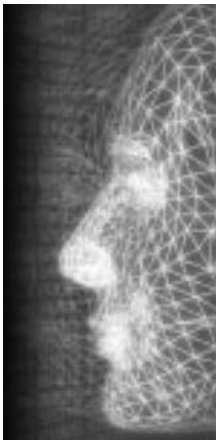
COMPUTER ANIMATION AND VIRTUAL WORLDS
Comp. Anim. Virtual Worlds (2009)
Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/cav.315

Angular momentum guided motion concatenation



By Hubert P. H. Shum, Taku Komura and Pranjul Yadav*

In this paper, we propose a new method to concatenate two dynamic full-body motions such as punches, kicks, and flips by using the angular momentum as a cue. Through the observation of real humans, we have identified two patterns of angular momentum that make the transition of such motions efficient. Based on these observations, we propose a new method to concatenate two full-body motions in a natural manner. Our method is useful for applications where dynamic, full-body motions are required, such as 3D computer games and animations. Copyright © 2009 John Wiley & Sons, Ltd.

Received: 25 March 2009; Accepted: 26 March 2009

KEY WORDS: motion editing; motion control

Introduction

Motion synthesis techniques such as motion blending and spacetime constraints are often used for producing new motions from existing motion capture data. Previous methods to concatenate or blend motions are less effective for dynamic, full-body motions such as punches, kicks and flips, as the postures to be blended are very dissimilar.

When we observe humans successively conducting two full-body motions, we realize that there are two important rules: (1) if the angular momenta of the two motions are in the same direction, humans try to make use of the angular momentum of the first motion to conduct the second motion, and (2) if they are in the opposite direction, the ending part of the first motion is used to prepare for the second one.

Based on these observations, we propose a method to simulate natural motions by concatenating two or more individual full-body motions. Parameters for blending are determined according to the angular momentum and the kinematics. Using our method, it is possible to produce animations of a character successively conducting flips in gymnastics and shadow boxing from individual movements (Figure 1). By simulating new motions using our method, the users can greatly simplify the motion capturing process as the number of motions needed to be captured is reduced. It can also overcome the difficulties due to the physical limitation of the human body and the spatial limitation of the capturing area.

Related Work

The main topic of this paper is to concatenate full-body motions such as those in gymnastics, dancing and fighting. Although there are many methods to synthesize/edit new movements from existing motions in computer animation, they are difficult to be applied for full-body motions. Previous methods can be categorized into spacetime optimization and motion blending.

Spacetime optimization[1] is suitable for producing ballistic motions in which the angular/linear momentum must be conserved. The method produces motions by minimizing the difference of the computed motion and the original captured motion, while using the conservation of the angular momentum and other conditions as constraints. Liu and Popović[2] propose a spacetime approach to generate realistic ballistic motions from a sequence of simple postures given by the user. Abe *et al.*[3] extend that method for editing motions such as jumping. Sulejmanpasic and Popović[4] propose a similar approach for computing ballistic motions. Fang and Pollard[5] also propose a spacetime approach to generate realistic motions such as running and acrobatic movements. Spacetime optimization is computational costly and can get caught into local minima when a large number

*Correspondence to: H. P. H. Shum, School of Informatics, University of Edinburgh, Informatics Forum, 10 Crichton Street, Edinburgh, EH8 9AB, Scotland, UK.
E-mail: hubert.shum@ed.ac.uk

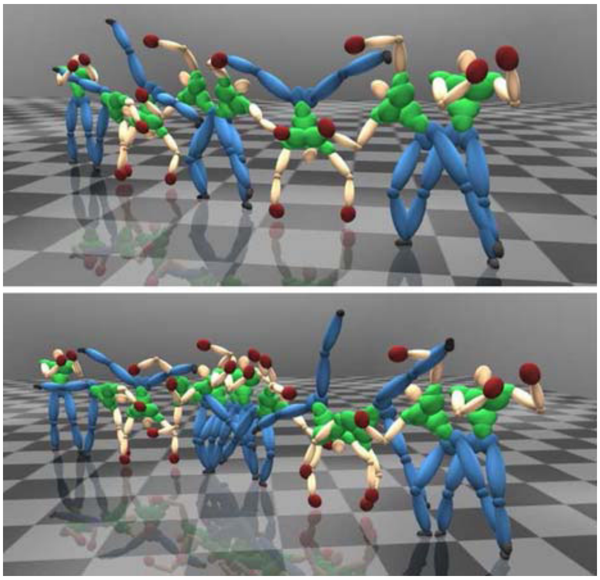


Figure 1. (Upper) An example of concatenating two cartwheel motions with our method. The character makes use of the angular momentum of the first cartwheel to start the second one. (Lower) The same motions concatenated based on the criteria used in Motion Graphs. The character needs to stop and stand still once in order to start the second trial, resulting in an unnatural motion.

of parameters in different dimensions are optimized simultaneously.

Motion blending is a technique to combine two or more existing motions. Motion Graphs[6–8] find similar frames inside a long motion sequence and linearly blend such sequences to switch from one motion to another. The sequences will not be blended unless the joint positions or the joint angles are very similar. Zhao and Safonova[9] propose to improve the connectivity of Motion Graphs by blending motion sequences that are similar in context but kinematically different. Heck and Gleicher[10] apply Motion Graph to generate continuous cartwheel motions. However, they assume the cartwheels start and end in a common pose, and the angular momentum of the first cartwheel is not transferred to the second one. Zordan *et al*.[11] and Arikan *et al*.[12] blend simulated and captured motions to produce effects of being pushed or attacked. While these movements only react to the external perturbation, we are more interested in actively controlling the momentum of the body when conducting successive full-body motions. Majkowska and Faloutsos[13] propose a method to produce flipping motions with multiple rotations from the motion of a single flip. One limitation of their method is that the concatenation of motions can be done only while the character is in the air. No method is provided to concatenate motions that are on the ground, which is often needed for creating a long motion sequence. Thorne[14] proposes to use a tablet device to synthesize a sequence of motions by a character. The short motion clips are concatenated by inserting appropriate keyframe postures in the database. The keyframes need to be predefined for arbitrary combinations.

In summary, there has been little work for concatenating dynamic, full-body motions in which the postures or momenta are dissimilar. For example, when a person conducts two consecutive back flips, the transition movement is completely different from the ending/initial motion of a single back flip. Therefore, we cannot simply blend/concatenate two single back flips to synthesize consecutive two back flips. On the other hand, spacetime optimization requires a huge amount of computation and can get caught in local minima. In this paper, we concatenate these full-body motions to generate smooth and natural resultant motion with limited amount of computation.

Methodology

In this section, the methodology to concatenate two arbitrary full-body motions is explained. We first discuss the typical patterns of angular momentum of whole body motions. Then, we introduce two blending rules and explain how they are used to concatenate motions in an energy efficient way. The discussion is followed by the method to determine the optimal timing for blending. Finally, we explain how the blended motion is scaled to generate a realistic motion with smooth angular momentum.

Typical Angular Momentum Trajectories

We handle motions that require large amount of rotation for the whole body. Such motions result in large angular momentum of the body around the center of mass and can normally be divided into three phases: the *preparatory phase*, the *main phase*, and the *posterior phase* (Figure 2 upper left).

The trajectory of the angular momentum is computed as follows:

$$L(t) = \sum_{j}^{n} \left(m_j \cdot r_j(t) \times p_j(t) \right) \tag{1}$$

DOI: 10.1002/cav

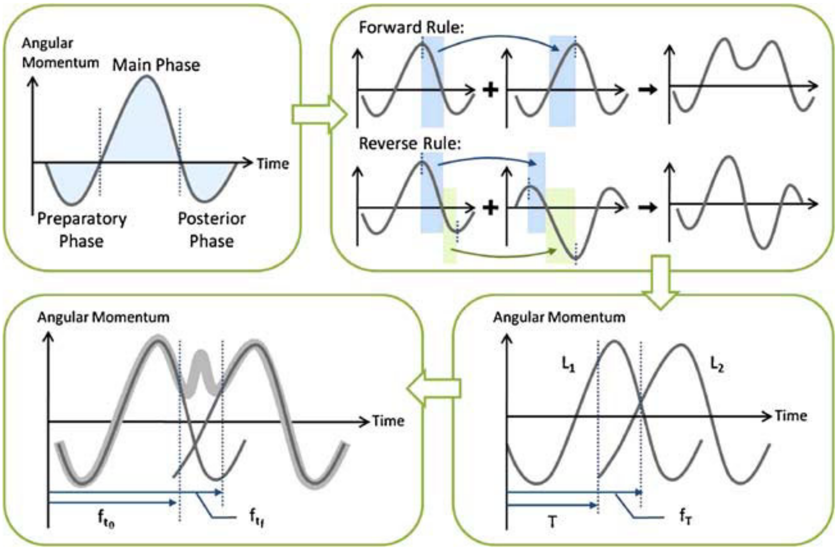


Figure 2. The overview of our system. Note that the trajectories of angular momentum are visualized by one dimensional curves, while they are three dimensional in practice. (Upper left) A typical angular momentum trajectory of a full-body motion. The angular momentum of the main phase is opposite to that of the preparatory phase and the posterior phase. (Upper right) A character can perform two successive motions based on forward rule or reverse rule. The shaded areas show the frame range of the blendable period, and the dotted lines indicate the frame when maximum angular momentum is reached for a phase. (Lower right) The two motions are aligned by adding an offset of time T to the second motion. f_T is the optimal blending center, where the angular momentum difference of the two motions is minimized. (Lower left) When blending the two motions (thin lines) to generate the blended motion (thick line), the blended period between f_{t_0} and f_{t_f} may contain unnatural high angular momentum. We apply motion scaling to smooth such angular momentum.

where n is the total number of segments, m_j is the mass of segment j, $r_j(t)$ is the moment arm vector connecting the mass center of the body and the mass center of segment j, and $p_j(t)$ is the velocity of segment j at frame t. The relationships of these variables are visualized in Figure 3(a).

The *main phase* is the part of the motion when the major objective of the motion is carried out. The angular momentum is maximized in such duration. This phase corresponds to, for example, the duration when a human is flying in the air while doing a back flip, or moving the forearm to the front while punching.

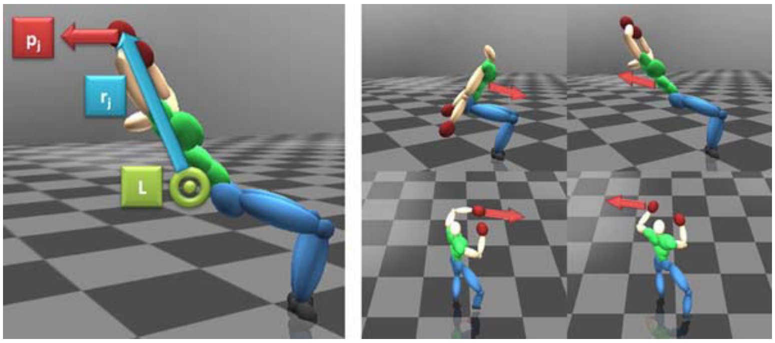


Figure 3. (a) The angular momentum L produced by the right hand, the velocity vector p_j, and the momentum arm vector r_j which connects the center of mass of the body and that of the right hand. (b) The preparatory (upper left) and main phases (upper right) of a flip. The main (lower left) and posterior phases (lower right) of a left hook punch. The red arrows show the direction of the linear momentum for the upper body (upper) and the left hand (lower).

DOI: 10.1002/cav

The *preparatory phase* happens before the main phase and the body moves in the direction opposite to the main phase to rewind the body. It corresponds to the movement of pulling down the arms before a back flip, or the rewinding motion to prepare for a punching motion.

The *posterior phase* happens when the overshot body is brought back to a standing-still, upright posture after the main phase. Again, the angular momentum is opposite to that of the main phase. This corresponds to the standing up motion after the landing of a back flip, or the motion to pull back the forearm towards the body after a punch.

The motions are segmented into these three phases according to the directions of the angular momenta. We first manually pick out a frame in the middle of the main phase. Let us define this frame as t_m and the angular momentum of the body at this frame as $L(t_m)$. Starting from frame t_m, we proceed backward and compute the angular momentum at every frame. The first t that satisfies $L(t) \cdot L(t_m) < 0$ is set as the frame when the preparatory phase ends and the main phase starts. Next we go forward along the time line and search for the first frame when $L(t) \cdot L(t_m) < 0$ is satisfied. This frame is set as the initial frame of the posterior phase.

For some motions, the preparatory/posterior phases may not exist. For instances, some punches start with the main phase and has no preparatory phase, and in some front flips there is no posterior phase (Figure 3b). We first segment the motions into different phases automatically using the above method, and then allow the users to manually fine tune the border frames of the phases.

Transition Rules and Spatial Alignment

We observe that there are two ways to successively conduct two full-body motions in an energy-efficient way. These observations are generalized as the *forward rule* and the *reverse rule* (Figure 2 upper right). Although there may be other ways to conduct two motions efficiently, the two rules we define here are more easy to be generalized and perform consistently when used to concatenate motions.

Forward rule: When humans conduct two successive motions whose angular momenta are in the same direction, they try to keep up the angular momentum of the first motion for starting the second motion. This happens, for example, when a character continuously flips twice in the same direction.

Reverse rule: When humans conduct two successive motions whose directions of the angular momentum are opposite, the main phase or posterior phase of the first motion is used to initiate the second motion. One example is the one-two punch in boxing, in which a left punch is followed by a right punch. In this case, the motion to pull back the left arm after the left punch is used to initiate the right punch.

The two rules are applied to concatenate two full-body motions. In our system, the user first decides whether he/she wants to apply the forward or reverse rule. The reason we allow this user's specification is that sometimes both rules can be applied: think of conducting a back flip after a side flip. The character may apply the forward rule to make use of the angular momentum of the side flip to do the next back flip. Or, it may try to use the latter half of the side flip as a preparatory motion of the next back flip in the opposite direction. Both motions can appear realistic.

Once the rule to apply is decided, the two bodies are spatially aligned based on the axis of angular momentum and the position of the supporting feet. Let us assume motion 1 and motion 2 are to be concatenated sequentially in this order. We first rotate motion 2 around the vertical axis so that the two axes of angular momenta are aligned. This is done by maximizing (if forward rule is to be applied) or minimizing (if reverse rule is to be applied) the dot product of the angular momentum vectors.

$$\theta_2 = \left\{ \arg\max_{\theta} \mid \arg\min_{\theta} \right\} \overline{L_1} \cdot (R_y(\theta)\overline{L_2}) \tag{2}$$

where θ_2 is the angle to rotate motion 2 around the vertical axis to align it with motion 1, $\overline{L_1}$, $\overline{L_2}$ are the average angular momenta of the two motions, and $R_y(\theta)$ is a rotation matrix for θ around the vertical axis (Figure 4). Then, motion 2 is translated so that the center of the feet in the first frame of the blending coincides with that in the corresponding frame in motion 1. After the alignment, in order to evaluate whether the two motions can be blended or not, the torso facing vectors of the two motions are projected onto the ground. If the angle made between these projected vectors is larger than $\frac{\pi}{4}$, we assume the two motions are not suitable to be concatenated.

The Blendable Period

Once the alignment is done and the conditions for concatenating the motions are met, the system proceeds to evaluate when the blending should occur (Figure 2 upper right). The duration we scan to find the best timing to blend the motions is called the *blendable period*,

DOI: 10.1002/cav

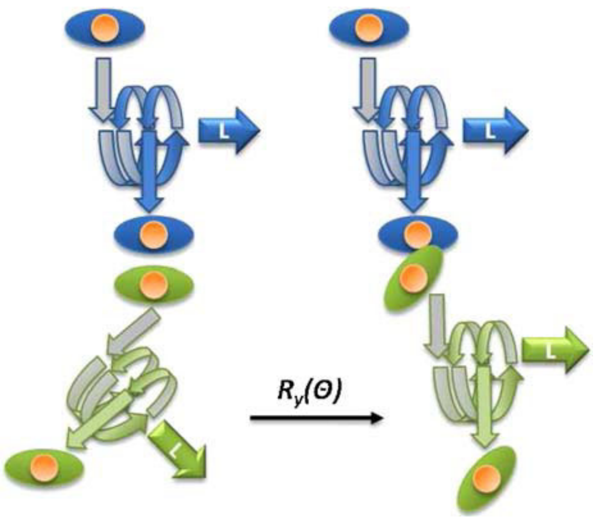


Figure 4. (Viewing from the top) Concatenating two motions by aligning them. The axes of angular momenta are compared and the second motion is oriented around the vertical axis until the dot product of the two angular momenta is maximized (if forward rule is applied) or minimized (if reverse rule is applied).

and is found by examining the trajectory of the angular momentum. Again, let us assume we are to concatenate motions 1 and 2. If the forward rule is to be applied, the latter half of the main phase in motion 1 and the initial half of the main phase in motion 2 become the blendable periods. If the reverse rule is to be applied, there are two cases: either (1) the latter half of the main phase in motion 1 and the latter half of the preparatory phase in motion 2, or (2) the initial half of the posterior phase in motion 1 and the initial half of the main phase in motion 2 become the blendable period. The case to be applied depends on whether any preparatory/posterior phases exist in the two motions. We limit the blendable period to the double support period when both feet are on the ground. In other words, the part of the blendable period that is not in double support is truncated.

The Optimal Timing of Blending

In this section, the optimal timing to align the two motions along the time line is calculated. This is done by analyzing the angular momentum during the blendable period.

Let us assume the motion 2 starts T after motion 1 (Figure 2 lower right). We need to find the optimal T that gives the best alignment of the two motions. This is calculated by minimizing the following objective function:

$$F = F_{\mathrm{dm}} + F_{\mathrm{inertia}} + F_{\mathrm{time}} \tag{3}$$

The first term, F_{dm}, is the square of the angular momentum difference at the blending center f_T:

$$F_{\mathrm{dm}} = \left\| L_1(f_T) - L_2(f_T - T) \right\|^2 \tag{4}$$

where L_1 and L_2 are the trajectories of the angular momentum of the first and second motions. The blending center f_T, which can be computed for every given T, is the frame where the angular momentum difference of the two motions is minimal:

$$f_T = \arg\min_{f} \| L_1(f) - L_2(f - T) \|^2 \tag{5}$$

The second term, F_{inertia}, is used to evaluate the difference of the body moment-of-inertia at f_T:

$$F_{\mathrm{inertia}} = \sum_{j}^{n} \left(m_j \cdot \| r_j^1(f_T) \cdot (r_j^2(f_T - T) - r_j^1(f_T)) \| \right)^2 \tag{6}$$

where n is the total number of joints, m_j is the mass of joint j and r_j^1, r_j^2 are the moment arms of segment j in the first and second motions.

The last term, F_{time}, represents the difference of the angular momentum at the blending center, f_T, and the first frame of the blendable period in the first motion, f_{start}.

$$F_{\mathrm{time}} = \left\| L_1(f_T) - L_1(f_{\mathrm{start}}) \right\|^2 \tag{7}$$

We minimize this term because we want the blending to start as early as possible.

The offset T that minimizes Equation (3) is selected and used to align the two motions. If F is larger than a predefined threshold, the two motions are considered too different to be blended. As the number of possible T is limited by the duration of blendable period in both motions, although we examine all the possible values of T in the range, the optimal value can be found quickly.

Linear Blending

Once the optimal offset T and the corresponding blending center f_T is computed, the two motions are linearly blended:

$$M_{\mathrm{blend}}(f) = \alpha M_1(f_T + f) + (1 - \alpha) M_2(f_T - T + f)$$
$$\forall f \in [f_{t_0},\ f_{t_f}] \tag{8}$$

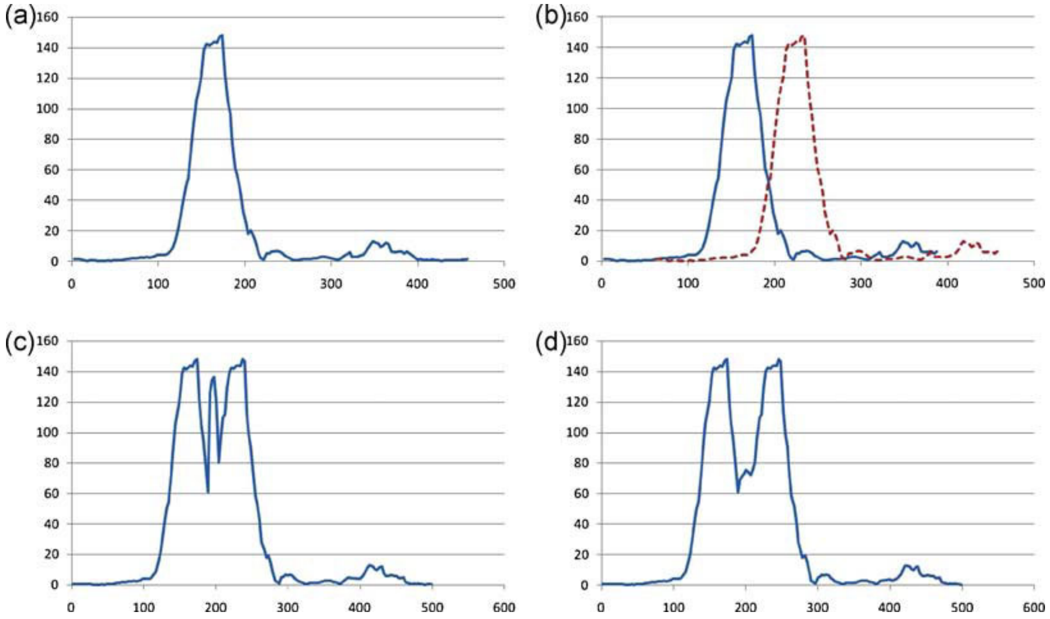


Figure 5. The magnitude of the angular momentum in our experiment for (a) a side flip, (b) two side flip motions aligned to be blended, (c) two consecutive side flips produced by linear blending of motions, and (d) the resultant motion smoothed after scaling the motion in the time domain.

where f_{t_0} and f_{t_f} represent the actual duration for blending the two motion, $\alpha = \frac{1-f}{(f_{t_f} - f_{t_0})}$ is a value between 1 and 0, M_{blend}, M_1 and M_2 are the postures at the blended motion, first motion and second motion, respectively.

f_{t_0} is found by tracing backward in time starting from f_T until either (1) the square difference of angular momentum of the two motions exceeds a predefined threshold, or (2) the blendable period of either motion ends. f_{t_f} is found similarly by tracing forward in time.

Angular Momentum Smoothing

Because the postures to be blended are usually different, blending the two motions may result in non-smooth trajectories of the angular momentum. In this section, we explain the method to smooth the angular momentum by scaling the blended motion along the time line.

In case the posture at f_{t_0} is very different from that at f_{t_f}, the movement during the blended period becomes very fast. This leads to an unnatural motion with large angular momentum (Figure 2 lower left). Here, we scale the motion along the time line. Such scaling operation provides an extra degree of freedom to smooth the angular momentum. The scaling factor is calculated by

$$s = \frac{\left\| L_{\text{blend}}(f_{t_0}) + L_{\text{blend}}(f_{t_f}) \right\|}{2} \Bigg/ \frac{\sum_{f=f_{t_0}}^{f_{t_f}} \left\| L_{\text{blend}}(f) \right\|}{f_{t_f} - f_{t_0}} \qquad (9)$$

where L_{blend} is the angular momentum trajectory of the blended motion. Effectively, the motion during the blended period is scaled such that the average angular momentum during the blended period is consistent with other parts of the motion.

An example of the angular momentum of a side flip is shown in Figure 5(a). A peak in the angular momentum appears in the middle which is produced by the flip of the whole body. We want to produce a character conducting two consecutive side flips by repeating the same motion. This is done by first aligning the side flip motion by minimizing Equation (3). The angular momenta of the two aligned motions are shown in Figure 5(b). When the two motions are simply blended by linear interpolation, the angular momentum appears as shown in Figure 5(c). A large peak appears in the middle of the two trajectories. This is because of the difference of the postures at the timing of blending the two motions. This results in very fast, unnatural movements. In order to solve this problem, we further scale the motions during

DOI: 10.1002/cav

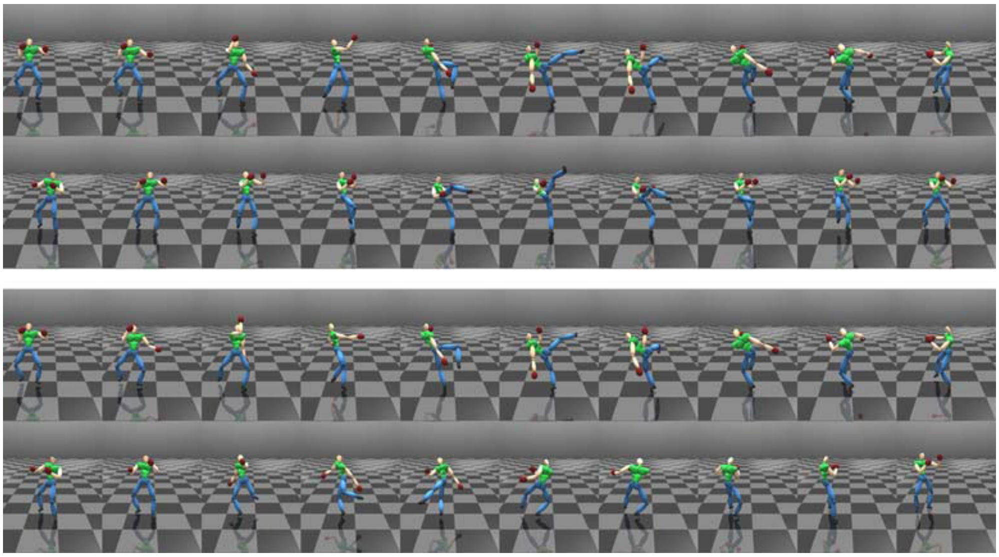


Figure 6. (Upper) A right kick rotating 360° around the vertical axis followed by a right kick toward the front. The body makes use of the angular momentum of the first kick to start the second kick. This motion sequence follows the forward rule. (Lower) A right kick followed by a left kick created by our method. The posterior phase of the right kick is used as the preparatory phase of the left kick. This motion sequence follows the reverse rule.

the blending period. This scaling results in an angular momentum trajectory shown in Figure 5(d).

Experiments

We create two motion databases, one composed of kickboxing motions and the other composed of gymnastic motions. We randomly select two motions, and check whether forward or reverse rule can be applied. The synthesized motions are evaluated by comparing them with motions generated by a Motion Graphs and captured motions. The readers are referred to the supplementary video for further details.

Motion Synthesis Based on Blending Rules

Here we show examples of synthesizing successive kicking motions based on the forward and reverse rules (Figure 6). First, two kicking motions are concatenated based on the forward rule. In the first kick, the character rotates around the vertical axis 360°. In the second kick, the character simply kicks to the front. When they are concatenated by our method, the angular momentum of the first kick is used for the second kick. Next, a sequence based on the reverse rule is created. In this example, the right kick used in the previous example was followed by a left kick in which angular momentum is in the opposite direction. The posterior phase of the first kick is used to initiate the next one. In both examples, our method can synthesize smooth, realistic transitions that cannot be produced by simply concatenating the two motions.

Comparison With Motions Synthesized by Motion Graphs

First, we produce a one-two punch, which is a left straight punch followed by a right straight punch, using individual left and right punches (Figure 7). If we synthesize such motions by using Motion Graphs, the right punch will only start after the left punch is completely finished as the postures in between are not similar enough to be connected. On the other hand, in the motion synthesized by our method, the right punch is initiated during the pulling back motion of the left punch. This is because the latter part of the left punch and the initial part of the right punch shares the common direction of the angular momentum. Our system can successfully find out such periods and overlap them.

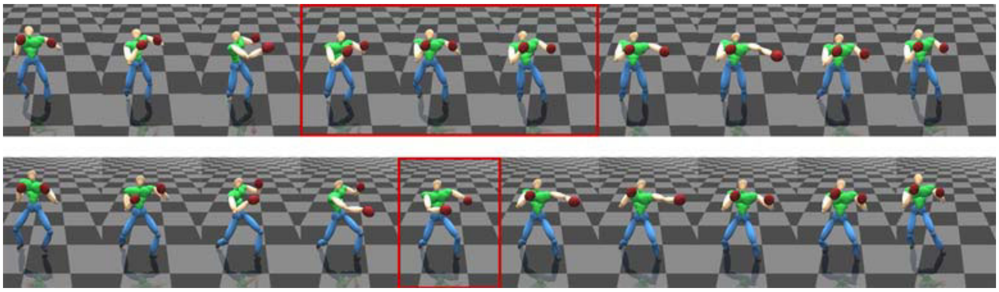


Figure 7. (Upper) A one-two punch created by a criterion used in Motion Graphs. Since similar postures only exist in the very end and the beginning of the two motions, the character needs to stop once in the middle as indicated by the red rectangle. (Lower) The one-two punch created by our method. The pulling back motion of the left punch is used to initiate the right punch as indicated by the red rectangle.

We also prepare an example of concatenating the same cartwheel motion from the gymnastic motion database (Figure 1). If they are concatenated by the criterion based on Motion Graphs, the character needs to stand still once before starting the second motion. On the other hand, in our system, the character can make use of the momentum of the first trial to start the second trial.

Comparison With Motions Performed by a Real Human

The angular momentum trajectories during two consecutive forward flips are analyzed. The dashed red line and the solid blue line in Figure 8 represent such trajectories of the motion conducted by a human performer and that synthesized by our method, respectively. We can observe that the angular momenta of the two motions are very similar. In the trajectory of the real double flip, it can be found that the performer tries to keep up the angular momentum of the first flip for starting the second flip. Our method reproduces such a behavior by blending the main phase of the two motions. The green dotted line represents the angular momentum trajectory of the motion synthesized by a Motion Graph. Since the Motion Graph requires two postures to be similar for concatenation, the characters need to reach the stable standing posture before launching another flip, which results in a long duration of low angular momentum.

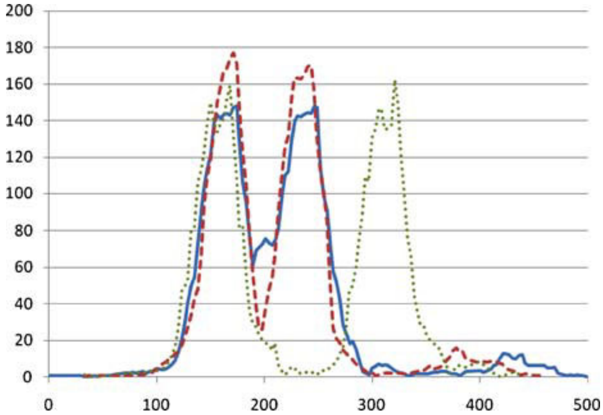


Figure 8. We synthesize a double flip by concatenating two single flips. This figure shows the magnitude of angular momentum for the synthesized motion by our method (solid blue line), the captured motion conducted by human performer (dashed red line), and the synthesized motion by a Motion Graph (dotted green line).

Synthesis of Long Motion Sequences

Finally, a long sequence of boxing motion is produced by concatenating a series of individually captured single attacks. The resultant motions performed by the character are comparable to the shadow boxing motion performed a boxer. Here, we only concatenated motions based on the forward and reverse rules. Therefore, for every motion, part of the angular momentum is used by the next motion.

Foot Sliding

Since the blending is performed in the joint angle space, the transition period generated by our method may suffer from foot sliding problems. In our experiments, we applied the particle inverse kinematics[15] to fix the supporting feet on the floor and prevent the feet from penetrating the floor during blending.

DOI: 10.1002/cav

Computational Costs

The bottle neck of the computation is in finding the optimal timing for blending by minimizing Equation (3). Currently, we simply apply a brute-force search by scanning every frame for finding the optimal T and f_T. Because f_T depends on T, the order of computational cost is in $O(n^2)$, where n is the number of frames in the overlapping blendable periods of the two motions. Even in such a case, with a Core Duo 1.73 GHz CPU and 1 GB of RAM, the computation can be finished within tens of seconds due to the simplicity of Equation (3). One of our future works will be applying numerical optimization techniques such as sequential quadratic programming to find the optimal timing of blending.

Discussions and Conclusion

In this paper, we propose a method to concatenate two full-body motions to produce a single successive motion. We target motions such as ballistic flipping motions and punching and kicking motions during fighting. The process of concatenation was led by the evaluation of the angular momentum. We successfully produced various combinations of full-body motions, which were difficult to be achieved by previous motion blending techniques.

Although we present our algorithm as an off-line process, it can easily be applied to online applications by precomputing the blending parameters for required motion combinations. During run-time, we can blend the motions with these parameters based on the user input. For example, when the user successively presses the punch button twice, instead of showing two individual punches, the system can display a blended double punch. As a result, the users can observe smooth, realistic and quick transition motions in applications such as games.

Our methodology does not strictly follow rules dynamics. For example, we do not conserve the angular momentum of the body during the aerial phase. We also do not model the impulse added to the body when it lands to the ground. Still, our method generates perceptually correct blended motions, which is an important factor for applications such as 3D computer games. One of our future works is to apply dynamics filter to verify the resultant motions such that it can be used in the field of robotics.

References

1. Witkin A, Kass M. Spacetime constraints. *Computer Graphics (Proceedings of SIGGRAPH 88)* 1988; **22**: 159–168.
2. Liu CK, Popović Z. Synthesis of complex dynamic character motion from simple animations. *ACM Transactions on Graphics* 2002; **21**(3): 408–416.
3. Abe Y, Liu CK, Popović Z. Momentum-based parameterization of dynamic character motion. *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2004; 173–182.
4. Sulejmanpasic A, Popović J. Adaptation of performed ballistic motion. *ACM Transactions on Graphics* 2005; **24**(1): 165–179.
5. Fang AC, Pollard NS. Efficient synthesis of physically valid human motion. *ACM Transactions on Graphics* 2003; **22**(3): 417–426.
6. Arikan O, Forsyth DA. Motion generation from examples. *ACM Transactions on Graphics* 2002; **21**(3): 483–490.
7. Lee J, Chai J, Reitsma PSA, Hodgins JK, Pollard NS. Interactive control of avatars animated with human motion data. *ACM Transactions on Graphics* 2002; **21**(3): 491–500.
8. Kovar L, Gleicher M, Pighin F. Motion graphs. *ACM Transactions on Graphics* 2002; **21**(3): 473–482.
9. Zhao L, Safonova A. Achieving good connectivity in motion graphs. In *Proceedings of the 2008 ACM/Eurographics Symposium on Computer Animation*, 2008.
10. Heck R, Gleicher M. Parametric motion graphs. In *I3D'07: Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games*, New York, NY, USA, 2007; 129–136 (ACM).
11. Zordan VB, Majkowska A, Chiu B, Fast M. Dynamic response for motion capture animation. *ACM Transactions on Graphics* 2005; **24**(3): 697–701.
12. Arikan O, Forsyth DA, O'Brien JF. Pushing people around. *Proceedings of 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2005; 59–66.
13. Majkowska A, Faloutsos P. Flipping with physics: motion editing for acrobatics. *Proceedings of 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2007; 79–87.
14. Thorne ME. Motion doodles: a sketch-based interface for character animation. *PhD Thesis*, University of Waterloo, 2001.
15. Jakobsen T. Advanced character physics. In *Game Developers Conference Proceedings*, 2001; 383–401.

Authors' biographies:

Hubert P. H. Shum is currently a final year PhD student in the School of Informatics from the University of

DOI: 10.1002/cav

Edinburgh. He received his MSc (2005) and BEng (2004) degrees in the City University of Hong Kong. His research interests include character animation and machine learning.

Taku Komura is currently a Lecturer in the School of Informatics, University of Edinburgh. Before joining University of Edinburgh on 2006, he worked as an Assistant Professor at City University of Hong Kong between 2002–2006, and as a postdoctoral researcher at RIKEN, Japan between 2000–2002. He received his PhD (2000), MSc (1997), and BSc (1995) in Information Science from the University of Tokyo. His research interests include human motion analysis and synthesis, physically based animation, and real-time computer graphics. His research area covers computer graphics, robotics, and biomechanics.

Pranjul Yadav is a final year undergraduate student doing his graduation from the department of Computer Science and Engineering at Indian Institute of Technology, Guwahati. He was a summer intern at UBI, Portugal in 2007 and at University of Edinburgh in 2008. He works in Geometric Modeling, Fractal Geometry and Computer Animation.

DOI: 10.1002/cav