Animating Articulated Characters using Wiggly Splines

Christian Schulz
MPI Informatik

Christoph von Tycowicz
Zuse Institute Berlin

Hans-Peter Seidel
MPI Informatik

Klaus Hildebrandt
Delft University of Technology

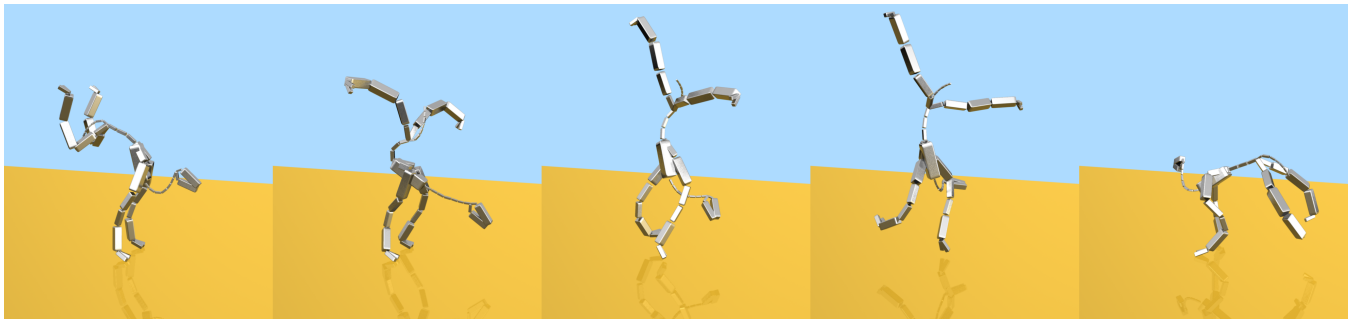

Figure 1: *Our framework for spacetime optimization efficiently generates planned motion of complex articulated characters. Snapshots of an animation of a four-legged character performing a handstand followed by a twisting jump are shown.*

Abstract

We propose a new framework for spacetime optimization that can generate artistic motion with a long planning horizon for complex virtual characters. The scheme can be used for generating general types of motion and neither requires motion capture data nor an initial motion that satisfies the constraints. Our modeling of the spacetime optimization combines linearized dynamics and a novel warping scheme for articulated characters. We show that the optimal motions can be described using a combination of vibration modes, wiggly splines, and our warping scheme. This enables us to restrict the optimization to low-dimensional spaces of explicitly parametrized motions. Thereby the computation of an optimal motion is reduced to a low-dimensional non-linear least squares problem, which can be solved with standard solvers. We show examples of motions created by specifying only a few constraints for positions and velocities.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation;

Keywords: spacetime constraints, physically-based animation, optimal control, wiggly splines

1 Introduction

Creating physically plausible motions of virtual characters that follow an animator's intent is a central task in computer animation. Spacetime optimization proved to be a valuable tool for planning, processing and synthesizing motion; and for controlling physical simulations. It is used for various tasks including creating realistic motion from keyframes, editing motions or simulations, and synthesizing specific motions (e.g., human walking) from captured motion. The forces needed to generate the resulting motions are optimally distributed over the whole animation or a selected time window. The motions are planned, efficient, and show effects like squash-and-stretch, timing, and anticipation. The price to pay is that the optimization problem to be solved is high-dimensional and highly non-linear.

We introduce a novel framework for spacetime optimization of the motion of articulated characters that can generate *various* types of motion (e.g. artistic motion) for *complex* virtual characters and does not rely on motion capture data. The long planning horizon increases the realism of the motions and automatically produces effects like anticipation (see the animation of the four-legged character in the teaser and video for an example). The scheme combines a linearization of the dynamics and a novel warping scheme for articulated characters in order to reduce the complexity of the optimization problem to be solved. Our modeling of the spacetime optimization provides us with a structure that can be used to create low-dimensional spaces of explicitly parametrized motions to which the optimization is restricted. Since the motions are explicitly parametrized, a time discretization is not required and the derivatives of the trajectories describing a motion can be explicitly evaluated. Our framework enables us to create motion by specifying only a few partial keyframes for positions and velocities. The resulting low-dimensional non-linear least squares problem can be solved with standard solvers like the Gauss–Newton method.

The framework we present is based on three main technical contributions. The first is a warping scheme for linearized articulated characters. The problem is that a linearization of the configuration space of an articulated character leads to artifacts for larger displacements, *e.g.*, the linearization of the rotations of the bones causes non-isometric deformations of the bones. The goal of the warping scheme is to reduce these artifacts. Our warping scheme uses the Lie-group structure of the rotation group and proceeds in two steps. First, we use the exponential map to compute a target rotation for the linearized rotation of every bone. Then, we construct a configuration that best matches the target rotations. The warping process is designed so that it can be differentiated and efficiently evaluated, which is important for the optimization. The warp map effectively reduces the linearization artifacts (see the warped vibration modes shown in the supplementary video). In addition to the application of the warping to spacetime optimization, we are convinced that the combination of vibration modes and warping can be useful for other problems concerning the simulation and anima-

tion of articulated characters. This approach contrasts from previous work on vibration modes for characters. Kry et al. [2009] and Jain and Liu [2011] interpreted the linear modes as vibrations in the generalized coordinates used to parametrize the configuration space. Similar to our warping, this leads to isometric deformations of the rigid bodies. However, their approaches are limited to unconstrained free vibrations of the articulated characters or constraints that are linear in the generalized coordinates. Hence, for many constraints, *e.g.* if the feet are fixed to the ground, these techniques cannot be applied. Being able to use constrained vibration modes is essential for our framework for spacetime optimization.

The second contribution concerns the preservation of angular momentum for in-air phases. We are proposing a technique that, after warping of the modes, modifies the global rotation of the character such that the angular moment is preserved during the motion. This process is integrated to the spacetime optimization so that the character can use the coupling of the pose and the angular velocity during the optimization, *e.g.* it can change its pose early in the animation in order to influence its final position.

The third contribution is our modeling of the spacetime optimization for articulated characters. Using the linearized dynamics and warping instead of the fully non-linear dynamics reduces the complexity of the optimization problem to be solved. The resulting spacetime optimization problem provides a structure that we can use to design an efficient specialized solver. In particular, we use vibration modes, wiggly splines and warping to explicitly describe the solutions of the optimization problem. This greatly reduces the complexity of the optimization problem since we can restrict the optimization to a low-dimensional space of wiggly spline coefficients instead of a general set of admissible motions. In addition, this modeling avoids using a time discretization so the derivatives of the motion can be explicitly evaluated.

2 Related Work

Constrained spacetime optimization, introduced by Witkin and Kass [1988], provides a powerful framework for creating character animations from constraints specified by an animator. The spacetime optimization selects, from the set of motions that satisfy the constraints, the motion that requires the least effort from the character. The effort is measured by the spacetime L^2-norm of the forces the character needs to generate to perform the motion. The resulting motion is planned and effective.

Spacetime constraints can produce realistic and interesting motions. However, the problem of spacetime optimization is that the high-dimensional and non-linear optimization problem has to be solved. This has limited the approach to simple characters and many methods are specialized for certain types of motions. Various strategies for improving solvers or reducing the complexity of the problem have been proposed. Cohen [1992] used hierarchical wavelets to develop an interactive spacetime control system that enables a user to interact with the iterative numerical minimization and to guide the optimization process. For the computation of the derivatives of the objective functional, automatic and symbolic differentiation have been used (see [Guenter 2007] and references therein). Fang and Pollard [2003] derived a strategy for efficiently evaluating the first derivatives of a broad range of constraints. Their system avoids the explicit evaluation of the torques, which speeds up the computation. Chai and Hodgins [2007] replace the physical model of the character dynamics with a statistical dynamics model learned from a motion capture database.

Reduction of the number of degrees of freedom has been used for efficiently synthesizing motion from motion capture data. For motion editing, Popović and Witkin [1999] selected the character's degrees of freedom most important for the task and restricted the spacetime optimization to these degrees. Safonova et al. [2004] used motion capture data to automatically construct low-dimensional spaces for a specific type of motion. For synthesizing new motion, the spacetime optimization is restricted to this space. Our framework also applies a dimensional reduction. In contrast to this work, our construction does not rely on motion capture data and therefore is applicable for more general types of characters and motions where captured data is not available. Jain and Liu [2011] use an analysis of the vibration modes of motion capture data for the design of their controller. Spacetime optimization has been used for human and animal locomotion. Wampler and Popović [2009] combine local spacetime optimization with a genetic algorithm to generate plausible locomotion gaits for a variety of virtual creatures. Mordatch et al. [2013] use spacetime optimzation for animation of the human lower limbs. Kim et al. [2009] introduce a scheme for editing the motions of multiple characters.

Reduction has also been used for the efficient *simulation* of coupled rigid- and softbodies by Barbič and Zhao [2011], Kim and James [2011], and Kim and Pollard [2011b]. Kim and Pollard [2011a] introduced a scheme for direct control of the coupled systems. It is an interesting direction of future work to design a system for long-horizon motion planning for such coupled systems.

Besides articulated characters, spacetime optimization has been used for controlling other types of physical systems including elastic ropes and strings [Barzel 1997], rigid bodies [Popović et al. 2003], and fluids [Treuille et al. 2003; McNamara et al. 2004; Wojtan et al. 2006]. Furthermore, the computation of optimal trajectories of deformable objects has recently received much attention. Our approach is inspired by this development and transfers techniques developed for controlling deformable objects to articulated characters. Barbič et al. [2009] use dimensional reduction to restrict the spacetime optimization problem for deformable objects to a low-dimensional space constructed from the keyframes, vibration modes, and tangent vectors of a deformation curve. Modal warping, introduced by Choi and Ko [2005], aims at reducing linearization artifacts in large deformations for simulations in reduced space spanned by linear vibration modes. Huang et al. [2011] introduce rotation-strain warping. This warp map couples the different modes for warping. Li et al. [2013] integrate spatial constraints to the rotation-strain warping process. To our knowledge, non of these warping processes can deal with the coupling of shape and angular velocity for in-air settings. Barbič et al. [2012] use this warp map for an interactive system for editing of simulations or animations of deformable objects. Kass and Anderson [2008] introduced the wiggly spline, a new family of splines that in addition to smooth interpolants can also produce oscillating functions. Hildebrandt et al. [2012] use wiggly splines for computing optimal trajectories of deformable objects. They derive a closed-form representation of the wiggly splines and show that the optimal trajectories of linearized deformable objects can be described using modal coordinates and wiggly splines. We use both of these results for our approach. To avoid linearization artifacts, they use linearizations of the equations of the motion around all the keyframes. To allow for more general constraints, Schulz et al. [2014] replace the multipoint linearization by rotation-strain warping. Li et al. [2014] propose a framework that not only optimizes the motion, but also the materials of a deformable object.

3 Modal Analysis for Articulated Characters

In this section, we review some basics concerned with vibration modes of articulated characters. First, we discuss the modal analysis of linear systems of second-order ODEs. Then, we show how

this framework can be applied to compute vibration modes of articulated characters.

3.1 Linear modal analysis

We consider a system with n degrees of freedom and linear dynamics. The state of the system is described by a time-dependent vector $u(t)$ and the dynamics are given by a coupled system of second-order ODEs of the form

$$M\,\ddot{u}(t) + D\,\dot{u}(t) + K\,u(t) + g = f(t), \tag{1}$$

where M, D, and K are the mass, damping, and stiffness matrix, g is a constant vector and f is a time-dependent force vector. We use the Rayleigh damping model, which means we assume that the damping matrix has the form $D = \alpha\,M + \beta\,K$ for some constant damping parameters α and β.

The eigenmodes ϕ_i and eigenfrequencies $\sqrt{\lambda_i}$ of (1) are the solutions to the generalized eigenvalue problem

$$K\,\phi_i = \lambda_i M\,\phi_i. \tag{2}$$

Let $\{\phi_1, \phi_2, ..., \phi_n\}$ be a set of eigenmodes that forms a basis of the n-dimensional space of possible configurations. When (1) is transformed into such a basis, the system of equations decouples into a system of n independent ODEs. These 1-dimensional equations have the form

$$\ddot{\omega}_i(t) + \delta_i\,\dot{\omega}(t) + \lambda_i\,\omega(t) + g_i = f_i(t), \tag{3}$$

where $\delta_i = \alpha + \beta\,\lambda_i$, $g_i = \phi_i^T g$, and $f_i(t) = \phi_i^T f(t)$. Each equation describes the motion of the system in the direction of the corresponding eigenmode ϕ_i.

3.2 Vibrations of Articulated Characters

The dynamics of articulated characters are non-linear, and, therefore, a linearization of the equations of motion is needed before the linear modal analysis can be applied. While modal analysis is widely used in graphics (*e.g.* for simulating deformable objects [Pentland and Williams 1989; Hauser et al. 2003; Barbič and James 2005] and shape analysis and modeling [Hildebrandt et al. 2010; Hildebrandt et al. 2011]) its application to animating articulated characters is a more recent development [Kry et al. 2009; Jain and Liu 2011; Nunes et al. 2012].

We consider a character that is described by a tree-structured skeleton, *i.e.*, a system of rigid bodies (bones) connected by joints. Let us assume that the character has μ bones and that for each bone, we have a representation in a fixed coordinate system. Then the configuration of each bone is described by a rigid motion (a translation and a rotation) and the joints impose holonomic constraints on the system.

Generalized coordinates q that parametrize the manifold of possible configurations can be obtained in the following way. To parametrize the rotation matrices, we use Euler angles. The first six generalized coordinates describe the rigid motion of the root bone. Then we traverse the skeleton from the root to the leaves. Each of the μ-1 joints is parametrized by one, two, or three Euler angles (depending on the joint type) that describe the relative rotations of the two bones connected by the joint. The (absolute) orientation of any bone is the product of the (relative) rotation matrices of all successors (in the skeleton tree) and the rotation matrix specifying the orientation of the root bone. We denote the translation of the root bone by $t(q)$ and the rotation matrices specifying the (absolute) orientation of the i^{th} bone by $R_i(q)$. The configuration of the character can be constructed by traversing the tree structure. During the traversal, each bone is rotated using the corresponding matrix $R_i(q)$ and translated so that it connects to its predecessor.

The non-linear dynamics of an articulated character can be modeled by a system of ODEs in the generalized coordinates

$$M(q)\ddot{q} + (C(q,\dot{q}) + D(q))\dot{q} + G(q) + K\,q + k_0 = f(t). \tag{4}$$

Here M is the mass matrix, C represents the centrifugal and Coriolis forces, G an external force field (*e.g.*, gravity), and f the generalized forces. Each of the joints is equipped with a spring and a damper to model passive forces. The matrices D and K represent the damping and stiffness coefficients and k_0 is a constant vector. Choices of stiffness coefficients for humans and animals are discussed in [Liu et al. 2005; Kry et al. 2009]. An alternative approach for modeling the stiffness matrix K was introduced by Hahn et al. [2012]. They model the virtual character as an elastic solid and use the mass and stiffness matrix of the elastic solid to compute vibration modes.

Let q_0 be a configuration of the character, and let $u = q - q_0$ denote a displacement. Following [Jain and Liu 2011], we linearize the equations of motion (4) around q_0 and obtain a system of the form (1), where M is the mass matrix $M(q_0)$ at q_0, K the stiffness matrix and $g = k_0 + G(q_0) + Kq_0$. Note that C does not appear in the equation because $C(q_0, 0)$ vanishes. As in Section 3.1, we use Rayleigh damping, *i.e.,* we set $D = \alpha\,M + \beta\,K$ for some constant parameters α and β.

3.3 Modes of constrained vibrations

In addition to free vibrations, we consider constrained vibrations where positions of points of the character, the orientation of bones, or the position of the center of mass are fixed. We assume that the constraints are satisfied for the configuration q_0 around which we linearize. Then the constraints on the displacement u are linear. We construct a basis of the set of admissible displacements by computing an SVD of the matrix representing the constraints and restrict the stiffness and mass matrices to this space. The vibration modes of the constrained character are the eigenvectors of the generalized eigenvalue problem (2) for the restricted matrices.

4 Warping Linearized Characters

After linearization of the equations of motion, the matrices that describe the linearized motion are no longer rotation matrices. Hence, the linearized motion is not an isometry of the rigid bodies. For example, a motion along a linear vibration mode is only plausible for small displacements u. We propose a warping strategy that aims to reduce these artifacts. The warping process differs for unconstrained (*in-air*) motion and motion with constraints. We discuss the case of constrained motion first.

The warping of the linearized deformation is carried out in two steps. First, a target rotation $\tilde{R}_i$ for every bone is determined. Then, a configuration of the character is computed that satisfies the constraints and is as-close-as-possible to the target rotations. For the first step, we use the Lie-group structure of the manifold of rotation matrices $SO(3)$. Each linearized rotation $R_i(q_0)+\mathrm{d}R_i\,u$ is an element of a tangent space of $SO(3)$. Here $\mathrm{d}R_i\,u$ denotes the derivative of R_i in direction u. We use the exponential map to map the linearized rotation (tangent vector) to the corresponding rotation (element of the manifold). This can be implemented using the matrix exponential exp. Let us consider the case $R_i(q_0) = Id_{\mathbb{R}^3}$ first. Then, $\mathrm{d}R_i\,u$ is an anti-symmetric 3×3 matrix and $\tilde{R}_i(q_0, u) = \exp(\mathrm{d}R_i\,u)$ is the corresponding target rotation matrix. In the gen-

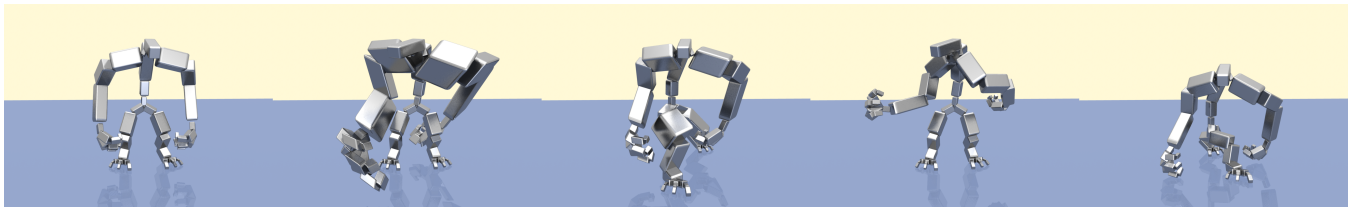

Figure 2: *Low-frequency vibration modes with fix and sliding constraints for the character. From left to right: the neutral pose, two linear modes (exhibiting artifacts), two warped poses (artifacts are removed).*

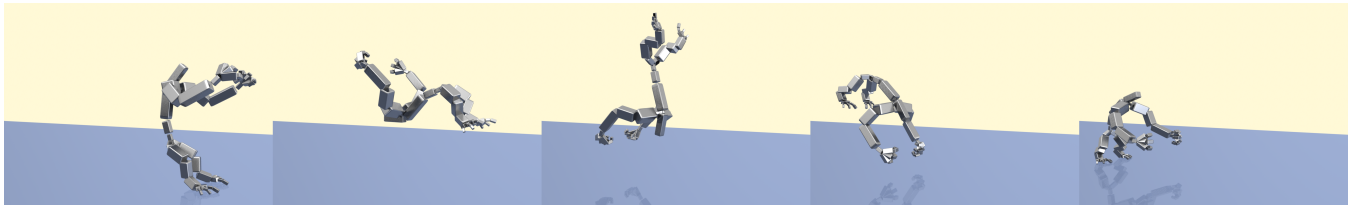

Figure 3: *Snapshots of an animation of a character performing a somersault computed with our in-air warp strategy.*

eral case (in which $R_i(q_0)$ is an arbitrary rotation) the target rotation is given by

$$\tilde{R}_i(q_0, u) = R_i(q_0) \exp((R_i(q_0))^T dR_i\, u). \tag{5}$$

The transpose $(R_i(q_0))^T$ maps $dR_i\, u$ from a tangent vector in $T_{R_i(q_0)}SO(3)$ to the corresponding tangent vector $(R_i(q_0))^T dR_i\, u$ in $T_{Id}SO(3)$, which is an antisymmetric matrix. The matrix exponential maps this matrix to an element of $SO(3)$. By multiplying this element with $R_i(q_0)$, we obtain the rotation matrix $\tilde{R}_i(q_0, u)$ corresponding to $dR_i\, u$.

In the second step, we construct a configuration of the character described by μ matrices W_i and the translation $\tau(q_0, u)$, such that the W_i best match the target rotations $\tilde{R}_i$. There are different possible ways to design this process. For the spacetime optimization, we want to have a warp map that is differentiable and allows for an efficient evaluation of the map itself as well as its derivative. To meet these criteria, we relax the condition that the matrices W_i describing the warped configuration must be rotations. Instead, we minimize over the all 3×3 matrices and penalize the deviation away from the target rotations $\tilde{R}_i$. Then the constraints on the position of bones, points on a bone, or the center of mass are linear constraints on the matrices W_i. We define the warped configuration to be the solution to the least-squares problem

$$W_i(q_0, u) = \arg\min_{W_i} \sum_{i=1}^{\mu} \left\| W_i - \tilde{R}_i(q_0, u) \right\|^2. \tag{6}$$

The matrices W_i cannot be computed separately since the constraints couple the W_is. The warped configuration, given by the W_is, depends non-linearly on the displacement u since the definition of $\tilde{R}_i(q, u)$ involves the matrix exponential. Still, the derivatives of W_i with respect to variations of u can be computed. After applying the chain rule to (6) the only difficulty is to compute the derivative of the matrix exponential. This can be efficiently done using Rodrigues' rotation formula and a truncated Taylor series for matrices with small Frobenius norm. For a detailed description of this technique, we refer to [Wisniewski 2010, Chapter 8.2].

In our experiments, the warp map nicely counteracts artifacts caused by linearization. Examples of warping of linear vibration modes are shown in the supplementary video. In addition to the application to spacetime optimization, we expect that this warp map can be helpful for other tasks that involve vibration modes of characters (e.g., the design of modal motion and locomotion controllers [Kry et al. 2009; Jain and Liu 2011; Nunes et al. 2012]).

4.1 Warping "in-air" motion

When computing motion for unconstrained characters (*e.g.* jumps), we adjust the global rotation of the warped motion to ensure that the angular momentum is preserved. As a first step, we decompose the interval $[a, b]$, over which the motion is parametrized, into segments $a = \tau_0 < \tau_1 < \ldots < \tau_\nu = b$. For every τ_j, we can compute the inertia tensor J_j and the angular velocity Ω_j of the warped motion. From this, we get the angular momentum $L_j = J_j\Omega_j$. We first compute L_0, which is the momentum to be preserved. Then, we traverse the interval and add a global rotation of the character so that the angular momentum at every τ_j equals L_0. The missing global angular velocity $\tilde{\Omega}_j$ at time τ_j is given by

$$L_0 = L_j + J_j\tilde{\Omega}_j,$$

which we can solve for $\tilde{\Omega}_j$,

$$\tilde{\Omega}_j = J_j^{-1}(L_0 - L_j).$$

A rotation with this angular velocity is given by

$$\exp((t - \tau_j)\, A_{\tilde{\Omega}_j}),$$

where $A_{\tilde{\Omega}_j}$ is the anti-symmetric matrix representing the linear map $v \mapsto \tilde{\Omega}_j \times v$. In the interval $[\tau_j, \tau_{j-1}]$, we multiply the matrices W_i describing the warped motion of the character with this rotation. After this process the warped motion's angular momentum L_j at τ_j equals L_0. This process is part of the warp map and in the optimization the deformation of the character is coupled to the angular velocity of the global rotation. Our example of somersault shows how the optimization uses this effect. To make the whole rotation in time, the character rounds its back and tucks in its legs in order to increase the angular velocity. Finally, it lands on its feet.

5 Spacetime Optimization

Spacetime optimization provides a framework for animating virtual characters. The control parameter that an animator can use are con-

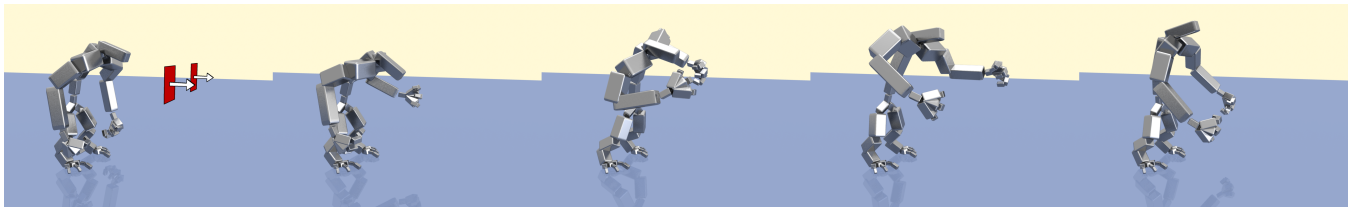

Figure 4: *Setup and snapshots of an animation of a character performing four fast punches. From left to right: the used start and end pose together with partial position and velocity constraints for the hands, four poses while punching.*

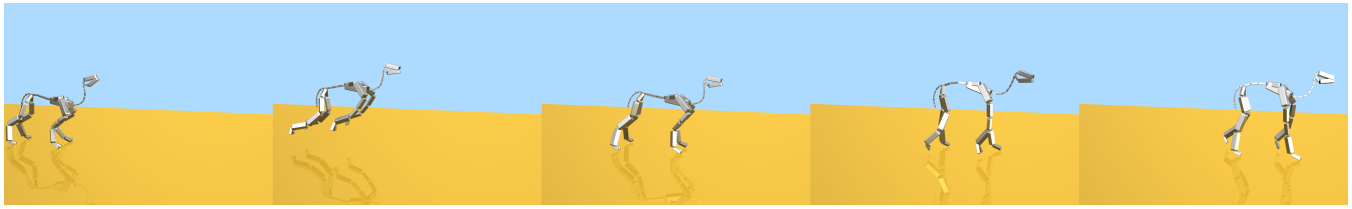

Figure 5: *Snapshots of an animation of a four-legged character jumping forward followed by some skating steps.*

straints in spacetime (*e.g.*, a set of keyframes). Then an optimal motion is computed that interpolates or approximates the keyframes. The motion is optimal in the sense that the squared spacetime L^2-norm of forces the character needs to generate for performing the motion is minimal.

In this section, we describe our modeling of the spacetime optimization. It combines the linearization of the dynamics and the warp map to obtain an optimization framework that produces plausible and interesting motion and can be efficiently and robustly solved. We want to emphasize that the warping is integrated with the optimization and not used as a post-process after the optimization. Furthermore, we want to point out that the warp map couples the modes. Hence, the modes are coupled in the optimization.

We first discuss the user-specified constraints and spacetime functional we want to minimize. Then, we show that the solutions of the optimization problem can be described using modal coordinates, wiggly splines, and the warp map. Finally, we show how this structure can be used to efficiently compute optimal motions.

User-specified constraints Our goal is to enable a user to prescribe only a few constraints and to determine the most degrees of freedom in the optimization. This means we prescribe only partial, as opposed to full, keyframes. For example, we prescribe the position of the feet after the somersault. In addition, the velocity of the motion can be explicitly constrained. For example, we can prescribe the average velocity at some specified point in time.

We want to emphasize that we control the warped motion and therefore formulate constraints for the motion after warping. Let $W(u(t))$ denote the vector-valued function that lists the coefficients of the translation $\tau(q_0, u(t))$ of the root bone and the warped matrices $W_i(q_0, u(t))$. Then, the vector $\frac{d}{dt}W(u(t)) = \mathrm{d}W(\dot{u}(t))$ describes the derivative of the warped motion. We denote the interval over which we want to optimize by $[t_0, t_m]$ and the nodes at which constraints are given by $t_0 < t_1 < \ldots < t_m$. For every $k \in \{0, 1, ..., m\}$, we have a vector-valued function C_k depending on $W(u(t_k))$ and $\mathrm{d}W(\dot{u}(t_k))$ defining the constraint on the warped motion at time t_k

$$C_k(W(u(t_k)), \mathrm{d}W(\dot{u}(t_k))) = 0.$$

For our experiments, we only used constraints that are linear in $W(u(t_k))$ and $\mathrm{d}W(\dot{u}(t_k))$. For example, constraints on the position of a point on a bone, the orientation of a bone, or the center of mass. Note that the functions C_k are linear in $W(u(t_k))$ and $\mathrm{d}W(\dot{u}(t_k))$ and non-linear in $u(t_k)$ and $\dot{u}(t_k)$. In the optimization, we will enforce these constraints in the least-squares sense using the energy

$$\begin{aligned} &E_C(u(t_0), \dot{u}(t_0), ..., u(t_m), \dot{u}(t_m)) \\ &= \textstyle\sum_k \left\| C_k(W(u(t_k)), \mathrm{d}W(\dot{u}(t_k))) \right\|^2 . \end{aligned}$$

Optimal forces To move, the character needs to generate forces. The motion of the character and the forces used are related by the equations of motion. Our goal is to compute the optimal motions for which the character spends the least amount of forces. To keep the functional that determines the forces simple, we use the linearized equations of motion to calculate the cost of a motion. The cost of a motion is modeled by an energy that measures the squared spacetime L^2-norm of the forces (the integral over the character and the time interval)

$$\begin{aligned} E(u) &= \frac{1}{2}\int_{t_0}^{t_m} \|f(t)\|^2_{M^{-1}} \, \mathrm{d}t \\ &= \frac{1}{2}\int_{t_0}^{t_m} \|M\,\ddot{u}(t) + D\,\dot{u}(t) + K\,u(t) + g\|^2_{M^{-1}} \, \mathrm{d}t. \end{aligned} \tag{7}$$

Here we use the notation $\|f(t)\|^2_{M^{-1}} = f^T(t)M^{-1}f(t)$. The M^{-1}-norm is used here, since f is the mass-weighted force and $M^{-1}f$ the pointwise force. Then $\|f(t)\|^2_{M^{-1}} = \|M^{-1}f(t)\|^2_M$ is the squared L^2-norm of the force at time t. The idea of measuring the squared norm of the forces goes back to [Witkin and Kass 1988]. The linearized equations of motion have been used for spacetime optimization of deformable objects in [Barbič et al. 2012] and [Hildebrandt et al. 2012].

Modal coordinates In a basis $\{\phi_1, \phi_2, ..., \phi_n\}$ of eigenmodes of the generalized eigenvalue problem (2), the equations of motion in (7) decouple. This is very helpful because by transforming u to this basis, we see that the Energy $E(u)$ can be written as a sum

$$E(u) = \textstyle\sum_i E_i(\omega_i), \tag{8}$$

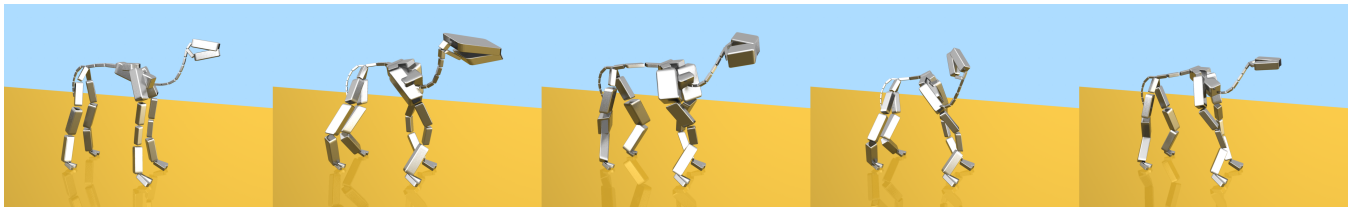

Figure 6: *Low-frequency vibration modes with constraints for the four-legged character. From left to right: the neutral pose, two linear modes (exhibiting artifacts), two warped poses (artifacts are removed).*

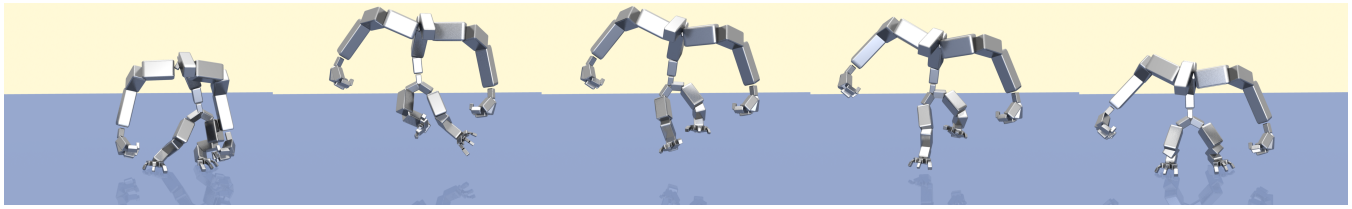

Figure 7: *Snapshots of an animation of a sidewards jumping character.*

where ω_i is the component of u in direction of the eigenmode ϕ_i and E_i measures the squared L^2-norm of the force in the direction of ϕ_i. The dynamics in the direction of an eigenmode is described by (3), and E_i takes the form

$$E_i(\omega_i) = \frac{1}{2}\int_{t_0}^{t_m} \left(\phi_i^T f(t)\right)^2 \,\mathrm{d}t \qquad (9)$$
$$= \frac{1}{2}\int_{t_0}^{t_m} \left(\ddot{\omega}_i(t) + 2\,\delta_i\,\dot{\omega}_i(t) + \lambda_i\,\omega_i(t) + g_i\right)^2 \,\mathrm{d}t.$$

The functions minimizing E_i are called *wiggly splines* [Kass and Anderson 2008]. A closed-form description of the wiggly splines was introduced in [Hildebrandt et al. 2012].

Wiggly splines The Euler-Lagrange equation of the variational problem is a linear fourth-order ODE that is described in [Hildebrandt et al. 2012]. In the generic case, *i.e.* $\delta_i, \lambda_i, \delta_i^2 - \lambda_i \neq 0$, the four-dimensional space of functions satisfying the Euler-Lagrange equation is spanned by the functions

$$b^{1,2,3,4}(t) = \mathrm{Re}\left(e^{\left(\pm\delta_i \pm \sqrt{\delta_i^2-\lambda_i}\right)t}\right). \qquad (10)$$

These are damped and driven oscillations. The two-dimensional space of damped oscillations, which is the solution space of (3), is a subspace of the four dimensional space. The fact that the damped and driven oscillations differ only by the sign of the damping parameter δ_i shows that the optimal way to inject forces into the system is to excite the oscillator in the same way as it is damped.

For a given set of constraints on the position and/or velocity at a set of $m+1$ nodes $\{t_0, t_1, \ldots, t_m\}$, the wiggly spline is the minimizer of the energy $E_i(\omega_i)$ among the twice-weakly differentiable functions that satisfy the constraints. Within each interval (t_k, t_{k+1}), the wiggly spline is a linear combination of the basis functions plus a constant

$$\omega_i(t)_{|[t_{k-1},t_k]} = \omega_{i,k}(t) = \sum_{l=1}^{4} w_{i,k}^l b^l(t) - c_i,$$

where $w_{i,k}^1, w_{i,k}^2, w_{i,k}^3$, and $w_{i,k}^4$ are coefficients and $c_i = g_i/|\lambda_i|$ if $\lambda_i \neq 0$ and $c_i = 0$ if $\lambda_i = 0$. This construction shares similarities with the construction of cubic splines, however, the basis functions are not polynomials but rather oscillatory functions.

Solving the optimization problem The functional we want to minimize is the weighted sum

$$\mathcal{E}(u) = E(u) + \gamma\, E_C(u(t_0), \dot{u}(t_0), ..., u(t_m), \dot{u}(t_m)). \qquad (11)$$

The energy E ensures that the motion uses the forces efficiently and E_C makes sure the motion approximates the constraints specified by the user. Since E_C depends only on the position and velocity of u at the nodes t_i, a minimizer u of (11) must satisfy

$$\nabla E(u) = 0$$

within each of the intervals (t_i, t_{i+1}). In a basis of eigenmodes $\{\phi_1, \phi_2, ..., \phi_n\}$, the energy takes the form (8). Since the functions satisfying $\nabla E_i(\omega_i) = 0$ are the wiggly splines, u must have the form

$$u(t) = \sum_i \omega_i(t)\phi_i, \qquad (12)$$

where each $\omega_i(t)$ is a wiggly spline. Note that the type of wiggly splines $\omega_i(t)$ (e.g. the parameter of the basis functions b_i) depends on the eigenvalue λ_i and the damping parameters α and β. For solving the optimization problem this means that instead of minimizing over arbitrary functions, we only need to minimize over a space of wiggly splines. We further reduce the complexity of the problem by restricting the optimization to the $10 - 30$ lowest frequency eigenmodes. This can be done by limiting the sum in (12) to these modes. Then, we have to determine $4\,m\,\bar{n}$ wiggly spline coefficients $w_{i,k}^l$ in the optimization, where $\bar{n}$ is the number of eigenmodes. Since a choice of coefficients $w_{i,k}^l$ uniquely describes a motion, we have constructed a low-dimensional space of explicitly parametrized motions for our problem.

The dimension of the space over which we optimize can be further reduced. The wiggly splines that describe the minimizer are once continuous and differentiable at all t_k. This means that the coefficients $w_{i,k}^l$ satisfy the linear conditions

$$\omega_{i,k}(t_k) = \omega_{i,k+1}(t_k) \quad \dot{\omega}_{i,k}(t_k) = \dot{\omega}_{i,k+1}(t_k)$$

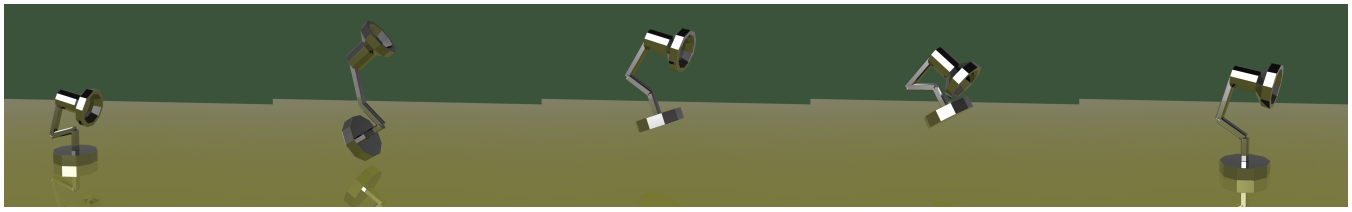

Figure 8: *Poses computed for a lamp performing a forward jump.*

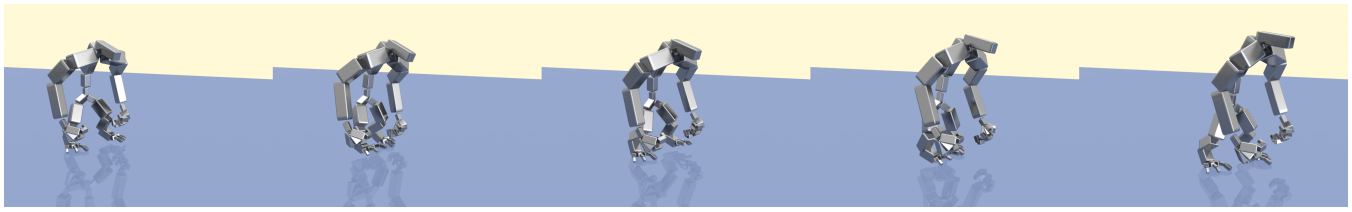

Figure 9: *Snapshots of an animation of a two legged character walking.*

for all $k \in \{1, 2, ..., m-1\}$. Furthermore, if no velocity is prescribed at node t_k, the spline is twice differentiable at t_k. Then, the linear condition

$$\ddot{\omega}_{i,k}(t_k) = \ddot{\omega}_{i,k+1}(t_k)$$

holds. In addition to prescribing keyframes in the least-squares sense, we can force the motion to interpolate a keyframe (or partial keyframe). We specify such conditions at the start and/or end of the motion. These constraints again impose linear conditions on the coefficients $w^l_{i,k}$. The space of wiggly coefficients $w^l_{i,k}$ that satisfy the condition can be computed using an SVD of the matrix representing the linear conditions. The resulting low-dimensional nonlinear least squares problem can be solved with standard solvers. We used a Gauss–Newton solver in our experiments.

6 Experiments

The supplementary video shows motions produced with our implementation of the proposed framework. Table 1 summarizes the details concerning the characters and the numerical optimization. We specified the characters' masses and the stiffness coefficients of the joints by hand. Following [Kry et al. 2009], we assign a stiffness coefficient to every joint, a high stiffness to the spine and lower stiffnesses to the knees and elbows. In our experiments, the Gauss–Newton scheme required 10-20 iterations to solve the optimization problems. The times listed in the table are the total times required to set up and solve all the optimization problems needed to generate a motion. The numbers of degrees of freedom of the characters, the number of splines used for the two and four legged characters, and the dimension of the reduced spaces are listed as well.

The video shows low-frequency eigenvibrations of the lamp, the four legged and the two legged character. Snapshots are shown in Figure 2 and 6. To illustrate the effect of our warp map, we first show vibrations along a linear mode and then the warped motions. The motion along the linear mode exhibits deformations of the bones. These artifacts are reduced by the warp map. The warp map, the modes and the wiggly splines are used for the construction of the reduced spaces, to which we restrict the optimization. Each element of the reduced space is a motion. We want to emphasize that since the warp map is non-linear, the space of warped motions is non-linear space. We perform the optimization in the linear space of wiggly spline coefficients. This space parametrizes the non-linear space of motions.

Animations	DoF	Ws	Modes	Dim	Opt
four punches	180	5	15	30	2.2s
walk	180	5	20	40	4.7s
side jump	180	3	20	40	1.5s
somersault	180	3	15	30	26s
camel ice-skating	150	7	20	40	4.5s
camel hand stand	150	5	20	40	45s

Table 1: *Statistics measured on a custom Laptop. From left to right: number of degrees of freedom of the character, number of composite splines, number of modes, dimension of the resulting optimization problems (number of wiggly splines coefficients to be determined for each spline), and total time for setting up and solving the optimization problems.*

Figure 4 shows snapshots of an animation of the two-legged character performing four consecutive punches. This sequences consist of five segments with a continuous but non-differentiable transition. The non-differentiable transition is needed to model the impact of the object which is punched by the character. Each segment is modeled by one wiggly spline for each modal coordinate. The first segment starts with a prescribed keyframe for the position and zero velocity and ends with a prescribed position and velocity only for the punching hand. The second, third, fourth and fifth segment start with the final position and velocity of the previous segment, except that the velocity at the previous punching hand is set to zero. The segments end with a partial position and velocity constraint on the punching hand, except for the last segment, which ends with a prescribed keyframe for the position and zero velocity.

Some animations contain walking and sliding steps, see Figure 5 and 9. These steps are modeled similarly to the punches, to get a continuous but non-differentiable transition. These segments start with the final position and velocity from the previous segment. The feet on one side are constrained to the ground, i.e. corresponding velocities are set to zero. And for the opposite feet we prescribe a partial position and velocity constraint at the end of the segment. By switching constraints between right and left feet, we create the walking and sliding steps.

The video shows six sequences that include jumps: two forward jumps see Figure 5 and 8, one sidewards jump see Figure 7, two somersaults see Figure 3 and 10 and a twisting jump see Figure 1.

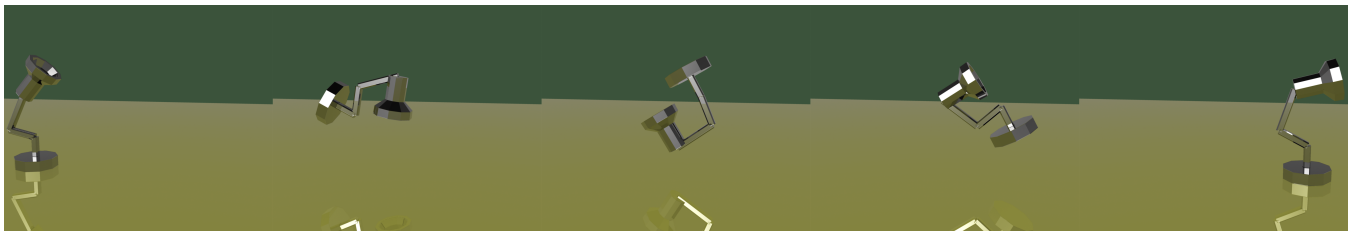

Figure 10: *Snapshots of the lamp performing a somersault.*

Each jump is modeled by three segments: preparing, in-air, and landing. Each segment is modeled by one wiggly spline for each modal coordinate. In the first segment, the base of the lamp, the feet of the two legged character and the feet of the forelegs of the four legged character are fixed to the ground. We prescribe the initial position and zero initial velocity. For the twisting jump, the preparation starts with the end pose of the previous segment. To reach the handstand pose, we start with a full keyframe and set only partial constraints on the feet of the hind legs.

For the forward and sidewards jumps, we only prescribe the velocity of the center of mass at the end of the preparation segment. For the somersaults and the twist, we additionally set a constraint for the angular velocity. Furthermore, we set position constraints on the feet of the hind legs and head for the preparation of the twisting jump. Once the first segment is computed, it defines initial positions and velocities for the in-air segment. In addition, the velocity of the center of mass determines the motion of the center of mass. This motion, in turn, determines the duration of the second segment, which ends when the center of mass has reached a certain height. To make sure that the character lands on its feet, we prescribe a partial keyframe for the position of the feet. The start pose and velocity of the landing segment are defined as the end of the in-air segment. At the end of the landing segment, full positions and velocities are prescribed. In this segment, we further allow a planar linear damped motion of the center of mass. This motion depends on the end velocity of the previous segment and results in overall damped sliding motion of the character on the ground plane.

During the in-air phase of the somersault, our in-air warping strategy is used. The deformation of the character is coupled to the bodies angular velocity, and, to make the whole rotation in time, the character has to round its back and tuck in its legs. These effects are automatically produced by our spacetime optimization framework.

7 Conclusion

We present a new framework for spacetime optimization of the motion of articulated characters that can be used for creating general types of motion and does not require motion capture data or an initial motion that satisfies the constraints. Our modeling of the spacetime optimization problem combines a linearization of the dynamics with a novel warping scheme for articulated characters. Using the structure provided by this modeling, we devise an efficient strategy for solving the problem that combines vibration modes, wiggly splines, and our warping scheme. This reduces the computation of an optimal motion to solving a low-dimensional and non-linear least-squares problem.

7.1 Limitations and challenges

Our current framework is limited to equality constraints, both at the keyframes and for the whole animation. A challenging problem is to effectively integrate inequality constraints into the spacetime optimization. Related is the problem of collision handling. Currently, we model collision using keyframes, velocities, and equality constraints.

Another challenge would be to extend the approach to coupled rigid- and softbody systems [Kim and Pollard 2011b], *e.g.*, to control characters with a soft belly. In addition, it would interesting to integrate the spacetime control directly into animators rigs (e.g. using rig-space physics [Hahn et al. 2012; Hahn et al. 2013]).

Acknowledgements

We would like to thank the anonymous reviewers for their comments and suggestions and Mimi Tsuruga for proofreading the text. This work was supported by the Max Planck Center for Visual Computing and Communication (MPC-VCC) and the DFG Project *KneeLaxity: Dynamic multi-modal knee joint registration* (EH 422/1-1).

References

BARBIČ, J., AND JAMES, D. L. 2005. Real-time subspace integration for St. Venant-Kirchhoff deformable models. *ACM Trans. Graph. 24*, 3, 982–990.

BARBIČ, J., AND ZHAO, Y. 2011. Real-time large-deformation substructuring. *ACM Trans. Graph. 30*, 4, 91:1–91:8.

BARBIČ, J., DA SILVA, M., AND POPOVIĆ, J. 2009. Deformable object animation using reduced optimal control. *ACM Trans. Graph. 28*, 53:1–53:9.

BARBIČ, J., SIN, F., AND GRINSPUN, E. 2012. Interactive editing of deformable simulations. *ACM Trans. Graph. 31*, 4.

BARZEL, R. 1997. Faking dynamics of ropes and springs. *IEEE Comput. Graph. Appl. 17*, 31–39.

CHAI, J., AND HODGINS, J. K. 2007. Constraint-based motion optimization using a statistical dynamic model. *ACM Trans. Graph. 26*.

CHOI, M. G., AND KO, H.-S. 2005. Modal warping: Real-time simulation of large rotational deformation and manipulation. *IEEE Trans. Vis. Comput. Graphics 11*, 1, 91–101.

COHEN, M. F. 1992. Interactive spacetime control for animation. *Proc of ACM SIGGRAPH 26*, 293–302.

FANG, A. C., AND POLLARD, N. S. 2003. Efficient synthesis of physically valid human motion. *ACM Trans. Graph. 22*, 3, 417–426.

GUENTER, B. 2007. Efficient symbolic differentiation for graphics applications. *ACM Trans. Graph. 26*.

HAHN, F., MARTIN, S., THOMASZEWSKI, B., SUMNER, R., COROS, S., AND GROSS, M. 2012. Rig-space physics. *ACM Trans. Graph. 31*, 4, 72:1–72:8.

HAHN, F., THOMASZEWSKI, B., COROS, S., SUMNER, R. W., AND GROSS, M. 2013. Efficient simulation of secondary motion in rig-space. In *Proceedings of the Symposium on Computer Animation*, 165–171.

HAUSER, K. K., SHEN, C., AND O'BRIEN, J. F. 2003. Interactive deformation using modal analysis with constraints. In *Graphics Interface*, 247–256.

HILDEBRANDT, K., SCHULZ, C., VON TYCOWICZ, C., AND POLTHIER, K. 2010. Eigenmodes of surface energies for shape analysis. *Proceedings of Geometric Modeling and Processing*, 296–314.

HILDEBRANDT, K., SCHULZ, C., VON TYCOWICZ, C., AND POLTHIER, K. 2011. Interactive surface modeling using modal analysis. *ACM Trans. Graph. 30*, 5, 119:1–119:11.

HILDEBRANDT, K., SCHULZ, C., VON TYCOWICZ, C., AND POLTHIER, K. 2012. Interactive spacetime control of deformable objects. *ACM Trans. Graph. 31*, 4, 71:1–71:8.

HUANG, J., TONG, Y., ZHOU, K., BAO, H., AND DESBRUN, M. 2011. Interactive shape interpolation through controllable dynamic deformation. *IEEE Transactions on Visualization and Computer Graphics 17*, 7, 983–992.

JAIN, S., AND LIU, C. K. 2011. Modal-space control for articulated characters. *ACM Trans. Graph. 30*, 5.

KASS, M., AND ANDERSON, J. 2008. Animating oscillatory motion with overlap: wiggly splines. *ACM Trans. Graph. 27*, 3, 28:1–28:8.

KIM, T., AND JAMES, D. L. 2011. Physics-based character skinning using multi-domain subspace deformations. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, New York, NY, USA, SCA '11, 63–72.

KIM, J., AND POLLARD, N. S. 2011. Direct control of simulated nonhuman characters. *IEEE Comput. Graph. Appl. 31*, 4 (July), 56–65.

KIM, J., AND POLLARD, N. S. 2011. Fast simulation of skeleton-driven deformable body characters. *ACM Trans. Graph. 30*, 5, 121:1–121:19.

KIM, M., HYUN, K., KIM, J., AND LEE, J. 2009. Synchronized multi-character motion editing. *ACM Trans. Graph. 28*, 3 (July), 79:1–79:9.

KRY, P., REVERET, L., FAURE, F., AND CANI, M.-P. 2009. Modal locomotion: Animating virtual characters with natural vibrations. *Computer Graphics Forum 28*, 2, 289–298.

LI, S., HUANG, J., DESBRUN, M., AND JIN, X. 2013. Interactive elastic motion editing through spacetime position constraints. *Computer Animation and Virtual Worlds 24*, 3-4, 409–417.

LI, S., HUANG, J., DE GOES, F., JIN, X., BAO, H., AND DESBRUN, M. 2014. Space-time editing of elastic motion through material optimization and reduction. *ACM Trans. Graph. 33*, 4, 108:1–108:10.

LIU, C. K., HERTZMANN, A., AND POPOVIĆ, Z. 2005. Learning physics-based motion style with nonlinear inverse optimization. *ACM Trans. Graph. 24*, 3.

MCNAMARA, A., TREUILLE, A., POPOVIĆ, Z., AND STAM, J. 2004. Fluid control using the adjoint method. *ACM Trans. Graph. 23*, 3, 449–456.

MORDATCH, I., WANG, J. M., TODOROV, E., AND KOLTUN, V. 2013. Animating human lower limbs using contact-invariant optimization. *ACM Trans. Graph. 32*, 6, 203:1–203:8.

NUNES, R. F., CAVALCANTE-NETO, J. B., VIDAL, C. A., KRY, P. G., AND ZORDAN, V. B. 2012. Using natural vibrations to guide control for locomotion. In *Proc. of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, 87–94.

PENTLAND, A., AND WILLIAMS, J. 1989. Good vibrations: modal dynamics for graphics and animation. *Proc. of ACM SIGGRAPH 23*, 207–214.

POPOVIĆ, Z., AND WITKIN, A. 1999. Physically based motion transformation. In *Proceedings of SIGGRAPH*, 11–20.

POPOVIĆ, J., SEITZ, S. M., AND ERDMANN, M. 2003. Motion sketching for control of rigid-body simulations. *ACM Trans. Graph. 22*, 4, 1034–1054.

SAFONOVA, A., HODGINS, J. K., AND POLLARD, N. S. 2004. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Trans. Graph. 23*, 3, 514–521.

SCHULZ, C., VON TYCOWICZ, C., SEIDEL, H.-P., AND HILDEBRANDT, K. 2014. Animating deformable objects using sparse spacetime constraints. *ACM Trans. Graph. 33*, 4.

TREUILLE, A., MCNAMARA, A., POPOVIĆ, Z., AND STAM, J. 2003. Keyframe control of smoke simulations. *ACM Trans. Graph. 22*, 3, 716–723.

WAMPLER, K., AND POPOVIĆ, Z. 2009. Optimal gait and form for animal locomotion. *ACM Trans. Graph. 28*, 3.

WISNIEWSKI, K. 2010. *Finite Rotation Shells*. Springer.

WITKIN, A., AND KASS, M. 1988. Spacetime constraints. *Proc. of ACM SIGGRAPH 22*, 159–168.

WOJTAN, C., MUCHA, P. J., AND TURK, G. 2006. Keyframe control of complex particle systems using the adjoint method. In *Proc. Symp. Comp. Anim.*, 15–23.