Technical Note

Motion Key-Frame Extraction by Using Optimized t-Stochastic Neighbor Embedding

Qiang Zhang *, Yi Yao, Dongsheng Zhou and Rui Liu

Key Laboratory of Advanced Design and Intelligent Computing, Dalian University, Ministry of Education, Dalian 116622, China; E-Mails: yaoyi28@126.com (Y.Y.); zhoudongsheng@dlu.edu.cn (D.Z.); liurui@dlu.edu.cn (R.L.)

* Author to whom correspondence should be addressed; E-Mail: zhangq@dlu.edu.cn; Tel./Fax: +86-411-87403733.

Academic Editor: Yuka Sasaki

Received: 11 May 2014 / Accepted: 16 April 2015 / Published: 21 April 2015

Abstract: Key-frame extracting technology has been widely used in the field of human motion synthesis. Efficient and accurate key frames extraction methods can improve the accuracy of motion synthesis. In this paper, we use an optimized t-Stochastic Neighbor Embedding (t-SNE for short) algorithm to reduce the data and on this basis extract the key frames. The experimental results show that the validity of this method is better than the existing methods under the same experimental data.

Keywords: motion synthesis; motion graph; dimension reduction; extract key-frame

1. Introduction

Intelligent algorithms for synthesizing human motion capture data are useful in many fields such as entertainment, education, training simulators and so on. At the same time, more and more human motion data is becoming freely available for public, such as CMU database [1]. The opening data enable researchers to synthesize human motions with some simple applications. In order to fully use these existing data, many methods of human motion synthesis have been proposed. Motion graph is a typical method to synthesize human motion data. Recently, extracting keyframe is almost the fundamental to reprocess those existing motion data. Figure 1 is an example of key-frame.

Figure 1. An example of key-frame, the numbers are the position of the key-frame.

For example, motion graph was emerged to automatic synthesis human motion data as a very promising technique [2–7]. A motion graph is built with many motion capture clips and added transitions motion between similar frames in these clips. Once constructed, a path through the graph represents a multi-behavior motion. In order to construct the motion graph, most researchers extract key-frames and calculate the comparability between them. Key-frame extraction is the fundamental for motion graph construction. Standard methods calculate similarity between all motion frames. Theoretically, it will be more time consuming and poorer efficiency following the growth of the motion frames. So, some researchers aimed to present more efficient solutions. For example, Krüger *et al.* [8] introduced a method which could fast search the local and global similarity in large motion capture databases. He also used a spatial data structure (KD-tree) to only search for k similar poses in a knn search. On the other hand, if people calculate the inter-frame similarity between all of motion sequences, the frames which satisfied the threshold may not be in keeping with the physical correctness. For example, one synthesizes a motion from “going” to “running”, the intermediate state of two typical movements is similar and the transitional frames which satisfied the threshold may appear in the time when the feet are off the ground. Under this circumstance, the transitional motion appears in the substandard physical laws position so that the result of motion syntheses is serious distortion. To prevent this problem, we need to extract key-frame and calculate similarity based upon those motions.

In this paper, we use an optimized manifold learning method based on t-SNE first presented in [9] to effectively extract key-frames. This method has the following advantages. Firstly, we use optimized parameters for human motion data based on standard t-SNE algorithm. Secondly, key-frames are selected according to the low-dimensional motion posture characteristic curves. The curves are mapped by high-dimensional motion data. In this way, it can not only ensure the reliable and accurate calculations but also improve the building efficiency of motion graph.

The experimental results show that the optimized t-SNE nonlinear manifold learning is faster and more accurate when extracting key-frame than the previous methods based on PCA or ISOMAP [10]. In order to prove the validity of this algorithm, we analyzed and contrasted the results by comparative experiments.

2. Related Works

Key-frame extracting technology has been widely used in many aspects of human motion synthesis. Many extracting algorithms have been proposed in recent years. Of course these algorithms have their strengths and weaknesses. The first way is fast similarity search which is based on clustering to extract human motion features. Liu *et al.* [11] clustered the motion data of N frame to K collection and took

first frame of each collection as a key frame. Park [12] took motion data with parameters and represented it by the quaternion. He linearized and clustered motion data with principal component analysis and K-mean method. This method could classify similar samples and the extracted key frames could be better represented original sample. However, it rarely considered the time correlation between the samples, which could lead to distortions of the analysis of motion sequences. The second way is a key-frame extracting method based on Low dimensional embedding algorithms. The methods can analyze high dimensional human motion data after dimensionality reduction. There are many nonlinear manifold learning methods such as ISOMAP, ST-ISOMAP, SOM, LLE, *etc.* [10,12,13], which mainly could be divided into two categories: global method and local method. There are many papers used the manifold learning method. Xiao *et al.* mapped the high-dimensional motion data into low-dimensional manifold by ISOMAP dimensionality reduction algorithm for more precise extracted key-frame and split human motion data fragments [14]. Liu broke up motion section with ISOMAP [11]. She extracted the boundary key-frame to distinguish between different types of sections in original motion data. Seward *et al.* used the ST-ISOMAP method to get new motion data fragments by rearranging motion data with projecting human motion data in low-dimensional manifold [15]. Lee *et al.* got the distribution of high-dimensional data in low-dimensional manifold with SOM method [16]. Lucas Kovar *et al.* [17] built a visualized parameterized space of human motions by using an automatic method which could extract the similar key-frames from the motion data set. The third way is motion segmentation. There are several representative researchers. For examples, Barbic *et al.* [18] presented an effective and easy to implement method which could segment motion capture data into distinct behaviors. Zhou *et al.* [19] provided a framework named hierarchical aligned cluster analysis (HACA), which is an unsupervised hierarchical bottom-up framework and could effectively segment complex human motions. Krüger *et al.* [20] presented a fully automatic method that could temporally segment the human motion sequences and similar time series, in which the self-similar structures of the human motion sequences were used. Also, there are still some distinguished methods, for examples, Müller mapped a series of segmentation methods, including content-based method [21] and motion templates-based method [22,23]. They are all efficient and automatic.

Stochastic Neighbor Embedding (SNE) was presented by Roweis [24]. It can convert the high-dimensional space into low-dimensional space. Maaten and Hinton [9] described a way of converting a high-dimensional data points set into a matrix of pairwise similarities which was called t-SNE. They proved that t-SNE is capable of capturing much of the local structure of the high-dimensional data very well, while also revealing global structure such as the presence of clusters at several scales [7,25].

Previous studies show that the method based on global nonlinear manifold learning can be applied to extract key frames of human motion data. Moreover, the low-dimensional data set can represent the motion sequence of high-dimensional posture well and explore the most essential characteristics of motion data effectively. Based on the above, this paper uses optimized t-SNE global nonlinear manifold dimensionality reduction algorithm to analyze human motion data of high-dimensional space and then key-frames can be extracted.

3. Algorithm Description

Stochastic Neighbor Embedding (SNE) algorithm describes the similarity of data points p_i to data points p_j with the conditional probability p_{ij} [24]:

$$p_{ij} = \frac{\exp(-d_{ij}^2)}{\sum_{k \neq i} \exp(-d_{ik}^2)} \quad (1)$$

where d_{ij}^2 can be regarded as a squared Euclidean distance between two high-dimensional points p_i, p_j :

$$d_{ij}^2 = \frac{\|x_i - x_j\|^2}{2\sigma_i^2} \quad (2)$$

where σ_i is the variance. The details of the SNE can be referred to as [24].

Although SNE constructs reasonably and has good visualizations, the cost functions are difficult to be optimized. For this reason Maaten and Hinton presented a new method called “t-Distributed Stochastic Neighbor embedding” (t-SNE for short) that tries to improve this algorithm [7,26,27].

The standard t-SNE algorithm used an $N \times D$ matrix as data set. The rows correspond to the instances and columns correspond to the D dimensions. In the case that the data set are specified, the code plots the intermediate solution every ten times iterations. The data set is only used in the visualization of the intermediate solutions: t-SNE is an unsupervised dimensionality reduction technique. The dimensionality of the visualization constructed by t-SNE can be specified through N dims. The standard t-SNE algorithm set 2 as the value [28].

In this paper, we used optimized t-SNE algorithm to extract key-frame. The technique is a variation of Stochastic Neighbor Embedding that is much easier to optimize, and produces significantly better visualizations by reducing the tendency to crowd points together in the center of the map. Firstly, we projected high dimensional human motion data to lower one by using optimized t-SNE algorithm. The data sets we used are $N \times D$ matrix. The N rows corresponded to the frame of human motion data and columns corresponded to the D dimensions (96 were set as default value of D). We also set the value of perplexity of the Gaussian distributions, which employed in the high-dimensional space as 30 to fit for human motion data. In this way, we can get an $N \times 3$ matrix that specified the coordinates of the three low-dimensional data points. Secondly, the characteristic curve could be drawn with the low-dimensional data set, which was a reflection of the high-dimensional motion data. Three dimensional low-dimensional embedding reflects the transform direction of human motion in three-dimensional. In order to extract the key-frames, the shifting trends of human motion were needed. Therefore, we get the characteristic curve through interpolation with the low-dimensional data, and the result of interpolation reflects the shifting trends of human motion. The inflection points with the x -axis of characteristic curve can represent the key-frame point of high-dimensional space human motion posture. The criteria of select those points is just based on this algorithm. Thirdly, the key-frame of the motion segment could be set with the characteristic curve. We got the key-frame at the inflection points of characteristic curve and the point of intersection of the x -axis and characteristic curve.

4. Experiments and Results Analysis

In this paper, we used CMU Graphics Lab Motion Capture Database [29] to test our algorithm. The format of the motion capture data used in the experimental section is BVH. We selected three clips including walking, running and jumping. Among these data, walking and running were regular motion, while jumping was chaotic motion. By this way, we could prove the versatility of our algorithm. The format of motion data was “BVH” (developed by Biovision) with 120 frame/second sampling rate and comprised with 31 key points in 96-dimensional data composition. The joint root is represented by a 3D translation and rotation vector. The translation of joint root determines the current position of the skeleton while the rotation of joint root determines the overall orientation of the skeleton. For the other joints, they are represented by a 3D rotation vector which describes their orientation in the local coordinates of their parent joints, and their position can be calculated by the orientation vector and the position of their parents. The data used in this paper contains the 3D translation vector of the root joint and the 3D rotation vectors with all of 31 key joints. The software environment was the Matlab 7.12 platform upon Windows 7 Ultimate. The hardware experimental environment was Intel Core i3-2100 3.10 GHz CPU, 2.00 GB RAM.

4.1. The Comparison with Different of Nonlinear Global Manifold Learning

In order to verify the effectiveness of the optimized t-SNE method for our algorithm, linear global dimension reduction method, *i.e.*, PCA (Principal Component Analysis) and nonlinear local dimensionality reduction method ISOMAP are compared. We used three representative and different data sets of motion sequences. Three data sets are walking, running and jumping. In terms of subject matter, the figures fall under five categories, and they are characteristic curves (including Figures 2–10), manifold (including Figures 11–19), key-frames (including Figures 20–23), comparative data 1 (Figure 24) and comparative data 2 (Figure 25).

Figure 2. Low-dimensional characteristic curves of walking with optimized t-Stochastic Neighbor Embedding (t-SNE) dimension reduction, red dots are the candidates of the key-frame.

Figure 3. Low-dimensional characteristic curves of walking by using PCA algorithms.

Figure 4. Low-dimensional characteristic curves of walking by using ISOMAP+ algorithm.

Figure 5. Low-dimensional characteristic curves of running with optimized t-SNE dimension reduction, red dots are the candidates of the key-frame.

Figure 6. Low-dimensional characteristic curves of running by using PCA algorithms.

Figure 7. Low-dimensional characteristic curves of running by using ISOMAP+ algorithm.

Figure 8. Low-dimensional characteristic curves of jumping with optimized t-SNE dimension reduction.

Figure 9. Low-dimensional characteristic curves of jumping by using PCA algorithms.

Figure 10. Low-dimensional characteristic curves of jumping.

Figure 11. Low-dimensional manifolds of walking with optimized t-SNE.

Figure 12. Manifold of walking with PCA dimension reduction.

Figure 13. Manifold of walking with ISOMAP dimension reduction.

Figure 14. Low-dimensional manifold of running with optimized t-SNE.

Figure 15. Manifold of running with PCA dimension reduction.

Figure 16. Manifold of running with ISOMAP dimension reduction.

Figure 17. Low-dimensional manifold of jumping with optimized t-SNE.

Figure 18. Manifold of jumping with PCA dimension reduction.

Figure 19. Manifold of jumping with ISOMAP dimension reduction.

Figure 20. Extracted key-frames of walking based on t-SNE, PCA and ISOMAP respectively.

Figure 21. Extracted key-frames of running based on t-SNE, PCA and ISOMAP respectively.

Figure 22. Extracted key-frames of jumping based on t-SNE, PCA and ISOMAP respectively.

Figure 23. Extracted key-frames of a long motion sequence which has 4592 frames based on t-SNE.

Figure 24. Compression ratios in different motion types.

Figure 25. Comparison of mean absolute error of five sampling motion. (a) Error of the kick ball motion; (b) Error of the jump motion; (c) Error of the walk motion; (d) Error of the dance motion; (e) Error of the walk-jump-walk motion.

In the first group experiment, we used optimized t-SNE algorithm. As shown in Figure 2, low-dimensional characteristic curves of walking by optimized t-SNE clearly represent the motion

characteristics of high-dimensional space. The low-dimensional feature curve was changed following with the changing of freedom of body movement. At the inflection points of the characteristic curve (Red dot) was the changing key-frame of human motion posture. Figure 11 shows the low-dimensional manifold of walking. One can see that the low-dimensional manifold with optimized t-SNE was smoother than Figures 12 and 13, which are drawing based on PCA and ISOMAP algorithm, respectively. That is to say, the PCA and ISOMAP dimensionality reduction characteristic curve could not clearly represent the characteristics of high-dimensional space of human motion. The extracted key-frames got by t-SNE, PCA and ISOMAP were shown in Figure 20 respectively (in order to make the result more clearly, the motion sequence is stretched according to a fixed ratio). From the comparison, it is easily to find that the key-frames extracted by t-SNE accurately represented human motion feature. Consequently, the experiments show that our method, the optimized t-SNE to reduce the dimension of the original data, is valid.

Similar tests were completed base on the data sets of running and jumping. Similar results could be found in the second group and the third group of experiments. Figures 9–22, similar conclusions as the first group experiments could be obtained.

Furthermore, the proposed approach is applied on a longer motion sequence which contains 4592 frames and the label of the motion sequence in the CMU motion capture data base is 13-29. In the experiment, 188 key-frames were extracted from the motion sequence, due to the limitation of the space, only the first 100 key-frames were exhibited in Figure 23. So, as the whole experiments showed, we could obtain accurate key-frames with optimized t-SNE algorithm for different kind of motion data, which ranged from simple regular motion data to complex irregular motion sequences.

4.2. Comparison of the Different Key Frames Extraction Algorithm

In this section, we apply our approach and other methods (such as uniform sampling, curve saliency and the quaternion distance) to test the different kinds of motion sequences.

From Table 1 and Figure 24, we found that the range of compression ratio varies around 5% in our method.

Table 1. Comparison of compression ratio in five different motion types.

Category	Kick ball	jump	walk	dance	Walk-jump-walk
Total number	801	439	343	1033	1199
key frames	35	24	16	37	50
Compression ratio (%)	4.3	5.5	4.6	3.5	4.17

Then, we compute the mean absolute error with the following formula:

$$E = [\sum (F(n) - F'(n))^2] / N \quad (3)$$

Here, $F(n)$ is the original motion data, $F'(n)$ is the reconstructed motion data by the linear interpolation algorithm according to the key frames. N is the number of frames multiplied by 96.

We apply four kinds of methods to get the same number of key frames from the same motion type. Then we reconstruct the motion sequence through the linear interpolation. As shown in Table 2 and Figure 25, we can find that mean absolute errors with our approach are almost less than the other

methods under the same compression ratio. In short, we know that our approach works well through testing a large amount of data.

Table 2. Comparison of the reconstructed mean absolute error in different methods.

Category	Kick ball (35)	Jump (24)	Walk (16)	Dance (37)	Walk-jump-walk (50)
Our approach	4.9609	2.3795	3.1628	5.8849	4.2502
quaternion	3.9236	1.9799	6.6882	9.4019	10.6691
curve saliency	9.7743	2.4829	5.2014	7.1775	7.1214
uniform sampling	14.5063	2.5363	2.9479	7.9382	4.7710

5. Conclusions

In this paper, we use optimized t-SNE algorithm as motion data processing and then extract the key frames. By performing two widely used experimental comparison algorithms, we verified the validity of this algorithm. The contribution of this paper was mainly as follows: an optimized t-SNE algorithm was applied to reduce dimension of human motion data and the motion data processed by this method were advantageous for extracting key-frame. The method used the characteristic curve of low-dimensional human motion data, and could dynamically extract the key-frames. Moreover, it was not only highly effective to simple regular motion data, but also equally effective for complex irregular motion sequences.

There are still many problems remaining, such as missing frames slightly when processing some complex chaotic motion data. So in the future works, we need to make the algorithm more intelligent and efficient by optimizing those issues, such as using other dimensionality reduction algorithm or optimizing the extraction method.

Acknowledgements

This work is supported by National Natural Science Foundation of China (Nos. 61370141, 61300015), Natural Science Foundation of Liaoning Province (No. 2013020007), the Program for New Jinzhou District Science and Technology Research (No. 2013-GX1-015, KJCX-ZTPY-2014-0012).

Author Contributions

Yi Yao drafts this manuscript; Qiang Zhang, Dongsheng Zhou and Rui Liu contributed to the direction, content, re-writing and also revised the manuscript.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Zhao, L.; Safonova, A. Achieving good connectivity in motion graphs. In Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Dublin, Ireland, 7–9 July 2008; pp. 127–136.

2. Kovar, L.; Gleicher, M.; Pighin, F. Motion graphs. *ACM Trans. Graph.* **2002**, *21*, 473–482.
3. Lee, J.; Chai, J.; Reitsma, P.S.A.; Hodgins, J.K.; Pollard, N.S. Interactive control of avatars animated with human motion data. *ACM Trans. Graph.* **2002**, *21*, 491–500.
4. Pullen, K.; Bregler, C. Motion capture assisted animation: Texturing and synthesis. *ACM Trans. Graph.* **2002**, *21*, 501–508.
5. Rahim, R.A.; Suaib, N.M.; Bade, A. Motion Graph for Character Animation: Design Considerations. *Int. Conf. Comput. Technol. Dev.* **2009**, *2*, 435–439.
6. Sakamoto, Y.; Kuriyama, S.; Kaneko, T. Motion map: Image-based retrieval and segmentation of motion data. In Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation, Grenoble, France, 27–29 August 2004; pp. 259–266.
7. Van der Maaten, L. Learning a Parametric Embedding by Preserving Local Structure. In Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics (AI-STATS), Clearwater Beach, FL, USA, 16–18 April 2009; Volume 5, pp. 384–391.
8. Krüger, B.; Tautges, J.; Weber, A.; Zinke, A. Fast local and global similarity searches in large motion capture databases. In Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Madrid, Spain, 2–4 July 2010; pp. 1–10.
9. Van der Maaten, L.; Hinton, G. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
10. Tenenbaum, J.B.; Silva, V.D.; Langford, J.C. A global geometric framework for nonlinear dimensionality reduction. *Science* **2000**, *290*, 2319–2323.
11. Liu, Y.Y.; Zhang, Q.; Wei, X.P.; Zhou, C.J. Key-frames extraction based on motion capture data. In Proceedings of Conference on Intelligent CAD and Digital Entertainment, Dalian, China, 22–24 July 2008.
12. Park, S.I.; Shin, H.J.; Shin, S.Y. On-line locomotion generation based on motion blending. In Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation, San Antonio, TX, USA, 21–22 July 2002; pp. 105–111.
13. Van der Maaten, L.J.P.; Postma, E.O.; Van den Herik, H.J. Dimensionality Reduction: A Comparative Review. *J. Mach. Learn. Res.* **2009**, *10*, 66–71.
14. Xiao, J.; Zhuang, Y.T.; Wu, F. Getting distinct movements from motion capture data. In Proceedings of the International Conference on Computer Animation and Social Agents, Geneva, Switzerland, 5–7 July 2006; pp. 33–42.
15. Seward, A.E.; Bodenheimer, B. Using nonlinear dimensionality reduction in 3D Figure animation. In Proceedings of the 43rd Annual Southeast Regional Conference, Kennesaw, GA, USA, 18–20 March 2005; Volume 2, pp. 388–392.
16. Lee, C.S.; Elgammal, A. Human motion synthesis by motion manifold learning and motion primitive segmentation. In Proceedings of 4th International Conference on Articulated Motion and Deformable Objects, Mallorca, Spain, 11–14 July 2006; pp. 464–473.
17. Kovar, L.; Michael, G. Automated extraction and parameterization of motions in large data sets. *ACM Trans. Graph.* **2004**, *23*, 559–568.
18. Barbič, J.; Safonova, A.; Pan, J.Y.; Faloutsos, C.; Hodgins, J.K.; Pollard, N.S. Segmenting motion capture data into distinct behaviors. In Proceedings of the 2004 Graphics Interface Conference, London, ON, Canada, 17–19 May 2004; pp. 185–194.

19. Zhou, F.; De la Torre, F.; Hodgins, J.K. Hierarchical aligned cluster analysis for temporal clustering of human motion. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 582–596.
20. Vögele, A.; Krüger, B.; Klein, R. Efficient Unsupervised Temporal Segmentation of Human Motion. In Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Copenhagen, Denmark, 21–23 July 2014; pp. 167–176.
21. Müller, M.; Röder, T.; Clausen, M. Efficient content-based retrieval of motion capture data. *ACM Trans. Graph.* **2005**, *24*, 677–685.
22. Müller, M.; Röder, T. Motion templates for automatic classification and retrieval of motion capture data. In Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation, Vienna, Austria, 2–4 September 2006; pp. 137–146.
23. Müller, M.; Baak, A.; Seidel, H.P. Efficient and robust annotation of motion capture data. In Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, New Orleans, LA, USA, 1–2 August 2009; pp. 17–26.
24. Hinton, G.E.; Roweis, S.T. Stochastic Neighbor Embedding. *Adv. Neural Inf. Process. Syst.* **2002**, *16*, 833–840.
25. Van der Maaten, L.; Hinton, G. Visualizing Non-Metric Similarities in Multiple Maps. *Mach. Learn.* **2012**, *87*, 33–55.
26. Van der Maaten, L.; Weinberger, K. Stochastic Triplet Embedding. In Proceedings of IEEE International Workshop on Machine Learning for Signal Processing (MLSP), Santander, Spain, 23–26 September 2012; pp. 1–6.
27. Van der Maaten, L.; Welling, M.; Saul, L.K. Hidden-Unit Conditional Random Fields. In Proceedings of the International Conference on Artificial Intelligence & Statistics (AI-STATS), Ft. Lauderdale, FL, USA, 11–13 April 2011; Volume 15; pp. 479–488.
28. Van Der Maaten, L. Fast Optimization for t-SNE. In Proceedings of Neural Information Processing Systems (NIPS) 2010 Workshop on Challenges in Data Visualization, Whistler, BC, Canada, 10–11 December 2010; p. 100.
29. CMU Graphics Lab Motion Capture Database. Available online: <http://mocap.cs.cmu.edu/> (accessed on 14 April 2015).