Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2006)
M.-P. Cani, J. O'Brien (Editors)

Composition of Complex Optimal Multi-Character Motions

C. Karen Liu [1,2] Aaron Hertzmann [3] Zoran Popović[1]

[1]University of Washington [2]University of Southern California [3]University of Toronto

Abstract

This paper presents a physics-based method for creating complex multi-character motions from short single-character sequences. We represent multi-character motion synthesis as a spacetime optimization problem where constraints represent the desired character interactions. We extend standard spacetime optimization with a novel timewarp parameterization in order to jointly optimize the motion and the interaction constraints. In addition, we present an optimization algorithm based on block coordinate descent and continuations that can be used to solve large problems multiple characters usually generate. This framework allows us to synthesize multi-character motion drastically different from the input motion. Consequently, a small set of input motion dataset is sufficient to express a wide variety of multi-character motions.

1. Introduction

Multiple character interaction is critical to the perceptual immersion for training simulation, tele-communication, and video games. One possible avenue to realistic multi-character motion is through reuse and editing of example motion sequences, using both physics-based and data-driven approaches. Despite the recent advent of automatic techniques for motion synthesis, the scope of most existing methods can only generate motion of a single character performing a single task.

Physics-based motion editing requires very little input data to modify basic dynamic characteristics of a motion, such as speed or orientation. However, these methods are limited to the low-level modifications of motion parameters, unable to change the high-level content of the motion. Furthermore, physics-based methods rely on intensive computation that scales poorly to a large optimization problem. In contrast, data-driven approaches take advantage of large motion databases by composing segments of example motions. This approach can create long sequences that achieve coarse-grained goals, but only if appropriate motion segments exist in the database. Clearly, data-driven methods cannot record all possible interactions among multiple characters in advance.

In this paper, we describe techniques for adapting a small collection of motions to create complex new motions with multiple characters. For example, beginning with one character walking and an another character running, we can create a new animation, in which one character attempts to tackle while the other character tries to dodge the tackle (Figure 1). The user can determine the scenario of the motion by specifying whether the dodge is successful, and, if so, by how much. These new motions are defined as solutions to a physics-based spacetime optimization problem.

When directly applied to multi-character motion, the standard spacetime constraints framework fails on two accounts. First, existing spacetime algorithms require the user to specify detailed environment constraints for a motion, such as specific locations and the timing of all footsteps. The realism of the output motion largely relies on the user's experience in setting the environment constraints. When multiple characters interact with each other, specifying these constraints manually could be exceedingly difficult. Second, the spacetime optimization framework is prohibitively expensive for large problems and increasingly prone to local minima. In this paper, we introduce methods that automatically optimize the timing and the placement of the constraints, avoiding suboptimal output motion due to unrealistic environment constraints specified by the user. This framework augments a spacetime optimization with a time-warping parameterization, in which both the timing of the motion and the constraints are optimized in synchrony. To address the second problem, we extend the spacetime windows approach with block coordinate descent and continuation strategies to solve for each character separately even when the characters' motion is mutually constrained (e.g. walking hand-in-hand).

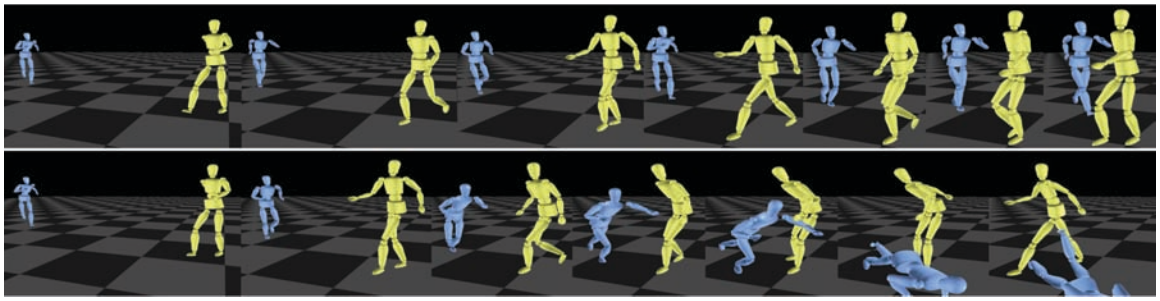

Figure 1: *Top: The input motion contains a walk and a run performed by two characters. Bottom: In the output motion, the blue character attempts to tackle while the yellow character tries to dodge the tackle.*

Combining these techniques significantly expands the flexibility of physics-based animation and motion editing to the domains of complex multi-character motion. We demonstrate our algorithms on both collaborative and adversarial interactions, that prior spacetime approaches have not been able to successfully synthesized. Furthermore, we believe that our techniques are general enough to be applied to all previous spacetime formulations in the literature, thus significantly enhancing their applicability.

2. Related work

Recent research in character animation has been focused on two main themes: physics-based and data-driven motion synthesis. We build our work on a physics-based framework while exploiting a small data set of basic motion from the real world.

In data-driven approaches, motions are created by editing and combining example motions. One simple approach is to perform direct filtering of joint angles [BW95, WP95] and interpolation of example motions [WH97, RCB98]. As in these methods, we adopt motion time-warping technique [WP95], and extend it to time-warping both motion and constraints. More recent approaches create motions by splicing together a collection of smaller motion segments. [AF02, KGP02, LCR*02, LWS02, PB02] or poses similar to examples poses [BH00, GMHP04, LWS02]. Since all poses come from real human data, the rich nuances of the motion are easily preserved. However, only poses that exist in the motion database (or very similar ones) can be generated, thus requiring potentially vast databases of example motions; handling multi-character interactions would make the requirements even more stringent.

The spacetime constraints framework casts motion synthesis as a variational constrained optimization problem [WK88, LGC94, PSE*00]. The algorithm minimizes a specific physical measure of energy, such as joint torques or power consumption, while satisfying dynamic or user-specified constraints. The optimal energy movement and intuitive user control are appealing for motion synthesis in low dimensional space. However, when applied to complex human motion, the optimization quickly becomes intractable and highly prone to local minima. Much research work has focused on optimizing simplified models, including simplified characters [PW99], momentum constraints [LP02], aggregate force models [FP03], and data-driven parameterization [SHP04]. In each case, the results of the optimization is strongly dependent upon choosing an appropriate parameterization.

Physics-based methods for editing motions combine the realism of example motions with dynamically plausible modifications [Gle97, PR01, RGBC96, ZH99], but are limited to relatively small, user-specified changes to constraint positions or timing. Gleicher [Gle98] described a method for footprint variation, when no dynamic constraints are present. Liu and Cohen [LC95] presented a technique that allows the keyframe timing to vary, thereby changing the parameterization of the motion, but holding all other constraints (e.g footprint constraints) fixed. We introduce a novel time-warping parameterization to jointly optimize the motion and the constraints in concert. Our algorithms extend spacetime framework to optimize not only the joint angles, but also the constraints, to achieve optimal motion for complex multi-character interactions.

Despite great advances in spacetime optimization, it remains applicable only for relatively short animation sequences with single character. Our framework of motion composition exploits strategies of block coordinate descent and continuations [Bet01]. Cohen [Coh92] introduced a spacetime method similar to block coordinate descent algorithm. This method breaks a large problem into a set of subproblems, optimized sequentially over windows of time. van de Panne [vL95] applied the continuation strategy to learning of balanced locomotion by using gradually reduced guiding forces to assist the process of learning. Combining the strategies of block coordinate descent and continuations, our optimization framework is able to scale to large problems that multi-character motions usually generate.

Multi-character animations has been demonstrated using data-driven methods [AF02], controller-based methods [Rey87], or a combination of both [ZMCF05]. However, data-driven methods based on kinematic models require sufficient example motions to describe the variability of multi-character motion. Our approach is also data-driven, but uses the data to learn a dynamic model that can generalize to new dynamic configurations not present in the training data. Robot controller design for multi-character motions remains a difficult process in spite of some recent advances towards automatic controller synthesis [HP97, FvdPT01]. Moreover, these methods usually lack the intuitive control provided by the spacetime framework. The goal of our work is to show that complex multi-character dynamics can also be synthesized with spacetime optimization.

Our work is also related to motion planning which searches for a motion sequence that satisfies complex kinematic constraints [KKKL94, YKH04]. While these methods satisfy complex constraints, they do not model dynamic properties of motion, resulting in less realistic motions. Kuffner et al. [KKK*02] describe a method for planning dynamically stable robot motions, but require user-specified keyframe constraints and fixed positions for either or both feet. Our method combines physics-based animation with a form of motion planning, allowing for minimally-specified constraints with time-varying dynamic configurations, although our system cannot solve fully-general planning problems.

3. Overview

We represent multi-character motion synthesis as a spacetime optimization problem where constraints represent the desired character interactions. Our system is composed of two major components: motion optimization and motion composition. The first component describes an extended spacetime optimization framework that solves for *both* motion and constraints. One distinctive feature of our framework is the novel constraint representation that allows the timing and the spatial coefficients (e.g. desired location of the foot for a footprint constraint) of a constraint to vary during the course of the optimization.

The flexibility to vary a constraint is crucial to the synthesis of interactive motion between multiple characters. It does not, however, alleviate the poor convergence and scaling issues raised by large optimization problems. In the second part of this paper, we describe a generalized framework that synthesizes complex motion as an optimal composition of smaller optimal sequences, each of which is the solution to a subproblem. The user can then arrange the subproblems into different schedules depending on the scenarios of the interaction among multiple characters.

The input to our system comprises input motion clips, initial environment constraints (e.g. footprints of the input motion clips), and the user-specified constraints that indicate the desired interaction. Our system automatically optimizes the motion subject to the user-specified constraints, and modifies the timing and the spatial coefficients of the initial environment constraints accordingly.

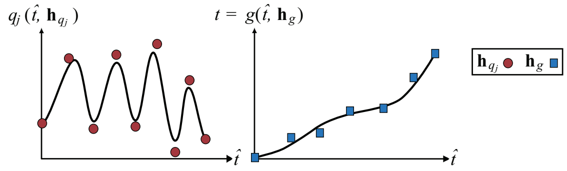

Figure 2: *Left: A joint DOF function plotted in warped time domain. B-spline coefficients* $\mathbf{h}_q$ *determine the joint angle values through* $\hat{t}$*. Right: Time transformation defined as a monotonic, non-linear function g from the warped time* $\hat{t}$ *to actual time t. B-spline coefficients* $\mathbf{h}_g$ *determine the shape of the time transformation.*

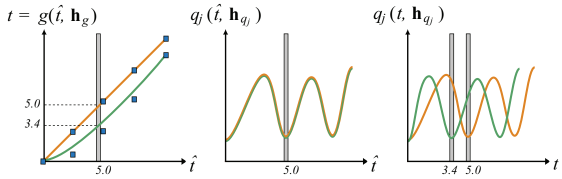

Figure 3: *Left: Time transformation from an initial identity function shown in orange, to an arbitrary non-linear function shown in green. The gray bar indicates a constraint that occurs at* $\hat{t} = 5.0$*. Middle: A joint DOF function plotted in warped time domain. Time transformation does not change the timing of the constraint in warped time domain. Right: The same joint DOF shown in actual time domain, where the actual timing of the constraint changes to* $t = 3.4$*. Note that the length of the motion is also shortened by the time transformation.*

4. Optimal constraints

The standard spacetime optimization usually formulates a constraint as a function of motion variables $\mathbf{q}$, with a time instance t and spatial constant $\mathbf{p}$ as function coefficients. For example, a foot position constraint $\mathbf{C}_p(\mathbf{q};\mathbf{p},t)$ would enforce the position of the foot at the desired location $\mathbf{p}$ at frame t. This framework allows the user to determine the timing and spatial coefficients of each constraint, as an intuitive way to provide controls to the user. However, manually specifying these constraints becomes difficult, if at all possible, when the interactions between characters take place.

One obvious approach to solve this problem is to directly optimize the timing and the spatial coefficients of each constraint. However, this approach leads to a highly discontinuous search space. Optimization with both motion variables and timing would almost always prefers to modify the motion variables, leaving the timing of the constraints unchanged. For example, consider changing the timing of a running sequence so that the foot constraint occurs slightly earlier. For this modification to be feasible while reducing the energy, the entire dynamics preceding and following the modified constraint needs to be updated in a global manner. Consequently, there is a huge energy barrier between the original state and a new state that satisfies all constraints with lower energy. This large, discontinuous gap in the search space is impossible for any derivative-based optimization to get across, effectively preventing the optimization from ever changing the timing of a constraint.

To address this problem, we design a new framework by augmenting spacetime optimization with a time-warping parameterization, in which both the timing of the motion and the timing of the constraints are optimized in synchrony. This coupling is crucial to avoiding large energy barriers while improving the objective function without violating a large number of constraints. A time-warping parameterization enables us to simultaneously optimize all motion variables $\mathbf{q}$, the spatial coefficients of the constraints $\mathbf{p}$, and the timing of both the motion and constraints, while maintaining the temporal relation between the motion and constraints.

4.1. Motion representation

We represent a motion as a sequence of body poses through time, using B-spline parameterization. Without time-warping, the value of a j^{th} joint degree of freedom (DOF) at time t is written $q_j(t;\mathbf{h}_{q_j})$, where $\mathbf{h}_{q_j}$ are the coefficients parameterizing the B-spline. In our formulation, we choose to represent the joint DOFs under warped time parameterization $q_j(\hat{t};\mathbf{h}_{q_j})$, and define a monotonic, non-linear function to represent time transformation from the warped time to the actual time $t = g(\hat{t},\mathbf{h}_g)$, where $\mathbf{h}_g$ are the one-dimensional B-spline coefficients describing the non-linear time transformation (Figure 2). We deliberately represent g as an *inverse* time warping function for the ability to vary the length of the actual motion sequence during the optimization. To prevent time from being reversed, we constrain $\mathbf{h}_g$ to ensure the monotonicity of time:

$$g(\hat{t}+1;\mathbf{h}_g) - g(\hat{t};\mathbf{h}_g) > 0 \qquad (1)$$

for all warped time indices $\hat{t}$.

To fully determine a motion sequence, we need to specify both motion variables $\mathbf{h}_{q_1}, \mathbf{h}_{q_2}, \ldots \mathbf{h}_{q_n}$ and the time variables $\mathbf{h}_g$. Note that this representation is largely redundant, as $\mathbf{h}_{q_j}$ can represent all possible joint variation, provided enough control points in the B-spline. However, the coefficients for time transformation ($\mathbf{h}_g$) allow the optimizer to vary the timing of the motion by changing a minimal set of motion variables, without disrupting the relative relationship among different joint DOFs.

4.2. Constraint representation

The constraint representation in our framework is based on the standard spacetime optimization with two modifications. First, the timing of each constraint is specified by a fixed warped time instance $\hat{t}_c$ instead of the actual time. This modification allows us to optimize the actual timing of the constraint ($t_c = g(\hat{t}_c;\mathbf{h}_g)$) by varying the time coefficients $\mathbf{h}_g$. Second, the spatial coefficients of each constraint are solved as free variables in the optimization. We represent a constraint as a function of motion variables and spatial coefficient, with a warped time instance as a function coefficient: $\mathbf{C}(\mathbf{h}_q,\mathbf{p};\hat{t}_c)$. We can visualize the timing changes as constraints travel through time together with joint DOFs that satisfy the constraints. Figure 3 demonstrates how the time transformation affects the timing of a constraint in the actual time domain, but leaves it unchanged in the warped time domain.

There are cases when constraints associated with actual time instances are useful. For example, the user might want the character's hand to reach the cup exactly two seconds after the motion starts. Our generative constraint representation can easily revert to a standard spacetime constraint by simply associating the constraint with an actual time instance: $\mathbf{C}(\mathbf{h}_q,\mathbf{p};t_c)$.

4.3. Environment constraints

The ability to change the timing and spatial coefficients is essential to environment constraints in multi-character animation. Although we only demonstrate our constraint representation on positional constraints, the representation can be applied to any type of environment constraints that enforce a spatial relation between the character and the environment at a particular time instance.

Current spacetime optimization algorithms require the desired footprint locations to be fixed prior to the synthesis process. These models are not suited for synthesis of multi-character motion, as the interactions between characters could change the footprints unexpectedly. Our coupled-warping formulation allows the optimizer to search for the best desired location and timing for a positional constraint $\mathbf{C}_p$:

$$\mathbf{C}_p(\mathbf{h}_q,\mathbf{p};\hat{t}_c) = \mathbf{d}(\mathbf{h}_q;\hat{t}_c) - \mathbf{p} = 0 \qquad (2)$$

where $\mathbf{d}(\mathbf{h}_q;\hat{t}_c)$ evaluates the position of the relevant handle on the character's body at $\hat{t}_c$ and $\mathbf{p}$ is the desired location of the handle. Note that varying $\mathbf{h}_g$ does not affect the value of $\mathbf{C}_p$, but changes the timing of $\mathbf{C}_p$ in the actual time domain.

This new constraint specification provides the flexibility to generate optimal motions that vary significantly from the initial state. For example, a single optimal walking cycle can be transformed into arbitrary walking, turning, pivoting, or maneuvers.

4.4. Dynamic constraints

To ensure physical realism, we enforce generalized dynamic constraints on each joint. Similar to environment constraints, we represent a dynamic constraint as a function of motion variables $\mathbf{h}_q$, associated with an warped time instance $\hat{t}_c$: $\mathbf{C}_d(\mathbf{h}_q;\hat{t}_c)$. In general, adjusting timing of environment constraints can easily violate a large number of dynamic constraint distributed over time, immediately requiring very large forces with high energy cost. Consequently, the optimization would almost never modify the time variables $\mathbf{h}_g$. To avoid this problem, we let the dynamic constraints travel through actual time domain in synchrony with environment constraints as time variables $\mathbf{h}_g$ vary.

We use the dynamics formulation described by Liu et. al. [LHP05]. Here we provide a brief summary of this formulation. A dynamic constraint enforces the Lagrangian dynamics that includes the effect of external and internal forces at each joint DOF j.

$$\sum_{i\in N(j)} \frac{d}{dt}\frac{\partial T_i}{\partial \dot{q}_j} - \frac{\partial T_i}{\partial q_j} - Q_{g_j} - Q_{c_j} - Q_{s_j} - Q_{t_j} - Q_{m_j} = 0 \quad (3)$$

where T_i indicates the kinetic energy of i^{th} body node, $N(j)$ is the set of body nodes in the subtree of joint DOF q_j, and the first two terms together represent the generalized net force at DOF q_j. Q_{m_j} is the force generated by muscles, Q_{g_j} is the generalized gravitational force, Q_{c_j} is the generalized ground contact force, Q_{s_j} is the generalized elastic force from the characters' shoes, and Q_{t_j} represents an aggregate spring force from the passive elements around joint q_j, such as tendons and ligaments. It is crucial to model the accurate impedance parameters for the passive elements, as passive forces Q_{t_j} significantly affect the realism of the resulting motion. The process of computing these values from a motion capture sequence is described in [LHP05]. Since there are no muscles or tendons around root DOFs, the root motion of the character is completely determined by the external forces Q_{g_j}, Q_{c_j}, and Q_{s_j} ([LHP05]).

Dynamic constraints introduce two sets of additional DOFs in the optimization: generalized muscle forces Q_m and ground contact forces λ in Cartesian coordinates.

4.5. Optimization

We measure effort in terms of muscle force usage by summing the magnitudes of the forces at all joint DOFs j over the warped time domain:

$$E = \sum_j \sum_{\hat{t}} \alpha_j \|Q_{m_j}(\hat{t})\|^2 \quad (4)$$

where the weights α represent the relative usage preference for each joint. These weights are also automatically determined from the motion capture data as described in [LHP05].

In summary, we formulate an optimization for solving a motion sequence $\mathbf{x} = \{\mathbf{h}_q, \mathbf{h}_g, \mathbf{p}, Q_m, \lambda\}$, which minimizes the muscle force usage E while satisfying environment constraints $\mathbf{C}_p$, dynamic constraints $\mathbf{C}_d$, and interaction constraints $\mathbf{C}_i$. Interaction constraints are defined based on specific scenarios. We will detail three examples in section 6.

$$\min_{\mathbf{x}} E(\mathbf{x}) \qquad \text{subject to} \qquad \begin{cases} \mathbf{C}_p = 0 \\ \mathbf{C}_d = 0 \\ \mathbf{C}_i = 0 \end{cases} \quad (5)$$

5. Motion composition

The flexibility of the framework described in the previous section works well with a short, single-character motion, but scales poorly when extended to a large problem with multiple characters. Although researchers have partly addressed these issues with spacetime windows [Coh92] and wavelet representation of DOFs [LGC94], these methods did not apply to multi-character motion. To solve such problems, we describe a generalization of the spacetime windows together with the continuation-based optimization strategy.

We view the process of synthesis of complex motions as an optimal composition of smaller optimal sequences. The user selects motion clips from a small set of primitive motion date set and specifies the important constraints to achieve desired interaction. Our framework solves for an optimal motion that carries out the interaction in a physically realistic manner. The main challenges of solving large multi-character motions lie in slow convergence rates and large number of local minima, causing the optimization to halt close to the initial state. In our experience, it is virtually impossible to solve such large problems with a single, simultaneous optimization over all variables.

We describe a framework that allows us to both reduce the number of unknowns and avoid local minima. Instead of optimizing all unknowns simultaneously, we use a cyclical sequence of smaller optimizations, which we refer to as a *schedule*, designed for a specific problem. For example, an adversarial chase scenario determines a schedule different from the schedule of a collaborative scenario where characters work towards a common goal. Our strategies use a combination of block coordinate descent and continuations. We outline these basic strategies in this section, and describe a number of useful schedules and their applications in the next section.

5.1. Block coordinate descent

This optimization strategy is frequently used for large optimization problems [Coh92]. The idea is to optimize with respect to a block of coordinates (or unknowns), while holding the remaining unknowns fixed. The active block of coordinates varies during the optimization. As long as the blocks cycle over all unknowns, this process will converge to a local minimum. In our framework, we use the results from the previous optimization to construct interaction constraints that need to be satisfied in the current optimization. In the two-character animation, the interaction constraint for character *A*'s optimization depends on the earlier optimization for character *B*, and vice versa. We can select the blocks by their spatial or temporal relations. For example, we can set unknowns to be only the DOFs of a specific character, treating the motion of all other characters as constant. Similarly, we can restrict the optimization to a smaller part of the whole animation. Whichever strategy is chosen, we interleave the blocks so that all unknowns are optimized. Coordinate-descent strategies always minimize the same objective function and the same constraints. In our framework, we also vary the objective function and constraints by continuation strategies.

5.2. Continuations

The idea of continuations is to solve a sequence of different problems that, in the limit, smoothly approach the objective function. For example, if a positional constraint is very difficult to satisfy, we may replace it with a spring of rest-length zero, and slowly increase the spring coefficient during the optimization. In the limit, we will strictly enforce the original positional constraint.

It is also crucial to apply block coordinate descent in conjunction with continuation to solve tightly coupled multi-character motion, such as a mother and a child walking hand-in-hand. If we were, for example, to first solve for mother's motion and then for the child's motion, the mother would change her entire motion to accommodate for the child's movement; the child would then change its motion minimally, resulting in undesirable behavior where the mother unrealistically tracks the child's hand movement. By using continuations, such as an increasingly tightening spring constraint between two hands, we can still use the character-decoupling block optimizations, without biasing one subproblem over another due to the ordering. This approach allows us to decouple all DOFs in virtually any grouping we choose. As long as we have a way of smoothly introducing the coupling constraints, in the final optimization they will revert to the true problem formulation.

6. Results

In our implementation, we used only three basic input motion clips: a walk cycle, a run cycle, and a child's walk cycle. We intentionally restricted ourselves to this small set of input motions to demonstrate the versatility of this approach. From this very limited input motion data set (5 second long), our framework was able to compose complex multi-character motions with interesting interaction in many different scenarios. The length of the clips ranges from 40 to 60 frames at 30 fps. We use SNOPT [GSM96] to solve spacetime optimization problems on a 2Ghz Pentium 4 machine. Each optimization takes approximately 8 to 20 minutes.

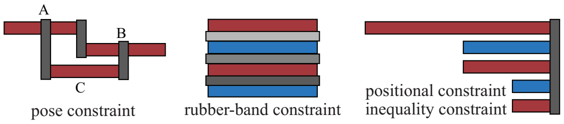

Figure 4: *Three different schedules. Left: Time-layered schedule. The gray bars indicate the pose constraints at transition points. Middle: Constrained multi-character schedule. The motions of two characters (red and blue bars) are tightly coupled by a continuation constraint. The influence of the constraint is indicated in grayscale. Right: Decreasing-horizon optimization. Two characters plan their motions according to the previous movement of the opponents.*

6.1. Time-layered schedule

This schedule is deployed for sequentially composition of small problems. First, we solve for two abutting optimizations A and B, and then we solve the optimization C that straddles across the transition from A to B. Each optimization has pose constraints obtained from the adjacent blocks. Repeated optimizations of this schedule converge to the optimal solution over the interval of A and B, thus effectively using smaller interval optimizations to solve for larger intervals (Figure 4).

Example: Transition synthesis. Since a complex motion is composed of smaller sequences that can be fundamentally different, there is a large discontinuity between the two sequences. Given two example sequences, our goal is to generate a realistic transition between them. First, we initialize the problem by connecting the two motions: we determine the average pose between the start and the end of the two sequences, and then solve two optimization problems (A and B) so that both motions smoothly lead to the average transition pose. We then remove the transition pose and solve for the overlap (C). The transition pose is only used for initializing the problem and has little effect on subsequent optimization. The video shows the synthesized transition between walking and running where the character correctly dips down and leans forward before reaching the running speed. This example was completed by three optimizations. The total computation time took 38 minutes.

6.2. Constrained multi-character schedule

When the motion of two characters is mutually constrained, we employ a schedule that alternates between optimizing each character's motion. We also use a continuation strategy to represent the constraint connecting the two characters. To find the optimal constraints for both characters, we need to allow each character to slowly adjust its motion according to the behavior of the other character.

Example: Hand in hand walking. When solving for the walk cycle with two people holding hands, we start out from

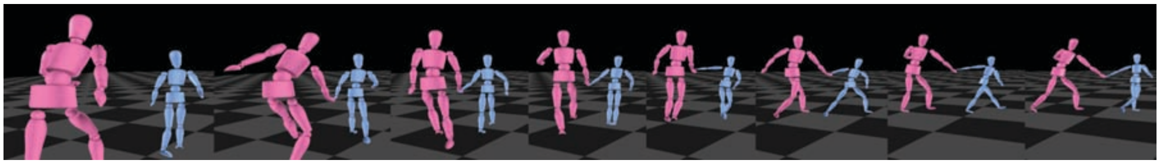

Figure 5: *Solving for the mother's motion more frequently than the child's motion would result in a final motion where the mother does all the work, while the child adjusts minimally.*

two people walking independently without any body contacts. We first optimize for the motion of one person and treat the other person's motion as constant. During the optimization, we apply a continuation-like "rubber-band" that pulls the hand of the character towards the other one. After every scheduled cycle, we increase the stiffness coefficient of the rubber-band by 100, pulling the characters further together (Figure 4). At convergence, the soft constraint is satisfied. Since we exert a spring force on the characters, the resulting force exchange across the hand constraint is dynamically accurate. By manipulating the order of the schedules, the user can control which character dominates the motion. For example, solving for character A more frequently than character B would result in the motion where character A does all the work, while character B adjusts minimally (Figure 5). The example shown in the accompanying video took 8 scheduled cycles before convergence. The total computation time took 85 minutes.

6.3. Decreasing-horizon optimizations

Spacetime optimization is inherently unsuited for creating animations where characters react to unexpected events, because all constraints are known *a priori* and the character can always plan well ahead. When two people are planning their motions in an adversarial setting, at every instance in time, each character adjusts to the latest movement of the opponent. We approximate this process by interleaving the solutions for both opponents, while consistently decreasing the reaction time for each character by reducing the optimization interval.

Example: Simulations of optimal adversary behavior. We synthesized several scenarios of a tackler and a target. The tackler's goal is to hit the torso of the target at specific velocity, and position his hands around the torso. The target's goal is to avoid the tackler at the collision time. We represent this as an inequality constraint that prevents the target from being inside the bounding sphere of the tackler. We use one running cycle and one walking cycle from the dataset as the initial motions for the tackler and the target. Each scheduling cycle, we choose to half the interval of time over which we are optimizing (Figure 4). First the tackler would find the optimal motion for tackling the target. Then the target character would solve for the optimal avoiding motion based on the tackler's final position, only we would solve for the last half of the animation. Then the tackler would plan the optimal motion based on the new avoidance strategy, only during the last half of the animation. This process would continue until a sufficiently small time interval. In the video, we optimize each character for two or three cycles depending on the scenarios. Effectively, we have created a coarse discretization of the optimal planning process. We show the two outcomes of the tackle by letting different characters have the last chance to adjust the motion. We also show an animation where the target avoids two tacklers coming from different directions.

7. Conclusion

We have described a framework for composing complex optimal motions from optimal motion building blocks. We introduce two simple but powerful extensions to spacetime optimization that make this possible. First, we exploit a time-warped parameterization to optimize the timing and the spatial coefficients of the constraints. Second, we describe how a block-coordinate descent approach can be extended to solve tightly coupled multi-character motions by use of continuations. The combination of these two techniques greatly expands the applicability of the spacetime optimization, including optimal transitions between different motions and collaborative and adversarial optimal motions for multiple characters.

This work presents only the first attempt at solving complex multiple character animations. We have by no means described a general framework that can deal with any situation, but merely shown that, with thoughtful structuring of the problem, it is possible to solve these large problems using spacetime optimizations. An important open problem is to automatically determine the appropriate schedule from high-level user specifications. Our current implementation uses inequality constraints to prevent self-penetration of a character. As the complexity and the number of the characters increase in the scenario, a more sophisticated collision detection method becomes necessary to our system. The intensive computational time also undermines the applicability of our algorithms. An adaptive optimization framework could potentially result in a more efficient algorithm for motion synthesis of a large group of characters.

References

[AF02] ARIKAN O., FORSYTH D. A.: Synthesizing Constrained Motions from Examples. *ACM Trans. on Graphics 21*, 3 (July 2002), 483–490.

[Bet01] BETTS J. T.: *Practical methods for optimal control using nonlinear programming*. SIAM, 2001.

[BH00] BRAND M., HERTZMANN A.: Style machines. *Proceedings of SIGGRAPH 2000* (July 2000), 183–192.

[BW95] BRUDERLIN A., WILLIAMS L.: Motion signal processing. In *Computer Graphics (SIGGRAPH 95 Proceedings)* (Aug. 1995), pp. 97–104.

[Coh92] COHEN M. F.: Interactive spacetime control for animation. In *Computer Graphics (SIGGRAPH 92 Proceedings)* (July 1992), vol. 26, pp. 293–302.

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

[FvdPT01] FALOUTSOS P., VAN DE PANNE M., TERZOPOULOS D.: Composable Controllers for Physics-Based Character Animation. In *Proceedings of SIGGRAPH 2001* (August 2001).

[Gle97] GLEICHER M.: Motion Editing with Spacetime Constraints. *1997 Symposium on Interactive 3D Graphics* (Apr. 1997), 139–148.

[Gle98] GLEICHER M.: Retargeting motion to new characters. In *Computer Graphics (SIGGRAPH 98 Proceedings)* (July 1998), pp. 33–42.

[GMHP04] GROCHOW K., MARTIN S. L., HERTZMANN A., POPOVIĆ Z.: Style-based Inverse Kinematics. *ACM Trans. on Graphics* (Aug. 2004), 522–531.

[GSM96] GILL P., SAUNDERS M., MURRAY W.: *SNOPT: An SQP algorithm for large-scale constrained optimization*. Tech. Rep. NA 96-2, University of California, San Diego, 1996.

[HP97] HODGINS J. K., POLLARD N. S.: Adapting Simulated Behaviors For New Characters. *Proc. SIGGRAPH 97* (1997), 153–162.

[KGP02] KOVAR L., GLEICHER M., PIGHIN F.: Motion Graphs. *ACM Trans. on Graphics 21*, 3 (July 2002), 473–482. (Proceedings of ACM SIGGRAPH 2002).

[KKK*02] KUFFNER JR. J. J., KAGAMI S., KISHIWAKI K., INABA M., INOUE H.: Dynamically-stable Motion Planning for Humanoid Robots. vol. 12, pp. 105–118.

[KKKL94] KOGA Y., KONDO K., KUFFNER J., LATOMBE J.-C.: Planning Motions with Intentions. In *Proc. SIGGRAPH 94* (July 1994), pp. 395–408.

[LC95] LIU Z., COHEN M. F.: Keyframe motion optimization by relaxing speed and timing. In *Computer Animation and Simulation Eurographics Workshop* (1995).

[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 Trans. on Graphics 21*, 3 (July 2002), 491–500.

[LGC94] LIU Z., GORTLER S. J., COHEN M. F.: Hierarchical spacetime control. In *Computer Graphics (SIGGRAPH 94 Proceedings)* (July 1994).

[LHP05] LIU C. K., HERTZMANN A., POPOVIC Z.: Learning physics-based motion style with nonlinear inverse optimization. *ACM Trans. Graph 24*, 3 (2005), 1071–1081.

[LP02] LIU C. K., POPOVIĆ Z.: Synthesis of Complex Dynamic Character Motion from Simple Animations. *ACM Trans. on Graphics 21*, 3 (July 2002), 408–416.

[LWS02] LI Y., WANG T., SHUM H.-Y.: Motion Texture: A Two-Level Statistical Model for Character Motion Synthesis. *ACM Trans. on Graphics 21*, 3 (July 2002), 465–472.

[PB02] PULLEN K., BREGLER C.: Motion Capture Assisted Animation: Texturing and Synthesis. *ACM Trans. on Graphics 21*, 3 (July 2002), 501–508.

[PR01] POLLARD N. S., REITSMA P. S. A.: Animation of humanlike characters: Dynamic motion filtering with a physically plausible contact model. In *Yale Workshop on Adaptive and Learning Systems* (2001).

[PSE*00] POPOVIĆ J., SEITZ S. M., ERDMANN M., POPOVIĆ Z., WITKIN A. P.: Interactive manipulation of rigid body simulations. *Proceedings of SIGGRAPH 2000* (July 2000), 209–218. ISBN 1-58113-208-5.

[PW99] POPOVIĆ Z., WITKIN A.: Physically Based Motion Transformation. *Proceedings of SIGGRAPH 99* (Aug. 1999), 11–20.

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

[Rey87] REYNOLDS C. W.: Flocks, Herds, and Schools: A Distributed Behavioral Model. In *Proc. SIGGRAPH 87* (July 1987), vol. 21, pp. 25–34.

[RGBC96] ROSE C., GUENTER B., BODENHEIMER B., COHEN M.: Efficient generation of motion transitions using spacetime constraints. In *Computer Graphics (SIGGRAPH 96 Proceedings)* (1996), pp. 147–154.

[SHP04] SAFONOVA A., HODGINS J. K., POLLARD N. S.: Synthesizing Physically Realistic Human Motion in Low-Dimensional Behavior-Specific Spaces. *ACM Trans. on Graphics* (Aug. 2004).

[vL95] VAN DE PANNE M., LAMOURET A.: Guided optimization for balanced locomotion. In *Computer Animation and Simulation '95* (Sept. 1995), Eurographics, Springer-Verlag, pp. 165–177.

[WH97] WILEY D. J., HAHN J. K.: Interpolation Synthesis of Articulated Figure Motion. *IEEE Computer Graphics & Applications 17*, 6 (Nov. 1997), 39–45.

[WK88] WITKIN A., KASS M.: Spacetime constraints. *Proc. SIGGRAPH 88 22*, 4 (August 1988), 159–168.

[WP95] WITKIN A., POPOVIĆ Z.: Motion Warping. *Proc. SIGGRAPH 95* (Aug. 1995), 105–108.

[YKH04] YAMANE K., KUFFNER J. J., HODGINS J. K.: Synthesizing animations of human manipulation tasks. *ACM Trans. on Graphics 23*, 3 (Aug. 2004), 532–539.

[ZH99] ZORDAN V. B., HODGINS J. K.: Tracking and modifying upper-body human motion data with dynamic simulation. In *Proc. CAS '99* (September 1999).

[ZMCF05] ZORDAN V. B., MAJKOWSKA A., CHIU B., FAST M.: Dynamic response for motion capture animation. *j-TOG 24*, 3 (July 2005), 697–701.