International Journal of Database Theory and Application
Vol.8, No.4 (2015), pp.231-240
http://dx.doi.org/10.14257/ijdta.2015.8.4.23

Low Level Segmentation of Motion Capture Data based on Hierarchical Clustering with Cosine Distance

Yang Yang*, Jinfu Chen*, Zhanzhan Liu**, Yongzhao Zhan* and Xinyu Wang*

** School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang 212013, China*
E-mail: yyoung@ujs.edu.cn

*** North Information Control Group Co., Ltd., Nanjing 210000, China*

Abstract

3D motion capture is to track and record human movements. In recent years, it has been applied into many fields, such as human computer interaction, animation, etc. Low-level segmentation of motion capture data is of significance to the various applications of 3D motion capture; however, due to the high dimensionality of motion capture data, traditional low-level segmentation methods can hardly work out a suitable segmentation for motion capture data. In order to solve this problem, a low-level temporal segmentation algorithm based on cosine distance is proposed, hierarchical clustering is explored so that similar velocity vectors are clustered together according to the cosine distance in a progressive way, the center of each cluster is updated as the vector derived with linear regression, the segment boundaries are determined as the point when the cosine distance between adjacent velocity vectors is greater than 1 (angle>90 degrees). We have conducted experiments on the motion capture database provided by Carnegie Mellon University (CMU), the experiment results show that the performance of the proposed method is optimistic.

Keywords: *Motion capture, Low level segmentation, Hierarchical clustering, Cosine distance*

1. Introduction

Much of the motion capture data must be carefully segmented into distinct motion primitives before it can be put into real usage. Automatic segmentation of motion capture data is becoming more and more important. According to [1], temporal segmentation of 3D motion capture data could be mainly divided into two categories, namely high level segmentation and low level segmentation.

High level segmentation refers to segment the motion into activities and behaviors, such as walk or jump, by annotating these behaviors, users could easily retrieve these behaviors according to the semantic meaning [2]. Recent years, a variety of methods have been proposed for high level segmentation, such as motion regularities [3], hierarchical aligned cluster analysis [4], kernelized temporal cut [5].

Low level segmentation tries to locate the start and end points of a motion, so that the motion can be divided into primitives. Theoretically, primitive refers to the motion segment which starts from and ends at a zero velocity point. Low level segmentation is of significance to applications such as animations [6] and video games [7]. In this paper, we will focus on low level segmentation.

Various kinds of features are explored for low level segmentation of motion capture data, in [8], angular velocity is used for segmentation, and the segment boundaries are determined as the zero-crossing of angular velocity, i.e., when the direction of angular velocity changes. For example, if at some point, the velocity changes from positive

ISSN: 2005-4270 IJDTA

direction to negative direction, then there must be point in-between when velocity equals to zero. Zero-crossing is a common segmentation method, however, it is sensitive to noise, which will result in severe false detection, in addition, it can't segment whole body motion which involves multiple dimensions. [9] explored energy based method, where segment boundaries are located as when the energy reaches local minimum, however, it has difficulties when the motions that begin or end slow. [10] proposed a entropy based method, the segmentation boundaries are determined according to the entropy of the difference between adjacent frames, however, it will also result in many false detected segment boundaries. In [11], the authors proposed a segmentation method based on pendulum, and the segment boundaries are located as the local extreme of pendulum, however, it can't find those points when directions change. The curvature based method is explored in [12], the curvature equals to the cross product of velocity and acceleration, the segment boundaries lie in the point when the curvature reaches its maximum, however, it can't detect segment boundaries when the directions do not change, a typical example would be a straight motion with a temporary pause in the middle, so curvature based method is rarely used alone.

Motion capture data is high dimensional, and the zero-crossing point in each dimension hardly coincide with each other, i.e., the zero-crossing point in one dimension may not be the zero-crossing point in the other dimensions, this leads to that there hardly exists any points when the velocity of whole body motion equals to 0. [9][13][15] propose to segment motions based on whole body motion velocity, when the velocity is less than a empirical threshold, it is determined as segment boundary. This method is simple and easy to implement, however, it will cause low precision and recall ratio. In [8], the author first find the zero-crossing point at each dimension, the segment boundaries are then determined as when these zero-crossing points in most of the dimensions coincide. This method is suitable for segmenting partial body motions, such as the motion of the arm, it is not applicable to whole body motion segmentation, since the segment boundaries at all dimensions could hardly coincide with each other. [16][17] tried to detect the segment boundaries by comparing the motions which are going to be segmented with template motions. This method is accurate and efficient, however, it can only be applied to specific fields when the template motions are available, such as in rehabilitation treatment, the doctor will prescribe the template motions for the patient to practice.

By observing how people segment the whole body motion into primitives, we have found a very interesting result, i.e., people tend to segment the motion when the direction changes more than 90 degrees, because that typically involves a great reduction in speed. In this paper, we proposed a low level segmentation method based on cosine distance, the overview of the proposed method is shown in Figure 1.

The proposed method first calculates the velocity vector of each frame, then the cosine distance between adjacent velocity vectors is computed, afterwards, hierarchical clustering is applied, so that similar velocity vectors are clustered together in a progressive way, the center of each cluster is updated as the vector derived with linear regression, finally, the segment boundaries are determined as the point when the cosine distance between adjacent velocity vectors is greater than 1. The experiment results on the CMU motion capture database [18] suggest that the performance of our proposed method is optimistic.

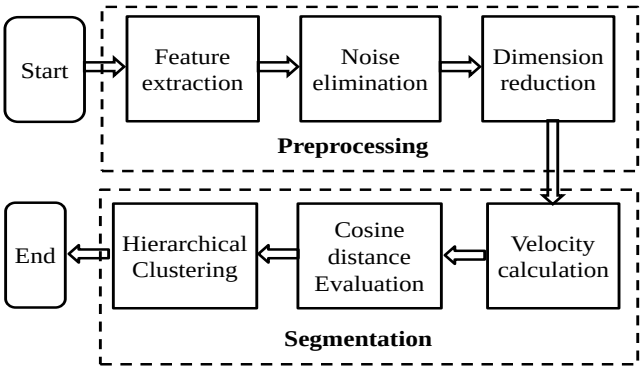

Figure 1. The Overview of the Proposed Method

The rest of this paper is organized as follows. Section 2 and Section 3 will cover the process to segment the motion capture data. In Section 4, the experiments and results are provided. Finally, in Section 5, we will conclude the paper.

2. Preprocessing of Data

2.1 Feature Extraction

Motion capture is the technology to capture and record human motions. The skeleton it captured typically contains several joints. As the skeleton shown in Figure 2, it contains 21 joints, each joint contains 3 dimensional data, this data could be the 3D position of the joint, or it could be the 3D rotation of the joint. Motion capture data could be denoted as a matrix with each row corresponds to a frame, and each column corresponds to a dimension.

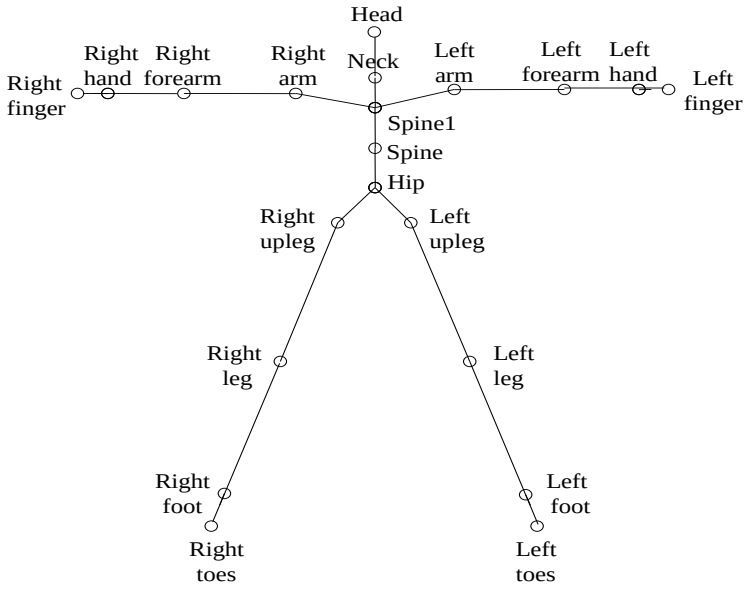

Figure 2. Captured Human Skeleton

Position of the joint is not suitable for low level segmentation of motion capture data, because it will cause severe false detection. One typical example would be a swing motion of the forearm, assume the forearm swing forward, this would be a motion primitive, however, the position of forearm achieves its local minimum while it's swing,

i.e., there is a zero-crossing point within this primitive, thereby the swing motion would be wrongly segmented into two primitives. So in this paper, we make use of the rotation of joint as the feature for segmenting motion capture data.

2.2 Noise Elimination

Due to the shift and occlusion problem, the motion capture data typically involves noise. As we discussed before, the method to segment the motion capture data is sensitive to noise. So before applying the segmentation algorithm, we should denoise the motion capture data. Here, a fifth-order two-way low pass Butterworth filter with cutoff frequency 0.1HZ is explored, as we can see in Figure 3, after filtering, the curve became much smoother. Notice that we use the two-way filter rather than the one-way filter, because otherwise it will cause phase shift which will change the segment boundaries.

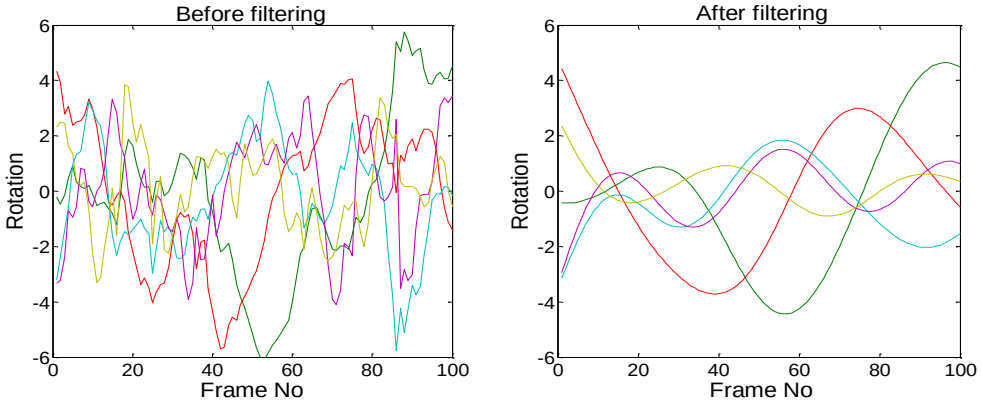

Figure 3. The Curve before and after Butterworth Filtering

2.3 Dimension Reduction

The high dimensionality of motion capture data has brought great inconvenience to the segmentation method, for example, as the skeleton shown in Figure 2, it contains 63 dimensions of data. Furthermore, the movements of some dimensions are very slight, which will have negative impact on the segmentation result. So in this paper, we apply principle component analysis to remove these slight movements. Principal component analysis is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. Those principle components in which accumulated variance reaches 80% are selected for further processing.

3. Segmentation Method

The cosine distances between adjacent velocity vectors are explored to segment the motion capture data. The cosine distance between two vectors v_i, v_{i+1} is a measure of the angle between them, as we can see in equation 3.1, it can be calculated as one minus the division of the dot product of v_i and v_{i+1} by the product of the magnitude of v_i and v_{i+1}, the range for cosine distance is from 0 to 2. v_i is the velocity vector of i^{th} frame, as shown in equation 3.2, it can be computed as the difference between two adjacent frames, namely f_{i+1} and f_i. The cosine distance between every pair of adjacent velocity vectors in a "walking" motion can be seen in Figure 4.

$$dist(v_i, v_{i+1}) = 1 - \frac{v_i \cdot v_{i+1}}{|v_i||v_{i+1}|} \quad (3.1)$$

$$v_i = f_{i+1} - f_i \quad (3.2)$$

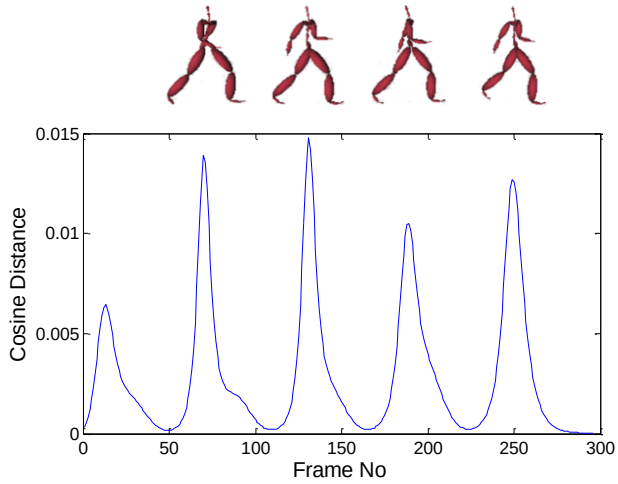

Figure 4. The Cosine Distance between every Pair of Adjacent Velocity Vectors in a "Walking" Motion

It should be noted that in Figure 4, the cosine distance between adjacent velocity vectors is quite near 0, i.e., the angles between adjacent velocity vectors are quite small, however, from our observations to the "walking" motion, at certain time points, the change of velocity direction is significant, such as the point between swing forward and swing backward during walking. The key factor for the inconsistency between calculation and observation is the high sample rate of the motion capture device. In our case, the sample rate is 60 fps, i.e., every frame lasts for 1/60=0.016 seconds, there are barely any significant changes in this tiny amount of time. Usually, in order to solve this problem and discover the segment boundaries, a cluster algorithm is required, but another problem would be raised that we don't know the number of clusters. Notice that the stopping criterion in our case is that when the cosine distance between all adjacent velocity vectors are greater than 1, so hierarchical clustering based on cosine distance tends to be the best option here, the algorithm is described below.

Algorithm *HCC*

```
Input: velocity vectors (vv); cosine distance (cd)
Output: segment boundaries (sb)
begin
   for i=1 to n              //n is the number of velocity vectors
     cluster{i}=i;           //initially, every velocity itself is a cluster
   while max(cd)>1           //when the maximum of cosine distance is greater than 1
      b=minindex(cd); e=b+1;  //minindex returns the index of minimum elements in
cd
      merge cluster{b} and cluster{e};
      merge vv[b] and vv[e] into one vector using linear regression;
      update cd[b-1] as the cosine distance between the merged vector and vv[b-1];
      update cd[e] as the cosine distance between the merged vector and vv[e+1];
      remove cd[b] from cd;
   end
   output the largest value in each cluster to sb;
end
```

As we can see from algorithm HCC, initially, every velocity itself is a cluster, then we try to find two adjacent velocity vectors with minimum distance, then the corresponding two clusters are merged into one, and these two vectors are merge into one using linear regression. Notice that, here we use regression rather than interpolation is because that regression better preserve the main direction of the two vectors. After merging two vectors, the cosine distance from the merged vector to the previous vector and the next vector should be updated, since the two vectors are merged, the cosine distance between them should be removed. This process keeps going until finally the maximum distance between adjacent vectors is greater than 1, i.e., the angles between all adjacent vectors are greater than 90 degree. The largest value in each cluster is then one of the segment boundaries. Figure 5 shows the result of hierarchical clustering of a "walking" motion.

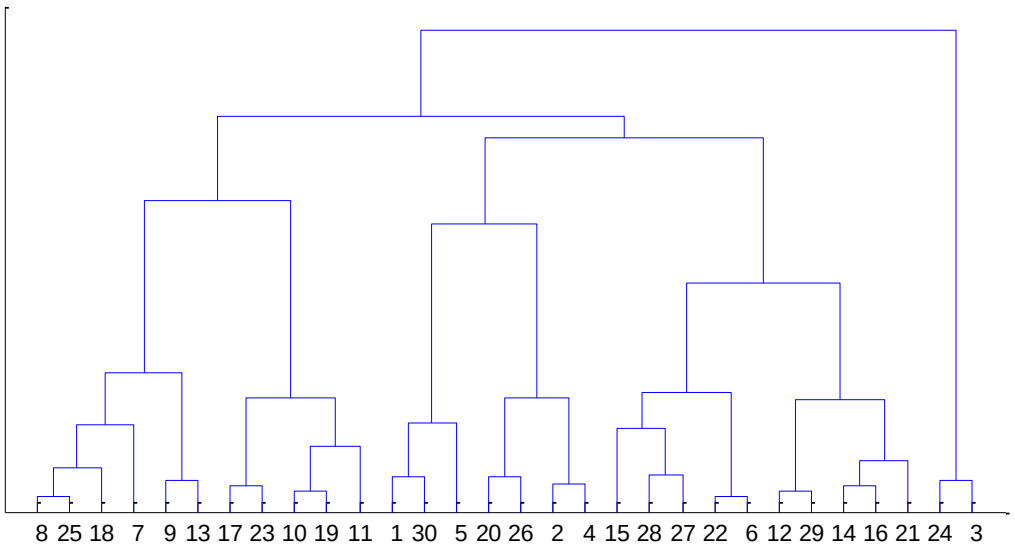

Figure 5. Hierarchical Clustering of a "Walking" Motion

4. Results and Discussion

The experiment dataset was selected from CMU motion capture database, it includes "walking", "running", "playing golf", "playing football", and "boxing", in total there are 6250 frames of data. Figure 6 shows the segmentation of the example motions in the 5 categories, each posture represents a segmentation boundary. Since the motions are too long, here, only the representative parts of them are shown. As we can see, the segment boundaries are these start and end postures of a motion.

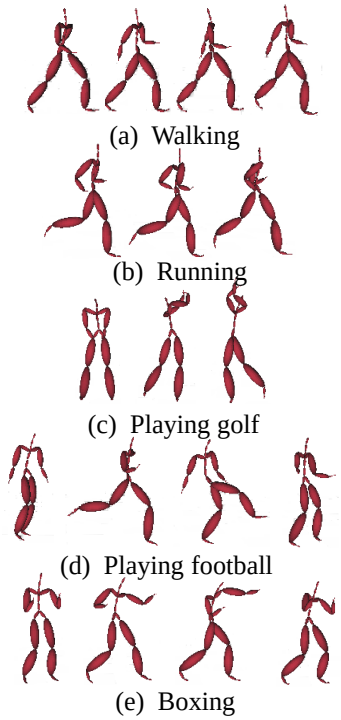

Figure 6. Segmentation of Example Motions

In order to obtain the ground truth of segmentation, we invite 20 students to manually segment these motion capture data in the dataset. A voting scheme is then applied to combine the manual segmentation results of these students so that 83 segment boundaries are obtained. The proposed method is implemented with Matlab 2010. We say a segmentation boundary derived by our method is correctly identified when there exists a manual segmentation boundary which is less than 10 frames away from it. By comparing the segmentation boundaries derived by the proposed method with the ground truth obtained manually, the precision and recall ratio of the proposed method can be derived as shown in equation 4.1 and 4.2.

$$precision = \frac{\text{number of segment boundaries which have been correctly identified}}{\text{total number of segments boundaries derived}} \quad (4.1)$$

$$recall = \frac{\text{number of segment boundaries which have been correctly identified}}{\text{total number of correct segment boundaries}} \quad (4.2)$$

Since precision and recall contradicts with each other, that means, for the same segmentation method, a higher precision ratio typically accompanied by a lower recall ratio, and vice versa. In order to provide an overall evaluation of the performance of different methods, the F1-measure is explored. F1-measure is the weighted harmonic average of precision and recall ratio, it takes into account both the precision and recall ratio. F1-measure can be calculated as shown in equation 4.3.

$$F1\text{-}measure = \frac{2 \times precision \times recall}{precision + recall} \quad (4.3)$$

The accuracies of different segmentation methods are compared as shown in Table 1. It can be seen that the proposed method outperforms both the velocity-based method and curvature based method in precision and recall ratio (84.3%>56.4%>55.4%,

89.2%>85.5%>74.7%), thus achieving a much better performance than the other two methods (86.7%>68.0%>63.6%).

Table 1. The Accuracy of Different Segmentation Methods

Methods	Precision/%	Recall/%	F1-measure/%
Velocity-based method	55.4	74.7	63.6
Curvature-based method	56.4	85.5	68.0
Our proposed method	83.1	89.2	86.0

Table 2 shows the comparison of time required by different segmentation methods in order to segment 6250 frames of data, as we can see, our proposed method requires more time than the other two methods. However, since most applications of the segmentation method are offline, so the time required by our method is still acceptable.

Table 2. The Time Required by Different Segmentation Methods

Methods	Time/s
Velocity-based method	3.68
Curvature-based method	7.43
Our proposed method	24.83

5. Conclusions

In this paper, we have discovered an interesting fact that people tend to segment the motion when the velocity direction changes more than 90 degrees because that typically involves a great reduction in speed. Based on this discovery, a low level segmentation method based on hierarchical clustering with cosine distance is proposed. The segment boundaries are determined as the point when the cosine distance between adjacent velocity vectors is greater than 1. The experiment results on the CMU motion capture database suggest that the accuracy of our proposed method is far better than the other two popular methods, and the running time required by the proposed method is acceptable.

Acknowledgements

The work described in this paper was fully supported by National Natural Science Foundation of China (61402205, 61202110), Natural Science Foundation of Jiangsu Province (BK2012284), Specialized Research Fund for the Doctoral Program of Higher Education (20113227110021), and a Research Fund granted by Jiangsu University (1281170027).

References

[1] J. Barbič, A. Safonova, J. Y. Pan, "Segmenting motion capture data into distinct behaviors", Proceedings of Graphics Interface, **(2004)**, pp. 185-194.
[2] Q. Xiao, Y. Wang, H. Wang, "Motion retrieval using weighted graph matching", Soft Computing, vol. 19, no.1, **(2015)**, pp. 133-144.
[3] R. Lan, H. Sun, "Automated human motion segmentation via motion regularities", The Visual Computer, vol. 31, no. 1, **(2015)**, pp. 33-53.
[4] F. Zhou, F. De la Torre, J. K. Hodgins, "Hierarchical aligned cluster analysis for temporal clustering of human motion", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 3, **(2013)**, pp. 582-596.
[5] D. Gong, G. Medioni, S. Zhu, "Kernelized temporal cut for online temporal segmentation and recognition", Computer Vision–ECCV, **(2012)**, pp. 229-243.

[6] W. Xing, X. Wei, J. Zhang, "Hybrid motion graph for character motion synthesis", Journal of Visual Languages & Computing, vol. 25, no. 1, **(2014)**, pp. 20-32.
[7] P. Chen, J. Li, M. Luo, "Real-Time Human Motion Capture Driven by a Wireless Sensor Network", International Journal of Computer Games Technology, vol. 2015, **(2015)**, pp. 1-14.
[8] W. Ilg, G. H. Bakir, J. Mezger, "On the representation, learning and transfer of spatio-temporal movement characteristics", International Journal of Humanoid Robotics, vol. 1, no. 4, **(2004)**, pp. 613-636.
[9] A. Fod, M. J. Matarić, O. C. Jenkins, "Automated derivation of primitives for movement classification", Autonomous robots, vol. 12, no. 1, **(2002)**, pp. 39-54.
[10] C. K. F. So, G. Baciu, "Entropy‐based motion extraction for motion capture animation", Computer Animation and Virtual Worlds, vol. 16, no. 3-4, **(2005)**, pp. 225-235.
[11] O. C. Jenkins, M. J. Mataric, "Automated derivation of behavior vocabularies for autonomous humanoid motion", International Joint Conference on Autonomous Agents and Multiagent Systems, **(2003)**, pp. 225-232.
[12] L. Zhao, N I. Badler, "Synthesis and acquisition of laban movement analysis qualitative parameters for communicative gestures", Ph.D Dissertation, University of Pennsylvania, **(2001)**.
[13] S. Gibet, P. F. Marteau, "Approximation of Curvature and Velocity for Gesture Segmentation and Synthesis", Gesture-Based Human-Computer Interaction and Simulation, **(2009)**, pp. 13-23.
[14] M. Pomplun, M. J. Mataric, "Evaluation metrics and results of human arm movement imitation", First IEEE- RAS International Conference on Humanoid Robotics, **(2000)**.
[15] K. Kahol, P. Tripathi, S. Panchanathan, "Automated gesture segmentation from dance sequences", IEEE International Conference on Automatic Face and Gesture Recognition, **(2004)**, pp. 883-888.
[16] J. F. S. Lin, D. Kulic, "Segmenting human motion for automated rehabilitation exercise analysis", Annual International Conference of the IEEE Engineering in Medicine and Biology Society, **(2012)**, pp. 2881-2884.
[17] W. Takano, Y. Nakamura, "Humanoid robot's autonomous acquisition of proto-symbols through motion segmentation. Humanoid Robots", IEEE-RAS, **(2006)**, pp. 425-431.
[18] CMU Motion Capture Database. Available: http://mocap.cs.cmu.edu/, **(2015)**.

Authors

Yang Yang, he is a ecturer at School of Computer Science and Communication Engineering, Jiangsu University. His research interest covers Computer Vision, Motion Synthesis, Pattern Recognition and E-learning.
Email: yyoung@ujs.edu.cn

Jinfu Chen, he is an associate professor at School of Computer Science and Communication Engineering, Jiangsu University. His research interest covers Software Engineering and Motion Analysis.
Email: jinfuchen@ujs.edu.cn

Zhanzhan Liu, he is a senior engineer at North Information Control Group Co., Ltd. His research interest covers Database System, Motion Analysis and Image Processing.
Email: zzliu@ustc.edu

Yongzhao Zhan, he is a professor at School of Computer Science and Communication Engineering, Jiangsu University. His research interest covers Human Computer Interaction, Distributed Computing and Video Processing.
Email: yzzhan@ujs.edu.cn

Xinyu Wang, he is an associate professor at School of Computer Science and Communication Engineering, Jiangsu University. His research interest covers Computer Graphics and 3D Watermarking.
Email: xywang@ujs.edu.cn

