Interactive 3D Animation Creation and Viewing System based on Motion Graph and Pose Estimation Method

Masayuki Furukawa, Hiroshi Kawasaki
Kagoshima University
1-21-24 Korimoto
Kagoshima, Japan
sc108044, kawasaki@ibe.kagoshima-u.ac.jp

Yasuhiro Akagi
Tokyo University of Agriculture and Technology
2-24-16 Higashi-cho, Koganei
Tokyo, Japan
akagi-y@cc.tuat.ac.jp

Yukiko Kawai
Kyoto Sangyo University
Kamikamomotoyama, Kitaku
Kyoto, Japan
kawai@cc.kyoto-su.ac.jp

ABSTRACT

This paper proposes an interactive 3D animation system specifically aiming efficient control of human motion. However there are various commercial products for creating movies and game contents, those are still difficult to deal with for non-professional users. To ease the creation process and encourage to utilize 3D animation for the general users, e.g., in the field of such as education, medicine and so on, we propose a system using Kinect. The data of skeleton models of human motion estimated by Kinect is processed to generate Motion Graph and finally restructure the data automatically for 3D character models. We also propose an efficient 3D animation viewing system based on touch interface for tablet device, which enables intuitive control of multiple motions of the human activity. To evaluate the effectiveness of the method, we implemented a prototype system and created several 3D animations.

Categories and Subject Descriptors

H.5.1 [Multimedia Information Systems]: Animations; H.5.2 [User Interfaces]: Interaction styles; K.3.1 [Computer Uses in Education]: Computer-assisted instruction

General Terms

Human Factors

Keywords

3D animation; Motion Graph; Realtime 3D scan;

1. INTRODUCTION

Nowadays, 3D computer graphics (3DCG) characters of human, animals and other moving objects can be seen everywhere in the movies, games and etc. Those become one of the important categories of multimedia contents.

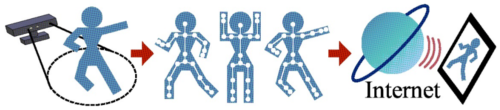
We call such animated 3DCG characters "3D animations" in the paper. To efficiently make 3D animations, a motion capturing system is commonly used to obtain the motion of the human skeleton. Since such a motion capturing system is basically for professional users and special skills are required to use the 3DCG softwares, it is difficult for non-professional users to create 3D animations. However, small-scale demands (content creation from non-professional users) are rapidly increasing for educational and/or medical purposes for small entities. To realize non-professional users to create a 3D animation without the motion capturing system and special skills on 3DCG, we propose a 3D animation creation system using Kinect [6]. Since a Kinect based system does not require any special devices and additional knowledge, non-professional users can create 3D animation easily without trainings.

The proposed system consists of three parts. The first one is a skeletal motion estimation part using a series of dense 3D point cloud which is captured by Kinect. The second one is an automatic Motion Graph [3] construction part. The third one is a shape interpolation part which realizes smooth transition between two motions. To implement the system, there are two critical issues which must be solved. The first one is low accuracy of skeletal motion estimation by Kinect. The other is a realization of smooth transition between different frames. For the first issue, we propose an exemplar based technique to re-estimate the motion; the technique is based on the assumption that similar motions frequently occur during human activity. For the second issue, we propose a new similarity calculation method for 3D point cloud which uses not only the target frame, but also neighboring frames. We also propose an efficient viewing system of animations using tablet devices, which shows the next candidates of the animation with superimposed information on the tablet devices. To evaluate the effectiveness of the system, we implement a prototype system, create several 3D animations and test the created contents with several users.

MM'14, November 3–7, 2014, Orlando, Florida, USA.
Copyright 2014 ACM 978-1-4503-3063-3/14/11 ...$15.00.
http://dx.doi.org/10.1145/2647868.2655055.

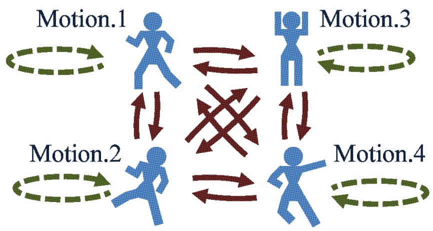
2. RELATED WORK

Although human movements seem complex, most of then are actually a repetition or combination of basic movements



(1) Capture (2) Structurization and editing (3) Interactive control of 3D animation

(a) Overview of the system.



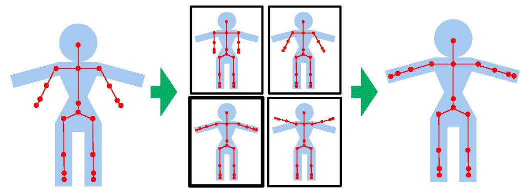
(b) Motion Graph of 3D animation. The solid lines show the transition-edge(s) and the dotted lines show the self-loop(s).

Figure 1: System overview and Motion Graph.

(e.g. walking, hand waving). We call such basic movements "motion primitive" or simply "primitive" in the paper. To create 3D animations, a state transition graph, which is named Motion Graph [3], is commonly used. Motion Graph represents self-loops and transitions between multiple motions. By using the Motion Graph, an editor can efficiently control the motion of a character to create animations. Sine the Motion Graph is useful to control animations interactively, methods which automatically construct Motion Graph by analyzing the captured motions have been proposed [8, 5]. Zhao et al. proposed a method which minimizes the size of Motion Graph [8]. This method is useful to create a digest animation from a long-term motion data. Apostolakis et al. proposed a software to create human animations [2]. This tool can control a human type character by using a rigging technique based on a skeleton structure. Another research defines the similarity of each skeletal motion based on the joint points of the skeleton model [4]. Since those methods are based on accurate skeletal motions captured by the motion capturing systems for professional use, the creation of Motion Graph from the sequential dense 3D point could which has large errors in motion have not been considered yet.

3. 3D ANIMATION CREATION SYSTEM BASED ON MOTION GRAPH

The overview of our content creation system is shown in Figure 1. The 3D animation is created by first capturing the moving human by Kinect (Figure 1(a)-(1)), then analyzing the motion to construct Motion Graph [3] that is a key technique to allow users to easily control 3D character (Figure 1(a)-(2)). In the paper, skeletal motions of the human body are represented by Motion Graph which consists of motion primitives of self-loop as the node and that of transition-edge between the nodes as the edge (Figure 1(b)). Finally, 3D characters are interactively controlled and animated by our viewing system (Figure 1(a)-(3)).



(a) Input frame (b) Exemplar (c) Re-estimation

Figure 2: Exemplar based skeleton re-estimation.

As the detailed steps of our proposed system, firstly we re-estimate the skeleton model which is estimated by Kinect, since the accuracy of the skeleton of Kinect is low (Sec. 3.1). Secondly, a method to create the Motion Graph is described in Sec.3.2. In the section, a similarity calculation method based on the series of the skeleton model is proposed. Finally, an interpolation method to realize the smooth transition of the 3D point cloud as well as the skeletal motion is explained in Sec. 3.3.

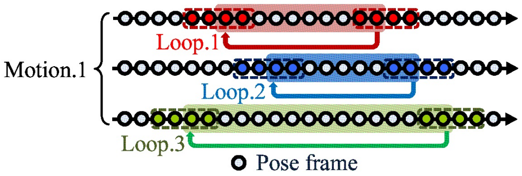
3.1 Re-estimation of skeleton by exemplar

Skeletal motions, which used to create Motion Graph, are estimated from the point cloud captured by Kinect. It is known that there are gaps between the skeletons and the point cloud due to inaccuracy of Kinect (Figure 2 (b)). If these frames are selected to be used for a self-loop or a transition-edge, an inconsistent transition will be generated. Therefore, to re-estimate the skeleton we propose a method as follows: (1) Detect frames with low accuracy on skeletal estimation by Kinect. (2) Apply an exemplar based skeleton re-estimation method for the detected low accuracy frames. (3) Reject frames which have lower accuracy than the threshold after step (2).

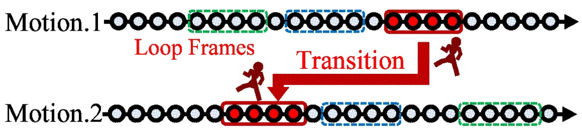
Frames with low accuracy after the re-estimation step will be excluded for further process to prevent inconsistent transition between motions. In order to detect frames with low accuracy on the skeleton estimation in the step (1), we first calculate the average distance between 3D points of point cloud and their nearest joint of the skeleton, and then, verify that it is below the threshold (Figure 2 (a)). If we leave frames which have the average distance less than 0.1m, only 10% of the total frames remained in the experiments. Based on the assumption that the skeleton includes many repetitions, remained skeletons are considered to be exemplar for the next step (Figure 2 (b)). In the step (2), the skeleton with the minimum average distance from the target 3D point cloud is searched as the correct skeleton (Figure 2 (c)). If the distance is above $0.1m$ after step (2), the frame is excluded from the further process. Valid skeletons increased to 50% of the total frames in the experiments.

3.2 Automatic Motion Graph creation

Motion Graph is created by self-loop and transition edge. Figure 3 shows the examples of detected self-loop and transition-edge on captured sequences. The actual detection process is as follows. First, all the distances for all two-frame pairs (p, q) within the skeletal motion are calculated. Then, frame sets which have the distance below the threshold are selected as candidates for self-loop. Since we consider the shorter loop is better for self-loop, we recalculate the distance by



(a) Example of multiple detection of self-loops.



(b) Example of transition-edges between motions.

Figure 3: Definition of self-loop and transition-edge.

taking advantage of the following weight function.

$$W(i,j) = \left| \frac{i-j}{FrameSize} \right|^2, \quad (1)$$

where $FrameSize$ represents the input number of frames of skeletal motion. Finally, we decide the pair of two frames (p, q) with minimum value of $W(p, q) * DistMotion(p, q)$ to define the self-loop motions (Figure 3).

After self-loop motions are detected, transition-edges are detected by calculating all the distances between all combinations of skeletal motions of two motions. Then, the set of two frames of the smallest distance which is below threshold is defined as the transition-edge. To calculate the distance between two skeletal motions, we use the equation proposed by Krüger et al. [4] as follows:

$$DistPose(i,j) = \sum_{k=1}^{n} ||\mathcal{F}_{E,j}^{k} - \mathcal{F}_{E,i}^{k}||), \quad (2)$$

where $\mathcal{F}_{E}^{n \times d}$ represents feature points of human body (hand, head, etc.), n is a number of feature points, d is a number of dimensions of the feature point, i and j are the time (number of frames) for two motions. When using equation (2) by itself, in some motion cases such as walking, two connections are created at the same pose, but opposite direction and one of them is wrong. To overcome this problem, we use the distance between multi-frame poses as follows:

$$DistMotion(p,q) = \sum_{i=1-N}^{N} \frac{DistPose(p+i, q+i)}{2N}, \quad (3)$$

where N is the window size. By using the equation (3), we can detect the distance between frames in consideration of the direction of the motion.

3.3 Interpolation of skeletal motion for smooth transition between frames

To make the motion primitives of self-loops and transition-edges, the most similar poses are detected to connect the motions as explained in the previous sections; however, they can not be the exact same poses. Therefore, the final 3D animation becomes unnatural, if two frames are simply connected without interpolation. To solve the problem, we replace the original n skeletal motions before/after connecting frames with the interpolated ones. To interpolate the motions, spline function is applied to five successive frames. Effectiveness of the method is evaluated in Sec. 4.3.

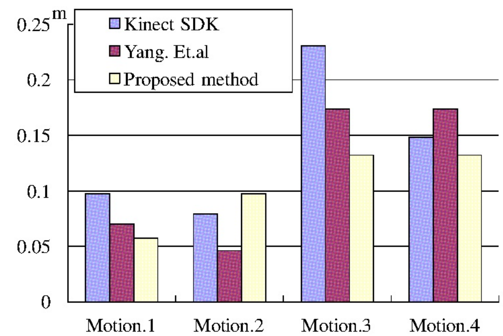


Figure 4: Comparison of the temporal smoothness of skeletal motions estimated by three methods.

4. EXPERIMENT AND EVALUATION

To evaluate our approach, we create 3D animations for experiments based on actual human motion captured by Kinect. We captured four types of gymnastic exercises as: "Stretching arms" (motion 1), "Spreading arms with knees-bent" (motion 2), "Swinging arms" (motion 3) and "Deep breathing" (motion 4).

In the experiments, we evaluate our re-estimation method in Section 4.1, and we also evaluate the effectiveness of the interpolation method for self-loops and transition-edges in Section 4.3. Finally, 3D interactive animations for our viewing system are shown, which can be easily operated by touch interface on tablet devices [1].

4.1 Evaluation of the exemplar based skeleton estimation method.

Since an actual skeletal motion of a human is usually smooth, if a skeletal motion is estimated correctly, the average of the difference of two skeletons between consecutive frames ($\sum_{t=0}^{N-1} DistMotion(t, t+1)/N$, see Eq.3) should have small value. Based on the assumption, we compare the proposed method to other two methods [6, 7]. Figure 4 shows the results. The results of the "Motion 1", "Motion 3" and "Motion 4" show that the proposed method is best among three. However, in the result of "Motion 2", the proposed method is worst. This is because "Motion 2" includes occluded regions and failed to track the movement of arms and legs by Kinect, and thus, enough exemplar data sets are not created for re-estimation. Therefore, the evaluation values of the previous methods are better than the proposed method.

4.2 Evaluation of Automatic Motion Graph creation

We evaluate the quality (the smoothness of the changing frame of motions) of animations which are created by the proposed Motion Graph creation method and those which are created by a human (manually). We ask 16 viewers which animation has smoother transitions than another between motions. In this evaluation, we use four types of self-loop motions (L1, L2, L3, L4) and one transition-edge (T) which consist of Motion 1-4. The results are shown in Fig-

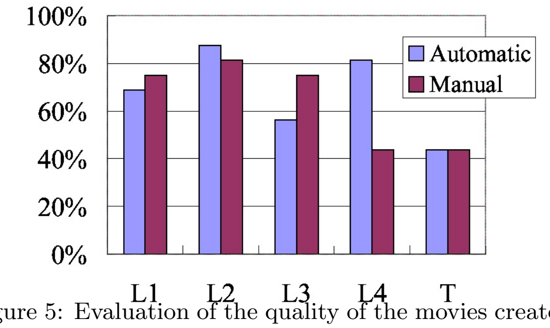


Figure 5: Evaluation of the quality of the movies created by manual process and our automatic Motion Graph construction method.

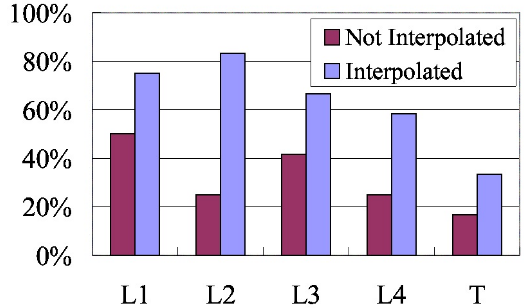


Figure 6: Evaluation of the quality of movie with/without interpolation method.

ure 5. The vertical axis of Figure 5 is the percentage of the viewers who answer the transitions are smooth. These results shows that the Motion Graph is successfully created with smooth motion transitions.

4.3 Evaluation of the interpolation method

We evaluate the effectiveness of the interpolation method which is described in the section 3.3. For this evaluation, we use the same animation in the previous section. Then, 15 viewers are asked which movie is more natural (smoothly change the motions) than the others. The results are shown in Figure 6. The results show that the proposed interpolation method successfully improves the quality on the smooth transitions of the 3D animations for all data.

4.4 Interactive control of 3D animation with tablet devices

Finally, our interactive 3D animation viewing system is evaluated by using our interactive viewing system with the tablet devices where 3D animations are efficiently controlled. 12 viewers are asked five questions after playing the 3D animations with the tablet device. The evaluation results are shown in Table 1. This results show that viewers felt that the contents were more interesting with 3D animation than an ordinary 2D video. It is also confirmed that novelty and user-friendliness of the system are higher than the 2D video. Furthermore, the result regarding the level of comprehension and usability were satisfactory.

5. CONCLUSION

In this paper, we propose an interactive 3D animation system which consists of an automatic 3D animation creation system and an interactive viewing system for tablet devices. For the automatic 3D animation creation system, the automatic Motion Graph creation method and the efficient interpolation technique of 3D point cloud are proposed. To realize automatic creation of Motion Graph, the distance calculation between the series of skeletal motions are proposed. For 3D point cloud interpolation, since the accuracy of estimated motions are generally low with Kinect, the exemplar based re-estimation method of skeletal motions is proposed. The advantages of the method are evaluated by creating the sample 3D animations. As the result, it is confirmed that the system can automatically create the Motion Graph with smooth transitions. More effective re-estimation of the skeleton model is our future research.

Table 1: Evaluation of the viewing system: (Q1) Attractiveness on contents, (Q2) Comprehension on contents, (Q3) Controllability of the contents, (Q4) Novelty of the system and (Q5) Intuitiveness the system interface.

	Percentages of the evaluation				
	Poor	Fair	Average	Good	Excellent
Q1	0.0	8.3	8.3	50.0	33.3
Q2	0.0	0.0	58.3	33.3	8.3
Q3	8.3	16.7	33.3	41.7	0.0
Q4	0.0	8.3	8.3	50.0	33.3
Q5	0.0	25.0	25.0	41.7	8.3

6. REFERENCES

[1] Y. Akagi, M. Furukawa, S. Fukumoto, and H. K. Y. Kawai. Interactive 3d animation system based on touch interface and efficient creation tools. In The 2013 IEEE International Conference on Multimedia and Expo (ICME) 2013 in "Multimedia for Humanity" theme track, 318, pages 1–7, 2013.

[2] K. Apostolakis and P. Daras. Raat - the reverie avatar authoring tool. In Digital Signal Processing (DSP), 2013 18th International Conference on, pages 1–6, July 2013.

[3] L. Kovar, M. Gleicher, and F. Pighin. Motion graphs. In Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '02, pages 473–482, New York, NY, USA, 2002. ACM.

[4] B. Krüger, J. Tautges, A. Weber, and A. Zinke. Fast local and global similarity searches in large motion capture databases. In Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '10, pages 1–10, Aire-la-Ville, Switzerland, Switzerland, 2010. Eurographics Association.

[5] S. Levine, Y. Lee, V. Koltun, and Z. Popović. Space-time planning with parameterized locomotion controllers. ACM Trans. Graph., 30(3):23:1–23:11, may 2011.

[6] Microsoft. Xbox 360 Kinect, 2010. http://www.xbox.com/en-US/kinect.

[7] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '11, pages 1385–1392, 2011.

[8] L. Zhao, A. Normoyle, S. Khanna, and A. Safonova. Automatic construction of a minimum size motion graph. In Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '09, pages 27–35, New York, NY, USA, 2009. ACM.