

Simulation and Control of Skeleton-driven Soft Body Characters

Libin Liu*

KangKang Yin[†]

Bin Wang[‡]

Baining Guo[‡]

*Tsinghua University

[†] National University of Singapore

[‡] Microsoft Research Asia

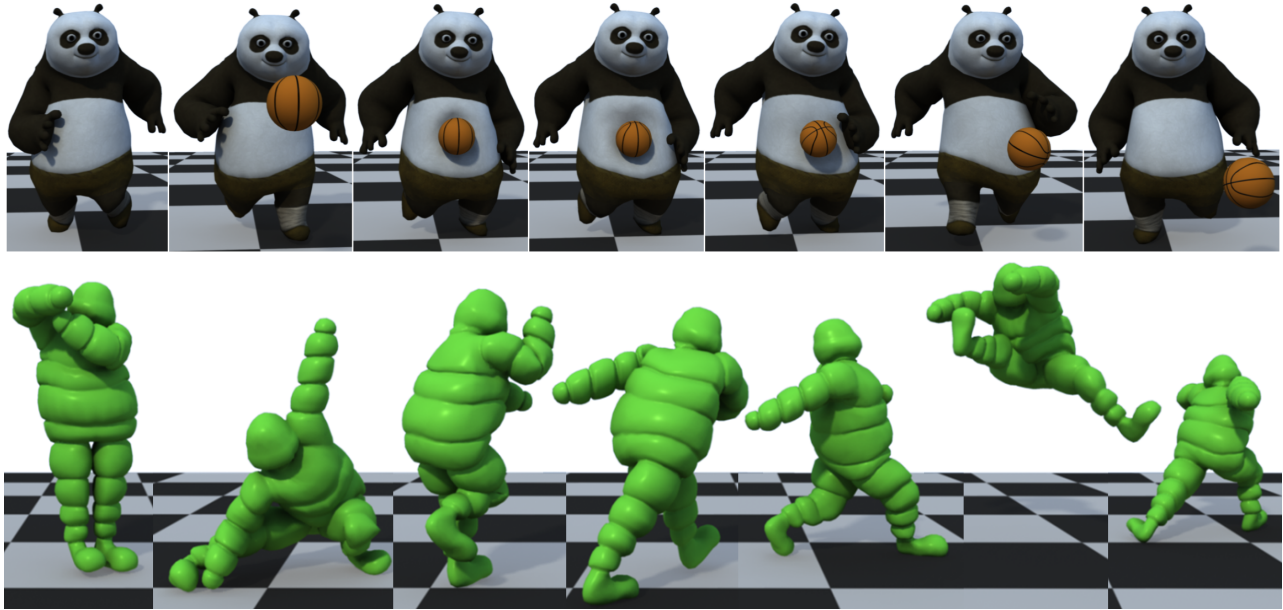


Figure 1: Our Panda model runs and responds to external perturbations at interactive rates. Our Michelin model does Kung Fu moves.

Abstract

In this paper we present a physics-based framework for simulation and control of human-like skeleton-driven soft body characters. We couple the skeleton dynamics and the soft body dynamics to enable two-way interactions between the skeleton, the skin geometry, and the environment. We propose a novel pose-based plasticity model that extends the corotated linear elasticity model to achieve large skin deformation around joints. We further reconstruct controls from reference trajectories captured from human subjects by augmenting a sampling-based algorithm. We demonstrate the effectiveness of our framework by results not attainable with a simple combination of previous methods.

Keywords: physics-based animation, motion control, soft body simulation, finite element method, plasticity model

Links: [DL](#) [PDF](#)

*e-mail: llb05@mails.tsinghua.edu.cn

[†]e-mail: {kkyin, wangb}@comp.nus.edu.sg

[‡]e-mail: bainguo@microsoft.com

1 Introduction

Recent advances in human-like articulated rigid body character control have demonstrated robust results for basic locomotion [Yin et al. 2007; Coros et al. 2010] and highly dynamic motions [Liu et al. 2012; Brown et al. 2013]. The rigidity assumption, however, is not applicable to fat characters, such as a panda with a big belly, whose skin and flesh deformations inevitably affect the underlying skeleton dynamics, especially in highly dynamic tasks. Simulation and control of human-like soft characters is extremely challenging due to coupling between the skeleton and the soft body dynamics, complexity of human skills, large numbers of Degrees of Freedom (DoFs), and large ranges of motion. In the long run, biomechanical approaches that truly model human anatomy are probably needed to completely solve the problem [Lee et al. 2009], but their high modeling and computational costs are prohibitive for graphics applications such as games in the foreseeable future.

We propose an affordable simulation and control framework for human-like soft body characters. Our interactive simulation framework is unique in its ability to conserve momentum, and its simplicity for reimplementation. We use the rigid skeletons for motion control and the surface geometries for deformation, and couple them properly to support two-way interactions. The rich literature on both topics provides basic building blocks for our system. However, a simple integration of prior arts does not work directly. First, joints of human-like characters have large ranges of motion and can rotate more than 90 degrees for example. The flesh around these joints thus experiences large deformations and exerts excessive elastic forces onto the bones when joints bend severely. Since the flesh and the bones are two-way coupled, these forces will prevent the bones from rotating to their target positions. This is not an issue for simple models such as fish demonstrated in previous work. But for human-like characters we need to properly address

this problem. In this paper, we introduce a novel pose-based plasticity model that extends the commonly used corotated linear elasticity model to achieve large deformations around skeletal joints.

Second, even though robust motion control algorithms exist for rigid human-like characters, their direct application for soft body character control often either fails or is computationally implausible. For example, in the sampling-based method of Liu et al. [2010], the initial open-loop PD targets specified by captured trajectories from human subjects are too far away from the solution when applied on soft body characters because: (a) our soft body characters are very different kinematically and dynamically from the human subjects; (b) the soft body dynamics adds another layer of complication and approximation; and (c) long airborne phases of some of the motions, such as jump kick and Kong Fu, add another dimension for failure. We present two key components, the Inverse Dynamics (ID) and the time scaling scheme, that enable successful control of soft body characters using the sampling-based construction method. More specifically, the ID provides a good initial solution, and the time scaling enables referencing the motion capture trajectories more flexibly.

2 Related Work

Control of rigid skeletons Recent years have seen significant progress on motion control of human-like characters abstracted as rigid skeletons, ranging from basic locomotion controllers [Yin et al. 2007; da Silva et al. 2008; Lee et al. 2010; de Lasa et al. 2010; Coros et al. 2010] to highly dynamic skills [Liu et al. 2010; Liu et al. 2012; Ha et al. 2012; Brown et al. 2013]. We build our motion control framework on the work of Liu and his colleagues [2010; 2012], due to its generality for a large range of motor skills.

Soft body simulation The seminal work of [Terzopoulos et al. 1987] stimulated various physics-based methods for simulating soft bodies. Among them we choose the corotational linear Finite Element Method (FEM) [Müller et al. 2002; Müller and Gross 2004] as our elasticity model because of its simplicity and rotational invariance. We use the mesh embedding technique of [Capell et al. 2002; Kim and Pollard 2011b] to deform a fine surface geometry embedded in simulated tetrahedral elements.

Coupling between rigid and soft bodies Most previous work on hybrid systems that contain soft as well as rigid parts only support one-way coupling between the two sub-systems, including physics-based skinning [Capell et al. 2005; McAdams et al. 2011; Hahn et al. 2012; Kavan and Sorkine 2012], and skeleton-driven skin deformations [Capell et al. 2002]. Shinar and colleagues proposed two-way coupling of rigid and deformable objects [Shinar et al. 2008]. Most related to our work is the two-way coupling of rigid skeletons and deformable skins for human-like characters [Kim and Pollard 2011b; Jain and Liu 2011], where the deformations of the soft flesh are not treated as just secondary animations but actually affect the underlying skeleton dynamics.

Control of soft bodies Various forms have been proposed for soft body control, including external forces [Barbič and Popović 2008], muscle fibers [Tan et al. 2012], rest shapes [Coros et al. 2012], and skeletons [Kim and Pollard 2011b; Kim and Pollard 2011a]. However, these methods often rely on optimization in a large parameter space with a great number of DoFs, which is computationally expensive and often runs into the problem of local minima. Therefore such methods are usually limited to simple characters and skills. The work of Kim and Pollard [2011b; 2011a] are the closest in spirit to our own. They propose a fast simulation

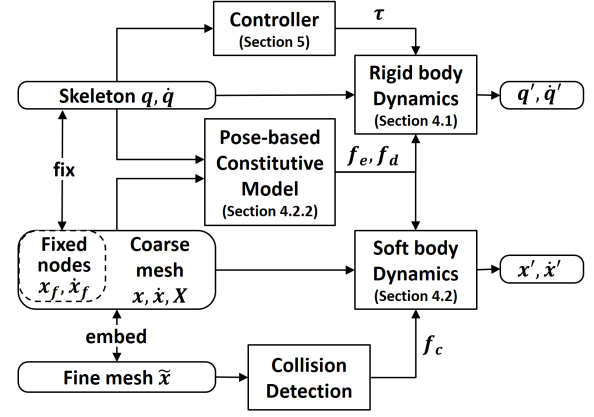


Figure 2: Overview of our simulation and control system.

framework for skeleton-driven soft characters. However, it is not clear in their work how to achieve more realistic large deformations around skeletal joints, or motion control for advanced motor skills. Skeleton control and deformable skin are also considered in [Jain and Liu 2011] and [Galoppo et al. 2007], but they handle the soft body dynamics and skeletal dynamics in an integrated fashion, while we simulate the two dynamical sub-systems separately (but coupled through fixed nodes). Thus we can achieve at least an order of magnitude faster simulation than the integrated dynamics scheme, which makes the control construction computationally tractable. Also only contact bodies (e.g. feet) are soft in [Jain and Liu 2011], while our system treats the full character as soft.

3 System Overview

Figure 2 illustrates the major components of our simulation and control system. The input to the system is a skeleton and a fine surface mesh representing the soft body character. A coarse volumetric mesh with a reference configuration X will then be constructed to couple the skeleton and the surface geometry. More specifically, the volumetric mesh is simulated with a soft body dynamics solver, and its state (x, \dot{x}) is used to update the shape of the fine surface mesh \tilde{x} ; the skeleton is simulated with a rigid body dynamics solver, and its state (q, \dot{q}) is used to constrain a subset of the nodes of the coarse mesh that are fixed to the skeleton.

4 Skeleton-driven Soft Body Dynamics

To explain the dynamics of our two-way skeleton-driven soft body character simulation framework, we first describe the articulated rigid body dynamics system, and then the soft body dynamics system. Afterwards we detail the coupling and integration of the two dynamics sub-systems.

4.1 Skeletal Dynamics

We model a skeleton as an under-actuated articulated rigid body system, and use PD-servos (Proportional Derivative) to actuate each DoF:

$$\tau = k_p(\tilde{q} - q) - k_d\dot{q} \quad (1)$$

where q and \tilde{q} represent the joint angles and their corresponding targets. The second term is a damping term that we will manipulate shortly to achieve stable PD controllers.

We augment the open-source Open Dynamic Engine (ODE) v0.12 to simulate the skeletal system. Denote the linear and angular velocities of the n rigid bodies of the articulated character as v . ODE

discretizes the equations of motion of the constrained multi-body system as:

$$\mathbf{M} \frac{\mathbf{v}_{t+1} - \mathbf{v}_t}{h} = \mathbf{f} + \mathbf{J}^T \lambda \quad (2)$$

$$\mathbf{J} \mathbf{v}_{t+1} = \mathbf{c} + \mathbf{C} \lambda \quad (3)$$

where \mathbf{M} is the inertia matrix of all bodies; \mathbf{J} is the constraint Jacobian, λ is the constraint force; and \mathbf{c} and \mathbf{C} are the error reduction term and the constraint stabilization term respectively. \mathbf{f} includes the inertia forces, joint torques, and external forces and torques.

Usually the PD torques τ in Equation 1 is directly accumulated to \mathbf{f} . Inspired by [Baraff and Witkin 1998; Tan et al. 2011], however, we compute the damping torques implicitly to achieve better stability. More specifically, we merge the first proportional term of Equation 1 into \mathbf{f} ; but move the derivative part to the left hand side of Equation 2 so that the damping term is computed from the velocity at the next time step as:

$$\mathbf{M} \frac{\mathbf{v}_{t+1} - \mathbf{v}_t}{h} + \mathbf{D} \mathbf{v}_{t+1} = \mathbf{f} + \mathbf{J}^T \lambda \quad (4)$$

We can then solve for the velocities at the next time step as:

$$\mathbf{v}_{t+1} = (\mathbf{M} + h\mathbf{D})^{-1} (\mathbf{M} \mathbf{v}_t + h\mathbf{f} + h\mathbf{J}^T \lambda) \quad (5)$$

Note that different from ODE's default solver, we need to invert the matrix $\mathbf{M} + h\mathbf{D}$ here. This matrix is sparse symmetric positive definite, however, so its inversion can be computed efficiently. We then solve for λ by substituting \mathbf{v}_{t+1} into Equation 3. The positions and rotations of the rigid bodies are then integrated in a semi-implicit fashion.

4.2 Soft Body Dynamics

The deformation of a soft object is a time dependent map from its undeformed material coordinates \mathbf{X} to the world coordinates \mathbf{x} . Following the commonly adopted Finite Element Method (FEM), we represent a soft body as a low-resolution volumetric tetrahedral mesh with a linear shape function. The dynamics of the soft body is then encoded by the tetrahedral elements of the mesh:

$$\mathbf{M} \ddot{\mathbf{x}} = \mathbf{f}_e + \mathbf{f}_d + \mathbf{f}_c + \mathbf{f}_g \quad (6)$$

where $\mathbf{M} = \text{diag}\{m_1, \dots, m_n\}$ is the mass matrix and m_i is the lumped mass of node i . \mathbf{f}_c and \mathbf{f}_g are the contact forces and the gravity respectively. The elastic force \mathbf{f}_e and the damping force \mathbf{f}_d are computed from the constitutive model of the material. In this paper, we devise a pose-based constitutive model that extends the widely adopted corotated linear model, which will be explained in details shortly. We again use the semi-implicit Euler method, also called symplectic Euler method, to integrate the equations of motion for the soft body.

We use a high-resolution triangular surface mesh of the soft body to achieve better visualization quality and better collision detection accuracy. We employ the mesh embedding technique of [Kim and Pollard 2011b] to couple the low-res volumetric mesh and the high-res surface mesh. More specifically, the position of a vertex on the surface geometry is computed by interpolating the nodal positions of the tetrahedral elements that enclose the vertex: $\tilde{\mathbf{x}}_i = \sum_{j=1}^4 \phi_j \mathbf{x}_j$, where $\sum_{j=1}^4 \phi_j = 1$ is the barycentric coordinates of the vertex. We assemble all the vertices and nodes together as $\tilde{\mathbf{x}} = \mathbf{J}_\phi \mathbf{x}$.

4.2.1 Corotated Linear Elasticity

The strain, stress and elastic force at node \mathbf{X} in material coordinates depend only on the deformation gradient $\mathbf{F} = \partial \mathbf{x} / \partial \mathbf{X}$. Their relationships are specified by the constitutive model of the given material. In this paper, we utilize the corotated linear elasticity [Müller et al. 2002; Müller and Gross 2004; Sifakis et al. 2012] for its simplicity and rotational invariance. It extracts the rotational part of the deformation by polar decomposition $\mathbf{F} = \mathbf{R}\mathbf{S}$, where \mathbf{R} is the rotation matrix corresponding to the rigid rotation, and \mathbf{S} is the symmetric matrix representing the soft stretch. Then the strain tensor is defined as follows to eliminate rigid rotation from the deformation:

$$\boldsymbol{\epsilon} = \mathbf{R}^T \mathbf{F} - \mathbf{I} \quad (7)$$

and the first Piola-Kirchhoff stress tensor is defined as

$$\begin{aligned} \mathbf{P} &= \mathbf{R}[2\mu\boldsymbol{\epsilon} + \lambda \text{tr}(\boldsymbol{\epsilon})\mathbf{I}] \\ &= 2\mu(\mathbf{F} - \mathbf{R}) + \lambda \text{tr}(\mathbf{R}^T \mathbf{F} - \mathbf{I})\mathbf{R} \end{aligned} \quad (8)$$

Each tetrahedral element has a constant deformation gradient that depends only on its nodal positions:

$$\mathbf{F} = \mathbf{B}_s \mathbf{B}_m^{-1} \quad (9)$$

The deformed shape matrix \mathbf{B}_s and the reference shape matrix \mathbf{B}_m are calculated from the nodal positions in world space and material space respectively:

$$\mathbf{B}_s = [\mathbf{x}_1 - \mathbf{x}_0, \mathbf{x}_2 - \mathbf{x}_0, \mathbf{x}_3 - \mathbf{x}_0] \quad (10)$$

$$\mathbf{B}_m = [\mathbf{X}_1 - \mathbf{X}_0, \mathbf{X}_2 - \mathbf{X}_0, \mathbf{X}_3 - \mathbf{X}_0] \quad (11)$$

where \mathbf{x}_i and \mathbf{X}_i , $i = 0 \dots 3$, are the four nodes of a tetrahedral element. From the deformation gradient, the total elastic force \mathbf{f}_e for one node can then be calculated. We refer interested readers to Section 4.2 of [Sifakis et al. 2012] for more details.

4.2.2 Pose-based Plasticity

Human-like characters can undergo large deformations near skeletal joints, such as shoulders and elbows, which are ineffective to model with elasticity. Musculoskeletal models, such as the biomechanical models in [Lee et al. 2009], are needed to model the joint deformations accurately. Such approaches, however, are computationally costly and hard to use for motion control. We propose a novel pose-based plasticity model to alleviate large elastic energies caused by joint rotations. This model is inspired by literatures that update reference shapes when simulating plastic materials [Bargteil et al. 2007; Wicke et al. 2010]. Different from the prior arts, however, we wish for pose-dependent plasticity rather than history dependent plasticity.

Our pose-based plasticity model updates the reference nodal position $\tilde{\mathbf{x}}_i$ of the volumetric mesh from the undeformed material coordinates \mathbf{X} as follows:

$$\tilde{\mathbf{x}}_i = \Gamma_{x_i}(\mathbf{q}, \mathbf{X}_i) \quad (12)$$

\mathbf{q} represents the bone positions. We realize the plasticity model through Linear Blend Skinning (LBS):

$$\Gamma_{x_i}(\mathbf{q}, \mathbf{X}_i) = \sum_{j=1}^n w_{ij} \mathbf{T}_j(\mathbf{q}_j) \mathbf{X}_i \quad (13)$$

$\mathbf{T}_j, j = 1, \dots, n$ are the transformation matrices of each bone. The skinning weights w_{ij} indicate how much the transformation of a bone j influences the node i . We solve for w_{ij} using the bounded

biharmonic method of [Jacobson et al. 2011]. All the fixed nodes that are rigidly attached to the skeleton are treated as control points for the optimization algorithm. We refer interested readers to [Jacobson et al. 2011] for more details.

We use additive plasticity as [O’Brien et al. 2002]:

$$\epsilon = \epsilon_e + \epsilon_p \quad (14)$$

where the total strain with respect to the original reference shape in Equation 7 is decomposed into the elastic strain ϵ_e and the plastic strain $\epsilon_p = \mathbf{R}_p^T \mathbf{F}_p - \mathbf{I}$. Thus the elastic strain becomes:

$$\epsilon_e = \mathbf{R}^T \mathbf{F} - \mathbf{R}_p^T \mathbf{F}_p \quad (15)$$

The plastic deformation gradient $\mathbf{F}_p = \mathbf{B}_p \mathbf{B}_m^{-1}$ is computed from the pose-based shape matrix $\mathbf{B}_p = [\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0, \bar{\mathbf{x}}_2 - \bar{\mathbf{x}}_0, \bar{\mathbf{x}}_3 - \bar{\mathbf{x}}_0]$. Then the first Piola-Kirchhoff stress tensor of Equation 8 becomes

$$\begin{aligned} \mathbf{P} &= \mathbf{R}[2\mu\epsilon_e + \lambda \text{tr}(\epsilon_e)\mathbf{I}] \\ &= 2\mu(\mathbf{F} - \mathbf{R}\mathbf{R}_p^T \mathbf{F}_p) + \lambda \text{tr}(\mathbf{R}^T \mathbf{F} - \mathbf{R}_p^T \mathbf{F}_p)\mathbf{R} \end{aligned} \quad (16)$$

To conserve angular momentum, $\mathbf{P}\mathbf{F}^T$ needs to be symmetric, which is equivalent to requiring $\mathbf{R}\mathbf{R}_p^T \mathbf{F}_p \mathbf{F}^T$ to be symmetric. We can then derive the solution for the rotation matrix $\mathbf{R}_p = \mathbf{R}^* \mathbf{R}$, where \mathbf{R}^* is from the polar decomposition $\mathbf{F}_p \mathbf{F}^T = \mathbf{R}^* \mathbf{S}^*$.

4.2.3 Pose-based Damping

The damping term \mathbf{f}_d is to dissipate high-frequency vibrations in soft bodies, and should not damp the bone motions or violate the conservation of momentum. We connect each node \mathbf{x}_i to its corresponding position in the updated reference pose $\bar{\mathbf{x}}_i$ with a damper:

$$\mathbf{f}_{d_i} = c m_i \frac{\delta \dot{\mathbf{x}}_i^T \delta \mathbf{x}_i}{\delta \mathbf{x}_i^T \delta \mathbf{x}_i} \delta \mathbf{x}_i \quad (17)$$

where $\delta \mathbf{x}_i = \bar{\mathbf{x}}_i - \mathbf{x}_i$ is the position difference, and $\delta \dot{\mathbf{x}}_i = \dot{\bar{\mathbf{x}}}_i - \dot{\mathbf{x}}_i$ is the velocity difference. The damping force is projected along the damper to eliminate undesired torques. We compute $\dot{\bar{\mathbf{x}}}_i$ by differentiating Equation 12, i.e., $\dot{\bar{\mathbf{x}}}_i = \mathbf{J}_{\Gamma_{\mathbf{x}_i}} \dot{\mathbf{q}}$, where $\mathbf{J}_{\Gamma_{\mathbf{x}_i}}$ is the skinning Jacobian.

Our damping method conserves momentum; does not require computation of stiffness matrices and thus is efficient; and is more consistent with our pose-based plasticity model. Other damping methods, such as Rayleigh damping, however, can be integrated into our simulation framework without any problem.

4.2.4 Collisions and Contacts

We detect collisions using the fine surface mesh rather than the coarse volumetric mesh for better accuracy. For speed and simplicity, we use a point-based collision detector in this paper and ignore self-collisions in most of our experiments. However, our framework also supports self-collision if interactivity is not essential to the application. In one of our demos in the accompanying video, we applied the image-based collision detection method of Wang et al. [2012] for self-collision detection, where the collision detection becomes the bottleneck of the whole simulation pipeline.

We model collisions and contacts using penalty forces with Coulomb friction, similar to [Kim and Pollard 2011b]. The contact forces \mathbf{f}_c are then transferred onto the coarse volumetric mesh by $\mathbf{f}_c = \mathbf{J}_\phi^T \hat{\mathbf{f}}_c$, where \mathbf{J}_ϕ is the mesh embedding Jacobian described in Section 4.2.

4.3 Two-way Coupling

To support two-way coupling between the soft body and the rigid skeleton, we attach the soft body to the skeleton by directly fixing some of the coarse mesh nodes to the bones. That is, the position and velocity of these fixed nodes are updated at the beginning of the time step according to the dynamic states of the bones. The masses of the fixed nodes are merged into the bones, or equivalently, the inertia of the bones are calculated from these attached nodes. We refer interested readers to [Kim and Pollard 2011b] for more details.

The elastic forces and contact forces on the soft body are transferred to the bones through the fixed nodes, and therefore affect the dynamic states of the bones in the next time step. In addition, the damping forces \mathbf{f}_{d_i} applied on the nodes \mathbf{x}_i computed in Equation 17 should be applied to $\bar{\mathbf{x}}_i$ in the opposite direction. However, $\bar{\mathbf{x}}_i$ is a virtual node that does not actually exist in the simulation. We therefore apply the forces to the corresponding bones with proper coordinate transformation. More specifically, the corresponding forces and torques applied to the bones due to the damping forces are computed by the skinning Jacobian as

$$\boldsymbol{\tau}_d = - \sum_i \mathbf{J}_{\Gamma_{\mathbf{x}_i}}^T \mathbf{f}_{d_i} \quad (18)$$

4.4 Simulation Pipeline

Our full simulation scheme consists of three major steps:

- (1) Updating kinematics: At the beginning of each time step, we update (a) the position and velocity of the fixed nodes according to the dynamic states of the bones; (b) the fine surface mesh according to the mesh embedding equation; (c) the reference pose and velocity for the pose-based plasticity and damping.
- (2) Accumulating forces and torques: Then we accumulate (a) the actuation PD torques for each joint, using the controller that will be described in the next section; (b) the elastic forces and the damping forces from the pose-based constitutive model for the soft body nodes; (c) the corresponding elastic forces on the fixed nodes and damping forces on the bones.
- (3) Integration: We solve and advance the skeleton dynamics system and the soft body dynamics system using semi-implicit Euler method as described above in Section 4.1 and Section 4.2.

Differences from Previous Approaches : Our two-way coupled skeleton and soft body simulation system employs many components from previous work. However, our framework is unique in several ways. First, most previous work supports only one-way coupling between the flesh and the skeleton [Capell et al. 2002; Lee et al. 2009]. That is, the skeleton dynamics affects the soft body but not the other way around. The work of [Kim and Pollard 2011b] does support two-way coupling, but not demonstrated on human-like characters. Their Fatman example is a one-way kinematically-driven simulation. Human-like characters pose more challenges for two-way coupling because (a) their range of motion is much larger, and (b) their control is much more difficult. We achieve two-way coupling using a novel pose-based plasticity model that extends the corotated linear elasticity model. Second, our two-way coupling method conserves momentum of the whole system. Yet previous methods, such as [Müller and Gross 2004; Kim and Pollard 2011b], can violate the conservation of momentum. Furthermore, most previous approaches for soft body character simulation, such as [Kim and Pollard 2011b], implement their own rigid body dynamics solver in generalized coordinates. We use a third-party simulator Open Dynamics Engine (ODE) to simulate the skeletal

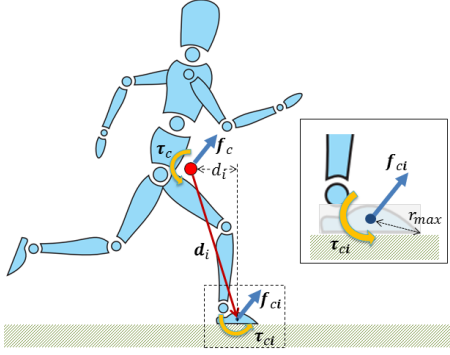


Figure 3: Variables in our inverse dynamics calculation.

system in Cartesian coordinates. We only augment it with a stable PD controller for better simulation stability. Utilizing a third-party simulator significantly reduces the difficulty and amount of effort required for reimplementing.

5 Motion Control

We build control for soft body characters in two steps, following the recent advances on rigid body character control. In the first step we search for open-loop controls that can imitate the style of an input reference trajectory. In the second step we learn feedback policies so that the controls are more robust to perturbations. For this step we directly follow [Liu et al. 2012] to build linear feedback policies around the open-loop controls constructed in the first step.

For the first step where we search for open-loop controls, we use a sampling-based scheme similar to [Liu et al. 2010]. Their method was designed for rigid body characters, however, and its direct application to soft body character control failed due to (a) the large mismatches between the soft body characters and the motion capture subjects; (b) the add-on complexity of the soft body dynamics; (c) motions with long airborne phases, such as jump kicks and Kong Fu, were not investigated in [Liu et al. 2010] and cause the control construction to fail. We thus augment the method with a trajectory-based inverse dynamics procedure to compute better initial solutions for the search of open-loop controls. We also augment the optimization with a time scaling scheme for more flexible control of timing in highly dynamic motions.

5.1 Inverse Dynamics

The method of [Liu et al. 2010] samples around the input reference trajectory and outputs perturbed target poses as the open-loop controls. Although effective for rigid body characters, the sampling procedure is not as effective and efficient for soft body characters. We improve the basic scheme of using the input reference trajectory as the initial solution for optimization with an inverse dynamics procedure, in order to better estimate an initial solution for sampling. We denote the Inverse Dynamics (ID) procedure as:

$$\tau = \text{ID}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{f}_{ext}, \tau_{ext}) \quad (19)$$

where $(\mathbf{q}, \dot{\mathbf{q}})$ represent the state of the skeletal system, and \mathbf{f}_{ext} and τ_{ext} are the external forces and torques applied to the bones. The τ on the left-hand side is the internal joint torques needed to actuate the system to achieve the desired acceleration $\ddot{\mathbf{q}}$. The inertia of the bones for ID are computed from all the nodes of the coarse mesh according to their skinning weights.

We estimate the kinematic quantities $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$ from the input reference motion. The reference trajectory comprises a series of time-

indexed poses denoted as $\tilde{\mathbf{q}}_T$ with a time step ΔT between successive keyframes. We estimate its velocities and accelerations by forward finite differencing:

$$\dot{\tilde{\mathbf{q}}}_T = \frac{\tilde{\mathbf{q}}_{T+1} - \tilde{\mathbf{q}}_T}{\Delta T} \quad (20)$$

$$\ddot{\tilde{\mathbf{q}}}_T = \frac{\dot{\tilde{\mathbf{q}}}_{T+1} - \dot{\tilde{\mathbf{q}}}_T}{\Delta T} \quad (21)$$

The time step ΔT is usually much larger than the simulation time step Δt . To estimate $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$ in the resolution of Δt , we assume constant accelerations and a linear variation in the velocities and positions within one time step ΔT , using an interpolation coefficient $\alpha = (t\Delta t - T\Delta T)/\Delta T$ as follows:

$$\ddot{\mathbf{q}}_t = \ddot{\tilde{\mathbf{q}}}_T \quad (22)$$

$$\dot{\mathbf{q}}_t = (1 - \alpha)\dot{\tilde{\mathbf{q}}}_T + \alpha\dot{\tilde{\mathbf{q}}}_{T+1} \quad (23)$$

$$\mathbf{q}_t = (1 - \alpha)\tilde{\mathbf{q}}_T + \alpha\tilde{\mathbf{q}}_{T+1} \quad (24)$$

The estimation of the external forces \mathbf{f}_{ext} and torques τ_{ext} is much more involved. We assume the reference trajectories were captured with no external forces other than the gravity and ground contacts. Then the resultant ground contact force and torque with respect to the center of mass can be computed as follows:

$$\mathbf{f}_c = \frac{\mathbf{P}_{t+1} - \mathbf{P}_t}{\Delta t} - m\mathbf{g} \quad (25)$$

$$\tau_c = \frac{\mathbf{H}_{t+1} - \mathbf{H}_t}{\Delta t} \quad (26)$$

where m is the total mass of the character, and \mathbf{P} and \mathbf{H} are the linear and angular momentum around the center of mass respectively. To estimate the contact forces and torques applied to individual bones that are in contact with the ground, we distribute the resultant \mathbf{f}_c and τ_c to contacting bones according to their relative positions with respect to the center of mass. More specifically, denote the set of contacting bones as \mathcal{B} , the contact force and torque acting on bone $i \in \mathcal{B}$ are estimated as:

$$\mathbf{f}_{ci} = \frac{1/d_i}{\sum_{i \in \mathcal{B}} 1/d_i} \mathbf{f}_c \quad (27)$$

$$\tau_{ci} = \frac{1/d_i}{\sum_{i \in \mathcal{B}} 1/d_i} \tau_c - \mathbf{d}_i \times \mathbf{f}_{ci} \quad (28)$$

where \mathbf{d}_i is the vector from the center of mass of the character to the center of mass of bone i (Figure 3), and d_i is its projected length on the ground plane.

The above estimations can violate physical constraints because of noise in the input trajectory and model discrepancies between the motion capture subject and our soft body characters. We therefore postprocess the estimated forces and torques using the following constraints:

$$f_y \geq 0 \quad (29)$$

$$\|\mathbf{f}_{xz}\| \leq \mu f_y \quad (30)$$

$$\|\tau\| \leq r_{max} f_{max} \quad (31)$$

The first constraint specifies that the ground reaction force can only push but not pull the character. The second constraint bounds the horizontal force by the normal force and the coefficient of friction μ . The third constraint clamps the estimated ground reaction torque with the maximal force and lever arm. r_{max} is the maximum distance between the center of bone i and the lowest point of its bounding box (see Figure 3), and $f_{max} = \sqrt{1 + \mu^2} f_y$ is the maximal contact force possibly applied on a single contact point.

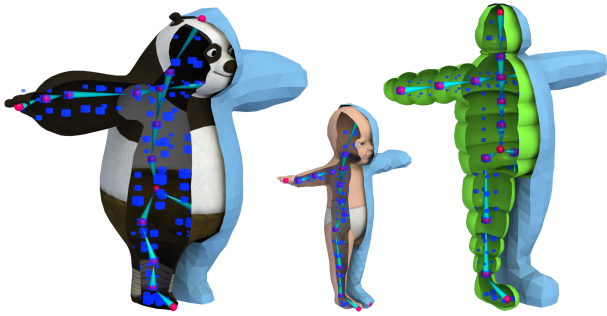


Figure 4: Our soft body characters: Panda, Baby, and Michelin. Light blue represents the coarse volumetric mesh. Dark blue boxes represent the fixed nodes. Red spheres are joints.

Model	Tetrahedral Mesh		Surface Mesh		#Joints
	# vertices	# elements	# vertices	# faces	
Michelin	1182	4160	5263	10522	15
Panda	1220	4099	4278	8552	17
Baby	1156	3690	8194	16384	17

Table 1: Various parameters of our soft character models.

The above estimated ground contact forces and torques are accumulated for each bone contacting the ground. Gravity are also added to all the bones. Finally, we perform inverse dynamics of Equation 19 to compute the joint torques τ for each simulation time step. Because [Liu et al. 2010; Liu et al. 2012] use target poses for PD servos as their control representation, we further transform the joint torques to target PD angles by rearranging Equation 1 as:

$$\mathbf{q} = \tilde{\mathbf{q}} + \frac{1}{k_p}(\boldsymbol{\tau} + k_d \dot{\tilde{\mathbf{q}}}) \quad (32)$$

5.2 Time Scaling

The original algorithm of [Liu et al. 2010] outputs controls that imitate the reference trajectory in both space and time. The reference timing, however, should not be strictly reinforced in our scenarios. For example, a fat character or a weak baby usually move slower than an average adult motion capture subject. We therefore augment the sampling algorithm with an additional degree of freedom in time. More specifically, we sample the PD target pose (in space) and the duration to reach the target pose (in time) for every sample. This is essential to reproduce highly dynamic motions on soft body characters, especially motions with long airborne phases, and similar to the time scaling parameter used in [Liu et al. 2012] for motion parameterization.

6 Results

We have tested our simulation and control framework with three soft body characters as shown in Figure 4. The fine surface meshes and the skeletons are given as input. We first manually generate a low resolution surface mesh to enclose the fine surface mesh. Then we specify fixed nodes on the bones (the blue boxes in Figure 4). Next we generate the tetrahedral volumetric mesh from the low-res surface mesh and the fixed nodes using the 3D Delaunay triangulator TETGEN [Si 2011]. The final tetrahedral meshes have roughly 1.2K vertices and 4K tetrahedral elements as shown in Table 1.

We have implemented the simulation systems on a desktop with 8-core 2.83GHz Intel Xeon E5440. After parallelizing the computation of elastic forces, the simulation speed is about four times

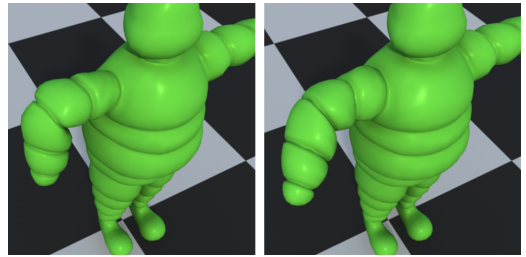


Figure 5: Comparison: the pose-based plasticity is turned on (left) and off (right).

Skills	Open-loop Control (hours)	Feedback Policy (hours)	Runtime (CPU) (ms/frame)	Runtime (GPU) (ms/frame)
Michelin run	0.5	1.0	2.1	1.2
Panda Run	0.8	3.4	2.3	1.4
Baby Roll	1.0		3.2	2.9

Table 2: Performance statistics of our parallel implementation using a simulation time step of 0.5ms.

slower than realtime as shown in Table 2, using a simulation time step of 0.5ms. We have also implemented the pose-based constitutive model on GPUs. It is slightly faster than the CPU version on an NVIDIA Graphics card GeForce GTX 680.

We demonstrate the effectiveness of our pose-based plasticity model in Figure 5. Without the plasticity model, the character fails to bend its elbows sufficiently. In most of our experiments we ignore self-collisions for faster performance. However, our framework easily supports self-collision when a proper collision detection engine is available, such as the imaged-based method of [Wang et al. 2012]. In the accompanying video, we show that with self-collisions turned on, the intersections of the arms with the body are effectively eliminated during running.

We compare our damping method with Rayleigh damping and the method of [Kim and Pollard 2011b]. We use the Michelin model to track multiple cycles (1.6s/cycle) of the kicking motion in a gravity-free and contact-free environment. Since there are no external forces and torques, the ground truth angular momentum should remain constant. Figure 7 shows the magnitude of the angular momentum over time using different damping schemes. We can see that the method of [Kim and Pollard 2011b] dramatically decreases the angular momentum and fluctuates a lot. Rayleigh Damping also results in momentum loss. In contrast, our method is stable and well conserves the angular momentum of the system. Such feature will be essential in integrating momentum-based motion control strategies into our framework in the future [Macchietto et al. 2009].

Open-loop Control Construction We have tested our control construction with a set of motions captured from human subjects, including running, kicking, rolling, and Kung Fu. The control construction is computationally expensive when simulating both the rigid bodies and the soft bodies. Detailed performance statistics on a small cluster of 40 cores can be found in Table 2. Screenshots for Panda Run, Michelin Kung Fu, Baby Roll, and Michelin Run are shown in Figures 1 and 6.

In the accompanying video, we show comparisons to justify the inverse dynamics component and the time scaling scheme. Without the initial solution generated by ID, the construction algorithm cannot find proper PD targets that can lift Michelin’s leg quickly enough before it starts to fall. Time scaling is also essential for successful control of highly dynamic motions. For example, the character cannot perform multiple jump-kicks without proper time scaling due to improper takeoff and landing time. Sometimes the

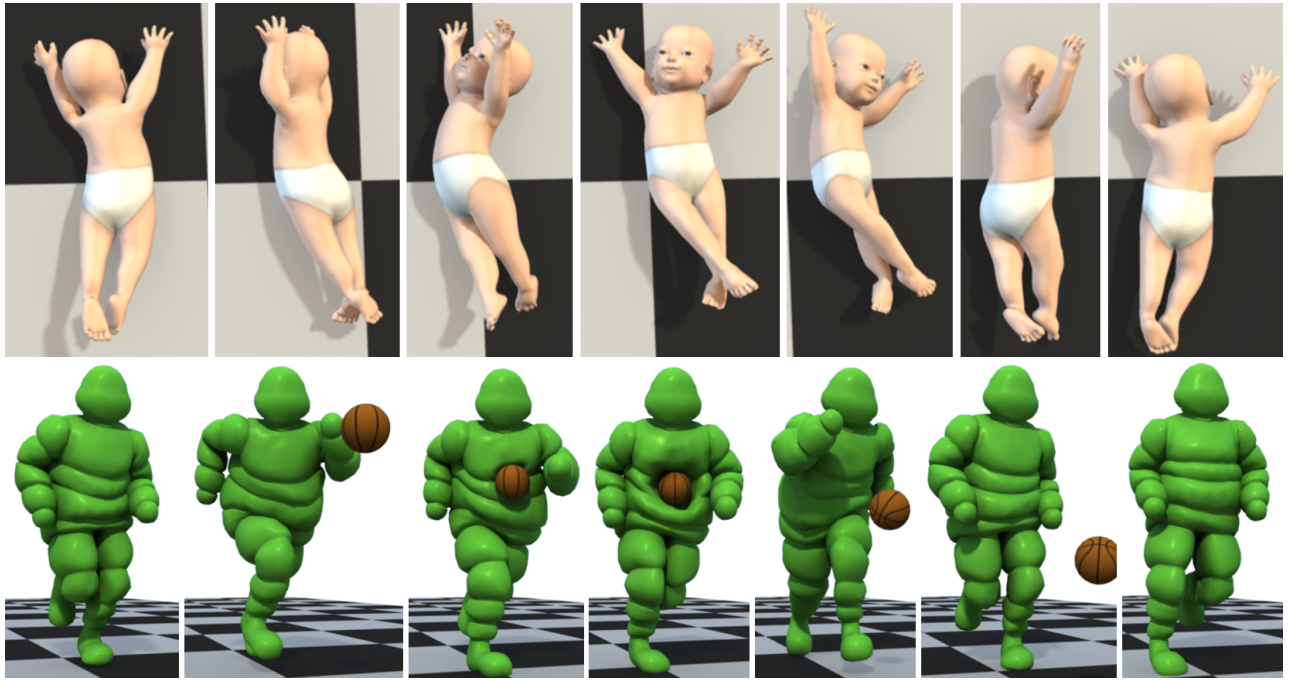


Figure 6: Top row: Baby roll (from right to left); Bottom row: Michelin run.

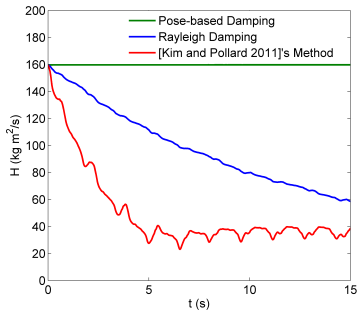


Figure 7: Comparison of damping methods.

effective time scaling can be as simple as slowing down the reference motions, for example for the Panda run and Baby roll. The intuition is that Panda moves slower and Baby is weaker.

Closed-loop Feedback Policies We have learned feedback policies for the Michelin and Panda runs. We optimize for the feedback policies on a cluster of 100 cores. It took about one hour to find a robust feedback policy for Michelin to run in a stable fashion. The feedback policy for Panda took much longer to learn due to its large model differences from humans, such as its big belly and short legs. The feedback policies are robust to some extent to external perturbations. We only optimized the feedback policies using hard material properties, but accidentally found them robust enough for a large range of flesh stiffness.

7 Discussion

We have presented a two-way simulation and control framework for soft characters with inherent skeletons. Our novel pose-based plasticity model supports proper deformation around joints with large ranges of motion. In extending a sampling-based control algorithm originally developed for rigid characters, we have devised two key components, inverse dynamics and time scaling, to lead the search

algorithm to success, despite large model and motion discrepancies between the soft characters and the captured subjects.

Where possible, we choose simple design options so that our system is fast and relatively easy to reimplement. The disadvantages of such choices include that the whole system requires tiny simulation time steps, well-designed volumetric meshes, and moderate material stiffness to remain numerically stable. We integrate the soft body dynamics, and couple the soft body and the rigid skeleton in an explicit fashion, because these schemes are simple and computationally efficient. Replacing them with implicit integrators is a worthy topic for future research. Additionally, our simple penalty method for contact modeling can result in foot sliding for highly dynamic motions such as jump kicks. In the future, we wish to use better contact force solvers. The performance of our system is interactive but not realtime. Better CPU and GPU implementation and integration is needed to achieve a faster system.

Currently we manually design the low-res volumetric mesh and place the fixed nodes. When the elements near joints are severely deformed or even inverted, unnatural wrinkles and rips may appear on the high-res surface mesh. A potential fix is to dynamically remesh severely deformed areas [Wicke et al. 2010]. The skinning algorithm affects the quality of the pose-based plasticity and therefore the simulated deformations. We currently use LBS for its simplicity. Better skinning methods, such as Dual Quaternion Skinning [Kavan et al. 2007], elasticity-inspired methods [Kavan and Sorkine 2012], or physically based methods [McAdams et al. 2011], may be investigated in the future. Besides skinning, the rest shapes of the soft body can also be specified by other means, such as interpolating example shapes [Coros et al. 2012].

The success rate of our control construction is not 100%. One failure case is that we cannot reproduce the rolling on Michelin with soft materials. This is because the reference motion is captured from a human athlete whose contact dynamics is significantly different from the soft Michelin. It would be interesting to investigate how to incorporate material properties and planning into the sampling-based control scheme.

Acknowledgements: This work was partially supported by Singapore MoE Academic Research Fund Tier 2 (MOE2011-T2-2-152).

References

- BARAFF, D., AND WITKIN, A. 1998. Large steps in cloth simulation. In *SIGGRAPH 1998*, 43–54.
- BARBIČ, J., AND POPOVIĆ, J. 2008. Real-time control of physically based simulations using gentle forces. *ACM Trans. Graph.* 27, 5, Article 163.
- BARGTEIL, A. W., WOJTAN, C., HODGINS, J. K., AND TURK, G. 2007. A finite element method for animating large viscoplastic flow. *ACM Trans. Graph.* 26, 3, Article 16.
- BROWN, D. F., MACCHIETTO, A., YIN, K., AND ZORDAN, V. 2013. Control of rotational dynamics for ground behaviors. In *SCA'13*, 55–61.
- CAPELL, S., GREEN, S., CURLESS, B., DUCHAMP, T., AND POPOVIĆ, Z. 2002. Interactive skeleton-driven dynamic deformations. *ACM Trans. Graph.* 21, 3, 586–593.
- CAPELL, S., BURKHART, M., CURLESS, B., DUCHAMP, T., AND POPOVIĆ, Z. 2005. Physically based rigging for deformable characters. In *SCA'05*, 301–310.
- COROS, S., BEAUDOIN, P., AND VAN DE PANNE, M. 2010. Generalized biped walking control. *ACM Trans. Graph.* 29, 4, Article 130.
- COROS, S., MARTIN, S., THOMASZEWSKI, B., SCHUMACHER, C., SUMNER, R., AND GROSS, M. 2012. Deformable objects alive! *ACM Trans. Graph.* 31, 4, Article 69.
- DA SILVA, M., ABE, Y., AND POPOVIĆ, J. 2008. Interactive simulation of stylized human locomotion. *ACM Trans. Graph.* 27, 3, Article 82.
- DE LASA, M., MORDATCH, I., AND HERTZMANN, A. 2010. Feature-based locomotion controllers. *ACM Trans. Graph.* 29, 4, Article 131.
- GALOPPO, N., OTADUY, M. A., TEKIN, S., GROSS, M., AND LIN, M. C. 2007. Soft articulated characters with fast contact handling. *Computer Graphics Forum* 26, 3, 243–253.
- HA, S., YE, Y., AND LIU, C. K. 2012. Falling and landing motion control for character animation. *ACM Trans. Graph.* 31, 6, Article 155.
- HAHN, F., MARTIN, S., THOMASZEWSKI, B., SUMNER, R., COROS, S., AND GROSS, M. 2012. Rig-space physics. *ACM Trans. Graph.* 31, 4, Article 72.
- JACOBSON, A., BARAN, I., POPOVIĆ, J., AND SORKINE, O. 2011. Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph.* 30, 4, Article 78.
- JAIN, S., AND LIU, C. K. 2011. Controlling physics-based characters using soft contacts. *ACM Trans. Graph.* 30, 6, Article 163.
- KAVAN, L., AND SORKINE, O. 2012. Elasticity-inspired deformers for character articulation. *ACM Trans. Graph.* 31, 6, Article 196.
- KAVAN, L., COLLINS, S., ŽÁRA, J., AND O'SULLIVAN, C. 2007. Skinning with dual quaternions. In *I3D '07*, 39–46.
- KIM, J., AND POLLARD, N. S. 2011. Direct control of simulated nonhuman characters. *IEEE Comput. Graph. Appl.* 31, 4, 56–65.
- KIM, J., AND POLLARD, N. S. 2011. Fast simulation of skeleton-driven deformable body characters. *ACM Trans. Graph.* 30, 5, Article 121.
- LEE, S.-H., SIFAKIS, E., AND TERZOPOULOS, D. 2009. Comprehensive biomechanical modeling and simulation of the upper body. *ACM Trans. Graph.* 28, 4, Article 99.
- LEE, Y., KIM, S., AND LEE, J. 2010. Data-driven biped control. *ACM Trans. Graph.* 29, 4, Article 129.
- LIU, L., YIN, K., VAN DE PANNE, M., SHAO, T., AND XU, W. 2010. Sampling-based contact-rich motion control. *ACM Trans. Graph.* 29, 4, Article 128.
- LIU, L., YIN, K., VAN DE PANNE, M., AND GUO, B. 2012. Terrain runner: control, parameterization, composition, and planning for highly dynamic motions. *ACM Trans. Graph.* 31, 6, Article 154.
- MACCHIETTO, A., ZORDAN, V., AND SHELTON, C. R. 2009. Momentum control for balance. *ACM Trans. Graph.* 28, 3, Article 80.
- MCADAMS, A., ZHU, Y., SELLE, A., EMPEY, M., TAMSTORE, R., TERAN, J., AND SIFAKIS, E. 2011. Efficient elasticity for character skinning with contact and collisions. *ACM Trans. Graph.* 30, 4, Article 37.
- MÜLLER, M., AND GROSS, M. 2004. Interactive virtual materials. In *GI'04*, 239–246.
- MÜLLER, M., DORSEY, J., MCMILLAN, L., JAGNOW, R., AND CUTLER, B. 2002. Stable real-time deformations. In *SCA'02*, 49–54.
- O'BRIEN, J. F., BARGTEIL, A. W., AND HODGINS, J. K. 2002. Graphical modeling and animation of ductile fracture. *ACM Trans. Graph.* 21, 3, 291–294.
- SHINAR, T., SCHROEDER, C., AND FEDKIW, R. 2008. Two-way coupling of rigid and deformable bodies. In *SCA'08*, 95–103.
- SI, H. 2011. TetGen, A Quality Tetrahedral Mesh Generator and a 3D Delaunay Triangulator. <http://wias-berlin.de/software/tetgen/>.
- SIFAKIS, E., SHINAR, T., IRVING, G., AND FEDKIW, R. 2012. FEM simulation of 3D deformable solids: A practitioners guide to theory, discretization and model reduction. In *SIGGRAPH 2012 Course*.
- TAN, J., LIU, C. K., AND TURK, G. 2011. Stable proportional-derivative controllers. *IEEE Comput. Graph. Appl.* 31, 4, 34–44.
- TAN, J., TURK, G., AND LIU, C. K. 2012. Soft body locomotion. *ACM Trans. Graph.* 31, 4, Article 26.
- TERZOPOULOS, D., PLATT, J., BARR, A., AND FLEISCHER, K. 1987. Elastically deformable models. In *SIGGRAPH*, vol. 21, 205–214.
- WANG, B., FAURE, F., AND PAI, D. K. 2012. Adaptive image-based intersection volume. *ACM Trans. Graph.* 31, 4, Article 97.
- WICKE, M., RITCHIE, D., KLINGNER, B. M., BURKE, S., SHEWCHUK, J. R., AND O'BRIEN, J. F. 2010. Dynamic local remeshing for elastoplastic simulation. *ACM Trans. Graph.* 29, Article 49.
- YIN, K., LOKEN, K., AND VAN DE PANNE, M. 2007. SIMBICON: Simple biped locomotion control. *ACM Trans. Graph.* 26, 3, Article 105.