

Dynamic Response for Motion Capture Animation

Victor B. Zordan, Anna Majkowska, Bill Chiu, Matthew Fast*
Riverside Graphics Lab
University of California, Riverside

Abstract

Human motion capture embeds rich detail and style which is difficult to generate with competing animation synthesis technologies. However, such recorded data requires principled means for creating responses in unpredicted situations, for example reactions immediately following impact. This paper introduces a novel technique for incorporating unexpected impacts into a motion capture-driven animation system through the combination of a physical simulation which responds to contact forces and a specialized search routine which determines the best plausible re-entry into motion library playback following the impact. Using an actuated dynamic model, our system generates a physics-based response while connecting motion capture segments. Our method allows characters to respond to unexpected changes in the environment based on the specific dynamic effects of a given contact while also taking advantage of the realistic movement made available through motion capture. We show the results of our system under various conditions and with varying responses using martial arts motion capture as a testbed.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation I.6.3 [Simulation and modeling]: Applications

Keywords: motion capture, character motion, motion control, physical modeling, animation

1 Introduction

Although motion capture animation has become prevalent in the computer graphics industry, the general adaptation and re-use of the data is limited by the available techniques for editing and modifying data in controllable ways. Under certain circumstances for example creating a believable response following an unexpected impact, modifications are particularly difficult to generate because they entail making careful changes to a characters' body accelerations. Changes to motion trajectories must appear natural or viewers will perceive the motion as flawed, as recent perception studies support [Reitsma and Pollard 2003; Wang and Bodenheimer 2004]. Further, in online applications, the hand-crafting of individual edits (a priori) is at odds with the goal of maintaining interactivity. Off- or online, motion capture modifications that mimic dynamic response are highly desirable because they allow characters to react and interact with their environment, upholding and promoting consistency and richness in generated imagery.

*email: vbz/annam/bill/mfast@cs.ucr.edu

Copyright © 2005 by the Association for Computing Machinery, Inc. Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions Dept, ACM Inc., fax +1 (212) 869-0481 or e-mail permissions@acm.org.
© 2005 ACM 0730-0301/05/0700-0697 \$5.00

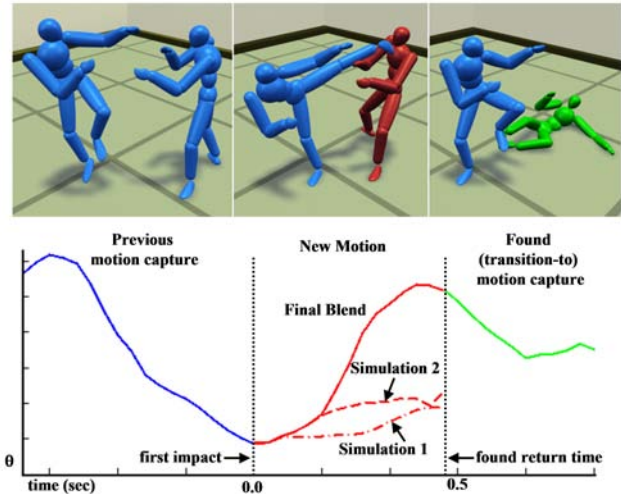


Figure 1: An example output from our system. The *red* synthesized (blend) motion fills in the gap between the *blue* and *green* motion capture segments played for the character being kicked. The synthesis and motion selection (of the *green* clip) are computed automatically by our system given the interaction which is about to take place in the left panel above. The plot shows the Y-value of the waist joint before, during, and after the interaction as well as two intermediate trajectories of simulated motion used by our system as described below.

Physical models are commonly used to add responsivity to characters by generating modifications through collision forces applied to a dynamic simulation that takes over during contact for motion capture (or keyframe motion.) In many games and movies, when impacts are detected, motion-capture driven characters are turned over to a dynamics engine to accept the impulse. But to date, the reaction of the character is most often a passive one which dissipates over the period just following the impact. The character, if possible, resumes following the motion sequence [Oshita and Makinouchi 2001; Zordan and Hodgins 2002] or, most often, goes limp and falls to the floor like a “ragdoll,” presumably rendered unconscious after the impact. The problem with this method is that once the effects of an impact are over, there are no general schemes to move the simulation in a meaningful way and return the simulation to motion capture control. In this paper, we present a technique which automatically computes a dynamic reaction to an unanticipated impact and returns the character to an available motion from a capture repository for the remainder of the dynamic response to the interaction.

A key insight in our approach is that dynamics is often only needed for a short time (a *burst*) during which the effects of a collision or impulse change the character's state. After that, the utility of the dynamics decreases and due to the lack of good behavior control the resulting motion of the simulation becomes less reliable than a motion capture sequence under similar conditions. To take advantage of this notion, we propose a complete system which selects a reasonable clip from a motion capture repository for soon after the interaction (the *green* clip in Figure 1) and computes new motion (shown *red* in Figure 1) that allows the character to respond to the

physical conditions and prepare for its newly determined, upcoming animation. Relying on a motion library with a modest set of real human reactions, in the proposed approach, we find good matches based on a preliminary, “naive” response trajectory computed using forward simulation (labeled as “Simulation 1” in the figure). A second, more active response (labeled “Simulation 2”) is generated using the knowledge of the selected match to respond based on (and in anticipation of) the found data. Finally, the system reaches the desired example by interpolating the dynamic motion (labeled “Final Blend”) so that it matches the desired state at the end of the generated segment.

Our system includes two critical components: a search engine that compares the initial simulated response with reaction segments from a motion library and a joint-torque controller that actuates the simulation to reach the desired posture, leading to a character that appears “conscious.” Based on these components, we draw from and contribute to two independent areas of previous work, building on reported techniques while combining them to meet our specific needs. Our search engine is similar to those suggested previously for reordering motion capture data but is crafted to give importance to the main bulk of the mass (i.e. the trunk) as well as the ground-support body, targeting relevant dynamic effects and contact constraints in the motion comparison. While our control approach is not novel, our usage of simulation to fill in the burst immediately following a forceful interaction is quite different than others reported to date. When considered as a whole, our complete approach contributes directly to the growing area of motion capture editing techniques which we describe next.

2 Background

Recent innovations in motion capture research focus on topics such as editing/retargeting, parameterizing motion based on behavior and style, reordering motion samples for continuous motion, as well as creating believable transitions. Because there are too many individual contributions to mention all in detail, we highlight only the most relevant to the approach described here. Following the seminal work in video textures [Schödl et al. 2000], search for smooth (and controllable) reordering of motion capture sequences exploded in 2002 with many animation researchers suggesting methods for finding and assessing good motion capture to motion capture transitions [Arikan and Forsyth 2002; Kovar et al. 2002; Lee et al. 2002; Li et al. 2002; Metoyer 2002]. Our search for matches is similar to these efforts except that we explicitly compare simulated and motion capture sequences, basing our search on a synthesized, physically based trajectory. Other researchers have proposed methods to use physical models to modify motion capture data to make energy-efficient transitions between motion sequences [Rose et al. 1996] and to transform motion based on user goals while minimizing physical characteristics [Popović and Witkin 1999; Pollard 1999; Pollard and Behmaram-Mosavat 2000].

A smaller number of researchers present approaches which combine physics and motion capture with goals similar to our own (i.e. to create responsive characters). Oshita and Makinouchi [2001] use inverse dynamics computed from data and add external inputs by modifying character velocity directly. Zordan and Hodgins [2002] suggest tracking motion capture data with controlled forward dynamics for characters, allowing the passive dynamics to take over during impacts. Komura et al. [2004] animate reactions for locomoting bipeds by modifying motion to maintain balance in response to external influences. In each of these works, the results reported focus on balanced motion and show limited types of interactions. Conversely, we propose a more flexible approach, showing results with heavy and light contact leading to responses including stepping, jumping, falling, and rolling.

Other closely related work presented by Shapiro et al. [2003] and Mandel [2004] introduces hybrid methods for moving from

kinematic (motion capture-driven) models to dynamics and back again. Shapiro and colleagues build off of a supervisory control scheme [Faloutsos et al. 2001] to create a general framework for moving between motion representations. Unlike their work, we focus our effort on a specialized framework to manage response including a customized search method. In addition, in our work, we intend to move from and to motion capture with a minimal, focused simulation phase. Mandel suggests specialized controllers for falling and recovery which take over once an impact is detected. In our framework, we give preference to the motion capture animation and use the simulation only as a vehicle for creating an immediate response and subsequently returning back to motion capture. We propose a distinct shift away from building specialized controllers and instead treat the controlled simulation as a sophisticated interim system with emphasis and explicit trust placed on the motion capture examples.

Due to the application-driven nature of this research, it is also important to compare our work with that seen in commercial products. In particular, Natural Motion’s software Endorphin [Natural-motion 2005] is closely related to the work described here. Endorphin 2.0 primarily focuses on simulation and supports general transitions from motion capture to simulation. However, currently this software only matches a final *static* simulation pose with motion capture in order to allow the simulation to return to motion capture. This implies that their transition happens only after the system has settled and has gone through a relatively long simulation phase compared to the approach we propose. In addition, through the second simulation pass which is unique to our system, we provide a prediction for the upcoming motion and therefore our controller is able to act in accordance with the desired reaction encapsulated in the upcoming motion example.

3 Motion Selection

In the first step of creating a response, we find a transition-to motion capture sequence from our repository to be used both as a reference for the response and for animation following the interaction. To do this, the system integrates the dynamics of a simulated version of the interaction over a short interval to generate a purely simulated, physically plausible character trajectory that acts in response to the interaction forces present. Then, we compare the simulated data with sequences in our motion library to find the desired transition-to sequence as well as the precise time and root transformation that aligns that found sequence with the simulated trajectory.

In our search, each motion sample (frame) is defined as a vector $(p_1, \theta_1, \dots, p_n, \theta_n)^T$ with the positions p_i and orientations θ_i of each of the n body parts. We compare *frame windows*, similar to Kovar et al. [2002], between the simulation and test motion capture sequences. Within windows, the distance function for pairs of frames f_{1i} and f_{2i} is computed as the weighted sum of distances between the positions and orientations of matching body parts. And the distance between two windows is found as the sum of distances between the corresponding frame pairs. To remove coordinate-frame differences, we normalize by aligning the roots in the start frames of each window. The distance D between windows $W_1 = \{f_{1i}\}_{i=s}^e$ and $W_2 = \{f_{2i}\}_{i=s}^e$ is then defined as

$$D(W_1, W_2) = \sum_{i=s}^e w_i \left(\sum_{b=1}^n w_{pb} \|p_b(f_{1i}) - p_b(f_{2i})\| + w_{\theta b} \|\theta_b(f_{1i}) - \theta_b(f_{2i})\| \right)$$

where w_i is the window weight, a quadratic function which returns the highest value for the start frame and decreases for subsequent frames in the window. The weights w_{pb} and $w_{\theta b}$ scale the linear and angular distances for each body b . To capture the dynamic

properties existing in the simulated motion, we assign high weights to the trunk parts. We use lower ones for the limbs, as we found differences in the positions of the limbs can be more easily modified when computing the transition motion. Additionally, to reduce the problem of sliding ground contact, we compute the center of mass at the first frame of each window and assign a high weight to the closest ground-support body for the duration of the window. To increase the efficiency of the search function we pre-process our database to find unique frames (based on a difference tolerance) and only compare windows starting with these frames. Once we find the closest window, we test all possible windows in the surrounding interval to find the best match.

4 Transition Motion Synthesis

Once we find the desired motion capture sequence to play next, we must generate motion to fill in the gap between the beginning of the interaction and the time before the found clip is played. We compute this transition motion with two goals in mind: react in a physically plausible manner which is consistent with the found motion and meet the desired state as closely as possible. Our system finds the transition in two steps based on these goals. We create a second simulated motion, now using a joint controller which is informed of the upcoming motion sequence and uses this motion sequence in its desired state calculation. The internal joint-torque actuation keeps the character “alive” across the transition but lacks the knowledge to make directed coordinated adjustments to the motion. Thus, we layer this simulated motion over the transition-to clip and perform an interpolation between them to create the final motion.

Over the duration of the transition, the torque controller follows a blended sequence of joint angles, $\theta_d(t)$, from the previous motion capture segment to the next. The controller uses an *inertia-scaled* PD-servo [Zordan and Hodgins 2002] at each joint to compute torques as

$$\tau = I(k_p (\theta_d(t) - \theta) - k_v (\dot{\theta}))$$

where θ and $\dot{\theta}$ correspond to the simulation’s current joint angles and velocities. I is the inertial matrix of the outboard body for each joint. The desired sequence to be tracked by the controller, $\theta_d(t)$, is generated on the fly by blending the intermediate postures from the two motion capture sequences before and after the transition. Spherical linear interpolation (slerp) with an ease-in/ease-out weighting parameter blends the desired joint angles smoothly over the computed length of time for the transition interval. A single, hand-selected gain value for each of k_p and k_v is held fixed for the entire transition. (After an initial tuning phase, we found satisfying results across many examples without needing to modify these selected values.) While this controller does not include joint limits, it is expected to make relatively small adjustments from beginning to end configuration and would likely break for poor matches.

Through the use of this controller, the simulation acts in correspondence with the desired motion, moving toward the end body posture over the course of the transition gap and making the character more lifelike (compared to ragdoll simulation). However, timing is critical to make the character’s action appear realistic. To control the time before the character begins its active response to the impact, we introduce a deliberate delay before changing the desired values to match the transition-to clip. With a small or no delay, the character appears to be anticipating the contact and with a large one, the character responds more slowly, as if stunned. With the proper delay time (0.1-0.2 s, which falls in the range of real human response delays), we found that the character’s response to the interaction became much more believable. Since the controller acts while the impact is happening, our system creates both active and passive dynamic response simultaneously.

From the dynamics we generate a completely physical response motion and blend this motion into the transitioned-to motion capture data using interpolation to remove remaining disturbances. We interpolate linearly the root node offset, distributing the root position error across the sequence. For rotations, our system interpolates by slerping quaternions, again using a simple linear weighting in time across the transition. While this approach is extremely simple, it produced more acceptable results than the more elaborate minimizations schemes we investigated.

5 Implementation and Results

The character we chose includes 51 degrees of freedom (DOFs): three each at the ankles, knees, hips, waist, back, shoulders, elbows, wrists, and neck plus six DOFs at the root. We selected a skeleton with the proportions between the two recorded subjects, both medium build with heights of 5’ 11” and 6’ 2”. (The motion comparison search intermittently selects data from each actor and does not show preference in selecting matches across the trained actors.) The dynamic simulation and general collisions are generated using Open Dynamic Engine (ODE) [Smith 2005]. We chose 3D ball and sockets for each joint of the articulation so that we could easily switch between motion capture and dynamics. Mass and inertial parameters are generated by ODE derived from basic geometric models for the bodies similar to those seen in the figures. These same models are used for contact with the ground and between characters with the exception that we wrote a custom collision handler for the foot/ground contact. Our model uses a penalty method with Coulomb friction plus additional friction derived from rotation to impede free spinning movement between the foot and the ground. ODE handled the rest of the collisions through contact constraints which allow the user to control parameters such as friction and the extent to which constraints are hard or soft. These constraint parameters allowed some compliance in the characters’ interactions with each other and the ground. Note, no collisions were computed during the final blend and so, some inter-penetration is visible in the final animation results.

To create believable exchanges between characters, we found that heavy impact-based interactions require simulation of both the recipient and the deliverer of the impacts generated. Simply following motion capture for the hitter lead to irresistible (and unrealistic) collision forces. But, simulating both characters simultaneously left us with managing the additional effect of the response for the attacker. Conveniently, we observed that simply following the completely simulated motion for a small number of frames after the impact and then blending back to the same original motion through interpolation lead to a convincing attack motion. With a quick but visible perturbation, the hitter responds to the impact and then continues with unmodified completion of the attack motion.

Motion capture reactions. Including a wide variety of believable responses in the motion library is important for producing high-quality results. To create our library, we recorded two actors sparing and practicing kenpo and judo strikes and recoveries using optical motion capture equipment. We mapped the recorded data to our character with a physically based model [Zordan and Horst 2003]. Our full repository contains segmented strikes and pushes as well as various reactions based on contact varying from light to heavy. While reactions to full contact strikes were omitted during recording to avoid injury, strong pushes were performed without harm to the actor. Pushes were made from the front, side and back with reactions including balanced recovery that required minimal foot repositioning, recovery requiring a leap or step, and complete loss of balance requiring a forward or backward judo-style somersault. Strikes included various kenpo-style kicks and punches. In total, our library contains 110 single-actor files which ranged from 3 to 10 s each. In the results shown, our system often found close matches between previously unmatched push reactions and strikes.

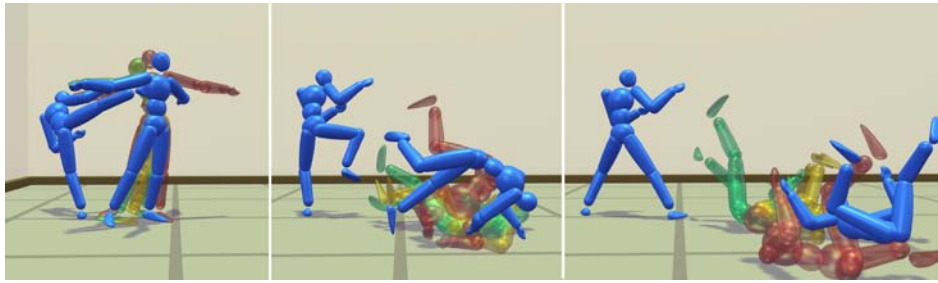


Figure 2: **Range of examples deriving from a single pair of motion clips, only varying the facing direction of the kicked character.**

Using the system described, we are able to create a variety of dynamic responses under various conditions. Figure 2 shows a range of such responses starting from a single pair of motion clips found by simply varying the facing direction of one of the characters. Figures 1 and 3 show several different scenarios and their resulting responses. Some care was taken to set-up the scenarios included in our results so that they show good contact. However, this is for demonstrative purposes only and our system should easily manage many other situations, especially when the contact leads to small disturbances (where the best thing to do is to complete the current motion capture playback, as we see in the attacker in the examples in this paper). For heavier impacts, we also anticipate good performance with the limiting factor being the collision handler. For very large contact forces, ODE’s collision solver (which uses constraints to resolve collisions) lead to undesirable inter-penetrating bodies. Currently, our system does not manage multiple contacts in series which would require a secondary search (or beyond) to find a good transition-to clip. This implies that a one-two punch would likely lead to an undesirable behavior because we do not anticipate the subsequent contact in the motion search. Also, sustained contact, like holds in wrestling, remains unexplored and provides an interesting direction for future work.

Some engineering decisions went into the system described. In particular timing was a tricky aspect of the problem we address. Because we wanted to minimize the time spent in simulation, we limited our motion search to a short time window just after the initial impact from 0.1 to 1.0 s. During operation, the system automatically shrinks this window by increasing the start time for the search (up to a limit of 0.25 s) based on contact to ensure that a match is not found in the time before contact is over (as the dynamic effects have not yet fully changed the state of the character.) The upper limit manages cases with very long or sustained contact. We found that good contact required running the collisions at a simulation time step of about 0.0005 s and so, to speed up the system, we built in an automatic rewind that looks for the first contact at 30 fps and then turns back time to the last motion capture frame, computes the simulation state and begins the simulation at the reduced time step. During run-time, the search was the slowest part of the system, taking about seventy percent of the computation time, and the results shown here were computed at approximately one minute each (without graphics.) While this is admittedly too slow for games, we note as Mandel describes in detail in his thesis [Mandel 2004] that search time can be controlled directly by choosing the number of reaction examples in the search database. And indeed, when we were developing our system, we found searching over a core set of reactions lead to runs much closer to real-time. We did choose to include our entire database of reactions to produce the highest quality motion for this paper and our timing reflects this extreme.

We found ground friction to be another sensitive parameter in our system and chose to set the ground friction in the first naive simulation calculation to 1.0 (a rough surface) but dropped the friction to 0.5 for the second pass to allow foot slippage. While we take some care to find a match which is sensitive to ground support, at times slippage is necessary, for example when both feet are in

sustained contact with the ground before and after the interaction. (This is because our transition takes place between two unmodified motion clips and the system must get the feet to the final location in some manner. Inverse kinematics could alleviate this constraint.) However, unlike pure kinematic foot-skate which is a common flaw in many interpolation-based approaches, the foot slippage of the simulation affects the rest of the dynamics in a consistent, physically based manner. And we found that the lower friction coefficient promotes the simulation slipping in a more visually pleasing manner.

6 Discussion and Conclusions

As our goal we set forth to generate physically plausible response using a reaction-based motion library. While our approach may seem simple, we argue that it takes advantage of the concept of the described *burst* following an impact without the need for a complicated implementation. Conversely, we argue that the components we do include are essential to our method’s success. It may seem tempting to consider performing basic interpolation following the first, naive ragdoll-like simulation trajectory described. But, we stress that the active control is critical for generating realistic, life-like motion. Foremost, our active simulation’s controller follows a changing trajectory (based on the transition-to motion) and makes the character move in a coherent, “conscious” manner. And by moving the simulation toward the desired posture found from the matched data, both an active reaction and passive response to the collision may be computed in a physically based manner simultaneously. This does not yield the same effect if the passive simulation is simply blended with the final motion because the interpolation can happen too slowly (leading to the ragdoll’s lack of direction revealing itself) or too quickly (when the dynamic effect of the contact, and possibly even the contact itself, is not fully realized.)

More subtly, but no less importantly, during simulation the active input yields visible secondary effects, such as physically based slipping which is far more desirable than kinematic-based, motion capture foot skate. Sometimes physically based slipping is appropriate, when a reaction calls for a fast corrective change in stance for example. Likewise, desired changes in momentum can be excited through active control, such as throwing out an arm or a leg for balance for example. These physically based characteristics can change the resulting motion in a manner consistent with the upcoming motion, but are only achievable if the simulated character is actively moving in anticipation of the “chosen” response in addition to passively responding to the collisions happening based on the current interaction.

We present a system that selects a motion capture sequence to follow an impact and synthesizes a transition to this found clip. In the process, the character is made to respond through an active controller which works while the impact forces are being delivered. To produce our results, we create a physically valid response and blend this into the desired transition-to motion. An important contribution

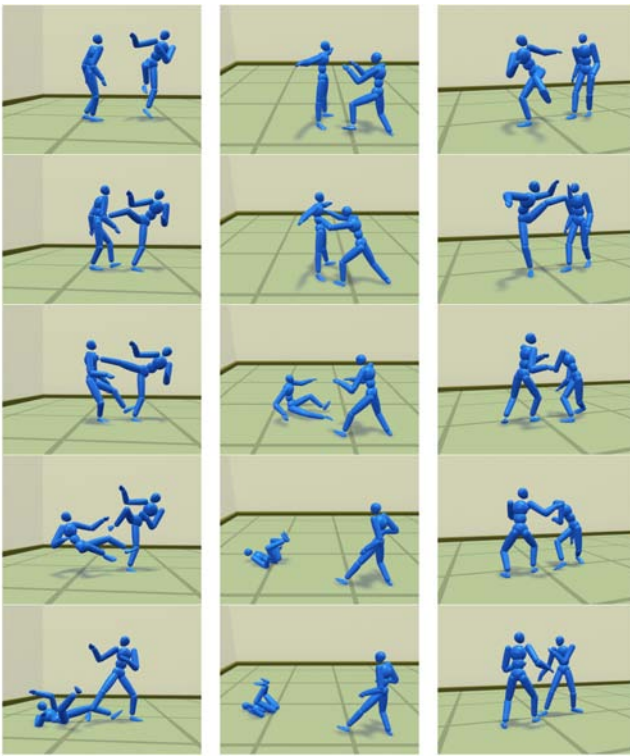


Figure 3: Three example animations (vertically top to bottom) show a range of possible uses for our technique.

of our straightforward technique is the use of a controller which acts in accordance with the upcoming motion. This is a key to avoid an unconscious (ragdoll-like) look for the character during the transition. There are continuing trends toward more physically simulated characters and moving beyond ragdoll-type effects for characters, especially for games. Through the use of the system we describe, a character can accept a hefty blow, recover, and keep fighting in a believable fashion. The commercial game industry has already shown that real-time physics of multiple characters on a mid-range processor is quite manageable with several current titles. Dynamics has until recently been a (computational) bottleneck, but with further advances, physics engines will become commonplace and afford more and more diverse, immersive, and interactive virtual worlds. Through techniques such as the one described here, we hope to continue to move toward more responsive and believable characters for these worlds.

Acknowledgments: The authors would like to thank Wei Xu, Brian Dawson, Dwayne Mason, and James Scarafone of Sony Computer Entertainment America for their support and generous help collecting motion capture data for this paper, as well as Nick Van der Horst as a motion capture talent. We would also like to acknowledge the very helpful discussion and remarks of Nancy Pollard, Christian Shelton, and the SIGGRAPH reviewers that made this paper stronger and more comprehensible.

References

ARIKAN, O., AND FORSYTH, D. A. 2002. Synthesizing constrained motions from examples. *ACM Transactions on Graphics* 21, 3 (July), 483–490.

- FALOUTSOS, P., VAN DE PANNE, M., AND TERZOPOULOS, D. 2001. Composable controllers for physics-based character animation. In *Proceedings of ACM SIGGRAPH 2001*, 251–260.
- KOMURA, T., LEUNG, H., AND KUFFNER, J. 2004. Animating reactive motions for biped locomotion. In *Proc. ACM Symp. on Virtual Reality Software and Technology (VRST '04)*.
- KOVAR, L., GLEICHER, M., AND PIGHIN, F. 2002. Motion graphs. *ACM Transactions on Graphics* 21, 3 (July), 473–482.
- LEE, J., CHAI, J., REITSMA, P. S. A., HODGINS, J. K., AND POLLARD, N. S. 2002. Interactive control of avatars animated with human motion data. *ACM Transactions on Graphics* 21, 3 (July), 491–500.
- LI, Y., WANG, T., AND SHUM, H.-Y. 2002. Motion texture: A two-level statistical model for character motion synthesis. *ACM Transactions on Graphics* 21, 3 (July), 465–472.
- MANDEL, M., 2004. Versatile and interactive virtual humans: Hybrid use of data-driven and dynamics-based motion synthesis. *Master's Thesis, Carnegie Mellon University*.
- METOYER, R., 2002. Building behaviors with examples. *Ph.D. Dissertation, Georgia Institute of Technology*.
- NATURALMOTION, 2005. Endorphin. www.naturalmotion.com.
- OSHITA, M., AND MAKINOCHI, A. 2001. A dynamic motion control technique for human-like articulated figures. *Computer Graphics Forum (Eurographics 2001)* 20, 3, 192–202.
- POLLARD, N. S., AND BEHMARAM-MOSAVAT, F. 2000. Force-based motion editing for locomotion tasks. In *Proceedings of the IEEE International Conference on Robotics and Automation*.
- POLLARD, N. S. 1999. Simple machines for scaling human motion. In *Computer Animation and Simulation '99, Eurographics Workshop*, 3–11.
- POPOVIĆ, Z., AND WITKIN, A. 1999. Physically based motion transformation. In *Proceedings of ACM SIGGRAPH 1999*, 11–20.
- REITSMA, P. S. A., AND POLLARD, N. S. 2003. Perceptual metrics for character animation: Sensitivity to errors in ballistic motion. *ACM Transactions on Graphics* 22, 3 (July), 537–542.
- ROSE, C., GUENTER, B., BODENHEIMER, B., AND COHEN, M. F. 1996. Efficient generation of motion transitions using spacetime constraints. In *Proceedings of ACM SIGGRAPH 1996*, 147–154.
- SCHÖDL, A., SZELISKI, R., SALESIN, D. H., AND ESSA, I. 2000. Video textures. In *Proceedings of ACM SIGGRAPH 2000*, 489–498.
- SHAPIRO, A., PIGHIN, F., AND FALOUTSOS, P. 2003. Hybrid control for interactive character animation. In *Pacific Graphics 2003*, 455–461.
- SMITH, R., 2005. Open dynamics engine. www.ode.org.
- WANG, J., AND BODENHEIMER, B. 2004. Computing the duration of motion transitions: an empirical approach. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 335–344.
- ZORDAN, V. B., AND HODGINS, J. K. 2002. Motion capture-driven simulations that hit and react. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 89–96.
- ZORDAN, V. B., AND HORST, N. V. D. 2003. Mapping optical motion capture data to skeletal motion using a physical model. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 140–145.