Online Motion Synthesis for Manipulating a Virtual Ball

Jongin Choi1, Junghuem Kwon2*

AbstractIn the real world, it takes considerable effort and extensive practice for a person who is poor at controlling a ball to become good at it. Moreover, manually synthesizing character animation for controlling a ball is arduous work, because the motion of the character must be synchronized with the ball, which moves according to the laws of physics. Therefore, this study proposes a novel method that generates the movement of a ball from motion capture data, and allows anyone to immediately become good at controlling a ball in the virtual world. Because it is difficult for a novice to control a ball, this study does not use an actual ball. Instead, motion that mimics control of a ball is captured and used. Frames and positions in which a character could interact with a ball are identified by analyzing the motion capture data, and then the movement of the ball is generated according to rules. Moreover, this study proposes a convenient method for controlling a character that controls a ball. Finally, we demonstrate the usefulness of our method through soccer ball juggling and basketball dribbling.

Keywords
Character Animation, Physics-Based Animation, Virtual Reality, Computer Graphics

1. INTRODUCTION

Most of us have seen someone juggling a soccer ball, skillfully controlling the ball with both feet. It takes considerable effort and extended practice for a person who is not skilled at controlling a ball to develop skills. Therefore, this study proposes an interesting system that allows a person to become good at controlling a ball in the virtual world, despite being unskilled at controlling a ball in the real world. Manually animating a character that controls a ball competently is problematic and tedious work, because the motion of the character must be synchronized with that of the ball, which moves according to the laws of physics, both temporally and spatially.

Because this system generates the movement of a ball from motion capture data, anyone can immediately become skillful at controlling the ball in the virtual world. We noticed, when we observed closely, that similar motions are repeated in juggling a soccer ball and dribbling a basketball. It is possible to find rules that describe the motions that are repeated when humans interact with objects. Using these rules, we generated animation for a character that controls a ball skillfully. This technique used only motions that mimicked controlling a ball, instead of using an actual ball, and difficult and flashy movements were excluded as much as possible. This was done to allow an unskilled novice to freely capture his/her own motion. We generated the natural movement of a ball according to the laws of physics by using a method [18] in which the user could freely control the rigid body path. This method extracts constraints for generating the movement of the ball by analyzing the motion of the character.

Moreover, we created a convenient interface for controlling a character that controls a ball. To do this, the motion capture data were divided into many motion clips (i.e., connectable minimum units), and then the information to connect them to each other was generated. The user could select the desired motion of the character simply by clicking the mouse. Animation for a character controlling a ball skillfully could be synthesized by connecting the motion clips in the order that the user selected, and then generating the movement of the ball. This method required three steps. First, we analyzed the motion capture data; second, the movement of a ball was generated according to the laws of physics; and third, we controlled the motion of an animated character by using the input from the user. This method is characterized by the following: anyone can freely capture and use his/her own motion; anyone can immediately learn to control a ball skillfully; anyone can easily control the motion of a character that controls a ball.

2. RELATED WORKS

There have been a number of studies on character interaction. Most physics-based character animations have generated characters that react to terrain or external forces. There have been many studies of example-based character animation for character interactions (e.g., interaction with outside environments [1,2,3,4], external forces [5,6,7], or other characters [8,9,10,11]). Most studies have focused on how external factors affect the character. In addition, hybrid methods [12,13,14,15] that mix physics- and example-based animation have created characters that are influenced by external environments and forces. Conversely, this study generated interesting results for controlling objects by using the motion of the character. Users were allowed to create an interaction animation in which the character controls an object according to the laws of physics by generating the motion of the object from motion capture data.

Motion analysis has primarily been applied to research for synchronizing music with character motion. T.H. Kim et al. extracted regular patterns from the motion of a character and synchronized them with input music [16],

Revised Manuscript Received on May 06, 2019

Jongin Choi1, Major of Game Contents, Youngsan University, 142 Bansongsunhwan- ro, Haeundae- Gu, Busan 48015, Korea
Junghuem Kwon, Department of Convergence, Korea University of Technology & Education, 1600 Chungjeol- ro, Byeongcheon- myeon, Dongnam- gu, Cheonan City, Chingnam Province 31253, Korea

thus generating a character rhythmically animated according to the music. Shiratori et al. classified character motion and input music according to its intensity, and generated dance motion that matched the music [17]. Popović et al. proposed a method for controlling the movement trajectory of a rigid body according to the laws of physics [18, 19]. They generated the natural movement trajectory of objects by calculating initial values, such as position or velocity, and by optimizing for various constraints. With the laws of physics method [18], we generated a ball trajectory, using the start and end position of interacting bones in a detached section and the bouncing of the ball as constraints. Jain and Liu proposed a method for editing the movement trajectory of an object and the motion of a character simultaneously [20]. This enabled users to edit the movement trajectory of the object and the motion of the character, and to change the movement trajectory of the object by detecting collision between the object and character.

Character motion control by user input has been studied actively in most example-based character animation. Both Kovar et al. and J. Lee et al. synthesized the animation of a character to move it to a goal position, or along a path specified by a user, by generating a motion graph from a huge motion capture database [21,22]. Heck and Gleicher controlled a character in detail by mixing similar motions effectively [23]. For example, the character could reach with an exact position, or punch with a wanted position. Gleicher created a character that moved along a path that a user had edited [24]. Thorne et al. controlled the movement of a character by using a sketch-based user interface [25]. A user could draw various movements and movement paths (e.g., normal walking, sneaking, marching, jumping) by using an electronic pen.

Both Gleicher et al. and Levine et al. controlled a character conveniently by generating a motion graph with a common pose as the center [26]. For example, in a competition fighting game, a character can return to an idle pose after attacking (e.g., punching or kicking). Gleicher et al., Y. Lee et al., S. Levine et al., and A. Treuille et al. created a character to interact with user input more rapidly and effectively by using reinforcement learning [27,28,29,30]. Levine et al. made a character that moved and effectively reached a goal that had been specified by a user in a dynamic environment [31]. Choi et al. generated various character motions that were controlled with a ball but the motions were not generated in real time [32]. In addition, it does not create the character's motion and only generates the movement of a ball according to the character's motion. They generated character motion that was limited to juggling [33]. By reviewing all related works, this study proposes a method for generating character motion in real time for various ways of controlling a ball with the hand or foot.

3. MOTION DATA ANALYSIS

The exact movement of a ball cannot know because we do not use an actual ball. Instead, we extract information to generate the motion of a ball by analyzing motion capture data. The movement of the animated ball is then generated with the information. This method requires information about the section in which the ball is attached to an interaction bone, and the section in which the ball is detached from it. To generate this information, we extract essential information by using the velocity graphs of bones that interact with the ball. Figure 1 represents in red the interaction bones that the user specifies. The left character specifies both feet as interaction bones, and the right character specifies both hands.

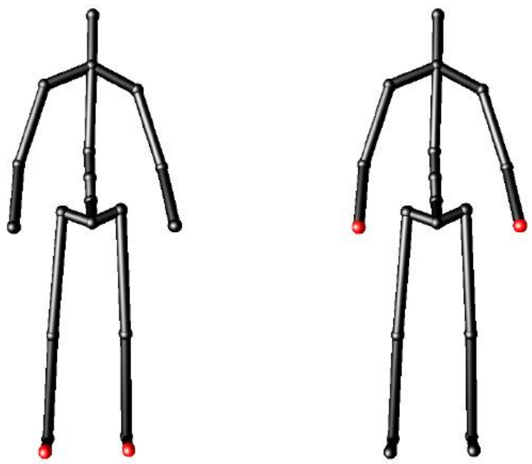

Fig 1. Interaction bones specified on the character

3.1 Motion Capture for Novices

We want to allow a person who is poor at controlling a ball to capture his/her own motion. Therefore, rather than using an actual ball, we captured the motion that mimics control of a ball. Furthermore, we excluded difficult and showy motions that are hard to do without an actual ball as much as possible. In addition, we captured and used several easy and simple motions that anyone can do. We also captured the motion of kicking a ball with both feet continuously for soccer ball juggling, and we captured the motion of bouncing a ball with both hands continuously for basketball dribbling.

3.2 Velocity Graph of Interaction Bone

An interaction bone is a character bone that contacts a ball directly, such as the hand or foot. Graphs are generated by using the vertical (y-axis) velocity of the interaction bones. Most of the movement of interaction bones is in the vertical direction, because one character alone controls the ball. We generate velocity graphs of interaction bones using the horizontal axis for time, and the vertical axis for velocity.

The top part of Figure 2 presents velocity graphs of both feet for soccer ball juggling, and the bottom presents velocity graphs of both hands for basketball dribbling. In the top graph, red represents the left foot and blue represents the right foot. On the velocity graph, the red and blue dots represent local maximum velocity, local minimum velocity, and zero speed, respectively. The red and blue dotted lines represent frames in which the ball was contacted (hit frame). The thick black lines under the graphs represent sections of the motion clip when the subject was standing. In the bottom graph, red represents the left hand and blue represents the right hand. On the velocity graphs, the red and blue dots represent local maximum velocity and local minimum velocity, respectively. The thick red

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

line under the graph represents the duration of contact between the left hand and the ball, and the thick blue line represents the duration of contact between the right hand and the ball. The characters at the bottom show the poses that correspond with each frame.

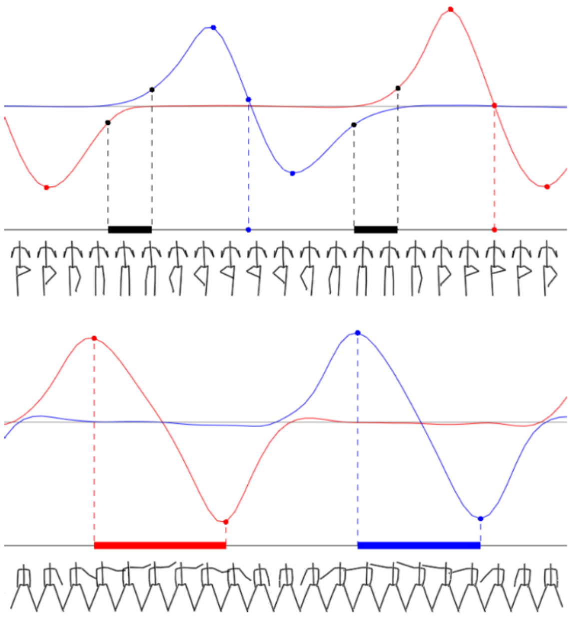

Fig 2. Motion data analysis

3.3 Interaction Information Generation

Frames in which an interaction bone has a local maximum or minimum velocity were identified as the interaction frames, because bio-mechanically, the interaction bone reaches maximum velocity when we throw or catch a ball. To exclude noise, we did not use frames that had a velocity between half the global minimum and half the global maximum. In the case of kicking a ball, we determined that it looked more natural at the zero-velocity frame than in the local maximum velocity frame because of movement direction of the foot when it hit the ball. Thus, the zero-velocity frame was specified as the hit frame. We searched interaction frames for all the interaction bones by using the following method:

Local maximum velocity frame. The frame with the maximum velocity in the section between the frame in which the velocity of an interaction bone changes from negative to positive and the frame in which the velocity of an interaction bone changes from positive to negative.

Local minimum velocity frame. The frame with the minimum velocity in the section between the frame in which the velocity of an interaction bone changes from positive to negative and the frame in which the velocity of an interaction bone changes from negative to positive.

Hit (zero-velocity) frame. A frame that captures the instant in which a ball is hit. The frame with velocity closest to zero in the section between one local maximum velocity frame and the next local minimum velocity frame.

Stand section. A section in which both feet are in contact with the ground simultaneously. The section in which the velocities of both feet are closest to zero between one local minimum velocity frame and the next local maximum velocity frame.

Attached section. A section in which a ball moves with an interaction bone between one local maximum velocity frame and the next local minimum velocity frame.

Detached section. The section in which a ball moves away from an interaction bone between two adjacent hit frames or two adjacent attached sections.

4. Character Motion Synthesis

This section presents a convenient method for synthesizing character motion for controlling a ball. We used a method similar to that of Treuille et al. [30] for synthesizing the motion of a character. Motion capture data were divided into several motion clips that were the minimum units that could be connected to each other. The motion clips were then classified according to the interaction bone used, and information was generated to connect the motion clips. Finally, by connecting the motion clips that were selected by a user, animation to interact freely with a ball could be synthesized for a character.

4.1 Motion Clip Generation

By dividing the motion capture data into sections, many motion clips were generated for synthesizing various motions in which a character controls a ball. The criterion for dividing the motion capture data differed according to the rule for controlling the ball.

In the case of soccer ball juggling, we divided the motion capture data using the stand section as a criterion. In other words, one motion clip was a span from one stand section to the next stand section. If we wanted to connect gaps between two neighboring stand sections, then one motion clip would be generated for soccer ball juggling. Similarly, in the case of basketball dribbling, motion capture data were divided using the attached section as a criterion. In other words, one motion clip was the span from one attached section to the next attached section. If we wanted to connect gaps between two neighboring attached sections, one motion clip was generated for basketball dribbling.

Figure 3 presents the results of the motion clip generation. At the top, the thick black lines represent stand sections, the thick red lines represent the sections in which the left foot was used, and the thick blue lines represent the sections in which the right foot was used in the motion clip. At the bottom, the thick red lines represent the attached section for the left hand, the thick blue lines represent the attached section for the right hand, and the thick black lines represent detached sections. Characters exhibit representative poses in the corresponding sections.

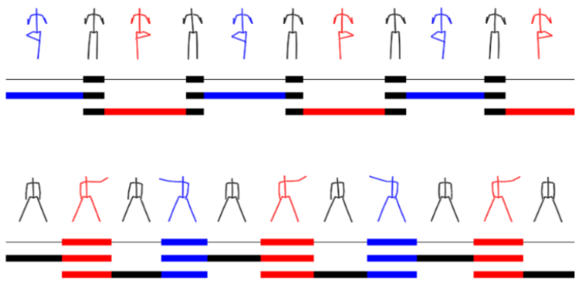

Fig 3 Motion clips generated from motion capture data

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

4.2 Motion Clip Connection

Motion clips were classified so as to connect them according to the interaction bone used. In the case of soccer ball juggling, we classified them into two groups ((L) and (R)) according to the foot used, because one motion clip used only one interaction bone. (L) represented the left foot, and (R) represented the right foot.

In the case of basketball dribbling, we classified the motion clips into four groups ((L, L), (L, R), (R, L), and (R, R)) according to the hand used, because one motion clip used two interaction bones. (L, L) represented using the left hand in the first attached section and the left hand in the second attached section. (R, R) represented the same with the right hand. Moreover, (R, L) represented using the right hand in the first attached section and the left hand in the second attached section. (L, R) represented the reverse of (R, L).

Figure 4 shows a connection graph used to generate the classification of motion clips. Unlimited character animation for controlling a ball could be synthesized by connecting the motion clips. In the case of soccer ball juggling, there were no constraints at all, because the transition between motion clips occurred in the stand section. Therefore, all motion clips could be connected freely. However, in the case of basketball dribbling, the hand used in the second attached section of the first motion clip had to be the same as the hand used in the first attached section of the second motion clip, because the transition between motion clips occurs in the attached section. Because all motion clips have transition sections of different lengths, the average length of the transition was used in the section of two motion clips as the transition length by using time warping.

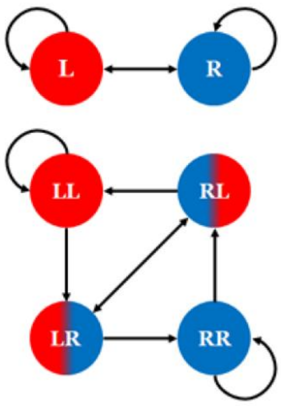

Fig4 Connection graph for motion groups classification

Figure 5 shows the connections between motion clips. At the top, the black parts represent the stand sections, the red parts represent sections in which the left foot was used, and the blue parts represent sections in which the right foot was used. At the bottom, the red parts represent the attached sections for the left hand, the blue parts represent the attached sections for the right hand, and the black parts represent the detached sections. In the transition section, time warping was used to blend the two motion clips.

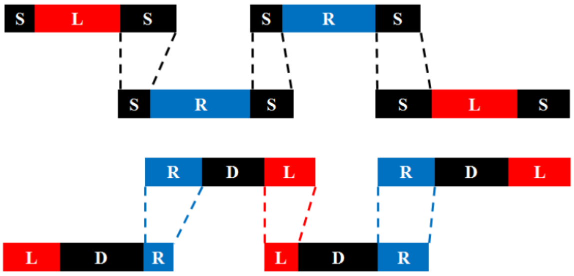

Fig 5 Connections between motion clips

To match poses, equation 1 was used in the transition section between the two motion clips as closely as possible. This was the method used in Kovar et al. [21] to convert the pose of the character to point cloud form in the transition section, and then to find the transformation that matched the pauses in the section. The linear transformation T_{θ,x_0,y_0} rotates the point p about the y (vertical) axis by θ, and translates it by (x_0,z_0). i is the index of the point in the point cloud. ω is the weight given to the point. θ, x_0, and z_0 are calculated by equations 2, 3, and 4, respectively.

$$\min_{\theta,x_0,y_0} \sum_i \omega_i \left\| p_i - T_{\theta,x_0,y_0} \right\|^2 \qquad (1)$$

$$\theta = \frac{\sum_i \omega_i (x_i z'_i - x'_i z_i) - \frac{1}{\sum_i \omega_i}(\bar{x}\bar{z}' - \bar{x}'\bar{z})}{\sum_i \omega_i (x_i x'_i - z'_i z_i) - \frac{1}{\sum_i \omega_i}(\bar{x}\bar{x}' - \bar{z}'\bar{z})} \qquad (2)$$

$$x_0 = \frac{1}{\sum_i \omega_i}(\bar{x} - \bar{x}' \cos\theta - \bar{z}' \sin\theta) \qquad (3)$$

$$z_0 = \frac{1}{\sum_i \omega_i}(\bar{z} - \bar{x}' \sin\theta - \bar{z}' \cos\theta) \qquad (4)$$

4.3 Foot-Slide Removal

The transition between motion clips was accompanied by a foot-sliding effect, because all the motion clips had different foot positions. In the case of basketball dribbling, because both feet were in contact with the ground, the problem could be overcome simply. We calculated the average foot position across all the frames, and then applied the average position to the entire transition frames by using inverse kinematics.

In the case of soccer ball juggling, the problem was solved with the method used in Gleicher et al. [26]. First, all poses in the center frames of the stand sections were assessed, and then the average pose was calculated. Next, the average pose was applied to all the center frames of the stand sections. Third, the difference between the original pose and the average pose in all the center frames of the stand sections was calculated. Finally, displacement maps were generated between two different neighboring poses, and were applied to all the frames by adding them to the original pose.

5. Ball Trajectory Generation

This study generates ball trajectory separately for attached sections and detached sections, because a ball moves together with an interaction bone in an attached section, but it moves according to the laws of physics in a detached section. If we connect the ball trajectories t generated in each section in order, a natural-looking whole ball trajectory can be

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

created.

5.1 Trajectory in Attached Sections

In the attached sections, the ball position is the same as the position of the interaction bone. Therefore, the trajectory of an interaction bone becomes the ball trajectory. In the case of basketball dribbling, because the transition section is the same as the attached section, the position that blends the two interaction bones of the two motion clips is the ball position in the transition section. In the case of soccer ball juggling, there is no attached section at all. Figure 6 shows the ball trajectory in an attached section. The black dotted lines represent ball trajectories; the left is the trajectory in the right hand, and the right is the trajectory in the left hand.

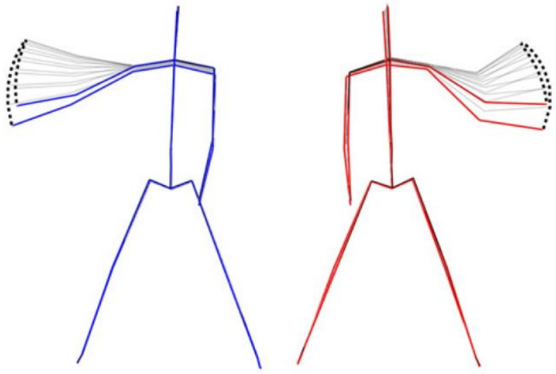

Fig 6. Ball trajectory in attached section

5.2 Trajectory in Detached Sections

Because the ball moves away from an interaction bone in a detached section, we generated a ball trajectory according to the laws of physics by using the start position and end position as constraints, similar to the method of Popović et al. [18]. In the case of soccer ball juggling, the detached section spanned from the hit frame of one motion clip to the hit frame of the next motion clip. Therefore, the start position of the ball was the position of an interaction bone in the hit frame of one motion clip, and its end position was the position of an interaction bone in the hit frame of the next clip.
In the case of basketball dribbling, the entire detached section was included in one motion clip. Therefore, the start position of the ball was the position of the interaction bone in the first frame of the detached section, and the end position was the position of the interaction bone in the last frame of the detached section. Figure 7 shows the ball trajectory in a detached section. The black dotted lines represent ball trajectories, the red character assumes the pose of the start frame, and the blue character assumes the pose of the end frame.

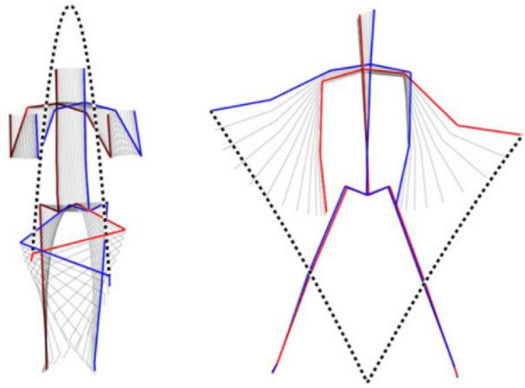

Fig 7. Ball trajectory in detached section

6. EXPERIMENTS

This study synthesized character animation for skillful control of a ball, as in soccer ball juggling and basketball dribbling. We provided a convenient user interface for controlling the motion of a character, with connection graphs of the classified motion groups.
Figure 8 shows the user interface for selecting the motion group. The selected motion group has many similar motion clips, and one motion clip is selected at random.
Figure 9 and Figure 11 show the synthesized interaction animation of soccer ball juggling and basketball dribbling with motions selected by the user. The large yellow slot at the top left shows the current motion, and the light green slot shows the next motion. The various small icons at the top of the screen represent motions selected by a user, and the motions are animated according to input order.

6.1 Soccer Ball Juggling

In this study, we captured and used four motions for soccer ball juggling: kicking a ball with the inside of the foot, with the outside of the foot, with the top of the foot, and with the heel. Figure 8(a) presents two user interfaces for soccer ball juggling. The four icons on the left represent the four motions using the left foot that were captured, and the right four icons represent the four motions using the right foot. If the user left clicks the mouse, the left image in Figure 8 appears. Moreover, if s/he right-clicks the mouse, the right image in Figure 8 appears. Figure 9 shows the synthesized character animation for soccer ball juggling by a user.

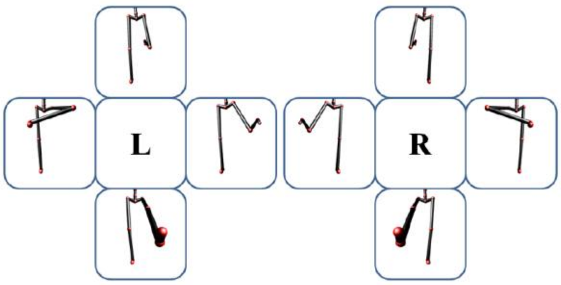

Fig 8. User interface for soccer ball juggling

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

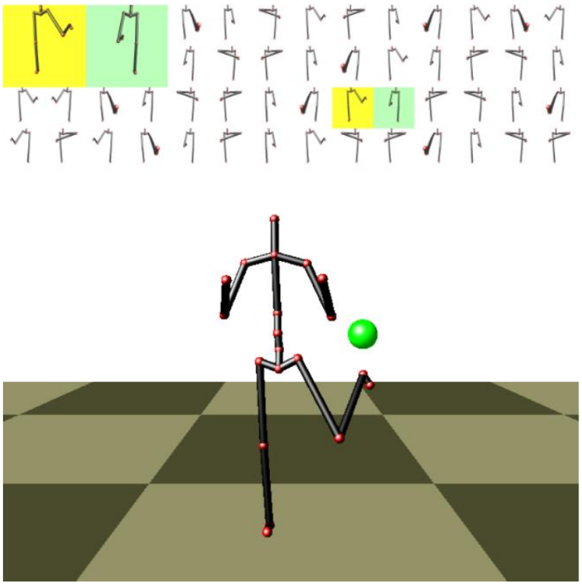

Fig 9. Soccer ball juggling animation

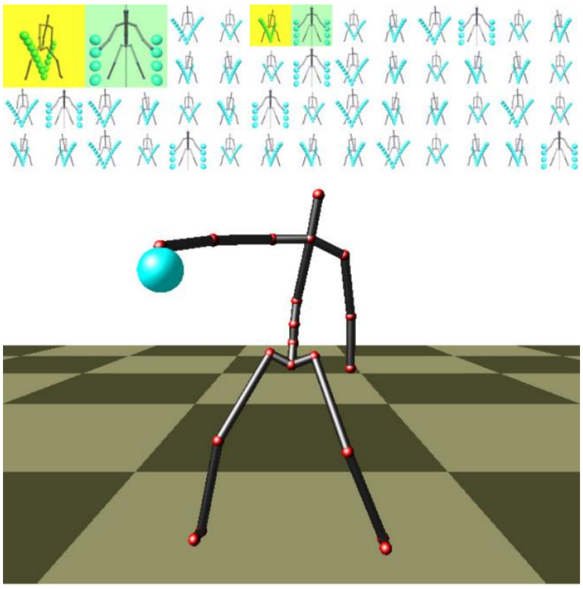

Fig. 11 Basketball dribbling animation

6.2 Basketball Dribbling

This study captured and used one-handed and two-handed dribbling motions for basketball dribbling. In the case of two-handed dribbling, we captured two motions: right-handed dribbling and left-handed dribbling. In the case of two-handed dribbling, we captured four motions: dribbling in front, in back, from left front to right back between the legs, and from right front to left back between the legs.

Figure 10 shows the user interface for basketball dribbling. The four icons at the left, right, top, and bottom represent the four captured two-handed dribbling motions, which are classified into two motion groups ((L, R) and (R, L)). The icon in the center represents the two captured one-handed dribbling motions, and it is classified into two motion groups ((L, L) and (R, R)). Each icon is related to two motion groups, as this enables a connection between all motion groups. Thus, the user does not need to manage the connection between the motion clips. Figure 11 shows the synthesized character animation for soccer ball juggling by a user.

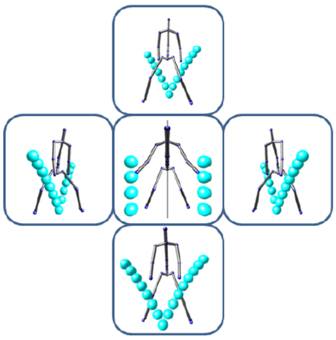

Fig 10. User interface for basketball dribbling

7. CONCLUSIONS AND FUTURE WORKS

This study proposed a method for synthesizing and controlling character motion for controlling a ball. Anyone can freely capture his/her own motion and use it. Moreover, because the motion of the character controlling a ball was generated by user input, the users can generate the kind of motion they prefer. The disadvantage of this method is that it does not consider the laws of physics, except for the generating of the ball trajectory in the detached section. For example, when a character kicks a ball, the collision angle or velocity between the ball and the foot are ignored. In addition, when the ball leaves the hand, the velocity and movement direction of the ball are ignored. Thus, such interactions can look somewhat unnatural.

Only five years ago, it was almost impossible to purchase motion capture equipment for personal use. Now, however, anyone can capture his/her own motion personally by using low-priced motion capture equipment such as Kinect for Xbox or Eye Toy for PlayStation. Therefore, if this method uses such inexpensive motion capture equipment, interesting software that makes anyone a skillful player able to freely control a ball can be created. We intend to research various recreational applications for this method by enhancing its entertainment value in the future.

ACKNOWLEDGMENT

This work was supported by a 2018 research grant from Youngsan University, and also supported by a research grant (2018-0423) from Korea University of Technology and Education, Republic of Korea.

REFERENCES

1. R.A. Al-Asqhar, T. Komura, and M.G. Choi, “Relationship descriptors for interactive motion adaptation,” Proceedings of the 12thACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA’13, New York, NY, USA, pp.45–53, ACM, 2013.

2. S. Cooper, A. Hertzmann, and Z. Popović, “Active learning for real-time motion controllers,” ACM SIGGRAPH 2007 Papers, SIGGRAPH ’07, New York, NY, USA, ACM, 2007.
3. K.H. Lee, M.G. Choi, and J. Lee, “Motion patches: Building blocks for virtual environments annotated with motion data,” ACMSIGGRAPH 2006 Papers, SIGGRAPH ’06, New York, NY, USA, pp.898–906, ACM, 2006.
4. Yamane, J.J. Kuffner, and J.K. Hodgins, “Synthesizing animations of human manipulation tasks,” ACM SIGGRAPH 2004 Papers, SIGGRAPH ’04, New York, NY, USA, pp.532–539, ACM, 2004.
5. Arikan, D.A. Forsyth, and J.F. O’Brien, “Pushing people around,” Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA ’05, New York, NY, USA, pp.59–66, ACM, 2005.
6. Komura, E.S.L. Ho, and R.W.H. Lau, “Animating reactive motion using momentum-based inverse kinematics: Motion capture and retrieval,” Computer Animation and Virtual Worlds, vol.16, no.3-4, pp.213–223, July 2005.
7. Ye and C.K. Liu, “Synthesis of responsive motion using a dynamic model,” Computer Graphics Forum, vol.29, no.2, pp.555–562, 2010.
8. E.S.L. Ho, T. Komura, and C.L. Tai, “Spatial relationship preserving character motion adaptation,” ACM SIGGRAPH 2010 Papers, SIGGRAPH ’10, New York, NY, USA, pp.33:1–33:8, ACM, 2010.
9. M. Kim, K. Hyun, J. Kim, and J. Lee, “Synchronized multi-character motion editing,” ACM SIGGRAPH 2009 Papers, SIGGRAPH ’09,New York, NY, USA, pp.79:1–79:9, ACM, 2009.
10. H.P.H. Shum, T. Komura, M. Shiraishi, and S. Yamazaki, “Interaction patches for multi-character animation,” ACM SIGGRAPH Asia 2008Papers, SIGGRAPH Asia ’08, New York, NY, USA, pp.114:1–114:8, ACM, 2008.
11. K. Wampler, E. Andersen, E. Herbst, Y. Lee, and Z. Popović,“Character animation in two-player adversarial games,” ACM Trans. Graph., vol.29, no.3, pp.26:1–26:13, July 2010.
12. A. Macchietto, V. Zordan, and C.R. Shelton, “Momentum control for balance,” ACM SIGGRAPH 2009 Papers, SIGGRAPH ’09, NewYork, NY, USA, pp.80:1–80:8, ACM, 2009.
13. N. Nguyen, N. Wheatland, D. Brown, B. Parise, C.K. Liu, and V. Zordan, “Performance capture with physical interaction,” Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA ’10, Aire-la-Ville, Switzerland, Switzer-land,pp.189–195, Erographics Association, 2010.
14. V. Zordan, A. Macchietto, J. Medin, M. Soriano, C.C. Wu,R. Metoyer, and R. Rose, “Anticipa-tion from example,” Proceedings of the 2007 ACM Symposium on Virtual Reality Software and Tecnology, VRST ’07, New York, NY, USA, pp.81–84, ACM, 2007.
15. V.B. Zordan, A. Majkowska, B. Chiu, and M. Fast, “Dynamic response for motion capture animation,” ACM SIGGRAPH 2005 Papers, SIGGRAPH ’05, New York, NY, USA, pp.697–701, ACM, 2005.
16. T.H. Kim, S.I. Park, and S.Y. Shin, “Rhythmic-motion synthesis based on motion-beat analy-sis,” ACM SIGGRAPH 2003 Papers, SIGGRAPH’03, pp.392–401, 2003.
17. T. Shiratori, A. Nakazawa, and K. Ikeuchi, “Dancing-to-music character animation,” Comput. Graph. Forum, vol.25, no.3, pp.449–458, 2006.
18. J. Popović, S.M. Seitz, M. Erdmann, Z. Popović, and A. Witkin, “Interactive manipulation of rigid body simulations,” Proceedings of the27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’00, pp.209–217, ACM Press/Addison-This studysley Publishing Co., 2000.
19. J. Popović, S.M. Seitz, and M. Erdmann, “Motion sketching for control of rigid-body simulations,” ACM Trans. Graph., vol.22, no.4, pp.1034–1054, Oct. 2003.
20. S. Jain and C.K. Liu, “Interactive synthesis of human-object inter-action,” Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA ’09, New York, NY, USA, pp.47–53, ACM, 2009.
21. L. Kovar, M. Gleicher, and F. Pighin, “Motion graphs,” Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’02, New York, NY, USA, pp.473–482,ACM, 2002.
22. J. Lee, J. Chai, P.S.A. Reitsma, J.K. Hodgins, and N.S. Pollard, “Interactive control of avatars animated with human motion data,” Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’02, New York, NY, USA,pp.491–500, ACM, 2002.
23. R. Heck and M. Gleicher, “Parametric motion graphs,” Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games, I3D’07, New York, NY, USA, pp.129–136, ACM, 2007.
24. M. Gleicher, “Motion path editing,” Proceedings of the 2001 Symposium on Interactive 3D Graphics, I3D ’01, New York, NY, USA, pp.195–202, ACM, 2001.
25. M. Thorne, D. Burke, and M. van de Panne, “Motion doodles: An interface for sketching character motion,” ACM SIGGRAPH 2006Courses, SIGGRAPH ’06, New York, NY, USA, ACM, 2006.
26. M. Gleicher, H.J. Shin, L. Kovar, and A. Jepsen, “Snap-together motion: Assembling run-time animations,” ACM SIGGRAPH 2008Classes, SIGGRAPH ’08, New York, NY, USA, pp.52:1–52:9, ACM, 2008.
27. J. Lee and K.H. Lee, “Precomputing avatar behavior from human motion data,” Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA ’04, Airela-Ville, Switzerland, Switzerland, pp.79–87, Eurographics Association,2004.
28. Y. Lee, K. Wampler, G. Bernstein, J. Popović, and Z. Popović, “Motion fields for interactive character locomotion,” ACM Trans. Graph., vol.29, no.6, pp.138:1–138:8, Dec. 2010.
29. S. Levine, J.M. Wang, A. Haraux, Z. Popović, and V. Koltun, “Continuous character control with low-dimensional embeddings,” ACM Trans. Graph., vol.31, no.4, pp.28:1–28:10, July 2012.
30. A. Treuille, Y. Lee, and Z. Popović, “Near-optimal character animation with continuous control,” ACM SIGGRAPH 2007 Papers, SIGGRAPH ’07, New York, NY, USA, ACM, 2007.
31. Levine, Y. Lee, V. Koltun, and Z. Popović, “Space-time planning with parameterized locomotion controllers,” ACM Trans. Graph., vol.30, no.3, pp.23:1–23:11, May 2011.
32. J.I. Choi, S.J. Kang, C.H. Kim, and J. Lee, “Virtual ball player: Synthesizing character animation to control a virtual ball from motion data using interaction patterns,” Visual Computer, vol.31, no.6-8, pp.905–914, June 2015.
33. J.I. Choi, S.J. Kim, C.H. Kim, and J. Lee, “Let’s be a virtual juggler,” Computer Animation and Virtual Worlds, vol.27, no.3-4, pp.443–450, May2016.

AUTHORS PROFILE

Jongin Choi, received an MS degree at Korea University in 2003 and received PhD in 2016 from the Department of Computer Science from Korea University. He has worked at Nexon Korea as a lead client programmer and NCSoft Korea as a lead animation programmer. And he is now a professor at the major of game contents in Youngsan University.

Email:gameai@ysu.ac.kr

Jounghuem Kwon, received PhD in 2010 at Visualization Group, WMG from Warwick University, UK. He has worked at NTU, SG as a research professor from 2009 to 2012, and CHIC, KIST as a principle researcher from 2012 to 2017. He is now an assistant professor at the department of future technology in Korea University of Technology and Education, Cheonan, Korea

Email: rjhkwon@koreatech.ac.kr

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication