Combining Active and Passive Simulations for Secondary Motion

James F. O'Brien Victor B. Zordan Jessica K. Hodgins

College of Computing and Graphics, Visualization, and Usability Center
Georgia Institute of Technology
Atlanta, GA 30332-0280
[obrienj|victor|jkh]@cc.gatech.edu

Abstract

Varied, realistic motion in a complex environment can bring an animated scene to life. While much of the required motion comes from the characters, an important contribution also comes from the passive motion of other objects in the scene. We use the term *secondary motion* to refer to passive motions that are generated in response to environmental forces or the movements of characters and other objects. For example, the movement of clothing and hair adds visual complexity to an animated scene of a jogging figure. In this paper, we describe how secondary motion can be generated by coupling physically based simulations of passive systems to active simulations of the main characters. We discuss three coupling methods for the interface between the passive and active systems: *two-way*, *one-way*, and *hybrid*. These three methods allow the animator to make an appropriate tradeoff between accuracy and computational speed. We use a basketball passing through a net as an illustrative example to demonstrate each of the three coupling methods. To provide guidance as to when each method is most appropriate, we present additional examples including a gymnast on a trampoline, a man on a bungee cord, a stunt kite, a gymnast landing on a flexible mat, a diver entering the water, and several human figures wearing clothing. The information gained from analyzing these examples is summarized in a decision tree and a set of guidelines for coupling active and passive systems.

Keywords

Computer animation, human motion, dynamic simulation, physically realistic modeling, deformable models, coupled systems, clothing, water.

I. Introduction

Objects that move in response to the actions of the main character often make an important contribution to the visual richness of an animated scene. We use the term *secondary motion* to refer to these passive motions. Secondary motion may be created by background elements such as flags in the wind or by a main character's accessories such as hair or clothing. Figure 1 shows the secondary motion of a trampoline as a gymnast bounces on it and of a young girl's skirt as it moves in response to her swinging motion.

Secondary motions are not normally the main focus of an animated scene, yet the absence of secondary motion can distract or disturb the viewer, destroying the illusion created by the scene. For example, if the skirt in Figure 1 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 are an important part of a compelling animation.

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,

Submitted to IEEE: Computer Graphics & Applications.

Fig. 1. **An animated scene with secondary motion.** Both the swinger's skirt and the bed of the trampoline must move if the scene is to be convincing. Additional elements, such as the kites flying in the wind, further enhance the realism of the scene.

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 animating complex secondary motion. As a result, specialized procedural methods have been developed for many of these objects.

While procedural models may be derived in any of 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. For some applications, such as character animation, the increased 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, it is most often the movement of the actor that is exaggerated and the passive secondary motions simply respond to the exaggerated motion.

Simulation methods have 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 mimic the complex interactions that would occur in the real world. The primary contribution of this work is an exploration of the coupling issues involved when passive systems are coupled to active systems that have an internal source of energy and a control system to govern their behavior.

We explore three different methods for coupling two systems together: *two-way*, *one-way*, and *hybrid*. To clarify the differences between the 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 the contact with the net and the net will be pushed out of the way by the ball. If the coupling is one-way, the motion of the ball is not affected by the net 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 will not match that of an actual basketball and net as closely. In between these two solutions are a variety of hybrid coupled 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, substantial computational savings can be achieved with little loss of realism, while in others, a tight two-way coupling is essential. To help illustrate some of these issues, and to demonstrate the generality of our approach for generating secondary motion, we explore the construction of several example systems that are built by coupling simulated components together: 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.

II. Background

A number of techniques have been developed that use physically based simulation to generate motion for animation. Most of the research has focused on the issue of designing a simulation method for a particular type of phenomenon or motion and, with the exception of work by Baraff and Witkin [1], techniques for coupling simulations have been largely unexplored. In this paper, we specifically look at methods for coupling active and passive simulations. This section discusses previous 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 phenomenon. Cloth modeling, in particular, has been an active research area. Clothing simulation packages are beginning to be commercially available and clothing simulation was used successfully in the Oscar winning short Geri's Game [2]. Many of the techniques developed to model cloth are based on the spring and mass techniques originally introduced to the animation community by Terzopoulos and his colleagues [3, 4]. Breen, House, and Wozny use inter-particle constraints based on empirically derived energy functions to account for observed macroscopic behaviors [5]. Their work focuses on realistically predicting the drape of woven fabric as it collides with other objects. Other cloth systems based on finite element methods introduce self-collision and interaction with synthetic actors [6, 7]. Ng and Grimsdale have published a comprehensive survey of physical and geometric modeling techniques for cloth and clothing [8].

Most of the water models presented in the literature focus on specific phenomenon such as splashing, waterfalls, and spray. The techniques provide varying levels of realism and interaction with other objects. Sims used large particle systems to generate convincing waterfalls and spray [9]. Miller and Pearce used particle systems with inter-particle forces to animate streams of flowing water [10]. Terzopoulos, Platt, and Fleischer modeled viscous interaction forces to simulate a range of liquid behaviors including solid to liquid phase transitions [11]. Kass and Miller used a height field governed by shallow water equations to model bodies of water [12]. O'Brien and Hodgins used a hybrid particle/height field formulation to model water splashes and interactions between the water and objects floating on its surface [13]. Chen and his colleagues developed an interactive-rate simulation of fluid flow that solved 2D Navier-Stokes equations [14]. Foster and Metaxas used a variation of 3D Navier-Stokes equations to animate liquids in complex environments, with realistic object interaction and subtle 3D wave effects [15].

Other phenomenon, including wind and atmospheric effects, deformable terrain, and hair, have been modeled with varying levels of accuracy. We are particularly interested in those systems that can be combined with others for generating secondary motion. Wejchert and Haumann presented a flexible model for creating custom wind fields from flow primitives [16]. They used this system to drive the motion of flexible leaves blowing in the wind. Li and Moshell modeled soil slippage and manipulation [17]. 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 that modeled imprints in sand, snow, and mud created by dynamically simulated characters [18]. Anjyo, Usami, and Kurihara introduced a force-based cantilever system for modeling hair [19].

The use of simulation for active systems is not as widespread as it is for passive systems because robust control algorithms that produce natural looking motions are difficult to design with existing techniques. A number of hand-tuned simulations for rigid-body human characters have been introduced [20, 21, 22]. Active systems with spring-like actuators for non-human models such as snakes [23] and fishes [24] have been investigated as well. Other researchers have experimented with optimization techniques to automatically generate control systems for simulated characters [25, 26, 27, 28, 29].

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 [6] and manipulating soil with a bulldozer [17] are similar to what we term one-way coupling. However, in these cases, the general concept of coupling has not been investigated, and responsive active simulations were not considered.

The work of Baraff and Witkin [1] is most closely related to the work presented in this paper and we feel that their techniques are complementary to our own. They present a method for combining groups of passive systems including particle, clothing, and passive rigid-body models. Their work focuses on a method that uses constraints to allow multiple systems to interact. 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.

III. Coupling

Our goal is to combine simulations of individual objects or phenomena so that they can interact with each other to produce secondary motion. The individual techniques described in Section II focus on modeling the behavior of particular objects or phenomena using specific simulation techniques and we would like to build on this existing work. Thus, we adopt a modular approach where two or more systems are coupled together and we focus on the design of the interfaces between these systems.

For physically based simulations, forces applied between the systems provide a natural method for one simulation to interact with another. Interactions may be designed to compute the forces to be applied between the two systems and simplifying approximations can be made in the design of these interactions. 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 describe two-way, one-way, and hybrid coupling. To illustrate the differences between these coupling techniques, we use an example of a basketball going through a net. In this simple example, the primary system is the basketball and the net is the secondary system. The ball is modeled as a spherical rigid body that is free to translate and rotate in space. The ball is initialized with a linear and angular velocity determined by the animator and once in flight it experiences a gravitational acceleration. The net is modeled with a spring and mass network that is held in place by springs connected to the hoop rim which is fixed in space. The mass points experience forces due to gravity and the actions of the springs. The collisions between the net and the ball are the interaction that we aim to model.

A. Two-Way Coupled

While any computer simulation will involve some level of approximation, the goal of a two-way coupled simulation is to model 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.

While this tight coupling allows realistic interactions, it may also make the combined simulation difficult to work with. Because the two systems must be computed in lock step, it is not possible to preview the result of one simulation without also computing the result of the other. If one simulation is significantly slower than the other, this constraint can be a serious drawback.

The computation time required to calculate the result of the coupled simulation is often greater than the total time required to simulate each of the systems separately, even allowing

Fig. 2. **Simulated basketball and net under different coupling conditions.** The yellow line highlights the path of the ball generated using two-way, one-way, and hybrid coupling. Images are sampled at 0.067 second intervals.

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 where the other system has a large cost for each time step but is stable at large time steps. Either system alone may 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 computational cost.

We implemented the basketball and net as a two-way coupled system. To model the interaction forces, collision constraints are imposed to prevent the points of the net's mesh from penetrating the surface of the basketball. When contact is detected, 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. Friction is implemented using a 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 path of the ball with two-way coupling is shown in the top sequence of images in Figure 2. 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 ball is slowed. Finally, the ball drops out from the bottom of the net with a substantially altered trajectory.

The main drawback to this type of coupling is the computation time required before the path of the ball may be viewed. The parameters for the net have been chosen to represent nylon cord which, while light and flexible, is highly resistant to 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. Furthermore, computing a time step for the ball is computationally much less expensive than computing a time step for the mesh. If simulated alone, the ball could be computed in real time on even very modest machines, 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 ball's path can be viewed. We refer to the time between specifying parameters and viewing the results as the debug cycle time. Because the animator would most likely be concerned mainly with the path of the ball, or in a more complex situation, the motion of the primary character, an extended primary debug cycle time is highly undesirable.

B. 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 if they were applied. 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 the active primary system would be able to trivially correct for any disturbances caused by the interaction with the secondary system.

The main benefit derived from this method of coupling is that the two systems may be simulated separately, avoiding an increased primary debug cycle time. This type of coupling is also easier to implement than two-way coupling because only the secondary system must be modified.

If the assumption that the interaction would have had a minimal effect on the primary system is wrong, then 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 will not notice consciously, but will nonetheless find disturbing and distracting. Additionally, the secondary system may be forced to violate interaction constraints because the primary object's motion will not be altered regardless of the magnitude of the interaction forces.

We also implemented the basketball and net as a one-way coupled system to illustrate this coupling technique. The interaction forces are computed as before, but no forces are applied to the ball. The resulting motion can be seen in the center row of Figure 2. The ball's path is not affected by the net and the resulting trajectory is unrealistic. The net is forced to stretch a great deal, despite its stiff material parameters, potentially causing the internal spring forces to become so large that a violation of the collision constraints occurs (fifth image of second row in Figure 2). Because the net is substantially deformed by the interaction, it becomes prone to instability and requires a time step of 10^{-6} seconds (compared to 10^{-5} seconds for the two-way coupled simulation).

C. Hybrid Coupled

A hybrid coupled 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 motion of the primary system is then used to drive the secondary system as in the one-way coupled case.

While the simulation of the secondary system must model all the visible behaviors of the secondary object, the stand-in only needs to 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 are many possible stand-ins with various levels of physical realism, and the appropriate stand-in depends on the level of realism required by the interaction.

The bottom row of Figure 2 shows the results of implementing the basketball and net as a hybrid coupled simulation. The effect of the net on the ball is approximated with a damping field co-located with the rest configuration of the net. As the ball passes through the field, its translational and rotational momentum are damped according to parameters selected by the animator. Once the path of the ball has been determined, the net is simulated using the generated ball path.

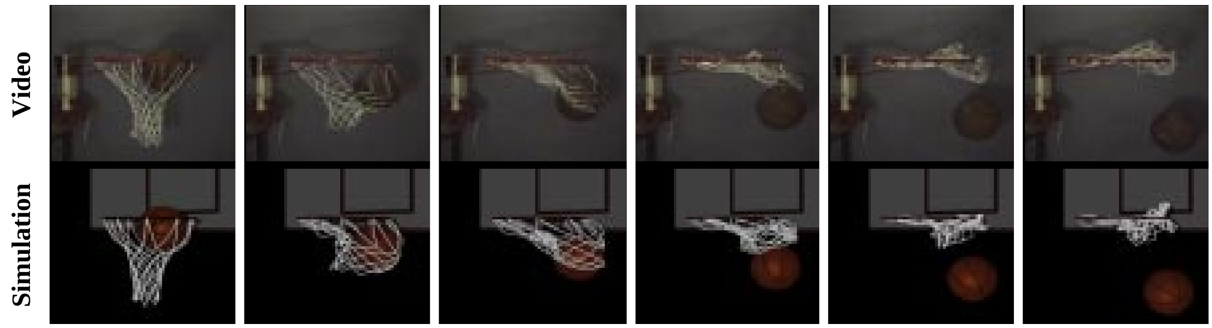

Fig. 3. **Comparison of simulation results for two-way coupling with video footage.** The top row of images shows frames captured from video footage of a real basketball and net. The bottom row shows a two-way coupled simulation with matching initial conditions. Images are sampled at 1/15 second intervals.

The motion of the ball generated with hybrid coupling is substantially different from the motions generated by the two- and one-way coupled simulations. In the hybrid simulation, the ball's horizontal and rotational velocity are slowed significantly, but, in contrast to the velocities seen with two-way coupling, they do not reverse because the simple approximation of a damping field is not capable of producing that behavior. However, the ball's path and the motion of the net are qualitatively similar to that seen with two-way coupling. The path generated by the hybrid simulation is also significantly different from the parabolic trajectory of the one-way coupled system and the net is not stretched in an unrealistic fashion.

The design and parameters of the approximation used for the hybrid coupled simulation provide 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 is likely to 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.

D. Comparison: 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 the results of the other two methods were compared. The true standard, however, is the motion of a real basketball and net. Figure 3 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.

We chose the basketball and net system as an example because it is a familiar system that is relatively simple to work with, yet still has sufficient complexity to be interesting. Because of the simplicity of the basketball and net simulation, we were able to implement it using each of the three coupling methods. For the remaining examples, we discuss a single implementation that employs the most suitable coupling technique for each pair of simulations.

IV. Simulation Methods

In this section, we briefly describe the details of the individual simulation techniques that we use as building blocks for constructing coupled systems. In Section V, we describe several examples of secondary motion created by coupling these individual simulation techniques together.

A. Spring and Mass Systems

We have found spring and mass systems to be an effective technique for modeling the behavior of a range of deformable materials including cloth, rope, and rubber. The object is represented as a heterogeneous network of point masses connected by springs. The springs generate forces with a linear response to deformation:

$$\mathbf{f_s} = -k_s \frac{\mathbf{x_a} - \mathbf{x_b}}{||\mathbf{x_a} - \mathbf{x_b}||}(||\mathbf{x_a} - \mathbf{x_b}|| - l_0) \tag{1}$$

where $\mathbf{f_s}$ is the force applied to mass point $\mathbf{a}$ ($-\mathbf{f_s}$ to $\mathbf{b}$), k_s is the elastic spring constant, $\mathbf{x_a}$ and $\mathbf{x_b}$ are the locations of the mass points, and l_0 is the spring rest length.

Energy dissipation within the system is modeled using the Rayleigh damping technique where damping forces are resolved into mass proportional and stiffness proportional components [30]:

$$\mathbf{f_m} = -k_m \dot{\mathbf{x}}_\mathbf{a} \tag{2}$$

$$\mathbf{f_d} = -k_d(\mathbf{x_a} - \mathbf{x_b}) \frac{(\dot{\mathbf{x}}_\mathbf{a} - \dot{\mathbf{x}}_\mathbf{b}) \cdot (\mathbf{x_a} - \mathbf{x_b})}{(\mathbf{x_a} - \mathbf{x_b}) \cdot (\mathbf{x_a} - \mathbf{x_b})} \tag{3}$$

where $\mathbf{f_m}$ is the mass proportional damping force applied to mass point $\mathbf{a}$, $\mathbf{f_d}$ is the stiffness proportional damping force applied to mass point $\mathbf{a}$ ($-\mathbf{f_d}$ to $\mathbf{b}$), k_m and k_d are the damping constants, and $\dot{\mathbf{x}}$ is the velocity corresponding to position $\mathbf{x}$. Although excessively large values for k_m can make the model appear as if it were moving through a viscous fluid, a small amount of mass proportional damping is often required to prevent explicit integrators from becoming numerically unstable. Conversely, stiffness proportional damping does not improve system stability, indeed increasing the value of $\mathbf{f_d}$ is likely to make the system less stable. Stiffness proportional damping, however, is important because without it, the system will behave as if it were constructed out of a rubbery elastic material.

The spring connections for the mesh are determined automatically from a geometric description of the object, and additional springs that span pairs of adjacent connected springs are added to enable control over shear deformation and out of plane surface bending. Spring constants are assigned so as to maintain a constant stress to strain ratio by setting k_s equal to the desired stress to strain ratio divided by the nominal rest length of the spring. Damping constants k_m and k_d are proportional to the appropriate point mass or spring constant. The mass of each point is determined by either distributing the overall system mass uniformly or by allocating the mass based on the polygonal area of the triangles attached to the mass point.

In addition to gravity and the spring and damping forces, the mass points experience an aerodynamic force in some of the example systems. We use a simplified aerodynamic model based on summing vector field flow primitives as described by Wejchert and Haumann [16]. Triangular faces experience a drag and lift force:

$$\mathbf{f_a} = \alpha_n s||\mathbf{v}||(\hat{\mathbf{n}} \cdot \mathbf{v})\hat{\mathbf{n}} + \alpha_t s(\mathbf{v} - (\hat{\mathbf{n}} \cdot \mathbf{v})\hat{\mathbf{n}}) \tag{4}$$

where $\mathbf{f_a}$ is the aerodynamic force, α_n and α_t are drag and lift coupling coefficients, $\mathbf{v}$ is the relative wind velocity averaged over the face, s is the surface area of the face, and $\hat{\mathbf{n}}$ is the unit normal of the face. The aerodynamic force is then distributed over the three mass points that define the triangular face.

B. Articulated Rigid Body Systems

The human actors in the example systems are modeled using rigid bodies connected with rotary joints. The dynamic models are generated automatically by computing the mass and moment of inertia of each body part from its geometric description and anatomical density information. The equations of motion for each system were generated using a commercially available package, SD/Fast [31].

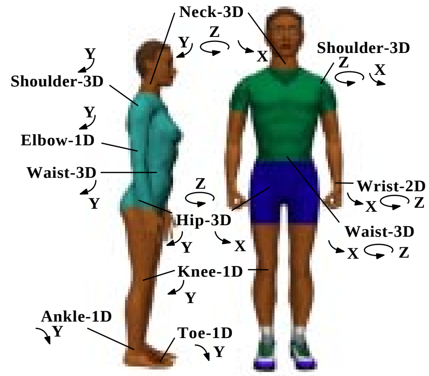

Fig. 4. **Simulation model for human figures.** The controlled degrees of freedom for male and female models. The models have 17 body segments and 30 controlled degrees of freedom.

The controlled degrees of freedom of the models are shown in Figure 4. Each internal joint of the model uses a torque source as a simple muscle model. The torques are computed using a proportional-derivative servo:

$$\tau = g_s(\theta_d - \theta) + g_d(\dot{\theta}_d - \dot{\theta}), \tag{5}$$

where g_s is the joint stiffness gain, g_d is the joint damping gain, θ and $\dot{\theta}$ are the joint position and velocity, and θ_d and $\dot{\theta}_d$ are the desired position and velocity.

The proportional-derivative servo is the lowest level of a control hierarchy. At the top level, a finite state machine selects appropriate control actions based on sensor events like ground contact or timing information. The control actions represent mid-level behaviors, such as balancing or leaping, and compute desired values for the joints of the simulated figure. The controllers used to generate the behaviors in this paper are described in more detail in previous publications [32, 22].

C. Water

To simulate the surface behavior of a body of fluid, we use a three-part system. Each subsystem models the effects of one aspect of the overall behavior of the fluid: the main volume, the free surface of the fluid, and disconnected components of the fluid (spray). Taken together, the subsystems, along with the interfaces between them, form a relatively simple, but visually appealing, water model [13].

To model the volume that makes up the main body of the fluid, we use a formulation that divides the body into a rectilinear grid of connected columns. The model assumes that all fluid properties are constant within a column. Pressure differences between adjacent columns induce flow through a set of virtual pipes that connect the columns.

The surface subsystem allows external objects to interact with the fluid system. Objects that collide with or float on the surface exert forces on the surface model. These forces are propagated as external pressure to the volume subsystem. The vertical positioning of the surface elements is determined by the volume of the columns.

A particle system is used to model droplets that are disconnected from the main body of the fluid. When an area of the surface has an upward velocity greater than a set threshold, particles are distributed uniformly over that area and the initial velocities for the particles are interpolated from the surface velocities. Once created, the particles fall under the influence of gravity and do not interact with each other. To conserve the total volume in the system, the volume of each particle is subtracted from the column from which it was created. When particles fall back onto the surface, the volume of the particle is added to the column that absorbed it.

D. Collision Detection and Response

Collision detection and response are essential for creating physically realistic coupling between independent simulated systems. Our collision detection algorithm is similar to that discussed by Snyder [33]. In preprocessing, both rigid and deformable bodies are broken into hierarchies of triangle sets. The leaf nodes of these hierarchies are individual triangles and an axis-aligned bounding box is stored at each leaf node. Interior nodes in the tree each have a bounding box that encompasses the union of the node's children. Separate objects each maintain their own hierarchies and when an object moves, the bounding boxes are recalculated to reflect the new configuration.

Collision detection is performed between two objects by testing for contact between the bounding box nodes for each object. Starting from the root nodes, testing is done recursively, until the objects are determined to not be in contact or until all triangle to triangle contacts have been detected. When a triangle contact is detected, the intersection locations are computed and used for collision response.

We compute collision response forces using a stabilized penalty constraint method with Coulomb friction. At each collision point, a normal and tangential force are computed:

$$\mathbf{f_n} = \mathbf{f_c} - [c_s \mathbf{p}_{\|} + c_d \dot{\mathbf{p}}_{\|}], \tag{6}$$

$$\mathbf{f_t} = -\mu |\mathbf{f_n}| \min(\xi, |\dot{\mathbf{p}}_{\perp}|) \frac{\dot{\mathbf{p}}_{\perp}}{|\dot{\mathbf{p}}_{\perp}|}, \tag{7}$$

where $\mathbf{f_c}$ is a constraint force that counteracts any acceleration pushing the two objects together, $\mathbf{p}_{\|}$ is the penetration vector, $\dot{\mathbf{p}}_{\|}$ and $\dot{\mathbf{p}}_{\perp}$ are the components of the relative velocity that are respectively parallel and perpendicular to $\mathbf{p}_{\|}$, μ is the coefficient of friction, c_s and c_d are stabilization coefficients, and ξ is an error tolerance. The normal and tangential force components are summed together and the resulting force is applied to the colliding objects at the appropriate contact points.

V. Example Systems

In Section III, we used the example of a basketball and net to illustrate the differences between the three coupling methods. In the following sections, we discuss additional examples and describe how they can be implemented as coupled systems. The examples are ordered approximately in ascending order of complexity. We focus on passive systems that are modeled with mass and spring systems or with simplified fluid dynamics models both for convenience and because these two modeling techniques are commonly used for animation. However, the ideas we describe should be applicable to other types of physically based systems.

Clothing. We have 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 that is generated automatically from a geometric model. Collisions between the clothing and the actor are detected by intersecting the triangle faces of the actor's polygonal model with the triangles of the clothing model. Figure 5 shows a runner wearing a tee-shirt and sweat pants and a child on a swing wearing a skirt.

Flags and Leaves. Figures 6 and 7 show flags and leaves blowing in the wind. These systems are examples of one-way coupled simulations that are modeled with a spring and mass system. In these simple simulations, the passive system is influenced only by environmental factors, such as wind. For example, the bicyclist shown in Figure 7 generates a wind field that stirs up leaves in the road as he moves past them. Because the actor does not experience any forces due to the motion of the leaves, the system is one-way coupled.

Fig. 5. **Synthetic actors wearing simulated clothing.** While the clothing worn by the actors moves in response to their actions, the effect of the clothing on the runner and child on the swing is negligible.

Fig. 6. **Simulated flags in the wind.** Flags are examples of simple background elements that move in response to environmental effects such as wind or the motion of other objects.

Fig. 7. **Spring and mass model of a leaf.** Like flags, leaves do not interact directly with the main characters. They are influenced indirectly by wind fields that are generated by moving objects, such as the bicyclist. The diagram on the right shows the texture map and the spring and mass network used to model a leaf.

Floor Mat. Figure 8 shows a gymnast landing on a deformable floor mat after performing a hand-spring vault. The floor mat makes the scene appear more realistic by softening the landing of the gymnast and by deforming to create a visual connection between the gymnast and the rest of the scene.

The floor mat is modeled using a mass and spring system and the gymnast is modeled with a controlled hierarchy of rigid bodies. 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 coupled solution. The forces applied to the gymnast's feet are computed as if she were landing on a grid of vertical springs that cannot slide horizontally or rotate. Although this simple model will not capture subtle effects, such as a 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 is used to drive the floor mat simulation and produce the desired deformation of the mat.

Water. The type of coupling that we use for water depends on the role that the water will play in the scene. Figure 9 shows a runner stepping in a puddle. Because his motion is not significantly affected by the water, a one-way coupling is used to model the interaction. On the other hand, a diver entering the water from a 10 meter platform should be significantly affected by the water (Figure 10). However, the degree to which the viewer is able to observe this effect is limited, particularly if the diver enters the water vertically. Therefore we use a hybrid coupling

Fig. 8. **Gymnast landing on a deformable mat.** This close up shows a gymnast landing on a deformable floor mat after a hand-spring vault. The give of the mat prevents the landing from having a painful, bone jarring appearance and the subsequent deformation creates an important connection between the actor and the background.

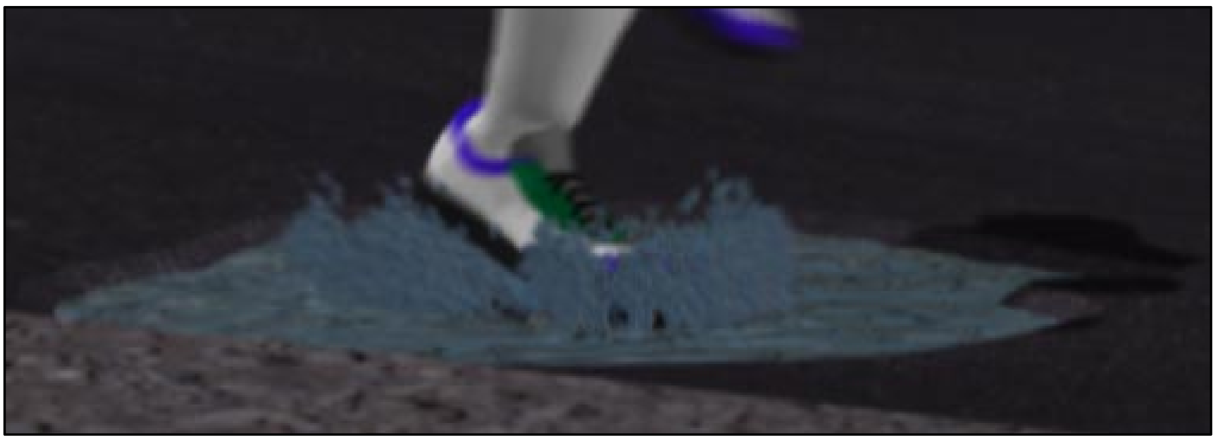

Fig. 9. **Foot stepping in a puddle of water.** Although the runner's motion is unaffected, the impact of the foot causes a splash.

where the diver encounters a viscous damping field that exerts drag forces on the parts of the diver that are below the water. The resulting motion is then used to drive the water simulation. A final water example involves objects floating on the surface of a pond, as shown in Figure 11. Two-way coupling is required here because the water's motion is affected by the motion of the floating objects and their motion is in turn affected by the water.

Kites and Stunt Kite. In addition to modeling the interactions between completely different 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, assumptions and simplifications can be made that are consistent with the specific qualities desired in the resulting motion for each component. We have used this approach to model kites flying in the air. We break each kite up into four components: cloth wing, frame, bridle and string, and tail, as shown in Figures 12 and 13.

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 simplified aerodynamic model described previously. The wing ripples and deforms as the wind acts on it, which in turn causes variation in the net aerodynamic forces that propagate to the frame and creates subtle variations in the kite's motion. The drag on the tail serves to stabilize the system.

In addition to the single-line kite, we have modeled a double-line stunt kite. The ground-ward end points of the strings are moved by a control system that directs the path of the kite mush as a person would fly a real stunt kite.

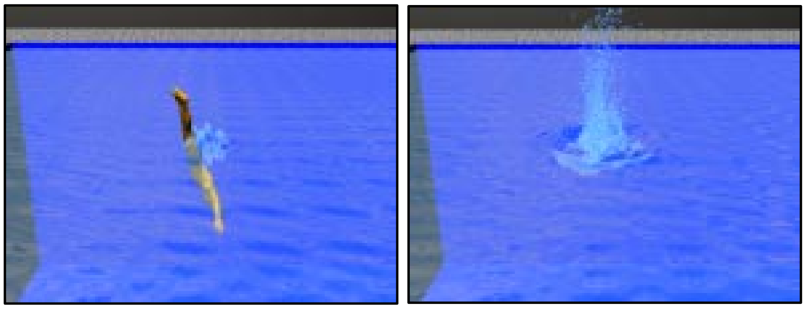

Fig. 10. **Diver entering the water.** As the diver enters the water, he slows down due to viscous drag and creates a splash.

Fig. 11. **Balls floating in water.** 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, more dense ball creates a larger disturbance while sinking through the water. The motions of the floating balls are affected by the motion of the water.

Bungee Jumper. The bungee jumper is an example of a two-way coupled system where the interactions play an important role in determining the motions of both the primary and the secondary objects. The bungee jumper is modeled with a rigid body hierarchy and the bungee cord is modeled with a spring and mass system. Because the cord does not significantly affect the motion of the jumper until after he has finished his leap from the platform, we tune the jumper's control system with the cord simulation disabled. When we are satisfied with the motion for the leap, we then run the two-way coupled system with the cord simulation enabled.

Gymnast and Trampoline. The simulation of a gymnast on a trampoline, shown in Figure 15, 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. Parameters for the frame springs and for the bed of the trampoline were selected to produce deformations matching those observed in still images and video footage under similar load conditions [34]. The control system is similar to those described previously, but simulated annealing search techniques were used 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 run too slowly for interactive hand tuning.

Other Examples. In addition to the examples described above, our coupling methodology has been 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. The images in Figure 16 show some scenes from the animation with secondary motion generated using the techniques described in this paper.

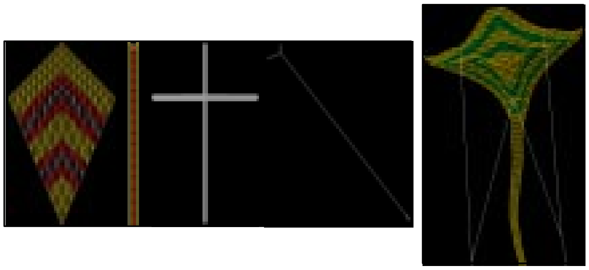

Fig. 12. **Diagram of kite assembly.** The four figures on the left show the components of the single line kite: the wing, tail, frame, and line. On the right, the two-line stunt kite is shown assembled.

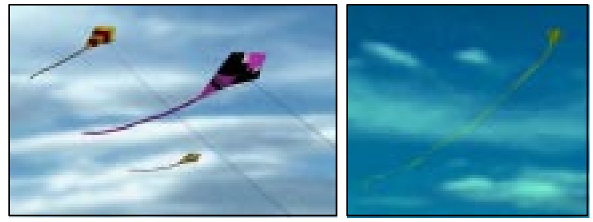

Fig. 13. **Kites in the sky.** The image on the left shows a close up of three single line kites in the air. The image on the right shows the two-line stunt kite as it performs a looping maneuver.

Fig. 14. **Bungee 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 feet.

VI. Selecting a Coupling Method

As the preceding examples demonstrate, the best coupling technique depends on the characteristics of the specific systems. Indeed, for a given pair of systems, the best choice is often influenced by the nature of the desired effect. For example, the splash created with a 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, a two-way or hybrid coupling would be required. The decision process can be greatly facilitated by systematically examining issues such as complexity, computational speed, interactivity, and stability. A decision tree based on an analysis of these factors is shown in Figure 17.

If the interaction does not have a significant effect on the primary system, then we can take advantage of the simplicity and speed of one-way coupling. An interaction may be insignificant because the primary object is not 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 I shows force and acceleration values for some of the examples presented in this paper. For the one-way coupled clothing, the acceleration on the primary system is very small (less than $1\,\mathrm{m/s^2}$), whereas for the two-way coupled trampoline, the accelerations are much larger (averaging $34\,\mathrm{m/s^2}$). The qualitative judgment about whether an effect is contextually significant is influenced by the desired level of realism. Objects that will be part of a busy background, far away from the camera, or partially obscured do not 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. The resulting instability may be handled with a smaller integration time step, but this solution increases the running time of the

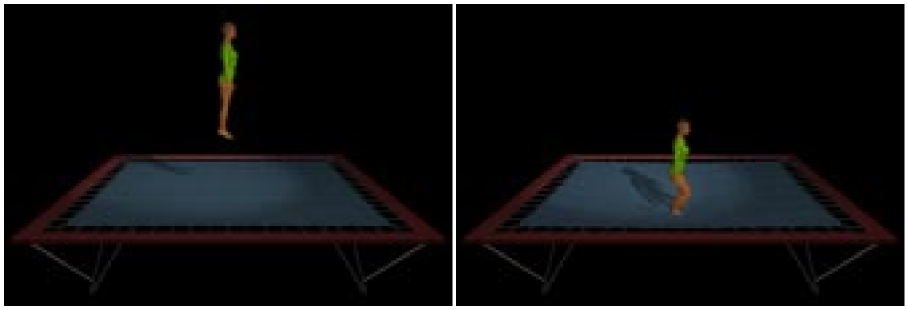

Fig. 15. **Gymnast on a deformable trampoline.** This system must be two-way coupled because the interaction between the gymnast and the trampoline has a significant effect on the motions of both systems. The first image shows the gymnast in a layout position prior to landing, the second image shows her as she hits the bed of the trampoline.

Fig. 16. **Scenes from the animated short *Alien Occurrence*.** Secondary elements include: Robe being cast off (A), moving drapes in background (A,B,C), tassels on spears (B), vest on condemned alien (C,D), and noose (C,D).

simulation and is not always a useful approach.

When one-way coupling is not 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. 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 accurately model the interaction.

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 stability problems, although the parameters of the stand-in can sometimes be adjusted to help alleviate the problem.

Finally, there are some systems for which the tradeoff between realism and complexity does not yield a reasonable compromise. For these systems, one-way coupling is inadequate, two-way

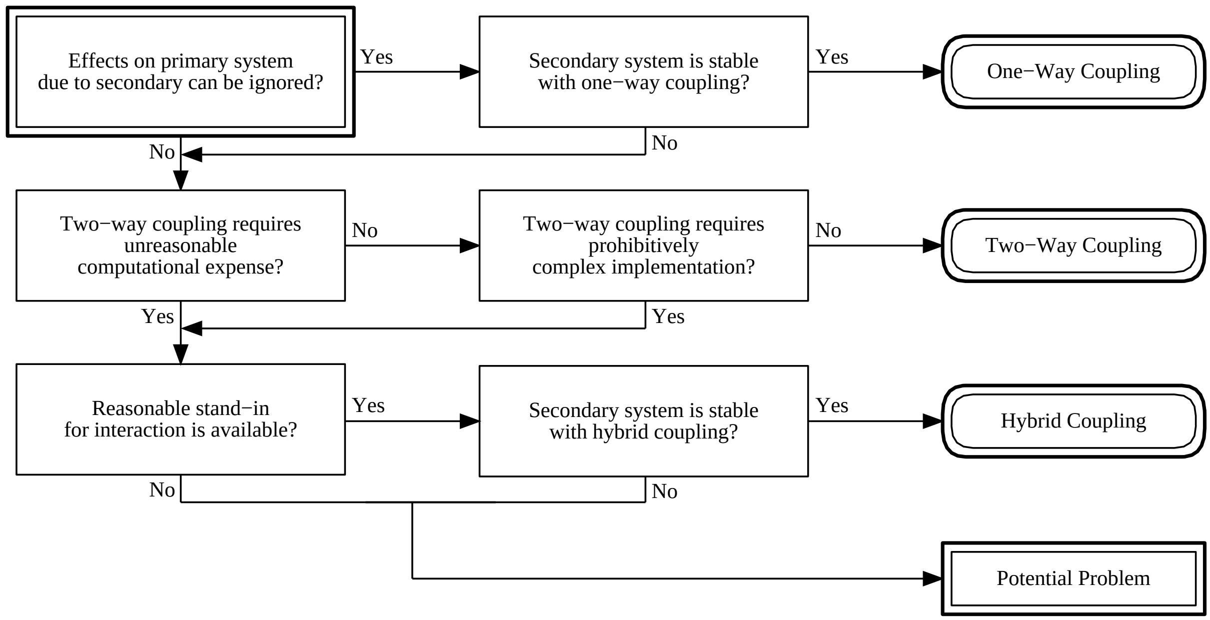

Fig. 17. **Decision tree for coupling selection.** This diagram outlines the process of selecting an appropriate method for coupling active and passive systems. As discussed in the text, it is intended as a general guide and the specific details of individual systems may dictate different choices.

Primary		Secondary		Force (N)			Acceleration (m/s²)		
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	Noose	3.50	137.9	4055.3	575.0	2.96	87.10	12.35
Alien	46.56	Vest	0.50	2.1	31.9	6.9	0.05	0.69	0.15

Table I. **Force and acceleration data from selected simulations.** This table shows the interaction forces that occur between two-way coupled simulations. The min, max, and mean forces are computed over 0.5 second intervals or the period of time that the objects are in contact. The accelerations are the effective acceleration on the primary system due to the action of the secondary system. The rows of this table correspond to Figures 2(top row), 15, 16.d, and 16.d.

coupling is too expensive, and no suitable stand-in can be devised for hybrid coupling.

The decision tree shown in Figure 17 is intended as a general guide for selecting a coupling technique. 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 allows programming errors and stability problems to be eliminated before the full system is assembled. Furthermore, debugging an active system with a fast, hybrid coupled system and then switching to two-way coupling may make designing the control system much easier.

VII. Discussion and Conclusions

In the physical world, all interacting objects are two-way coupled and 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 have explored three methods of coupling that allow a tradeoff between speed and realism. By explicitly considering the interface between simulations, we have given the animator the ability to choose a suitable compromise. This decision about the appropriate level of coupling is similar to the modeling decision about the level of detail required for a physical simulation.

Realism is not always the primary goal when creating an animation and the animator often exaggerates motions to more strongly convey a particular impression or emotional content [35]. However, the need for exaggeration does not preclude the use of simulation, because most often it is the motion of the main character that is exaggerated while the secondary objects merely respond to the primary motion.

While we have focused on the interactions between active and passive systems, these techniques should be applicable 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 have not 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 interaction. Furthermore, when two active simulations are cooperating to perform a single task, such as a lift during a ballet dance, the two control systems must coordinate the timing and purpose of their actions.

Simulation is a powerful technique for animation and in particular for generating secondary motion. Unfortunately, using a simulation often requires manipulating unintuitive parameters to specify physical properties and to determine the behavior of the control system. Developing user-friendly techniques for high-level specification of these parameters remains an important research area. One possible improvement would be to automatically tune the parameters based on an evaluation function, user-guided optimization, or comparison with real world footage.

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 help add visual richness to an animated scene. While we have simulated the secondary motion of many of the objects in the scene, a number objects are still 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, this increase in the fidelity of the modeling may increase the viewer's expectations.

Acknowledgments

The authors would 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. The first author was supported by a Fellowship from the Intel Foundation.

References

[1] David Baraff and Andy Witkin, "Partitioned dynamics", Tech. Rep. CMU-RI-TR-97-33, Carnegie Mellon University, 1997.

[2] Pixar, *Geri's Game*, 1998.

[3] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer, "Elastically deformable models", in *Proceedings of SIGGRAPH '87*. ACM SIGGRAPH, July 1987, pp. 205–214, Held in Anaheim, California, 27 - 31 July 1987.

[4] Demetri Terzopoulos and Kurt Fleischer, "Deformable models", *The Visual Computer*, vol. 4, no. 6, pp. 306–331, Dec. 1988.

[5] David E. Breen, Donald H. House, and Michael J. Wozny, "Predicting the drape of woven cloth using interacting particles", in *Proceedings of SIGGRAPH '94*. ACM SIGGRAPH, July 1994, pp. 365–372, Held in Orlando, Florida, 24-29 July 1994.

[6] Michel Carignan, Ying Yang, Nadia Magnenat Thalmann, and Daniel Thalmann, "Dressing animated synthetic actors with complex deformable clothes", in *Proceedings of SIGGRAPH '92*. ACM SIGGRAPH, July 1992, vol. 26, pp. 99–104, Held in Chicago, Illinois, 26-31 July 1992.

[7] Pascal Volino, Martin Courshesnes, and Nadia Magnenat Thalmann, "Versatile and efficient techniques for

simulating cloth and other deformable objects", in *Proceedings of SIGGRAPH '95*. ACM SIGGRAPH, Aug. 1995, pp. 137–144, Held in Los Angeles, California, 06-11 August 1995.

[8] Hing N. Ng and Richard L. Grimsdale, "Computer graphics techniques for modeling cloth", *IEEE Computer Graphics and Applications*, vol. 16, no. 5, pp. 28–41, Sept. 1996.

[9] Karl Sims, "Particle animation and rendering using data parallel computation", in *Proceedings of SIGGRAPH '90*. ACM SIGGRAPH, Aug. 1990, pp. 405–413, Held in Dallas, Texas, 6-10 August 1990.

[10] Gavin Miller and Andrew Pearce, "Globular dynamics: A connected particle system for animating viscous fluids", *Computers and Graphics*, vol. 13, no. 3, pp. 305–309, 1989.

[11] Demetri Terzopoulos, John Platt, and Kurt Fleischer, "Heating and melting deformable models (from goop to glop)", in *Proceedings of Graphics Interface '89*. Canadian Human-Computer Communications Society, June 1989, pp. 219–226, Held in London, Ontario, 19-23 June 1989.

[12] Michael Kass and Gavin Miller, "Rapid, stable fluid dynamics for computer graphics", in *Proceedings of SIGGRAPH '90*. ACM SIGGRAPH, Aug. 1990, vol. 24, pp. 49–57, Held in Dallas, Texas, 6-10 August 1990.

[13] James F. O'Brien and Jessica K. Hodgins, "Dynamic simulation of splashing fluids", in *Proceedings of Computer Animation '95*, Apr. 1995, pp. 198–205, Held in Geneva, Switzerland, 19-21 April 1995.

[14] Jim X. Chen, Niels da Vitoria Lobo, Charles E. Hughes, and J. Michael Moshell, "Real-Time fluid simulation in a dynamic virtual environment", *IEEE Computer Graphics & Applications*, vol. 17, no. 3, pp. 52–61, May-June 1997.

[15] Nick Foster and Demitri Metaxas, "Realistic animation of liquids", in *Proceedings of Graphics Interface '96*. Canadian Human-Computer Communications Society, May 1996, pp. 204–212, Held in Toronto, Ontario, 22-24 May 1996.

[16] Jakub Wejchert and David Haumann, "Animation aerodynamics", in *Proceedings of SIGGRAPH '91*. ACM SIGGRAPH, July 1991, pp. 19–22, Held in Las Vegas, Nevada, 28 July - 2 August 1991.

[17] Xin Li and J. Michael Moshell, "Modeling soil: Realtime dynamic models for soil slippage and manipulation", in *Proceedings of SIGGRAPH '93*. ACM SIGGRAPH, Aug. 1993, pp. 361–368, Held in Orlando, Florida, 24-29 July 1994.

[18] Robert W. Sumner, James F. O'Brien, and Jessica K. Hodgins, "Animating sand, mud, and snow", in *Proceedings of Graphics Interface '98*, 1998.

[19] Ken-ichi Anjyo, Yoshiaki Usami, and Tsuneya Kurihara, "A simple method for extracting the natural beauty of hair", in *Proceedings of SIGGRAPH '92*. ACM SIGGRAPH, July 1992, pp. 111–120, Held in Chicago, Illinois, 26-31 July 1992.

[20] Dinesh Pai, "Programming anthropoid walking: Control and simulation", Tech. Rep. 90-1178, Cornell Computer Science, 1990.

[21] Norman I. Badler, Cary B. Phillips, and Bonnie Lynn Webber, *Simulating Humans: Computer Graphics Animation and Control*, Oxford University Press, New York, 1993.

[22] Jessica K. Hodgins, Wayne L. Wooten, David C. Brogan, and James F. O'Brien, "Animating human athletics", in *Proceedings of SIGGRAPH '95*. ACM SIGGRAPH, Aug. 1995, pp. 71–78, Held in Los Angeles, California, 06-11 August 1995.

[23] Gavin Miller, "The motion dynamics of snakes and worms", in *Proceedings of SIGGRAPH '88*, Aug. 1988, pp. 169–178, Held in Atlanta, Georgia, 1-5 August 1988.

[24] Xiaoyuan Tu and Demetri Terzopoulos, "Artificial fishes: Physics, locomotion, perception, behavior", in *Proceedings of SIGGRAPH '94*. ACM SIGGRAPH, July 1994, pp. 43–50, Held in Orlando, Florida, 24-29 July 1994.

[25] Andrew Witkin and Michael Kass, "Spacetime constraints", in *Proceedings of SIGGRAPH '88*. ACM SIGGRAPH, Aug. 1988, pp. 159–168, Held in Atlanta, Georgia, 1-5 August 1988.

[26] Michael F. Cohen, "Interactive spacetime control for animation", in *Proceedings of SIGGRAPH '92*. ACM SIGGRAPH, July 1992, vol. 26, pp. 293–302, Held in Chicago, Illinois, 26-31 July 1992.

[27] Michiel van de Panne and Eugene Fiume, "Sensor-actuator networks", in *Proceedings of SIGGRAPH '93*. ACM SIGGRAPH, Aug. 1993, pp. 335–342, Held in Orlando, Florida, 24-29 July 1994.

[28] J. Thomas Ngo and Joe Marks, "Spacetime constraints revisited", in *Proceedings of SIGGRAPH '93*. ACM SIGGRAPH, Aug. 1993, pp. 343–350, Held in Orlando, Florida, 24-29 July 1994.

[29] Karl Sims, "Evolving 3d morphology and behavior by competition", in *Artificial Life IV*, 1994, pp. 28–39.

[30] R. Cook, D. Malkus, and M. Plesha, *Concepts and Applications of Finite Element Analysis*, John Wiley & Sons, 3rd edition, 1989.

[31] Michael Hollars, Dan Rosenthal, and Michael Sherman, *SD-Fast Users's Manual*, Symbolic Dynamics, Inc., Mountain View, CA, b.1.1 edition, 1991.

[32] Wayne L. Wooten and Jessica K. Hodgins, "Simulation of human diving", in *Graphics Interface '95*. Canadian Human-Computer Communications Society, May 1995, pp. 1–9.

[33] John M. Snyder, "An interactive tool for placing curved surfaces without interpenetration", in *Proceedings of SIGGRAPH '95*. ACM SIGGRAPH, Aug. 1995, pp. 209–218, Held in Los Angeles, California, 06-11 August 1995.

[34] Erika Phelps and Brian Phelps, *Trampolining: The skills of the Game*, The Crowood Press, Ramsbury, Marlborough, Wiltshire, 1990.

[35] Frank Thomas and Ollie Johnston, *Disney Animation: The Illusion of Life*, Abbeville Press, New York, 1984.