Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2010)
M. Otaduy and Z. Popovic (Editors)

Performance capture with physical interaction

Nam Nguyen†[1] Nkenge Wheatland[1] David Brown[1] Brian Parise[2] C. Karen Liu[2] Victor Zordan[1]

[1]University of California Riverside [2]Georgia Institute of Technology

---

Abstract

This paper introduces a technique for combining performance-based animation with a physical model in order to synthesize complex interactions in an animated scene. The approach is to previsualize interaction of final integrated scene, online, while the performance is being recorded. To accomplish this goal, we propose a framework which unifies kinematic playback of motion capture and dynamic motion synthesis. The proposed method augments a real-time recording of a human actor with dynamics-based response in order to modify motion data based on the conditions of the character. The system unifies kinematic and dynamic aspects of the final motion while allowing user control over the outcome both temporally and spatially across the character's body. Examples of complex interactions interleaved with intelligent response underscore the power of the technique along with multi-person captures in which remote users interact physically in a shared virtual world.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.7]: Three-Dimensional Graphics and Realism – Animation—

---

1. Introduction

Creating scenes with complex interaction between actors and the environment is crucial to many applications in movies and games. Although computer-generated (CG) effects enable actors to appear visually realistic in virtual worlds, producing realistic interaction between real actors and virtual objects, features, or characters remains challenging. Because actors often perform outside the context of the virtual scene, seamless integration relies heavily on manual post-processing efforts to synchronize an actor's performance with CG effects.

We introduce a technique that previsualizes final, integrated scenes at the time of performance acquisition. Our technique seeks to enforce realistic dynamic interaction in the virtual world while faithfully preserving the nuances of the actor's performance. The system combines motion capture in the form of a real-time performance with physical simulation to generate visible response to the virtual interaction. We present a hybrid approach for combining pure (kinematic) motion capture and a dynamic simulation of the character in order to create the appearance of a more immersive performance, in real time. The seamless integration of these two animation signals (kinematic and dynamic) is accomplished by transforming the kinematic signal into a dynamics representation and then balancing the influence of

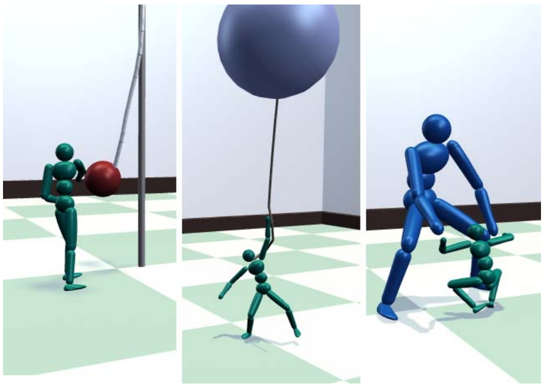

Figure 1: *Interactions from various scenarios created easily by combining motion performance and a physics-based representation of the character(s).*

---

† nnguyen@cs.ucr.edu

DOI: 10.2312/SCA/SCA10/189-195

the original data and the physical response across the body and across time.

To date, the concept of *online* performance animation has been narrowly perceived as a real-time motion reconstruction technique that is used in commercial applications such as digital puppetry [SLGS01]. However, this limited definition largely ignores an important feature of the performance-based approach, that is, the reactivity provided by human intelligence in real-time. In our use of performance animation, the actor/performer is not only providing motion trajectories, but also revealing how humans make decisions and strategic actions in response to situations. Several examples appear in Figure 1.

Fundamentally, integrating real-time performance with a physically simulated character introduces two major challenges. First, due to inconsistencies between the virtual and real world, the performance can become unrealistic (or invalid) when directly applied to the virtual character. This issue has been discussed by Ishigaki et al [2009] using an example where the performer is standing on the floor attempting to control a suspended virtual character traversing a monkeybar set. However, if there is physical interaction between the character and a virtual object, the character's motion should respond to the impact from the interaction and accommodate an appropriate deviation from the performance. In this case, the character movement should remain physically plausible for the duration of the interaction, obey the performance to the extent possible, and return to the performer's motion in a timely and believable fashion. This paper largely tackles this second challenge: integrating performance with physics for synthesizing virtual interaction.

2. Background

Offline human performance has already proven extremely valuable for generating highly humanlike motion in the form of recorded motion capture data. The growing use of motion capture animation has resulted in many research efforts on adapting, generalizing, and combining motion data (too many to mention here). However, our novel approach deviates from the normal use of motion capture specifically because it is online and therefore editing apriori is not possible.

Using online performance to control a virtual avatar has been proposed by many researchers with input devices that range from video cameras [Kru88, CH05], accelerometers [SH08], foot pressure sensors [YP03], and full-body motion capture devices [SLGS01, IWZL09]. The underlying algorithms solve the problem of mapping the performance to the character's action. Ishigaki et al. [IWZL09] introduce a control interface that integrates an online performance with offline example motions to produce realistic motion while preserving the style of the performer/user. Like our technique, they also use a dynamic model but only simulate the root of the character. Due to this simplification, the physical interaction with the environment does not result in detailed, context-appropriate responses. In contrast, our method employs a full-body physics model that simulates realistic responsive behaviors, yielding a lifelike character participating in a cognizant fashion within a rich dynamic environment.

Closely related to this paper is the growing body of work in responsive characters which employs both physical and procedural methods for synthesizing interaction. During an interaction in procedural approaches that do not include physics models, a character moves according to defined methods which account for the environment [AFO05, YPvdP05, YL10]. Physically based response techniques afford character interactions through impulses or collision detection and force-based response. Several researchers have proposed systems which transform an (offline) motion example to a physical representation and compute reactions to an impact by deviating and returning to the sample motion [ZH02, YCP03, ACSF07, AdP07, YCBvdP08, SAP08, YL08, MLPP09]. However, these techniques cannot create a large deviation from the original performance and add a very specific level of *physical* responsivity to the character. A handful of hybrid techniques transition to a new motion examples and/or control to generate larger responses [FvdPT01, SPF03, Man04, ZMCF05, NVCNZ08]. While some of these techniques share similar frameworks, such as Nunes et al., none seamlessly combine physics and kinematic playback spatially as we do in this system.

Further, our technique is novel in that it can inject a strategic, context-appropriate response via online performance. Our approach is to immersively include the actor as an intelligent decision-maker for the character. This distinction creates the appearance that the character can assess and make timely choices in addition to react through physical response. In this fashion, our technique is similar to previous work on interfaces for animation such as [LvdPF00, DYP03] in that timing and human intelligence can be used to guide the appropriate action based on context inferred through visual representation of the virtual scene.

3. System overview

The description of our system architecture follows the layout diagram appearing in Figure 2. The raw performance starts with the actor and is input into a combined kinematic-dynamic (KD) controller. The forward simulation of the virtual world along with the physical model of the character is advanced in time based on the output of the KD control inputs. In this step, collisions are resolved by modifying the character motion and the objects in the scene. The actor gets visual feedback from the projection output and modifies the online performance appropriately.

At the core of our technique is the KD controller which seamlessly combines the (kinematic) motion performance with a physics model. The domain for our KD controller is defined as the continuous space that combines two distinct

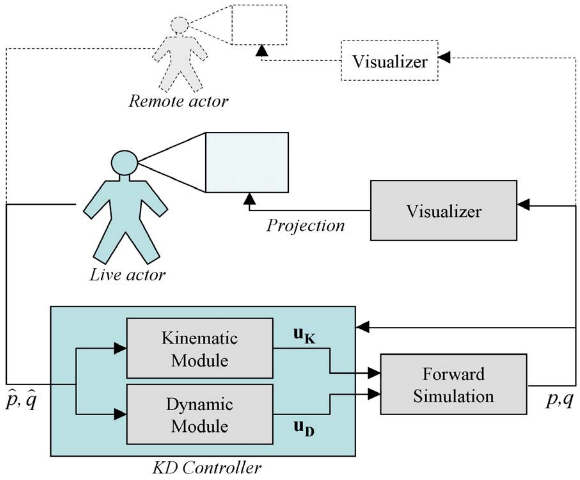

Figure 2: *The layout of the system shows the basic flow of information from the actor(s) through the motion capture system to the KD controller and the simulation. The simulation state is fed back to the control and input to the visualizer(s) which project(s) the scene for the actor(s).*

interpretations of the motion performance, first as a pure kinematic playback and second as a physical model following the performance as a desired setpoint trajectory. In our system, we implemented the latter as a forward simulation following the performance via a tracking control routine. We sidestep coordinated control in lieu of a naive tracking controller and incorporate intelligence and coordination through online performance capture.

Our system creates a single simulation within which a host of interactions are possible. For example, by adding additional simulations we can created coupled simulations like those seen in previous work [OZH00, LHP06, SSF08] Also shown, with dashed lines, in Figure 2 is a networked remote actor. We extend our basic system to allow a remote performance capture and interaction between two actors in long-distant real-world locations. The networked clients remotely report the motion performance of the actors while a single local server manages the simulation that resolves interactions and generates a consistent state. This state is fed back to each remote client which uses its own visualizer that can uniquely control camera view, etc. for each actor.

4. KD control

In this paper, we introduce two distinct notions of the performance, one as playback of the raw data through a kinematic model of the character, and the other as the result of a full-body simulation of the character driven to follow the raw performance data. For clarity, let us define vector $\hat{\mathbf{q}} \in R^{3n}$ where n is the number of 3D joints in the skeleton and assign it the given sample of the raw orientation data from the motion capture performance. $\mathbf{q}$ similarly is the output state of the full-body simulation. Further, if we apply forward kinematics for the character skeleton to the data sample, we obtain the positions of each body part from that data. We call this position vector $\hat{\mathbf{p}} \in R^{3m}$ where m is the number of body parts in the skeleton. And, from the simulation output we yield the simulated body positions $\mathbf{p}$.

For our goals in this research, we would like a unified controller that can encompass both representations described for the performance. However, we must resolve the discrepancy between the kinematic $(\hat{\mathbf{q}}, \hat{\mathbf{p}})$ and dynamic $(\mathbf{q}, \mathbf{p})$ representations. To unify the two KD representations, we propose to implement the kinematics as a special case of the dynamics. In this manner, inputs computed for each can be applied to the same unified model. We label those inputs $\mathbf{u_K}$ and $\mathbf{u_D}$. In our technique, $\mathbf{u_K} \in R^{3m}$ is a vector of Cartesian forces derived from our kinematic controller while $\mathbf{u_D} \in R^{3n}$ is a vector of joint torques derived from our dynamic controller. In the remainder of this section, we define the mathematical operations that transform the two input signals into the set of control inputs applied to the character simulation.

4.1. Physically based kinematics

Conceptually, one could interpret motion capture playback as a set of accelerations for each body across each timestep. By assigning mass and inertia to each body, there is an equivalency between these accelerations and a set of forces and torques which will lead to the analogous accelerations of these bodies. This hypothetical transformation is equivalent to inverse dynamics. Effects due to gravity and external forces can also be accounted for within this framework.

Rather than performing inverse dynamics explicitly, we propose a simple solution to create a physical representation of the motion performance which derives a set of forces, $\mathbf{u_K}$, which will lead a physical simulation of the character to follow a performance spatially. Specifically, to compute $\mathbf{u_K}$, we use a Cartesian-based PD-servo as

$$\mathbf{u_K} = k_p(\hat{p} - p) - d_p(\dot{p}) \tag{1}$$

where $\hat{p}$ is the body position derived from the performance fit to the kinematic skeleton and gains k_p and d_p are manually tuned constants. In tuning, these values are chosen such that they are large enough to be the dominant influences (overcoming gravity and contact forces) while not leading to instability.

4.2. Dynamic control

In contrast, our dynamic controller follows the actor's performance by tracking the joint angles of the motion capture data using internal joint torques, $\mathbf{u_D}$. Without the presence of disturbances, these torques drive the character to match the joint trajectories of the performance. However, the joint tracker does not control the root body and lack of coordination would lead the character to fall over using $\mathbf{u_D}$ alone. Rather than focusing on coordination using joint torques, we apply $\mathbf{u_D}$ in conjunction with $\mathbf{u_K}$ and transform the character from a purely dynamic one to a hybrid which combines both coordination and responsivity. Specifically, the joint tracker computes joint torques based on the tracking error $(\hat{q} - q)$ and joint velocities $(\dot{q})$ as follows

$$\mathbf{u_D} = k_q I(\hat{q} - q) - d_q(\dot{q}) \qquad (2)$$

which follows the raw performance $\hat{q}$ based on the value k_q and damping d_q. I is the inertia of the joint's outboard bodies as in [ZH02, ACSF07].

Our choice of tracker is based on its simplicity. This choice of tracking control is not critical but the system must be responsive to the physical constraints of the space while also reflecting the pose $\hat{\mathbf{q}}$. Any tracking control system which suffices to meet this objective will do and alternatives such as nonlinear optimization [YL08] and multi-objective frameworks [AdP07, SAP08] also offer possible solutions. We opt for the one described because the incorporation of (non-recorded) performance data is straightforward and also the controller affords a method for unifying the kinematic and dynamic representations as described below.

4.3. Integrated KD control

The described controllers both guide the character to follow the performance. However, each has its own characteristics. The forces computed in Equation 1 maintain global Cartesian positions but resist external forces in an unrealistic manner, in the limit appearing to ignore them completely while following the motion kinematically. In contrast, the joint torques in Equation 2 make corrections in local joint coordinates and, therefore, will respond in a more believable manner. The Cartesian forces provide more precise control from the perspective of the performance and can aid in balancing the full-body character, while the joint torques only provide a limited amount of control and no balance. We wish to blend these two complementary inputs in order to get the best of both.

In regards to balance, we choose to join the KD signals within the body based on the proximity of each body part to the ground plane. Empirically, a stronger influence of kinematics in the lower body can aid in balance while a lower value in the upper body can afford more directed, physically plausible interactions. This assumption is also justifiable drawing from the observation that the ground is the source of the support forces which lead to the actions that are embedded in the performance and, subsequently, in the forces from Equation 1. In our implementation, we add this influence by scaling the lower body in the following manner: feet feel a multiple of 125x; lower legs feel a multiple of 25x; and the upper legs 5x. These multipliers scale the gain value applied to the rest of the body from Equation 1.

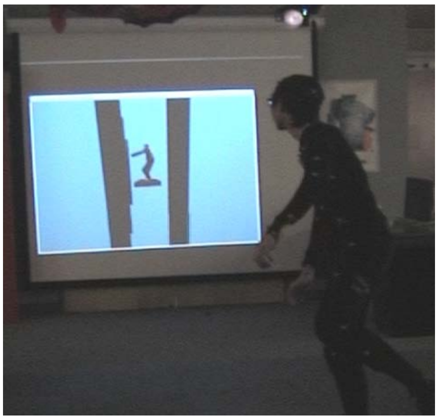

Figure 3: *Live capture of system in use - projection screen displays animation being generated to allow real-time feedback from the performer. In this scene, the actor is controlling the glider interface prototype.*

5. Implementation

We employ the proposed framework to synthesize motion in a capture studio-type setting. In this context, we view the actor as an accomplice who will aid in the process of creating the desired animation, as opposed to a game application where the player would be a naive (adversarial) user of the interface. Our implementation combines a real-time solver for motion capture (from Vicon Motion Systems, www.vicon.com) with a real-time simulation engine employing ODE (www.ode.org) for the dynamics. An analogous skeleton is imported into ODE to align the character representation with that of the actor. The character skeleton has degrees of freedom with ball and sockets at each joint. We employ ODE's collision handler to manage all collisions. In the results shown, the actor views the world from a large projection screen adjacent to the capture area (See Figure 3).

In practice, we found it necessary to include an additional user to give commands to the system, for example to rotate the camera for the visualizer. We call this person the operator. The purpose of the operator is to control the other aspects of the system not available to the performer. For example, the operator commands the initiation and termination of grasping, the camera control, and the motion of the virtual objects (such as controlling the size/bouyancy of the balloon and steering the car). Further, the operator can enable/disable the kinematic forces (for freefall) and helps in efficiently tuning the system.

Several extensions are added to the basic implementation. The system computes the support polygon and the center of mass for the body and visually informs the actor about their status through a visualization in the projection. By paying attention to these balance features, the actor can adjust her/his posture accordingly in order to keep the character standing in a more realistic fashion. We add the ability to grab objects in two ways, first as a soft grasp via an additional PD-spring force which “holds” the hand and an object together, and also as a hard grasp by creating a joint constraint (through ODE) between the hand and the object to be grasped. Currently, we do not have a functioning hand (and so the character can not carefully manipulate objects) in part, because the real-time motion capture is too poor to reliably determine what the actor is doing with a hand. We also found it was critical to perform a careful treatment of the real-time motion capture data. Our custom plugin for the Vicon software time stamps all out-going packets and the simulation software ensures the data is fresh, checks and throws out poor samples, and filters the data as well.

6. Results

We crafted a series of different scenarios to show off specific features of the system as well as its overall flexibility. The results appear in the filmstrips in Figure 4 as well as in the paper video. Respective gains appear in the table below.

Tether Ball. This example highlights the actor being able to manipulate objects in the environment. The character can hit the ball object and affect its motion. It also highlights the use of real time motion capture. The actor swings at the virtual ball in real time as if the ball is a physical object in front of him. Without fully immersing the actor in the interative virtual world, timing-critical scenarios like this example would be difficult to create.

Balloon. To show that an object can also affect the motion of the character, we present this example of the character holding onto a balloon. Initially, the character influences the motion of the balloon. As the balloon increases in buoyancy, it begins to counteract the charcter’s weight. It becomes difficult for the character to remain on the ground. When the buoyancy reaches a certain level, the balloon floats away with the character. As detailed in the gains table, the character in this scene has no forces and has decreased torques. This allows the character to be more affected by the buoynacy of the balloon.

Stapler. This example shows the actor’s character interacting with an oversized stapler that has a naive ‘wind-up’ toy control [vdPKF94]. The actor is able to dampen the vertical motion of the stapler. The character also reacts to the stapler’s motion which in turn affects how the character is moving. The actor’s character struggles to suppress the movement of the stapler. Note how the changes in the stapler’s motion lead to the actor readjusting his hold. The character in this scene has increased spring forces to better control the virtual stapler.

Tug of war. This example shows the system in effect with a network connection. Two actors are recorded in remote studios (crosscontinental). With the system, the two actors are able to interact and influence each other in a shared virtual space in real time. In our version of "tug of war" both actors struggle over a virtual object and pull on the object in a way that affects the motion of the other actor’s character. We see how the effect of a pull on the object causes an immediate physical response in the other character.

Giant. One common problem with capturing multiple character interaction is the occlusion of mocap markers when the characters cluttered in a small space. By allowing two actors performing in two physical locations and interacting virtually, we can create scenarios with contact-intensive interaction among multiple characters. A scene of a giant and a regular sized character struggling is produced. The scene was created with two actors working in parallel as opposed to the scene being created sequentially with each actor recording his parts separately or an animator creating the appearance of a giant in post production. During the capture, the giant character picks up and holds onto the smaller character. This animation reveals how the system can manipulate various aspects of the characters’ performance. The giant character in the scene weighs considerably more than the regular-sized character. The giant is given increased strength to hold and manipulate the other character.

Glider. The glider example demonstrates the actor dextrously manipulating a virtual object that has unique behavior. The glider has an upward local force applied to it similar to that of a helicopter. This force keeps the glider and the actor in the air. When the actor changes the level of his feet, the glider pitches correspondingly. The glider’s force pushes the character in the appropriate direction. The actor can also shift his facing direction to rotate the glider so that the glider moves in different directions. The actor uses these interface controls to navigate a slalom course. Increased kinematic forces allows the character to more easily stay on the glider.

The table below includes all KD gain and damping values used. The default is used for several examples while deviations are highlighted. To derive these values required some tuning, but the process was fairly straightforward. That is, we start with a fair initial guess, based on the values of previous scenes, and iterate a few times while adjusting the values until we find the results to our liking.

	Kinematics		Dynamics	
	k_p	d_p	k_q	d_q
*Default**	160	25	450	10
Balloon (in the air)	0	0	200	2
Stapler	200	25	450	10
Giant	420	50	1200	15
Glider	1000	125	450	10

*Scenes not mentioned here use default values

In terms of usability, we have not yet performed careful subject testing. However, we point out that currently there is about a one eighth-of-a-second delay between the performer and the on-screen character. Also, the character’s pose often diverges from the performer’s as a result of effects from the virtual environment. Even so, across the several actors that used the system in our trials, these discrepancies required only a short acclimation period. Our main performer gained a proficiency with the system in a few hours and was able to compensated for the lag well enough to perform careful manipulations (e.g. wrangling the stapler.) Further, he was able discern what features of the motion were important, where to add "flourish", and which aspects of the motion to ignore, for example, because they were under complete dynamic control. While we would not claim the system is ready for novice users, evidence supports that the system is usable by individuals willing to go through some basic training.

7. Limitations

While the power of our technique is revealed by the unique interactive *pre-visualization* animations in the accompanying video. There are several limitations to the current system. We highlight a few of the key issues here and offer some suggestions for future extensions to the described system. But first we point out that we found that nothing is more paramount to high-quality results than a strong acting performance - which indicates that our system is both conveying the performance and dependent on good, real-time motion capture data. Further, in juxtaposition, with access to better actors and facilities, our technique is likely to shine more brightly than appears in the accompanying results.

In the described implementation, the performer's "window" into the animated world is a projection within the capture studio. However, this set-up creates a need for the actor to constantly look in a particular direction which can lead to problematic results that would require additional clean-up to animate the head properly for a production quality animation. As an alternative, we also added a head-mounted display (HMD) to the implementation with some success. The HMD allows the actor to see the virtual environment in first person, which eases the discrepancies between the actor and the character with respect to the viewing direction and avoids having the actor look at the screen during performance. However, the actor found the screen easier because it gave more contextual information about the scene and a third person view of the character.

We currently require that the performer act as an accomplice to the system because, as is, the system can lead to undesirable situations. For example, we are currently unable to handle a case where the primary support of the character changes from the ground to another object. Thus, the character can put a hand on top of a wall but cannot lift himself onto or over the wall automatically because our system assumes the ground is the primary support and distributes the kinematic corrective forces based on this assumption. Likewise, the actor can easily "break" the system, for example, by using the strength in the feet and legs to an unfair advantage in interaction. We particularly require that the actor be cooperative in cases where the character falls over or loses balance. Obviously, these limitations would need to be addressed if the current system was to be implemented for a game setting. We note that Ishigaki et al. [2009] do offer solutions to some of these limitations.

Of course there are also several directions for future work. Foremost, we are eager to make more prudent use of the kinematic control forces - as is, they remain too powerful and at times dissipate the effects of interactions. Also, there is a bias in our system to results with standing interactions and further investigation of the described approach for more interesting balance scenarios is pending. We are also eager to further our investigation in the direction of multi-person capture as well. Overall, we feel this paper makes a valuable contribution by exposing the unique possibilities of combining performance and physics, but there is a certainly great deal more to explore and discover in this exciting domain.

8. Conclusions

We describe a technique that incorporates a physical simulation with a motion capture performance in order to allow an actor to interact with virtual objects. In this setting, acting as an observer and intelligent agent for the character, the actor guides the movement through her/his performance in real-time. We consider the result akin to raw motion capture data and envision our technique to be used for rough previsualization, a precursor to high quality post-production.

With this paper, we introduce a novel continuum between kinematics and dynamics that we exploit both spatially across the character's body parts and temporally as the character makes and breaks contact with objects in the environment. We show the power of the proposed technique through a spectrum of test scenarios which reveal the capabilities of the system for use in manipulation and intelligent, physically based responses to complex interactions. The resulting animations showcase character animation with motion that is dictated by an actor's performance but still responds to the contact of objects in the environment, including other actors.

References

[ACSF07] ALLEN B., CHU D., SHAPIRO A., FALOUTSOS P.: On the beat!: timing and tension for dynamic characters. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2007).

[AdP07] ABE Y., DA SILVA M., POPOVIĆ J.: Multiobjective control with frictional contacts. In *Eurographics/SIGGRAPH Symposium on Computer Animation* (2007).

[AFO05] ARIKAN O., FORSYTH D. A., O'BRIEN J. F.: Pushing people around. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation* (New York, NY, USA, 2005), ACM Press, pp. 59–66.

[CH05] CHAI J., HODGINS J. K.: Performance animation from low-dimensional control signals. *ACM Transactions on Graphics (SIGGRAPH 2005) 24*, 3 (Aug. 2005).

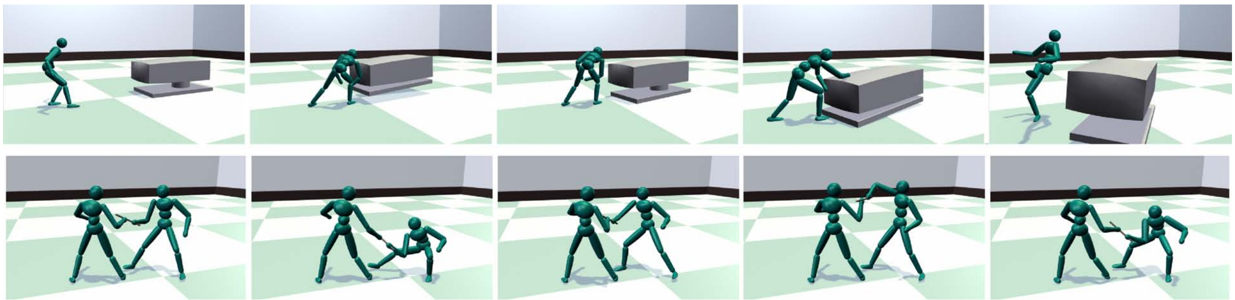

Figure 4: *Stills from the stapler and tug of war animation examples.*

[DYP03] DONTCHEVA M., YNGVE G., POPOVIĆ Z.: Layered acting for character animation. *ACM Transactions on Graphics 22*, 3 (July 2003).

[FvdPT01] FALOUTSOS P., VAN DE PANNE M., TERZOPOULOS D.: Composable controllers for physics-based character animation. In *SIGGRAPH* (Aug. 2001), pp. 251–260.

[IWZL09] ISHIGAKI S., WHITE T., ZORDAN V., LIU K. C.: Performance-based control interface for character animation. *ACM Transactions on Graphics (SIGGRAPH) 28*, 3 (2009).

[Kru88] KRUEGER M.: Hanging by a thread. *SIGGRAPH 1988 - Electronic Art* (1988).

[LHP06] LIU C. K., HERTZMANN A., POPOVIĆ Z.: Composition of complex optimal multi-character motions. In *2006 ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (Sept. 2006).

[LvdPF00] LASZLO J., VAN DE PANNE M., FIUME E. L.: Interactive control for physically-based animation. In *Proceedings of ACM SIGGRAPH 2000* (July 2000), Computer Graphics Proceedings, Annual Conference Series.

[Man04] MANDEL M.: Versatile and interactive virtual humans: Hybrid use of data-driven and dynamics-based motion synthesis, 2004. *Master's Thesis, Carnegie Mellon University*.

[MLPP09] MUICO U., LEE Y., POPOVIĆ J., POPOVIĆ Z.: Contact-aware nonlinear control of dynamic characters. *ACM Transactions on Graphics 28*, 3 (July 2009).

[NVCNZ08] NUNES R. F., VIDAL C. A., CAVALCANTE-NETO J. B., ZORDAN V.: Simple feedforward control for responsive motion capture-driven simulations. *International Symposium on Visual Computing (ISVC)* (2008).

[OZH00] O'BRIEN J. F., ZORDAN V. B., HODGINS J. K.: Combining active and passive simulations for secondary motion. *IEEE Computer Graphics & Applications 20*, 4 (2000).

[SAP08] SILVA M. D., ABE Y., POPOVIC J.: Interactive simulation of stylized human locomotion. *ACM Transactions on Graphics 27*, 3 (July 2008).

[SH08] SHIRATORI T., HODGINS J. K.: Accelerometer-based user interfaces for the control of a physically simulated character. *ACM Transactions on Graphics 27*, 5 (Dec. 2008).

[SLGS01] SHIN H. J., LEE J., GLEICHER M., SHIN S. Y.: Computer puppetry: An importance-based approach. *ACM Trans. on Graphics 20*, 2 (Apr. 2001), 67–94.

[SPF03] SHAPIRO A., PIGHIN F., FALOUTSOS P.: Hybrid control for interactive character animation. In *Pacific Graphics 2003* (2003).

[SSF08] SHINAR T., SCHROEDER C., FEDKIW R.: Two-way coupling of rigid and deformable bodies. In *2008 ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (July 2008).

[vdPKF94] VAN DE PANNE M., KIM R., FIUME E.: Virtual wind-up toys for animation. In *Graphics Interface* (May 1994), pp. 208–215.

[YCBvdP08] YIN K., COROS S., BEAUDOIN P., VAN DE PANNE M.: Continuation methods for adapting simulated skills. *ACM Transactions on Graphics 27*, 3 (July 2008).

[YCP03] YIN K., CLINE M., PAI D.: Motion perturbation based on simple neuromotor control models. In *Pacific Graphics* (2003), pp. 445–449.

[YL08] YE Y., LIU C. K.: Responsive characters with dynamic constraints in near-unactuated coordinates. *ACM Transactions on Graphics 27*, 5 (Dec. 2008).

[YL10] YE Y., LIU C. K.: Motion transition using a nonlinear statistical dynamic model. In *Eurographics* (Norrköping, Sweden, May 2010).

[YP03] YIN K., PAI D. K.: Footsee: an interactive animation system. In *2003 ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (Aug. 2003).

[YPvdP05] YIN K., PAI D., VAN DE PANNE M.: Data-driven interactive balancing behaviors. *Pacific Graphics* (2005).

[ZH02] ZORDAN V., HODGINS J.: Motion capture-driven simulations that hit and react. In *ACM SIGGRAPH Symposium on Computer Animation (SCA-02)* (July 21–22 2002), ACM Press, pp. 89–96.

[ZMCF05] ZORDAN V., MAJKOWSKA A., CHIU B., FAST M.: Dynamic response for motion capture animation. *ACM Trans. on Graphics (SIGGRAPH) 24*, 3 (July 2005), 697–701.