Combining Active and Passive Simulations for Secondary Motion

James F. O'Brien, Victor B. Zordan, and Jessica K. Hodgins
Georgia Institute of Technology

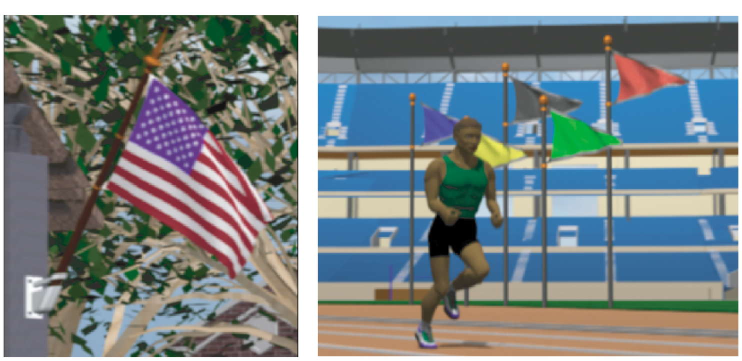
We describe how to generate secondary motion by coupling physically based simulations of passive objects to actively controlled characters.

Objects that move in response to the actions of a main character often make an important contribution to the visual richness of an animated scene. We use the term *secondary motion* to refer to passive motions generated in response to the movements of characters and other objects or environmental forces. Secondary motion may be created by background elements or by objects interacting with an active character. The flags shown in Figure 1 are examples of secondary motion generated by environmental forces, while the trampoline and skirt in Figure 2 are objects that exhibit secondary motion in response to the actions of active characters.

Secondary motions aren't normally the main focus of an animated scene, yet their absence can distract or disturb the viewer, destroying the illusion of reality created by the scene. For example, if the skirt in Figure 2 were rigid, the scene would be less believable; with painted-on, skin-tight clothing, the scene would be less interesting. While the viewer may not always be explicitly aware of secondary motions, they're an important part of many animated scenes.

Much of the research in computer animation has focused on the difficult problem of animating the primary characters. Because objects that exhibit secondary motions tend to be complex, deformable objects with many degrees of freedom, the techniques that have been developed for character animation are usually not appropriate for animating secondary motion. In particular, methods based on motion capture or key-framing are often impractical for secondary motion. As a result, researchers have developed specialized procedural methods for many of these objects.

While procedural models may be derived in a number of ways, physically based simulation has proven to be both a highly effective and an elegant solution, particularly for passive systems with many degrees of freedom. One advantage of simulation is that the motion is generated automatically from the initial specification of the environment and the applicable physical laws. For some applications, such as character animation, this automation results in an undesirable loss of direct control over the details of the motion. However, for secondary motion this lack of control is usually not a significant problem because these motions are passive, dictated only by forces from the environment or the actions of the primary characters. Even in situations where aesthetic considerations call for an exaggerated or otherwise unrealistic motion, often the movement of the actor is exaggerated while the passive secondary motions simply respond to the exaggerated motion.



1 **Simulated flags in the wind. Flags are examples of simple background elements that move in response to environmental effects.**

Simulation has been successfully used to model many isolated phenomena, but secondary motion by definition involves interactions between objects. Specialized simulations can be coupled together using inter-system constraints and forces to model the complex interactions that occur in the real world. The main contribution of this work is an exploration of the issues involved when passive secondary systems are coupled to another, primary, system. Typically, but not nec-

0272-1716/00/$10.00 © 2000 IEEE

2 **An animated scene with secondary motion. Both the swinger's skirt and the bed of the trampoline must move if the animation is to be convincing. Additional moving elements, such as the kites flying in the wind, further enhance the realism of the scene.**

essarily, the primary system will be active, having an internal source of energy and a control system to govern its behavior.

We classify methods for coupling two systems together as *two-way*, *one-way*, or *hybrid*. To clarify the differences between these three forms of coupling, we use the interaction between a basketball (primary) and net (secondary) as an illustrative example. If the simulations are two-way coupled, the rotational and linear velocity of the ball will be changed by its contact with the net and the net will be pushed out of the way by the ball. If the coupling is one-way, the net doesn't affect the motion of the ball, and the ball continues on a ballistic trajectory. The deformation of the net will be more extreme than in the two-way coupled case, and the motion won't match that of an actual basketball and net as closely. Between these two solutions lie a variety of hybrid solutions where the interaction model is approximate.

The physics of a particular situation and the fidelity of the required motion determine how the simulations should be coupled. In some situations, one-way or hybrid coupling can result in substantial computational savings with little loss of realism. In others, a tight two-way coupling is essential. To illustrate some of these issues, and to demonstrate the generality of our approach for generating secondary motion, we built several systems by coupling simulated components: a gymnast on a trampoline, a man on a bungee cord, a flying stunt kite, a gymnast landing on a flexible mat, a diver entering the water, and several human figures wearing clothing.

Background

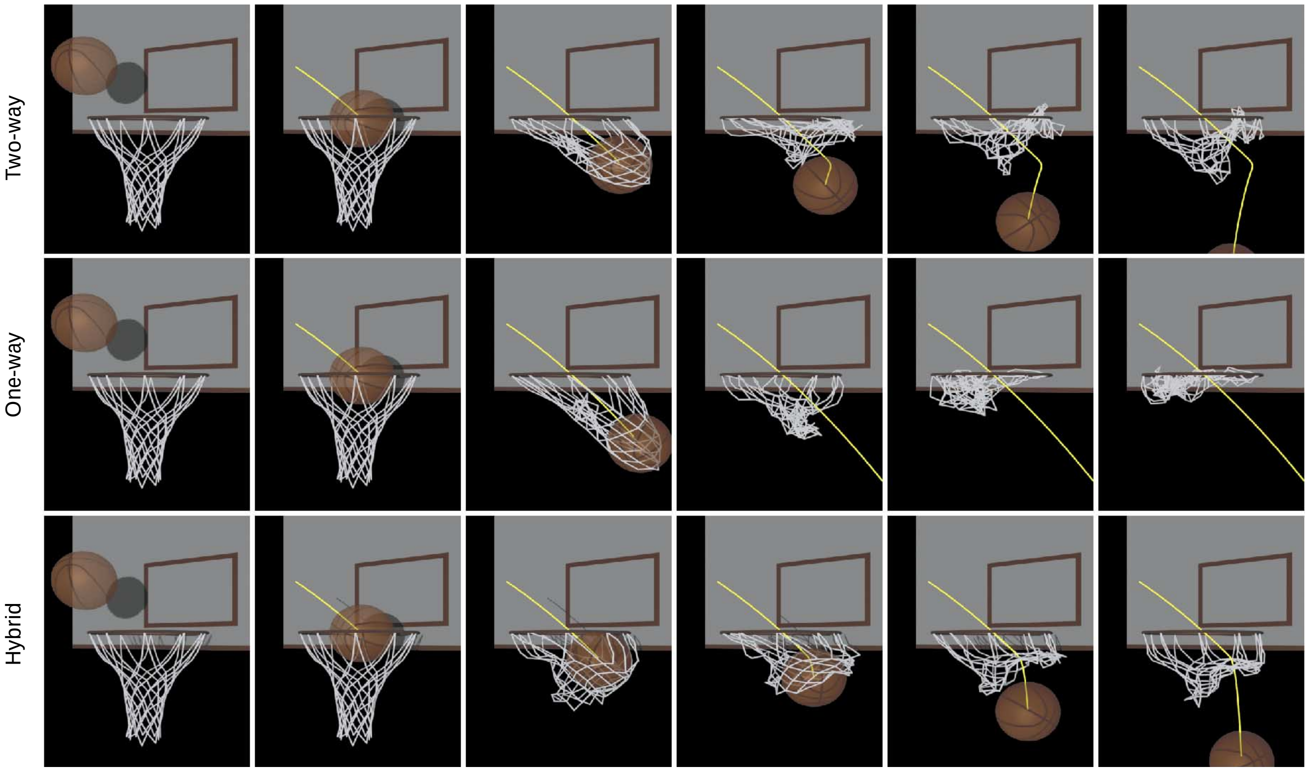
A number of techniques have been developed that use physically based simulation to generate motion for animation. Most research has focused on the issue of designing a simulation method for a particular type of phenomenon or motion. With the exception of work by Baraff and Witkin,[1] techniques for coupling simulations remain largely unexplored. This section discusses techniques for simulating passive and active systems as well as previous work related to combining systems.

Simulation has proven particularly successful in animating passive systems with many degrees of freedom such as cloth, water, hair, and other natural phenomena. Cloth simulation packages are even appearing in commercial packages, and clothing simulation was used successfully in the Oscar-winning short *Geri's Game*[2]. Many of the techniques developed to model cloth build on the spring and mass techniques originally introduced to the animation community by Terzopoulos and his colleagues in 1987.[3] Other cloth systems developed since then use finite element methods and include self-collision as well as interaction with synthetic actors (see, for example, the work of Volino, Courshesnes, and Magnenat-Thalmann[4]). Recent work has shown interactive cloth simulation to be feasible,[5] and simple cloth objects are appearing as effects in physically based electronic games.

Most of the water models presented in the literature focus on such specific phenomena as splashing, waterfalls, and spray. The techniques provide varying levels of realism and interaction with external objects. Highly realistic results have been achieved using a variation of 3D Navier-Stokes equations to animate liquids in complex environments.[6] As with cloth simulation, fluid simulation techniques have progressed to the point where both interactive simulations[7] and their use in commercial productions[8] are feasible.

Other natural phenomena modeled include wind and atmospheric effects, deformable terrain, and natural hair motion. Some of these systems, combined with external elements, generate secondary motion. Li and Moshell modeled soil slippage and manipulation.[9] Their system supported interaction through a controllable bulldozer and other earth-moving equipment. Sumner, O'Brien, and Hodgins introduced a system for animating deformable terrain to create imprints from simulated characters in sand, snow, and mud.[10] Physical models have also been used to model how objects fracture. O'Brien and Hodgins developed a finite-element technique for modeling deformable objects that can break, crack, or tear when they deform in response to external forces.[11]

The use of simulation for active systems isn't as widespread as for passive systems because robust control algorithms that produce natural-looking motions are difficult to design with existing techniques. A number of hand-



3 Simulated basketball and net with different couplings. The yellow line highlights the path of the ball generated using two-way, one-way, and hybrid coupling. Images are sampled at 0.0833-second intervals.

tuned simulations for rigid-body human and deformable nonhuman characters have been introduced.[12-14]

Some of the work on passive systems includes specific examples of coupling two systems together. For example, combining deformable clothing with the motions of synthetic actors[15] and manipulating soil with a bulldozer[9] resemble what we term one-way coupling. However, these papers didn't investigate the general concept of coupling and didn't consider responsive active simulations, as would be the case for two-way coupling.

The work of Baraff and Witkin[1] most closely relates to the work presented in this article. They presented a method for combining groups of passive systems including particle, clothing, and passive rigid-body models. Their work focused on a method that uses constraints to allow multiple systems to interact. They included examples of complex interaction such as two-way coupling between a stack of rigid objects and a cloth object or particle spray. In our work, we focus on higher level issues including when coupling two systems is appropriate, how approximations can be introduced to increase interactivity and efficiency without significantly degrading the results, and issues specific to coupling active systems to passive ones.

Coupling

We aim to combine simulations of individual objects or phenomena so that they can interact with each other to produce secondary motion. The techniques referenced in the previous section address modeling the behavior of particular objects or phenomena using specific simulation techniques, and we build on this existing work. Thus, we adopt a modular approach where two or more systems are coupled together and emphasize the design of the interfaces between these systems.

Forces applied between the systems provide a natural way for one simulation to interact with another. We group the interactions into three categories based on the method of approximating the inter-system forces: two-way coupled, one-way coupled, and hybrid.

In the remainder of this section, we illustrate the differences between these coupling techniques with an example of a basketball going through a net. In this simple example, the primary system is the basketball and the secondary system is the net. The collisions between the net and the ball are the interactions that we aim to model. We represented the ball as a spherical rigid body that can translate and rotate freely in space. The ball is initialized with a linear and angular velocity determined by the animator. Once in flight, it experiences gravitational acceleration. We modeled the net with a spring and mass network attached by springs to a fixed-hoop rim. The mass points experience forces due to gravity, the actions of the springs, and their interaction with the ball.

Two-way coupled

While any computer simulation involves some level of approximation, a two-way coupled simulation models the interaction as realistically as possible given the component systems. Two-way interactions affect both components, and the forces applied to one system are mirrored by equal and opposite forces applied to the other. The systems are simulated in lock step with each other, and the actions of each system directly affect the other.

We implemented the basketball and net as a two-way coupled system. To model the interaction forces, we imposed collision constraints to prevent the points of the net's mesh from penetrating the basketball's surface. When the system detects contact, a constraint force prevents further penetration, a stabilizing damping force absorbs a portion of the impact energy, and a restoring force corrects any penetration error. We implemented friction with a static Coulomb friction model. The resultant force is applied to the appropriate points of the net, and an equal and opposite force, with a corresponding moment, is applied to the ball.

The top sequence of Figure 3 shows the ball's path with two-way coupling. The ball enters the net at a shallow angle while spinning clockwise, causing the net to deflect from its rest configuration until the strings in the net are pulled tight and the sideways velocity of the ball is reduced. The ball drops out of the net with a substantially altered trajectory.

The main drawback to this type of coupling is the computation time required before the ball's path can be viewed. We chose the parameters for the net to represent nylon cord which, while light and flexible, resists stretching. As a result, the simulation of the net is numerically stiff and requires a small time step, on the order of 10^{-5} seconds. Additionally, the net contains hundreds of masses, each of which must be integrated for every time step. The ball, on the other hand, is a rigid body with isotropic inertial moments, and its ballistic flight may be simulated with arbitrarily large time steps. Therefore, computing a time step for the ball is computationally much less expensive than computing a time step for the mesh of the net. If simulated alone, the ball could be computed in real time, allowing the animator to interactively view and refine the motion by changing the initial conditions. When the two simulations are coupled together, the animator must wait several minutes before the motion can be viewed. We refer to the time between specifying parameters and viewing the results as the *debug cycle time*.

Finally, the total computation time required to calculate the result of the two-way coupled simulation may exceed the total time required to simulate each of the systems separately, even allowing for the additional work required to compute the interaction. For example, consider coupling two simulations where one system has a small computational cost per time step but requires small time steps, and the other system has a large cost for each time step but is stable at large time steps. Either system alone may be fast enough to be usable, but when the two systems are combined, the poor stability of the first is likely to dictate a small time step for both, thus greatly increasing the total computational cost.

One-way coupled

With a one-way coupled system, the interaction forces are applied only to the secondary system, leaving the primary system unaffected by the interaction. This approach relies on the assumption that the neglected forces would have a minimal effect on the primary system. This situation is likely to occur when the mass of one component system is several times the mass of the other, when one system is constrained in a way that would counteract the interaction forces, or when an active primary system would be able to trivially correct for any disturbances caused by the interaction with the secondary system.

We implemented the basketball and net as a one-way coupled system to illustrate this coupling technique. The system computes the interaction forces as before, but no forces are applied to the ball. The center row of Figure 3 shows the resulting motion. The net doesn't affect the ball's path, and the resulting trajectory is ballistic and therefore unrealistic. The net is forced to stretch a great deal, despite its stiff material parameters, eventually causing a violation of the collision constraints (fifth image of second row in Figure 3). Because the net is substantially deformed by the interaction, it requires a significantly smaller time step to prevent instability.

One benefit derived from this method of coupling is that the two systems may be simulated separately, potentially avoiding a long primary debug cycle. In general, this type of coupling is easier to implement than two-way coupling because only the secondary system is modified. It also allows coupling where it's not possible or desirable to modify the primary system, as in the case of a motion-capture-driven or hand-animated character.

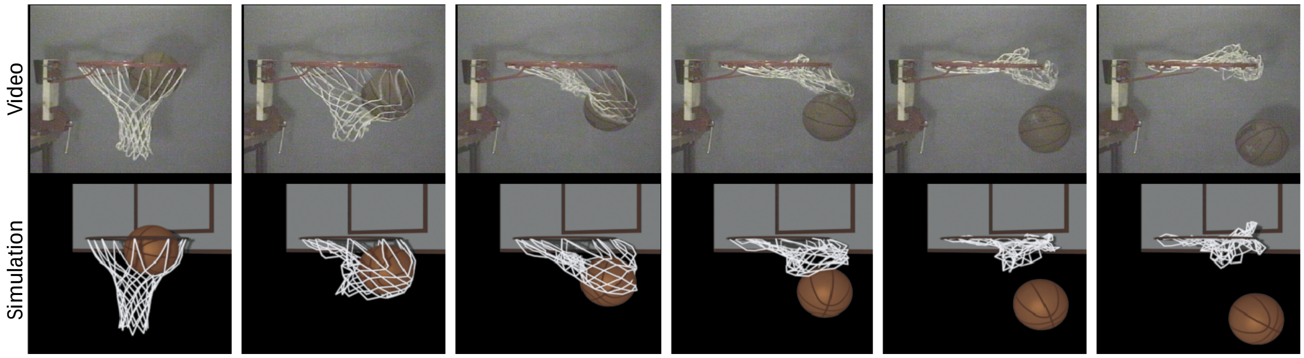
If the assumption that the interaction would have had a minimal effect on the primary system is wrong, as in the basketball and net example, the resulting motion will appear unrealistic. Even in cases where the effect would have been quite subtle, the resulting motion can appear incorrect in a way that most viewers might not notice consciously, but may nonetheless find disturbing and distracting.

Hybrid

A hybrid system is a compromise between the accuracy of two-way coupling and the speed of one-way coupling. As with one-way coupling, the primary system is computed independently of the secondary system. However, rather than ignoring the effect of the interaction on the primary system entirely, a simple approximation of the secondary system—a stand-in—interacts with the primary system. The primary system's motion then drives the secondary system as in the one-way coupled case.

The bottom row of Figure 3 shows the results of implementing the basketball and net with a hybrid coupling. The effect of the net on the ball is approximated with a damping field colocated with the net's rest configuration. As the ball passes through the field, its translational and rotational momenta are damped according to parameters the animator selected. Once the path of the ball has been determined, the net is simulated using the generated ball path.

The motion of the ball generated with the hybrid system is substantially different from the motions generated by the two- and one-way coupled simulations. The ball's horizontal and rotational velocity are slowed significantly, but, in contrast to the velocities seen with two-way coupling, they don't reverse direction because the simple approximation of a damping field isn't capable



4 **Comparison between simulation results for two-way coupling and video footage. The top row of images shows frames captured from video footage of a real basketball and net. The bottom row shows a two-way coupled simulation with matching initial conditions. Images are sampled at 0.067-second intervals.**

5 **The leaves are influenced by wind fields generated by moving objects such as a bicyclist. The diagram on the right shows the texture map and the spring and mass network used to model a leaf. For clarity, additional springs that resist bending and shear aren't shown.**

of producing that behavior. The ball's path and the net's motion, however, qualitatively resemble that seen with two-way coupling. The path generated by the hybrid simulation also differs significantly from the parabolic trajectory of the one-way coupled system, and the net isn't stretched in an unrealistic fashion.

This example uses a relatively simple stand-in to model the action of the net on the ball, but an arbitrarily realistic model could be used for the stand-in. The distinction is that while the simulation of the secondary system must model all the visible behaviors of the secondary object, the stand-in need only approximate the desired interactions. Designing a stand-in that can be simulated quickly and efficiently is much easier than designing a secondary system that can be fully coupled to the primary system. For any given secondary system, there exist many possible stand-ins with various levels of physical realism, and the appropriate stand-in depends on the level of realism the interaction requires.

The design and parameters of the approximation used for the hybrid simulation provide additional control handles for the animator. For the basketball example, the location, size, and damping constants of the field can be adjusted to achieve a desirable path for the ball. Because hybrid coupling should provide a shorter debug cycle time for the primary system than two-way coupling, the animator may interactively adjust these parameters until the desired trajectory is achieved.

Simulated versus real world

In the above discussion, we referred to the two-way coupled simulation as the most realistic of the three methods and implicitly used it as a standard against which to compare the results of the other two methods. The true standard, however, is the motion of a real basketball and net. Figure 4 compares images from video footage to rendered images of the two-way coupled simulation with similar initial conditions. The simulated ball and net move in a way that closely resembles the motion shown in the video images.

Example systems

In this section, we discuss a variety of examples and how they can be implemented as coupled systems. In the previous section, we used the basketball and net system as an illustrative example because it's a familiar system that's relatively simple to work with, and because we were able to implement it using each of the three coupling methods. For each of the examples in this section, we discuss a single implementation that employs the most suitable coupling technique. The examples are presented in approximately ascending order of complexity. We focus on passive systems modeled with mass and spring systems or with simplified fluid dynamics models. However, the ideas we describe should apply to other types of physically based systems.

Leaves

Figure 5 shows leaves blowing in the wind. The bicyclist generates a wind field that stirs up leaves in the road as he moves past them. We use Wejchert and Haumann's simplified aerodynamics model to drive the motion of flexible leaves blowing in the wind.[16] Because the actor doesn't experience any forces due to the motion of the leaves, the system is one-way coupled.

Clothing

We modeled clothing as a one-way coupled system. This choice is appropriate because the effect of the clothing on the simulated human is negligible. The clothing is modeled with a mass and spring system generated automatically from a geometric model. The system detects collisions between the clothing and the actor by intersecting the triangle faces of the actor's polygonal model with the triangles of the clothing model. Figure 6 shows a runner wearing a tee-shirt and sweat pants, and a child on a swing wearing a skirt.

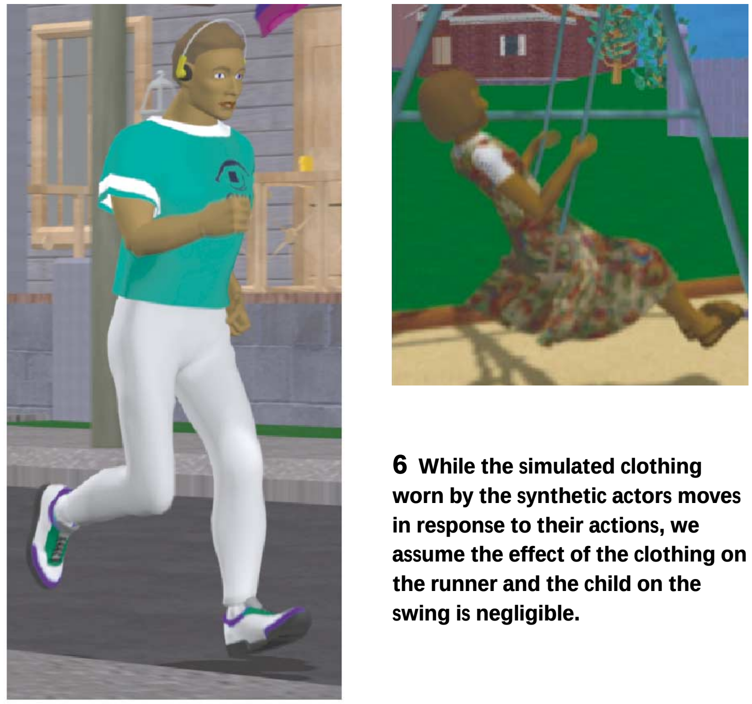
Floor mat

Figure 7 shows a gymnast landing on a deformable floor mat after performing a handspring vault. The floor mat makes the scene appear more realistic by softening the landing and by deforming to create a visual connection between the gymnast and the rest of the scene.

We modeled the floor mat with a mass and spring system, and the gymnast with a hierarchy of rigid bodies governed by an active control system. Because the gymnast's controller is tuned by hand, a quick debug cycle time is important. The gymnast simulation is relatively fast and can be run interactively, but the mat simulation is several times slower. Using a two-way coupling to link these systems would result in an unacceptably slow debug cycle time for the gymnast, but a one-way coupling would not have the desired result of softening the landing. Instead, we use a hybrid solution. The forces applied to the gymnast's feet are computed as if she were landing on a grid of vertical springs. Although this simple model will not capture subtle effects, such as sideways slip, the approximation has the desired result of softening the landing while still being very fast to simulate. Once the gymnast's motion has been computed, it drives the floor mat simulation and produces the desired deformation of the mat.

Water

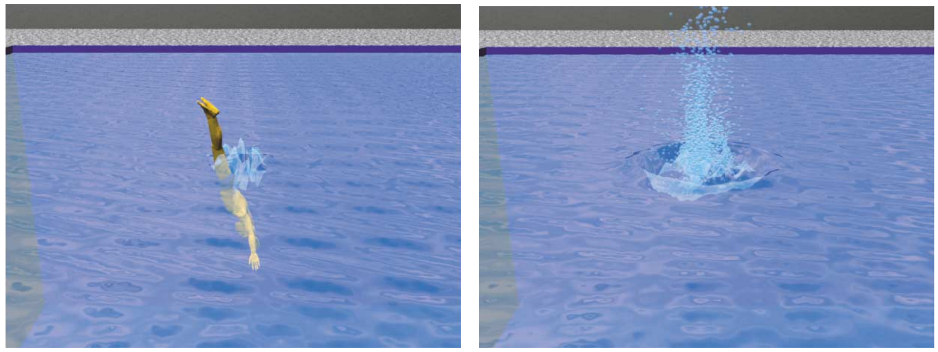
We've used one-way, two-way, and hybrid couplings to combine rigid body models with a height-field-based water simulation technique.[17] Figure 8 shows a runner stepping in a puddle. Because the water doesn't signif-



6 While the simulated clothing worn by the synthetic actors moves in response to their actions, we assume the effect of the clothing on the runner and the child on the swing is negligible.

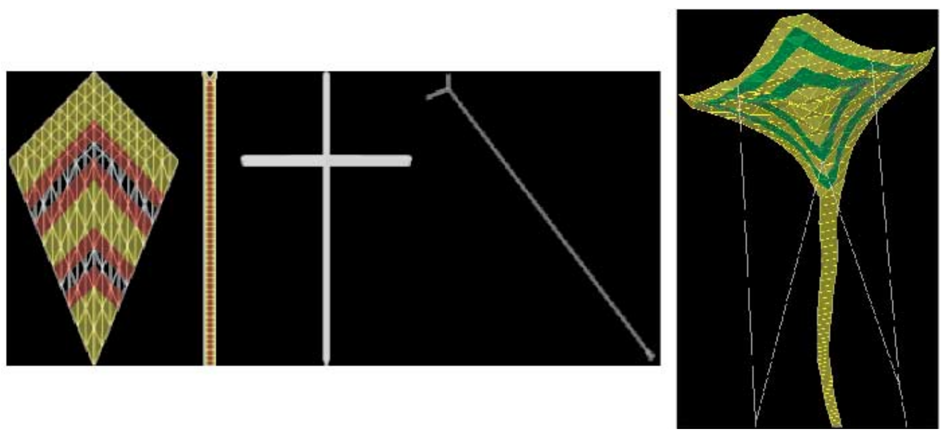
7 This closeup shows a gymnast landing on a deformable floor mat after a handspring vault. The compliance of the mat prevents the landing from having a painful, bone jarring appearance. The deformation also creates an important connection between the actor and the background.

8 Although the runner's motion is unaffected, the impact of the foot stepping in a puddle of water causes a splash.



9 **As the diver enters the water, he slows down due to viscous drag and creates a splash.**

10 **Two-way coupling is used to model the interaction between floating balls and water in a small pond. When the lighter balls are dropped into the water, they create small disturbances and float on the surface. The larger, denser ball creates a larger disturbance and sinks. The resulting waves affect motions of the floating balls.**



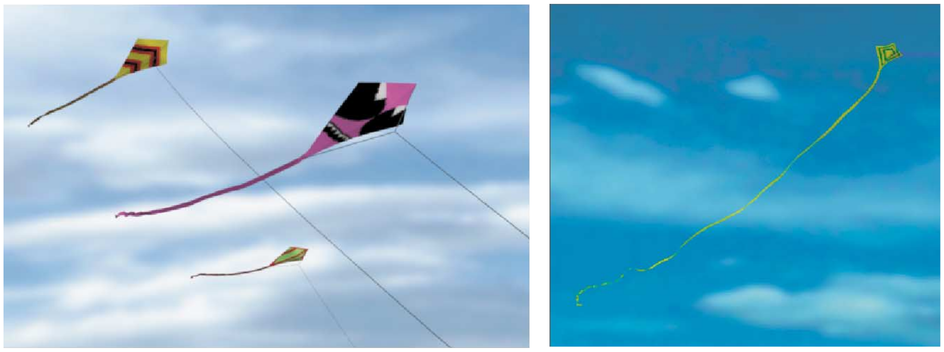
11 **Diagram of kite assembly. The four figures on the left show the components of the single-line kite: the wing, tail, frame, and line. On the right, the two-line stunt kite is shown assembled.**

icantly affect his motion, we used a one-way coupling to model the interaction. On the other hand, a diver entering the water from a 10-meter platform (Figure 9) should be significantly affected by the water, although the degree to which the viewer can observe this effect is limited. Therefore, we use a hybrid coupling where the diver encounters a viscous damping field that exerts drag forces on the body parts under water. The resulting motion then drives the water simulation. Objects floating on the surface of a pond, on the other hand, require two-way coupling because the motion of the floating objects affects the water's motion, and their motion is in turn affected by the water (Figure 10).

Kites and stunt kite

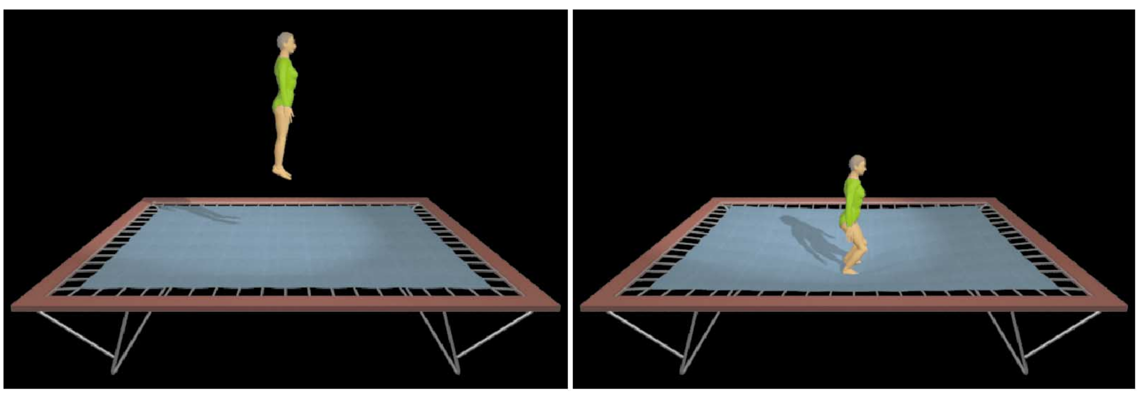
In addition to modeling the interactions between separate objects, two-way coupling can also be used to model the interactions of different components within a single object. By separating the object into components, we can make simplifications that are consistent with the specific qualities desired in the resulting motion of each component. We used this approach to model single- and double-line kites flying in the air. We divide each kite into four components: cloth wing, frame, bridle and string, and tail, as shown in Figures 11 and 12.

The kite is held aloft in the presence of gravity by the combined action of a horizontal wind field and the tension in the string. Lift and drag forces are generated on the wing and tail using the same simplified aerodynamic model as we used for the leaves. The wing ripples and deforms as the wind acts on it, causing variations in the net aerodynamic forces that propagate to the frame and creating subtle variations in the kite's motion. The drag on the tail serves to stabilize the system. The lower ends of the strings on the double-line stunt kite are moved by a control system that directs the path of the kite much as a person would fly a real stunt kite.



12 **Kites in the sky. The image on the left shows three single-line kites in the air. The image on the right shows the two-line stunt kite as it performs a looping maneuver.**

13 **Jumper on elastic bungee cord. The actor's control system causes him to leap from the bridge, and his fall is arrested by the action of the bungee cord attached to his ankles.**



14 **Gymnast on a deformable trampoline. This system must be two-way coupled because the interaction has a significant effect on the motions of both systems. The first image shows the gymnast in a layout position prior to landing; the second image shows her as she lands on the bed of the trampoline.**

Bungee jumper

The bungee jumper shown in Figure 13 is an example of a two-way coupled system where interactions play an important role in determining the motions of both the primary and secondary objects. We modeled the bungee jumper with a rigid body hierarchy and the bungee cord with a spring and mass system. Because the cord doesn't significantly affect the motion of the jumper until after he has left the platform, we debug the leaping control system with the cord simulation disabled. When we're satisfied with the motion for the leap, we use the two-way coupled system to compute the final motion.

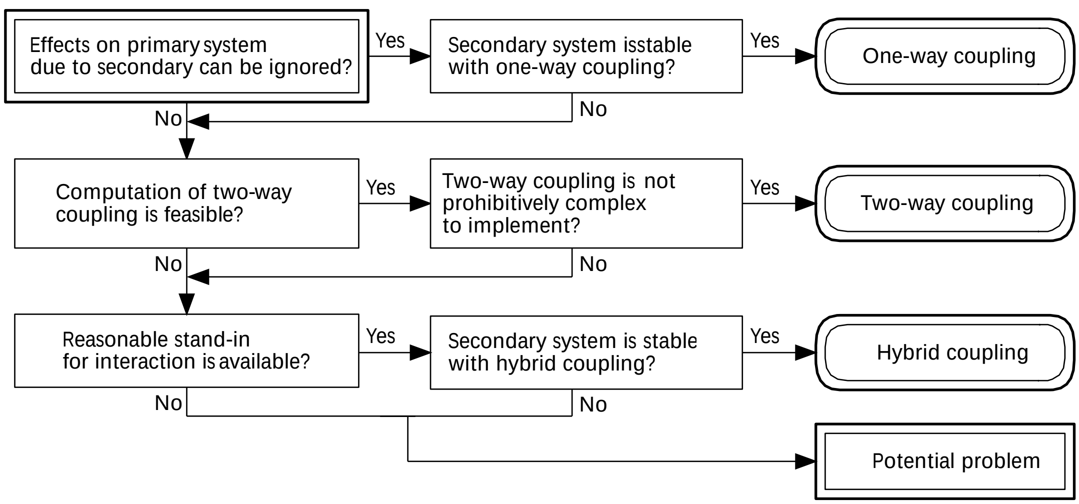
Gymnast and trampoline

The simulation of a gymnast on a trampoline, shown in Figure 14, is the most complex of our two-way coupled examples. To model this system correctly requires a physically realistic model of the gymnast, the trampoline, and the interactions between them, as well as a control system capable of dynamically balancing the gymnast on the deformable trampoline. The trampoline is a spring and mass system. We selected the parameters for the frame springs and for the bed of the trampoline to produce deformations matching those observed in still images and video footage under similar load conditions.[18] The control system resembles those described previously, but we used simulated annealing search techniques to automatically determine parameters that would allow the gymnast to bounce repeatedly. We chose this approach because the two-way coupled simulation of the gymnast and the trampoline was too slow for interactive hand tuning.

Other examples

In addition to the examples described above, our cou-

15 **Scenes from the animated short *Alien Occurrence*. Secondary elements include: (a) Robe being cast off, (a, b, c) moving drapes in background, (b) tassels on spears, (c, d) vest on condemned alien, and (c, d) noose.**



16 **Decision tree for selecting a coupling method.**

pling methodology was used to generate secondary motion for the animated short, *Alien Occurrence*. Based on the classic short story *An Occurrence at Owl Creek Bridge* by Ambrose Bierce, this animation portrays the sentencing, imagined escape, and final execution of the main character. Figure 15 shows some scenes from the animation with secondary motion generated using the techniques described in this article.

Selecting a coupling method

As the preceding examples demonstrate, the best coupling technique depends on the characteristics of the specific systems and the nature of the desired effect. For example, the splash created with one-way coupling between the runner's foot and the water is visually appealing, but if the animator needed to have the runner slip in the water, two-way or hybrid coupling would be required. The decision process can be facilitated by systematically examining issues such as complexity, computational speed, interactivity, and stability. Figure 16 shows a decision tree based on an analysis of these factors.

If the interaction doesn't have a significant effect on the primary system, we can take advantage of the simplicity and speed of one-way coupling. An interaction may be insignificant because the primary object isn't influenced by the interaction or because the effect is contextually unimportant. The influence on the primary system can be determined by measuring the effective acceleration due to the sum of the interaction forces. Interactions that cause very small accelerations or accelerations that are overwhelmed by other forces can probably be ignored.

Table 1. Force and acceleration data from selected simulations.

Primary		Secondary		Force (N)			Acceleration (m/s^2)		
Object	Mass (kg)	Object	Mass (kg)	Min	Max	Mean	Min	Max	Mean
Ball	0.68	Net	0.03	0.0	59.9	15.7	0.00	88.08	23.14
Gymnast	64.38	Trampoline	20.00	60.4	5298.8	2215.2	0.93	82.30	34.41
Alien	46.56	Vest	0.50	2.1	31.9	6.9	0.05	0.69	0.15
Alien	46.56	Noose	3.50	137.9	4055.3	575.0	2.96	87.10	12.35

Table 1 shows force and acceleration values for some of the examples presented in this article. For the one-way coupled clothing, the acceleration on the primary system is very small (less than 1 m/s^2), whereas for the two-way coupled trampoline, the accelerations are much larger (averaging 34 m/s^2). The minimum, maximum, and mean forces, in Newtons (N), are computed over the period of time that the objects are in contact or a 0.5-second interval in the case of sustained contact. The accelerations are the effective acceleration on the primary system due to the action of the secondary system. The rows of Table 1 correspond to Figures 3 (top row), 14, 15c, and 15d. The qualitative judgment about whether an effect is contextually significant is often determined by the desired level of realism. Objects that will be part of a busy background, far away from the camera, or partially obscured don't require the same level of realism as do objects that are the focus of attention.

When the interaction is contextually unimportant, system stability may still rule out the use of one-way coupling. Because the primary object's motion is not altered by the interaction, the secondary system can be driven into unstable configurations or deformed in a visually unappealing fashion.

When one-way coupling isn't feasible, the choice between two-way and hybrid coupling can be made based on the computational expense and the complexity of the implementation. Two-way coupling will result in a combined system that is, at best, as fast to compute as the slowest component and possibly much slower because the combined system will inherit the requirements of both systems. The greater computational cost may make the system unusable by increasing the debug cycle time beyond the user's interactivity threshold. Two-way coupling may also be prohibitively complex to implement because of the detailed physical laws that must be included to model the interaction accurately.

Hybrid coupling is a reasonable choice when a stand-in that cheaply models the salient elements of the interaction is available. For example, our hybrid systems often include vector fields that apply forces based on the object's position, orientation, and velocity. Like one-way coupling, hybrid coupling can lead to a problem with stability, although adjusting the parameters of the stand-in may alleviate the problem.

Finally, for some systems, the trade-off between realism and complexity doesn't yield a reasonable compromise. For these systems, one-way coupling is inadequate, two-way coupling is too expensive, and no suitable stand-in can be devised for hybrid coupling. The gymnast and trampoline fall into this category, and we had to employ automated search techniques in the gymnast's controller.

The parameters that determine the appropriate type of coupling may change during the development cycle. In particular, building two simulations and the interaction between them in stages help eliminate programming errors and stability problems before the full system is assembled. Furthermore, debugging an active system with a fast, hybrid-coupled system, then switching to two-way coupling may make designing an effective control system much easier.

Discussion and conclusions

In the physical world, all pairs of interacting objects are two-way coupled. The resulting movement includes a remarkable amount of perceptible detail. However, simulation is computationally expensive, and completely simulating even a simple real-world scene would be difficult on current computing hardware. For this reason, we explored three methods of coupling that allow a trade-off between speed and realism. By explicitly considering the interface between simulations, we've given the animator the ability to choose a suitable compromise. This decision about the appropriate level of coupling resembles the modeling decision about the level of detail required for a physical simulation.

While we focused on the interactions between active and passive systems, these techniques should apply to situations where both systems are passive or both are active. The components of the kites and the initial example of the ball and net demonstrate passive-to-passive coupling, but we haven't shown a system where two active systems are coupled together, such as would be required for pairs figure skating. The simulation of an active-to-active interaction would be similar to the active-to-passive examples, but both control systems would have to be robust enough to allow for the disturbances caused by the changes to the dynamic systems. Furthermore, when two active simulations cooperate to perform a single task, such as a ballet lift, the two control systems must coordinate the timing and purpose of the simulated actions.

The examples described above demonstrate that our approach of using coupled simulations is general, can be applied to a wide range of phenomena, and can add visual richness to an animated scene. While we simulated the secondary motion of many of the objects in the scene, a number of objects remain motionless. In some cases, modeling a few of the moving and flexible objects appears to emphasize the lack of motion in the others. Like the progression in models from wireframe to polygonal to subdivision surfaces, this increase in fidelity may also increase the viewer's expectations.

Animations corresponding to the figures in this article can be viewed online at http://www.cc.gatech.edu/gvu/animation/Areas/secondary/secondary.html. ■

Acknowledgments

We'd like to thank Wayne Wooten for his vaulting and diving simulations, and Nancy Pollard for her help with the automated tuning of the trampolinist's control system. The production of the animated short, *Alien Occurrence*, involved the dedicated efforts of many students at the Georgia Institute of Technology's Graphics, Visualization, and Usability Center.

This project was supported in part by NSF NYI Grant No. IRI-9457621, Mitsubishi Electric Research Laboratory, and a Packard fellowship. James F. O'Brien was supported by a fellowship from the Intel Foundation.

References

1. D. Baraff and A. Witkin, *Partitioned Dynamics*, Tech. Report CMU-RI-TR-97-33, Robotics Institute, Carnegie Mellon Univ., Pittsburgh, 1997.
2. T. DeRose, M. Kass, and T. Truong, "Subdivision Surfaces in Character Animation," *Proc. Siggraph 98*, Annual Conference Series, ACM, New York, July 1998, pp. 85-94.
3. D. Terzopoulos et al., "Elastically Deformable Models," *Proc. Siggraph 87*, ACM, New York, July 1987, pp. 205-214.
4. P. Volino, M. Courshesnes, and N. Magnenat-Thalmann, "Versatile and Efficient Techniques for Simulating Cloth and Other Deformable Objects," *Proc. Siggraph 95*, ACM, New York, Aug. 1995, pp. 137-144.
5. D. Baraff and A. Witkin, "Large Steps in Cloth Simulation," *Proc. Siggraph 98*, Annual Conference Series, ACM, New York, July 1998, pp. 43-54.
6. N. Foster and D. Metaxas, "Realistic Animation of Liquids," *Proc. Graphics Interface 96*, Canadian Human-Computer Communications Society, Toronto, May 1996, pp. 204-212.
7. J. Stam, "Stable Fluids," *Proc. Siggraph 99*, ACM, New York, Aug. 1999, pp. 121-128.
8. P. Witting, "Computational Fluid Dynamics in a Traditional Animation Environment," *Proc. Siggraph 99*, ACM, New York, Aug. 1999, pp. 129-136.
9. X. Li and J.M. Moshell, "Modeling Soil: Realtime Dynamic Models for Soil Slippage and Manipulation," *Proc. Siggraph 93*, ACM, New York, Aug. 1993, pp. 361-368.
10. R.W. Sumner, J.F. O'Brien, and J.K. Hodgins, "Animating Sand, Mud, and Snow," *Computer Graphics Forum*, Vol. 18, No. 1, Mar. 1999.
11. J.F. O'Brien and J.K. Hodgins, "Graphical Modeling and Animation of Brittle Fracture," *Proc. Siggraph 99*, ACM, New York, Aug. 1999, pp. 137-146.
12. N.I. Badler, C.B. Phillips, and B.L. Webber, *Simulating Humans: Computer Graphics Animation and Control*, Oxford Univ. Press, New York, 1993.
13. J.K. Hodgins et al., "Animating Human Athletics," *Proc. Siggraph 95*, ACM, New York, Aug. 1995, pp. 71-78.
14. X. Tu and D. Terzopoulos, "Artificial Fishes: Physics, Locomotion, Perception, Behavior," *Proc. Siggraph 94*, ACM, New York, July 1994, pp. 43-50.
15. M. Carignan et al., "Dressing Animated Synthetic Actors with Complex Deformable Clothes," *Proc. Siggraph 92*, ACM, New York, July 1992, Vol. 26, pp. 99-104.
16. J. Wejchert and D. Haumann, "Animation Aerodynamics," *Proc. Siggraph 91*, ACM, New York, July 1991, pp. 19-22.
17. J.F. O'Brien and J.K. Hodgins, "Dynamic Simulation of Splashing Fluids," *Proc. Computer Animation 95*, IEEE CS Press, Los Alamitos, Calif., Apr. 1995, pp. 198-205.
18. E. Phelps and B. Phelps, *Trampolining: The Skills of the Game*, Crowood Press, Ramsbury, Marlborough, Wiltshire, Great Britain, 1990.

James F. O'Brien *is a doctoral candidate in the College of Computing at the Georgia Institute of Technology, and a member of the Graphics, Visualization, and Usability Center. He received an MS in computer science from the Georgia Institute of Technology in 1996 and a BS in computer science from Florida International University in 1992. His research interests include physically based animation and geometric modeling. In 1997 he received a graduate fellowship from the Intel Foundation.*

Victor B. Zordan *is pursuing his PhD with the Graphics, Visualization, and Usability Center in the College of Computing at Georgia Institute of Technology. He received a BS degree in mechanical engineering from Boston University in 1992. His research currently focuses on automatic control techniques for simulated characters. He is interested in various forms of physical models and their interaction with visions of creating consistent and compelling animated worlds.*

Jessica K. Hodgins *received her PhD from the Computer Science Department at Carnegie Mellon University in 1989. She is currently an associate professor in the College of Computing at the Georgia Institute of Technology and a member of the Graphics, Visualization and Usability Center. Her research explores techniques that may someday allow robots and animated creatures to plan and control their actions in complex and unpredictable environments. In 1994 she received an NSF Young Investigator Award and was awarded a Packard Fellowship. In 1995 she received a Sloan Foundation Fellowship. She is editor-in-chief of* ACM Transactions on Graphics.

Readers may contact the authors at the Georgia Institute of Technology, Graphics, Visualization, and Usability Center, 801 Atlantic Dr., Atlanta, GA 30332-0280, e-mail {obrienj, victor, jkh}@cc.gatech.edu.