

# Generalizing Locomotion Style to New Animals With Inverse Optimal Regression

Kevin Wampler\*  
Adobe Research

Zoran Popović†  
University of Washington

Jovan Popović‡  
Adobe Research

## Abstract

We present a technique for analyzing a set of animal gaits to predict the gait of a new animal from its shape alone. This method works on a wide range of bipeds and quadrupeds, and adapts the motion style to the size and shape of the animal. We achieve this by combining inverse optimization with sparse data interpolation. Starting with a set of reference walking gaits extracted from sagittal plane video footage, we first use inverse optimization to learn physically motivated parameters describing the style of each of these gaits. Given a new animal, we estimate the parameters describing its gait with sparse data interpolation, then solve a forward optimization problem to synthesize the final gait. To improve the realism of the results, we introduce a novel algorithm called *joint inverse optimization* which learns coherent patterns in motion style from a database of example animal-gait pairs. We quantify the predictive performance of our model by comparing its synthesized gaits to ground truth walking motions for a range of different animals. We also apply our method to the prediction of gaits for dinosaurs and other extinct creatures.

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

**Keywords:** character animation, optimization, locomotion

**Links:**  DL  PDF

## 1 Introduction

The animals seen in nature come in a great variety of shapes and sizes, and move in a similarly diverse range of ways. Although the availability of video footage and motion capture data makes it easy to see how these animals locomote, natural curiosity compels us to wonder beyond this data to the motions of other creatures. How might animals which we don't have data for move? Can the gaits of living animals be used to guess at the gait for a dinosaur?

Our approach to gait synthesis takes a step toward answering these questions by using a database of real-world gaits captured from 2D sagittal plane video data to learn a model of how different animals walk. We can then synthesize physically valid (sagittally dominant) walking gaits for other animals according to the similarity of their shapes to the different animals in the database. This form of generalization is made difficult by the fact that many of its most

interesting applications are on to creatures (such as dinosaurs) with a shape substantially different than anything currently living. This necessitates a method which not only captures the salient aspects of the gaits for the animals for which we do have data, but which is also well-suited for generalizing these gaits to new creatures.

Our approach to addressing these issues is based on a combination of inverse optimization and sparse data interpolation, and in particular upon a novel algorithm called *joint inverse optimization* which unifies these two concepts. The term *joint* in this case does not refer to the animal's joints, but rather to the fact that this technique solves an inverse optimization problem jointly across multiple animals simultaneously. This method takes as input a database of animal gaits extracted from video footage, and processes it into a compact form describing the style of each gait. This style is captured with a set of biologically meaningful parameters describing, for instance, the relative preference for using different joints, the stiffness of these joints, and the preference for avoiding motions likely to cause the animal to trip. Unlike approaches such as [Liu et al. 2005] relying on traditional inverse optimization, our joint inverse optimization approach ensures that the learned parameters are well-suited to generalization onto new animals. Thus, instead of learning specific values of these parameters from a single motion, we learn coherent patterns between the parameters for an entire set of different motions. These coherent patterns in style are then used to guess the manner in which a new input animal should move. To the best of our knowledge this is the first method attempting to synthesize realistic motions across a wide range of both bipedal and quadrupedal animals. This allows the synthesis of visually plausible gaits for a wide range of extinct animals, and is validated to be more accurate than several potential alternative approaches at reproducing the gaits of a range of living animals for which we do have data.

## 2 Related Work

The synthesis of realistic legged locomotion is a well-studied but difficult problem in character animation. Although part of this difficulty is inherent in the synthesis of locomotion (realistic or otherwise) for complex characters, synthesizing *realistic* locomotion presents the particular challenge in that it requires a precise definition of what constitutes “realistic”. Although this problem can be addressed by relying on an artist or expert in the creation of the motion [Raibert and Hodgins 1991; Yin et al. 2007; Coros et al. 2010; Kry et al. 2009; de Lasa et al. 2010], the most common method for realistic locomotion synthesis is probably to rely on motion capture or video data as a guide. This use of real-world motion data is seen in methods which directly rearrange and replay existing motions [Bruderlin and Williams 1995; Witkin and Popović 1995; Lee et al. 2002; Arikian and Forsyth 2002; Kovar et al. 2002] or which take a set of related motions for a character and use these to create new motions for the same character [Safonova et al. 2004; Wei et al. 2011].

The reliance of many methods on pre-existing motion data poses a special problem when it comes to the synthesis of realistic locomotion for animals. Not only is it substantially harder to acquire motion data for animals than for humans, but there is also the more fundamental problem that acquiring motion data for extinct or fictional animals is impossible even in principle. Thus while some techniques have been developed to successfully synthesize animal

\*e-mail:kwampler@adobe.com

†e-mail:zoran@cs.washington.edu

‡e-mail:jovan@adobe.com

locomotion, most existing approaches either rely on motion data or artist interaction in specifying the style of the motions [Coros et al. 2011; Nunes et al. 2012], apply only to non-legged locomotion [Sims 1994; Wu and Popović 2003; Tan et al. 2011], or do not focus on generating realistic motions [Fang and Pollard 2003; Wampler and Popović 2009; Mordatch et al. 2012].

When it comes to the synthesis of *realistic* animal motions, one approach which has been taken is to increase the realism of the synthesized gaits by using a more faithful biomechanical model of an animal. This approach has been applied to humanoid bipedal locomotion by the modeling of soft tissue deformations in the feet [Jain and Liu 2011] or by including a more accurate biomechanical and metabolic modeling of leg muscles for humans [Wang et al. 2012; Mordatch et al. 2013] or general bipedal animals [Geijtenbeek et al. 2013]. Although these approaches have shown impressive results, our work differs in a few key ways. Firstly, while current approaches have only been shown to work on bipedal characters and do not focus on animals with highly different sizes, our approach can be applied equally to both bipeds and quadrupeds and gives plausible results on a very wide range of animals (covering factor of ten difference in height or factor of two hundred difference in mass) with no per-animal tuning. Secondly, our approach is simpler to apply to the synthesis of new animals since it does not require any specification of the system of muscles the animal uses to locomote. This makes, for instance, the addition of a tail or arms in order to synthesize a dinosaur gait a trivial task. Still, this flexibility comes at a cost in that our method requires a database of real-world animal motions as input to a preprocessing step. Once this data is collected and processed, however, our method can synthesize gaits for a wide range of new animals without any additional input other than the shapes of these animals.

On the other hand, one might attempt to synthesize an animal’s gait by forgoing any consideration of the principles underlying the motion and instead attempting to directly match the appearance of motions for which data does exist. The approach has the potential to work well when one has data for animals with shapes similar to the one to be synthesized, but it is not clear how well this sort of approach can work when this is not the case. Indeed, the primary successful application of this approach to creatures with diverse shapes has thus far been limited to highly stylized motions [Hecker et al. 2008]. Unfortunately, these cases where the animal is relatively unlike anything one could obtain data for cover some of the most interesting applications of computational locomotion synthesis. A restriction to synthesizing gaits only for animals similar to those that exist would preclude the generation of gaits for most dinosaurs and many other extinct animals.

Our method (first described in the dissertation [Wampler 2012]) takes a middle ground and combines data-based interpolation with biologically and physically motivated factors. An instance of this type of ‘middle ground’ approach has been previously applied to human locomotion [Liu et al. 2005; Lee and Popović 2010] by learning the passive actuation characteristics at a character’s joints from a sequence of motion capture and then using this information to create new motions for the same character. Although these passive actuation characteristics are relatively simple biologically motivated entities approximating the nature of tendons and ligaments, their particular values were derived from a sequence of motion capture data. Our method is in the same spirit of these approaches, but applies to the analysis of a *set* of gaits for animals with widely varying shapes. The primary complication that arises in doing so is that in order to synthesize motions for new creatures, the parameters used to represent the style of each animal’s gait must be suited to interpolation on to animals for which there is no motion capture data available.

### 3 Algorithm Overview

Our approach to gait synthesis rests upon three primary components. As input we require a *motion database*, and a *generative model*. Given these inputs, we then employ an algorithm for tuning the generative model to agree with the motion database which we call *joint inverse optimization*. The motion database consists of a set of pairings  $(A_1, M_1), \dots, (A_n, M_n)$ , each of which associates the shape of an animal  $A_i$  with its ground-truth gait  $M_i$  as extracted from real-world video data. While this motion database covers a set of known animal motions, the synthesis of new motions is handled by a generative model, denoted  $f$ . This generative model takes as input an animal  $A$  and a vector of parameters  $\phi$  describing the style of the motion to be synthesized, and outputs an associated gait  $f(A, \phi)$ . The goal of our gait synthesis technique can be summarized as follows: given a new animal  $A_{\text{new}}$ , find the parameters  $\phi_{\text{new}}$  of the generative model such that the style of the resulting motion  $f(A_{\text{new}}, \phi_{\text{new}})$  matches what would be expected given the motion database.

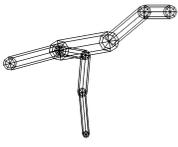
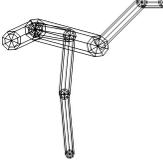
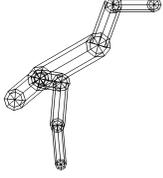
Intuitively this approach mimics that which an artist might take when animating an animal for which they do not have any video footage – try to guess the the motion for the new animal by appealing to animals for which the artist *does* have video footage. Synthesizing locomotion in this manner requires that several sub-problems be addressed:

**Motion Database Creation** The motion database contains ground-truth motions for a wide range of different animals. This database is created by tracking a set of points in a real-world sagittal plane video of the animal walking, then solving a spacetime constraints optimization to fit a physically realistic cyclic gait to the motion of these points.

**Generative Model** The generative model  $f(A, \phi)$  is used to synthesize new motions. In keeping with many existing locomotion synthesis techniques and literature on the underlying principles of animal locomotion [Alexander 1996], we base the generative model on an optimization. That is, given an animal  $A$ , the motion  $f(A, \phi)$  is chosen so as to minimize some objective function which is itself parameterized by  $\phi$ . A key ingredient in the definition of  $f$  lies in choosing a set of parameters  $\phi$  which is expressive enough to capture the variations in style between the different animals in the motion database.

**Joint Inverse Optimization** The core of our algorithm uses the motion database and the generative model to synthesize gaits for new animals. This is achieved by estimating a  $\phi_i$  for each  $(A_i, M_i)$  in the motion database, and performing sparse data interpolation on these  $\phi_i$  values to predict what  $\phi$  should be for a new input animal. We do this by solving an inverse optimization problem *jointly* across all of the animals in the motion database. This optimization estimates a vector of parameters  $\phi_i$  for each  $(A_i, M_i)$  in the motion database such that  $f(A_i, \phi_i) \approx M_i$  and such that the  $\phi_i$  parameters are well-suited for generalization onto new animals by a simple sparse data interpolation technique.

With these three components in place, a motion for a new animal  $A_{\text{new}}$  can be synthesized by first interpolating  $\phi_{\text{new}}$  from the  $\phi_i$  parameters associated with similar animals in the motion database, then solving for the final motion with  $f(A_{\text{new}}, \phi_{\text{new}})$ . The following three sections will cover each of these components in further detail.

greater flamingo  1.1m 3kg	ostrich  2.3m 94.5kg	secretary bird  0.8m 3.3kg
greater roadrunner  0.22m 0.3kg	emu  1.5m 30.0kg	southern ground hornbill  0.9m 3.6kg

**Figure 1:** The six bipeds in the motion database along with their associated heights and masses.

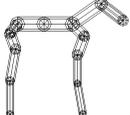
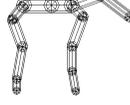
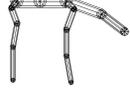
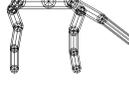
## 4 Motion Database Creation

The motion database is used to define a mapping from the shape of an animal to the way that animal moves in the real world. Each entry associates a particular animal  $A_i$  with a cyclic gait  $M_i$  prescribing the animal’s ground truth walking motion. Although in principle our approach is applicable to non-walking gaits, all of the gaits in a motion database should be of the same type (i.e. all walks, or all runs, etc). As a practical matter, we have focused only on sagittally dominant walks due to the relative ease of obtaining 2D sagittal plane video data of walks for a wide range of different animals, but expect our approach to be applicable to 3D motion data with little modification. In synthesizing a gait for a new animal (described later in section 6) this motion database is used as a reference to estimate what parameters of the generative model are likely to result in a realistic motion for a new animal.

An animal  $A$  is represented as a kinematic tree of limbs connected by joints. Each joint describes a parameterized rotation from its parent limb to its child limb, and each limb has an associated length and mass. In addition, each animal’s representation marks the limb corresponding to the head and the position of each foot. A *pose* for an animal is described by a vector giving the rotational parameters of each of its joints, the global translation and rotation of the animal, and the ground reaction forces and torques at each foot in contact with the ground. A sequence of such poses concatenated into a single vector forms a motion  $M$ , and the pose associated with a particular frame at time  $t$  in  $M$  is denoted  $M(t)$ .

Although an ideal motion database would be constructed using 3D motion capture and force plates, this sort of data is currently difficult to obtain for a wide range of different animals. Instead, each of the  $M_i$  motions in our motion database is created by fitting a cyclic 3D motion to standard 2D sagittal plane video footage of the animal walking. We have obtained this data from online video sharing sites such as YouTube and Flickr Video, as they represent the most easily accessible resource for such video footage. Each video consists of a side-on view of the animal walking with only rotational camera motion so as to avoid parallax.

Our motion database consists of six bipeds and twelve quadrupeds spanning a wide range of animal shapes and sizes as shown in fig-

horse  1.55m 500kg	Thomson’s gazelle  0.55m 20kg	pronghorn antelope  0.9m 50kg	bison  1.7m 700kg
house cat  0.24m 4.5kg	cheetah  0.85m 50kg	elephant  3.7m 5200kg	giraffe  5.5m 1000kg
moose  1.8m 320kg	rhinoceros  1.6m 1600kg	steenbok  0.5m 17kg	tiger  1.0m 180kg

**Figure 2:** The twelve quadrupeds in the motion database along with their associated heights and masses.

ures 1 and 2. For each of these animals we determine the relative lengths and masses of the animal’s limbs by manually constructing a crude 3D model from a selected frame in the video, then uniformly scaling each limb’s length and mass so that the animal’s height and total mass match the values in figures 1 and 2. For each video we also remove the effects of any camera motion by translationally stabilizing based on SURF features [Bay et al. 2008] tracked between the frames in the video.

Next, a number of points on the animal are manually tracked across the video. These points vary somewhat from animal to animal, but always include at least two points on the torso, at least one on the head, as well as points on the knees and feet. Each of these points gives a 2D trace of some point on the animal tracked through each frame in the video. A sub-segment of the frames in the video representing a single gait cycle is also selected. In order to match each 2D point trace with the motion of a 3D point on the animal, a few frames are chosen on which a 3D model for the animal is superimposed over the video. Using this superposition, each 2D point is attached to one of the animal’s limbs, allowing coordinates for each point to be determined in a coordinate system local to the point’s associated limb. The local coordinates of these points combined with their associated 2D trajectories from video form the input required to solve a fitting problem matching a physically accurate 3D gait to the motions in the video.

The final step is to solve for a cyclic gait  $M$  for each animal  $A$  such that the motion of this gait matches that of the 2D point traces extracted from the animal’s associated video. For a given frame at time  $t_i$  in the video let  $d_j(t_i)$  denote the 2D position of the  $j$ th point in the extracted traces, and let  $p_j(M(t_i))$  denote the 2D X-Y projection of the associated point on the animal in a motion  $M$ .

If  $M$  were a perfect fit to the data then it would hold that  $d_j(t_i) = p_j(M(t_i))$  for each frame  $i$  and point  $j$ . Since in general a perfect fit to the data will not be possible, the degree to which a motion  $M$  matches the data is quantified independently at each frame with a sum of squared distances between each point in the traces extracted from video and the associated point on the animal in the

pose  $M(t_i)$ :

$$\text{fit}(M(t_i)) = \sum_j w_j \|d_j(t_i) - p_j(M(t_i))\|_2^2 \quad (1)$$

Here  $w_j$  is a weight assigned to each point on the animal to give preference to a more accurate fit of the knee and foot positions (weight of 3) while allowing greater errors in the fit of points on the torso (weight of 0.5), all other points having weight 1.0.

The motion  $M$  used to represent the ground truth gait for the animal in the motion database is then solved for with a spacetime constraints optimization. This generates a 3D gait closely matching the data to as the solution of the nonlinear programming problem:

$$\begin{aligned} M = \operatorname{argmin} \sum_i [\text{fit}(M(t_i)) + \alpha \cdot \text{cost}(M(t_i))] \\ \text{s.t. } g_{k_{eq}}(M(t_i)) = 0 & \quad \forall k_{eq}, i \\ h_{k_{ieq}}(M(t_i)) \leq 0 & \quad \forall k_{ieq}, i \end{aligned} \quad (2)$$

where  $\text{cost}(M(t_i))$ ,  $g_{k_{eq}}(M(t_i))$ , and  $h_{k_{ieq}}(M(t_i))$  are all defined to match the objective, equality, and inequality constraints respectively described in [Wampler and Popović 2009]. This definition of  $\text{cost}(M(t_i))$  penalizes high-energy motions, while  $g_{k_{eq}}(M(t_i))$ , and  $h_{k_{ieq}}(M(t_i))$  enforce the laws of Newtonian mechanics. Some contribution of  $\text{cost}(M(t_i))$  is included in the objective as a regularization term and helps to avoid overfitting to the data. The term  $\alpha$  controls how strongly minimizing  $\text{cost}(M(t_i))$  is weighted versus minimizing the data fit error. Larger values of  $\alpha$  tend to lead to smoother motions at the cost of a poorer data fit. Although  $\alpha$  is occasionally altered on a per-animal basis, a value of  $10^{-5}$  works well for most of the animals in the database, with other values ranging from  $10^{-3}$  to  $10^{-8}$ . In a few cases we were unable to choose a value for  $\alpha$  which avoids both oversmoothing and ‘glitches’ from undersmoothing. In these situations, we prefer a slightly undersmoothed motion since the inverse optimization and regression steps employed later will introduce a small amount of additional smoothing.

## 5 Generative Model

While the motion database represents the gaits of a set of animals as extracted from video footage, the synthesis of new gaits is handled by a generative model  $f$ , taking as input the shape of an animal  $A$  and a vector  $\phi$  parameterizing the style of the gait to be generated. This generative model is capable of synthesizing gaits for animals with greatly different shapes and sizes, as well as for extinct or imaginary creatures. The remainder of this section will be concerned with the definition of  $f$  and  $\phi$ , while section 6 will address which particular value of  $\phi$  should be used for a given input animal.

A good generative model should be flexible enough so that for any of a wide range of different input animals, it is possible to synthesize a realistic gait of that animal by an appropriate choice of  $\phi$ . More concretely, for each  $A_i$  and  $M_i$  in the motion database there should be some  $\phi_i$  such that  $M_i \approx f(A_i, \phi_i)$ . In addition, when synthesizing motions for new animals it will be useful for the values of  $\phi$  to be both relatively compact and coherently interpolable between different animals.

Our generative model  $f$  is based on the spacetime constraints optimization described by [Wampler and Popović 2009] because of its ability to automatically and relatively quickly synthesize motions for a wide range of different animals. In this approach, a motion is

synthesized by finding the minimum of a large nonlinear programming problem:

$$\begin{aligned} f(A, \phi) = \operatorname{argmin}_M \sum_i \text{cost}(M(t_i), \phi) \\ \text{s.t. } g_{k_{eq}}(M(t_i)) = 0 & \quad \forall k_{eq}, i \\ h_{k_{ieq}}(M(t_i)) \leq 0 & \quad \forall k_{ieq}, i \end{aligned} \quad (3)$$

Where  $g_{k_{eq}}$  and  $h_{k_{ieq}}$  are again the equality and inequality constraints enforcing the physical validity of the resulting motion as described by [Wampler and Popović 2009]. In our implementation, this optimization is initialized with the animal in a default stand-still pose and is solved with the SNOPT nonlinear programming library [Gill et al. 2005].

The objective function used in equation 3 dictates which sorts of gaits should be preferred over others, and thus altering this objective function allows different gaits to be synthesized for the same animal. In previous work by [Liu et al. 2005] this idea has been used to synthesize different styles of human locomotion by choosing  $\phi$  to represent to passive actuation characteristics at the character’s joints. For our task we extend this set to include parameters describing the strength and coactuation of different joints, the uncertainty of interaction with the ground, and the relative preference for low-torque, low-impact, and smooth motions.

### 5.1 Generative parameters

The elements of  $\phi$  serving as parameters to our generative model are listed in table 1 and are used to alter the per-frame objective function  $\text{cost}(M(t_i), \phi)$  in equation 3. To facilitate easily spotting these parameters in the following equations, all variables representing elements in  $\phi$  have been colored **dark blue**. Adjusting the value of  $\phi$  thus provides a means by which the style of a synthesized gait can be controlled. Our choice of generative parameters and the associated objective function were found by starting with an objective function based purely on torque-minimization, and iteratively adding terms until the gaits of the animals within the motion database could be reproduced. The result is a combination of six terms:

$$\begin{aligned} \text{cost}(M(t_i), \phi) = & \text{cost}_{\text{torque}}(M(t_i), \phi) \\ & + w_f \text{cost}_{\text{force}}(M(t_i), \phi) \\ & + w_a \text{cost}_{\text{smooth}}(M(t_i), \phi) \\ & + (e^{w_h} - 1) \text{cost}_{\text{head}}(M(t_i), \phi) \\ & + \text{cost}_{\text{coactuate}}(M(t_i), \phi) \\ & + (e^{w_g} - 1) \text{cost}_{\text{ground}}(M(t_i), \phi) \end{aligned} \quad (4)$$

There are several sub-quantities which must be used to calculate these components, all of which are computed as a function of  $M(t_i)$ . For notational cleanliness, however, we will leave this dependence on  $M(t_i)$  implicit and write for instance  $\tau$  instead of the more explicit  $\tau(M(t_i))$ . Accordingly,  $\tau$ ,  $\mathbf{f}$ , and  $\mathbf{q}$  will respectively represent a vector of the concatenated torques, forces, and joint rotation angles at each of the character’s joints at the frame  $M(t_i)$ . The quantities  $\mathbf{R}_h$ ,  $\mathbf{p}_h$ , and  $\ddot{\theta}_h$  will refer to the rotation matrix, position, and angular acceleration for the limb representing the animal’s head as measured about its center of mass. Similarly,  $\mathbf{p}_f$  represents the position of a foot. Any temporal derivatives represented by a superscript dot are computed using finite differences.

The first three components of the objective function in equation 4 penalize torques, forces, and angular accelerations at the animal’s

name	biped	description
$h_g$	✓	falloff of ground uncertainty distribution
$w_g$	✓	weight for vertical ground penalties
$w_v$	✓	weight for horizontal ground penalties
$h_w$	✓	maximum height of near-ground drag
$d_w$	✓	drag coefficient for near-ground drag
$c_l$	✓	weight on knee-ankle coactuation
$r_l$	✓	target ratio for knee vs. ankle velocities
$c_a$		weight on elbow-wrist coactuation
$r_a$		target ratio for elbow vs. wrist velocities
$t_h$	✓	scaling for torques exerted at the hip
$t_k$	✓	scaling for torques exerted at the knee
$t_a$	✓	scaling for torques exerted at the ankle
$t_s$		scaling for torques exerted at the shoulder
$t_e$		scaling for torques exerted at the elbow
$t_w$		scaling for torques exerted at the wrist
$w_f$	✓	scaling for joint-force penalties
$w_a$	✓	scaling for joint-acceleration penalties
$w_h$	✓	scaling for head stability penalties
$k_h$	✓	spring constant for hip
$\bar{q}_h$	✓	spring rest angle for hip
$d_h$	✓	damper coefficient for hip
$k_k$	✓	spring constant for knee
$\bar{q}_k$	✓	spring rest angle for knee
$d_k$	✓	damper coefficient for knee
$k_a$	✓	spring constant for ankle
$\bar{q}_a$	✓	spring rest angle for ankle
$d_a$	✓	damper coefficient for ankle
$k_s$		spring constant for shoulder
$\bar{q}_s$		spring rest angle for shoulder
$d_s$		damper coefficient for shoulder
$k_e$		spring constant for elbow
$\bar{q}_e$		spring rest angle for elbow
$d_e$		damper coefficient for elbow
$k_w$		spring constant for wrist
$\bar{q}_w$		spring rest angle for wrist
$d_w$		damper coefficient for wrist

**Table 1:** A table of the inverse parameters used to specify the style of an animal’s gait. Quadrupedal animals make use of the whole set of parameters, while bipeds make use only of those marked.

joints:

$$\text{cost}_{\text{torque}}(M(t_i), \phi) = \mathbf{w}_j \cdot \boldsymbol{\tau} \quad (5)$$

$$\text{cost}_{\text{force}}(M(t_i), \phi) = \mathbf{w}_j \cdot \mathbf{f} \quad (6)$$

$$\text{cost}_{\text{smooth}}(M(t_i), \phi) = \|\dot{\mathbf{q}}\|^2 \quad (7)$$

where  $\mathbf{w}_j$  is a vector of weights with elements equal to  $e^{t_h} - 1$ ,  $e^{t_k} - 1$ ,  $e^{t_a} - 1$ ,  $e^{t_s} - 1$ ,  $e^{t_e} - 1$ , or  $e^{t_w} - 1$  at indices corresponding to the hip, knee, ankle, shoulder, elbow or wrist joints respectively. All other elements in  $\mathbf{w}_j$  are fixed equal to 1. Lower settings of an element in  $\mathbf{w}$  approximates the effect of ‘stronger’ joints, which are actuated by stronger muscles and better able to withstand large forces.

The term  $\text{cost}_{\text{head}}(M(t_i), \phi)$  penalizes motion in the animal’s head and is computed as the sum of four sub-terms respectively penalizing the rotation of the head away from forward, its velocity, linear acceleration, and angular acceleration:

$$\text{cost}_{\text{head}}(M(t_i), \phi) = 100\|\mathbf{R}_h - \mathbf{I}\|_2^2 + 2.5\|\dot{\mathbf{p}}_{h_{yz}}\|_2^2 + \|\ddot{\mathbf{p}}_h\|_2^2 + \|\ddot{\theta}_h\|_2^2 \quad (8)$$

The weights 100 and 2.5 scaling the first two sub-terms where chosen empirically. Although  $\text{cost}_{\text{head}}(M(t_i), \phi)$  does not depend on

$\phi$ , the strength with which it factors into equation 4 is scaled by  $(e^{w_h} - 1)$ . The term  $\dot{\mathbf{p}}_{h_{yz}}$  represents the motion of the head in the vertical and lateral directions. The function  $\text{cost}_{\text{head}}$  is useful in modeling the fact that animals often attempt to stabilize their head motion to help with visual perception [MacIver et al. 2010].

The standard spacetime constraints formulation used to synthesize motions treats each joint in the animal as being capable of moving entirely independently of all the other joints. In reality, however, some pairs of joints exhibit a tendency to be *coactuated* such that their motions occur in concert rather than independently. Rather than directly modeling the muscles responsible for this as in [Wang et al. 2012; Mordatch et al. 2013; Geijtenbeek et al. 2013], we take a simplified approach where  $\text{cost}_{\text{coactuate}}(M(t_i), \phi)$  penalizes deviations of the relative velocities of the animal’s knee and ankle joints from  $r_l$ :

$$(e^{c_l} - 1)(r_l \dot{\mathbf{q}}_k - \dot{\mathbf{q}}_a)^2 \quad (9)$$

Where  $\dot{\mathbf{q}}_k$  and  $\dot{\mathbf{q}}_a$  are the rotational velocities at the knee and ankle respectively. For a quadruped the parameters  $c_a$  and  $r_a$  are used to add an analogous additional penalty related to the relative velocities of the elbow and wrist joints.

In the optimization defined by equation 3, the animal is implicitly assumed to have perfect knowledge of its environment. In reality this is of course not the case, and in order to avoid tripping an animal will often lift its feet higher than is strictly energetically optimal. We use the term  $\text{cost}_{\text{ground}}(M(t_i), \phi)$  to penalize motions where the animal’s foot moves quickly while too close to the ground as:

$$\text{cost}_{\text{ground}}(M(t_i), \phi) = \sum_f p_{c_x} \|\dot{\mathbf{p}}_{f_{xz}}\|_2 + \max\{0, -p_{c_y} \dot{\mathbf{p}}_{f_y}\} \quad (10)$$

where the sum is taken over each of the animal’s feet and  $\dot{\mathbf{p}}_{f_{xz}}$  and  $\dot{\mathbf{p}}_{f_y}$  represent the horizontal and vertical components of the foot’s velocity. The values  $p_{c_x}$  and  $p_{c_y}$  approximate the probability of an ‘unexpected’ contact between the foot and the ground due to the foot’s horizontal and vertical motion respectively, calculated as:

$$p_{c_x} = w_v(1 - e^{-h_g \mathbf{p}_{f_y}}) \quad (11)$$

$$p_{c_y} = \frac{e^{-h_g \mathbf{p}_{f_y}^{t+1}} - e^{-h_g \mathbf{p}_{f_y}}}{1 - e^{-h_g \mathbf{p}_{f_y}}} \quad (12)$$

The value  $\mathbf{p}_f^{t+1}$  indicates the position of the foot in the next frame in the motion, i.e.  $\mathbf{p}_f(M(t_{i+1}))$ .

In addition to the term  $\text{cost}_{\text{ground}}(M(t_i), \phi)$  directly penalizing motions where the foot skirts too close to the ground, we also indirectly penalize such motions by approximating the resistive forces on the leg resulting from walking through shallow water or near-ground vegetation. For each foot  $f$ , we add a force  $\mathbf{f}$  resisting the foot’s velocity based on the speed of the foot and its depth below the height defined by  $h_w$ :

$$\mathbf{f} = -\dot{\mathbf{p}}_f \|\dot{\mathbf{p}}_f\|_2 \max\{0, h_w - \mathbf{p}_{f_y}\} e^{d_w} \quad (13)$$

In practice, we have found the effects of this near-ground resistance are useful in conjunction with those from  $\text{cost}_{\text{ground}}(M(t_i), \phi)$  to shape the trajectory of the animal’s feet during their air phases.

Finally, we include a number of parameters modeling the passive actuation characteristics of the animal’s leg joints. For each of the hip, knee, ankle, shoulder, elbow, and wrist joints we include three parameters specifying a spring rest length, spring constant, and damper coefficient for the joint. The use of these parameters is identical to that in [Liu et al. 2005], where they were successfully used to capture stylistic variations in human locomotion.

## 6 Joint Inverse Optimization

The motion database provides a reference for the real-world gaits of a number of different animals, but to create gaits for new animals it is necessary to generalize beyond those contained within the motion database. While the generative model  $f(A, \phi)$  described in section 5 is capable of synthesizing new motions given an animal  $A$  and a vector of parameters  $\phi$ , it remains to be determined which value of  $\phi$  is likely to lead to a realistic motion for a given animal.

We achieve this by fitting a vector  $\phi_i$  of parameters to each  $M_i$  in the motion database, then interpolating between these parameters to estimate  $\phi$  for a new animal. Although reasonable results can sometimes be obtained by fitting each of the  $\phi_i$  independently, we find that interpolation between the  $\phi_i$  found in this manner is often difficult. Instead, we propose a new algorithm termed *joint inverse optimization* which jointly learns all of the  $\phi_i$  together. The resulting parameters are constructed to better allow  $\phi$  to be determined for a new animal by simple interpolation. In the remainder of this section we will first look more closely at the naïve case of independently fitting each  $\phi_i$ . We will then introduce our algorithm of joint inverse optimization, followed by details on its implementation.

### 6.1 Independent Inverse Optimization

As its name implies, joint inverse optimization is based on the approach of *inverse optimization* that has been previously employed in character animation [Liu et al. 2005; Lee and Popović 2010]. One possible formulation of an inverse optimization problem as applied to animal gaits would determine the  $\phi_i$  parameters for each  $(A_i, M_i)$  in the motion database by solving:

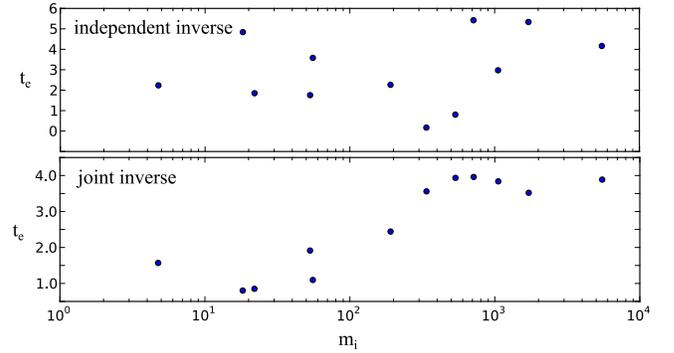
$$\phi_i = \operatorname{argmin}_{\phi} D(f(A_i, \phi), M_i) \quad (14)$$

Where the function  $D(M_a, M_b)$  defines a distance metric representing in the error in how closely the motion  $M_a$  matches  $M_b$  (see section 6.2.1 for the specifics of the distance function used in our implementation). Essentially, for each animal  $A_i$  this optimization solves for the vector of parameters  $\phi_i$  such that the motion  $f(A_i, \phi_i)$  synthesized by the generative model with these parameters best matches the ground truth motion  $M_i$  for the animal.

The problem with using a standard inverse optimization formulation to determine the  $\phi_i$  parameters for each entry in the motion database comes when trying to estimate the  $\phi_{\text{new}}$  which will lead to a realistic motion for a new animal. In particular, since equation 14 solves for each  $\phi_i$  independently, there is no guarantee of any consistency of these parameters between different animals. Indeed, as illustrated in figure 3 the  $\phi_i$  parameters found using this method do not form any easily describable coherent pattern. In general this makes the use of sparse data interpolation to determine which  $\phi_{\text{new}}$  should be used to synthesize a gait for a new animal  $A_{\text{new}}$  problematic, and better results can be obtained by incorporating a notion of sparse data interpolation directly into the inverse optimization problem.

### 6.2 Joint Inverse Optimization

Because the end goal of our approach is to use the examples in the motion database to determine how a new animal should move, our technique of joint inverse optimization explicitly incorporates this requirement into its formulation. The result is an optimization which not only attempts to fit each  $\phi_i$  with its associated motion  $M_i$ , but which also minimizes a term ensuring that the result is



**Figure 3:** Plots of the  $t_e$  parameter (see table 1) for the quadrupeds in the motion database against the log of the animal’s mass. The parameters found by joint inverse optimization are better suited to regression. Note that distances along the x-axis are only an approximation to the similarity of the animals as measured by equation 21, so a perfectly smooth curve should not be expected in the lower plot.

well-suited to sparse data interpolation:

$$\operatorname{argmin}_{\theta, \phi_1, \dots, \phi_n} \sum_i [D(f(A_i, \phi_i), M_i) + \beta (\|\phi_i - R(A_i, \theta)\|_2^2 + r(\theta))] \quad (15)$$

This formulation differs from traditional inverse optimization (equation 14) by the addition of two functions: A *regression function* and a *regularization function*. The regression function  $R(A, \theta)$  takes as input an animal  $A$  (not necessarily in the motion database) and a vector of *regression parameters*  $\theta$ , and returns a vector of generative parameters  $\phi$  suitable for using to synthesize a gait for  $A$ . The regularization function  $r(\theta)$  is then used in preventing an overfitting of the regression parameters  $\theta$ . It is the vector of regression parameters  $\theta$  that determines the motion of a new animal  $A_{\text{new}}$ , by first calculating  $\phi_{\text{new}} = R(A_{\text{new}}, \theta)$ , then solving for the final motion with  $f(A_{\text{new}}, \phi_{\text{new}})$ .

Intuitively, the goal of a joint inverse optimization is to find a value of the regression parameters  $\theta$  which simultaneously accurately reproduces each motion in the motion database via minimizing each  $D(f(A_i, R(A_i, \theta)), M_i)$ , and which avoids overfitting by minimizing  $r(\theta)$ . Unfortunately, the optimization resulting from attempting to directly minimize this quantity is extremely brittle due to the compounding factor that the generative function  $f$  is *not* guaranteed to converge to a physically valid result, and thus for many values of  $\theta$  there are some animals in the motion database for which  $D(f(A_i, R(A_i, \theta)), M_i)$  is impossible to evaluate.

The formulation for joint inverse optimization given in equation 15 provides a more robust and efficient approach. In this approach, the  $\phi_i$  parameters used to solve for an animal’s motion are allowed to differ from the parameters resulting from the regression function  $R(A_i, \theta)$ . In essence, although the  $\phi$  parameters used to synthesize a gait for a new animal will be found via  $\phi_{\text{new}} = R(A, \theta)$ , this is treated as a soft constraint for the purposes for fitting  $\theta$  to the motion database. We also note that in the case where the motion database contains only a single animal, equation 15 reduces to standard inverse optimization (equation 14) for most sensible choices of  $R$  and  $r$ , including ours described in section 6.2.2.

When accounting for the possibility that  $f(A_i, \phi_i)$  might fail to converge to a physically valid result, note that in equation 15 the only term which relies on the result of the generative model is  $D(f(A_i, \phi_i), M_i)$ . The fact that this term can be computed in-

dependently for each animal makes it substantially easier to create an optimization which solves equation 15 by simply discarding any failed evaluations of  $f(A_i, \phi_i)$ . Our approach for achieving this (including the values used for the term  $\beta$ ) is described in section 6.2.3, but we will first provide the specific definitions of  $D$ ,  $R$ , and  $r$  which we employ.

### 6.2.1 Gait distance metric

Solving the joint inverse optimization problem defined by equation 15 requires a definition of  $D(M_a, M_b)$ , describing a distance metric between different gaits. Although a simple sum-of-squared differences in the rotational degrees of freedom of the animal's joints over the course of the two gaits gives reasonable results, we have found that more visually pleasing results can be obtained with a slightly more involved definition. Intuitively, this is because some aspects of a motion, such as the height by which the feet are raised or the stability of the head, are more visually important than the precise angles of rotation for the animal's joints. We compute  $D(M_a, M_b)$  as a sum of per-frame costs, each computed as:

$$D(M_a(t_i), M_b(t_i)) = \|\mathbf{0.25} \mathbf{w} \cdot (\mathbf{q}(M_b(t_i)) - \mathbf{q}(M_a(t_i)))\|_2^2 + \quad (16)$$

$$\|\mathbf{0.6} \mathbf{w} \cdot (\dot{\mathbf{q}}(M_b(t_i)) - \dot{\mathbf{q}}(M_a(t_i)))\|_2^2 + \quad (17)$$

$$\|\mathbf{0.5} \mathbf{w} \cdot (\ddot{\mathbf{q}}(M_b(t_i)) - \ddot{\mathbf{q}}(M_a(t_i)))\|_2^2 + \quad (18)$$

$$2.25 \sum_f (\mathbf{p}_f(M_b(t_i))_y - \mathbf{p}_f(M_a(t_i))_y)^2 + \quad (19)$$

$$3.25 \sum_{h \in \{h_1, h_2\}} \|\ddot{\mathbf{p}}_h(M_b(t_i)) - \ddot{\mathbf{p}}_h(M_a(t_i))\|_2^2 + \quad (20)$$

Where  $\mathbf{q}(M(t_i))$  is a vector of the rotational degrees of freedom in  $M$  at frame  $i$ , and  $\mathbf{p}_{h_1}(M(t_i))$ ,  $\mathbf{p}_{h_2}(M(t_i))$  give the position of the front and back of the head at frame  $i$ . Similarly, for each foot  $f$ ,  $\mathbf{p}_f(M(t_i))$  gives the position of the foot at frame  $i$ . The vector  $\mathbf{w}$  is used to scale the weight given to differences in the rotations at different joints. For bipeds  $\mathbf{w}$  is 2 for the knee and 3 for ankle joints while for quadrupeds  $\mathbf{w}$  is 3 for the knee/elbow joints and 10 for the ankle/wrist joints. All other elements in  $\mathbf{w}$  are set to 1.

### 6.2.2 Regression and Regularization Functions

A joint inverse optimization additionally requires definitions for the regression and regularization functions  $R(A_i, \theta)$  and  $r(\theta)$ . We initially experimented with a regression function which estimated parameters of physically-motivated scaling functions in the spirit of [Hodgins and Pollard 1997]. We were not able to achieve good results with this approach, and instead settled on a relatively simple formulation based on regression with radial basis functions and regularization with quadratic smoothing [Boyd and Vandenberghe 2004]. This approach employs a definition of  $\theta$  which concatenates  $\theta_1, \dots, \theta_n$  for each of the  $n$  animals in the motion database, where each  $\theta_i$  has the same dimension as the generative parameters  $\phi_i$ .

The distance metric underlying the radial basis function interpolation measures the dissimilarity between two animals  $A_a$  and  $A_b$  with a weighted combination of the difference of the log-masses of the two animals and the difference in the lengths of their various limbs normalized by the total size of each animal:

$$d(A_a, A_b)^2 = \sum_i \left( \frac{l_{a_i}}{\sum_j l_{a_j}} - \frac{l_{b_i}}{\sum_j l_{b_j}} \right)^2 + 0.002(\ln(m_a) - \ln(m_b))^2 \quad (21)$$

where  $l_{a_j}$  and  $l_{b_j}$  are the lengths of the  $j$ th limbs in  $A_a$  and  $A_b$  respectively, and  $m_a$  and  $m_b$  are the total respective masses of  $A_a$  and  $A_b$ . The logarithmic scaling of the masses captures the fact that, for instance, a 10kg difference in mass is much more meaningful between a 10kg and a 20kg animal than between a 1000kg and a 1010kg animal.

Using this distance metric, the regression function is then defined in a manner similar to [Zhang et al. 2004] as:

$$R(A, \theta) = \frac{\sum_i \theta_i e^{-d_i^2}}{\sum_i e^{-d_i^2}} \quad (22)$$

where  $d_i = \frac{d(A, A_i)}{d_{min}}$  with  $d_{min} = \min_i d(A, A_i)$ . Here  $\theta_i$  are the regression parameters directly associated with  $A_i$  (which in general need not be equal to  $\phi_i$  due the decoupling of  $\theta$  from  $\phi_1, \dots, \phi_n$  in equation 15).

To avoid overfitting, we employ a regularization function based on quadratic smoothing:

$$r(\theta) = \sum_i \|R(A_i, \theta') - \theta_i\|_2^2 \quad (23)$$

where  $\theta'$  are the regression parameters omitting the  $\theta_i$  parameters associated with  $A_i$ , so this regularization function essentially computes a sum of leave-one-out errors.

Finally, the period  $p_i$  of an animal's gait cycle and the speed  $v_i$  at which it should move are determined with a separate method by:

$$p_i = \alpha_p m_i^{\beta_p} \quad (24)$$

$$v_i = \alpha_v l_{leg_i}^{\beta_v} \quad (25)$$

In these equations  $m_i$  is the total mass of  $A_i$  and  $l_{leg_i}$  is the average length of the legs of  $A_i$ . Additionally,  $\alpha_p$  and  $\beta_p$  are parameters used to model the relationship between an animal's mass and the period of its gait cycle, while  $\alpha_v$ , and  $\beta_v$  are parameters modeling relationship between the length of the animal's legs and its speed. The parameters  $\alpha_p$ ,  $\beta_p$ ,  $\alpha_v$ , and  $\beta_v$  are found from the motion database using a least-squares fit. Because the animals in our motion database do not exhibit significant differences in the timings for their foot contacts once normalized for period and speed, we fix the relative timings for the foot contacts in all synthesized gaits to match those of a default walk.

### 6.2.3 Numerical Solution

In order to solve the joint inverse optimization problem defined by equation 15, we first note that the optimization is in a form which is partially decoupled. In particular, only the  $\|\phi_i - R(A_i, \theta)\|_2^2 + r(\theta)$  regression error term relates the different animals to each other (via  $\theta$ ), and that remaining term  $D(f(A_i, \phi_i), M_i)$  can be independently evaluated for each  $\phi_i$ . This leads to an optimization technique in which a series of otherwise independent optimizations for each  $\phi_i$  are coupled together by the regression function  $R(A_i, \phi)$  and the regularization function  $r(\theta)$ . This is achieved in a manner reminiscent of coordinate descent by alternating between minimizing  $\theta$  and minimizing  $\phi_1, \dots, \phi_n$ .

Our approach to minimizing equation 15 involves a set of coupled instances of the covariance matrix adaptation (CMA) algorithm [Hansen et al. 1996]. We maintain a separate mean and covariance matrix to solve for for each  $\phi_i$ , denoted  $\mu_i$  and  $C_i$  respectively. At the beginning of each iteration a fixed number of samples  $\phi_{i,1}, \dots, \phi_{i,m}$  (we use  $m = 64$  in our tests) is drawn for each  $\phi_i$  distributed according to  $\mu_i$  and  $C_i$ . Treating these

method	mean error	median error
default	0.495	0.435
kinematic interpolation	0.451	0.387
independent inverse interpolation	0.473	0.360
joint-inverse interpolation	0.327	0.245
(base inverse fit)	0.142	0.127

**Table 2:** The mean and median errors over the combined biped and quadruped motion databases of leave-one-out tests in which one animal was excluded and its gait synthesized based on the other animals. The errors are measured using metric given by equations 16-20. Note that the cat and emu were excluded from these statistics because the ‘independent inverse interpolation’ spacetime constraints optimization failed to converge for them.

samples as the current estimates for each  $\phi_i$ , equation 15 is minimized for  $\theta$  while holding all  $\phi_{i,1}, \dots, \phi_{i,m}$  fixed. As only the  $\|\phi_i - R(A_i, \theta)\|_2^2 + r(\theta)$  term depends on  $\theta$ , solving for  $\theta$  reduces to a straightforward regularized least-squares optimization and can be solved relatively efficiently with an off-the-shelf unconstrained optimizer (our choices of  $R$  and  $r$  actually allow a solution with a single linear least squares solve, but we use LBFGS instead since it allows increased flexibility in the code and is sufficiently efficient). This yields a new estimate for  $\theta$ , and allows the cost associated with each  $\phi_{i,j}$  sample to be computed as defined by equation 15, after which the mean  $\mu_i$  and covariance  $C_i$  associated with each  $A_i$  are independently updated using the standard CMA update [Hansen et al. 1996] omitting any samples for which  $f(A_i, \phi_{i,j})$  fails.

Solving a joint inverse optimization problem also requires a setting for the parameter  $\beta$  used to weight the terms in the objective function resulting from how closely each  $\phi_i$  matches  $R(A_i, \theta)$ . Because lower values of  $\beta$  tend to converge more quickly, a continuation is performed where  $\beta$  is started out at a low value and then gradually increased over the course of the optimization. For quadrupeds we set the value of  $\beta$  in iteration  $i$  to  $\beta_i = 0.01 + 0.02 i$  over a total of 50 iterations while for bipeds we use  $\beta_i = 0.001 + 0.004 i$  over a total of 35 iterations.

## 7 Results

We demonstrate our approach of animal gait synthesis using joint inverse optimization on a motion database of walking gaits for six bipeds and twelve quadrupeds as illustrated in figures 1 and 2. Although in principle we could use both the bipeds and quadrupeds simultaneously, for simplicity we automatically select whether to synthesize a motion using only the bipeds or only the quadrupeds depending of whether the input animal is a biped or a quadruped. Processing this motion database with the joint inverse optimization algorithm takes several days when run on a cluster of 96 computers, but need only be done once. The main computational bottleneck in this preprocessing lies in the use of the generative model in the inner loop of the optimization, since each evaluation involves solving an expensive spacetime constraints problem. In practice this means that, the time to run the joint inverse algorithm described in section 6.2.3 scales approximately linearly with the number of animals in the motion database, limiting its application to databases with at most a few dozen animals, although a more efficient generative model could in principle substantially improve on this. Synthesizing a gait for a new animal once the motion database has been preprocessed takes only a few minutes on a single core.

In order to quantitatively compare our approach with other alternatives, table 2 shows the result of a leave-one-out cross-validation for several potential synthesis techniques, including ours:

**default** The gait is synthesized using a fixed hand-chosen default value of  $\phi_i$  for all animals.

**kinematic interpolation** The gaits in the motion database are directly interpolated weighted according to equation 22. The resulting motion will generally not satisfy the laws of physics, so a final spacetime constraints optimization is performed which attempts to match the kinematic interpolation as closely as possible while satisfying the laws of physics.

**independent inverse interpolation** The gait is synthesized using parameters interpolated using equation 22, but *without* any joint inverse optimization (i.e. the generative parameters for each animal are optimized independently). We note that because the main computational bottleneck in joint inverse optimization is in the evaluation of the generative model rather than in fitting the regression parameters, an independent inverse optimization is not significantly more efficient to solve.

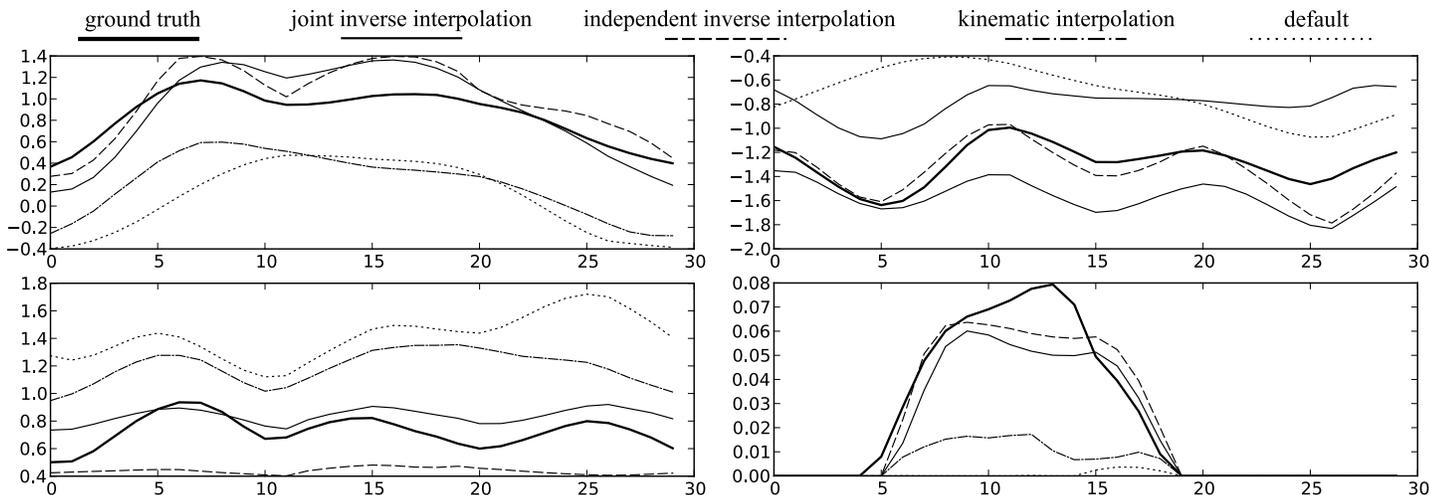
**joint-inverse interpolation** Our proposed approach in which a gait is synthesized using parameters interpolated using equation 22 *with* joint inverse optimization.

**(base inverse fit)** For comparison, each gait is synthesized using the optimal independently-fit inverse parameters. In contrast to the other approaches listed here this one *does* make use of the ground-truth motion of the animal and is thus analogous to the approach of [Liu et al. 2005]. Although this approach gives the least error, it is fundamentally incapable of synthesizing motions for new animals.

As shown in table 2, these other potential approaches are outperformed by joint inverse optimization. We also illustrate a comparison of the values of the foot height and several DOFs for the ground truth and leave-one-out synthesized gaits for the Thomson’s gazelle in figure 4. Additional examples to allow a more qualitative comparison can be found in the accompanying video, with additional detail provided in the supplemental video and PDF. In general, we find that the gaits we synthesize often form a close visual match with the animal’s ground truth motion.

It is useful to contrast the approach of independent inverse optimization with that of joint inverse optimization. Sometimes, the independent inverse optimization approach performs moderately well, and often outperforms kinematic interpolation. We suspect that this is because our generative parameters are in general better suited for smooth interpolation onto new animals than are kinematic parameters. It is this interpolability which is exploited by joint inverse optimization to achieve higher quality results. In addition to improving on the quality of the results, joint inverse optimization has a secondary advantage over independent inverse optimization in that it is more robust. Because the algorithm in section 6.2.3 ignores samples of  $\phi_{i,j}$  for which  $f(A_i, \phi_{i,j})$  fails, these failing samples are essentially assigned infinite cost. This automatically guides the algorithm to values of the inverse parameters which robustly allow successful synthesis. Independent inverse interpolation, on the other hand occasionally fails badly, and for instance was unable to synthesize leave-one-out gaits for the emu or cat due to the resulting spacetime optimization not converging to a physically valid result.

We have also tested our approach on synthesizing gaits for a number of extinct animals, including land-birds, mammals, and dinosaurs. Even when the shape of the animals differs significantly from anything in the motion database (or even from anything currently living) the gaits appear to be visually reasonable. A snapshot of these motions appears in figure 5. We also have found that the synthesized gaits adjust in a visually reasonable manner to the size of the animal. For example, figure 6 shows gaits for two dinosaurs



**Figure 4:** From top left to bottom right: plots of the hip angle, knee angle, ankle angle, and foot height over the frames in gaits for a Thomson's gazelle.

with the same relative shape but different total sizes. The synthesized gaits are significantly different, and on a qualitative level capture the sort of differences which would be expected considering the triceratops' substantially greater size and mass. Of course in these cases there is no possibility of a 'ground truth' motion to validate these synthesized gaits against. Nevertheless, the synthetic gaits still provide visually compelling hypotheses for the motions of these creatures.

There are a couple of situations where our proposed method does not perform as well. The first of these situations is when little data exists for a type of animal in the motion database. For instance, our motion database contains only three cats, and the leave-one-out cross-validation for these motions shows an error greater than for the other animals in the database. The second situation in which this method can perform poorly is when large extrapolations are used. For instance the motions synthesized for a paraceratherium (four times the mass of an elephant) or for *argentinosauros huinculensis* (4.5 times the mass of an elephant) display some moderately serious artifacts. Extrapolating further to the motion of a *amphicoelias fragillimus* (20 times the mass of an elephant) essentially fails, yielding clearly unrealistic results. The amount of extrapolation allowed before the synthesized gait deviates from the true gait in a visually obvious way depends on the animal. For instance, the gazelle's gait begins to show artifacts if we exclude the three most similar animals from the motion database (the gazelle, steenbok, and antelope, with the most similar remaining animal being the cat). The giraffe's gait, on the other hand, is relatively insensitive to variations in the inverse parameters and remains visually plausible for even wide extrapolations. Nevertheless, when used on animals with a size near those in the motion database, the method given here performs well, and yields visually plausible results over a wide range of dinosaurs and other animals with a shape relatively unlike anything currently living.

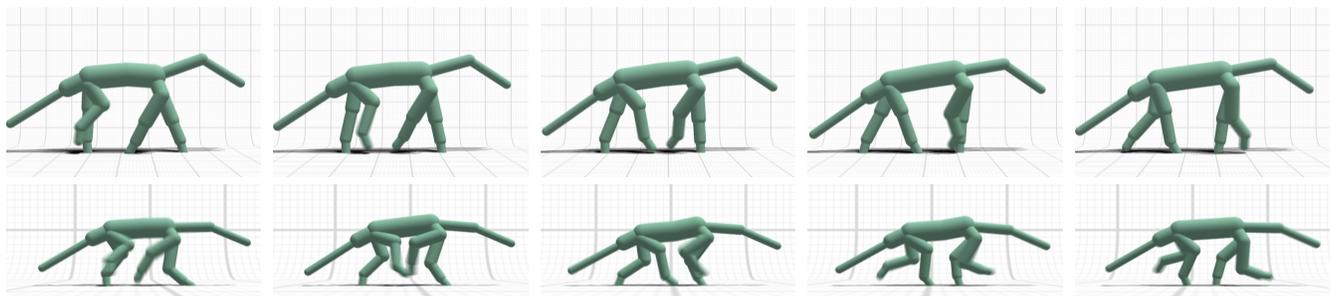
## 8 Conclusion

We have presented a technique for learning a model representing the style of walking gaits for a range of different animals, then using this model to synthesize gaits for new animals. Our method is based on a novel algorithm called *joint inverse optimization* which learns coherent patterns underlying the gaits of different animals.

This allows the synthesis of visually plausible gaits for a variety of different extinct creatures, and has been verified to be superior to several other potential techniques at estimating the gaits of a range of animals for which we do have data.

Although this paper applies the joint inverse optimization algorithm to animal gait synthesis using a specific choice of generative model and motion database, the joint inverse optimization algorithm at the core of our approach is more general as well as relatively simple. Once the motion database and generative model have been specified, a basic implementation can be written in a few hundred lines of code. An interesting avenue for future research is thus to apply joint inverse optimization with an improved motion database or generative model. One simple and obvious extension would be to construct a richer motion database including a wider range of animals and gaits. A more involved but exciting extension would be to combine this work with recent research synthesizing gaits using a model of an animal's musculature [Wang et al. 2012; Mordatch et al. 2013; Geijtenbeek et al. 2013]. The more biologically realistic generative model of this approach could potentially allow for the synthesis of highly realistic gaits across a very wide range of animals by estimating how different biological properties change with the shape and size of the animal. Although what properties are most useful is a question for future research, possibilities include the percentage of the body devoted to muscle vs. other elements such as skeletal structure and internal organs, the tendency to adopt gaits which minimize the possibility of injury, or selecting gaits which are suited to agility versus metabolic efficiency. Finally, since joint inverse optimization is not limited to animal locomotion, its potential application to other areas is an interesting avenue for future work.

Another interesting area for future work is in the development of a better regression function. As mentioned in section 6.2.2, we initially experimented with a regression function capturing physical scaling laws in the spirit of [Hodgins and Pollard 1997]. It is worth considering why our attempts to use these scaling laws were unsuccessful. One reason likely lies in the fact that our generative model does not capture all aspects of a real animal which are relevant to its motion. For instance, since our generative model does not accurately model muscles nor the possibility of the animal breaking a bone or otherwise injuring itself, we cannot directly model certain scaling laws relating cross-sectional muscle area of



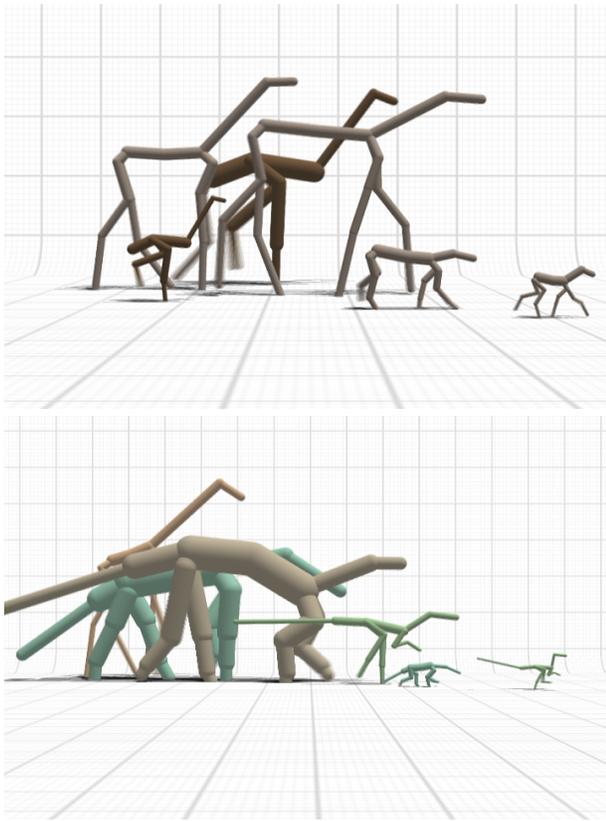
**Figure 6:** A comparison of the gait cycles of two dinosaurs with identical shapes but different sizes. The top shows a triceratops, while the bottom shows the same animal shape scaled down to the size of a protoceratops (one-fifth the height of a triceratops).

safety factors in bone sizes to the animal's mass [Alexander 1996]. Our nearest-neighbor based regression function, however, still allows the *effect* of these scaling laws to be approximately modeled, even when the available generative parameters only indirectly relate to the true scaling laws. A more fundamental issue with synthesis based on physical scaling laws stems from the difficulty in capturing the more subtle stylistic elements of an animal's gait. For instance, the cheetah and antelope in our motion database have very similar masses, yet significantly different motions. In order to capture this distinction, the regression function must capture more aspects of an animal's form than just its overall size. Our regression function achieves this by incorporating the relative lengths of an animal's limbs into the nearest-neighbor interpolation (figure 7). Furthermore, while it is easy to imagine still further features which may impact an animal's motion (for instance, the type of terrain the animal lives in), it is not always clear what the appropriate scaling laws should be. That said, we do think that biomechanical scaling laws have the potential to be useful, and a regression function combining the strengths of these laws with those of nearest-neighbor interpolation could potentially allow more accurate results in the case of large extrapolations.

On a more theoretical level, our approach captures variations in style between different animals, but cannot account for the multitude of different motions often performed by a single animal. This is because we treat the style of an animal's gait as fixed by the shape of the animal. Instead, it would be fruitful for future techniques to be able to model the *set* of styles likely to be exhibited in the motions of an animal with a given shape. Even further in this direction, an ideal method would not be limited to just gait synthesis, but would be able to produce a diverse range of realistic motions, or even control strategies, which accurately depict how an animal might move.

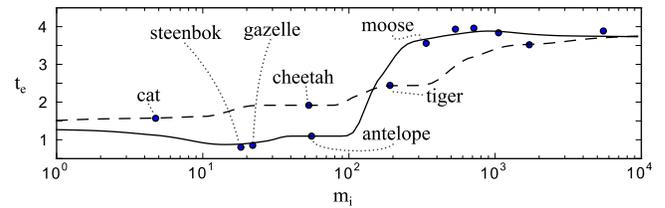
## References

- ALEXANDER, R. 1996. *Optima for Animals*. Princeton paperbacks. Princeton University Press.
- ARIKAN, O., AND FORSYTH, D. A. 2002. Interactive motion generation from examples. *ACM Transactions on Graphics (ACM SIGGRAPH 2002)* 21, 3, 483–490.
- BAY, H., ESS, A., TUYTELAARS, T., AND VAN GOOL, L. 2008. Speeded-up robust features (surf). *Comput. Vis. Image Underst.* 110, 3 (June), 346–359.
- BOYD, S., AND VANDENBERGHE, L. 2004. *Convex Optimization*. Cambridge University Press, New York, NY, USA.
- BRUDERLIN, A., AND WILLIAMS, L. 1995. Motion signal processing. In *Proceedings of SIGGRAPH 95*, Computer Graphics Proceedings, Annual Conference Series, 97–104.
- COROS, S., BEAUDOIN, P., AND VAN DE PANNE, M. 2010. Generalized biped walking control. *ACM Transactions on Graphics* 29, 4, Article 130.
- COROS, S., KARPATHY, A., JONES, B., REVERET, L., AND VAN DE PANNE, M. 2011. Locomotion skills for simulated quadrupeds. *ACM Transactions on Graphics* 30, 4, Article TBD.
- DE LASA, M., MORDATCH, I., AND HERTZMANN, A. 2010. Feature-Based Locomotion Controllers. *ACM Transactions on Graphics* 29, 3.
- FANG, A. C., AND POLLARD, N. S. 2003. Efficient synthesis of physically valid human motion. *ACM Trans. Graph.* 22, 3, 417–426.
- GEIJTENBEEK, T., VAN DE PANNE, M., AND VAN DER STAPPEN, A. F. 2013. Flexible muscle-based locomotion for bipedal creatures. *ACM Transactions on Graphics* 32, 6.
- GILL, P. E., MURRAY, W., AND SAUNDERS, M. A. 2005. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM Review* 47, 1, 99–131.
- HANSEN, N., HANSEN, N., OSTERMEIER, A., AND OSTERMEIER, A. 1996. Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. Morgan Kaufmann, 312–317.
- HECKER, C., RAABE, B., ENSLOW, R. W., DEWEESE, J., MAYNARD, J., AND VAN PROOIJEN, K. 2008. Real-time motion retargeting to highly varied user-created morphologies. *ACM Trans. Graph.* 27, 3, 1–11.
- HODGINS, J. K., AND POLLARD, N. S. 1997. Adapting simulated behaviors for new characters. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, SIGGRAPH '97, 153–162.
- JAIN, S., AND LIU, C. K. 2011. Controlling physics-based characters using soft contacts. *ACM Trans. Graph. (SIGGRAPH Asia)* 30 (Dec.), 163:1–163:10.
- KOVAR, L., GLEICHER, M., AND PIGHIN, F. 2002. Motion graphs. *ACM Transactions on Graphics* 21, 3 (July), 473–482.
- KRY, P. G., REVERET, L., FAURE, F., AND CANI, M.-P. 2009. Modal locomotion: Animating virtual characters with natural vibrations. *Computer Graphics Forum*.



**Figure 5:** The top illustrates the gaits synthesized for a number of extinct mammals and land birds, while the bottom shows the gaits synthesized for a several dinosaurs. In the bottom image a giraffe appears in the background for size comparison.

- LEE, S. J., AND POPOVIĆ, Z. 2010. Learning behavior styles with inverse reinforcement learning. *ACM Trans. Graph.* 29, 4, 1–7.
- 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.
- LIU, C. K., HERTZMANN, A., AND POPOVIĆ, Z. 2005. Learning physics-based motion style with nonlinear inverse optimization. *ACM Trans. Graph.* 24, 3, 1071–1081.
- MACIVER, M. A., PATANKAR, N. A., AND SHIRGAONKAR, A. A. 2010. Energy-information trade-offs between movement and sensing. *PLoS computational biology* 6, 5, e1000769.
- MORDATCH, I., TODOROV, E., AND POPOVIĆ, Z. 2012. Discovery of complex behaviors through contact-invariant optimization. *ACM Trans. Graph.* 31, 4 (July), 43:1–43:8.
- MORDATCH, I., WANG, J. M., TODOROV, E., AND KOLTUN, V. 2013. Animating human lower limbs using contact-invariant optimization. *ACM Trans. Graph.* 32, 6 (Nov.), 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 *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ACM, New York, NY, USA, I3D '12, 87–94.



**Figure 7:** Plots of the  $t_e$  parameter (see table 1) for the quadrupeds in the motion database against the log of the animal's mass as found by joint inverse optimization. The solid and dotted curves respectively show the inverse parameters for an animal with the shape of a gazelle or cheetah, but uniformly scaled to a different mass. The difference between these curves illustrates the dependence of the generative parameters on the shape of the animal, while the shape of each individual curve shows the depends on the animal's mass.

- RAIBERT, M. H., AND HODGINS, J. K. 1991. Animation of dynamic legged locomotion. *SIGGRAPH Comput. Graph.* 25 (July), 349–358.
- 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.
- SIMS, K. 1994. Evolving virtual creatures. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '94, 15–22.
- TAN, J., GU, Y., TURK, G., AND LIU, C. K. 2011. Articulated swimming creatures. *ACM Trans. Graph.* 30, 4 (July), 58:1–58:12.
- WAMPLER, K., AND POPOVIĆ, Z. 2009. Optimal gait and form for animal locomotion. *ACM Trans. Graph.* 28, 3, 1–8.
- WAMPLER, K. 2012. *Computational Generation of Terrestrial Animal Locomotion*. PhD thesis, Seattle, WA, USA. AAI3552870.
- WANG, J. M., HAMNER, S. R., DELP, S. L., AND KOLTUN, V. 2012. Optimizing locomotion controllers using biologically-based actuators and objectives. *ACM Trans. Graph.* 31, 4, 25.
- WEI, X., MIN, J., AND CHAI, J. 2011. Physically valid statistical models for human motion generation. *ACM Trans. Graph.* 30 (May), 19:1–19:10.
- WITKIN, A. P., AND POPOVIĆ, Z. 1995. Motion warping. In *Proceedings of SIGGRAPH 95*, Computer Graphics Proceedings, Annual Conference Series, 105–108.
- WU, J.-C., AND POPOVIĆ, Z. 2003. Realistic modeling of bird flight animations. *ACM Trans. Graph.* 22, 3 (July), 888–895.
- YIN, K., LOKEN, K., AND VAN DE PANNE, M. 2007. Simbicon: Simple biped locomotion control. *ACM Trans. Graph.* 26, 3, Article 105.
- ZHANG, L., SNAVELY, N., CURLESS, B., AND SEITZ, S. M. 2004. Spacetime faces: High-resolution capture for modeling and animation. In *ACM Annual Conference on Computer Graphics*, 548–558.