Pacific Graphics 2008
T. Igarashi, N. Max, and F. Sillion
(Guest Editors)

Volume 27 (2008), Number 7

Smart Motion Synthesis

Masaki Oshita

Kyushu Institute of Technology

Abstract

Creating long motion sequences is a time-consuming task even when motion capture equipment or motion editing tools are used. In this paper, we propose a system for creating a long motion sequence by combining elementary motion clips. The user is asked to first input motions on a timeline. The system then automatically generates a continuous and natural motion. Our system employs four motion synthesis methods: motion transition, motion connection, motion adaptation, and motion composition. Based on the constraints between the feet of the animated character and the ground, and the timing of the input motions, the appropriate method is determined for each pair of overlapped or sequential motions. As the user changes the arrangement of the motion clips, the system interactively changes the output motion. Alternatively, the user can make the system execute an input motion as soon as possible so that it follows the previous motion smoothly. Using our system, users can make use of existing motion clips. Because the entire process is automatic, even novices can easily use our system. A prototype system demonstrates the effectiveness of our approach.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Interaction Techniques; I.3.7 [Computer Graphics]: Animation

1. Introduction

Creating long motion sequences is a time-consuming task even when motion capture equipment or motion editing tools are used. Animators would like to be able to reuse existing motion clips to create one long, complex clip. In theory, by combining elementary short motion clips, an animator could create such a motion sequence, but the actual process is not so simple. Although recent animation production systems include functions for this purpose, to get a continuous and natural motion sequence, animators still need to adjust the appropriate blending ranges, additional constraints, and motion speeds, etc. Because such motion editing is very tedious, animators generally choose to use motion capture even if they have a sufficient amount of elementary motion clips, despite the fact that motion capture also requires a lot of effort to be used in production.

In this paper, we propose a system for creating a long motion sequence by combining a number of elementary motion clips. The user is first asked to input motions on a timeline. The system then automatically generates a continuous and natural motion (Figure 1). Our system employs four motion synthesis methods: motion transition, motion connection, motion adaptation, and motion composition. Based on

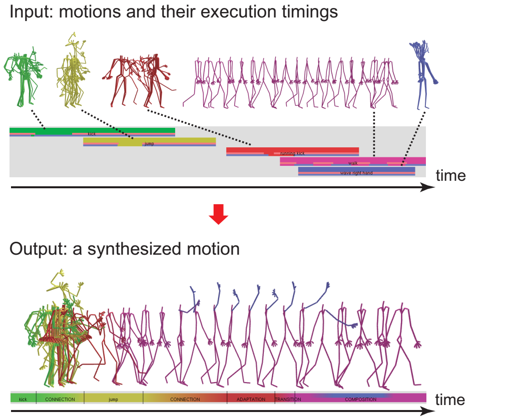

Figure 1: *Example of our motion synthesis system. The user arranges the input motions on the timeline (five motions in this example). The system automatically generates one continuous output motion by synthesizing the given motions. The most appropriate motion synthesis methods are determined based on the constraints of the input motions. The user can interactively edit the output motion by changing the execution timings of input motions on the timeline.*

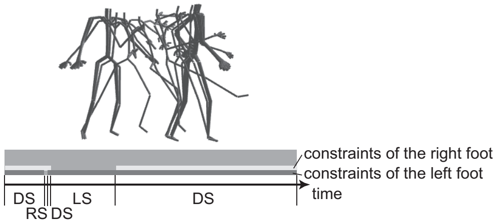

Figure 2: *Example of the support phases and constraints between the feet and ground. In this kick motion, after the left foot is slightly moved (RS), the right foot is moved (LS) and then both feet touch the ground (DS).*

the constraints between the feet of the animated character and the ground, and the timings of the input motions, an appropriate method is determined for each pair of overlapped or sequential motions. As the user changes the arrangement of the input motions, the system interactively changes the output motion. Alternatively, the user can make the system execute an input motion as soon as possible so that it follows the previous motion smoothly. Using our system, users can make use of existing motion clips. Because the entire process is automatic, even novices can easily use our system.

Because maintaining constraints is very important during motion synthesis, our system first analyzes the input motions to detect support phases, which are the constraints between the feet of the animated character and the ground during the motions (Figure 2). Based on the detected support phases, the system determines the appropriate motion synthesis method from the following four methods.

- Motion transition (Figure 3) refers to making a smooth transition from one motion to another. This method is applicable when two motions have the same support phases in which the same leg is moved and the two phases overlap each other on the timeline. During motion transition, the two phases of motion are blended.
- Motion connection (Figure 4) refers to connecting one motion to another. The previous motion is slightly changed to connect to the following motion smoothly. This method is applicable when the previous motion has a phase in which a leg is moved. The following motion does not have to have a leg-moving phase. During motion connection, the phase of the previous motion is blended with the connection posture of the following motion.
- Motion adaptation (Figure 5) refers to adapting one motion so that it follows another. This method is applicable even when the two motions do not have a leg-moving phase. During motion adaptation, the posture of the following motion is transformed so that it maintains the constraints of the previous motion. In addition, when the following motion has a leg-moving phase, the position of the foot is gradually changed from that in the previous motion

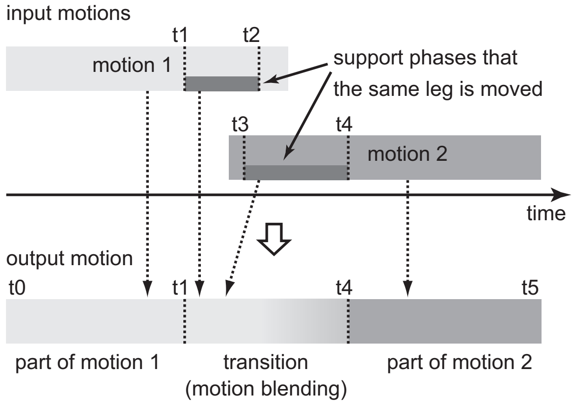

Figure 3: *Motion transition.*

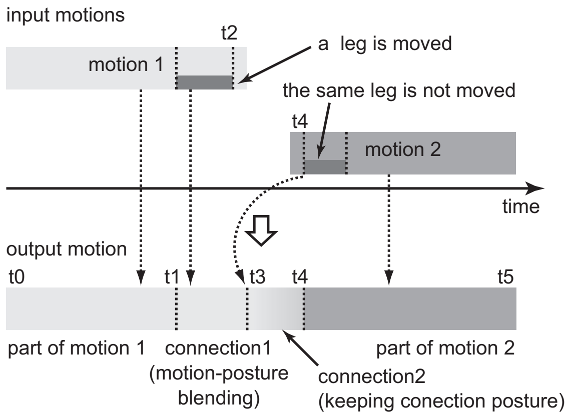

Figure 4: *Motion connection.*

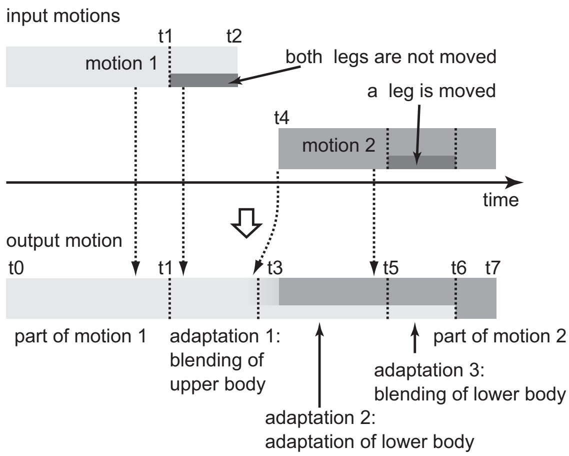

Figure 5: *Motion adaptation.*

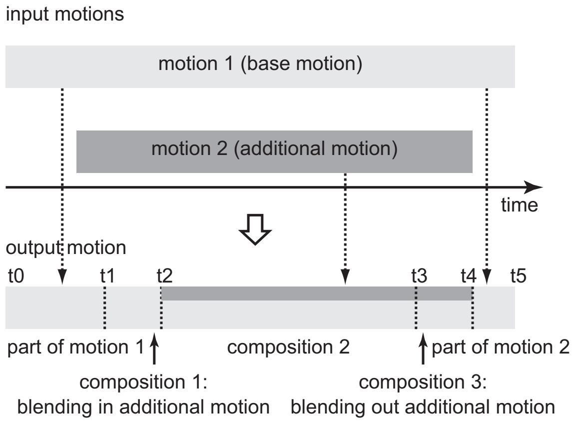

Figure 6: *Motion composition.*

to that in the following motion. Inverse kinematics is used to transform postures based on the foot positions.

- Motion composition (Figure 6) refers to making a composite of two motions by combining the movements of some body parts of one motion with the movements of other body parts of another motion. This method is applicable when two motions overlap each other and one is a motion in which mainly the upper body is moved. We introduce a new method for extracting the effects of the movement on the spine and pelvis caused by movement of the upper body and for applying it to the composite motion to produce natural motion.

In this paper we use the word 'motion synthesis' to represent all of the above methods as a whole. Note that the names of our methods may have been used to represent different concepts in previous studies. Our methods are named based on their purposes, not the techniques used to implement them., such as motion blending, inverse kinematics, and dynamic time warping. We have also developed new techniques to realize the above methods, including computation of body orientation and transformation of root position.

The main contributions of this paper are as follows:

- Three motion synthesis methods for executing two motions sequentially: motion transition, connection, and adaptation. These methods are implemented using conventional techniques such as dynamic time warping, motion blending, and inverse kinematics. Most importantly, we introduce a mechanism that chooses an appropriate method and timings based on the detected support phases.
- A method for executing two motions simultaneously. We introduce a new method for extracting the vibration of body parts caused by the movement of primary body parts in one motion and applying them to another motion.
- A framework for a smart motion synthesis system which generates a continuous output motion from input motions given on the timeline by utilizing the above methods.

The main application of our system is in animation production. However, the system can also be used for controlling characters in computer games or interactive applications.

The rest of this paper is organized as follows. In section 2, we review related works. Section 3 provides an overview of our motion synthesis system. Section 4 explains motion analysis applied to input motions. Sections 5, 6, 7 and 8 describe the methods of motion transition, connection, adaptation, and composition, respectively. In Section 9, we demonstrate our system and discuss the effectiveness of our method. Finally, Section 10 concludes the paper.

2. Related work

This section discusses related works from the viewpoints of motion synthesis framework and motion synthesis methods.

2.1. Motion synthesis framework

Commercial animation systems such as Maya, Softimage, and 3ds Max are widely used by animators. Some of these systems provide motion synthesis functions. MotionBulder from Autodesk [Aut07], for example, provides a smooth blending function. Using MotionBulder, a user can make smooth transitions between motions. However, to create natural motion, the user has to specify the appropriate blend ranges and additional constraints necessary for keeping the character's feet on the ground. Softimage [AT07] provides similar interface and bridge transitions which blend the same cycles of two motions. This method can only be applied to cyclic motions (e.g., walking and running) and the user has to set the elementary motion cycle information manually. These systems provide rather simple synthesis methods and leave users to perform additional editing to get the expected results. Our system, on the other hand, tries to provide the expected results automatically.

Most previous works have focused on a specific type of motion synthesis method; only a few studies have investigated generic motion synthesis systems. Boulic et al. [BBET97] proposed a framework that dynamically composes motions based on a given priority. Perlin and Goldberg [PG96] developed an interactive animation system that synthesizes motions based on manually programmed scripts. Ménardais et al. [MKMA04] proposed a motion synthesis system that synchronizes and blends motions based on foot constraints. A more detailed comparison between [MKMA04] and our method will be discussed in the next subsection. Generally speaking, however, all the above systems require users to set appropriate priorities or weights for input motions for the system to work successfully. In most cases, users simply wish to combine motions to make one continuous motion and do not want to deal with the priorities or weights of individual motions. The only thing a user of our system has to do is simply put input motions on the timeline, and the system takes care of the rest.

Arikan et al. [AFO03] developed an animation system in which a user can specify a combination of desired motion types (e.g., walking, running, or jumping) at each moment on the timeline. The system automatically generates a continuous motion that satisfies the specified constraints by making a composite of pieces in a motion database. However, this system requires a large number of motions in the database, and the user has to label the motions in advance.

2.2. Motion synthesis methods

In this subsection, we compare the individual methods comprising our motion synthesis system with existing methods.

Many researchers have employed motion interpolation techniques [RCB98] [PShKS04] for motion transition [AW00] [GR96]. Using motion interpolation and by changing the blending weighs for sample motions, a user can create motion that gradually changes from one motion to another. However, motion interpolation is limited to the same kinds of motions (e.g., walking and running) and thus cannot be applied to different ones (e.g., walking and kicking). Moreover, the sample motions must be carefully created and synchronized in advance. Therefore, this approach does not fit our purpose.

Researchers have also developed methods for motion transition that use motion blending. Wang and Bodenheimer [WB08] proposed methods to find appropriate blending points and the duration between two motions. However, they did not consider the constraints between the feet and the ground, and this caused unnatural motions such as foot sliding. Other previous studies [MKMA04] [GBT05] have utilized support phases to determine the appropriate blending segments of motions. Glardon et al. [GBT05] blended locomotion cycles of interpolated motions to generate smooth transitions. Ménardais et al. [MKMA04] introduced an algebraic relation to define the bendable support phases of two motions and proposed an algorithm that blends a pair of bendable support phases of a given motion. This approach is similar to ours. However, since they used only motion transition, unnatural results were obtained when there were double support phases in two adjacent motions. Moreover, their method automatically determines the alignment of the given motions. Therefore, users could not specify a timeline of elementary motions. Our system, on the other hand, determines the appropriate method from multiple motion synthesis methods, including the motion transition method, based on a specified timeline of elementary motions.

Rose et al. [RGBC95] investigated the use of dynamics to generate connective motions. Using numerical optimization with an objective function that minimizes the joint torques during a connective motion, a physically correct connective motion is generated. The terminal posture of the previous motion is connected to the initial posture of the following motion. Therefore this method may produce unnatural results when the figure is moving along the postures by a foot sliding during connective motion.

The motion graph [KGP02] [LCR*02] [AF02] is also a technique closely related to motion synthesis. By making transitions between similar postures on given motion sequences and converting all postures and transitions to nodes and edges, respectively, a motion graph is constructed. The motion graph is used to generate continuous motion by connecting short transitions while traversing nodes and edges. When a motion graph is constructed, transitions are intended to connect very similar postures [KGP02] [AF02]. Therefore, transitions cannot be applied to connect any given motions that do not share similar postures. For example, we could use a motion graph to generate connective motion from the two end-postures of two motions [IAF96]. However, it is difficult to find an appropriate path that satisfies all given postures and timelines.

Motion editing techniques similar to our motion connection, adaptation, and insertion techniques are commonly used by animators. Using such techniques, they can edit motion sequences manually using commercial animation systems [Aut07] [AT07]. However, these methods have yet to be formalized or automated. As such, animators must choose the appropriate approach carefully and edit motion sequences step-by-step using keyframe animations, inverse kinematics, constraints and other fundamental motion editing tools. Our method releases animators from such tedious tasks and allows even novice users to combine motion sequences without expert knowledge.

Motion composition is also an important technique. Earlier works [RGBC95] [BBET97] used a straightforward approach that combines the movements of some body parts of one motion with the movements of other body parts of another motion. Although this approach is also supported by commercial animation systems [Aut07] [AT07], it still has two main problems. First, unnatural results may be produced when two segments are not synchronized. Second, because each motion does not affect any other motion, unnatural motion may be produced. For example, when the upper-body waving motion and the lower-body walking motion are made into a composite, the lower-body is not affected by the waving motion, although a small lateral swing synchronized to the movements of the arm would be expected. Heck et al. [HKG06] solved the first problem by finding the appropriate segments of the given motions for a composite based on the synchronization of two motions. Perlin and Goldberg [PG96] tried to solve the second problem by adding random noises to composite motions. However, random noises do not necessarily reflect the original motions. Al-Ghreimil and Hahn [AGH03] extracted the influences of the movements of primary body parts to the movements of other parts by subtracting the corresponding base motion from the given motion, and applied the resulting influence to composite motions. To employ this approach, however, users must prepare

(a) Motion transition

next / prev	NS	RS	LS	DS
NS	○	○	○	×
RS	○	○	×	×
LS	○	×	○	×
DS	×	×	×	×

(b) Motion connection

next / prev	NS	RS	LS	DS
NS	×	○	○	○
RS	×	○	×	○
LS	×	×	○	○
DS	×	×	×	×

(c) Motion adaptation

next / prev	NS	RS	LS	DS
NS	×	×	×	×
RS	×	×	×	×
LS	×	×	×	×
DS	×	○	○	○

Figure 7: *Support phase conditions used to determine the applicability of motion transition, connection, and adaptation.*

base motions for all motions, which is not practical. Our approach, on the other hand, is similar to that of [HKG06] and [AGH03]. Because we extract the influence of primary body part movements by approximating base motions, the user is not required to provide base motions.

3. Smart motion synthesis

3.1. System overview

In this section, we provide an overview of our smart motion synthesis system. As shown in Figure 1, a user is asked to input motions along the timeline. The system then automatically generates a continuous and natural motion.

Our system first analyzes the input motions and detects support phases, which are the constraints between the feet and ground during the motion (Figure 2). Maintaining these constraints is very important during motion synthesis. Our method currently considers only standing biped human-like figures. Therefore, all motion frames are categorized into one of the following support phases: double leg support (DS), right leg support (RS), left leg support (LS), and no support (NS).

Based on the support phases, the system determines which motion synthesis method should be applied to each segment of the output motion. As introduced in Section 1, the system uses four methods: motion transition, motion connection, motion adaptation, and motion composition.

We developed a prototype system in which users can not only add motions along the timeline, but also adjust the temporal arrangement of the motions by dragging individual motion clips using a mouse. As the user changes the arrangement of the motions, the system interactively changes the output motion. Alternatively, the user can make the system decide the execution timing of an input motion automatically by simply double-clicking the input motion. In this case, the motion is executed as soon as possible so that it follows the previous motion smoothly. In this way, both the appropriate timing and synthesis method is determined.

3.2. Automatic selection of motion synthesis method

The output motion is divided into segments, as shown in Figure 1. Each segment uses either part of an input motion or one of the motion synthesis methods to generates part of the output motion based on two input motions. Segmentation is performed based on the timings and support phases of the input motions.

When two input motions are executed sequentially, either motion transition, motion connection, or motion adaptation is used as the intermediate segment for connecting the two motions. The appropriate method is selected in each case in the following priority: Transition > Connection > Adaptation. The applicability of each method is determined based on the support phases. Depending on which of the conditions in Figure 7 a pair of support phases best satisfies, the appropriate method is chosen. These conditions are designed to prevent foot sliding, and more detailed explanations of these conditions are given in the following sections (Sections 5.1, 6.1, and 7.1). The system checks every pair of support phases for two sequential motions to find the pairs that best satisfy the constraints. Since each method is applicable to different types of support phases, in most cases at least one of the above methods is selected.

When two motions are executed simultaneously, that is, when one motion covers another motion entirely on the timeline (Figure 6), motion composition can be used. If one of the two motions can be classified as an additional motion (see Section 8), then motion composition is applied. Otherwise, the two motions are executed sequentially and the above process for creating sequential motion is applied. By repeating these processes from the first to the last motion, segmentation of the output motion can be determined based on the selected methods and support phases. Although the above processes consider only a pair of motions at a time, our methods can also be applied to multiple motions by repeating above process on synthesized motions. However, the execution of more than two motions simultaneously does not happen in normal situations.

3.3. Automatic adjustment of execution timing

After some motions are specified by the user, the system can determine the appropriate execution timings of the motions. The execution timing of subsequent motions is determined so that the middle of the support phase of the following motion matches the middle of the corresponding support phase of the previous motion. The determination of appropriate synthesis methods and execution timings is also solved one-by-one, from the first to last motion.

4. Motion analysis

A user of our system does not have to manually provide any additional information with respect to input motions. Instead, to apply our motion synthesis method, any additional necessary information, such as support phases, body orientations, and base foot information, are automatically detected for each input motion.

4.1. Detecting support phases

We use a common approach [MKMA04] to detect support phases during motion. If the velocity of a foot is lower than a threshold and its height is still relatively close to the ground, the foot is considered to be supporting the body. At each frame during a motion, the constraints of both feet are detected and the support phase is labeled.

4.2. Detecting the base foot

Each input motion has its own coordinates and initial positions and orientations. When two motions are executed sequentially, the second motion should be aligned with the first motion. To do this, our method aligns the positions of the base feet and body orientations of the two motions. The second motion is moved so that the position of the base foot of the first motion at the end of the motion matches the position of the same foot of the second motion at its beginning. In addition, the second motion is rotated so that the body orientations of both motions match.

Although the simpler option of using the pelvis position and orientation instead of the base foot and body orientation exists, this can cause problems, as shown in Figure 8. When the positions of the feet are different between two motions, the orientations of the pelvis are also different, and thus the orientation of the second motion will need to be changed when the two motions are connected. In addition, to maintain the constraints between the supporting foot and the ground, the two motions should be connected based on the foot positions instead of the pelvis positions.

We consider a base foot to be the foot that mainly supports the body. As the names imply, during a right leg support phase (RS) the right foot is the base foot, and during a left leg support phase (LS) the left foot is the base foot. During a no support phase, however, there is no base foot. This is not a problem, however, because we normally have no need to connect motions between two no support phases unless both motions are comprised only of no-support phases (e.g., flying motions). In such cases, we use the pelvis position to align the two motions. Conversely, in cases where there are double support phases, the foot that is closer to the center of mass is chosen as the base foot.

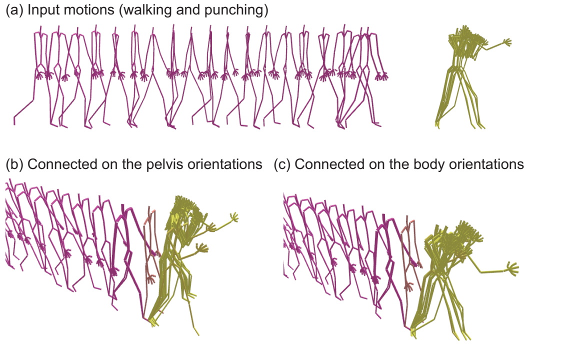

Figure 8: *Example of body orientations and motion connection based on them. If two motions are connected based on the pelvis directions (b), the following motion goes into wrong direction. This is solved by using body orientations that are computed from the movement of the pelvis (c).*

4.3. Computing body orientation

We assume that a figure basically moves in the direction of body orientation. Therefore, the body orientation at the starting point of a motion is computed from the vector of the initial pelvis position to a subsequent position of the pelvis after it has moved a certain distance. However, in some motions a figure may move backward (e.g., backward walking) or laterally (e.g., lateral step or jump). Therefore, we assume that a figure can moves forward, backward, right, or left at the beginning of a motion. Among the four possible movement vectors computed by rotating the movement direction 0, 90, 180, and 270 degrees, the vector whose angle (dot product) in relation to pelvis orientation vector is smallest is used as the body orientation vector. The body orientation at the end of a motion is computed in the same way.

5. Motion transition

In the following sections, we explain individual motion synthesis methods, beginning with motion transition. Motion transition is applied when the corresponding support phases of two input motions involve movement of the same leg.

5.1. Conditions

Motion transition is applied if the two support phases of two sequential input motions satisfy the following conditions.

First, the support phases must satisfy Figure 7 (a). Although [MKMA04] used similar conditions, they applied motion blending to a wider range of phases, including double support phases. We do not allow such blending of double support phases, because it can cause foot sliding and produce unnatural motion. We blend two motion segments only when the same foot, or both feet, are moving in both motions.

Second, the two segments that satisfy the first condition should be sufficiently close on the timeline. If they are too far apart, the resulting transition may be too long and slow. Currently we determine the timings of two segments using the following equation,

$$s_{min} < \frac{t4-t1}{t2-t1} < s_{max}, s_{min} < \frac{t4-t1}{t4-t3} < s_{max}, \qquad (1)$$

where t1 and t2 are the beginning and ending times, respectively, of the support phase of the previous motion, as shown in Figure 3. t3 and t4 are the corresponding times of the support phase of the next motion. The transition segment begins at t1 and ends at t4. The above conditions ensure that the time scaling of both original segments is kept between $s_{min}$and s_{max}, which are given as parameters to the system. These parameters can be tuned by the user. In our experiments we use $s_{min} = 0.7$and $s_{max} = 1.5$.

If a pair of support phases satisfies the above conditions, the pair is used for motion transition. If there is more than one pair that satisfies the conditions, the latest one on the timeline is used. As explained in Section 3.2, if the system cannot find any support phases that satisfy the conditions, another method such as motion connection (Section 6) or adaptation (Section 7) is applied instead of motion transition.

5.2. Implementation

A motion transition segment starts at the beginning of the support phase of the previous motion and ends at the end of the support phase of the following motion (t1 and t4 in Figure 3). During motion transition, the corresponding segments of the input motions are blended so that the blending motion transits from the previous motion to the following motion. Before the transition segment, a part of the previous motion (before the support phase for motion transition) is executed. Similarly, after the transition segment, a part of the following motion is executed.

To make a transition over two segments, the segments are time-warped and blended. We use a common method. First, each motion segment is expanded or shrunken using the time-warping technique [RCB98] [PShKS04] so that it fits the transition segment. On each frame the postures of the two segments are blended with a weight function that linearly varies from 0.0 to 1.0. Based on the resulting two time-warped motion segments, their joint rotations, root positions, and orientations are blended [RCB98] [PShKS04].

6. Motion connection

Motion connection refers to changing the previous motion so that it connects to the beginning point of the following motion. The differences between motion connection and transition are as follows. First, motion connection allows for connection to a double support phase, which is not allowed in motion transition. Second, motion-posture blending is used in motion connection instead of motion blending.

6.1. Conditions

The conditions between the two support phases of two motions when motion connection is allowed are shown in Figure 7 (b). There is no temporal condition on the support phases for motion connection provided except the support phase of the previous motion must end before the support phase of the following motion ends. This means that motion connection can be applied even when two motions are separated on the timeline. Note that motion connection can be applied to a wider range of motion segments compared to motion transition. However, if both motion transition and connection are applicable, motion transition is applied because it generally produces smoother transitions.

6.2. Implementation

During motion connection, the motion segment of the previous motion (t1 to t2 in Figure 4) is blended with the connection posture (t4) of the following motion. However, if we simply use the original motion segment for motion connection, the resulting motion may be too long and slow. Therefore, we first determine the terminal timing of motion blending (t3). Once the terminal timing point is reached, the figure maintains the connection posture until the following motion starts (t3 to t4). However, such stillness may look unnatural since people usually do not stay still for very long. If a user does not like such prolonged stillness, they can change the timings of the input motions so that the following motion starts earlier.

For the connection posture, the initial posture of the support phase of the following motion (t4 in Figure 4) is used. However, if the terminal time of the support phase of the previous motion (t2 in Figure 4) is later than that of the following motion (t4), the posture at t2 is used for the connection posture. The terminal time of the previous motion (t2) is also used as the terminal time of the connection segment (t3) so that the duration of the original motion is maintained. Once the connection posture and the terminal time of motion connection are determined, the motion is computed by blending the time-warped original motion and the connection posture.

7. Motion adaptation

Motion transition or connection is applied only when the previous motion includes a foot-moving-phase. If both feet

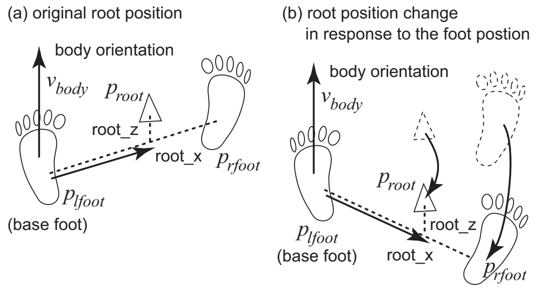

Figure 9: *Transformation of the horizontal position of the root (pelvis) in response to the changes in foot positions.*

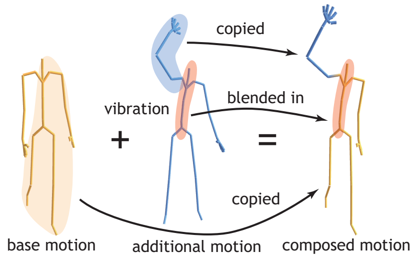

Figure 10: *Motion composition.*

keep contact with the ground, it is difficult to transit or connect to the following motion, because the figure cannot move its legs. In such cases, we apply motion adaptation, which adapts the foot positions of the following motion to the previous motion.

7.1. Conditions

The conditions of the support phases for motion adaptation are shown in Figure 7 (c). Note that motion adaptation is now applicable when the previous motion does not have any leg moving phases (NS, LS, RS) and when motion transition and connection are not applicable.

7.2. Implementation

First, the upper body postures of the previous motion are changed so that it can connect to the following motion (between t1 and t2 in Figure 5). We use the same method here as in motion connection, except the lower body posture of the original motion is not changed because both feet are constrained on the ground. In order to compute the posture at t3 in Figure 5, the lower body posture of the following motion at t4 is transformed so that its foot positions match the posture at t2 in the previous motion. However, if we simply change the foot positions while keeping the position of the root (pelvis) the same relative to the base foot, the position of the root may be unnatural. Therefore, the root position is also transformed in response to the transformation of the foot positions, as shown in Figure 9. We represent the root position before transformation P_{root} using parameter ($root\_x$, $root\_z$) as follows:

$$P_{root} = root\_x\left(P_{rfoot} - P_{lfoot}\right) + root\_z\, V_{body}, \qquad (2)$$

where $P_{rfoot}, P_{lfoot}, V_{body}$represent the foot positions and body orientation vector. The transformed position of the root is computed based on the transformed foot positions.

At the beginning of the following motion (t3 in Figure 5), the foot positions are constrained. At each frame of the following motion, based on the foot and pelvis positions, the posture of the original motion is changed using inverse kinematics. If the following motion has a foot-moving- phase, the foot position is blended from the constrained position to the original position (t5 to t6 in Figure 5).

8. Motion composition

Motion composition refers to the combining of the movements of some body parts of one motion with the movements of other body parts of another motion. For example, as shown in Figure 10, based on a waving motion and a walking motion, motion composition generates a waving-while-walking motion. Here, 'additional motion' refers to an input motion in which the movement of some upper body pars are used as additional motion in the composition (waving motion in Figure 10), while 'base motion' refers to another input motion in which the movements of the other body parts used as base motion (walking motion in Figure 10). In theory, there exist many more complex combinations of body parts from multiple input motions. However, we consider only the abovementioned combination types because the combining of upper-body and lower-body motions is very common [HKG06] [AGH03]. Our method chooses which body parts of an additional motion should be used based on the motion of only the right arm, left arm, or the entire upper body.

If we simply combined the movements of different body parts from different motions, the resulting motion would be unnatural, because the different moving body parts do not affect each other. For example, when the right arm movement of a waving motion is combined with a walking motion, as shown in Figure 10, in addition to the right arm movement of the waving motion, the spine and pelvis should also be moved in coordination with the movement of the right arm. However, if we simply used the spine and pelvis motion of the waving motion instead of the walking motion, the correct movement of the walking motion would be lost and the combined motion would not be the expected motion. Therefore, we extract only the vibration elements on the spine and pelvis caused by the additional motion related to arm movement and add them to the movement of the base motion.

8.1. Analysis of additional motion

When an input motion is given to the system, the system analyzes it whether it can be used as an additional motion and which body parts of the upper body should be used. In addition, vibration elements of the motion are extracted.

In order to determine the body parts to be used, variations of the joint angles of the right and left shoulders are computed. If the difference between the minimum and maximum angles exceeds a threshold, then the movement of the arm is used. If both arms are used, the spine including the neck is also used. When one arm is used as the primary body part, the vibrations of the angles of the spine and the position of the pelvis caused by the arm movements are extracted. If the whole upper body (both arms and spine) is used as the primary body parts, the vibrations of the position of the pelvis are extracted. In addition, based on the variations of the shoulder angles, the beginning and end timings of the movement of the primary body parts are also determined (t2 and t3 in Figure 6).

To extract the vibration elements, we estimate the joint and special trajectories of the spine and pelvis when the primary upper body parts are not moving. We assume that the trajectories should be smooth if the primary body parts are not moving because there would be no influence from non-moving body parts. Therefore we use a spline curve that fits the joint trajectories. The difference between the original trajectories and the spline trajectories is then used to add the small vibrations synchronized to the movements of the upper body.

$$\theta_v(t) = \theta(t) - \theta_b(t), \tag{3}$$

where $\theta(t)$ is a spine angle or root position of the original motion, and $\theta_v(t)$ is the extracted vibration. For $\theta_b(t)$, we use the Hermite curve, which is a continuous curve computed based on the values and velocities of the beginning and end of the movement (t2 and t3 in Figure 6). Vibrations are computed for each degree-of-freedom (DOF) of the spine angles and pelvis position.

8.2. Implementation

In motion composition, the base motion and the additional motion are synchronized based on the movements of additional body parts in both motions. The movements of the additional body parts in the additional motion are then blended with the base motion using motion-posture blending. The small vibrations $\theta_v(t)$ are applied to the spine angles and the pelvis position of the base motion.

9. Experiments and discussion

We implemented the proposed framework and motion synthesis methods. We tested our system using various commercially available motion data. Some of the resulting animations are shown in the accompanying videos.

Since our method is simple, motion synthesis can be computed very quickly. Once the segmentation of the output motion is computed, the posture of each frame is computed on the fly. The system does not have to store all frames of synthesized motion along with the input motions.

To evaluate the effectiveness of our system, we compared our system with MotionBuilder from Autodesk [Aut07]. We combined the same motions using both systems. The accompanying videos contain three kinds of examples: motion transition (walking and backward walking), motion connection (jump and walking) and motion adaptation (kicking and running kick). Regarding the required time and effort, our system was much easier to use than MotionBuilder. While MotionBuilder also provides a user interface in which a user can arrange input motions on the timeline and automatically blend overlapped motions, in order to create natural motion, the user must also, through trial an error, specify the trimming of input motions along with the spatial arrangement of the motions and any additional constraints or keyframes to ensure the feet stay on the ground. It required 10 to 20 minutes to create each motion with MotionBuilder, while it required only less than one minute with our system. Regarding the quality of the generated motions, our system basically generated motions similar to those generated by MotionBuilder. However, our system sometimes generated unnatural motions, especially when the motion connection method was used. Since motion connection methods use motion-posture blending, unnatural motions are sometimes created when the blending pose is different from the previous motion, or when the pose is an unstable posture, as can be seen in the example video. This problem can be solved by finding the appropriate blending poses or by introducing a dynamic as explained below.

The motion blending method used in our implementation is very simple. Corresponding motion segments in the input motions are linearly blended. Automatic execution timings are also simply computed from the durations of the support phases of the input motions, as explained in Section 3.3. To improve the quality of output motions, evaluations of blended motions should be introduced to determine the appropriate blending timings, time-warping, and blending weights [WB08]. Further, evaluation of the effectiveness and applicable range of the motion synthesis methods will need to be pursued in future studies.

Currently the use of dynamics is not introduced in our method. Because all input motions are comprised of motion-captured data and physically correct, the synthesized motions are also expected to be physically correct. The most important motion constraint is the constraint related to the positions of the feet and ground; this constraint is ensured in our method. However, because we simply blend postures during motion transition, connection, and adaptation, when the postures of motions are different, the resulting blended motions may be unnatural. To deal with this issue, we suggest the future introduction of a motion filter [FP03] [TySK02] to our system that changes the output motion to better satisfy physics-based constraints such as balancing and energy minimization, etc.

Similar to other animation software, our system can also edit the motions of multiple characters simultaneously. The motions of multiple characters are synthesized independently. However, when multiple characters interact with each other, their motions should be synchronized. If the timing of one motion is changed, then the timings of the corresponding motions of other characters should also be changed. Because the timings of input motions can be freely changed in our system, such temporal constraint issues can be handled without any problem.

Based on the above experiments using various motions, we have demonstrated the effectiveness of our system. We plan to make our system public and ask both professional and non-professional users to try our system in production.

10. Conclusion

In this paper we proposed a system for making long motion sequences by combining elementary motion clips. Using our system, even novice users can make use of existing motion clips. Our future work will include the integration of more advanced techniques for motion blending, inverse kinematics, and dynamics filters to improve the quality of output motions as well as the further development of a practical system.

References

[AF02] ARIKAN O., FORSYTH D. A.: Interactive motion generation from examples. *ACM TOG 21*, 3 (2002), 483–490.

[AFO03] ARIKAN O., FORSYTH D. A., O'BRIEN J. F.: Motion synthesis from annotations. *ACM TOG 22*, 3 (2003), 402–408.

[AGH03] AL-GHREIMIL N., HAHN J. K.: Combined partial motion clips. In *The 11th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2003* (2003).

[AT07] AVID TECHNOLOGY I.: *Softimage XSI 6.0*. 2007.

[Aut07] AUTODESK I.: *Autodesk MotionBuilder 7.5*. 2007.

[AW00] ASHRAF G., WONG K. C.: Generating consistent motion transition via decoupled framespace interpolation. *ACM TOG 19*, 3 (2000), 447–456.

[BBET97] BOULIC R., BECHEIRAZ P., EMERING L., THALMANN D.: Integration of motion control techniques for virtual human and avatar real-time animation. In *Proceedings of the ACM symposium on Virtual Reality Software and Technology (VRST '97)* (1997), pp. 111–118.

[FP03] FANG A. C., POLLARD N. S.: Efficient synthesis of physically valid human motion. *ACM TOG 22*, 3 (2003), 417–426.

[GBT05] GLARDON P., BOULIC R., THALMANN D.: On-line adapted transition between locomotion and jump. In *Computer Graphics International 2005* (2005), pp. 44–50.

[GR96] GUO S., ROBERGE J.: A high-level control mechanism for human locomotion based on parametric frame space interpolation. In *Eurographics workshop on Computer animation and simulation '96* (1996), pp. 95–107.

[HKG06] HECK R., KOVAR L., GLEICHER M.: Splicing upper-body actions with locomotion. *Computer Graphics Forum 25*, 3 (2006), 459–466.

[IAF96] IKEMOTO L., ARIKAN O., FORSYTH D.: Quick transitions with cached multi-way blends. In *Symposium on Interactive 3D Graphics and Games 2007* (1996), pp. 145–151.

[KGP02] KOVAR L., GLEICHER M., PIGHIN F.: Motion graphs. *ACM TOG 21*, 3 (2002), 473–482.

[LCR*02] LEE J., CHAI J., REITSMA P. S. A., HODGINS J. K., POLLARD N. S.: Interactive control of avatars animated with human motion data. *ACM TOG 21*, 3 (2002), 491–500.

[MKMA04] MÉNARDAIS S., KULPA R., MULTON F., ARNALDI B.: Synchronization for dynamic blending of motions. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2004* (2004), pp. 325–335.

[PG96] PERLIN K., GOLDBERG A.: Improv: A system for scripting interactive actors in virtual worlds. In *SIGGRAPH '96 Proceedings* (1996), pp. 205–216.

[PShKS04] PARK S. I., SHIN H. J., HOON KIM T., SHIN S. Y.: On-line motion blending for real-time locomotion generation. *Computer Animation and Virtual Worlds 15*, 3 (2004), 125–138.

[RCB98] ROSE C., COHEN M. F., BODENHEIMER B.: Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics and Applications 18*, 5 (1998), 32–40.

[RGBC95] ROSE C., GUENTER B., BODENHEIMER B., COHEN M. F.: Efficient generation of motion transitions using spacetime constraints. In *SIGGRAPH '95 Proceedings* (1995), pp. 147–154.

[TySK02] TAK S., YOUNG SONG O., KO H.-S.: Spacetime sweeping: An interactive dynamic constraints solver. In *Computer Animation 2002* (2002), pp. 261–270.

[WB08] WANG J., BODENHEIMER B.: Synthesis and evaluation of linear motion transitions. *ACM TOG 27*, 1 (2008).