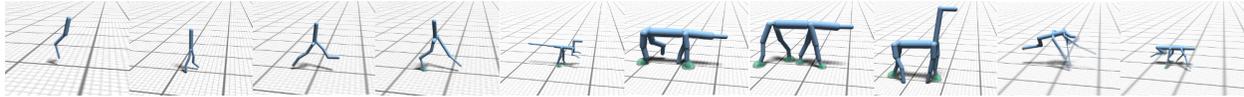


Optimal Gait and Form for Animal Locomotion

Kevin Wampler*

Zoran Popović

University of Washington



Abstract

We present a fully automatic method for generating gaits and morphologies for legged animal locomotion. Given a specific animal's shape we can determine an efficient gait with which it can move. Similarly, we can also adapt the animal's morphology to be optimal for a specific locomotion task. We show that determining such gaits is possible without the need to specify a good initial motion, and without manually restricting the allowed gaits of each animal. Our approach is based on a hybrid optimization method which combines an efficient derivative-aware spacetime constraints optimization with a derivative-free approach able to find non-local solutions in high-dimensional discontinuous spaces. We demonstrate the effectiveness of this approach by synthesizing dynamic locomotions of bipeds, a quadruped, and an imaginary five-legged creature.

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

Keywords: animation, character dynamics, spacetime optimization, gait

1 Introduction

Both scientists and artists have long been fascinated by the interplay between the form and motion of animals. It is intuitively clear that an elephant's thick and sturdy limbs are related to its weight just as a cheetah's light stature and spring-loaded legs and spine are related to its ability to run quickly. Despite this longstanding interest, there are relatively few options in the way of automatic tools to aid in determining how an animal should run given only information on its shape, or for determining its shape given constraints on its motion. This is a particularly noticeable problem when modeling animals which are extinct or entirely fanciful.

Our approach to this problem focuses on terrestrial locomotion. Given only the basic shape of a legged animal, we can fully automatically synthesize a visually plausible gait without relying on any pre-authored or recorded motions. Similarly, given an initial guess as to the shape of an animal and constraints on its motion, such as the required speed of its gait, we can simultaneously solve for the animal's motion while reportioning its skeleton so that it can move more efficiently. We can also solve for the pattern and timing in which an animal's feet should contact the ground.

*email: wampler@cs.washington.edu

We base our method for attacking these problems on an optimization which attempts to solve for the most efficient shape and motion for an animal, subject to a set of constraints dictated by the laws of physics. Such approaches have been used extensively for animation in computer graphics, but due to the dimensionality and nonlinearity of the equations involved they have been notoriously difficult to reliably apply to complex characters without a good initial guess for the motion. This problem is made worse by the difficulty of applying efficient derivative-based optimization techniques to solve for variables which are not easily formulated in a differentiable manner, such as the order of the foot contacts.

We address these difficulties with a novel combination of an efficient derivative-free global optimization with a derivative-aware optimization which is particularly well suited to solving for an animal's form and motion without requiring a good initialization. This allows us to leverage the ability of derivative-based optimizers to efficiently generate closed-loop motions for complicated animals with the ability of population based approaches to avoid local minima and to manage poorly- or non-differentiable variables.

The motions and shapes generated by our approach appear visually plausible and do not require any per-animal user authoring or preset motion patterns. As far as we are aware, ours is the first method to achieve this fully automatically for non-simplified characters with highly underactuated motions. Although the biomechanical model we employ is currently somewhat approximate, we hope that our method's ability to perform a *de novo* synthesis from basic physical principles is a first step toward frameworks in which more accurate biomechanical models may be applied to create truly accurate motions for animals alive, extinct, and imaginary.

2 Related Work

There is an old and extensive body of scientific literature on the relationship between shape and motion in animals. One of the best known of these works is *On Growth and Form*[Thompson 1992] which provides a fascinating investigation into the relationships between the forms of animals and physical or mathematical forms. Later work, for instance *Optima for Animals*[Alexander 1996] and *Principles of Animal Design*[Weibel et al. 1998] present optimization as a technique for describing the shape and motion of animals.

The interrelationship between morphology and locomotion is also of interest in robotics, biomechanics and paleontology. Optimization based approaches are again common in these contexts but the dimensionality and nonlinearity of these optimizations often limits the problems which can be studied. Hutchinson and Gatesy[2006] describe some of the difficulties related to the synthesis of dinosaur gaits. This difficulty has typically limited its applicability to highly simplified models or has required a good initialization for the optimization. For instance, optimization over morphology and gait [Paul and Bongard 2001], and optimization over different gait styles [Srinivasan and Ruina 2006] have both been applied to simplified bipedal models.

In computer graphics, optimal dynamic character motions are synthesized primarily by solving a large variational problem, often referred to as spacetime optimization [Witkin and Kass 1988]. The continuous optimization methods often used for these problems tend to get “stuck” easily in local minima, and consequently have been used primarily to alter an existing motion sequence when applied to complex characters [Popović and Witkin 1999; Fang and Pollard 2003; Liu et al. 2005; Liu et al. 1994; Popović and Witkin 1999; Rose et al. 1996; Witkin and Kass 1988]. Some techniques have been developed to alleviate the need for an initial motion by either using a database of similar motions as in [Safonova et al. 2004] or by utilizing a specialized dynamics formulation as in [Fang and Pollard 2003], which has produced reasonable flipping and tumbling motions with little initialization. Unfortunately these methods are not yet able to generate plausible *de novo* gaits for animals for which we do not have example motions. This problem is further compounded when we wish to allow the foot timings to change in the optimization. Time warping has been used to allow small changes in foot timings [Liu et al. 2006], but cannot capture changes such as that between a walk and a run. Our work tries to remove these limitations of the spacetime optimization in order to optimize for larger characters, their morphology and optimal gait.

An alternative to the use of a continuous optimization to solve for a character’s motion is to apply population based methods such as evolutionary and genetic algorithms. Auslander et al.[1995] optimized for simplified characters with fixed morphology while Pollard and Hodgins[1997] used repeated simulated annealing to adapt a controller to a human character with different proportions. Simulated annealing was also used to determine the sinusoidal control pattern of swimming creatures [Grzeszczuk et al. 1998]. Synthesis of different gaits using a pose control graph optimized through a combination of a local sampling technique with a variation on random restart has been achieved by [van de Panne 1996]. One of the best known works in this area is *Evolving Virtual Creatures*[Sims 1994]. This paper presents a genetic algorithm to search for both a shape and a control strategy which would allow a creature to best perform a pre-specified task, such as swimming or moving on land. The resulting animations are highly entertaining and often lifelike, but rarely resemble those of actual animals in the case of terrestrial locomotion. These sorts of population based optimization approaches have the advantage that they do not rely on derivatives and can avoid local minima to an extent, but they appear to have trouble achieving the highly under-actuated motions shown by many real animals. There is also some evidence to suggest that they scale less well to the high dimensional spaces required by these problems [Koh et al. 2008].

More recently, authored parametric methods have been used to generate morphology-dependent locomotion patterns for an interactive game environment (Spore) [Hecker et al. 2008]. This paper was one of the key motivations to automatically synthesize more physically and energetically realistic motions. Other approaches provide the user with tools to author the torque actuations of a gait and then simulate the animal via forward dynamics [Raibert and Hodgins 1991; Kry et al. 2009].

Our method is similar in spirit to that of Sims’ and others in that it is based upon an optimization which simultaneously solves for both form and motion, but the specifics of how we achieve this are different. Instead of optimizing over completely different skeleton topologies, we only vary the lengths and radii of an animal’s limbs. This restriction allows us to use efficient derivative-based optimization methods to solve for shapes and motions more closely resembling those found in nature. Although this means that we cannot automatically add an extra leg to an animal, in practice the range of possible shapes is still large – a dog and a horse have the same topology.

In order to optimize over the foot contact times and aid in avoiding local minima, we employ a hybrid optimization technique which combines a derivative-aware spacetime constraints approach as the inner loop to a derivative-free global optimization based on the covariance matrix adaptation evolution strategy [Hansen et al. 1996]. This approach appears to have a greatly increased efficiency over existing population based constrained optimizers (see appendix A) and is well suited to the problem of gait and morphology optimization.

3 Continuous Optimization

The inner loop to our optimization is formed by a derivative-based spacetime constraints approach. Our formulation is not fundamentally different from previous approaches, but is notable in two key respects. Firstly, we have specifically focused the optimization on generating a closed-loop gait cycle, and it is capable of solving for such a motion without a good initialization. Secondly, in order to avoid unrealistic results without relying on a good initialization we must be more thorough in the optimization’s formulation than is typical.

We take as input an animal defined as a kinematic tree of connected limbs. We model each limb as a cylinder with a length, radius, and mass. Pairs of limbs are connected by *joints* which define a parametrized transformation from the endpoint of the parent limb to the endpoint of the child limb. Generally this transformation consists of a rotation followed by a translation along the length of the child limb. We further define a special joint at the root giving the global rotation and translation of the animal. In addition we require that the limb endpoints of the animal’s feet and the limb corresponding to its head (if any) be specified.

Given such a description of an animal we define its motion over a single gait cycle with a set of variables describing the parameters for its joints at each of a fixed number of frames. For each foot, at each frame when it is in contact with the ground, we define six variables giving the force and torque exerted on the foot via this contact, denoted by \mathbf{f}_{i,j_c} and \mathbf{t}_{i,j_c} . We also include three variables for each rotational joint parameter defining its passive actuation characteristics, represented by a spring stiffness, rest state, and a dampening constant. In later sections we will include additional variables controlling the shape of the animal and when each foot is in contact with the ground.

To optimize for a gait we take a “treadmill” approach. That is, if we wish to derive a gait for an animal running at some velocity \mathbf{V}_{gait} , we solve an optimization where the animal stays stationary overall and instead has its feet moving at $-\mathbf{V}_{gait}$. This allows us to easily define the optimization to be cyclic by applying all terms which rely on temporal derivatives, such as the physical constraints, in a cyclically looping manner.

To aid in writing later equations, we define here some common terms which will appear in our equations:

- m the default mass of the animal
- \mathbf{q} a single joint degree of freedom
- \mathbf{f}, \mathbf{t} force and torque, respectively
- $p(i, j)$ position of bone endpoint (node) or joint j at frame i
- $v(i, j)$ the linear velocity of node j at frame i
- $R(i, j)$ The 3×3 rotation matrix of node j at frame i

In addition, unless noted otherwise, we will use i to index over frames, j to index over joints or bone endpoints, l to index over limbs, and j_c to index over ground contacts. So for instance $\mathbf{f}_{i,j}$ will refer to the force at joint j in frame i .

Our optimization utilizes several dynamic and kinematic terms, discussed next. For the moment, note that the current formulation requires that the frames at which each foot is in contact with the ground be fixed in advance. We will later introduce method by which we can allow the foot contact times to vary in the global optimization.

3.1 Kinematic constraints

The simplest of the constraints used in our optimization are those enforcing a kinematic validity to the motion. The first of these enforce that all parts of the skeleton should remain above the plane of the ground and that the feet should be on the ground and have the correct velocity when in contact:

$$\forall_{i,p} : p(i,p)_y \geq 0 \quad (1)$$

$$\forall_{i,j_c} : p(i,j_c)_y = 0 \quad (2)$$

$$\forall_{i,j_c} : v(i,j_c) = -\mathbf{V}_{gait} \quad (3)$$

For efficiency and ease of differentiation we enforce non-self-intersection only approximately by having the user provide a list of pairs of points in the animal $(p_{1,1}, p_{1,2}), \dots, (p_{n,1}, p_{1,2})$ and enforce a minimum distance constraint between these points:

$$\forall_{i,k} : \|p(i, p_{k,2}) - p(i, p_{k,1})\|^2 \geq r^2 \quad (4)$$

Geometrically we model each foot contact as occurring at a single point. This simple approach does not allow more detailed modeling of a foot's structure such as the heel-toe roll in human gaits or the use of the toes to push off the ground in some animal gaits. Fortunately most non-human animals have small ground contact areas, but we have still found it necessary to explicitly define a preferred orientation for a foot on the ground by adding constraints enforcing each foot's rotation about the y -axis to be zero when in contact:

$$\forall_{i,j_c} : R(i,j_c)_{x,z} - R(i,j_c)_{z,x} = 0 \quad (5)$$

Finally, to remove the translational invariance of the optimization and keep the resulting gait from drifting arbitrarily we add a term to keep the average position of the animal along the x - z plane centered at the origin.

$$\sum_i p(i, root)_x = \sum_i p(i, root)_z = 0 \quad (6)$$

3.2 Mass normalization

In the following description of the dynamical constraints and objective function to our optimization some terms are normalized by the default mass of the animal. This helps keep the optimization better conditioned for animals of large mass and is similar to methods employed by [Srinivasan and Ruina 2006; Hodgins and Pollard 1997]. We achieve this normalization by treating the optimization variables for the ground reaction forces as scaled by the mass of the animal $\frac{\mathbf{f}_{i,j_c}}{m}$ and $\frac{\mathbf{t}_{i,j_c}}{m}$ and introducing a similar multiplicative normalization term to the other equations which deal with dynamics. This allows us keep all intermediate computations in standard SI units – a benefit in defining terms which depend on constants such as the maximum stress a muscle can support, as discussed later in section 3.6.

3.3 Dynamic constraints

We phrase the dynamical constraints of our method with a Newton-Euler formulation reminiscent of Fang and Pollard [2003] by expressing the dynamical constraints about the animal's center of

mass, yielding a total of six dynamical constraints per frame. In order to calculate these constraints more efficiently we approximate each limb by a point mass located at the limb's center of mass. At each frame, we ensure that time derivatives of the animal's linear and angular momenta are equal to the net force and torque on the animal respectively:

$$\forall_i : \dot{\mathbf{p}}_i = mg + \sum_{j_c} \mathbf{f}_{i,j_c} \quad (7)$$

$$\forall_i : \dot{\mathbf{L}}_i = \sum_{j_c} (p_i, j_c - \mathbf{CM}_i) \times \mathbf{f}_{i,j_c} + \mathbf{t}_{i,j_c} \quad (8)$$

Where p_{cm} and v_{cm} give the position and velocity of a limb's center of mass and \mathbf{CM}_i gives the position of the center of mass of the animal at frame i .

In order to ensure physical validity we also constrain the ground contact forces to be within a friction cone:

$$\forall_{i,j_c} : \frac{1}{m} \left(\mu \mathbf{f}_{i,j_c \perp} - \left\| \mathbf{f}_{i,j_c \parallel} \right\| \right) \geq 0 \quad (9)$$

Where \mathbf{f}_{\parallel} and \mathbf{f}_{\perp} represent the components of the contact force parallel and perpendicular to the ground respectively. We similarly constrain the torque components of the ground reactions to lie within an ellipsoid scaled by the force component of the reaction, approximating the behavior of a foot which has an area of contact with the ground, even though the foot in our optimization is still geometrically a point:

$$\forall_{i,j_c} : \frac{1}{m} \left(\mathbf{f}_{i,j_c \perp}^2 - \left\| \frac{\mathbf{t}_{i,j_c \parallel}}{\nu_b} \right\|^2 - \left(\frac{\mathbf{t}_{i,j_c \perp}}{\nu_t} \right)^2 \right) \geq 0 \quad (10)$$

Where μ is the coefficient of static friction and ν_b and ν_t give bounds on the maximum allowed torques in the directions parallel and perpendicular to the ground respectively. In all our optimizations we use $\mu = 0.5$ and $\nu_b = \nu_t = 0.02$

3.4 Passive elements

In addition to being able to actuate their joints through muscular exertion, many animals also have musculo-skeletal structures that behave like springs or dampers and passively actuate these joints. The importance of these in reproducing styles of walk in humans was noted in [Liu et al. 2005], which may also be referred to for a more thorough discussion of these terms.

We support passive elements in our optimization by representing the total torque at each joint as the sum of a passive and an active component:

$$\mathbf{t} = \mathbf{t}_a + \mathbf{t}_p \quad (11)$$

Where \mathbf{t}_a is the component of the torque arising directly from muscular exertion while \mathbf{t}_p is the component arising from passive actuation. This allows the animal to achieve more efficient gaits and is reflected in our objective function (13). We compute \mathbf{t}_p for a gait by including three variables for each joint degree-of-freedom, \mathbf{q}_j , in the animal: A spring constant k_{s_j} , a spring rest length, $\bar{\mathbf{q}}_j$, and a dampening coefficient k_{d_j} . The passive force for that degree of freedom can then be written as:

$$\mathbf{t}_{p_j} = -k_{s_j}(\mathbf{q}_j - \bar{\mathbf{q}}_j) - k_{d_j}\dot{\mathbf{q}}_j \quad (12)$$

In our tests, we have found that the inclusion of passive elements does not drastically alter the overall form of a gait, but does make the resulting motions significantly smoother and more believable.

3.5 Objective

The objective function used in our optimization, if we are not optimizing for morphology, is the sum of four terms:

$$C_1 \frac{1}{m} \sum_{i,j} \|\mathbf{t}_{\mathbf{a},i,j}\|^2 + \quad (13)$$

$$C_2 \sum_{i,\mathbf{q}} \dot{\mathbf{q}}^2 + \quad (14)$$

$$C_3 \sum_i \|\dot{\mathbf{p}}(i, head)_\perp\|^2 + \quad (15)$$

$$C_4 \sum_i \|R(i, head) - \mathbf{I}\|^2 \quad (16)$$

The first of these terms is standard in spacetime optimizations and minimizes the muscular exertion of the animal, where the torques used in this equation are found analytically from the motion and contact forces using inverse dynamics. The second term penalizes high-velocity joint motions and is necessary to avoid low-torque but unrealistic ‘wiggling’ motions that would otherwise occur in the flight phase of some gaits.

We also note that animals often attempt to keep their heads largely stable. This is both so that the brain is not jostled too much, and because motion of the head interferes with the visual processes needed to move through an environment. We thus use the third term to penalize translational motion of the head. Since motion of the head in the direction an animal is running does not much interfere with vision, we only penalize the component of this motion which is perpendicular to the velocity of the gait. The fourth term functions similarly and keeps the animal’s head facing in its direction of motion.

We weight each of these terms with a constant so that their contributions to the overall objective function are of the same general magnitude. Note that although these constants were hand tuned, we use the same values for all animals and gaits: $C_1 = 25$, $C_2 = 0.1$, $C_3 = 25$, $C_4 = 100$

3.6 Morphological terms

The spacetime optimization described thus far follows closely in the mold of previous approaches and is sufficient for generating gaits on a fixed morphology when the foot contact timings are known in advance. In order to allow optimization over morphology we simply add two new optimization variables per limb controlling the radius and length of the limb, scaling the mass of the bone proportionally so that it maintains a constant density. Thus larger limbs, while often providing an advantage in terms of locomotion, also carry the disadvantage that they increase the animal’s mass.

Once we have added these variables to parametrize over different shapes we must alter our objective function to better reflect the tradeoffs of having larger versus smaller limbs. We achieve this by redefining the torque minimization term in equation 13, replacing it with a term attempting to minimize the active forces exerted by the muscles, E_f , and a term attempting to minimize the stress put upon the muscle fibers, E_σ :

$$C_1 \left[\sum_{i,j} E_f(i,j)^2 + \sum_{i,j} E_\sigma(i,j)^2 \right] \quad (17)$$

In order to calculate E_f and E_σ we must estimate at each actuated joint, j , both the cross-sectional muscle areas of those muscles controlling the joint, denoted a_j , and a scaling factor for converting muscle forces into joint torques, denoted d_j . In our approach

we take only the basic shape of the animal given by the optimization variables and instead use a few simple heuristics to guess the values of a_j and d_j . This has the advantage of simplifying the optimization but has the disadvantage that our model is a rather crude biomechanical approximation of reality. Nevertheless this approach has proved sufficient for qualitatively reasonable results in solving for an animal’s shape.

We estimate a_j for each joint, j , by assuming that 25% of a limb’s cross-sectional area is taken up by muscle. Furthermore we note that generally (although certainly not always) in nature large muscles are connected at both ends to large limbs, partially because large muscle loads may damage the bones in a small limb. We represent these properties with the equation:

$$a_j = 0.25 \pi \min(r_{j,1}, r_{j,2})^2 \quad (18)$$

where $r_{j,1}$ and $r_{j,2}$ are the radii of the limbs adjacent to joint j and \min is a smooth and differentiable approximation to the min function in the non-negative quadrant:

$$\min(a, b) \doteq (a^{-10} + b^{-10})^{-\frac{1}{10}} \quad (19)$$

We estimate d_j by first choosing a value for the muscle’s attachment distance on each of the two limbs at joint j , $d_{j,\{1,2\}}$, and then combine these into a single scaling factor. To estimate $d_{j,\{1,2\}}$ we note that the attachment arm is constrained both by the length and the radius of each limb. We represent this by setting each limb’s attachment arm to be a minimum of a proportion of the length and radius of the respective limb:

$$d_{j,\{1,2\}} = \min(0.2 l_{\{1,2\}}, 1.5 r_{\{2,1\}}) \quad (20)$$

The most accurate way to combine $d_{j,1}$ and $d_{j,2}$ into a single scaling factor would include the dependence on the angle of the joint’s extension. We take a simplified approach and just derive the force-to-torque conversion factor for when the limb is extended at a right angle:

$$d_j = \frac{d_{j,1} d_{j,2}}{\sqrt{d_{j,1}^2 + d_{j,2}^2}} \quad (21)$$

Having computed values for a_j and d_j we can now define $E_f(i, j)$ and $E_\sigma(i, j)$ as used in equation 17. The definition of $E_f(i, j)$ is simple and merely computes the force exerted by the muscle at joint j in frame i :

$$E_f(i, j) = \frac{|\mathbf{t}_{\mathbf{a},i,j}|}{d_j} \quad (22)$$

The definition of E_σ is moderately more involved. We begin by estimating the magnitude of the force exerted upon the muscles attached to joint j , $|\mathbf{f}_{\mathbf{m},i,j}|$, which differs from $E_f(i, j)$ in that it accounts for forces due to both active and passive actuation:

$$|\mathbf{f}_{\mathbf{m},i,j}| = \frac{1}{d_j} \left(|\mathbf{t}_{\mathbf{a},i,j}| + |\mathbf{t}_{\mathbf{p},i,j}| + |\mathbf{t}_{\mathbf{e},i,j}| \right) \quad (23)$$

In addition we include a third term, $|\mathbf{t}_{\mathbf{e},i,j}|$, in computing this force. This prevents the optimization from allowing some limbs to become very thin by keeping all forces parallel to the limb’s axis and eliminating any torques at the adjacent joints. Due to inevitable errors in a real animal’s motion such precise forces are impossible to actually achieve. We thus assume that it is possible for any force through a limb to actually be exerted slightly more off-axis than determined by $\mathbf{f}_{i,j}$. We calculate the magnitude of the extra torque from this displacement by $|\mathbf{t}_{\mathbf{e}}|_{i,j} = 0.05 |\mathbf{f}_{i,j} \cdot v_{limb}|$ where v_{limb} is the vector from one endpoint of the limb to the other.

We can now compute $E_\sigma(i, j)$ as a scaled multiple of $|\mathbf{f}_{\mathbf{m},i,j}|$:

$$E_\sigma(i, j) = h(\sigma_{i,j}) |\mathbf{f}_{\mathbf{m},i,j}| \quad (24)$$

The scaling factor, $h(\sigma_{i,j})$, is chosen so that this term starts to dominate $E_f(i, j)$ when the stress upon the muscles at joint j , $\sigma_{i,j}$, reaches some value, σ_{max} . This stress is computed as the force per cross-sectional muscle area: $\sigma_{i,j} = |\mathbf{f}_{m_{i,j}}|/a_j$. In our case we use $h(\sigma) \doteq \sqrt{3(\sigma/\sigma_{max})^2 + 1} - 1$, and choose a value of $\sigma_{max} = 50 \frac{N}{cm^2}$, which corresponds to the maximum muscle stress exerted by endurance-trained human athletes [Häkkinen and Keskinen 2006].

4 Gait Optimization

The optimization as described so far requires that the foot contacts be specified in advance. This is a disadvantage in defining such optimizations, as it is not always obvious what the foot contact timings should be for a given animal moving at a particular speed.

To represent a foot contact optimization we assume that each foot touches the ground for only a single interval during a gait cycle. We then represent a gait's timing by introducing a single variable controlling the overall period of the gait and add a pair of optimization variables for each foot giving the start time and duration of the foot's contact interval, both measured as fractions of the gait's period.

In order to optimize over the foot contact timings it is necessary to formulate the spacetime optimization so that they can be quickly changed at runtime. Since we pre-generate code to calculate derivatives we must first formulate a problem that includes ground reaction forces for each foot in every frame, thus including all derivatives that might be needed later for any possible contact timing. At runtime we modify this optimization by removing the variables and constraints corresponding to inactive foot contacts. The ground reaction force variables can be removed by forcing them to be equal to zero. The inactive constraints can be removed directly from the optimization's constraint vector and Jacobian. If automatic instead of symbolic differentiation is used this step will probably not be necessary.

Because the variables for the foot contact times and durations are not differentiable, standard derivative-based constrained optimization methods cannot be applied directly. Furthermore, this problem contains many local minima which must be avoided in determining a good gait. We address both of these difficulties with a hybrid optimization scheme combining derivative aware and derivative-free techniques.

4.1 Hybrid optimization

In order to solve the gait optimization problem we use a novel hybrid optimization technique which combines a spacetime optimization as an inner loop to a sampling-based derivative-free optimization method based on a variant of the covariance matrix adaptation evolution strategy (CMA). This combines the efficiency in high dimensional spaces and ability to handle general constraints of spacetime optimization with the ability to handle non-differentiable variables and avoid many local minima. We will first describe the standard CMA algorithm briefly. For further details we refer readers to [Hansen and Kern 2004].

We begin with a function to be minimized, f and an initial Gaussian defined by a mean, \mathbf{m}_0 , and a covariance matrix \mathbf{C}_0 . We first draw λ samples from this distribution and evaluate f at each. We then select the μ samples with the lowest associated values of f . These samples are termed the *elites* and we will denote them by $\mathbf{x}_1, \dots, \mathbf{x}_\mu$ where $f(\mathbf{x}_1) \leq \dots \leq f(\mathbf{x}_\mu)$. With each elite we also associate a weight as defined by:

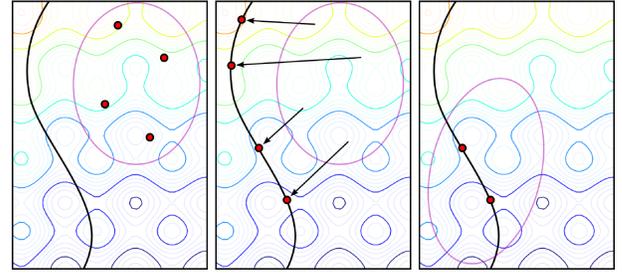


Figure 1: An illustration of a single iteration of basin-CMA. First λ samples are chosen from the current distribution. Next each sample is projected to a local constrained minimum. Finally the mean and covariance are updated according to the elite samples.

$$w_j = \frac{\ln(\mu + 1) - \ln(j)}{\mu \ln(\mu + 1) - \sum_{k=1}^{\mu} \ln(k)} \quad (25)$$

These weights are chosen so as to favor those elites with the lowest values of f , and other weighting schemes may also work, so long as $w_1 > \dots > w_\mu > 0$. Using these values for w and \mathbf{x} we then update the mean as:

$$\mathbf{m}_{i+1} = \sum_{j=1}^{\mu} w_j \mathbf{x}_j \quad (26)$$

and similarly update the covariance matrix as:

$$\mathbf{C}_{i+1} = (1 - c_{cov})\mathbf{C}_i + c_{cov} \sum_{j=1}^{\mu} w_j (\mathbf{x}_j - \mathbf{m}_j)(\mathbf{x}_j - \mathbf{m}_j)^T \quad (27)$$

As we iterate this method the multivariate Gaussian moves and shrinks until it becomes a point at the function's minimum or until a preset maximum number of iterations is reached.

4.1.1 Basin-CMA

The CMA algorithm has a few disadvantages which make it unsuitable for direct use in our optimization. Firstly it does not take advantage of derivatives when they are available, and thus is somewhat inefficient on our largely differentiable problem. More importantly, however, CMA is an entirely *unconstrained* optimization approach so it is unable to handle kinematic and dynamical constraints in our problem.

The solution to both of these problems is to use a spacetime optimization at each CMA sample point instead of evaluating the objective function directly. More formally, let f be our objective function and c_1, \dots, c_n be the constraint functions. Furthermore, let g be a function such that if $\mathbf{y} = g(\mathbf{x}|f, c_1, \dots, c_n)$ then \mathbf{y} is a local minimizer of f which satisfies the constraints c_1, \dots, c_n where the minimization is started from the initial point \mathbf{x} . Generally, g can be thought of as projecting \mathbf{x} to the nearest constrained local minimum.

We then modify the CMA algorithm so that it searches over $f \circ g(\mathbf{x}|f, c_1, \dots, c_n)$ instead of $f(x)$. As this search is essentially over the function defined by the basins of attraction for $g(\cdot|f, c_1, \dots, c_n)$, we term this optimization basin-CMA. This allows basin-CMA to leverage the local optimization's constraint handling behavior to quickly adapt to the constraint manifold. An illustration of this can be seen in figures 1 and 2.

In addition to changing the objective function over which we are searching, we achieve significantly greater efficiency by harmoniously altering the update equations. Given our elite sample points

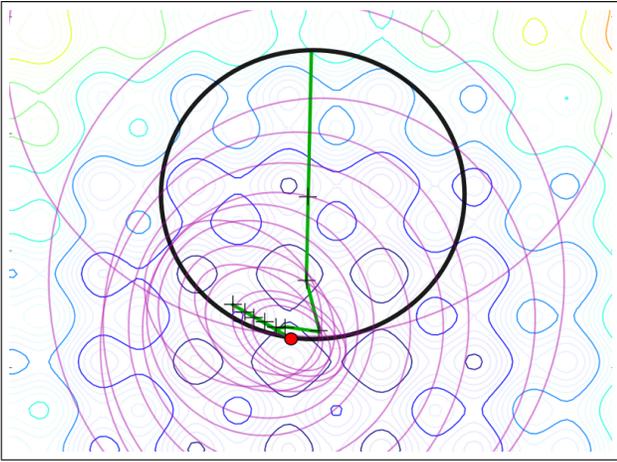


Figure 2: The progress of basin-CMA on finding the global optimum of a constrained minimization problem. The constraints enforce the solution to be on the unit circle, shown in black. The means follow a path displayed in green while the covariance matrices at each iteration are drawn in purple.

$\mathbf{x}_1, \dots, \mathbf{x}_\mu$, define $\mathbf{y}_j = g(\mathbf{x}_j | f, c_1, \dots, c_n)$ for $1 \leq j \leq \mu$. We then alter both the mean and covariance update equations (26, 27) by replacing each use of \mathbf{x}_j with \mathbf{y}_j .

Although this change is very simple, the resulting optimization is, in our experience, both surprisingly efficient and quite powerful. We have tested this method against the problems from the 2006 Congress on Evolutionary Computation (CEC-06) real-parameter constrained optimization competition. The problem set consists of 24 non-convex constrained optimization problems ranging from 2 to 24 dimensions, all of which exhibit local minima. We found that basin-CMA was able to match the ability to find the global constrained optimum of the best of the other entrants while running faster than the competition’s best entrant for each problem by an average factor of 51 times if derivative evaluations are as cheap as function evaluations and 6.3 times if finite differences are used for all derivatives. Full details of these tests are given in appendix A.

4.1.2 Application to gait optimization

We use basin-CMA as the outer loop in our optimization and employ it in handling non-differentiable or poorly conditioned terms such as the foot contact timings. In doing so, however, we must define the local optimization function, $g(\mathbf{x}_j | f, c_1, \dots, c_n)$ for $1 \leq j \leq \mu$, appropriately so that we do not pass the variables dictating the contact times to the spacetime optimization, which has no means to handle them.

We define g by labeling each variable in the optimization with one of three labels dictating its use as excluded from the local optimization, included in both the CMA and local optimizations, or used in the local optimization only. This labeling gives us flexibility both in excluding variables from the local optimization which it cannot handle properly, and in reducing the dimensionality of the problem solved by the basin-CMA outer loop in the case that the local optimization can handle some variables well when given only their default values. We typically only include the foot timings in CMA while excluding the morphology, pose, passive element, and ground force variables. This means that the outer loop normally only has to optimize over three to ten dimensions. If needed, however, the pose and morphology variables can be managed in the outer loop as well, although the resulting basin-CMA optimization will generally have

several hundred dimensions and converge somewhat more slowly.

In order to initialize the optimization we choose a Gaussian distribution wide enough so that it covers the entire allowed parameter space of the CMA variables. In our case the mean is set to a default in which all variables are zero except for the foot timings, which are set to be evenly spaced within a one-second period gait, and the torso height which is set so that the feet are level with the ground. The covariance matrix is diagonal with each element being the maximum distance from the mean to one of the two bounds in the corresponding dimension (or 500 if the dimension is partially or fully unbounded). In the local optimization the variables not dictated by CMA (i.e. marked as “local optimization only”) are initialized to their default values. So long as the initial CMA distribution is wide enough to provide good coverage of the allowed parameter space many other initialization schemes would give equivalent results.

5 Implementation

The definition of our spacetime optimization is implemented in Python. Using an operator overloading scheme similar to that in [Gunter 2007] we build a function composition graph for these terms and compute its derivatives. We then automatically generate C++ code to calculate these terms. This approach yields relatively high-speed results and allows us to analytically compute the sparsity pattern of the Jacobian of an optimization’s constraints. We then solve this spacetime optimization problem using the SQP based nonlinear programming package SNOPT [Gill et al. 2005].

Running times for a single spacetime optimization run from around a minute for very simple animals such as a simplified biped creature to 10-20 minutes for more complicated ones such as a horse. In the Basin-CMA computation these spacetime optimizations are run in parallel on a cluster to speed up the time needed to evaluate the samples in each iteration. We use the CMA parameters of $\lambda = 96$, $\mu = 32$, and $c_{cov} = 0.3$ and find that the problem is generally close enough to convergence for the solution to be useful at around 50 iterations.

6 Results

We have tested our methods on models for five different animals: a monopod, simplified biped, velociraptor, horse, and a pentaped, with 10, 16, 33, 29, and 32 DOFs respectively. All optimizations are fully 3D and we use 30 frames to sample the gait cycle. When optimizing for morphology we keep the torso, neck, and head size of the animals fixed and allow the proportions of the other limbs to vary. The morphological degrees of freedom are parametrized to preserve left-right symmetry, but we do not enforce any symmetry on the gait itself.

In all cases we have observed that our local spacetime optimization can solve for a reasonable gait giving just a single static pose and the foot contact times. Including the morphological DOFs sometimes causes the spacetime optimization to fail on more complicated skeletons, but this is easily fixed when necessary in our approach by having Basin-CMA determine the morphological terms, holding them fixed in each sampled spacetime optimization. We have also observed that multiple runs of basin-CMA generally converge to similar gaits, or to one of a set of reasonable gaits, for instance running versus hopping.

We have further observed successful determination of foot contact timings for our animals. This includes automatic determination of a walk/run gait depending on the desired speed as, well as picking gaits which effectively balance the animal’s mass between the feet of both the horse and pentaped animals. Examples of these

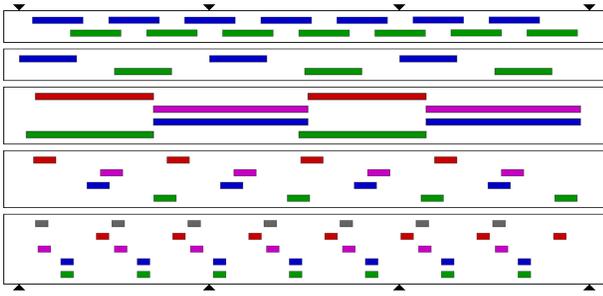


Figure 3: Examples of the foot contact timings resulting from our method. From top to bottom: simplified biped at $0.7 \frac{m}{s}$, simplified biped at $3.0 \frac{m}{s}$, horse at $1.0 \frac{m}{s}$, horse at $10.0 \frac{m}{s}$, and pentaped at $4.0 \frac{m}{s}$. The tic marks on the top and bottom are at one second intervals.

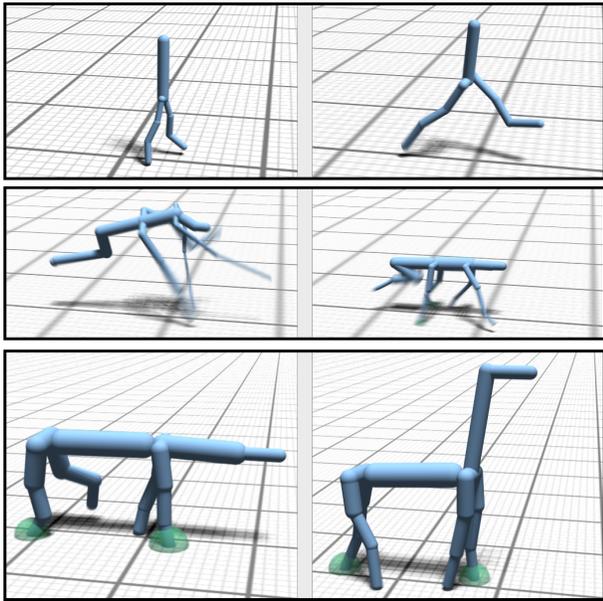


Figure 4: Some examples showing varying morphologies. From top to bottom: same contact times but different speeds, same speed but different contact times, and a user constraint setting the minimum height of the head.

foot timings are shown in figure 3. We have also observed successful adaptation of morphology to tasks such as moving at different speeds, maintaining different gaits, and keeping the head above a certain height. Some examples of results of our approach are shown in figure 4.

Our approach does not yet fully capture all the details exhibited in animal motions, and we were not able to obtain, for example, a gallop gait for the horse – getting instead a four-beat phase shifted trot. We suspect that this is due to our incomplete modeling of biomechanical elements such as an animal’s preference for using some muscles over others, robustness, and complexity of control. Nevertheless our method can easily be used to optimize over a reduced subspace of all possible foot timings, so an artist can design some aspects of a gait they desire and our method can search within this for an optimal gait. We have also found that for low-energy motions the gaits for the horse sometimes exhibit limp-like asymmetries. We suspect that this is a result of a flatter energy landscape with many different motions that result in very similar energies, and it is

worth noting that such low-energy problems have been a consistent trouble for spacetime constraint approaches.

7 Conclusion

We have shown a method for generating optimal gaits and morphologies for an animal without requiring a starting motion or foot contact timings. This allows us to automatically animate real creatures as well as those which are extinct or entirely imaginary. We achieved this through an efficient hybridization of a spacetime constraints optimization with a variation of the derivative-free optimization technique of covariance matrix adaptation. The gaits and morphologies produced are lifelike and exhibit many qualitative traits seen real animals.

We feel that future variations of our method could benefit from a more accurate biomechanical model. In particular, some constants and functions in the optimization determining the weighting between various factors were set by hand. A particularly interesting possibility for determining these tradeoffs would be to employ inverse optimization techniques such as those used to find the values for passive elements by [Liu et al. 2005]. Modeling muscles explicitly and optimizing over their properties could also serve to increase the biomechanical realism of the results.

Another entirely different avenue for future work would lie in designing tools to allow artist interaction with our method. Although the fully automatic nature of our method is desirable in some circumstances, it would also be useful to allow an artist to specify some aspects of a gait or form and then refine these while computing the terms which were not specified.

Ideally we would like to be able to automatically reproduce all of the many motions seen in nature given only simple information about the physical structure of the animal. Part of this could even involve determining probable forms and motions for variations on this animal depending on the particulars of the environments they inhabit. Although this full vision remains some distance off, we hope our approach provides a valuable step towards it.

Acknowledgments

The authors would like to thank Adrien Treuille for his many suggestions and implementation of CMA, Erik Andersen for creating the video, and the anonymous reviewers for their helpful comments. This work was supported by the UW Animation Research Labs, NSF grant HCC-0811902, Intel, and Microsoft Research.

References

ALEXANDER, M. 1996. *Optima for Animals*. Princeton University Press.

AUSLANDER, J., FUKUNAGA, A., PARTOVI, H., CHRISTENSEN, J., HSU, L., REISS, P., SHUMAN, A., MARKS, J., AND NGO, J. T. 1995. Further experiences with controller-based automatic motion synthesis for articulated figures. *ACM Transactions on Graphics* 14, 4 (Oct.), 311–336.

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

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.

GRZESZCZUK, R., TERZOPOULOS, D., AND HINTON, G. 1998. Neuroanimator: Fast neural network emulation and control of physics-based models. In *Proceedings of SIGGRAPH 98*, Computer Graphics Proceedings, Annual Conference Series, 9–20.

GUENTER, B. 2007. Efficient symbolic differentiation for graphics applications. *ACM Trans. Graph.* 26, 3, 108.

HÄKKINEN, K., AND KESKINEN, K. L. 2006. Muscle cross-sectional area and voluntary force production characteristics in elite strength- and endurance-trained athletes and sprinters. *European Journal of Applied Physiology* (Apr), 215–220.

HANSEN, N., AND KERN, S. 2004. Evaluating the CMA evolution strategy on multimodal test functions. In *Parallel Problem Solving from Nature PPSN VIII*, Springer, X. Yao et al., Eds., vol. 3242 of LNCS, 282–291.

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 SIGGRAPH 97*, 153–162.

HUTCHINSON, J. R., AND GATESY, S. M. 2006. Dinosaur locomotion: Beyond the bones. *Nature* 440, 7082 (Mar), 292–294.

KOH, B.-I., REINBOLT, J. A., GEORGE, A. D., HAFTKA, R. T., AND FREGLY, B. J. 2008. Limitations of parallel global optimization for large-scale human movement problems. *Medical Engineering and Physics*.

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

LIU, Z., GORTLER, S. J., AND COHEN, M. F. 1994. F: Hierarchical spacetime control of linked figures. In *In Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, ACM Press, 35–42.

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.

LIU, C. K., HERTZMANN, A., AND POPOVIĆ, Z. 2006. Composition of complex optimal multi-character motions. In *SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 215–222.

PAUL, A., AND BONGARD, J. C. 2001. The road less travelled: Morphology in the optimization of biped robot locomotion. In *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2001)*, IEEE Press, 226–232.

POPOVIĆ, Z., AND WITKIN, A. 1999. Physically based motion transformation. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 11–20.

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

ROSE, C., GUENTER, B., BODENHEIMER, B., AND COHEN, M. F. 1996. Efficient generation of motion transitions using spacetime constraints. 147–154.

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 *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 15–22.

SRINIVASAN, M., AND RUINA, A. 2006. Computer optimization of a minimal biped model discovers walking and running. *Nature* 439, 7072 (Jan), 72–75.

THOMPSON, D. W. 1992. *On Growth and Form: The Complete Revised Edition*. Dover.

VAN DE PANNE, M. 1996. Parameterized gait synthesis. *IEEE Comput. Graph. Appl.* 16, 2, 40–49.

WEIBEL, E. R., TAYOR, C. R., AND BOLIS, L. 1998. *Principles of Animal Design*. Cambridge University Press.

WITKIN, A., AND KASS, M. 1988. Spacetime constraints. In *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 159–168.

A CEC-2006 problem results

This table summarizes the results of our method when applied to the CEC-06 constrained real parameter optimization competition. The efficiency of our method in finding each problem’s global minimum is measured by the ratio of the number of function evaluations of the top scoring CEC contestant over that required by basin-CMA. Thus a ratio of 10 means that our method required ten times fewer function evaluations. Note that we select the best CEC contestant on a *per problem* basis, so no single algorithm in the competition would rate as well against ours and the table indicates. Since we use derivatives in our approach we provide two efficiency ratios. fevals ratio₁ gives a best case ratio where derivative solutions only count for addition evaluation. fevals ratio₂ gives a worst-case ratio when finite differences is used.

#	dimension	opt?	fevals ratio ₁	fevals ratio ₂
1	13	yes	105.970464	7.569319
2	20	yes	10.735468	0.511213
3	10	yes	20.546281	1.867844
4	5	yes	29.500000	4.916667
5	4	yes	59.019391	11.803878
6	2	yes	24.537736	8.179245
7	10	yes	89.488215	8.135292
8	2	yes	0.269050	0.089683
9	7	yes	6.083616	0.760452
10	8	yes	11.086012	1.231779
11	2	yes	12.345679	4.115226
12	3	yes	16.769231	4.192308
13	5	yes	29.487110	4.914518
14	10	yes	39.905063	3.627733
15	3	yes	40.223077	10.055769
16	5	yes	4.163090	0.693848
17	6	no	40.748068	5.821153
18	9	yes	109.538760	10.953876
19	15	yes	19.098863	1.193679
20	24	no	–	–
21	7	yes	26.338387	3.292298
22	22	no	–	–
23	9	yes	442.150171	44.215017
24	2	yes	2.457534	0.819178