EUROGRAPHICS 2006 / E. Gröller and L. Szirmay-Kalos
(Guest Editors)

Volume 25 (*2006*), *Number 3*

Splicing Upper-Body Actions with Locomotion

Rachel Heck
University of Wisconsin-Madison

Lucas Kovar
Industrial Light and Magic

Michael Gleicher
University of Wisconsin-Madison

Abstract

This paper presents a simple and efficient technique for synthesizing high-fidelity motions by attaching, or splicing, *the upper-body action of one motion example to the lower-body locomotion of another. Existing splicing algorithms do little more than copy degrees of freedom (DOFs) from one motion onto another. This naïve DOF replacement can produce unrealistic results because it ignores both physical and stylistic correlations between various joints in the body. Our approach uses spatial and temporal relationships found within the example motions to retain the overall posture of the upper-body action while adding secondary motion details appropriate to the timing and configuration of the lower body. By decoupling upper-body action from lower-body locomotion, our motion synthesis technique allows example motions to be captured independently and later combined to create new natural looking motions.*

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

1. Introduction

Often the problem of character control is divided into locomotion and action. For example, a character can carry a box, punch, carry a suitcase, or wave, independent of whether it is walking straight, walking in a curve, walking up some stairs, jogging, or just moving from one foot to the other. Applications that allow many different actions and kinds of locomotion, such as video games, must be able to generate motions for every possible *combination* of locomotion and action. Synthesizing these motions using current example-based methods requires capturing examples of the character performing all possible combinations. If n locomotion examples are needed and there are m actions that the character can perform, nm examples would need to be collected all together. It would be attractive if a character's upper-body action and lower-body locomotion could be captured independently, requiring only $n+m$ examples. However, this independent treatment requires a method for *splicing* the separate pieces together. For example, to create a motion of someone stepping up onto a platform while carrying a cup, an application could splice the lower body of a character stepping up onto a platform with the upper body of someone carrying a cup. Unfortunately, existing methods for character splicing do not produce motions of the same quality as the original example motions.

In this paper, we propose a method for splicing a character's upper body action onto its lower body locomotion in a manner that preserves the fidelity of the original source motions. The key challenge addressed by our method is that the motion of the upper body and lower body can be be highly correlated – for example when a person walks, the swing phase of the right arm is tightly coupled to that of the left leg – and these correlations are an important part of many motions [PB00]. Natural correlations not only stem from

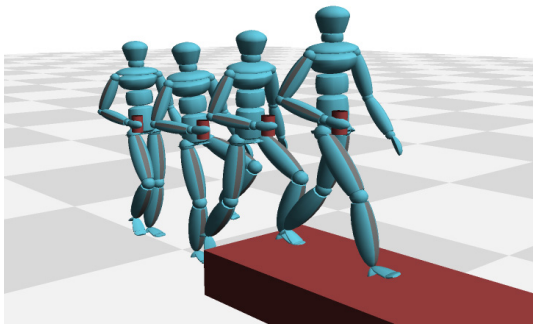

Figure 1: *A motion generated by splicing the upper body of a person carrying a cup with the lower body of a person stepping up onto a platform.*

physical needs, such as balance preservation, but also from harder-to-characterize properties associated with the stylistic form of human motion. In light of the complicated and subtle form of the relationships we wish to preserve, we adopt an example-based approach; our method produces natural motions by identifying and enforcing temporal and spatial relationships between different parts of the body. Our simple and efficient method synthesizes spliced motions that appropriately preserve correlations; an example is provided in Figure 1.

Our technique is motivated by two key observations. First, a captured locomotion example encodes temporal and spatial relationships between the upper body and lower body that can be used for splicing; in other words, if we have one example of a character performing an action while moving around, we can transfer that action onto a different example of a character moving around. For this reason, we require that all captured examples of upper-body actions be performed while locomoting. Secondly, changes made within the upper body or the lower body of a motion can affect a viewer's perception of the action or locomotion. For example, when carrying a heavy box, a person's arms move little relative to the torso, and any editing operation that causes additional arm movement will incorrectly make the box appear lighter. This leads us to restrict the kinds of changes that can be made during splicing. Specifically, we only allow a temporal alignment and a per-frame rigid transformation at the attachment point. This helps retain the meaning of the upper-body and lower-body motions while allowing better correlation between the two pieces.

To illustrate our splicing technique, we give a concrete example. Imagine that we have motions of a character walking in a curved path and of a character walking in a straight line while carrying a heavy box. Using motion splicing, one can generate a motion of the character carrying a heavy box down a curved path. This motion will display many essential characteristics of the original box carrying motion. For instance, the arms will stay the same distance apart throughout the motion, and the upper body will lean back slightly because of the weight of the box. The motion will also exhibit characteristics that are a direct result of the correlations between the upper body and lower body. For example, the shoulders will rotate and bob slightly with the stride of the lower body, and the entire motion will lean in towards the turn. In particular, note that we can produce motions with correct physical effects without explicitly modeling physics; we produce motions with physical effects because the original example motions encode them.

Our focus on the practical and important problem of splicing upper-body actions with lower-body locomotions rather than the more general problem of transferring any subset of the body from one motion to another has allowed us to develop a fast, simple, and reliable algorithm. See Section 5 for more discussion on the general splicing problem.

The rest of this paper is organized as follows. First, Section 2 discusses related work. Section 3 then provides details of our motion splicing algorithm, and Section 4 presents the results of some experiments. Finally, we conclude in Section 5 with a general discussion of the technique, its advantages, and its disadvantages.

2. Related Work

The general strategy of combining existing motions to synthesize new ones has been studied by the animation community for many years. Previous research has primarily focused on two techniques: transitioning and interpolation. A transition is a segment of motion that seamlessly attaches two motions to form a single, longer motion. Earlier work focused on generating individual transitions [RGBC96], and more recent work has built graph structures that use transitions to connect many short segments of motion data into arbitrarily long streams of motion [AF02, KGP02, LCR*02, AFO03, KPS03]. An interpolation is a synthesized motion that is in between a set of example motions [BW95]. Interpolations are particularly useful for constructing parameterized motions where high-level motion properties (such as the target location of a reach or the style of walk) are tied to interpolation weights [WH97, RCB98, RSC01, PSS02, KG03]. While transitions and interpolations are usually constructed for all degrees of freedom (DOFs) simultaneously, in principle they can be applied to subsets of a character's DOFs to achieve results similar to motion splicing. For example, Rose et al [RGBC96] generated transitions from waving to walking and from saluting to walking for the DOFs of the right arm, creating motions where a character walks while waving or saluting. While this can sometimes yield reasonable results, in general its failure to respect correlations with other DOFs can produce unrealistic motions. Our algorithm seeks to splice motions while explicitly preserving the correlations between the upper and lower halves of the body.

To our knowledge, splicing motions together is common practice in the video game industry (for example, to make a character hold a gun while running). While we are aware of no published descriptions of the methods employed, our experience suggests that splicing is usually accomplished by simply swapping data between two motion, henceforth referred to as naïve DOF replacement. Unfortunately, unless the original motions are carefully tailored, naïve DOF replacement will often produce unnatural results because it does not respect the natural correlations between the upper body and lower body. Even in simple cases, such as transferring the upper body of a walking motion to a different walking motion, naïve DOF replacement can cause disturbing visual artifacts. Naïve DOF replacement was also used by Perlin [Per95] and Perlin and Goldberg [PG96] to layer procedurally-defined motions into composite actions. However, the goal of this work was to produce scriptable characters that could flexibly interact with each other and with

human participants, rather than to synthesize high-fidelity motion. We seek to splice captured examples in a way that preserves the realism of the original data. More recently, Ikemoto and Forsyth used naïve DOF replacement to transplant pieces of motion [IF04]. Using a classifier to evaluate the results, they added "human" looking motions to a motion database. Ikemoto and Forsyth acknowledge the failures of naïve DOF replacement in their work by using a classifier to guarantee that the results look correct. Our work seeks to perform splicing in a more sophisticated manner, so as to increase the likelihood that it produces a realistic result.

To build a representation of how the DOFs of a set of "base" actions (such as walking straight forward, standing, or sitting) are affected by the addition of an auxiliary action (such as throwing), Al-Ghreimil and Hahn [AGH03] captured examples of each base action with and without the auxiliary action and computed the differences. Joint trajectories that varied little in these difference motions were replaced with an average, yielding a more compact encoding. The result was a set of joint trajectories that could be added on to new instances of a known base motion, e.g., a different actor walking straight forward. It is unclear how well this method generalized to different base motions, such as turning or walking in a curve.

Ashraf and Wong specifically addressed the correlations of the upper body and lower body of a human by using frame space interpolation to blend the upper body and lower body independently [AW00]. Like us, they observe that the lower body of the character often is devoted to locomotion and the upper body is devoted to articulated activities. Ashraf and Wong's results show how the upper-body action and lower-body locomotion can be treated independently during blending. We also focus on the decoupling of the upper body and the lower body, but in our case the goal is to decouple for splicing purposes rather than for interpolation.

Given keyframed values of a small number of a character's DOFs, Pullen and Bregler [PB02] identified segments of a motion data set where these DOFs were similar at user-specified frequency bands. These segments of motion data were then used to add higher-frequency content to the keyframed DOFs and to fill in the DOFs that were not keyframed. This work is similar in spirit to our own in that a motion is synthesized based on partial specifications of its DOFs. However, its goals are quite different: they sought to add detail to sparse keyframes of a small number of DOFs, whereas we seek to directly combine different DOFs of separate segments of motion data. Also, while they implicitly respected joint correlations through their matching algorithm, we explicitly adjust the timing and orientation of the spliced upper body so that it better fits the lower-body motion.

Our focus on locomotion is not unique; a great deal of previous research has sought ways of synthesizing this important and common activity. Multon et al [MFCD99] provide a survey of work through 1999, and more recent work has been done by Sun and Metaxas [SM01], Park et al [PSS02], Choi et al [CLS03], and Kwon et al [KS05]. Particularly relevant is the work of Ko and Badler [KB96], where inverse dynamics was used to adjust walking motions so a character would stay in balance and have reasonable joint torques. This approach is well suited for producing affects like leaning to compensate for a heavy weight that is being carried, but it is less appropriate for generating motions that are primarily driven by stylistic preferences (e.g., there are many ways to carry a light tray while staying in balance). We use motion capture data to incorporate these effects. Additionally, these methods for synthesizing locomotion can be useful during motion splicing for producing locomotion examples.

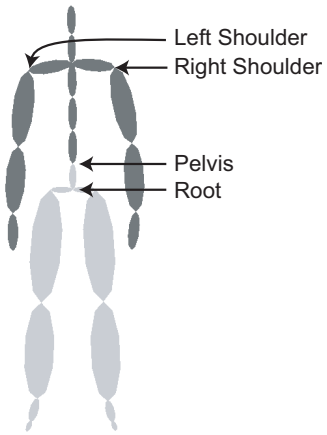

Figure 2: *The skeleton hierarchy used in our system. The upper body of the skeleton is the darkly shaded region while the lower body is the lightly shaded region.*

Other researchers have used the principles of physics to enforce physical properties on simulated motion. The work of Tak et al. [TSK02] uses Kalman filters to enforce physical constraints on motions and adjust for balance. Shin et al. [SKG03] used the physical laws of ballistic motion and zero moment point constraints to make a motion more physically plausible. While our technique does not directly take physics into account, because the physical requirements of the motion are encoded in the original motions, we often transfer the physical properties automatically when preserving temporal and spatial correlations. See Section 5 for more discussion on the advantages and disadvantages of not explicitly modeling physics.

3. Motion Splicing

We represent motions using a standard skeleton model. Our skeleton hierarchy is shown in Figure 2, with the upper body and lower body respectively indicated by darker and lighter regions. We define a motion as a continuous function

$$\mathbf{M}(t) = \{\mathbf{p}(t), \mathbf{q}_1(t), \ldots, \mathbf{q}_k(t)\} \quad (1)$$

that is regularly sampled into frames $\mathbf{M}(t_1), \ldots, \mathbf{M}(t_n)$. Here $\mathbf{p}$ is the position of the root with respect to the origin and $\mathbf{q}_j$ is the orientation of the j^{th} joint with respect to its parent (or, for the root, with respect to the global coordinate system). Values of $\mathbf{M}$ in between samples are computed by linearly interpolating the root position and using spherical linear interpolation on joint orientations. For splicing, it is convenient to rewrite $\mathbf{M}$ as

$$\mathbf{M}(t) = \{\mathbf{p}(t), \mathbf{Q}^l(t), \mathbf{q}_P(t), \mathbf{Q}^u(t)\} \quad (2)$$

where $\mathbf{Q}^l$ represents the orientations of the root and all joints in the lower body, $\mathbf{q}_P$ is the orientation of the pelvis with respect to the root, and $\mathbf{Q}^u$ represents the orientations of all joints in the upper body except for the pelvis.

This paper is concerned with attaching the upper body of one locomotion example, $\mathbf{M}_U$, onto the lower body of another locomotion example, $\mathbf{M}_L$, yielding a spliced motion, $\mathbf{M}_S$. Our goal is to preserve the relative locations of the joints within the upper-body of $\mathbf{M}_U$ and within the lower-body of $\mathbf{M}_L$ while still exhibiting details related to the correlations between the two halves. To do this, we construct $\mathbf{M}_S$ by identifying and enforcing temporal and spatial relationships within the motions. The process consists of three stages:

1. **Time Alignment.** Using the configuration of the lower bodies of the two example motions, we find a *time alignment curve*, $\lambda(t)$, that relates corresponding frames of $\mathbf{M}_L$ and $\mathbf{M}_U$. In particular, $\mathbf{M}_L(t_i)$ is at the same phase of the locomotion cycle as $\mathbf{M}_U(\lambda(t_i))$. The time alignment curve is used to identify temporal correlations between the example motions.

2. **Spatial Alignment.** For each frame $\mathbf{M}_L(t_i)$, we find a rotation about the pelvis that best aligns the upper body of $\mathbf{M}_U(\lambda(t_i))$ with the upper body of $\mathbf{M}_L(t_i)$. Intuitively, this alignment correlates the upper-body motion of $\mathbf{M}_U$ with the lower-body motion of $\mathbf{M}_L$ by using the upper-body motion of $\mathbf{M}_L$ as a reference.

3. **Posture Transfer.** The *posture* of a motion is the global relationship between the shoulders and hips of the character. A potentially undesirable side effect of spatial alignment is that it will alter the posture of $\mathbf{M}_U$ so that it roughly matches the posture of $\mathbf{M}_L$. We apply a posture transfer technique to retain the posture of $\mathbf{M}_U$ while still preserving the high-frequency details necessary to correlate it with $\mathbf{M}_L$'s lower body.

The result is a spliced motion, $\mathbf{M}_S$, of the form

$$\mathbf{M}_S = \{\mathbf{p}_L(t), \mathbf{Q}^l_L(t), \mathbf{q}^S_P(t), \mathbf{Q}^u_U(\lambda(t))\} \quad (3)$$

where the only quantities that need to be computed are the time alignment curve, $\lambda(t)$, and the pelvis orientations, $\mathbf{q}^S_P(t)$. Note in particular that, up to timewarping, the parameters of every joint except for the pelvis come directly from the captured motions $\mathbf{M}_L$ and $\mathbf{M}_U$.

The remainder of this section expands on each step of our algorithm.

3.1. Time Alignment

Before the upper body of $\mathbf{M}_U$ can be attached to the lower body of $\mathbf{M}_L$, it must be warped in time in order to retain important temporal correlations. For example, without timewarping the arms may appear to swing out of time with the legs. Timewarping is accomplished by constructing a time alignment curve, $\lambda(t)$, which is a strictly increasing function that maps frames of $\mathbf{M}_L$ to corresponding frames of $\mathbf{M}_U$. Since the characters in both example motions are locomoting, we can build λ by using the lower-body configurations of $\mathbf{M}_L$ and $\mathbf{M}_U$ as references. The dynamic timewarping method that we use is based on the work of Kovar and Gleicher [KG03], but there are several other published methods that would work as well. In particular, the timewarping methods of Bruderlin and Williams [BW95], Dontcheva et al. [DYP03], and Hsu et al. [HPP05] also use dynamic programming to find a temporal alignment between motions. Even greedy timewarping can work in cases where the motions have very similar lower bodies. But we have found that greedy search does not work for many of the more complex examples found in this paper and recommend using a method based on dynamic programming.

To compute the optimal time alignment, we use the following distance metric between two frames. First, for each frame, a point cloud is formed based on the positions of the root and both knees. We then compute the optimal sum of squared distances between corresponding points given that each cloud may be translated in the floor plane and rotated about the vertical axis. This metric was originally used by Kovar et al. [KGP02]; refer to this paper for a closed form solution to the optimization.

Our goal is to find a mapping between frames of $\mathbf{M}_L$ and $\mathbf{M}_U$ that minimizes the average distance between corresponding frames. First, assume that the initial frames $\mathbf{M}_L(t_0)$ and $\mathbf{M}_U(t_0)$ are in phase, i.e. both motions begin at the same point in the locomotion cycle. We start by calculating the distance between every pair of frames, forming a grid. Then we use the dynamic programming method of Kovar and Gleicher [KG03] to compute a continuous, monotonic, and non-degenerate path for every cell in the grid that connects to the lower left corner, while minimizing the sum of its cells. Next we scan the cells in the top and right edges of the grid for the path whose cells have the smallest average value. This optimal path provides a discrete, monotonically increasing time alignment, as shown in Figure 3. To generate the final time alignment curve, we fit a strictly increasing, endpoint interpolating B-spline to the optimal path.

If the initial frames of $\mathbf{M}_L$ and $\mathbf{M}_U$ are not in phase, then we crop an appropriate number of frames from the beginning of $\mathbf{M}_U$. This is done by computing the distance between

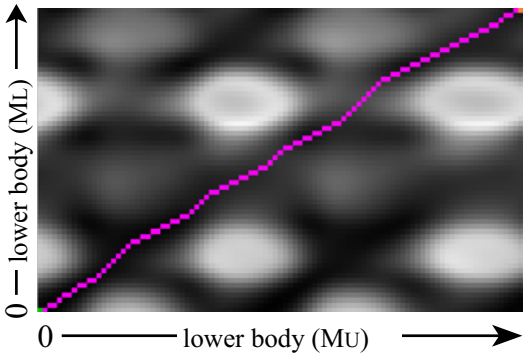

Figure 3: *A grid depicting the difference between the lower body of motion $\mathbf{M}_L$ and the lower body of motion $\mathbf{M}_U$. Dark areas denote similarity between the corresponding frames. An optimal time alignment is shown by the path through the grid.*

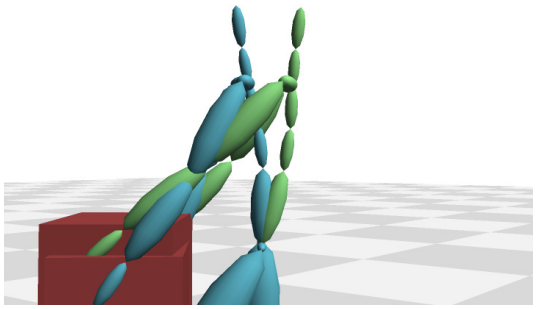

Figure 4: *A comparison between two motions spliced from a person carrying a heavy box and a person walking in a curve. The green motion was generated using our entire technique, while the blue motion lacks the adjustments made by posture transfer. Note that the green motion leans back, balancing the weight of the heavy box. The character is rendered as a stick figure to better illustrate this difference.*

$\mathbf{M}_L(t_0)$ and every frame of $\mathbf{M}_U$ and cropping $\mathbf{M}_U$ at the first local minimum.

3.2. Spatial Alignment

Once a time alignment curve is available, we are ready to attach corresponding frames together. The question at this stage is what local rotation should be used for the pelvis of the new motion. One possible choice is the local orientation of the pelvis in $\mathbf{M}_U$. However, the local orientation of the pelvis is tightly coupled to that of the root, and because the root orientations of $\mathbf{M}_U$ and $\mathbf{M}_L$ are not identical, simply copying the local pelvis orientation will destroy the coordination of the movement. Another possibility is to preserve the global orientation of the pelvis, but this can also be problematic. For instance, if we splice a motion of a character walking north with one of a character walking south, the upper body will face backwards in the result.

We instead approach this problem as one of spatial alignment. Just as we used the lower bodies of the two example motions for reference during time alignment, we use the upper bodies for spatial alignment. Specifically, for each pair of corresponding frames, we find a local pelvis orientation that best aligns the shoulders and spine of $\mathbf{M}_U$ with those of $\mathbf{M}_L$. This method ensures that the upper body faces the correct general direction while also adding details to the orientation that are appropriately correlated with the motion of the lower body. For example, when a person moves, their upper body rotates with each step. Step size, speed, and path curvature are just a few characteristics that can affect the exact details of these rotations. By aligning the upper body of $\mathbf{M}_U$ with the upper body of $\mathbf{M}_L$, these orientation details are transferred to the spliced motion.

As with the temporal alignment step, we use point clouds to perform spatial alignment. To spatially align the upper bodies of $\mathbf{M}_L(t_i)$ and $\mathbf{M}_U(\lambda(t_i))$, we first form a point cloud for each of the two motions based on the locations of the pelvis, spinal joints, and both shoulders. We then translate and rotate the point clouds so that the coordinate systems of the pelvis coincide with the origin. Finally, using the method of Horn [Hor87], we find the 3D rotation, $\mathbf{q}_A$, that minimizes the sum of squared distances between corresponding points. Then the local pelvis orientation in relation to the root of $\mathbf{M}_L$ that best aligns the upper body of $\mathbf{M}_U$ with the upper body of $\mathbf{M}_L$ is simply $\mathbf{q}_P^L * \mathbf{q}_A$, where $\mathbf{q}_P^L$ is the local pelvis orientation of $\mathbf{M}_L$.

3.3. Posture Transfer

For many upper-body actions, an important and necessary aspect of the motion is the overall orientation of the shoulders in relation to the hips over the duration of the motion. For example, a person carrying a heavy box leans back so as to counter the weight of the box. We call this overall relationship between the shoulders and the hips the *posture* of the motion. The spatial alignment procedure described in Section 3.2 preserves subtle correlations between the upper-body and lower-body motions by tracking the movements of the torso and shoulders, but it has the potentially undesirable side-effect that it changes the motion's posture. For instance, if the upper body of the person carrying the box were to be attached to the lower body of a person walking normally, the per-frame spatial alignment would rotate the upper body forward so that it would line up better with the straight-backed upper body of the normal walking motion (Figure 4).

Let $\mathbf{M}_{S'}$ be the result of splicing the upper body of $\mathbf{M}_U$ with the lower body of $\mathbf{M}_L$ using only time alignment (Section 3.1) and spatial alignment (Section 3.2). The goal of

Figure 5: *A motion generated by splicing the upper body of a person carrying a heavy box with the lower body of a person walking along a curved path.*

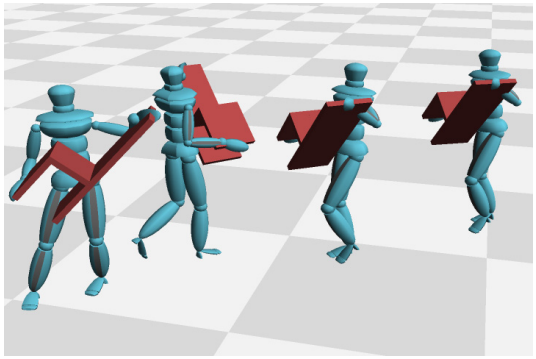

Figure 6: *A motion generated by splicing the upper body of a person carrying a chair with the lower body of a person making a sharp 180 degree turn.*

posture transfer is to replace the posture of $\mathbf{M}_{S'}$ with the posture of $\mathbf{M}_U$ while still preserving the high frequency details dictated by the lower-body motion of $\mathbf{M}_L$. Our general strategy is to compute a point cloud representation for the overall posture of each motion, and then to find the 3D rotation, $\mathbf{q}_G$, that best aligns these temporally global point clouds as we did in Section 3.2. This rotation specifies how $\mathbf{M}_{S'}$ should be globally adjusted in order to transfer the posture of $\mathbf{M}_U$.

To compute the point cloud representation for a single motion, we form a point cloud for each frame of the motion based on the locations of both shoulders. We then translate and rotate each point cloud so that the pelvis is at the origin and the vector defined by the hips is aligned with the x-axis. The point cloud posture representation is simply the average of these aligned per-frame point clouds. We can then find the 3D rotation, $\mathbf{q}_G$, that minimizes the sum of squared distances between corresponding points, again using the method of Horn [Hor87]. So for each frame the final local orientation of the pelvis, $\mathbf{q}_P^S$, is $\mathbf{q}_P^{S'} * \mathbf{q}_G$.

4. Results

All of the examples for this paper were computed on a 2.0GHz Intel Pentium 4. Each example took less computation time to produce than the duration of the final clip. For example, the 110-frame spliced motion (1.83 seconds) depicted in Figure 1 took .67 seconds to compute.

We tested a number of different motion combinations while developing this motion splicing technique; for example, Figure 5 shows the result of splicing the upper body of a person carrying a heavy box with the lower body of a person walking along a curved path. We have found that our method works well even in cases where the two lower bodies look very different. For example, a motion of a person making a sharp 180 degree turn looks considerably different from a typical forward walking motion. Yet the principles of this paper remain valid, and we have successfully spliced

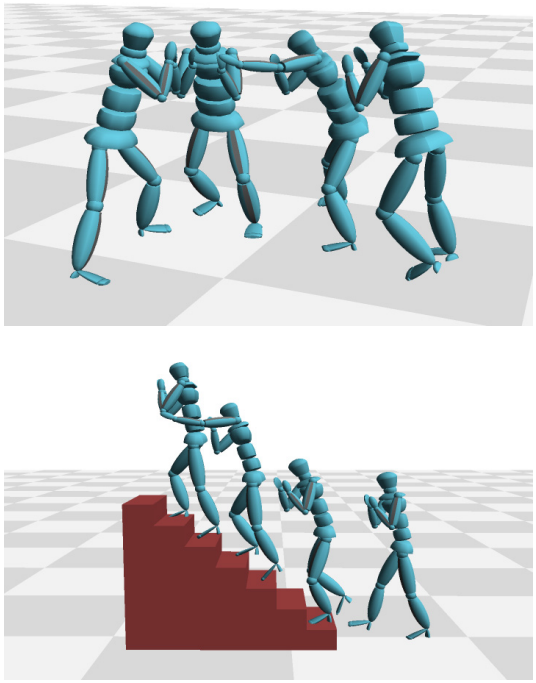

Figure 7: *The top motion is an original motion captured example of a person boxing. The bottom motion was generated by splicing the upper body of the person boxing with the lower body of a person walking up a set of stairs.*

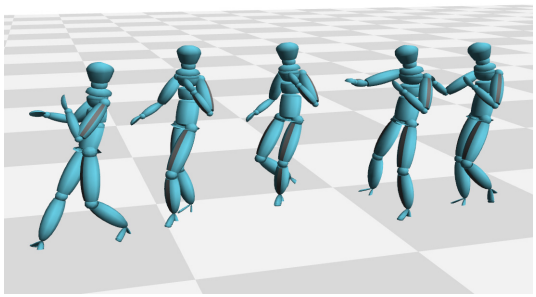

Figure 8: *A motion generated by splicing the upper body of a person pushing something with the lower body of a person side stepping around an obstacle.*

motions where a person walks forward while performing a variety of actions, such as carrying a chair, onto a sharp turn (see Figure 6).

Figure 7 shows another example of a challenging splice: an upper body of a boxer is spliced onto the lower body of a person climbing a set of stairs. Notice how the lower body of the boxing motion is not a traditional locomotion example, but since the character generally moves from one foot to the other throughout the motion, our algorithm is able to correlate the upper-body punching motion with the lower-body stair climbing motion. The punch is timed correctly with the motion of the legs as the character climbs the steps.

While the method presented in this paper works well on a wide variety of motions, there are limitations. Most importantly, since the method adjusts the way in which the shoulders twist, turn, and bend in relation to the lower body, an upper-body action whose shoulder *motion* is important to the meaning of the action cannot be spliced safely. For instance, when a person throws a football, not only does the arm rotate backward, so does the shoulder. Our method cannot guarantee preservation of this shoulder twist during splicing. Additionally, because it depends on the locomotion cycle for time alignment, our algorithm will not work on motions where the character is not locomoting. But because of the speed of our algorithm, it is feasible to try it on non-obvious motions, such as the boxing motion presented above. A few other locomotion examples that we have tested our algorithm on include motions of a person running, jogging, side-stepping, walking with a jaunt, and tip-toeing. We have also run tests using a number of different upper-body actions including acting like a zombie, pushing someone, and carrying a ceramic pot (see Figure 8). We have found that our motion splicing algorithm performs predictably, identifying and enforcing the desired temporal and spatial relationships.

5. Discussion

This paper has presented a simple and efficient method for splicing the upper-body action of one motion sequence onto the lower-body locomotion of another, thereby decoupling a character's action from the method and path of locomotion. To preserve realism, the spliced upper body is timewarped and rotated about its attachment point on a per-frame basis in order to achieve better correlation with the lower body. Given enough example locomotion data to allow flexible control, adding different upper-body actions requires capturing just one additional example motion. In contrast, with previous algorithms for data-driven motion synthesis, creating a character that can perform multiple motions simultaneously requires a combinatorial explosion in the number of example motions that must be captured. Motion splicing has the potential of reducing the data requirements to more manageable levels by allowing independent examples to be layered atop one another.

For this paper, we primarily used motion capture data in our experiments, but procedural, keyframed, or edited motions would serve just as well. For example, we often used standard techniques for automatically cyclifying a motion in order to extend it in length [LCR*02, AF02, KGP02].

While splicing characters at the waist in order to decouple upper-body action from locomotion is useful for a large number of applications, there may be applications where splicing the body in a different location is desirable. For example, one could imagine splicing only the arm of a waving motion in order to make a character wave. While the details of our algorithm do not apply in this single limb case, our general technique of using the example motions as references in order to identify and enforce spatial and temporal relationships is still relevant. Furthermore, it is likely that the three steps of our algorithm - temporal alignment, spatial alignment, and overall spatial relationship transfer - can be generalized to deal with these other splicing problems.

Our method does not explicitly take into account the physics of the original motions. In some cases, not explicitly dealing with dynamics could affect the perceived physical characteristics of the objects in the scene. For example, a heavy object may look less heavy because the character is able to accelerate with relative ease. Some of these potential problems could be fixed by running a physics cleanup algorithm, such as those in [TSK02] and [SKG03]. However, our results show that many effects due to physics are transferred from one motion to another using our example-based method. Even more importantly, our method has the added advantage of preserving stylistic properties that are hard to capture with purely physically-based methods.

References

[AF02] ARIKAN O., FORSYTHE D. A.: Interactive motion generation from examples. *ACM Transactions on*

Graphics 21, 3 (2002), 483–490.

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

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

[AW00] ASHRAF G., WONG K. C.: Generating consistent motion transition via decoupled framespace interpolation. In *Proceedings of Eurographics* (2000).

[BW95] BRUDERLIN A., WILLIAMS L.: Motion signal processing. In *Proceedings of ACM SIGGRAPH 1995* (Aug. 1995), Annual Conference Series, pp. 97–104.

[CLS03] CHOI M. G., LEE J., SHIN S. Y.: Planning biped locomotion using motion capture data and probabilistic roadmaps. *ACM Transactions on Graphics 22*, 2 (2003), 182–203.

[DYP03] DONTCHEVA M., YNGVE G., POPOVIC Z.: Layered acting for character animation. In *Proceedings of ACM SIGGRAPH 2003* (July 2003), Annual Conference Series, ACM SIGGRAPH.

[Hor87] HORN B. K. P.: Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A 4* (1987), 629–642.

[HPP05] HSU E., PULLI K., POPOVIC J.: Style translation for human motion. In *Proceedings of ACM SIGGRAPH 2005* (July 2005), Annual Conference Series, ACM SIGGRAPH, pp. 1082–1089.

[IF04] IKEMOTO L., FORSYTH D. A.: Enriching a motion collection by transplanting limbs. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2004* (Aug. 2004).

[KB96] KO H., BADLER N.: Animating human locomotion with inverse dynamics. *IEEE Computer Graphics and Application 16*, 2 (1996), 58–59.

[KG03] KOVAR L., GLEICHER M.: Flexible automatic motion blending with registration curves. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2003* (July 2003).

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

[KPS03] KIM T., PARK S., SHIN S.: Rhythmic-motion synthesis base on motion-beat analysis. *ACM Transactions on Graphics 22*, 3 (2003), 392–401.

[KS05] KWON T., SHIN S. Y.: Motion modeling for online locomotion synthesis. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2005* (July 2005).

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

[MFCD99] MULTON F., FRANCE L., CANI M.-P., DEBUNNE G.: Computer animation of human walking: a survey. *The Journal of Visualization and Computer Animation 10* (1999), 39–54.

[PB00] PULLEN K., BREGLER C.: Animating by multilevel sampling. In *IEEE Computer Animation Conference* (May 2000), CGS and IEEE, pp. 36–42.

[PB02] PULLEN K., BREGLER C.: Motion capture assisted animation: Texturing and synthesis. In *Proceedings of ACM SIGGRAPH 2002* (July 2002), Annual Conference Series, ACM SIGGRAPH.

[Per95] PERLIN K.: Real time responsive animation with personality. *IEEE Transactions on Visualization and Computer Graphics 1*, 1 (Mar. 1995), 5–15.

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

[PSS02] PARK S. I., SHIN H. J., SHIN S. Y.: On-line locomotion generation based on motion blending. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2002* (July 2002).

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

[RGBC96] ROSE C., GUENTER B., BODENHEIMER B., COHEN M.: Efficient generation of motion transitions using spacetime constraints. In *Proceedings of ACM SIGGRAPH 1996* (Aug. 1996), Annual Conference Series, ACM SIGGRAPH, pp. 147–154.

[RSC01] ROSE C., SLOAN P., COHEN M.: Artist-directed inverse-kinematics using radial basis function interpolation. *Computer Graphics Forum 20*, 3 (2001).

[SKG03] SHIN H., KOVAR L., GLEICHER M.: Physical touch-up of human motions. In *Proceeding of Pacific Graphics 2003* (Oct. 2003).

[SM01] SUN H., METAXAS D.: Automating gait animation. In *Proceedings of ACM SIGGRAPH 2001* (August 2001), Annual Conference Series, ACM SIGGRAPH, pp. 261–270.

[TSK02] TAK S., SONG O., KO H.: Spacetime sweeping: An interactive dynamic constraints solver. *Computer Animation 2002* (June 2002), 261–270.

[WH97] WILEY D., HAHN J.: Interpolation synthesis of articulated figure motion. *IEEE Computer Graphics and Application 17*, 6 (1997), 39–45.