Chapter 4

Splicing Upper-body Actions with Locomotion

As described in Section 1.2.1, interactive applications often divide character control into locomotion and action. Yet example-based motion synthesis fails to treat locomotion and action independently. This inability to decouple locomotion and action makes using an example-based approach to motion synthesis in interactive applications infeasible since it results in a combinatorial number of required example motions. This problem is of particular interest to the video game industry as many video games require locomotion and action to be decoupled during motion synthesis.

Existing methods for decoupling locomotion and action during motion synthesis do little to account for the natural correlations within the body. 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, MZF06]. To complicate the matter, these natural correlations not only stem from physical needs, such as balance preservation, but also from harder-to-characterize properties associated with the stylistic form of human motion.

In this chapter, I describe 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. In light of the complicated and subtle form of the relationships I wish to preserve, I adopt an example-based approach that produces natural motions by identifying and enforcing temporal and spatial relationships between different parts of the body. The simple and efficient method synthesizes spliced motions that appropriately preserve correlations. For instance, Figure 4.1 shows the results of splicing the upper body of a person holding a cup onto the lower body of a person stepping

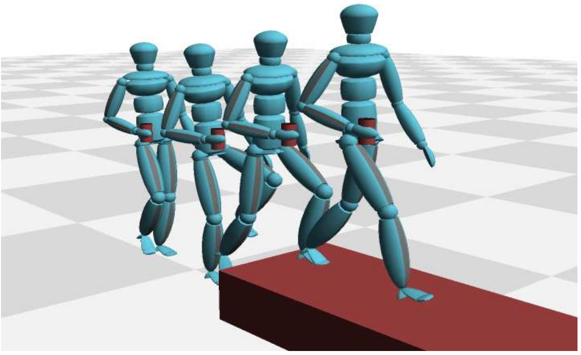


Figure 4.1 A motion generated by splicing the upper body of a person carrying a cup with the lower body of a person stepping up onto a platform.

up onto a platform. My algorithm allows the upper-body action to be decoupled from the method of locomotion yet preserves important details, such as the shifting of the character's weight as he pulls himself up onto the platform, the character's cup-carrying posture, and the forward lean of the upper body prior to the step up.

The rest of this chapter will describe the results from a study that looks at natural correlations in motion (Section 4.1), detail my algorithm for splicing upper-body actions with locomotion (Section 4.2), provide some additional insights into using my algorithm at runtime (Section 4.3), provide the results of some experiments involving motion splicing (Section 4.4), and conclude with a general discussion of the technique, its advantages, and its disadvantages (Section 4.5).

4.1 Correlation Study

As defined by the American Heritage Dictionary [Ame04], a correlation is

> [a] casual, complementary, parallel, or reciprocal relationship ... between two comparable entities

In other words, there is a correlation between two things if one changes in a relatable when the other changes.

The key challenge to splicing one part of a motion onto another is that the body is highly correlated. For example, when a person does jumping jacks, her arms move up at the same time that her legs spread out. This correlation between the upper-body motion and the lower-body motion is part of what makes the jumping jack motion recognizable.

This chapter is concerned with the splicing of an upper-body action onto a lower-body locomotion. For this specific, yet important, motion splicing problem, parts of the body that seem unrelated are actually closely tied together. While working on my algorithm for splicing upper-body actions with lower-body locomotion, I analyzed these correlations in locomotion examples.

My method for analyzing the upper body and lower body correlations in a human locomotion example consists of five steps:

1. Perform forward kinematics on the motion under analysis.
2. Plot the global orientations of each upper-body joint and the global x-rotation of the hip joints (i.e., the part of the hip's rotation which controls the forward and backward motion of the leg) with respect to the root.
3. Fit a polynomial to each data line, producing a trend curve.
4. Identify correlations by finding trend curves that are in phase with the hip trend curves or 180 degrees out of phase with the hip trend curves.
5. Repeat steps 3, 4, and 5 on the local orientations of the upper body joints and the x-rotation of the hip joints.

For the results presented here, a computer was used to perform forward kinematics, plot data lines, and fit trend curves. The final correlation identification was done manually.

I analyzed the correlations for a number of locomotion examples; I will discuss here some of the correlations that appeared consistently in all of my examples. The global analysis of locomotion shows a consistent bobbing of the entire upper body (e.g., the upper body leans forward and backward in time with the motion of the legs). This correlation is a lot stronger in motions where the upper body is highly constrained, for example, when a walking character carries a heavy box (see Figure 4.2).

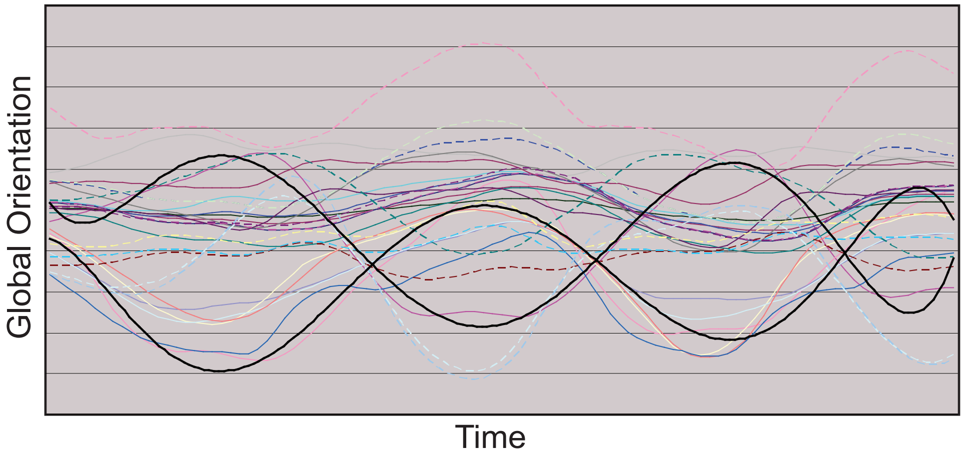


Figure 4.2 Plots of the global orientations of correlated joints in a walking motion as determined by my method. This graph shows the large number of joints that are globally correlated in a simple walking motion.

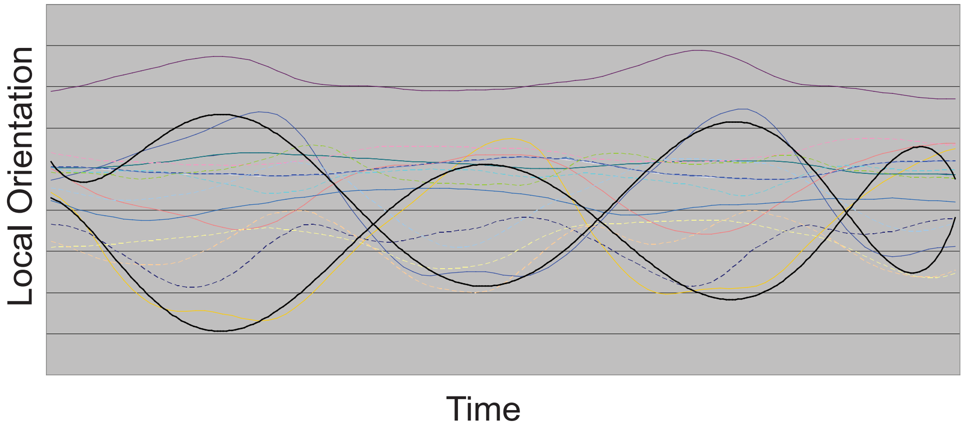


Figure 4.3 Plots of the local orientations of correlated joints in a walking motion as determined by my method. This graph shows the large number of joints that are locally correlated in a simple walking motion.

The local orientation analysis consistently showed that as the character moves, the upper body twists through the spine causing the relationship between the shoulders and the hips to change. And, in general, the head also needs to twist in the opposite direction to allow the character to continue looking in the original direction of travel (see Figure 4.3).

Many other correlations show themselves using this type of analysis. The number and degree of the correlations varies greatly depending on the exact motion, but the importance of correlations to natural human locomotion becomes obvious when looking at this data. The importance of these correlations is the driving motivation behind my algorithm for quality splicing of upper-body action and locomotion presented in the next section.

4.2 Splicing Algorithm

My technique for splicing the upper-body action of one motion onto the lower-body locomotion of another 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 I have one example of a character performing an action while moving around, I can transfer that action onto a different example of a character moving around. For this reason, I require that all captured examples of upper-body actions be performed while locomoting. In particular, this requirement provides a basis for temporal correlations, which are extraordinarily important during locomotion (see Section 4.1).

The second key observation is that 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. Since I do not know what the upper body or lower body is doing, only that the upper body is performing an action and the lower body is locomoting, I cannot safely make any changes within the upper body or lower body. This leads me to restrict the kinds of changes that can be made during splicing. Specifically, I 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.

4.2.1 A Technical Overview

Motion splicing 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, $\mathbf{M}_S$. My 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, I construct $\mathbf{M}_S$ by identifying and enforcing temporal and spatial relationships within the motions. The process consists of three stages:

1. **Time Alignment (Section 4.2.2).** Using the configuration of the lower bodies of the two example motions, 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 (Section 4.2.3).** For each frame $\mathbf{M}_L(t_i)$, 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 (Section 4.2.4).** 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 $\mathbf{M}_L$. Thus, the final step is to 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

$$\mathbf{M}_S = \{\mathbf{p}_L(t), \mathbf{Q}_L^l(t), \mathbf{q}_P^S(t), \mathbf{Q}_U^u(\lambda(t))\} \qquad [4.1]$$

where the only quantities that need to be computed are the time alignment curve, $\lambda(t)$, and the pelvis orientations, $\mathbf{q}_P^S(t)$. 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 $\mathbf{M}_U$.

Each step of the motion splicing algorithm uses a different part of the original motion examples as a reference, thereby transferring and preserving important correlations: time alignment uses the lower body of $\mathbf{M}_L$; spatial alignment uses the upper body of $\mathbf{M}_L$; and posture transfer uses the posture of $\mathbf{M}_U$.

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

4.2.2 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. Because naïve DOF replacement (see Section 2.1.3) does not consider the temporal alignment of natural locomotion correlations, it can easily cause motions to be spliced out of sync. In practice, example motions are carefully hand-tailored to avoid timing issues, but this process is extraordinarily time-consuming and greatly limits the motions that can be spliced together without temporal correlation problems. Figure 4.4 shows a graph of a motion that has been spliced together without any timing adjustments.

Timewarping is accomplished by constructing a time alignment curve, $\lambda(t)$, that maps frames of $\mathbf{M}_L$ to corresponding frames of $\mathbf{M}_U$. Since the characters in both example motions are locomoting, λ can be built by using the similarity of the lower-body motions of $\mathbf{M}_L$ and $\mathbf{M}_U$ to build the alignment curve.

Specifically, in this step of the motion splicing algorithm, the goal is to find a mapping between frames of $\mathbf{M}_L$ and frames of $\mathbf{M}_U$ that minimizes the average distance between corresponding frames. The distance between frames is calculated as described in Section 3.2, using the positions of the root and both knees to form the point clouds. To compute the optimal time alignment, 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. Then, use the algorithm presented in Section 3.4 to compute

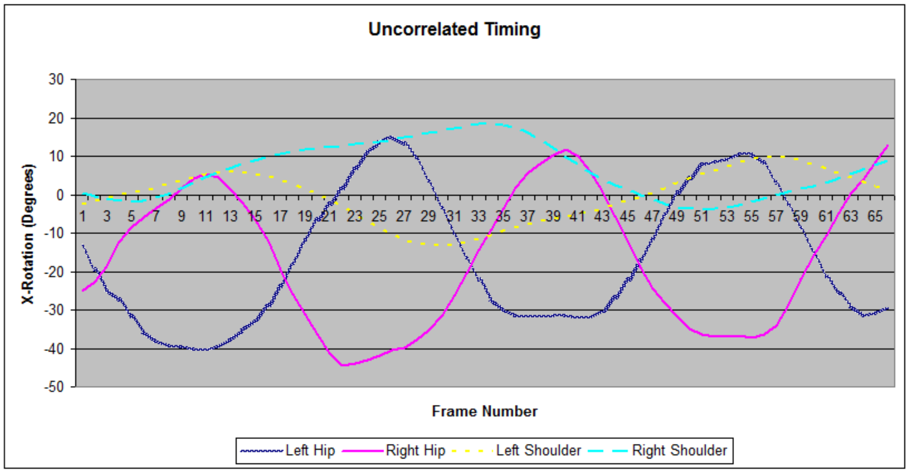


Figure 4.4 Timing difference caused by splicing two motions that have not been time aligned. Notice how the rotations of the arms are in sync and the rotations of the legs are in sync, yet the arms and legs are not correlated with each other. These correlation issues make the motion appear odd, even to someone unfamiliar with the reason why.

the optimal time alignment curve grid. Next scan the cells in the top and right edges of the grid for the path whose cells have the smallest average value. As described in Section 3.4, this optimal path defines a continuous time alignment curve for the two motions.

If the initial frames of $\mathbf{M}_L$ and $\mathbf{M}_U$ are not in phase, then crop an appropriate number of frames from the beginning of $\mathbf{M}_U$ before computing the time alignment curve. This is done by computing the distance between $\mathbf{M}_L(t_0)$ and every frame of $\mathbf{M}_U$ and cropping $\mathbf{M}_U$ at the first local minimum.

4.2.3 Spatial Alignment

Once a time alignment curve is available, the system can 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. The local orientation not only controls how the upper body should turn, bend, and twist in relation to the lower body, but it also compensates for movement within the root, effectively stabilizing the upper body. 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. This coupling is one of the main reasons why naïve DOF replacement often produces wobbly looking upper body motions (see Section 2.1.3). Figure 4.5 illustrates the wobbling issues associated with naïve DOF replacement.

Another possibility for the local rotation of the pelvis is to preserve the joint's original global orientation, but this can also be problematic. For instance, if a motion of a character walking east is spliced with one of a character walking west, the upper body will face backwards in the result (see Figure 4.6).

I instead approach this problem as one of spatial alignment. Just as the lower bodies of the two example motions were used for time alignment, the upper bodies can be used for spatial alignment. Specifically, for each pair of corresponding frames, the goal is to 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

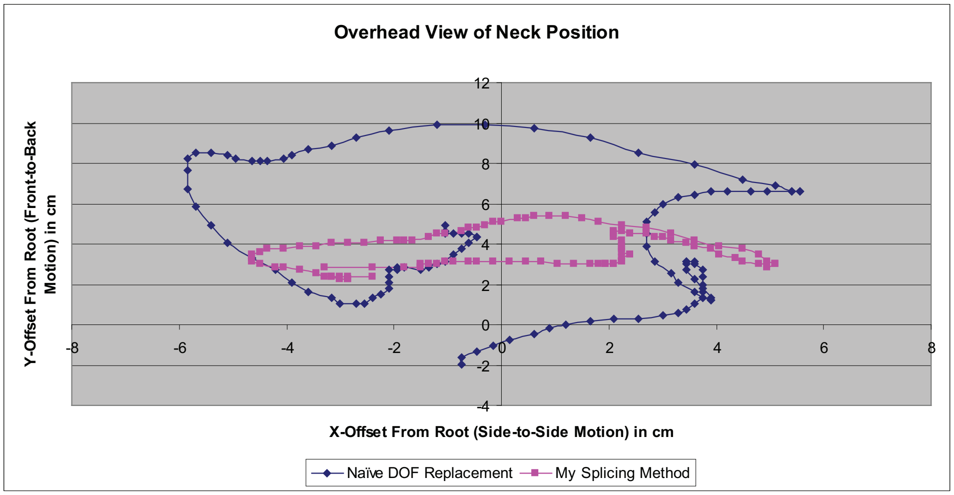


Figure 4.5 Stability comparison of naïve DOF replacement and my motion splicing algorithm. This graph shows the position of the neck in relation to the root for a motion spliced together using naïve DOF replacement and a motion spliced together using my method. Both motions were spliced from the same source data - the upper body of a person carrying a cup and the lower body of a person stepping up onto a platform. Notice how the naïve DOF replacement result moves erratically and wildly when compared with the more stable result produced using my technique.

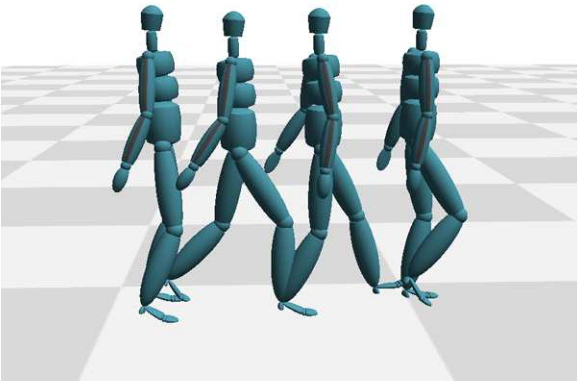


Figure 4.6 The result of splicing the upper body of a person walking west with the lower body of a person walking east using the global orientation of the characters for alignment. The result does not look natural because the upper body faces backwards on the lower body.

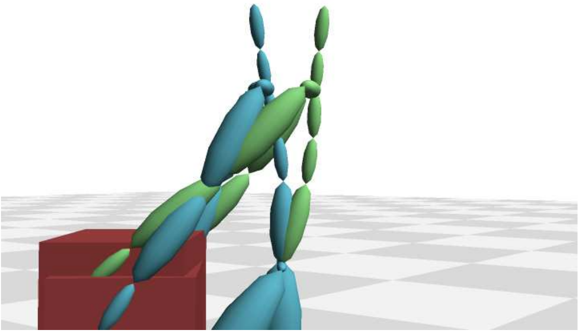


Figure 4.7 A comparison between two motions spliced from a person carrying a heavy box and a person walking in a curve. The green motion was generated using my entire technique, while the blue motion lacks the adjustments made by posture transfer. Note that the green motion leans back, balancing the weight of the heavy box. The character is rendered as a stick figure to better illustrate this difference.

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, point clouds are used to perform spatial alignment (see Section 3.2 for more information on point clouds). To spatially align the upper bodies of $\mathbf{M}_L(t_i)$ and $\mathbf{M}_U(\lambda(t_i))$, first form a point cloud for each of the two motions based on the locations of the pelvis, spinal joints, and both shoulders. 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], find the 3D rotation at the origin, $\mathbf{q}_A$, 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 $\mathbf{M}_L$.

4.2.4 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. I call this overall relationship between the shoulders and the hips the *posture* of the motion. The spatial alignment procedure described in Section 4.2.3 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 (see Figure 4.7).

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 4.2.2) and spatial alignment (Section 4.2.3). The goal of 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 $\mathbf{M}_L$. The general strategy is to compute a point cloud representation (again, see Section 3.2 for more details on point clouds) 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 was done in Section 4.2.3. 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, form a point cloud for each frame of the motion based on the locations of both shoulders. 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. 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 $\mathbf{q}_P^{S'} * \mathbf{q}_G$.

4.3 Runtime Modifications

To effectively use the methods presented in this chapter to synthesize a motion clip at runtime in an interactive application, it is necessary to apply the algorithm in a slightly different manner from that presented. In particular, the time alignment (Section 4.2.2) and posture transfer (Section 4.2.4) steps require that the motions be processed globally. Even more problematic is that the time alignment algorithm scales poorly with the length of the motions being compared; if there are u frames in motion $\mathbf{M}_U$ and there are l frames in motion $\mathbf{M}_L$, the algorithm scales with $\mathbf{O}(ul)$. Thus, I make the following two suggestions for performing motion splicing at runtime:

1. The time alignment curve, $\lambda(t)$, should be precalculated and stored in a lookup table for easy access. This can either be done directly between every pair of possible spliced motions, or done indirectly by time aligning each motion to a baseline reference locomotion example. The latter method is overall a better approach as it is considerably faster to compute and requires only $\mathbf{O}(n)$ storage space, where n is the total number of example motions. The former method does have a slight advantage over using a baseline motion - determining the time alignment between two motions at runtime only requires a single time alignment curve lookup while the baseline motion method requires the composition of two time alignment curve lookups. Yet the storage savings garnered by using the baseline motion method far outweigh this constant time increase in computation time. For either case, these stored time alignments can be used to temporally align motions for quick splicing at runtime.

2. For each of the upper-body example motions, I also suggest computing and storing the posture of the motion prior to runtime. Again, these precomputed postures can be accessed quickly for posture transfer. The posture computations of $\mathbf{M}_{S'}$ can be calculated in one of two ways: as a running average, a method that should work well given that posture could change slightly over the course of a motion; or, since the spatial alignment step (Section 4.2.3) causes the posture of $\mathbf{M}_{S'}$ to roughly match the posture of $\mathbf{M}_L$, a precomputed posture for $\mathbf{M}_L$ can be used as an approximation of the posture of $\mathbf{M}_{S'}$.

These simple modifications to the application of the algorithm make it applicable as a direct replacement for runtime naïve DOF replacement in existing interactive applications.

4.4 Results

I tested a number of different motion combinations while developing this motion splicing technique. Here I describe in detail some of these splicing results:

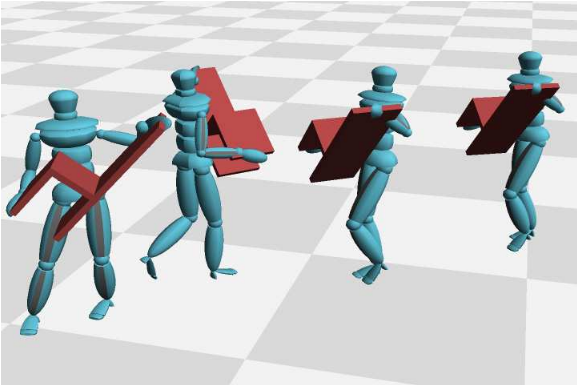
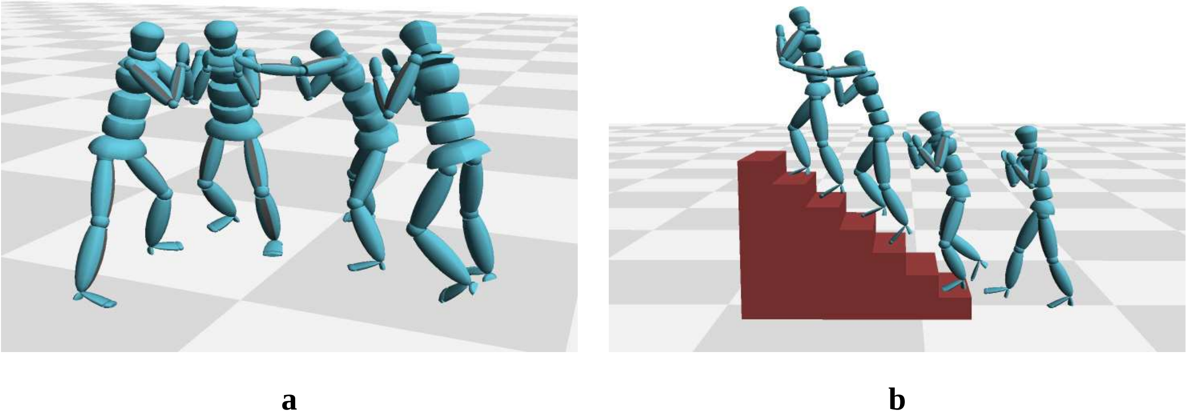


Figure 4.8 A motion generated by splicing the upper body of a person carrying a chair with the lower body of a person making a sharp 180 degree turn.



a

b

Figure 4.9 An unusual motion splice. (a) is an original motion-captured example of a person boxing. (b) was generated by splicing the upper body of the person boxing with the lower body of a person walking up a set of stairs.

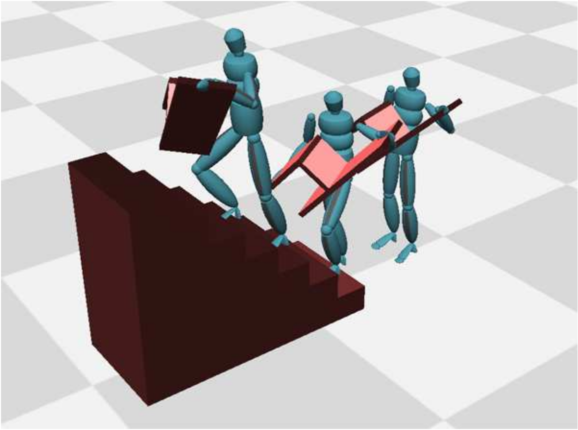


Figure 4.10 A motion generated by splicing the upper body of a person carrying a chair with the lower body of a person walking up a set of stairs.

Carrying a Box in a Curve. Figure 1.3 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. As described in Section 1.2.1, the spliced character leans into the turn, leans back to counter the weight of the box he is carrying, and twists his upper body slightly with each step. When this result is compared with ground truth, motion-captured data of an actual person walking along a curved path while carrying a heavy box, it is difficult to distinguish one from the other; both characters twist, turn, and bend in similar ways.

Stepping-Up While Carrying Cup. Figure 4.1 shows the results of splicing the upper body of a person holding a cup onto the lower body of a person stepping up onto a platform. Important correlating details are clearly seen in the motion; the character shifts his weight as he pulls himself up onto the platform, the character retains a rigid cup-carrying posture that would keep the contents of the cup inside the cup, and the character leans forward with his upper body prior to stepping up onto the platform.

Sharp 180 with a Chair. I have found that my 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 presented in this chapter remain valid, and I have successfully spliced motions where a person walks forward while performing a variety of actions, such as carrying a chair, onto a sharp turn (see Figure 4.8).

Punching while Climbing Stairs. Figure 4.9 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, the 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.

Climbing Stair with a Chair. Figure 4.10 shows the result of splicing the upper body of a person carrying a chair with the lower body of someone climbing a set of stairs. With each step up the stairs, the character's upper body twists slightly more than it does when walking straight forward, a characteristic of step climbing motions. The character also leans forward with his upper body before taking the first step.

Descending Stairs with a Heavy Box. I can also splice the upper body of a person carrying a heavy box with the lower body of a person descending a set of stairs. The character correctly settles his weight backwards when he reaches the bottom of the steps in the spliced motion (see Figure 4.11).

Running While Carrying a Cup. Motions of people walking can also be spliced onto motions where the character is moving much faster. For instance, Figure 4.12 shows the result of splicing the upper body of the cup-carrying character onto the lower body of a person running in a curve. Again, the cup stays upright throughout the motion, the left arm swings in time with the motion of the legs, and the entire motion leans into the curve.

Walk-to-Run with a Heavy Box. The motion splicing algorithm also works when the timing of the locomotion changes. Figure 4.13 shows the result of splicing the upper body of the character carrying a heavy box onto the lower body of a person who is walking and then begins to run. The character leans back throughout the motion to counter the weight of the heavy box, and the upper body twists from side-to-side with varying intensities as he changes from walking to running.

Side-stepping While Pushing. The last example shows another unusual splice. The upper body comes from a character who is stepping randomly while pushing something, and the lower body is from a character who steps around an obstacle. The spliced motion is of the character pushing something as he steps around it (see Figure 4.14). Notice how the character appears to look to the left as he steps around the object, a characteristic that is transferred from the original lower body motion.

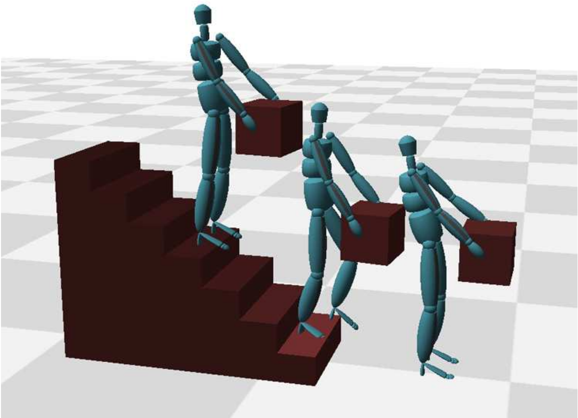


Figure 4.11 A motion generated by splicing the upper body of a person carrying a heavy box with the lower body of a person walking down a set of stairs.

A few other locomotion examples that I have tested my algorithm on include motions of a person jogging, walking with a jaunt, and tip-toeing. I have also run tests using a number of different upper-body actions including acting like a zombie and carrying a ceramic pot with both hands.

4.4.1 Limitations

While the method presented in this chapter 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. My splicing method cannot guarantee preservation of this important shoulder twist.

One potential drawback of the motion splicing method is that it 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 very 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, my results show that many effects due to physics are transferred from one motion to another using my example-based method. Even more importantly,

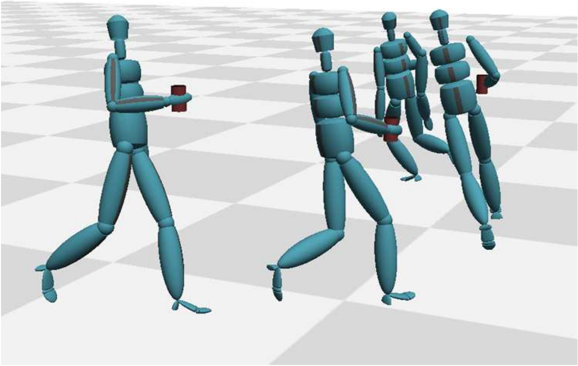


Figure 4.12 A motion generated by splicing the upper body of a person carrying a cup with the lower body of a person running.

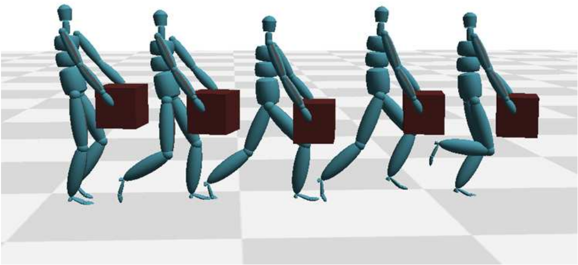


Figure 4.13 A motion generated by splicing the upper body of a person carrying a heavy box with the lower body of a person walking and then running.

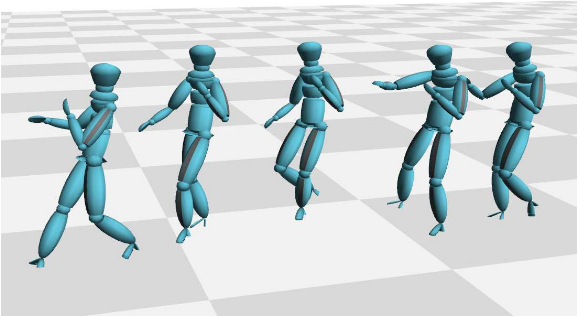


Figure 4.14 A motion generated by splicing the upper body of a person pushing something with the lower body of a person side stepping around an obstacle.

my method has the added advantage of preserving stylistic properties that are hard to capture with purely physically-based methods.

Additionally, because it depends on the locomotion cycle for time alignment, my motion splicing algorithm will not work on motions where the character is not locomoting. But because of the speed of my algorithm, it is feasible to try it on non-obvious motions, such as the boxing motion presented earlier. Unlike the results produced using the naïve DOF replacement algorithm, I have found that my motion splicing algorithm performs predictably, identifying and enforcing the desired temporal and spatial relationships.

4.4.2 Algorithm Performance

In this section, I describe how my motion splicing algorithm performs in each of the six categories described in Section 1.1.

Efficient Synthesis All of the examples in this chapter 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 4.1 took $.67$ seconds to compute. This timing data includes the time it takes to build a time alignment curve and calculate the global posture information. If temporal and postural information is precomputed, then each frame of motion can be computed in $\mathbf{O}(1)$ time. On average, this constant amount of time is only $.0008$ seconds. This data shows that the splicing algorithm works quickly and in a predictable amount of time per frame.

Efficient Data Storage Because the algorithm effectively decouples locomotion from action, it reduces the amount of required storage space for example motions and associated data from $\mathbf{O}(nm)$ to $\mathbf{O}(n + m)$, where n is the number of locomotion examples needed for flexible navigation and m is the number of actions that the character can perform while locomoting.

Low Latency or Response Time Because the motion splicing algorithm is designed for motion clip generation and not motion stream generation, this characteristic is not applicable.

Accurate Motion Generation For splicing motions, requests take the form of which two motions should be spliced together. Since the algorithm always does the splice between the two requested motions, it accurately meets this request.

Visual Quality Since motion quality is partially subjective, it can be difficult to measure the quality of the motions produced. However, by limiting the changes that are made to the original motions (see Section 4.2.1), there is little deviation from the original example motions, which are assumed to be of high-quality. In addition, all of the changes made to temporally and spatially align the spliced motions are smooth and continuous.

Automated Authoring The algorithm for splicing the upper body of one motion with the lower body of another is completely automated and uses no user-defined parameters.

4.5 Discussion

This chapter 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 the examples in this chapter, I primarily used motion-captured example motions in the experiments, but procedural, keyframed, or edited motions would serve just as well as example motions. For instance, standard techniques can be used to automatically cyclify a motion in order to extend it in length [LCR$^+$02, AF02, KGP02]. This is a technique that was used to create some of the examples presented in this chapter.

My focus on the practical and important problem of splicing upper-body actions with lower-body locomotion rather than the more general problem of transferring any subset of the body from one motion to another has allowed me to develop a fast, simple, and reliable algorithm. But 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 my algorithm do not apply in this single limb case, the general technique of using 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 the algorithm - temporal alignment, spatial alignment, and overall spatial relationship transfer - can be generalized to deal with these other splicing problems. This belief is supported by the similarities between my algorithm and the algorithm of Majkowska et al. for splicing hands onto a full body motion [MZF06]. By showing the utility of decoupling motion parameters through splicing, I hope that my algorithm with inspire other related work.