.
7. Remove paths without sufficient lengths and form similar posture path sets.

In our system, a valid bridge has no more than 300 milliseconds (ms); a path with Manhattan distance less than 600 ms is removed. In [8], the time-aligned paths with sharp or flat slopes were regarded as degenerate cases. On the contrary, such paths may still provide high redundancy for compression. Since the self-distance plot is symmetric along the diagonal, we only evaluate the upper half and reflect the result to the lower half. Besides, for a large group, we can perform RMA on a lower-resolution distance plot to reduce computation. The right of Fig. 6 shows the similar posture paths extracted from the left.

Extracting the most effective repetitions

With similar posture paths, the next step is to find out the reference clips, so called primary clips, and their corresponding repeated clips for maximum compression gains.

We take Fig. 7a as an example. An arbitrary segment along a similar posture path represents a mapping from the projected frame interval $[px_s, px_e]$ to $[py_s, py_e]$. Since the repetition numbers of an interval are determined by the projections of similar posture paths, we project the end-points of similar posture paths to form preliminary clip

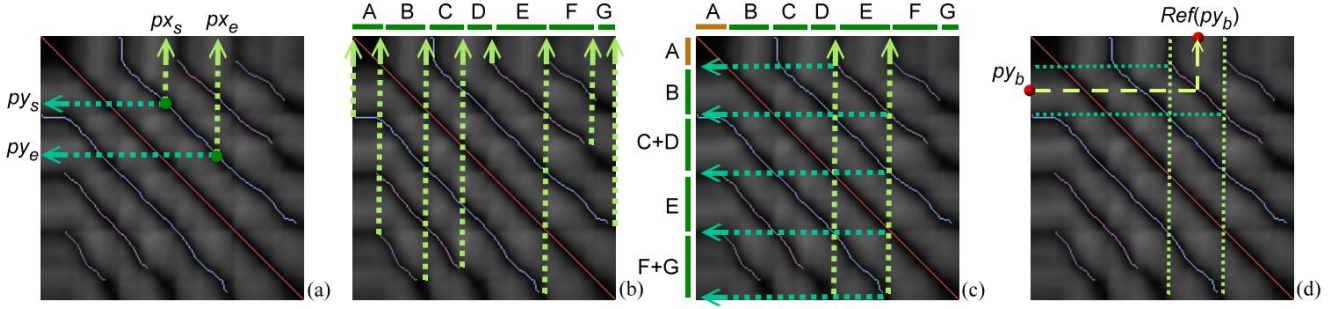


Fig. 7. Similar posture paths and repeated clip extraction of a simple ballet example. (a) The similar posture paths. (b) By projecting end-points of similar posture paths, we can divide the sequence into seven candidate clips. (c) Candidate clip “E”, “B”, “C+D”, “F+G” are in the same equivalent class of repetition. (d) Per-frame mapping from “B” to “E” can be retrieved through the similar posture path segment.

candidates as shown in Fig. 7b.

In Fig. 7c, if we take interval E as our primary clip, there will be three repeated intervals B , $C+D$, $F+G$. If we take interval C as the primary clip, its repetition becomes F , partial B and partial E . More generally, we can also take both C and E as primary ones. Selection of primary and repeated pairs significantly affects the following data compression.

Therefore, we formulate an objective cost for compressing primary and repeated clips. The exact compression costs are unknown at the current clipping phase. However, a sequence with more variations needs more samples and bits for approximation. Therefore, as the first term in (2), we define our expected cost to record the relative variation in a primary clip as standard deviation of the target motion interval $[s, e]$ multiplied by the frame number. Besides, we also have to record the median posture compared to the initial posture, as the second term in (2). We alleviate the extra computation through the pixel values in the self-distance plot.

$$\begin{aligned} \text{PriCost}(s, e) \\ = \text{stdev}([Dist(m, f)]_{f=s \rightarrow e}) \cdot (e - s + 1) + Dist(m, init) \end{aligned} \quad (2)$$

, where s and e are the start and end frame indices of a target interval; m is the median frame number; $init$ is the frame number of initial posture; stdev is the standard deviation function. $Dist$, the L2-norm of coefficient differences between two frames i and j , can be retrieved from the corresponding pixel (i, j) in the self-distance plot.

For a repeated clip, we would like to encode and minimize the difference between the original and referred postures, and the index overhead is relatively small. Hence, a repeated clip cost becomes

$$\text{RepCost}(s, e) = \text{stdev}([Dist(Ref(f), f)]_{f=s \rightarrow e}) \cdot (e - s + 1) \quad (3)$$

, where $Ref(f)$ is the reference frame of f in primary interval. For instance, the reference frame of py_b is $Ref(py_b)$ in Fig. 7d. Additional small constants can also be added to (2) and (3), respectively, to represent the clip overheads.

Hence, our goal becomes to find out the best primary and repeated clip sets that can minimize the following equation.

$$\arg \min \sum_{(s, e) \in \text{PriClipSet}} \text{PriCost}(s, e) + \sum_{(s, e) \in \text{RepClipSet}} \text{RepCost}(s, e) \quad (4)$$

, where PriClipSet contains all primary clip intervals and

RepClipSet contains all repeated clip intervals.

Finding the optimal solution of (4) is a combinatorial optimization problem of extremely high complexity. Assume that we divide the target motion sequence into a certain set of clips with designated lengths respectively. If there is no partial matching, the problem (4) can be reduced to generalized Llyod-Max problem, proven to be NP-complete. Moreover, finding a proper clip set, which needs to evaluate all possible clip numbers in variable lengths, is infeasible in polynomial time as well. To tackle this problem, we propose using a greedy method for a feasible and efficient solution.

We divide the problem into two parts and perform them iteratively. The first part is to define a set of valid intervals as clip candidates. These candidates should have sufficient lengths for temporal coherence and explicit amounts of repetitions. As mentioned above, during initialization, we generate valid intervals by projecting endpoints of similar postures as shown in Fig. 7b. Full or partial intervals that are later selected as primary or repeated clips will be removed from the valid list.

Given a set of valid intervals, the second part is to find clip pairs for maximum gains. In other words, the retrieved repeated clips should be similar to the candidate primary interval and the sum of repeated frame numbers should be larger than that of the primary one. Since the similar posture paths restrict the distance between frames of reference pairs. Once an interval becomes a primary clip, we can obtain the largest encoding improvement by setting all its valid repetitions as repeated clips. If only part of an interval is assigned as a repeated clip, the residual part will be put back to the

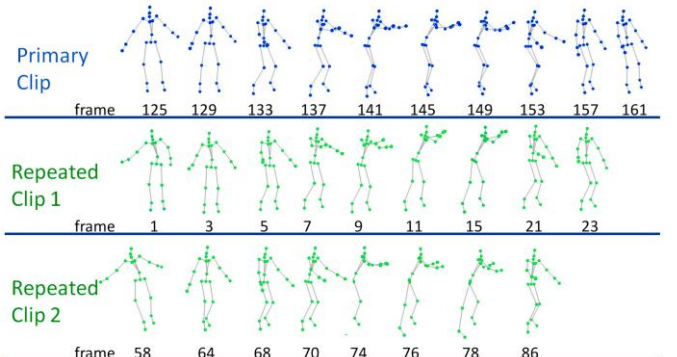


Fig. 8. Examples of repeated clips and referred primary postures. Motion with similar postures but difference speeds or orientations can be extracted.

valid interval set.

We define our repetition encoding gain of an interval as the original coding cost divided by the improved cost:

$$\begin{aligned} \text{Gain}(px_s, px_e) &= \frac{\text{Original coding cost}}{\text{Improved coding cost}} \\ &= \frac{\text{PriCost}(px_s, px_e) + \sum_i \text{PriCost}(py_s^i, py_e^i)}{\text{PriCost}(px_s, px_e) + \sum_i \text{RepCost}(py_s^i, py_e^i)} \end{aligned} \quad (5)$$

, where $\text{PriCost}(px_s, px_e)$ is the cost to encode frame px_s to px_e as a primary clip; $\text{RepCost}(py_s^i, py_e^i)$ is the cost to encode the candidate repeated interval (py_s^i, py_e^i) as a repeated clip; i is the index of repeated intervals.

The procedure of our repeated motion analysis (RMA) is:

1. Given similar posture path sets, project path boundaries to the x axis.
2. Take all projected boundaries as cuts and form a valid projected interval set.
3. Find the projected interval with the maximum gain (5) as a primary clip and label its corresponding unlabeled intervals as repeated clips.
4. Repeat Step 3 until there is no valid interval.
5. Set the primary intervals without repetition as unique clips.

Since there are encoding overheads for each clip, we merge unique or repeated clips less than 200 ms with neighbor clips. With the proposed clip extraction method, repeated clips across the whole motion sequence can efficiently be retrieved. Each frame in a repeated clip is then represented by a reference frame index from the similar posture path and its posture (coefficient) differences from the referred posture. Assume a clip B is referred to clip A and their mapping indices are $\text{Ref}_{B \rightarrow A}$. The mathematical notation is

$$H_{\text{rep}}^{vB}(t) = H_{\text{orig}}^{vB}(t) - H_{\text{pri}}^{vA}(\text{Ref}_{B \rightarrow A}(t)) \quad (6)$$

, where t is the time index; H_{orig}^{vB} is the v^{th} projected trajectory of the original data in clip B ; H_{pri}^{vA} is the v^{th} trajectory of primary clip A ; H_{rep}^{vB} is the v^{th} trajectory represented by repeated clip B . Fig. 8 shows motions from a set of primary and repeated clips.

5 CLIP APPROXIMATION

With the above processes, we utilize PCA on behavior groups to grasp the joint correlations; the short- to mid-term repetitions are extracted by RMA. In the following two subsections, we approximate the motion trajectories within clips by curve fitting and quantization. Subsection 5.3 introduces how we record and rectify high-frequency motion, especially the ground contact, with an inverse kinematics (IK) approach.

5.1 Fitting Coefficient Trajectories

Human motions have high temporal coherence, a sequence of projected coefficients (in primary and unique clips) or coefficient differences (in repeated clips) can be regarded as a curve. To keep efficiency and feature-aware sampling, we fit these trajectories by Catmull-Rom splines. As a modifica-

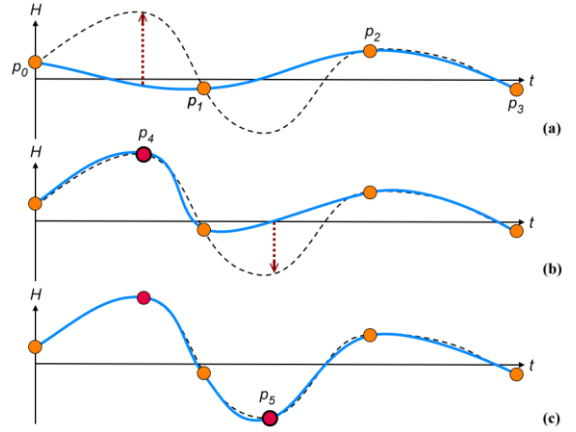


Fig. 9. Conceptual diagram of coefficient approximation by curve fitting. The black dot curve represents a projected coefficient trajectory; the approximate trajectory is shown in blue. (a) Initial control points are placed in a fixed span. (b)(c) By setting the point with maximum error as a new control point, the approximate trajectory can iteratively approach the target one.

tion of Hermite spline, a Catmull-Rom spline passes through all the control points and uses the vector between preceding and following control points as the tangent vector of a control point.

Given a trajectory $H(t)$, the iterative curve fitting procedure is:

1. Set initial control points in a fixed span.
2. Construct an approximate Catmull-Rom curve.
3. Find point p_i with the largest approximate error.
4. If approximate error is larger than threshold, include p_i in the control point set P and go back to Step 3.

Fig. 9 shows an example of this curve fitting method. Compared with least-square or fixed span curve fitting, such dynamically allocated control points preserve more abrupt variations. We put initial control points every 7~9 steps for 120Hz motion data. During the later encoding, for the regularly initialized control points, we only need to record its value H_{pi} ; on the other hand, we need to record both the time index t_{pi} and value H_{pi} for newly included control points.

5.2 Adaptive Quantization of Projected Coefficients

In the above subsection, we have approximated PC projected coefficients by splines. For further compact storage, most research performed quantization to turn floating-point parameters to integer ones. For example, related articles [3] utilized 16-bit uniform quantization to encode coefficients of clip-PCA.

However, using too small-step quantization will make the later process, entropy coding, inefficient. For instance, assume the range of a projected coefficient trajectory, $\max(H_p) - \min(H_p)$, is 3000 millimeters(mm). Under 16-bit quantization, the step is $(\max(H_p) - \min(H_p)) / 2^{16} \cong 0.0458\text{mm}$. Thus, projected coefficients of 115 mm and 115.05mm are distinct in frequency counting during entropy or dictionary coding. In contrast, the error of motion capture data due to noise, imperfect cleaning, etc. and the error of spline approximation can be far larger than the quantization error.

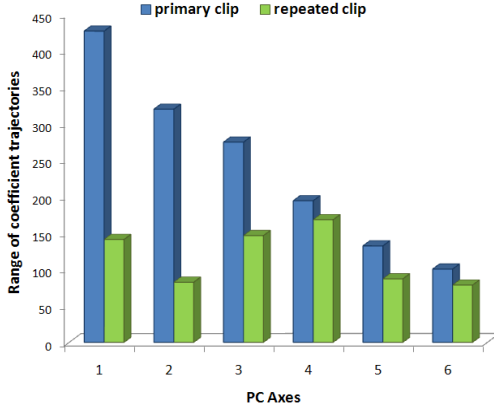


Fig. 10. Coefficient ranges of a hand-waving group in CMU motion data 15_04. Projected coefficient ranges usually decrease as the PC indices increase.

Besides, PCA is designed to order the variances of projected dimension from large to small. As an example in Fig. 10, the ranges of projected coefficients mostly decrease from major to minor PC axes. We can use fewer bits to encode data with smaller ranges.

Hence, we propose using more aggressive uniform quantization strategy. According to the ranges and importance, we use different quantization bits for coefficients of different PC axes in a group. For the v^{th} curve of clip u , we record its minimum value as $\min(Hp^{uv})$; for each group, we evaluate the curve ranges of PC axes as $Range_v$. Therefore, the n -bit quantization step of v^{th} axis is $Range_v / (2^n)$. The quantized integer of Hp_i^{uv} becomes

$$\tilde{Hp}_i^{uv} = \frac{(Hp_i^{uv} - \min(Hp^{uv})) \cdot 2^n}{Range_v} \quad (7)$$

While adding a half of quantization step during dequantization, the maximum quantization error becomes a half of the quantization step; the expected error is a quarter of the quantization step. Therefore, we can evaluate the required bits by a ceiling function according to the range and expected tolerable quantization error.

$$n = \lceil \log(Range_v / E(err_{qt})) - 2 \rceil \quad (8)$$

Our experiment shows that the expected error of the uniform-sampled spline under 95% of major PCs is around 2~3 cm per joint. We set our expected quantization error to 0.6 cm for axes of repeated clips and 0.5 cm for axes of primary clips to reduce the effects on ranges at repeated clips. While scattering the accumulated errors at subspace axes back to joints, they are relatively small. Instead of 16-bit per coefficient, we can therefore encode a coefficient of primary clips by only 5~11 bits and 3~7bits for repeated clips with comparable accuracy. The spare storage can be used to increase the number of spline control points or increase the number of major PC axes.

5.3 IK and Contact Trajectories

Using spline-based approximation at subspace can not always preserve extreme high frequency motion, like ground contact. In our system, we separately record such contact trajectories for later rectification by IK.

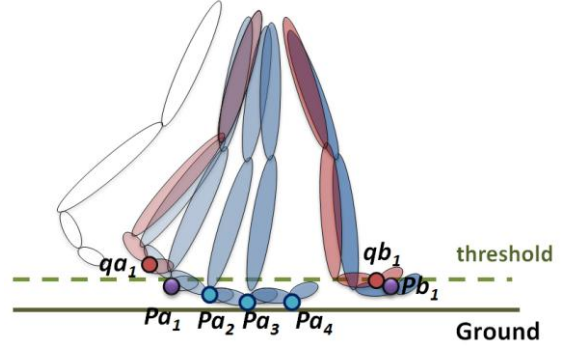


Fig. 11. Encoding contact trajectories. As a new contact motion occurs, we create a contact segment a . We store the difference between the reconstructed point qa_1 and the starting point pa_1 ; for the following contact points, we store the small motion difference $pa_i - pa_{i-1}$. As a large contact motion occurs, we create a new segment b .

The value $y=0$ may not always be the ground height in different motion capture data. We cluster the y positions of joints and take the mean value of the lowest cluster as the floor height.

As a joint near the floor height, we start to record the joint's x, y, z positions until its y value leaves the range or the contact point moves more than 2.5cm between two frames. As shown in Fig. 11, for the starting contact point, we use 8-bit quantization to store its x, y, z differences compared to our reconstructed joint positions, respectively; for the following contact points, we use 4-bit quantization to store the x, y, z differences compared to the preceding points. Other identification method, e.g. learning system of L. Ikemoto et al. [19], could also be used to retrieve contacts besides the ground.

5.4 Details of Data Encoding

Except the minimum and maximum values of quantization, most floating-point data can also be represented by integer through adaptive quantization. We use 16-bit quantization for mean postures in groups; use 8-bit and 6-bit for $\min(Hp^{uv})$ in primary and repeated clips, respectively. For elements of PC axes, whose range is around 0.5~0.6, we can even use 12-bit quantization with only few errors.

Other time-varying data can also benefit from quantization or their motion differences. Root orientation and translations are encoded by 16-bit quantization of Catmull-Rom spline control points. The posture reference indices in repeated clips can also be encoded by four to five bits of differences between B-spline control points. At last, we utilize RAR compression for further lossless encoding, and it can be replaced by other dictionary or entropy-based compression. Since a byte is the minimum counting unit in such methods. For an n -bit quantized value ($n < 8$), we still use one byte space before lossless compression, but it gains more compression due to highly frequent occurrence.

To avoid propagation of primary clip error to repeated ones, we encode and approximate the primary motion first. Coefficient differences of repeated clips are extracted through the reconstructed coefficients instead of the original ones at referred primary clips.

TABLE 1
THE COMPARISON OF THE PROPOSED METHOD WITH CLIP-PCA METHOD FOR A 120HZ MOCAP DATA SET

Method	PC variance	Avg. joint error	Compression ratio	Encoding fps	Decoding fps	Avg. valid PC axes	Avg. quant. bits (Primary/Repeat)	Clip % (Pri/Uni/Rep)
proposed	98%	1.60 cm	88.7 : 1	372.4	4286.8	12.56	6.44/ 4.89	24%/8%/68%
proposed	95%	2.39 cm	108.4 : 1	425.1	4383.2	8.41	6.68/ 5.18	25%/8%/67%
Clip-PCA[3]	95%	2.32 cm	39.8 : 1	530.5	4902.3	20.33	16	—

The target data set is a subset of CMU mocap database (subject: 6, 15, 16, 17, 35, 94 and 135), where 31 joints are recorded; the frame amount (with padded postures to fit clip length in [3]) is 341472; the original binary data in 32-bit floating point number is 130.9MB.

TABLE 2
THE COMPARISON OF THE PROPOSED METHOD WITH CLIP-PCA METHOD FOR A 30HZ MOCAP DATA SET

Method	PC variance	Avg. joint error	Compression ratio	Encoding fps	Decoding fps	Avg. valid PC axes	Avg. quant. bits (Primary/Repeat)	Clip % (Pri/Uni/Rep)
proposed	98%	2.25 cm	28.9 : 1	129.5	3248.8	17.68	6.25/4.84	10%/3%/87%
proposed	95%	2.89 cm	35.9 : 1	124.9	3644.2	11.53	6.33/5.14	11%/3%/86%
Clip-PCA[3]	95%	3.32 cm	12.3 : 1	786.5	3576.2	36.46	16	—

The target data set is recorded in 30Hz with 18 joints. The frame amount (with padded postures) is 91920; the original binary data in 32-bit floating point number is 20.96MB.

5.5 Decoding and Data Reconstruction

As shown in Fig. 5, the reconstruction process is more straightforward and efficient in computation than the compression process. The first step is to decompress RAR encoded data and dequantize the quantized values. Through spline interpolation, we can reconstruct the whole sequences of projected PC coefficients or coefficient differences.

For primary and unique clips, these coefficient trajectories can directly be projected from PCA to the original space. For repeated clips, the reconstructed trajectories are used to compensate the differences from its referred primary coefficients before PCA reverse-projection. The decompressed root orientations and translation vectors are then applied to move the skeleton from their aligned postures.

Due to lossy compression, there may be discontinuities in joint motions at clip boundaries; we connect clips by motion continuity method proposed in [3]. Besides, low-pass filtering with a 60ms-wide window is also applied to smooth the possible discontinuity between group and clip boundaries. After transferring the reconstructed motion from virtual marker sets to joint angles, we use jacobian-based IK to rectify contact motion according to the contact trajectories.

6 EXPERIMENTS AND DISCUSSION

In addition to the proposed method, we reproduce O. Arikian's method of clip-based PCA with IK [3]. Besides, authors of PGA-IK compression also provide their reconstructed motion data as listed in the reference [13]. We carried out experiments and user evaluations with three different data sets. These experiments performed on a desktop with 3.0 GHz quad-core CPU and 3.25GB memory. Currently, only RMA and IK are implemented in multi-threading without specific performance optimization.

6.1 Experiment

The first experiment compared the proposed method with clip PCA method [3] for a motion capture data set with 31 joints in 120fps. The BVH data set including sporting, dancing, walking and running motion, are transformed from subject no.6, 15, 16, 17, 35, 94 and 135 of CMU motion capture database [9]. Compression methods are applied to data of each subject, respectively. Besides, because the compared method used a fixed clip length, we padded unfilled clips with their last postures.

The original data in 32-bit floating-point numbers are about 130.9MB. We set the clip length of clip PCA method as 200ms and 95% of variances were covered by valid PC axes. The proposed methods with 95% variance and 98% variance PC subspace were also performed. The upper-bound of control point amount per trajectory (PC axis) in spline approximation was set identical to these methods.

As the result shown in Table 1, the proposed method with 95% PC variance reaches a 108:1 compression ratio with comparable accuracy. The spare space can be used for more accurate PC space like PCA space of 98% variance.

The second experiment performed the comparison on a motion data set with 18 joints at 30fps. The captured data are aimed at real-time application, such as 3D games. We set the clip length of clip PCA method as 533ms for balance of accuracy and compression gains.

This is a challenging data set since fewer temporal redundancy and joint correlations can be utilized. Besides, it also includes several intense motions. As shown in Table 2, the proposed feature-aware method is more accurate and nearly three times of compression benefits than the compared method.

In the third experiment, we compare the proposed method (with major PCs of 95% variances), clip-PCA method [3] with the reconstructed data of PGA-IK [13]. The five original motion data are also from CMU motion database. As the result in Table 3, PGA-IK method get benefits from run-time optimization and can have consid-

TABLE 3

THE COMPARISON OF THE PROPOSED METHOD WITH CLIP-PCA AND PGA-IK METHODS FOR 120HZ MOCAP CLIPS

09_06: 141 frames (Running, short)				
Method	Avg. joint error	Comp. ratio	Encoding fps	Decoding fps
proposed	2.26 cm	22.8 : 1	437.9	1719.5
Clip-PCA[3]	1.76 cm	2.5 : 1	22.5	2285.7
PGA-IK[13]	3.02 cm	18 : 1	—	126.9

17_08: 6179 frames (Walking, slow)				
Method	Avg. joint error	Comp. ratio	Encoding fps	Decoding fps
proposed	1.57 cm	91.7 : 2	539.1	2945.2
Clip-PCA[3]	1.63 cm	30 : 1	778.8	2979.8
PGA-IK[13]	1.93 cm	182 : 1	—	61.7

15_04: 22549 frames (Various, dancing, boxing)				
Method	Avg. joint error	Comp. ratio	Encoding fps	Decoding fps
proposed	1.86 cm	114.7 : 1	219.1	2530.8
Clip-PCA[3]	2.30 cm	42.5 : 1	705.6	2693.7
PGA-IK[13]	1.63 cm	69 : 1	—	32.7

85_12: 4499 frames (Breakdance)				
Method	Avg. joint error	Comp. ratio	Encoding fps	Decoding fps
proposed	5.01 cm	60.1 : 1	506.4	3096.4
Clip-PCA[3]	4.78 cm	11.1 : 1	585.7	3903.1
PGA-IK[13]	4.02 cm	97 : 1	—	49.0

17_10: 2783 frames (Boxing)				
Method	Avg. joint error	Comp. ratio	Encoding fps	Decoding fps
proposed	2.65 cm	62.6 : 1	486.6	4288.1
ClipPCA[3]	3.16 cm	9.2 : 1	376.7	4569.8
PGA-IK[13]	3.15 cm	61 : 1	—	62.6

The PGA-IK motions, as listed in [13], are provided by the authors for comparison. The encoding fps does not list is the paper.

erable compression gains for a slow walking motion with smooth end-effector moves, like 17_08 data.

On the other hand, the proposed and clip-PCA method finding the similar motion clips across motion data can improve compression ratios as the motion length increases. For the 15_04 data, our compression ratio can reach 114:1. Besides, the encoding and decoding performance of these two methods are adequate to real-time applications.

Furthermore, we compare the adaptive-range and uniform methods for coefficient quantization. The uniform method evaluated the quantization steps according to the designated bits and range of each group, respectively. Table 4 shows that using 16-bit quantization has barely gains in accuracy but dramatically lowers the compression ratios. According to (7) and (8), 9- and 7-bit uniform quantizations to primary and repeated data fit for our expected quantization error. In this case, the compression ratios are closer to those of adaptive methods. That is because the small magnitude data at minor axes or repeated clips have high frequency of occurrence and result in high

TABLE 4

THE COMPARISON OF ADAPTIVE AND UNIFORM QUANTIZATION FOR A 120HZ MOCAP DATA SET

Quant. bits (Pri/Rep)	PC variance	Avg. joint error	Compression ratio
uniform(16/16)	98%	1.580 cm	55.3 : 1
uniform(9/7)	98%	1.586 cm	81.0 : 1
adaptive	98%	1.598 cm	88.7 : 1
uniform(16/16)	95%	2.385 cm	72.6 : 1
uniform(9/7)	95%	2.388 cm	101.0 : 1
adaptive	95%	2.394 cm	108.4 : 1

The target data set is the same as that in Table 1.

effectiveness at entropy or dictionary coding, but the adaptive method with trajectory-level *min* bases and ranges still have superior ratios.

6.2 User Evaluation

To evaluate the perceptual effects by different compression methods, we perform two data sets for user study, and each of them has three topics: “defect counting”, “naturalness” and “faithfulness”.

In the “defect counting” and “naturalness”, a user had to watch the original and all reconstructed motions in random order without any prior information. She or he then reported the numbers of apparent defects and scored the naturalness (one is very poor; three is acceptable; five is very satisfactory). In “faithfulness” topic, the original and the reconstructed motion were played simultaneously. Users were informed of the original motion and were asked to score the degree of similarity from one to five points.

The first evaluation performed eleven motion clips of 20 second by the proposed and clip-PCA methods. Twenty one volunteers with graphics or animation-related background participated in the test. These volunteers were not informed of how these compression methods worked. The result is shown in the first row of Table 5.

The second evaluation compared the whole motion sequences by three methods as listed in Table 3. Fourteen volunteers participated in this test. The result is shown in the 2nd to 6th rows in Table 5.

Interestingly, due to imperfect cleaning, the original motion capture data sometimes even have more defects and get lower scores in “naturalness”. In contrast, those abrupt tremble motions could usually be omitted by PCA or PGA subspace projection.

6.3 Discussion

We discuss the advantages and weaknesses of these methods according to our experiments.

PCA on clips fitted by bezier curves

This seminal work [3] proposed using virtual markers at position domains to linearly encode joint angles by PCA. The uncompressed dimension of a clip is 3 (x, y, z DOFs) $\times$ 4 (Bezier control points) $\times$ 3 (virtual markers) $\times$ *number_of_joints*. For a group with 95% PC variances, the valid PC axes are more than two times of those by frame-based PCA. The total amounts of projected coefficients, contrarily, are nearly a half of those by frame-based PCA

TABLE 5
USER EVALUATION OF THE ORIGINAL MOTIONS AND RECONSTRUCTED MOTIONS OF THREE METHODS

Motion	Original			The proposed method			Clip-based PCA with IK [3]			PGA-IK [13]		
	Defect	Natural.	Faithful.	Defect	Natural.	Faithful.	Defect	Natural.	Faithful.	Defect	Natural.	Faithful.
Mix set	6.48	3.56	—	5.81	3.61	3.81	7.48	3.39	3.54	—	—	—
09_06	0	4.86	—	0	4.86	4.86	0	4.86	4.86	0	4.86	4.86
17_08	11.62	2.54	—	1.85	4.31	4.46	2.77	4.08	4.77	1.31	4.38	4.35
15_04	12.14	3.71	—	8.07	3.75	3.96	7.79	3.11	3.21	6.14	4.11	3.18
85_12	4.79	4.14	—	4.21	3.96	3.93	7.14	3.57	3.5	4.57	3.64	3.82
17_10	8.29	3	—	1.14	4.64	4.57	3.21	3.89	4.57	5.29	3.29	4.14

The mix data set contains eleven clips of walking, dancing, jumping, sporting motions from dataset in Table 1 and 2. Others are motions in Table 3. “Defect” represents the number of visible defects noticed by subjects; “natural.” represents the score of naturalness (from 1 to 5. 1: very poor, 3: acceptable, 5: very satisfactory); “faithful.” represents the score of faithfulness (from 1 to 5).

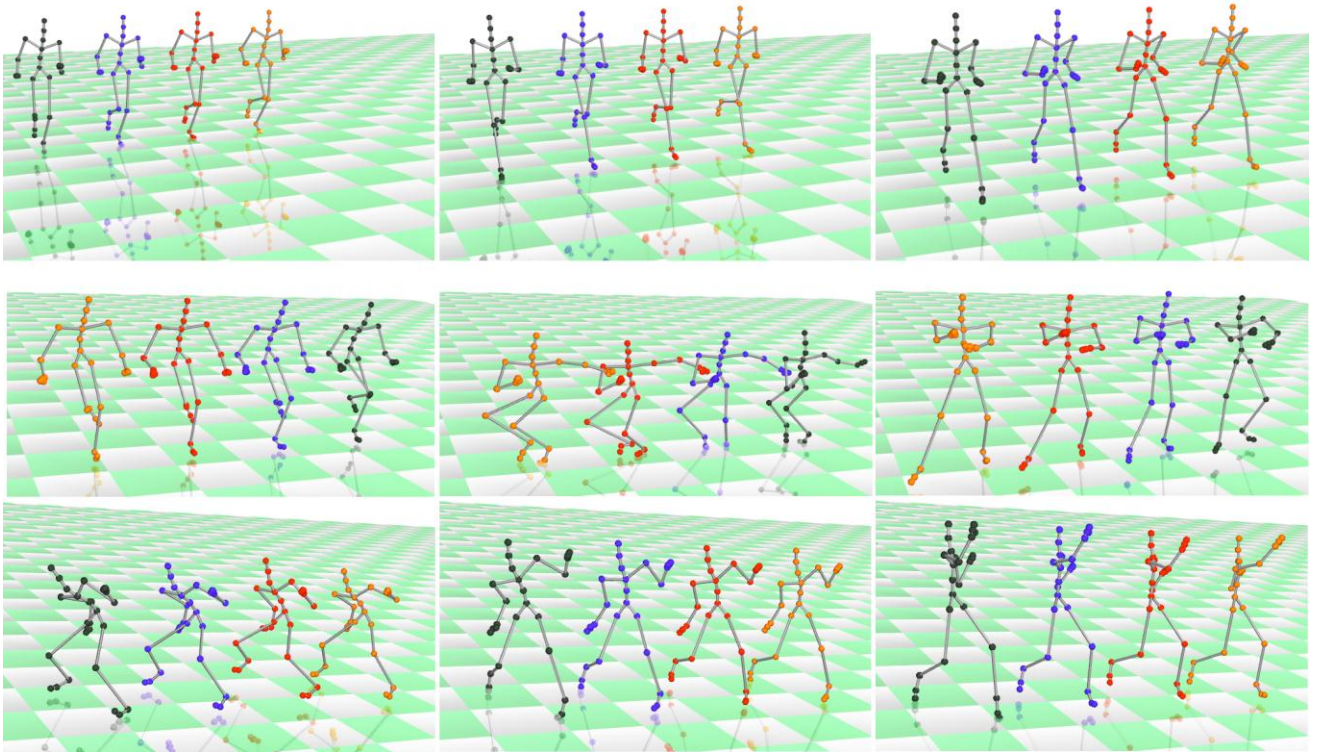


Fig. 12. Comparison of the original motion and decompressed motions by three methods. The original motions are shown in orange; the proposed results are shown in red; results of clip-based PCA [3] are shown in indigo; results of PGA-IK [13] are shown in black.

with an identical control point sampling period. The elements of clip-based PCA axes are more than 4 (Bezier control points) * 2 (valid PC axes) times of that by frame-based PCA. It is a serious overhead. As shown in Table 3, for short motion clips, these overheads substantially lower the compression ratios. Besides, since it takes uniform clip lengths and least-square curve fitting. Users reported that this method sometimes cannot precisely follow abrupt motions especially at 30Hz data sets. On the other hand, due to its regularity, this method is of highest computation performance in both encoding and decoding. The overhead problem can alleviate as the group data increase. Clip-based PCA method is adequate to large motion database or incremental compressions, where the PC axes have been pre-stored or computed.

PGA-IK

Due to the temporal discontinuity penalty in its opti-

mization formula, this method has few discontinuous defects and high naturalness scores in user study. It also has considerable compression gains for simple or smooth motion. However, as the motion getting complex, this method without motion segmentation, needs more PGA axes and control points for end effector fitting. Besides, it only records the end-effector motions. This method got lower scores in faithfulness, especially at elbow and knee angles. Moreover, from our experience, different solvers with various parameters could result in different postures by optimization.

The intensive computation makes PGA-IK not appropriate for real-time systems with moderate computing ability. Nevertheless, it can be a highly compact representation for short motion clips or motion with plenty environment contacts. It may easily be combined with other optimization-based motion editing methods.

The proposed method

The proposed method used feature-aware clipping and curve fitting. It got satisfactory results in faithfulness. We use adaptive quantization and compact virtual markers with frame-based PCA. The uncompressed dimension of a frame is only 3 (x, y, z DOFs) $\times$ 2 (virtual markers) $\times$ *number_of_joints*. The PC axes overheads are much fewer than the clip-PCA method. In addition, the RMA can reduce the coefficient ranges of a repeated clip about two to sixteen times compared to primary ones. The adaptive quantization can further reduce the bit usages with few additional errors.

The weakness of the proposed method is that we use external IK for contact motion. If the contact point is far from the reconstructed point, the IK postures can be dissimilar to the decompressed ones and result in discontinuity. We propagate the effects of IK postures about 30ms to alleviate this situation.

Notwithstanding, we believe the proposed method is a good balance of computation and compression efficiency. It provides a substantial compression ratio for mid- to large-size data. It is also of high computational efficiency in encoding and decompression and is adequate to large data sets or short clips for real-time applications. Example motions of these three methods are shown in Fig.12. Please refer to the demo video for further comparison.

7 CONCLUSION AND FUTURE WORK

In this paper, a novel motion representation and compression method is presented. We formulate and associate the repetition matching with data encoding as an optimization problem. A feasible algorithm, repeated motion analysis, is proposed to efficiently find the mapping indices that can maximize the compression gains with comparable accuracy.

With the feature-aware motion clipping and curve fitting, the proposed method can grasp both intensive and gradual motions. The proposed adaptive method finds the appropriate quantization steps and can further compact the data size.

We also performed experiments and user evaluation to compare the proposed method with two state-of-the-art methods. The results demonstrate that the proposed method can simultaneously achieve a significant compression ratio and high computational efficiency.

The proposed system is also possibly extended to online compression if we apply behavior grouping and RMA between the new motion and existing data. Besides motion compression, the clip segmentation, parameters and reference indices extracted by RMA inherently provide adequate notations for further progressive motion representation, motion data classification, retrieval or editing. We believe the proposed repetition analysis, adaptive data fitting and quantization can also be applied to other applications like multimedia or geometric model processing.

ACKNOWLEDGMENT

The authors appreciate the helpful comments from the anonymous reviewers. They also thank volunteers participating in the user evaluation. Especially, they thank M.

Tournier and other authors of PGA-IK method, who cordially provided their data for comparison. This paper was partially supported by the National Science Council, Taiwan under grant no. NSC 98-2221-E-009 -151 and 99-2221-E-009-136.

REFERENCES

- [1] J. Barbic, A. Safonova, J. Pan, C. Faloutsos, J. Hodgins and N. Pollard, "Segmenting Motion Capture Data into Distinct Behaviors," *Proc. Graphics Interface (GI '04)*, pp. 185-194, 2004.
- [2] G. Liu and L. McMillan, "Segment-based Human Motion Compression," *Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation (SCA '06)*, pp. 127-135, 2006.
- [3] O. Arikan, "Compression of Motion Capture Database," *ACM Trans. Graphics*, vol. 25, no. 3, pp. 890-897, 2006.
- [4] A. Schödl, R. Szeliski, D. Salesin and I. Essa, "Video Textures," *Proc. ACM SIGGRAPH*, pp. 489-498, 2000.
- [5] O. Arikan and D. Forsyth, "Interactive Motion Generation from Examples," *ACM Trans. Graphics*, vol. 21, no. 3, pp. 483-490, 2002.
- [6] J. Lee, J. Chai, P. Reitsma, J. Hodgins and N. Pollard, "Interactive Control of Avatars Animated with Human Motion Data," *ACM Trans. on Graphics*, vol. 21, no.3, pp. 491-500, 2002.
- [7] L. Kovar, M. Gleicher and F. Pighin, "Motion Graph," *ACM Trans. Graphics*, vol. 21, no. 3, pp. 473-482, 2002.
- [8] 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, 2004.
- [9] CMU Graphics Lab Motion Capture Database, <http://mocap.cs.cmu.edu/>.
- [10] H. Hoppe, "Progressive Meshes," *Proc. ACM SIGGRAPH*, pp. 99-109, 1996.
- [11] M. Alexa and W. Muller, "Representing Animations by Principal Components," *Computer Graphics Forum*, vol. 19, no. 3, pp. 411-418, August 2000.
- [12] S. Chattopadhyay, S.M. Bhandarkar and K. Li, "Human Motion Capture Data Compression by Model-based Indexing: A Power Aware Approach," *IEEE Trans. Visualization and Computer Graphics*, vol. 13, no. 1, pp. 5-14, Jan/Feb 2007.
- [13] M. Tournier, X. Wu, C. Nicolas, A. Élise and R. Lionel, "Motion Compression using Principal Geodesics Analysis," *Computer Graphics Forum*, vol.28, no.2, pp.355-364, 2009.
- [14] Y. Li, T. Wang and H.Y. Shum, "Motion texture: A Two-level Statistical Model for Character Motion Synthesis," *ACM Trans. Graphics*, vol. 21, no. 3, pp. 465-472, 2002.
- [15] R. Cutler and L.S. Davis, "Robust Real-time Periodic Motion Detection, Analysis, and Application," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 781-796, 2000.
- [16] Q. Gu, J. Peng and Z. Deng, "Compression of Motion Capture Data Using Motion Pattern Indexing," *Computer Graphics Forum*, vol. 27, no. 9, 2008.
- [17] B. Horn, "Closed-form Solution of Absolute Orientation Using Unit Quaternions," *J. the Optical Society of America A*, vol. 4, no. 4, pp. 629-642, 1987.
- [18] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888-905, 2000.
- [19] L. Ikemoto, O. Arikan and D. Forsyth, "Knowing when to put your foot down," *Proc. ACM Symp. on Interactive 3D Graphics and Games (I3D'05)*, pp. 49-53, 2005.

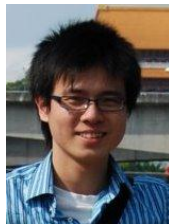


I-Chen Lin received the B.S. and Ph.D. degree in computer science from National Taiwan University, Taiwan, in 1998 and 2003, respectively. In 2005, he joined Department of Computer Science, National Chiao Tung University, Taiwan. He is currently an assistant professor. His research interests include computer graphics, animation, image-based modeling and virtual reality. He is a member of IEEE and ACM SIG-

GRAPH.



Jen-Yu Peng received the B.S. and M.S. degree in computer science from National Dong Hwa University, Taiwan, in 2003 and 2005, respectively. He is a PhD student in the Department of Computer Science, National Chiao Tung University, Taiwan. His research interests include interactive computer animation and game programming.



Chao-Chih Lin received the B.S. degree in electronics engineering from National Cheng Kung University, Taiwan, in 2005, and he received the M.S. degree in computer science from National Chiao Tung University in 2007. In 2009, he joined XPEC Entertainment Inc. His research interests include facial and character animation and surface approximation.



Ming-Han Tsai received the B.S. and M.S. degree in computer science from National Chiao Tung University, Taiwan, in 2007 and 2009, respectively. He is now a Ph.D. student in the Department of Computer Science, National Chiao Tung University, Taiwan. His research interests include image-based 3D modeling and motion capture.