

Article

Keyframe Extraction from Human Motion Capture Data Based on a Multiple Population Genetic Algorithm

Qiang Zhang *, Shulu Zhang and Dongsheng Zhou

Key Laboratory of Advanced Design and Intelligent Computing, Ministry of Education, Dalian University, Dalian 116622, China; E-Mails: zhangshulu163@163.com (S.Z.); zhoudongsheng@dlu.edu.cn (D.Z.)

* Author to whom correspondence should be addressed; E-Mail: zhangq@dlu.edu.cn; Tel.: +86-411-8740-3728; Fax: +86-411-8740-3733.

Received: 7 July 2014; in revised form: 5 November 2014 / Accepted: 7 November 2014 /

Published: 12 November 2014

Abstract: To reduce reconstruction errors during keyframe extraction and to control the optimal compression ratio, this study proposes a method for keyframe extraction from human motion capture data based on a multiple population genetic algorithm. The fitness function is defined to meet the goals of minimal reconstruction errors and the optimal compression rate, where multiple initial populations are subjected to co-evolution. The multiple population genetic algorithm considers global and local search. Experimental results showed that the algorithm can effectively extract the keyframe from motion capture data and it satisfied the desired reconstruction error.

Keywords: keyframe extraction; motion capture data; multiple population genetic algorithm

1. Introduction

Motion capture data with highly realistic and real-time processing capabilities has important applications in simulation training, film and television production, entertainment, virtual reality, and many other fields. However, the original motion capture data comprise large volumes of data and their reusability is poor. In order to facilitate the transmission, storage, and reuse of motion capture data, many keyframe extraction techniques have been proposed, which can reduce the data storage space and accelerate data processing.

The keyframe extraction techniques applied to motion capture data extract representative frames from the motion data based on the entire movement sequence. At present, popular keyframe extraction methods can be divided into categories based on clustering, curve simplification, and others.

Methods based on clustering divide the motion sequence into groups with similar features, before selecting one frame from each class as the keyframe. For example, Liu *et al.* [1] divided N-frames of motion data into K sets according to the features of the motion data, before selecting the first frame in each set as the keyframe. Park and Shin [2] employed a quaternion to represent the motion sequences, before applying PCA (Principal Component Analysis) and the K-means method to process the motion data, and the scattered data were used for interpolation to extract the keyframe. Zhu and Wang [3] proposed a similar clustering method for extracting keyframes, where the high-dimensional original data were mapped to a low-dimensional space and the mean squared error was then used to segment the low-dimensional data. The segmentation point reflected an attitude change in the motion data, thus the segmentation point was selected as the keyframe. Halit and Capin [4] also reduced the dimensionality of the data and obtained the motion feature based on the Gaussian-weighted average of the frame, before selecting frames higher than the average as candidate frames and then removing the unimportant frames to yield the final keyframe.

In methods based on curve simplification, each frame of the motion capture data is regarded as a high-dimensional space point, where these points in time comprise a trajectory curve. The curve simplification algorithm is used to extract feature points to obtain the keyframe from the motion sequence. Lim and Thalmann [5] extended the curve simplification algorithm and determined the highest curve that indicated the motion capture data, before finally extracting the extreme points of the curve as the keyframe. However, this method required repeated attempts with different error levels, depending on the compression rate, to meet the requirements. Togawa and Okuda [6] proposed a key frame extraction method based on location. First, movement was regarded as a set of curves formed by all the joint rotations of each frame data and the locations of all the joints were calculated to determine the key points to reduce the number of keyframes, before finally obtaining the keyframe. Assa *et al.* [7] used multidimensional scaling analysis to map the movement of high-dimensional data to low-dimensional space, and then used the simplified approach curves in the low-dimensional space to extract the keyframes. Yang *et al.* [8] proposed the double curve simplification algorithm, which utilized the set of angles formed by human limb bones and the center of the bones as characteristics, before using a stratified curve simplification algorithm to extract the keyframe. Peng [9] proposed a method based on the center distance to extract the keyframe. Zhang *et al.* [10] extracted the keyframe based on the joint distance by using nine joint features to represent the original motion sequences, before applying PCA to reduce the dimensionality of the data and obtain one-dimensional curves. The extreme curve was then extracted as the initial keyframe point and the final keyframe interpolation procedure was based on the threshold between the extreme points.

In another keyframe extraction method, Liu [11] used spherical linear interpolation to replace linear interpolation, and also introduced the speed error, which increased the measurement of the joint rotation characteristics, where the frame cut method was used to obtain a reconstruction error curve and the keyframe was extracted automatically based on the best compression ratio. However, this method could not guarantee that the keyframe obtained in the optimal compression ratio had the minimum reconstruction error. Lee *et al.* [12] constructed an objective function that treated the key

frames and the compression ratio as the goal, before using genetic algorithms to extract key frames from a dynamic grid sequence. Liu *et al.* [13,14] improved this method and proposed a keyframe extraction algorithm based on a simplex hybrid genetic algorithm (SMHGA) in order to increase the local search ability of the genetic algorithm, which used the relatively powerful local search simplex method. This method could eliminate the parameter adjustment process and it improved the local search ability. However, genetic algorithms are readily trapped by local extremum in the background knowledge, which makes it difficult to find the global optimal solution. Cai *et al.* [15] proposed a keyframe extraction method for motion capture data based on the compression ratio or reconstruction error, which was divided into two stages that employed a preselected frame and optimization selection based on the reconstruction error.

The present study presents a human motion capture data keyframe extraction method based on a multiple population genetic algorithm (MPGA), where we define a fitness function that assesses the reasonableness of the keyframe extraction result. This method sets the minimum fitness value as the optimization target. The algorithm does not require a manually specified threshold and the optimal keyframe set can be extracted.

2. Multiple Population Genetic Algorithm

Genetic algorithms are highly parallel, global, randomized, and adaptive optimization probabilistic search algorithms that imitate natural selection and the mechanism of evolution. They are highly robust and they have the capacity for global search. However, genetic algorithms have many shortcomings and defects, such as the immature convergence problem. Thus, the MPGA was introduced to address these problems.

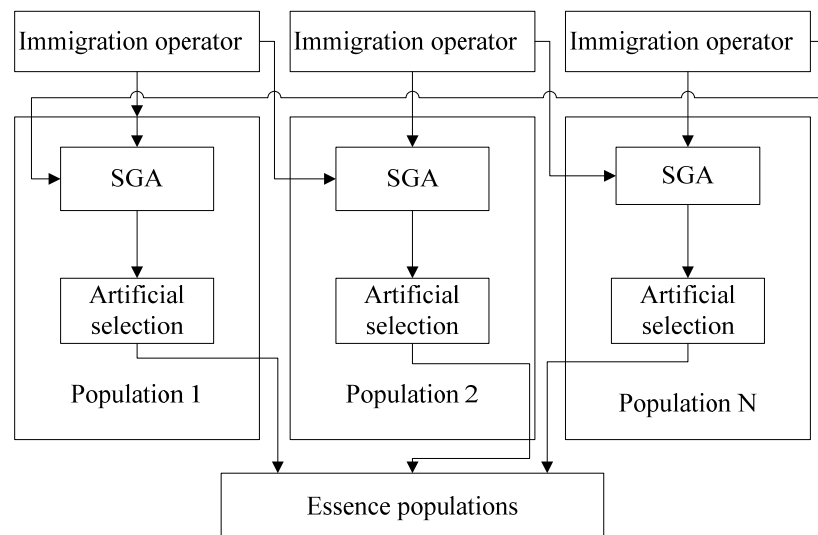
MPGA breaks with the standard genetic algorithm (SGA) framework that relies on the evolution of a single group of genes by introducing multiple populations to optimize the search process. The different populations are assigned with different control parameters to achieve different search purposes. The algorithm also considers global search and local search. Various groups are independent and they communicate with each other via an immigration operator, where an operator regularly (every generation) introduces the best individual obtained from each population in the evolution process into the other populations, thereby achieving information exchange between populations. The specific operation rule requires that the best individual in the source population replaces the worst individual in the target population. During each generation of evolution, the best individual in other populations is selected with an artificial selection operator, thereby preserving the best individuals in the overall population.

SMHGA and MPGA belong to the same group of genetic algorithms, which all have the ability to eliminate the parameter adjustment process and improve the capacity for local search. The SMHGA is a hybrid approach that combines GA with a probabilistic simplex (SM), which is a method for directing promising search in a specific direction to speed up convergence and improve the quality of the solutions [14]. SMHGA uses a single group to evolve, whereas the MPGA employs multiple populations to optimize the search process; thus, it has a large search space and it can readily determine the global optimal value.

2.1. Structure of MPGA

In this section, we introduce the structure of MPGA. The structure of MPGA can be seen in Figure 1.

Figure 1. Structure of the multiple population genetic algorithm.



2.2. Description of MPGA

Step 1. Generate the initial population. Using a binary coding to generate *MP* initial populations *Chrom*, each initial population *Chrom* contains *N* individuals. Different populations are endowed with different control parameters (e.g., mutation probability and crossover probability).

Step 2. Calculate the fitness of each individual in each initial population *Chrom* and then sort the fitness values from highest to lowest within the scope of the various groups.

Step 3. Apply selection, crossover, and mutation operations to each separate population. Different populations are independent but co-evolution occurs.

Step 4. Immigration operations. Each initial population is relatively independent but the exchange of information occurs between populations via the immigration operation, where the worst individual in the target population is replaced by the best individual in the source population.

Step 5. Artificial selection operations. During each generation of evolution, select the best individual in other populations with an artificial selection operator and maintain the best individuals in the essence population.

Step 6. Find the best individual in the essence population. Check whether the current optimal value and the previous optimal value are the same. If this is not the case, update the optimal values.

Step 7. Check whether the evolutionary process satisfies the termination condition. If it meets the condition, the algorithm proceeds to Step 8. If it is not satisfied, then return to Step 2.

Step 8. End of the evolutionary process. Output the optimal value and the best individual to obtain the optimal value.

The algorithm is repeated in an iterative manner and the algorithm ends when the number of iterations satisfies the value in the initial settings. The individuals in the essence population can obtain

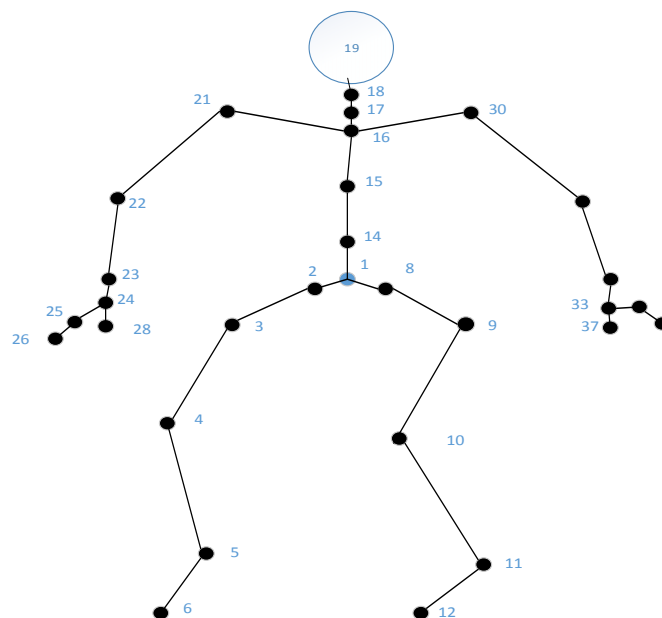
a good optimized value and the best value is then selected as the movement sequence keyframe. Genetic manipulation is terminated when the following condition is satisfied: the optimal value remains unchanged *MAXGEN* times.

3. Extracting Keyframes Based on MPGA

3.1. Motion Capture Data

In this study, the format of the motion capture data used in the experimental section is BVH. The skeleton model is employed, as shown in Figure 2. This is the standard skeleton from the CMU (Carnegie Mellon University) human database, which includes 31 joints where the ROOT is the hip joint root.

Figure 2. Standard skeleton.



The captured data essentially comprise the period of the time sequence and each point in the sequence represents the data from each frame. The data from the frame comprises 31 joints. Therefore, human motion capture data can be defined by the following formula:

$$M = \{F(1) \quad F(2) \quad \dots \quad F(i) \quad \dots \quad F(n)\}$$

where n is the total number of frames and each frame can be defined by the following formula:

$$F(i) = \{P_i(1) \quad R_i(1) \quad R_i(2) \quad \dots \quad R_i(t) \quad \dots \quad R_i(31)\}$$

where $P_i(1)$ represents the translation data from the ROOT node, $R_i(1)$ represents the rotation of the ROOT node, and $R_i(t)$ ($t = 2, \dots, 31$) is the local rotation, excluding the ROOT node.

3.2. Encoding and Initialization of the Population

To determine whether a frame in the motion sequence is the keyframe, it can be labeled with two state variables: 0 or 1. Thus, binary code is used to encode the population in this study. After encoding

the motion sequence, we get a sequence of binary numbers represented by 0–1, 0 or 1 represents a frame in the motion sequence. The encoded motion sequence can be expressed by the following formula:

$$m_k = [\alpha_{k1}, \alpha_{k2}, \dots, \alpha_{kn}]$$

where m_k represents an individual in the population (*i.e.*, a motion sequence) and $\alpha_{ki}(i = 1, \dots, n)$ is a coded location of an individual (a frame in the movement sequence). We take all the binary code 1 which is in the encoded motion sequence as the keyframe. Namely, when α_{kn} takes 0, this means that the frame is not the keyframe; when it takes 1, the frame is the keyframe. In addition, since the key-frames are generally not adjacent, if α_{ki} and $\alpha_{k(i+1)}$ are both 1 (namely, they are adjacent), only one of them is taken as the keyframe. Furthermore, if $\alpha_{k(i-1)}$, α_{ki} and $\alpha_{k(i+1)}$ all are 1, $\alpha_{k(i-1)}$ and $\alpha_{k(i+1)}$ are taken as the keyframes. In this way, we make sure that the selected keyframes are not adjacent in the corresponding encoded motion sequences.

In this study, the population is initialized randomly. The population size is 30, the number of populations is 30, and the number of generations is 150. The crossover probability in each population is generated randomly within the range of 0.7–0.9 and the mutation probability in each population is generated randomly within the range of 0.001–0.05. The first and last frames in the motion sequence are the default key frames.

3.3. Calculation of the Motion Sequence Reconstruction Error

To simplify the experimental process within a certain error, the linear interpolation method is employed to interpolate the keyframe and to obtain the motion sequence reconstruction. The Euler angle is used to interpolate the rotation. The average inter-frame distance between the original motion sequence and the motion sequence reconstruction is selected as the reconstruction error [10]:

$$E = [\sum (F(n) - F'(n))^2] / N$$

where $F(n)$ are the original motion data, $F'(n)$ are the corresponding reconstructed motion data, and N is the total number of frames in the motion sequence.

3.4. Fitness Function

The evolution of the genetic algorithm depends on the fitness function, so it is particularly important to define the fitness function. During the motion sequence keyframe extraction process, there are two target values: the minimum reconstruction error and the optimal number of keyframes, *i.e.*, reconstruction error minimization and compression ratio optimization are the goals.

The reconstruction error and the number of keyframes comprises a pair of contradictions, thus a smaller reconstruction error indicates a larger number of keyframes. The fitness function is defined according to the following formula:

$$f(\text{keyArray}) = wE(\text{keyArray}) + (1 - w)R(\text{keyArray}),$$

where w is the weight, which can be varied according to different situations, and keyArray represents the current keyframe sequence.

$$R(\text{keyArray}) = \text{length}(\text{keyArray}) / \text{frame}_{\text{total}}$$

The formula above is the compression ratio, *i.e.*, the ratio of the number of keyframes relative to the number of frames in the original motion sequence, where $length(keyArray)$ represents the number of key frames and $frame_{total}$ is the number of frames in the original motion sequence.

Because the reconstruction error and the compression ratio have different dimensions, it is necessary to normalize the reconstruction error. The standardization process for reconstruction can be expressed as:

$$E(keyArray) = E_{keyArray} / E_{max}$$

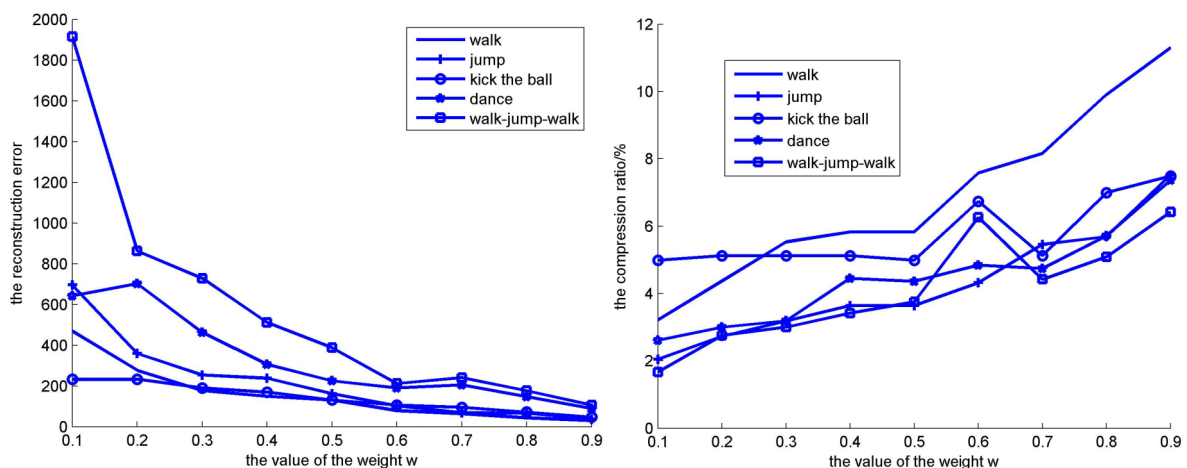
where $E_{keyArray}$ is the reconstruction error when the keyframe is $keyArray$ and E_{max} is the reconstruction error when the keyframe simply takes the first frame and the last frame.

4. Experimental Analyses

Experiments were performed to verify the effectiveness of the algorithm. The data were derived from the CMU database [16] and the sampling frequency for the human motion capture data was 120 frames/s. In the experiments, the population size was 30, the number of populations was 30, and *MAXGEN* was 150.

Experiment 1. Comparison with different values of the weight w . We selected five representative motions: walk, jump, kick the ball, dance, and walk–jump–walk. The value of the weight w was increased at intervals of 0.1 in the range of 0–1. The results are shown in Figure 3.

Figure 3. Reconstruction error and compression ratio with different values of the weight w .



As shown in Figure 3, when the value of the weight w changed, we obtained the corresponding compression ratio and the reconstruction error for the same motion. As the value of the weight w increased, the corresponding reconstruction error decreased and the corresponding compression ratio increased. This is because a smaller reconstruction error indicates higher compression. When the reconstruction error was less than 200 (the value of the weight w was higher than 0.5), the reconstructed sequence was usable. When the reconstruction error was higher, there was a significant difference between the reconstructed sequence and the original motion sequence. Thus, the value of the weight w was selected as 0.5.

Experiment 2. Comparison of motion capture sequence keyframe extraction from data with different physical characteristics. We selected four types of waking motion data with the same logic (the number of frames was the same, *i.e.*, 129). However, their physical characteristics were different (different speeds and different styles). With the same number of keyframes (seven), the extraction results were as shown in Figure 4.

Figure 4. (a) Large swing of the arms; (b) Small swing of the arms; (c) Happy walk; (d) Wild walk.

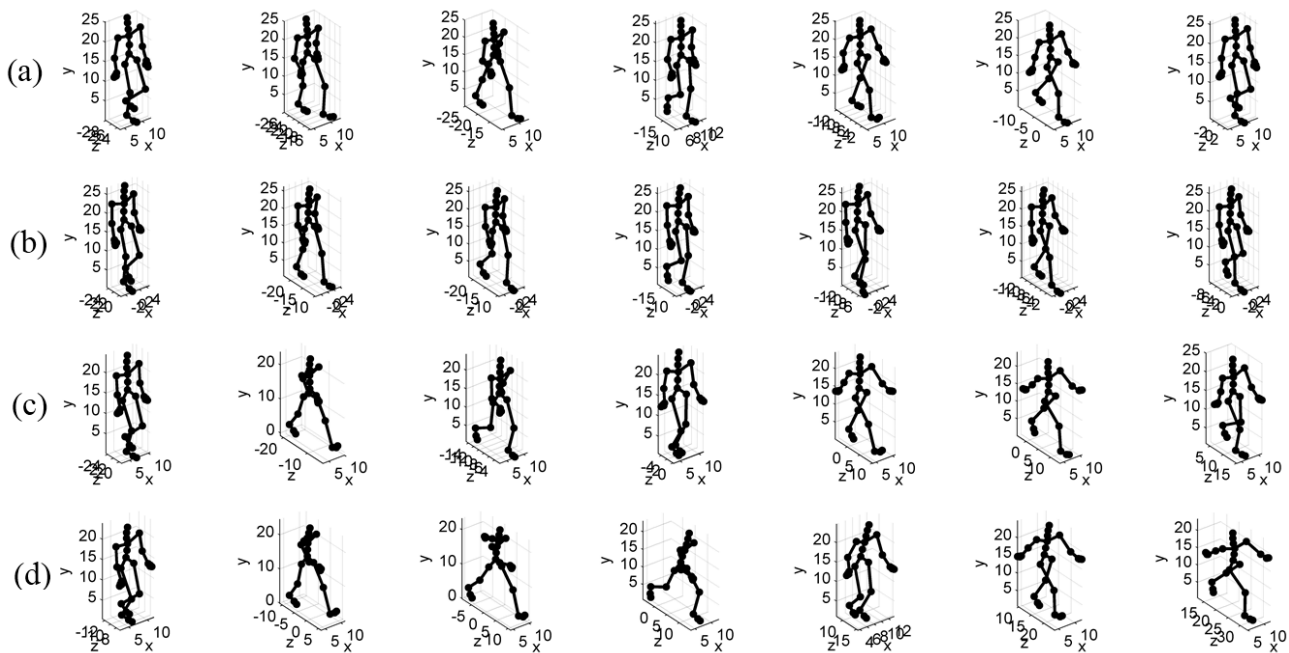


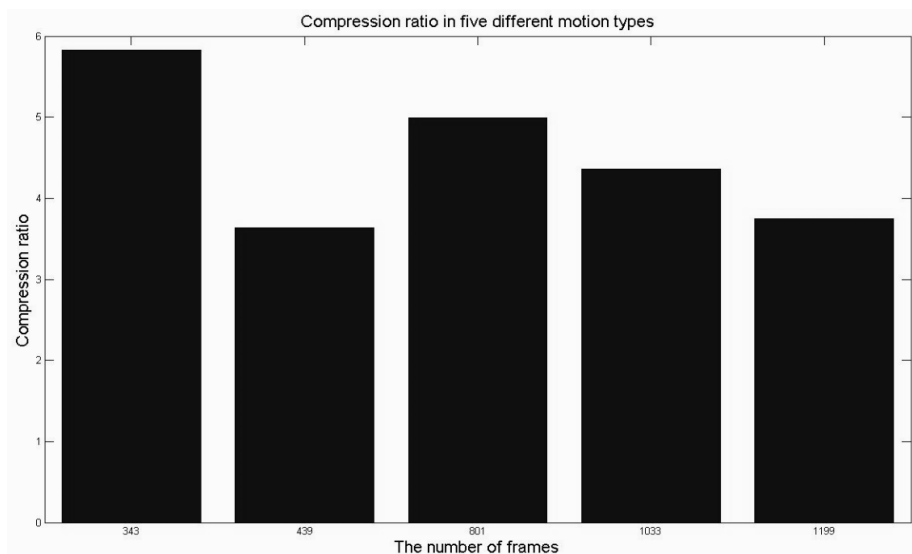
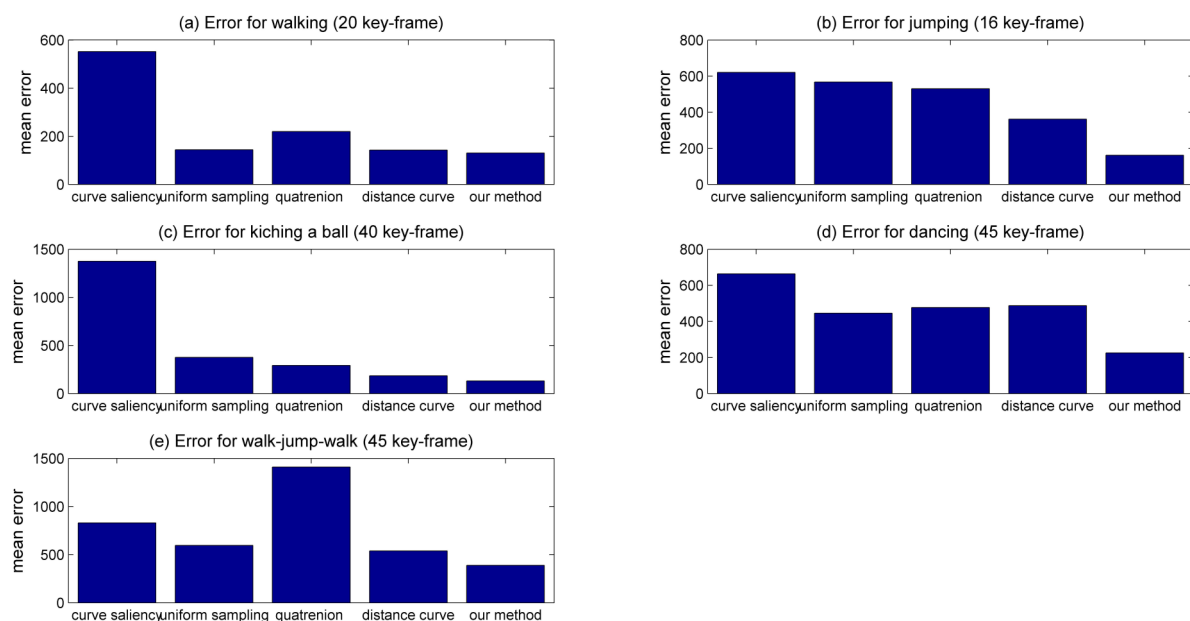
Figure 4 shows that by using MPGA to extract the keyframes from the four types of motion sequences with the same logic but different physical characteristics, the consistency of the keyframe sets was maintained and similar motion sequences were well distinguished.

Experiment 3. Comparison of the compression ratio and reconstruction error. We tested five different types of motion data, *i.e.*, walking, jumping, kicking a ball, dancing, and walk–jump–walk, where we use the proposed method to extract the keyframes from the different motion sequences. According to the number of keyframes determined by our method, we then used four other methods to extract the keyframes from the motion capture data.

Table 1 and Figure 5 show that the compression ratio was generally around 5% after keyframe extraction based on MPGA was used to extract the keyframes from different motion types. Figure 6a–f shows that the reconstruction error was minimized when the keyframes were extracted using our method, and the error value was significantly better than that with the other four methods (curve saliency, uniform sampling, quaternion, and distance curve). In summary, the proposed keyframe extraction method minimized the error but it also achieved a good compression ratio, thereby demonstrating that it was the best keyframe extraction algorithm. This is because we could obtain the optimum keyframe from the motion capture data using our method.

Table 1. Comparison of the compression ratio and reconstruction error.

Motion type	Total frames	Number of keyframes	Compression ratio (%)	Reconstruction error				
				Curve saliency [17]	Uniform sampling	Quaternion	Distance curve [10]	Our method
Walking	343	20	5.83	551.5709	144.0324	219.8285	142.8489	130.5158
Jumping	439	16	3.64	620.7511	567.3376	530.1455	361.0555	161.7257
Kicking a ball	801	40	4.99	1374.7	375.5223	291.2531	184.7565	131.1673
Dancing	1033	45	4.36	663.4626	445.0226	476.2979	487.1763	225.1592
Walk–jump–walk	1199	45	3.75	828.8955	595.2192	1410.6	538.4884	388.2128

Figure 5. Compression ratio with different motion types using our method.**Figure 6.** Comparison of the reconstruction error with five methods for five sampling motions: (a) error for walking; (b) error for jumping; (c) error for kicking a ball; (d) error for dancing; and (e) error for walk–jump–walk.

Experiment 4. Comparison of using the MPGA and a genetic algorithm to extract the keyframes. In this experiment, the value of w was 0.5, the crossover ratio value was 0.95, and the mutation rate value was 0.05. Two different motions were tested in this experiment: walking (343 frames) and jumping (439 frames).

Table 2 shows that for the same target optimization fitness function, the keyframe extraction algorithm based on the MPGA obtained a better compression ratio than the method based on a genetic algorithm, while it also achieved a satisfactory reconstruction error. Because the reconstruction error and the compression ratio is a pair of contradictions, a smaller reconstruction error indicates a higher compression ratio. The keyframes extracted by the two algorithms could reconstruct the original data well. However, in terms of the transmission, storage, and reuse of motion capture data, our method is more suitable for keyframe extraction.

Table 2. Comparison of the results for Experiment 4.

	Multiple population Genetic algorithm		Genetic algorithm	
	Walking	Jumping	Walking	Jumping
Motion type	Walking	Jumping	Walking	Jumping
Total frames	343	439	343	439
Number of keyframes	55	72	97	135
Compression ratio	16.03%	16.40%	28.28%	30.75%
Reconstruction error	54.389	36.110	21.8254	21.8020

Experiment 5. We also compared the performance of our proposed method and other methods using an irregular exercise, *i.e.*, kicking a ball (801 frames), where the number of keyframes was 40, and the results are shown in Figure 7.

Figure 7. Results obtained using: (a) our method; (b) curve saliency; (c) uniform sampling; and (d) distance curve.

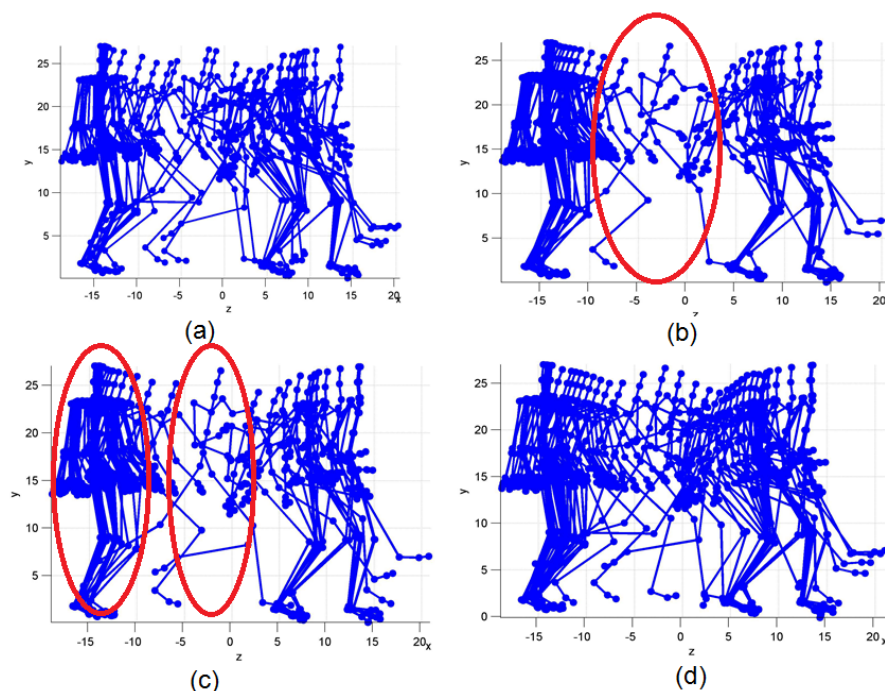


Figure 7 shows that the keyframes extracted by the MPGA and the method based on the distance curve could obtain good representations of the original motion sequences. However, the methods based on uniform sampling and curve saliency had oversampling and undersampling issues (red circles shown in Figure 7b–c), thus they did not obtain good representations of the original motion sequences. These results demonstrate that keyframes can be extracted from irregular exercises using our method. With the same compression ratio, our method obtained the best results and it exhibited a greater capacity for representing the original motion data than the other methods.

5 Conclusions

In this study, we proposed a method for keyframe extraction from human motion capture data based on MPGA. Our algorithm aims to achieve the minimum reconstruction error and the optimum number of keyframes according to a defined fitness function based on a number of initial populations that undergo co-evolution, where the genetic algorithm considers global and local search. The evolutionary process does not require the artificial designation of threshold parameters. In Experiment 1, the keyframes extracted by our method distinguished similar motion sequences satisfactorily. Therefore, our method can also be used for classifying long motion sequences and for new motion synthesis. This method can extract the keyframes with a better compression ratio and it can reconstruct the original motion sequence well. However, the algorithm still has some shortcomings because it is not possible to specify the number of keyframes. Moreover, the MPGA is a global search algorithm that aims to determine the optimal value, thus the algorithm is highly time-consuming.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61370141, No. 61300015), by Program for Changjiang Scholars and Innovative Research Team in University (No. IRT1109), the Program for Liaoning Excellent Talents in University (No. LR201003, No. LJQ2013131), and the Program for Science and Technology Research in New Jinzhou District (No. 2013-GX1-015).

Author Contributions

S.Z. wrote this manuscript; Q.Z. and D.Z. contributed to the writing, direction and content and also revised the manuscript.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Liu, F.; Zhuang, Y.T.; Wu, F.; Pan, Y. Motion retrieval with motion index tree. *Comput. Vis. Image Underst.* **2003**, *92*, 265–284.
2. Park, M.J.; Shin, S.Y. Example-based motion cloning. *Comput. Animat. Virtual Worlds* **2004**, *15*, 245–257.

3. Zhu, D.M.; Wang, Z. Extraction of keyframe from motion capture data based on motion sequence segmentation. *J. Comput.-Aided Des. Comput. Graph.* **2008**, *20*, 787–792. (In Chinese)
4. Halit, C.; Capin, T. Multiscale motion saliency for keyframe extraction from motion capture sequences. *Comput. Animat. Virtual Worlds* **2011**, *22*, 3–14.
5. Lim, I.S.; Thalmann, D. Key-posture extraction out of human motion data. In Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society Istanbul, Turkey, 25–28 October 2001; pp. 1167–1169.
6. Togawa, H.; Okuda, M. Position-based keyframe selection for human motion animation. In Proceedings of 11th International Conference on Parallel and Distributed Systems, Fukuoka, Japan, 2005; pp. 182–185.
7. Assa, J.; Caspi, Y.; Cohen, D. Action synopsis: Pose selection and illustration. *ACM Trans. Graph.* **2005**, *24*, 667–676.
8. Yang, T.; Xiao, J.; Wu, F.; Zhuang, Y.T. Extraction of keyframe of motion capture data based on layered curve simplification. *J. Comput.-Aided Des. Comput. Graph.* **2006**, *18*, 1692–1697. (In Chinese)
9. Peng, S. Key frame extraction using central distance feature for human motion data. *J. Syst. Simul.* **2012**, *3*, 1–5.
10. Zhang, Q.; Xue, X.; Zhou, D.; Wei, X. Motion Key-frames extraction based on amplitude of distance characteristic curve. *Int. J. Comput. Intell. Syst.* **2013**, *7*, 1–9.
11. Liu, Y. Keyframe extraction from motion capture data by optimal reconstruction error. *J. Comput.-Aided Des. Comput. Graph.* **2010**, *22*, 670–675. (In Chinese)
12. Lee, T.-Y.; Lin, C.-H.; Wang, Y.-S.; Chen, T.-G. Animation key-frame extraction and simplification using deformation analysis. *IEEE Trans. Circuits Syst. Video Technol.* **2008**, *18*, 478–486.
13. Liu, X.; Zhao, D.; Zhao, Y. Survey on keyframe technology of data-driven human animation. *Comput. Eng. Des.* **2011**, *32*, 1006–1009.
14. Liu, X.; Hao, A.; Zhao, D. Optimization-based keyframe extraction for motion captures animation. *Vis. Comput.* **2013**, *29*, 85–95.
15. Cai, M.; Zou, B.; Xin, G. Extractions of keyframe from motion capture data based on pre-selection and reconstruction error optimization. *J. Comput.-Aided Des. Comput. Graph.* **2012**, *24*, 1485–1492. (In Chinese)
16. CMU Graphics Lab Motion Capture Database. Available online: <http://mocap.cs.cmu.edu/> (accessed on 7 July 2014).
17. Bulut, E.; Capin, T. Keyframe extraction from motion capture data by curve saliency. Available online: <http://www.cs.rpi.edu/~bulute/casa.pdf> (accessed on 7 November 2014).