Learning Physics-Based Motion Style with Nonlinear Inverse Optimization

C. Karen Liu
University of Washington

Aaron Hertzmann
University of Toronto

Zoran Popović
University of Washington

Abstract

This paper presents a novel physics-based representation of realistic character motion. The dynamical model incorporates several factors of locomotion derived from the biomechanical literature, including relative preferences for using some muscles more than others, elastic mechanisms at joints due to the mechanical properties of tendons, ligaments, and muscles, and variable stiffness at joints depending on the task. When used in a spacetime optimization framework, the parameters of this model define a wide range of styles of natural human movement.

Due to the complexity of biological motion, these style parameters are too difficult to design by hand. To address this, we introduce Nonlinear Inverse Optimization, a novel algorithm for estimating optimization parameters from motion capture data. Our method can extract the physical parameters from a single short motion sequence. Once captured, this representation of style is extremely flexible: motions can be generated in the same style but performing different tasks, and styles may be edited to change the physical properties of the body.

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

Keywords: Character animation, motion style, physics-based animation, inverse optimization

1 Introduction

Creating expressive and realistic character motion remains one of the main challenges in computer animation. Traditional keyframing techniques, while expressive, are not well-suited for achieving realism. Physics-based methods for locomotion synthesis show promise for highly dynamic motions such as jumping, diving, and gymnastics, but it remains very difficult to specify styles of motion. Dynamic simulation of low-energy motions — such as walking, jogging, and other common movements — are even more challenging, because these motions are not tightly constrained by physical requirements, and so physical style plays a significant role in determining motion. Style itself is very difficult to parameterize, especially in terms that can be applied to dynamic motion representation. More recent data-driven approaches to motion synthesis can preserve the realism provided by example motion capture data, but cannot produce new motions. Consequently, data-driven methods require a large database of training motions in order to allow flexibility. In this representation, the style and dynamics of motion are tightly coupled, so there is no way to reason about how the style of the motion would transfer to a motion with different dynamics.

In this paper, we present a physics-based approach to creating realistic, expressive motion. Our dynamic model includes an abstracted representation of an actor's muscles and tendons, sufficient to capture the essential qualities of locomotion arising from musculoskeletal structure. Furthermore, the model includes parameters that encode an actor's relative preference for applying torques at some joints more than others. New motions are created by spacetime optimization, minimizing the total muscle torques according to those preferences. The individual physics and style of an actor are described by the complete set of musculoskeletal parameters and muscle preferences, and modifying these parameters yields new motion styles.

Due to the complexity of biological motion, these style parameters are too difficult to design by hand. Moreover, it is controversial whether optimization is even a good model for human motion [Alexander 2001]. To address these questions, we introduce Nonlinear Inverse Optimization (NIO), a new algorithm for automatic estimation of physics parameters from motion capture data. NIO assumes that the motion capture is optimal for a spacetime optimization problem with unknown parameters and known constraints, and solves for physics parameters to make the observed motion optimal. We can then generate new motion sequences as if performed by that actor, in the same style as the real actor, but satisfying entirely new constraints. Because our method learns a high-level description of style, we do not require large training databases; the styles in this paper are estimated from a single short motion sequence each. For example, once we have extracted the style parameters of a specific walk, we can determine how this same person would move with a large briefcase in their hand.

Our physical model incorporates several hypotheses about locomotion from the biomechanics literature. First, there is a distinct preference for using specific joints rather than others, due to variations in joint strength, stability, and other factors [Full et al. 2002]. Second, biological systems use passive elements in their musculoskeletal structure, such as tendons and ligaments, to store and release energy, thereby reducing total power consumption [Alexander 1988]. Third, animals vary stiffness of their joints when performing different tasks. For example, leg stiffness is considerably higher during running than during walking [Farley and Morgenroth 1999; Ferris et al. 1999]. Incorporating these factors leads to increased realism in our model. Although some of these factors have been used in animation systems, they have not been used together in physics-based animation. This is likely due to the difficulty of selecting a large number of simulation parameters by hand, a problem we address by learning these parameters from data. Moreover, we anticipate that our approach can be used as a means to explore biomechanical theories; to this end, we show a preliminary experiment in which our system accurately predicts the overall features of a new motion, as compared to ground truth measurements.

In this paper, we focus on modeling human locomotion for two reasons. First, locomotion is central to human movement. Second, in contrast to high-energy motions such as high-jumping, it is much more difficult to generate realistic walking and other low-energy motions by optimization. Whereas high-energy motions are determined primarily by a small number of dynamic and physical constraints, low-energy motions require much more accurate, detailed models of dynamics and style. In fact, it has not previously been shown that full-body human walking is optimal with respect

email: {karenliu,zoran}@cs.washington.edu, hertzman@dgp.toronto.edu
web: http://grail.cs.washington.edu/projects/charanim/phys-style.html

to muscle-usage. Our learning and synthesis procedures are general and we anticipate that they will enable analysis of more general types of motions, as well as analysis of animals with different kinematics or dynamics from humans.

All biomechanical models involve simplifications, and ours is no exception. We use an abstracted representation of dynamics in order to capture the most salient elements of motion. The most significant simplification is that we treat joint stiffness as an element of style that does not vary during a motion. Consequently, walking and running — which normally entail different degrees of muscle stiffness — are treated as two different styles. Additionally, we employ a minimal model of the musculoskeletal system that represents aggregate forces at each joint, rather than the specific structure of individual muscles, bones, and tendons.

2 Related Work

Robot controller simulation has been successfully applied to the domain of realistic computer animation, yielding a variety of types of motions [Faloutsos et al. 2001; Hodgins et al. 1995; Hodgins and Pollard 1997; Raibert and Hodgins 1991; Laszlo et al. 2000; Sun and Metaxas 2001; Torkos and van de Panne 1998; van de Panne et al. 1994; van de Panne and Fiume 1993]. These methods yield physically valid motions, often in real-time. However, creating controllers for a given task remains a difficult process, and it is even more difficult to create a controller to represent a specific style of motion.

The spacetime constraints framework, in contrast to simulation, casts motion synthesis as a variational optimization problem of minimizing some physical measure of energy, such as muscle exertion [Liu et al. 1994; Rose et al. 1996; Liu and Popović 2002; Pandy 2001; Popović and Witkin 1999; Witkin and Kass 1988], or joint angle acceleration [Fang and Pollard 2003]. Optimal energy movement and intuitive control give this method great appeal. Unfortunately, for complex characters, Newtonian physics constraints are highly nonlinear, often preventing the spacetime optimization from converging to a good solution. This problem prevents spacetime optimization from being used when the starting guess for the optimization is far away from the desired solution. Because many aspects of the real-life physics are abstracted away from the model, the optimization tends to produce reasonable results only for high-energy motion (jumping, diving, acrobatics, etc.), because these motions are largely constrained by what is physically possible. Low-energy motions, such as walking and running, depend more on the fine details of the physical model, because there are many ways to perform these motions while still satisfying the physical constraints. Much of the motion style is determined by musculoskeletal intricacies that are not usually modeled. For this reason, when applied to low-energy motion, spacetime optimization is highly sensitive to the starting position of the optimization — the optimization often converges to a physically-valid but unrealistic solution. Safonova et al. [2004] obtain better convergence and more realistic motions by parameterizing motion within a low-dimensional subspace obtained from a collection of example motions. Our framework shows that realistic motions can be obtained within a purely energy-based model without a subspace projection or extra penalty terms. Additionally, our method requires only a single example motion to define a style, rather than a database of motions in the same style.

Because of the difficulties in directly modeling physics and style, learning simple models of style from examples has recently been an extremely active and productive area of research [Arikan and Forsyth 2002; Arikan et al. 2003; Brand and Hertzmann 2000; Grochow et al. 2004; Kovar et al. 2002; Kovar and Gleicher 2004; Lee et al. 2002; Li et al. 2002; Pullen and Bregler 2002]. These methods modify existing motion clips to create new motions according to some constraints, while maintaining the specific style and expressiveness of the original motions. However, since these methods do not explicitly model physics, the output is limited to direct modifications to the available motions. For example, if we only have clips of an actor walking, then we can only synthesize more walking, and not, say, climbing or descending stairs. Consequently, extremely large motion databases may be required for general-purpose synthesis. Our work aims to infer the physical system that produced a given motion, which provides the ability to generalize to many new motions that were not included in the training data; the representation of style is much more compact. Our work has the disadvantage that it is more computationally intensive, and can only capture styles described by the physical model. Motion filtering, warping, and retargetting methods [Gleicher 1998; Rose et al. 1998; Tak and Ko 2005; Unuma et al. 1995; Vasilescu 2002; Witkin and Popović 1995] can be used to modify existing motions, but are limited to small modifications of motion trajectories without changing constraints, such as the number of footsteps, and without maintaining dynamic validity of the motion. In constrast, our system is not tied to the particular events in the example motion, and can generate new physically-correct motions with new sequences of constraints and new lengths.

Neff and Fiume [2002] point out the importance of muscle and spring tension in motion, and apply these observations to keyframe animation. In their system, all parameters must be determined by an animator.

Previous Inverse Optimization algorithms search for energy functions in which the measured data is optimal; Heuberger [2004] provides a detailed survey of inverse optimization. Existing methods apply only when the forward optimization problem has restricted structure, such as linear programming and network-flow problems. Approximate inverse optimization is an open problem [Heuberger 2004]; we present NIO, a first attempt at addressing this problem area. NIO does not require special structure in the energy function, except that it be differentiable. NIO does not ensure that an inverse is found, but we have found it to produce good results nonetheless.

Alternatively, maximum likelihood and Bayesian learning methods can learn energy functions defined in terms of probabilities. However, these methods lead to objective functions with intractable integrals (Appendix C). Previous methods have used random sampling techniques to optimize this integral [Geyer and Thompson 1992; Hinton and Sejnowski 1986; Hinton 2002]. However, no existing algorithm is capable of efficient random sampling in our case, where the problems have thousands of dimensions and are subject to hard nonlinear constraints. However, NIO is inspired by Contrastive Divergence [Hinton 2002], a probabilistic method. We also show a connection between inverse optimization and maximum likelihood. In concurrent work, LeCun and Huang [2005] describe related energy learning methods for classification and regression.

Our work also relates to methods that learn dynamical systems from data. NeuroAnimator [Grzeszczuk et al. 1998] fits a neural network to a known dynamical system, whereas we focus on learning dynamics and a physical energy function from motion capture data. Bhat et al. estimates the parameters of a 2D rigid-body system [2002] or a cloth simulation [2003] from a video sequence. These methods focus on passive systems or systems in which all forces are known. In contrast, we address problems involving unknown forces designed to minimize an unknown energy function.

3 Overview

We view realistic human locomotion as a result of an energy-optimal process that achieves a given set of tasks represented by environment and goal constraints $\mathbf{C}$. To compute a new motion $\mathbf{X}$, we minimize the energy objective function $E(\mathbf{X};\theta)$ which computes the total amount of torque due to muscle forces (Section 5). The parameter vector θ encapsulates all elements of physical style: muscle/tendon elastic properties, shoe elastic parameters, and relative preferences for muscle usage at each joint. In Section 4, we describe our model of motion as a function of all external and internal forces: muscle torques, gravity, spring forces, internal elastic forces, ground contact forces, and shoe elastic forces.

Given a motion capture sequence $\mathbf{X}_T$ and constraints $\mathbf{C}$, we can estimate the parameter vector θ that gave rise to it. This is done by finding a θ for which $\mathbf{X}_T$ is the minimizer of $E(\mathbf{X};\theta)$. This search is performed by Nonlinear Inverse Optimization (NIO), as described in Section 6. The constraints $\mathbf{C}$ are estimated in a preprocess described in Appendix A. Having extracted the physical style θ, we can generate a wide range of motions in the same style as the example motion, by minimizing the energy function with the same θ but new constraints; examples are shown in Section 7.

4 Motion dynamics

The distinctive feature of our spacetime optimization framework is a representation that accounts for key aspects of the musculoskeletal structure: relative strength of muscles, impedance, and neutral position parameters of passive structures around each joint. We represent the character skeleton as a transformation hierarchy that comprises 18 body nodes, 29 joint DOFs and 6 root DOFs, and rotational joints are parameterized by exponential maps [Grassia 1998]. We write the Lagrangian equations of motion[1] so as to include the effect of generalized forces associated with DOF q_j:

$$\sum_{i\in N(j)} \frac{d}{dt}\frac{\partial T_i}{\partial \dot{q}_j} - \frac{\partial T_i}{\partial q_j} = Q_j \tag{1}$$

where T_i denotes the kinetic energy of body node i and $N(j)$ is the set of body nodes in the subtree of joint DOF q_j, and Q_j is the aggregate generalized forces acting on q_j. The kinetic energy of body node i can be computed as:

$$T_i = \frac{1}{2}\mathrm{tr}\left(\dot{\mathbf{W}}_i \mathbf{M}_i \dot{\mathbf{W}}_i^T\right) \tag{2}$$

where $\mathbf{W}_i$ is the chain of the transformations from the root of the skeleton to body node i and $\mathbf{M}_i$ is the mass tensor of the body node i. The left-hand side terms of Equation 1 can be computed as:

$$\frac{d}{dt}\frac{\partial T_i}{\partial \dot{q}_j} - \frac{\partial T_i}{\partial q_j} = \mathrm{tr}\left(\frac{\partial \mathbf{W}_i}{\partial q_j}\mathbf{M}_i \ddot{\mathbf{W}}_i^T\right) \tag{3}$$

The aggregate generalized force Q_j acting on a DOF q_j is a sum of generalized forces:

$$Q_j = Q_{m_j} + Q_{g_j} + Q_{p_j} + Q_{c_j} + Q_{s_j} \tag{4}$$

The right-hand-side terms in this expression represent the aggregate generalized forces due to muscles (Q_{m_j}), gravity (Q_{g_j}), passive springs and dampers (Q_{m_j}), ground contact (Q_{c_j}), and shoe springs (Q_{s_j}). These equations represent the forces at a specific time instant t; for brevity, the dependence on t is omitted from these equations.

We next describe the generalized forces in detail.

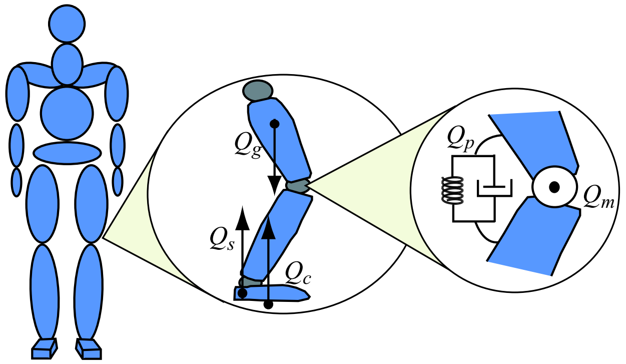


Figure 1: The character consists of 18 body nodes and 35 DOFs. The aggregate generalized forces acting on each joint are: muscles (Q_{m_j}), gravity (Q_{g_j}), passive springs and dampers (Q_{m_j}), ground contact (Q_{c_j}), and shoe springs (Q_{s_j}). The aggregate spring force from the passive elements (Q_p) is illustrated as a spring and a damper, and the active muscle force (Q_m) is illustrated as a motor.

Gravity. Gravity can be viewed as a constant force $m_i\mathbf{g}$ acting on the center of mass of each body part i. The generalized force due to gravity acting upon joint DOF q_j is computed as:

$$Q_{g_j} = \sum_{i\in N(j)} \left(\frac{\partial \mathbf{W}_i}{\partial q_j}\mathbf{c}_i\right)\cdot(m_i\mathbf{g}) \tag{5}$$

where $\mathbf{c}_i$ is the center of the body node i in its local coordinate frame, m_i is the mass of the body node i, and $\mathbf{g}$ is the gravitational acceleration.

Passive Joint Forces. Our model accounts for the passive joint forces due to the stretching of opposing muscles, tendons and ligaments. Tendons are stretchy tissue connecting muscles to the bones, and ligaments are fibrous tissue that join one bone to another across a joint, keeping the joint in place. Both tendons and ligaments act as spring-like elements that dampen motion. It is worth noting that these passive generalized forces are used extensively in natural locomotion to reduce energy consumption, increase stability and simplify the control. As the tissue around each joint stretches and contracts, energy is temporarily stored and released, thus increasing the efficiency of locomotion. In running, this mechanism of exchange between kinetic and elastic potential energy appears to conserve about 20-30% of the energy that would otherwise be supplied by muscles [Alexander 1988]. Similarly, although opposing muscles are the only real torque generators around each joint, they are also quite elastic and contribute to the aggregate passive forces around the joint. Our model separates the generalized force contribution of all muscles around a joint into a passive and active portion. If all muscle loads around a joint are kept constant, the entire joint system can be viewed as passive, even though all muscles might be actuated. Any variation of muscle loads away from this equilibrium is considered an active component of the generalized force, and is subsequently minimized with the objective function. Some studies suggest that animals keep overall leg stiffness fixed during walking and running [He et al. 1991], but vary stiffness according to the specific locomotion task, such as running on a surface of varying stiffness [Farley and Morgenroth 1999]. These collective spring-like effects are also significantly different for each joint. In the absence of all muscle forces and gravity, each joint also has a default rest state at the equilibrium of all muscle, tendon and ligament forces. NASA experiments have reported on these equilibrium joint positions for humans in a relaxed state in outer space, and reported

[1] The more common definition of the Lagrangian incorporates potential energy. We include gravity in the aggregate joint forces instead (which is equivalent to the more common form).

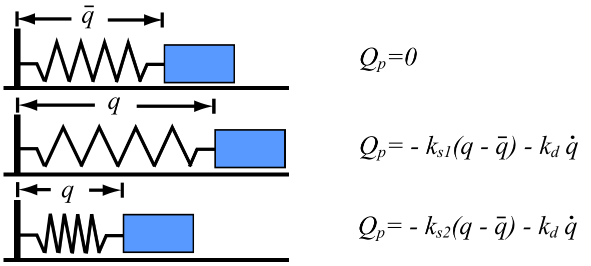

$\bar{q}$
q
q
$Q_p=0$
$Q_p = - k_{s1}(q - \bar{q}) - k_d\,\dot{q}$
$Q_p = - k_{s2}(q - \bar{q}) - k_d\,\dot{q}$

Figure 2: We use two different spring coefficients (k_{s1} and k_{s2}) to model the passive elements in stretching state and contracting state respectively. $\bar{q}_j$ is the joint angle of q_j at rest in absence of all external forces.

that the values are different for different people [Mount et al. 2003]. Opposing muscles around each joint can easily set these neutral positions to different values depending on the locomotion task.

We write the force due to passive elements as a linear damped spring force:

$$Q_{p_j} = -k_{s_j}(q_j - \bar{q}_j) - k_{d_j}\dot{q}_j \tag{6}$$

where k_{s_j} and k_{d_j} are the spring coefficient and damping coefficient that model the spring force caused by the stretchy tissue around joint DOF q_j, and $\bar{q}_j$ is the joint angle of q_j at rest in absence of all external forces. We use two different spring coefficients, k_{s1_j} and k_{s2_j}, to model the passive elements in stretching state and contracting state respectively (Figure 2). Since our optimizer requires forces to be continuous, we use a sigmoid function, $g(x) = (\tanh(500x)+1)/2$ to approximate the discontinuity at $q_j = \bar{q}_j$:

$$k_{s_j}(q_j) = g(q_j - \bar{q}_j)k_{s1_j} + (1 - g(q_j - \bar{q}_j))k_{s2_j} \tag{7}$$

Environment Constraint Forces. During ground contact, we use Coulomb's friction model as described by Pollard and Reitsma [2001] to compute the force caused by the friction between the character and the environment. A friction cone is defined to be the range of possible forces satisfying Coulomb's friction model for an object at rest. We ensure the contact forces stay within a basis that approximates the cones, with nonnegative basis coefficients λ_p:

$$Q_{c_j} = \sum_p \lambda_p \mathbf{V} \frac{\partial \mathbf{C}_p}{\partial q_j} \tag{8}$$

where $\mathbf{V}$ is a 4×3 matrix consisting of 4 basis vectors that approximately span the friction cone. Finally, $\frac{\partial \mathbf{C}_p}{\partial q_j}$ projects the contact force into the space of q_j, where $\mathbf{C}_p$ is a positional constraint that fixes a point on the character to its environment.

Shoe Forces. The spring-like nature of shoes contribute to the overall "bounciness" of locomotion. To simulate this elastic force, we use a spring that only activates when the distance between the foot and the floor is less than the rest length of the spring (Figure 3). Again, we use a sigmoid to approximate a step function:

$$Q_{s_j} = g(\bar{h} - h(\mathbf{q}))\, k_{shoe}(\bar{h} - h(\mathbf{q}))\frac{\partial h(\mathbf{q})}{\partial q_j} \tag{9}$$

where $\bar{h}$ denotes the rest length of the shoe spring, $h(\mathbf{q})$ indicates the vertical distance between the heel and the floor, and k_{shoe} denotes the spring constant for the shoe. As with the contact force, $\frac{\partial h(\mathbf{q})}{\partial q_j}$ projects the elastic force into the space of q_j.

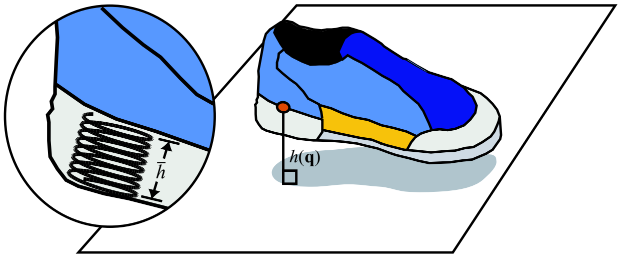


Figure 3: The elastic force of the shoes is modeled by a spring that only activates when the distance between the foot and the floor is less than the rest length $\bar{h}$ of the spring. $h(\mathbf{q})$ indicates the vertical distance between the heel and the floor.

4.1 Determining muscle forces

A complete motion is represented as a vector $\mathbf{X}$ containing joint angle configurations $\mathbf{q}$ and coefficients of ground contact forces λ: $\mathbf{X} = \{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_L, \lambda_1, \lambda_2, \ldots, \lambda_P\}$, where L is the number of the frames in the motion, and P is the number of footstep constraints. For efficiency, we parameterize motions as cubic B-splines, with a sufficient number of control points to allow detailed motion. The muscle forces Q_{m_j} can easily be computed from Equations 1 and 4, as a function of a motion $\mathbf{X}$, the physical parameters θ, and the time instant t:

$$Q_{m_j}(t, \mathbf{X}, \theta) = \sum_{i \in N(j)} \frac{d}{dt}\frac{\partial T_i}{\partial \dot{q}_j} - \frac{\partial T_i}{\partial q_j} - Q_{g_j} - Q_{c_j} - Q_{s_j} - Q_{p_j} \tag{10}$$

Since there are no muscles or tendons that apply forces directly to the root DOFs, a separate equation applies at the root; this equation says that the global motion of the character is completely determined by the aggregate external forces:

$$Q_{0_k}(t, \mathbf{X}, \theta) = \sum_{i \in N(0)} \frac{d}{dt}\frac{\partial T_i}{\partial \dot{q}_k} - \frac{\partial T_i}{\partial q_k} - Q_{g_k} - Q_{c_k} - Q_{s_k} = 0 \tag{11}$$

where k indexes over the 6 global DOFs at the root and $N(0)$ is the set of all body nodes.

5 Motion synthesis by minimizing muscle usage

The main functionality of muscles is to move bones around their joints by contracting and relaxing. While minimizing muscle usage certainly makes sense, optimization methods often neglect the large variability in muscle strength and usage preference for each joint. For example, the muscles driving the hip joint can generate significantly larger torque than the shoulder or elbow muscles. In addition, animals prefer to use certain muscles and joints simply because they may be more robust (less likely to sprain or tear) [Full et al. 2002]. Different muscle preferences significantly change the resulting style of the optimal motion. We will refer to the relative preference of power usage for joint DOF q_j by a corresponding scalar α_j. We specifically measure effort in terms of muscle force usage by summing the squared magnitudes of the forces at all joint DOFs j over all time steps t:

$$E^*(\mathbf{X}; \theta) = \sum_j \sum_t \alpha_j \left(Q_{m_j}(t, \mathbf{X}, \theta)\right)^2 \tag{12}$$

The weights α capture the relative preference for usage of different joint DOFs, and are normalized to sum to 1. The complete physical style of a character is collected in a parameter vector

$\theta = \{\alpha, \mathbf{k_s}, \mathbf{k_d}, \bar{\mathbf{q}}, k_{shoe}, \bar{h}\}$. In our system, the parameter vector θ is 147-dimensional.

In order for a motion to be physically valid, it should satisfy $Q_0 = 0$. However, our simplified skeleton does not provide enough accuracy to satisfy this constraint exactly. Instead, we add a soft constraint:

$$E(\mathbf{X};\theta) = \sum_j \sum_t \alpha_j (Q_{m_j}(t,\mathbf{X},\theta))^2 + w_r \sum_k \sum_t (Q_{0_k}(t,\mathbf{X},\theta))^2 \quad (13)$$

We use $w_r = 100$, a large value compared to α.

The motion with the specific physical style θ is computed as a solution to the following nonlinear optimization problem:

$$\min_{\mathbf{X}} E(\mathbf{X};\theta) \qquad \text{subject to} \qquad \mathbf{C}(\mathbf{X}) = 0 \quad (14)$$

where $\mathbf{C}$ denotes the footstep constraints and the bounds on $\mathbf{X}$. As a short-hand, we will also write this minimization as:

$$\min_{\mathbf{X} \in C} E(\mathbf{X};\theta) \quad (15)$$

For all examples in this paper, motion constraints were expressed in the form of constraints on footsteps. Specifically, each constraint fixes a point on one of the character's feet to a specific point in the environment for a specific period of time. These constraints are either provided by the user using a simple sketching tool we designed (see accompanying video), or extracted from a captured motion sequence, as described in [Liu and Popović 2002]. In our experience, manually creating footprints with reasonable positions, durations, and the frequency is not easy. We use captured footprints for those motions containing complex steps, such as for a sharp 180° turn.

6 Nonlinear Inverse Optimization

We now describe Nonlinear Inverse Optimization (NIO), a method for determining optimization parameters from measured data. Given an observed energy-optimal motion[2] $\mathbf{X}_T$; how can we determine the physical parameters θ that gave rise to it? One approach would be to minimize the least-squares difference between the observed motion and the result of spacetime optimization; however, as discussed in Appendix B, this approach leads to many difficulties.

We begin with the assumption that the motion $\mathbf{X}_T$ was generated by spacetime optimization as in Section 5. Consequently, the true motion parameters θ should satisfy

$$E(\mathbf{X}_T;\theta) = \min_{\mathbf{X} \in C} E(\mathbf{X};\theta) \quad (16)$$

However, it is not immediately apparent how one would search for a θ that satisfies Equation 16. Moreover, there is no guarantee that such a θ exists, because of noise and inaccuracies in the model.

Instead, we propose the following *Inverse Optimization Objective:*

$$G(\theta) = E(\mathbf{X}_T;\theta) - \min_{\mathbf{X} \in C} E(\mathbf{X};\theta) \quad (17)$$

This objective function has the property that $G(\theta) = 0$ only when θ satisfies Equation 16; $G(\theta) > 0$ otherwise. This means that any parameters θ that satisfy Equation 16 are global minima of $G(\theta)$. Even if we cannot find a θ that satisfies Equation 16, minimizing $G(\theta)$ will try to get $\mathbf{X}_T$ as "close" to being optimal as possible. Hence, we use $G(\theta)$ as an objective function for estimating θ. Additionally, in order to avoid degenerate solutions where $\alpha_j \equiv 0$, we would like to ensure $\sum_j \alpha_j = 1$, and, in order for muscle preferences to be plausible, we also require $\alpha_j \geq 0$ for all j. In practice, we use soft constraints:

$$D(\theta) = w || \sum_j \alpha_j - 1 ||^2 + w \sum_j S(\alpha_j) \quad (18)$$

where w is a large weight (we use $w = 10^4$), and $S(x)$ penalizes negative values ($S(x) = 0$ for $x \geq 0$; $S(x) = x^2$ for $x < 0$). The problem of determining θ from the observed motion is then

$$\arg\min_{\theta} G(\theta) + D(\theta) \quad (19)$$

The second term in Equation 17 cannot be evaluated exactly, as it would require global optimization. We evaluate it approximately using SNOPT [Gill et al. 1996], a non-linear optimizer. Equivalently, one may also modify the objective function to consider the local minimum discovered by SNOPT, rather than the global minimum. In this latter view, it is possible to design optimization algorithms for $G(\theta)$ that are guaranteed never to increase the objective function.

6.1 Learning algorithm

We now describe an algorithm for learning θ by minimizing $G(\theta)$. Standard search techniques cannot be applied because the objective function is highly nonlinear and non-differentiable: evaluation of $G(\theta)$ requires a solution to a complex non-linear minimization problem. For example, since θ is 147-dimensional, and each evaluation of $G(\theta)$ in our examples takes 4-8 minutes, computing a single gradient would take approximately 15 hours; even then there is a question of whether the gradient would be accurate. However, suppose, for a given estimate $\hat{\theta}$, we compute the optimal motion $\mathbf{X}_S = \arg\min_{\mathbf{X} \in C} E(\mathbf{X};\hat{\theta})$. The key idea of our algorithm is to locally approximate $G(\theta)$ with

$$\tilde{G}(\theta) = E(\mathbf{X}_T;\theta) - E(\mathbf{X}_S;\theta) \quad (20)$$

so that we may approximate the gradient of $G(\theta)$ at $\hat{\theta}$ as

$$\frac{d}{d\theta} G(\theta) \approx \frac{d}{d\theta} \tilde{G}(\theta) = \frac{\partial}{\partial\theta} E(\mathbf{X}_T;\theta) - \frac{\partial}{\partial\theta} E(\mathbf{X}_S;\theta) \quad (21)$$

This gives us an approximate gradient direction that can be used as a search direction within an iterative numerical optimization procedure: at each iteration, the algorithm computes a "counterexample motion" $\mathbf{X}_S$, evaluates Equation 21, and then updates $\hat{\theta}$ by taking a small step in the negative approximate gradient direction. We can interpret this algorithm as follows. During optimization, the current parameter estimate $\hat{\theta}$ views $\mathbf{X}_S$ as a motion that has lower energy than the observed motion $\mathbf{X}_T$ (Figure 4). Taking a step in the negative approximate gradient direction causes $\mathbf{X}_S$ to have higher energy and $\mathbf{X}_T$ to have lower energy, thus moving closer to a θ in which $\mathbf{X}_T$ has the lowest energy of all possible motions. The step-size is determined by a line-search with respect to $G(\theta) + D(\theta)$; this prevents a step from inadvertently making some other motion much better than $\mathbf{X}_T$. If $\hat{\theta}$ is optimal, then $\mathbf{X}_T$ and $\mathbf{X}_S$ have the same energy, and the approximate gradient is zero.

[2] We extract motion parameters $\mathbf{X}_T$ and constraints $\mathbf{C}$ from raw marker data as described in Appendix A.

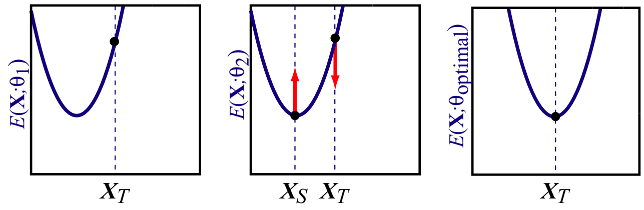


Figure 4: Intuition for NIO. The horizontal axis of each plot corresponds to a space of possible motions, and the vertical axis indicates the energy of each motion. The plot at the right shows our goal, namely, to find a θ for which $\mathbf{X}_T$ is "at the bottom" of the energy function. During optimization, however, we may have an energy function more like the one on the left, in which $\mathbf{X}_T$ is not at the bottom. In each step of the optimization process, we generate a motion $\mathbf{X}_S$ that has lower energy than $\mathbf{X}_T$, and then adjust θ to "push" $\mathbf{X}_T$ slightly downward and to "push" $\mathbf{X}_S$ slightly upward.

The entire algorithm may be summarized as follows:[3]

```
function NONLINEARINVERSEOPTIMIZATION($\mathbf{X}_T$)
    initialize $\hat{\theta}$
    while not done do
        $\mathbf{X}_S \leftarrow \arg\min_{\mathbf{X}\in C} E(\mathbf{X};\hat{\theta})$
        $\Delta\theta \leftarrow \frac{\partial}{\partial\theta}E(\mathbf{X}_T;\theta) - \frac{\partial}{\partial\theta}E(\mathbf{X}_S;\theta) + \frac{d}{d\theta}D(\theta)$
        $\beta \leftarrow \arg\min_\beta G(\hat{\theta}-\beta\Delta\theta) + D(\hat{\theta}-\beta\Delta\theta)$
        $\hat{\theta} \leftarrow \hat{\theta} - \beta\Delta\theta$
    end while
    return $\hat{\theta}$
```

We initialize α_j to be $1/M$, where M is the number of joint DOFs, for each DOF j. The rest pose $\bar{\mathbf{q}}$ is initialized as the average pose of $\mathbf{X}_T$. The shoe parameters k_{shoe} and $\bar{h}$ are initialized to the values obtained during preprocessing (Appendix A). We select initial values for $k_{s_1,j}$, $k_{s_2,j}$, and $k_{d,j}$ for each joint, by minimizing the inferred muscle forces:

$$\min_{\{k_{s1_j}\},\{k_{s2_j}\},\{k_{d_j}\}} \sum_j (Q_{m_j}(t,\mathbf{X},\hat{\theta}))^2 \qquad (22)$$

We ran the algorithm for exactly 50 iterations in each of our tests, although convergence could also be detected automatically by comparing successive values of the objective function. We found that the objective function typically decreased by several orders of magnitude within the first 10 steps, and then made tiny improvements after that (Figure 5), in a manner reminiscent of the linear convergence of gradient descent.

The bottleneck in this algorithm is in computing $\mathbf{X}_S$; however, this may be sped up by initializing SNOPT with $\mathbf{X}_T$, and by not running it to convergence (so that $\mathbf{X}_S$ is not necessarily optimal for $\hat{\theta}$, but is rather some motion which has lower energy than $\mathbf{X}_T$.)

7 Experiments

We tested our algorithm by learning the styles of several walking and running motion capture sequences. Each style is learned from a single motion sequence of 50-90 frames at 30 fps (2-3 seconds duration). We then used these dynamic style parameters to synthesize a wide range of different motions (Figure 6). We solve spacetime optimization problems using SNOPT [Gill et al. 1996]. The learning process took on the order of 4 to 6 hours per style, on a 2Ghz Pentium 4 machine. Synthesis took approximately 10 to 30 minutes per motion. During synthesis, we obtained somewhat faster convergence by optimizing explicit Q_{m_j} and Q_{0_k} variables together with the motion, and introducing explicit dynamics constraints (Equations 10 and 11).

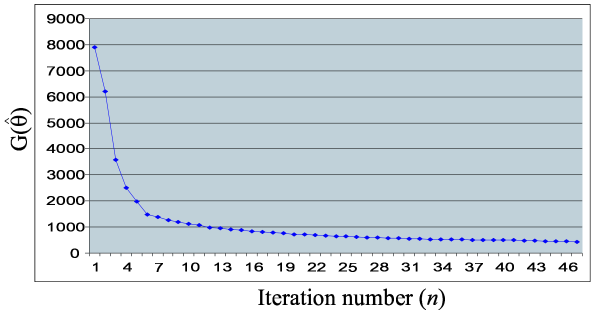


Figure 5: NIO of the neutral walk. The horizontal axis shows the iteration number n, and the vertical axis shows the value of $G(\hat{\theta})$. NIO obtains a good solution in very few steps. After learning, $E(\mathbf{X}_T;\hat{\theta}) = 8009.37$, and $E(\mathbf{X}_S;\hat{\theta}) = 7535.32$, indicating that, for the learned style $\hat{\theta}$, the energy of the optimal motion is very close to the energy of the observed motion.

Our synthesis algorithm does require a reasonable initial state. From our experiments, simple initializations, such as a default pose translating through space, lead to poor local minima. The following procedure was used for initialization in all of our experiments. Given new footstep constraints ($\mathbf{C}$) and the target motion ($\mathbf{X}_T$), we generate an appropriate initial sequence in the following three steps. First, we fit a spline to the horizontal coordinates of the footstep constraints, and initialize the horizontal coordinates of the root position to be the spline's position at each time instant t. Second, we initialize the global rotations with the spline tangents at each time t. Third, for each time t, we find the pose in the example sequence $\mathbf{X}_T$ that has the most similar footstep constraints to the constraints in time t, and copy the joint angles and root height to time t.

Estimating style parameters. We used NIO to learn the style of a neutral, balanced walking sequence. To evaluate the style parameters learned from this input sequence, we generated a motion with the learned style and with the same footprints as the input motion. As shown in the accompanying video, the synthesized motion is visually identical to the input motion. To demonstrate the importance of muscle preferences and passive elements in synthesis of natural motions, we designed following two experiments. First, we synthesized a motion with the same footprints as the input motion, but without considering muscle preferences ($\alpha = 1$ for all the body nodes) and without passive elements ($k_{s1_j} = k_{s2_j} = k_{d_j} = k_{shoe} = 0$). In the second experiment, we learned muscle preferences α in a model without passive elements and used the learned α to synthesize a motion constrained by the same footprints as the input motion. Note that learning the muscle preferences alone produces a motion reasonably close to the input motion. However, without spring and damper forces, the movement of some joints appear loose and unnatural.

Creating motions with new constraints. We can synthesize new motions in the same style as the previous walking sequence by

[3] The line search procedure is:

```
$\beta \leftarrow 1$
while $G(\hat{\theta}-\beta\Delta\theta) + D(\hat{\theta}-\beta\Delta\theta) > G(\hat{\theta}-\beta\Delta\theta/2) + D(\hat{\theta}-\beta\Delta\theta/2)$
    and $\beta > 10^{-6}$ do $\beta \leftarrow \beta/2$ end while
return $\beta$
```

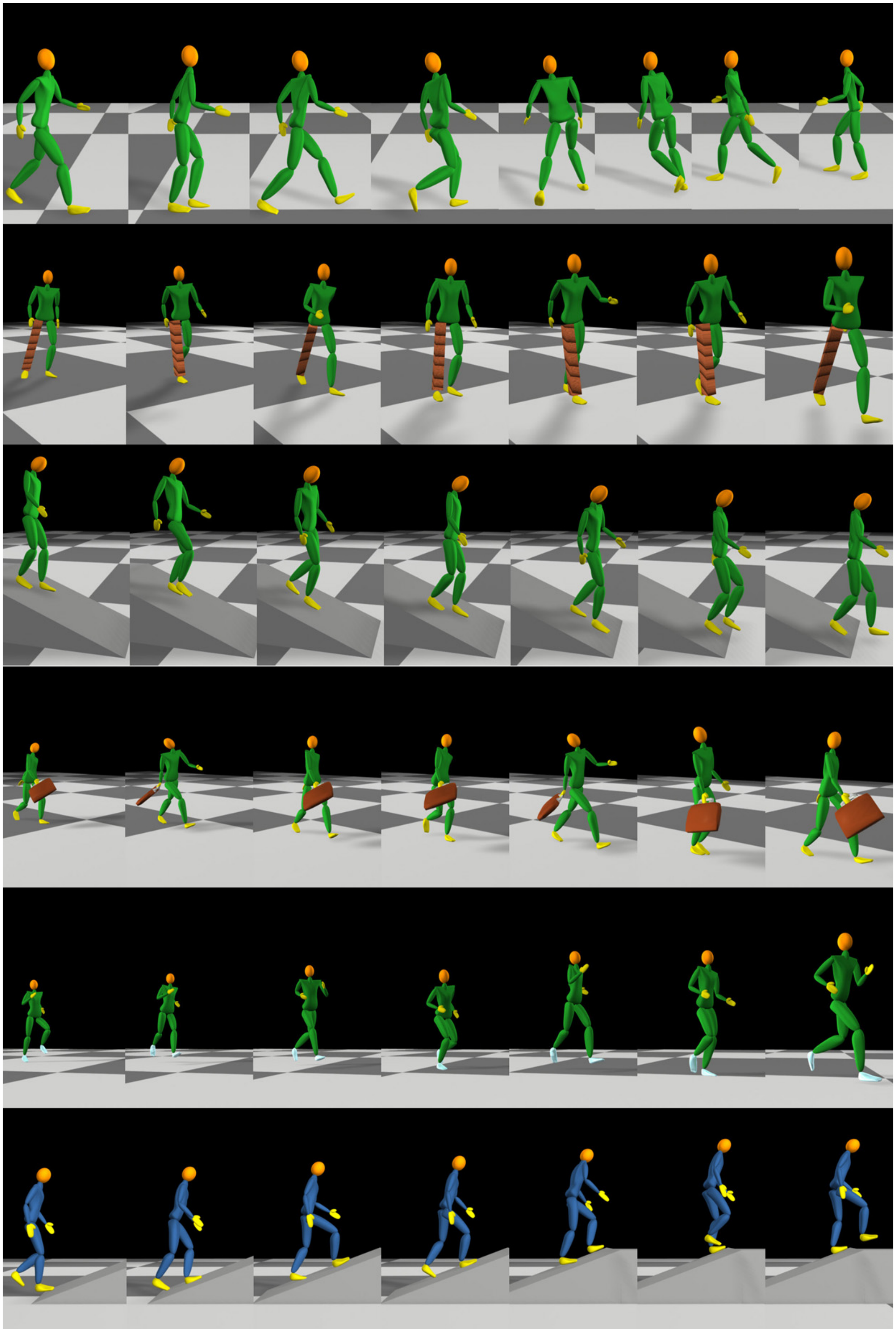


Figure 6: Examples of synthesized motions in various walking and running styles. From top to bottom: 180-degree walking turn, limp walk, descending an incline, walking with a suitcase, running with springy shoes, ascending an incline.

providing new footprint constraints. In the first example (shown in the accompanying video), we show a new walking sequence on a curved path. The new footprints caused the character to lean her torso into the turn. We also show the same style applied to a sharp 180° turn, where the character leans even further towards the center of rotation (Figure 6, first row). In addition to creating new footprints, we can also modify the character's skeleton. We show a motion sequence where we "locked" the character's left knee and decreased the range of movement on the joint of the left hip (Figure 6, second row). To perform the same gait, the character has to twist her torso more aggressively.

Capturing different styles. We have tested our style learning algorithm on a range of phenomena, such as variations due to emotional state, individual body shape, and functional activity such as walking or running. We learned a "sad" style from a captured walking sequence and synthesized walking uphill and downhill in the same "sad" style (Figure 6, third row). In another example, the actor was asked to act "happy" when we captured her walking motion. We allowed the footprint constraints to slide on the floor to create a skating motion in the "happy" style. Despite changes in constraints, the resulting motions still exhibit the same styles as the examples.

Our learning algorithm can learn different styles for different individuals. We recorded motions of two subjects walking on a level surface and synthesized walking uphill in their personal styles (Figure 6, sixth row). In our framework, running is considered a different style from walking because of the difference in muscle stiffness in these two actions. To illustrate this, we used the style parameters learned from running and applied them to walking. The character exhibits a lot of tension in her movements, since muscles are stiffer in running; the resulting motion resembles power-walking.

Editing styles and dynamics. We can also edit the style parameters and the dynamic properties. To illustrate this, we changed the mass of the character's right hand corresponding to carrying a 3 kilogram suitcase. As a result, the character leans to the left to counteract the weight and swings her right arm much less than before. Applying this change to different styles yields different optimal walking motions. For example, in the sad style, the character carries the suitcase in front of her body, whereas, in the happy style, she swings it back and forth (Figure 6, fourth row).

Our physics-based framework also models the elasticity of the character's shoes. By increasing the elasticity of the shoes, we create a bouncier running motion (Figure 6, fifth row).

Comparison to ground truth and warping. In order to evaluate our method, we compared it to a ground truth motion and to a motion warping method, in the case of walking uphill (Figure 7). We performed motion capture of an actor walking up a ramp. Then, using the neutral walking style learned from an actor walking on level ground, we synthesized a new motion with the same footstep constraints as the captured uphill motion. Note that our method accurately predicts the overall features of the ground truth motion, including leaning into the slope and applying larger forces at each step, even though these features are not present in the example motion. For comparison, we also generated the motion using a motion warping method that does not model dynamics; instead, it warps the example motion to the new constraints, and uses this motion as initialization in an optimization of the smoothness of the motion subject to footstep constraints. The warped motion does not capture the proper dynamics of the motion, e.g., the character does not lean into the slope.

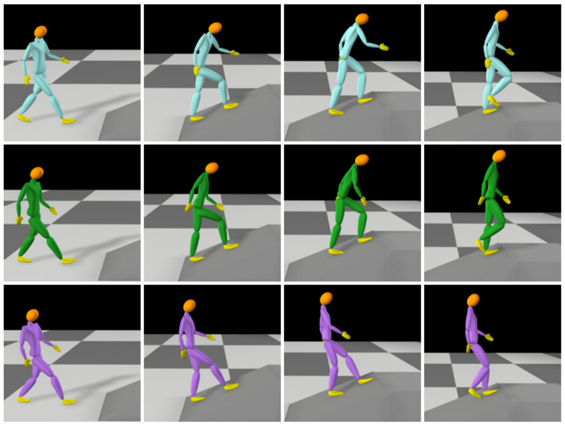


Figure 7: Comparison to motion warping and ground truth. *Top:* Motion capture of a person walking up a ramp. *Middle:* Motion predicted by our method, using a style learned from walking on a level surface. Although the prediction is not identical to the motion capture sequence, our method has accurately predicted the overall dynamic nature of the motion, such as leaning into the slope, and exerting more force at each step. *Bottom:* Motion predicted by warping the level motion and smoothing the motion while satisfying foot constraints. Many dynamic features of the ground truth are absent from the warped motion.

8 Discussion and future work

We have described a model of human locomotion that incorporates several important hypotheses of biological motion: optimality of locomotion, relative preferences for applying torques at different joints, the importance of spring and damper elements, and the importance of variable tension to style. We have also described a novel framework for learning biomechanical parameters from examples. We have found each of these components to be essential to producing realistic motions. For example, without springs, the character is unnaturally loose; without learning, it is too difficult to determine reasonable model parameters. The ability of our system to create realistic-looking motions, and, in the cases we have tested, to accurately predict real motions, strongly suggests that the system has accurately modeled the essential features of human locomotion.

Many open questions remain, as well as exciting avenues for future work. We anticipate that generalizations of this model can be used to model a very wide range of animal motion.

Musculoskeletal modeling. We have used a highly-abstracted model of dynamics, in order to capture the essential features of motion. There are a number of ways to generalize the model, such as detailed geometric models of bones, muscles, and tendons, and detailed models of muscle activation. One important simplification we have made is to keep muscle tension fixed, whereas humans vary stiffness for different tasks. A more sophisticated model would learn the energy cost due to varying muscle activations, although this may require a larger training database. Hence, our present system will not be able to accurately predict motions with different stiffness characteristics, such as accurately predicting the nature of a walking motion from running data.

We have found the learning process to be effective when given different biomechanical models. For example, an early version of our system used a poor model of ground contact forces; NIO was able to learn a reasonable model of most aspects of motion, but ground contact appeared unrealistic in synthesized motions. For this reason, we are optimistic that NIO will work well with a biomechanical models of greater or lesser complexity, subject to the descriptive power of the model. Determining the appropriate model complexity for various specific problems remains an open question.

Other types of motions and characters. We anticipate that our general approach can be applied to other types of motions and other types of animals, although the details of the biomechanical model may vary in different cases.

Model accuracy and uniqueness. The main assumption of our approach is that the example motion is energy-optimal. It seems reasonable to hypothesize that a "neutral" walking motion is optimal with respect to, for example, metabolic energy consumption. On the other hand, the energetic happy walk may not be the most energy-efficient method for locomotion. Nonetheless, it may be optimal with respect to a different energy function, one that, for example, reflects the happy person's strong preference for more exaggerated gestures than necessary. Our approach models these cases by a physical system that explains the motion and can generate new motions, but, in doing so, conflates emotional state with biomechanical properties.

An open question is to determine in what cases are the model parameters uniquely-defined by the motion capture data, and when the estimation is stable. For purposes of animation, uniqueness of the model parameters is not essential; what matters is the ability of the system to accurately generate new motions. We suspect that unique style of the model could be determined by learning θ parameters that fit a set of motions, rather than a single short motion. Furthermore, it would be valuable to perform detailed biomechanic analysis of locomotion variability by measuring the forces and tensions from real subjects performing a set of tasks, and comparing these measurements to those produced by our generative model.

Properties of NIO and extensions. We have found NIO to work well in practice, and it has a number of appealing theoretical properties, such as convergence when $G(\theta) = 0$. Yet we have an incomplete understanding of NIO's properties, such as whether it is guaranteed to solve $G(\theta) = 0$ when a solution exists. One intriguing question is whether we can learn the structure of problems, i.e., to determine biomechanical models or determine the constraints that were required to create motions.

There are a number of practical extensions, such as handling nuisance parameters, non-trivial noise levels, and model selection. These issues would be straightforward to model in a probabilistic setting, but this would lead to substantial computational challenges.

Stylistic variation. Since our model of style represents physical properties of motion, we anticipate that it can be used to generate new styles; one possible application is to create a linear space of styles that can be used to generate new motions or to recognize existing styles.

Biomechanics research. An important and controversial question in biomechanics is whether movements are "optimal" in any sense [Alexander 2001]. Based on our preliminary tests, we believe that NIO can be used in human motion research to create highly predictive models of motion based on optimization, thereby lending support to the optimization theory of motion.

Acknowledgements

We are grateful to Geoff Hinton for discussions, and to Marc Thyng and Mira Dontcheva for help with video preparation. This work was supported by the UW Animation Research Labs, NSF grants EIA-0121326, CCR-0092970, IIS-0113007, an NSERC Discovery Grant, the Connaught fund, an Alfred P. Sloan Fellowship, an NVIDIA Fellowship, Electronic Arts, Sony, and Microsoft Research.

A Motion preprocessing

We reconstruct a motion ($\mathbf{X}$) and the mass tensors ($\mathbf{M}$) directly from the raw data acquired by a motion capture system. The motion sequences were captured at the rate of 120 frames/second and then down-sampled to 30 frames/second. Reconstructing the motion entails estimating the joint angles and the ground contact forces at every time step. We have found that using standard inverse kinematics to estimate joint angles yields motions that appear accurate visually, but that contain unrealistic levels of noise. These minute variations correspond to very large derivatives, and thus to unrealistic forces. Instead, we formulated an spacetime optimization that fits each handle $h_i(\mathbf{q})$ on the character to the corresponding recorded marker p_i, subject to a dynamic constraints on the global DOFs:

$$\min_{\mathbf{X}, k_{shoe}, \bar{h}} \sum_i ||h_i(\mathbf{q}) - p_i||^2 \qquad \text{subject to} \qquad Q_0 = 0 \tag{23}$$

Motions obtained this way have much smoother second derivatives, while still matching the originally markers faithfully. Consequently, style parameters θ extracted from these motions are more robust for synthesizing motions with new constraints. Note that this optimization also estimates ground contact forces for all time steps (parameterized by λ coefficients) based solely on the motion of the character's center of mass. Inspecting measured motions suggests that vertical translation due to the root DOFs dominates all that due to all other DOFs, and so the above procedure should yield reasonably accurate ground contact forces. Measurements could also be performed on a force platform in order to obtain exact ground contact forces.

In order to determine the mass tensor for each ellipsoidal body node, we first set the major axis length to match the corresponding limb length, and then scale the other axes equally in order for the limb's volume to match the mass distribution for humans described in the biomechanics literature [de Leva 1996; Pearsall et al. 1994].

B Least squares learning

A tempting approach to learning θ is to solve for the θ that minimizes the following least-squares objective function:

$$||\mathbf{X}_T - \arg\min_{\mathbf{X} \in C} E(\mathbf{X}; \theta)||^2 \tag{24}$$

Our early tests with this approach were entirely unsuccessful. This approach is fraught with many difficulties. First, this objective function presumes that the observed motion is the unique minimizer of the energy function; if there is noise in the system, if there are approximations in the model, or if the energy function does not have a unique minimum, then the motion $\mathbf{X}_T$ may be different from that returned by an optimizer. Second, this objective function is likely to have many spurious local minima, because adjustments to θ may make very unpredictable changes to the optimal motion. Third, there does not appear to be a reliable procedure for producing search directions for this objective function; for example, gradient descent cannot be applied because we cannot compute the gradient of the objective. As a result, expensive search methods such as simulated annealing or finite differences would be required. These methods are very expensive even for low-dimensional problems; in our case, θ is 147-dimensional, which suggests that optimization could take days or even weeks. In contrast, NIO suffers from none of these problems: it is very fast, robust to initialization, and does not require a user-designed mutation function.

C Relation to Maximum Likelihood

In this section, we discuss theoretical properties of Inverse Optimization and how it relates to maximum likelihood (ML) learning. A common way to define the probability of an energy-based system is with a Gibbs distribution:

$$p(\mathbf{X}|\theta,\tau) = \frac{e^{-E(\mathbf{X};\theta)/\tau}}{\int_{\mathbf{X}\in C} e^{-E(\mathbf{X};\theta)/\tau} d\mathbf{X}} \tag{25}$$

where τ is called the "temperature." In ML, we would normally remove the constraint that $\sum_i \alpha_i = 1$, and remove the temperature; we then search for the θ that maximizes $p(\mathbf{X}_T|\theta)$. However, suppose we fix the value of τ; learning θ by maximizing $p(\mathbf{X}|\theta,\tau)$ is equivalent to minimizing

$$\begin{aligned} L_\tau(\theta) &= -\tau \ln p(\mathbf{X}_T|\theta,\tau) && (26) \\ &= E(\mathbf{X}_T;\theta) + \tau \ln \int_{\mathbf{X}\in C} e^{-E(\mathbf{X};\theta)/\tau} d\mathbf{X} && (27) \\ &= E(\mathbf{X}_T;\theta) - \int p(\mathbf{X}|\theta,\tau) E(\mathbf{X};\theta) d\mathbf{X} \\ &\quad -\tau \int p(\mathbf{X}|\theta,\tau) \ln p(\mathbf{X}|\theta,\tau) d\mathbf{X} && (28) \end{aligned}$$

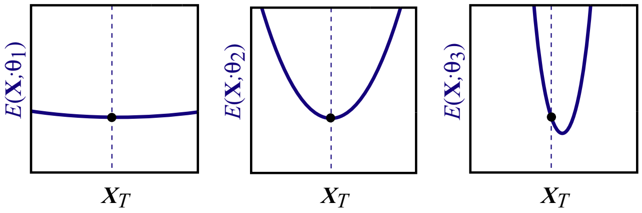
The equivalence of Equations 27 and 28 may be shown by substituting Equation 25 into the final instance of $p(\mathbf{X}|\theta,\tau)$ in Equation 28.

Now, consider the behavior of this optimization in the limit as $\tau \to 0$: $p(\mathbf{X}|\theta,\tau)$ will become a delta-function around the minimum-energy motion. Hence

$$\lim_{\tau\to 0} L_\tau(\theta) = E(\mathbf{X}_T;\theta) - \min_{\mathbf{X}\in C} E(\mathbf{X};\theta) = G(\theta) \tag{29}$$

Hence, the Inverse Optimization Objective can be viewed as ML learning in the zero-temperature limit. Furthermore, our optimization algorithm can be viewed as a zero-temperature form of Contrastive Divergence [Hinton 2002], since sampling from the delta-function is equivalent to finding the minimum-energy motion.

Developing algorithms for ML learning of θ is a promising but challenging avenue for future work. We suspect that the ML estimate of θ would be more useful than the one produced by our algorithm, as it would likely handle noise more robustly, and provide a proper probability distribution over motions. Moreover, consider the following optimization scenarios with three possible choices of θ that all assign the same energy to the target motion $\mathbf{X}_T$:



The Inverse Optimization Objective views both θ_1 and θ_2 as optimal, since $G(\theta_1) = G(\theta_2) = 0$. However, ML prefers θ_2 to θ_1, since it assigns higher probability to the target motion $\mathbf{X}_T$. (This follows from $\int_{\mathbf{X}\in C} e^{-E(\mathbf{X};\theta_1)} d\mathbf{X} > \int_{\mathbf{X}\in C} e^{-E(\mathbf{X};\theta_2)} d\mathbf{X}$). Similarly, ML would usually prefer θ_3 over θ_1 and θ_2, whereas Inverse Optimization would prefer θ_1 or θ_2. Intuitively, not only do we want the observed motion to be at the bottom of a "bowl" in the energy function, but the bowl should be as deep as possible.

References

Alexander, R. M. 1988. *Elastic Mechanisms in Animal Movement*. Cambridge University Press.

Alexander, R. M. 2001. Design By Numbers. *Nature 412* (Aug.), 591.

Arikan, O., and Forsyth, D. A. 2002. Synthesizing Constrained Motions from Examples. *ACM Transactions on Graphics 21*, 3 (July), 483–490. (Proceedings of ACM SIGGRAPH 2002).

Arikan, O., Forsyth, D. A., and O'Brien, J. F. 2003. Motion synthesis from annotations. *ACM Transactions on Graphics 22*, 3 (July), 402–408.

Bhat, K. S., Seitz, S. M., Popović, J., and Khosla, P. K. 2002. Computing the physical parameters of rigid-body motion from video. *Lecture Notes in Computer Science 2350*, 551–566.

Bhat, K. S., Twigg, C. D., Hodgins, J. K., Khosla, P. K., Popović, Z., and Seitz, S. M. 2003. Estimating cloth simulation parameters from video. In *Eurographics/SIGGRAPH Symposium on Computer Animation*, ACM Press, 37–51.

Brand, M., and Hertzmann, A. 2000. Style machines. *Proceedings of SIGGRAPH 2000* (July), 183–192.

de Leva, P. 1996. Adjustments to Zatsiorsky-Seluyanov's segment inertia parameters. *J. of Biomechanics 29*, 9, 1223–1230.

Faloutsos, P., van de Panne, M., and Terzopoulos, D. 2001. Composable Controllers for Physics-Based Character Animation. In *Proceedings of SIGGRAPH 2001*, 251–260.

Fang, A. C., and Pollard, N. S. 2003. Efficient synthesis of physically valid human motion. *ACM Transactions on Graphics 22*, 3 (July), 417–426.

Farley, C. T., and Morgenroth, D. C. 1999. Leg Stiffness Primarily Depends on Ankle Stiffness During Human Hopping. *Journal of Biomechanics 32*, 267–273.

Ferris, D. P., Liang, K., and Farley, C. T. 1999. Runners Adjust Leg Stiffness for Their First Step on a New Running Surface. *Journal of Biomechanics 32*, 787–794.

Full, R. J., Kubow, T., Schmitt, J., Holmes, P., and Koditschek, D. 2002. Quantifying dynamic stability and maneuverability in legged locomotion. *Integ. and Comp. Biol 42*, 129–157.

Geyer, C. J., and Thompson, E. A. 1992. Constrained Monte Carlo maximum likelihood for dependent data. *J. Roy. Statist. Soc. Ser. B 54*, 657–699.

Gill, P., Saunders, M., and Murray, W. 1996. SNOPT: An SQP algorithm for large-scale constrained optimization. Tech. Rep. NA 96-2, University of California, San Diego.

Gleicher, M. 1998. Retargeting Motion to New Characters. *Proceedings of SIGGRAPH 98* (July), 33–42.

Grassia, F. S. 1998. Practical parameterization of rotations using the exponential map. *Journal of Graphics Tools 3*, 3, 29–48.

Grochow, K., Martin, S. L., Hertzmann, A., and Popović, Z. 2004. Style-based Inverse Kinematics. *ACM Transactions on Graphics* (Aug.), 522–531.

Grzeszczuk, R., Terzopoulos, D., and Hinton, G. 1998. NeuroAnimator: Fast Neural Network Emulation and Control of Physics-Based Models. *Proceedings of SIGGRAPH 98* (July), 9–20.

He, J., Kram, R., and McMahon, T. A. 1991. Mechanics of running under simulated low gravity. *J. of Applied Physiology 71*, 863–870.

Heuberger, C. 2004. Inverse Combinatorial Optimization: A Survey on Problems, Methods, and Results. *J. Comb. Optim. 8*, 329–361.

Hinton, G. E., and Sejnowski, T. J. 1986. Learning and relearning in Boltzmann machines. In *Parallel Distributed Processing, Volume 1: Foundations*, D. E. Rumelhart and J. L. McClelland, Eds. 282–317.

Hinton, G. E. 2002. Training Products of Experts by Minimizing Contrastive Divergence. *Neural Computation 14*, 8, 1771–1800.

Hodgins, J. K., and Pollard, N. S. 1997. Adapting Simulated Behaviors For New Characters. *Proc. SIGGRAPH 97*, 153–162.

Hodgins, J. K., Wooten, W. L., Brogan, D. C., and O'Brien, J. F. 1995. Animating Human Athletics. *Proc. SIGGRAPH 95* (August), 71–78.

Kovar, L., and Gleicher, M. 2004. Automated Extraction and Parameterization of Motions in Large Data Sets. *ACM Transactions on Graphics* (Aug.), 559–568.

Kovar, L., Gleicher, M., and Pighin, F. 2002. Motion Graphs. *ACM Transactions on Graphics 21*, 3 (July), 473–482. (Proceedings of ACM SIGGRAPH 2002).

Laszlo, J., van de Panne, M., and Fiume, E. L. 2000. Interactive Control For Physically-Based Animation. *Proceedings of SIGGRAPH 2000* (July), 201–208.

LeCun, Y., and Huang, F. 2005. Loss Functions for Discriminative Training of Energy-Based Models. In *Proc. AIStats*.

Lee, J., Chai, J., Reitsma, P. S. A., Hodgins, J. K., and Pollard, N. S. 2002. Interactive Control of Avatars Animated With Human Motion Data. *ACM Transactions on Graphics 21*, 3 (July), 491–500. (Proceedings of ACM SIGGRAPH 2002).

Li, Y., Wang, T., and Shum, H.-Y. 2002. Motion Texture: A Two-Level Statistical Model for Character Motion Synthesis. *ACM Transactions on Graphics 21*, 3 (July), 465–472.

Liu, C. K., and Popović, Z. 2002. Synthesis of Complex Dynamic Character Motion from Simple Animations. *ACM Transactions on Graphics 21*, 3 (July), 408–416. Proceedings of ACM SIGGRAPH 2002.

Liu, Z., Gortler, S. J., and Cohen, M. F. 1994. Hierarchical spacetime control. In *Computer Graphics (SIGGRAPH 94 Proceedings)*, 35–42.

Mount, F. E., Whitmore, M., and Stealey, S. L. 2003. Evaluation of Neutral Body Posture on Shuttle Mission STS-57 (SPACEHAB-1). Tech. Rep. TM-2003-104805, NASA, Feb.

Neff, M., and Fiume, E. 2002. Modeling Tension and Relaxation for Computer Animation. In *ACM SIGGRAPH Symposium on Computer Animation*, 81–88.

Pandy, M. G. 2001. Computer Modeling and Simulation of Human Movement. *Annu. Rev. Biomed. Eng. 3*, 245–273.

Pearsall, D., Reid, J., and Ross, R. 1994. Inertial properties of the human trunk of males determined from magnetic resonance imaging. *Annals of Biomed. Eng. 22*, 692–706.

Pollard, N. S., and Reitsma, P. S. A. 2001. Animation of humanlike characters: Dynamic motion filtering with a physically plausible contact model. In *Yale Workshop on Adaptive and Learning Systems*.

Popović, Z., and Witkin, A. 1999. Physically Based Motion Transformation. *Proceedings of SIGGRAPH 99* (Aug.), 11–20.

Pullen, K., and Bregler, C. 2002. Motion Capture Assisted Animation: Texturing and Synthesis. *ACM Transactions on Graphics 21*, 3 (July), 501–508. Proceedings of ACM SIGGRAPH 2002.

Raibert, M. H., and Hodgins, J. K. 1991. Animation of dynamic legged locomotion. In *Computer Graphics (SIGGRAPH 91 Proceedings)*, vol. 25, 349–358.

Rose, C., Guenter, B., Bodenheimer, B., and Cohen, M. 1996. Efficient generation of motion transitions using spacetime constraints. In *Computer Graphics (SIGGRAPH 96 Proceedings)*, 147–154.

Rose, C., Cohen, M. F., and Bodenheimer, B. 1998. Verbs and Adverbs: Multidimensional Motion Interpolation. *IEEE Computer Graphics & Applications 18*, 5, 32–40.

Safonova, A., Hodgins, J. K., and Pollard, N. S. 2004. Synthesizing Physically Realistic Human Motion in Low-Dimensional Behavior-Specific Spaces. *ACM Transactions on Graphics* (Aug.).

Sun, H. C., and Metaxas, D. N. 2001. Automating gait animation. In *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, 261–270.

Tak, S., and Ko, H.-S. 2005. A physically-based motion retargeting filter. *ACM Trans. Graphics 24*, 1 (Jan.), 98–117.

Torkos, N., and van de Panne, M. 1998. Footprint-based Quadruped Motion Synthesis. In *Graphics Interface '98*, 151–160.

Unuma, M., Anjyo, K., and Takeuchi, R. 1995. Fourier Principles for Emotion-based Human Figure Animation. In *Proc. SIGGRAPH 95*, 91–96.

van de Panne, M., and Fiume, E. 1993. Sensor-actuator networks. In *Computer Graphics (SIGGRAPH 93 Proceedings)*, vol. 27, 335–342.

van de Panne, M., Kim, R., and Fiume, E. 1994. Virtual Wind-up Toys for Animation. *Graphics Interface '94* (May), 208–215. Held in Banff, Alberta, Canada.

Vasilescu, M. A. O. 2002. Human Motion Signatures: Analysis, Synthesis, Recognition. *Proc. ICPR '02 3* (Aug.), 456–460.

Witkin, A., and Kass, M. 1988. Spacetime constraints. In *Computer Graphics (SIGGRAPH 88 Proceedings)*, vol. 22, 159–168.

Witkin, A., and Popović, Z. 1995. Motion Warping. *Proceedings of SIGGRAPH 95* (Aug.), 105–108.