Footprint-based Quadruped Motion Synthesis

Nick Torkos
Michiel van de Panne
Department of Computer Science
University of Toronto[1]

Abstract

This paper applies trajectory-based optimization techniques to the synthesis of quadruped motions. The animator specifies hard constraints, consisting of footprint locations and their timings, and soft constraints that encode both physically-plausible behavior and the notion of comfortable positions. By dealing first and foremost with the spline trajectories representing the gross motion of the quadruped, the resulting optimization problem can be solved efficiently and robustly. Results include walking, jumping, and galloping quadrupeds.

Résumé

Nous présentons une technique pour la synthèse des mouvements de quadrupèdes par optimisation de trajectoires. Les animateurs utilisent des contraintes rigides pour la position dans l'espace et le temps des traces de pieds, et les contraintes douces pour les lois physiques et la notion de positions confortables. La technique s'occupe principalement de la synthèse d'une trajectoire pour le centre de gravité, ce qui produit un algorithme efficace et robuste. Nos résultats comprennent des quadrupèdes qui marchent, sautent, et galopent.

Keywords: *physically-based animation, kinematics, motion modeling, optimization, legged locomotion.*

1 Introduction

Tools which automate the creation of animated legged locomotion have long been proposed and a few are now available commercially. However, the problem of legged locomotion is ill-posed. As a result, existing methods employ varying degrees of animator control over the motion and may or may not take the physics of motion into consideration. The method presented here builds on a combination of several existing techniques and ideas in order to yield a novel and flexible motion synthesis tool for animating quadruped motion.

Our system represents motions using a set of trajectories which describe the state of the quadruped over time. These trajectories are then optimized to satisfy a variety of constraints. The constraints include footprint positions, physics, and stylistic considerations. While the use of a trajectory-based representation of the problem is shared with previous work on *spacetime constraints[18]*, other aspects of our approach differ in order to successfully apply this problem representation to complex figures such as quadrupeds. Most notably, we restrict ourselves to optimizing the motion trajectories of two key mass points, as opposed to optimizing the motion trajectories of all the degrees of freedom. This optimized pair of motion trajectories then defines the basic structure of the motion, around which the remaining details are filled in.

The use a simplified model has several advantages. First, optimizing the family of trajectories which describe the complete detailed motion of a quadruped is a daunting task and has thus far been infeasible. Our system avoids some of the difficulties by virtue of the simplified model. We also afford more freedom in the design of the optimization functional by not requiring expressions for the first and second partial derivatives, thereby obviating the need to generate these complex expressions. Second, local minima are less likely to occur with a smaller optimization problem. Our optimization procedure optimizes the gross motion alone, relying on other techniques to produce the detailed motion. Third, the criteria which govern the realism of the gross motion and fine motion details are potentially very different. To take a leaping dog as an example, the basic laws of physics are of primary importance in constraining the gross motion of the center of mass, while other factors such as the kinematics of the hind legs are of key importance in the synthesis of joint motion within the legs.

An animator provides two types of input to our system. The first is the specification of the quadruped footprint locations and their timing. These act as hard constraints in the optimization process and their

---

[1] `{torkos|van}dgp.utoronto.ca`

use is motivated in part by earlier work on extracting gait information from dinosaur footprints[1]. The second type of input consists of stylistic hints which we call *comfort constraints*. These are treated as soft constraints in our system, as are the laws of physics. Taken together, the hard and soft constraints govern the shape of the gross motion of the quadruped. The results we present explore specific examples of how these types of constraints can be used to reconstruct a desired quadruped motion.

A feature of our method is the capability to stretch the laws of physics if necessary. This is accomplished by treating both physics and comfort as soft constraints, thus allowing for compromise solutions which can balance style requirements and physical plausibility.

The remainder of this paper is divided into six sections. We first review previous work in Section 2. An animator begins to create a motion by specifying the required footprints, the details of which are presented in Section 3. The first step in creating a motion fitting the desired footprints uses trajectory optimization to satisfy the desired constraints. This is described in Section 4. The second step in creating a motion is *motion reconstruction*, which is explained in Section 5. Our results are given in Section 6. Finally, we present our conclusions in Section 7.

2 Related Work

One logical way of animating the motion of a quadruped is to use a physically-based simulation. The animation problem then becomes one of controlling the simulated quadruped to make it perform the desired motion. Control is a difficult problem, however, because simulated muscles provide a very indirect means of controlling the final motion. Nevertheless, a growing body of work is built around the use of fully-dynamic forward simulations and appropriately-designed control schemes. Representative examples of this approach as applied to animating quadrupeds include the work of Raibert and Hodgins[12], van de Panne[16], and Kokkevis et al.[9]. Developing control models which allow for more animator control remains problematic with this type of approach.

Another body of work has approached quadruped animation using refined sets of prescribed kinematic behaviors which are used to directly synthesize the motion. The work of Girard[7, 8] has successfully used this approach and has been extended into a commercial product[14]. Most notably, this product makes use of footprints as part of the motion specification. Our work is similar in spirit and was partly motivated by the success of Girard's early work. However, we choose to synthesize the key aspects of the motions through optimization instead of sets of rules and forward simulation. We optimize the motion on a global scale, thereby allowing for extensive anticipation and follow-through motion, and allowing the laws of physics to be stretched. Other interesting algorithms for kinematic and hybrid kinematic/dynamic work have been explored[2, 3, 4, 5], although these techniques are not shown to be generalizable to quadrupeds.

The concept of trajectory optimization for animation can best be traced back to the *spacetime constraints* work of Witkin and Kass[18]. This area has subsequently been explored in greater depth by Cohen[6], Liu and Cohen[11], Rose et al.[13] and others. Our work focusses on how to make a trajectory optimization approach practical for the problem of quadruped animation. Trajectory optimization is a general way of framing motion synthesis problems, but there are a great number of choices that need to be made in order to make it workable for a particular problem. Distinctive features of our work include the separation of the gross character motion from the motion detail, our use of footprint and comfort constraints, and the useful compromises that can be reached as a results of treating physics and style considerations as soft constraints. Lastly, the implementation of the physics constraints is quite different from those previously used in trajectory optimization methods.

The quadruped work in this paper builds on previous work using footprint-based techniques for biped animation[17]. The quadruped model is considerably more complex than the previous biped model in all respects: footprint specification, gait types, trajectory optimization incorporating a model of a flexible back, and a more difficult motion reconstruction problem. Given that few animation techniques have been shown to apply to both bipeds and quadrupeds[2], the extension of this previous biped work to quadrupeds has been an informative challenge[15].

3 Footprints

We begin the description of our system by describing the principal input provided by an animator, which consists of footprint position and timing information.

3.1 Footprint Model

A footprint is a six-tuple $< i, x, z, \theta, t_1, t_2 >$, where i represents the foot ($i \in \{1, 2, 3, 4\}$ for a quadruped),

---

[2] [7, 12] are notable exceptions.

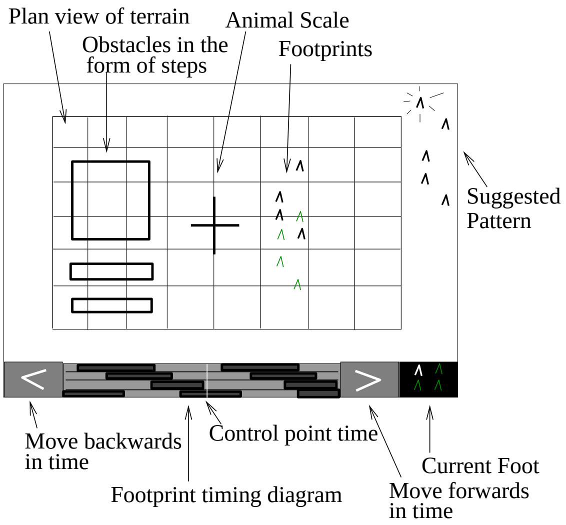


Figure 1: User interface for manual footprint entry.

x and z indicate the position of the foot on the terrain, θ indicates the orientation on the terrain, and t_1 and t_2 are the time of contact and time of departure, respectively, of the foot with the ground. The height, y, and the resting pitch of the foot on the terrain are implicitly specified by the terrain surface.

3.2 Direct Positioning of Footprints

Our user interface is illustrated in Figure 1. The complete environment is seen from a plan view. An indication of which quadruped leg is to be associated with the current footprint is given in the lower right corner of the screen. In the upper right corner, a sample template suggests a reasonable position for the new footprint, given the current choice of gait.

Screen clutter is a potential problem with using footprints, as the hind footprints tend to overlap the front ones. We minimize this through the use of colour coding, as well as making the most recent footprints stand out using bold lines. A cross in the centre of the screen provides a reference scale. The dimensions of the cross correspond to the length of the animal. Finally, a timing diagram is located at the bottom of the screen which can be used to directly alter the timing information at any time.

3.3 Timing and Gaits

The timing associated with a footprint is dependant on the desired gait and can be created in several ways. An animator can choose to exercise full control by specifying the timing information directly. An alternative solution we provide is to let the system guide the user based upon a choice among three the gaits which our system currently supports: the walk, trot, and gallop. The sequencing and timing of the leg stance phases are then controlled automatically in accordance with the desired gait. The animator retains responsibility for specifying the position and orientation of the feet.

3.4 Footprints Along a Path

Higher level motion descriptions can be used to generate the footprint data if further automation is desired. A simple and effective abstraction uses splines to model the path of the animal and an automatic process to generate a reasonable set of footprints along the path. An example of this is shown in Figure 2, where the footprints of a quadruped are placed along a curve defined by Catmull-Rom spline segments.

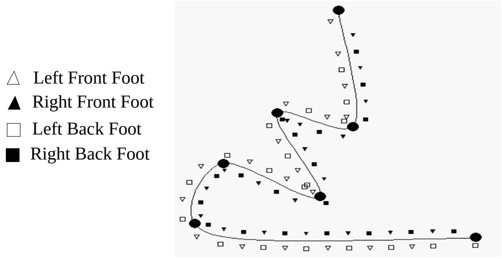


Figure 2: Footprint generation along a curved path.

3.5 Autonomous Motion

Completely autonomous motion is useful in interactive games and virtual worlds where the environment is unpredictable. Our animation system has an interactive mode of operation where new footprints are generated on the fly. As each footprint is placed, the motion of the quadruped is synthesized at interactive rates. Our automated footprint procedure currently implements a random wandering behavior. Incorporating goals and obstacle avoidance is an obvious next step. A selling feature of footprint constraints is thus their utility in building high level motion planners with little effort.

4 Trajectory Optimization

This section explains the first step in our motion synthesis algorithm: *trajectory optimization.* Trajectory optimization allows us to efficiently generate the underlying motion of a simplified approximation of the quadruped's body over the footprints. We reconstruct the rest of the motion afterwards using the methods described in Section 5.

4.1 Overview

Trajectory optimization determines the overall motion of the quadruped's body. It is assumed that the motion of the head, the swing phase of the legs, and the swing of the tail are visually important, but that these have a minimal effect on the gross motion of the animal.

The optimization process is based upon minimizing the integral of an optimization functional over the duration of the trajectory. This integral is approximated numerically as a sum using evenly-spaced sample points in time over the motion trajectory. Given a point in time on the motion trajectory, the functional evaluates the fidelity of the physics and the compliance with the style constraints. The former involves computing the current accelerations of mass points, which in our case can be analytically evaluated from the spline-based trajectories. The accelerations are then used to compute the extent of any violations of the basic laws of physics. The gross position of the quadruped, combined with the known footprint supports provide the basis for evaluating the style constraints. With this intuition, we now describe the process in more detail.

4.2 Quadruped Model

In Figure 3 we show the full skeletal quadruped model used for display purposes, indicating all the joints and skeletal links in the model. Aside from the single degree-of-freedom (d.o.f.) joints in the legs (excluding the hips), all the joints are assumed to have three d.o.f. This model has all the degrees of freedom necessary for a reasonably sophisticated animation, but we consider it too complex to animate directly using optimization techniques.

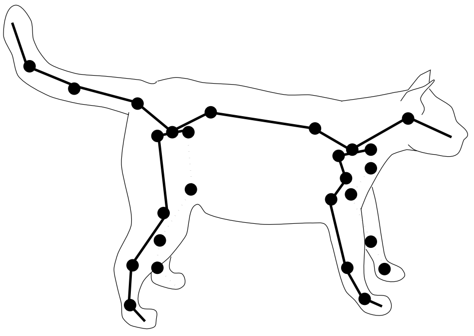


Figure 3: Initial Model of Quadruped.

4.2.1 Approximate Quadruped Model

Our simplified model consists of two point masses, connected by a spring which serves to model the internal forces of the back. This model is shown in Figure 4. The position of the stance feet also play an important role, as shall be described shortly.

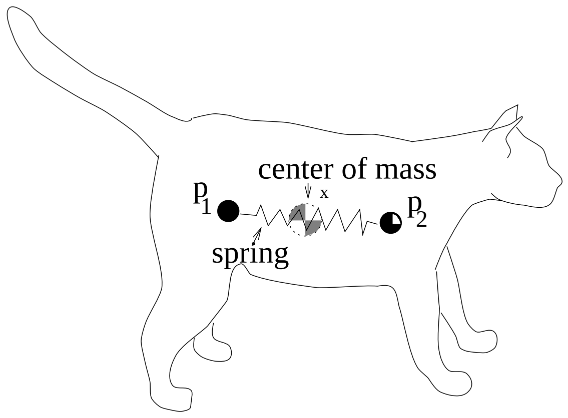


Figure 4: The simplified physical model.

4.3 Trajectory Representation

The trajectories of the two point masses are represented using C^2 piecewise-continuous spline curves in order to yield smooth motions for the respective masses. The assumption used to justify the smoothness is that the legs are assumed to be effective at intermediating between the body motion and the ground. The number of spline segments used in modeling the trajectory is an important choice to make. Too few spline segments results in the motion being overly constrained and smooth. Too many spline segments can slow the optimization process unnecessarily and lead to an increased number of undesireable local minima. In practise, the use of one or two spline segments per stride[3] is a reasonable choice. It should be noted that our representation does not use a separate curve to offer an independant parameterization of the trajectory speed, as is common in some keyframe systems.

4.4 Optimization Functional

The optimization enforces the soft constraints, namely the physics and style. We denote the optimization functional as $E(V)$,

$$(1) \quad E(V) = \int_0^T \left(E_{physics}(V) + E_{comfort}(V)\right) dt.$$

where $V = \{V_1, V_2, \ldots, V_n\}$ represents the trajectories of P_1 and P_2, n is the number of spline segments,

---

[3] A stride is defined as a complete gait cycle.

V_i is the set of free parameters for curve segment i, and T is the duration of the animation. $E_{physics}$ incorporates the physics constraints. $E_{comfort}$ keeps the animal from contorting into awkward positions by penalizing trajectories which violate stylistic constraints.

4.5 Physics Energy

$E_{physics}$ is composed of three terms: one for each of $\mathbf{p_1}$ and $\mathbf{p_2}$, and one for the center of mass. We assume that the hind legs support the back half of the body, $\mathbf{p_1}$, the front legs support the front half of the body, $\mathbf{p_2}$, and that all of the legs support the center of mass. The position, velocity, and acceleration of the center of mass are known at any point in the optimization process as they are computed directly from the spline trajectories for the two point masses. The key to evaluating the physical plausibility of the motion at a particular point in time lies in determining whether the current acceleration achieved by a point mass is indeed plausible, based upon the current configuration of the points of support.

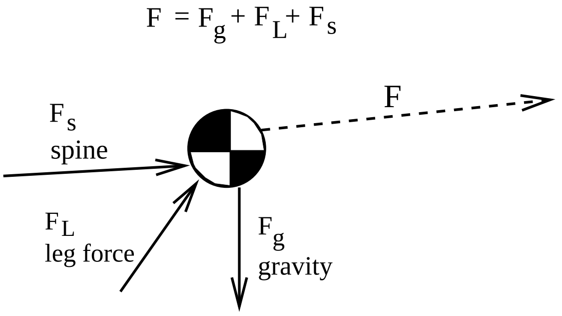


Figure 5: Forces acting on a point mass.

In Figure 5 we show all of the forces active on one of the point masses in our model. Even though the magnitude of the leg forces are not known, we shall make use of constraints on their line of action in order to evaluate the plausibility of the physics. Using the forces shown in this diagram, the general procedure to determine the physics energy term for a mass point is:

1. compute the acceleration $\mathbf{a}$ from the trajectory
2. compute the leg force, $\mathbf{F_L}$
3. determine the feasible and infeasible components of $\mathbf{F_L}$
4. calculate a penalty contribution for the infeasible components

In step 2, the total force applied on the point mass, $\mathbf{F}$, is determined as $m\mathbf{a}$. The force of gravity and the spring force, $\mathbf{F_s}$ are also known. For the center of mass $\mathbf{F_s} = \mathbf{0}$. For $\mathbf{p_1}$ the spring force is

$$\mathbf{F_s} = k_s(l - l_o)\frac{(\mathbf{p_2}(t) - \mathbf{p_1}(t))}{||(\mathbf{p_2}(t) - \mathbf{p_1}(t))||}, \tag{2}$$

where k_s is a spring constant, l is the distance between the two points, and l_0 is the rest length of the spring. The spring force for $\mathbf{p_2}$ is equal to that of $\mathbf{p_1}$, but in the opposite direction.

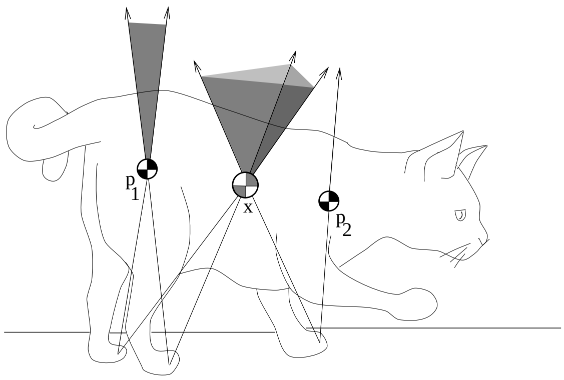


Figure 6: Example of plausible leg force regions on point masses.

The leg force, $\mathbf{F_L}$, remains the only unknown and can thus be solved for. However, a stance leg cannot simply exert an arbitrary force. Given a point mass and the position of its associated points of support, we can determine a range of feasible directions for leg forces. An example is shown in Figure 6. In this figure, the cat has three legs on the ground, which support the center of mass. Both of the hind feet support $\mathbf{p_1}$, while only one leg supports $\mathbf{p_2}$. This means that the regions of plausibility for the leg forces are bounded by three planes for $\mathbf{x}$, embedded in a plane and bounded by two lines for $\mathbf{p_1}$, and restricted to a line for $\mathbf{p_2}$. We can rewrite $\mathbf{F_L}$ as $\mathbf{F_L} = \mathbf{F_{Lp}} + \mathbf{F_{Ln}}$, where $\mathbf{F_{Lp}}$ is the component of the leg force that can be accounted for by our physical model, and $\mathbf{F_{Ln}}$ is the implausible force. We minimize the non-physical forces using a physics penalty term proportional to $||\mathbf{F_{Ln}}||^2$.

For a quadruped, there are five possible cases that we must treat in order to calculate the physics penalty term for a particular leg force. These cases are single, double, triple, and quadruple support, as well as flight. All five cases are relevant for the center of mass. For points $\mathbf{p_1}$ and $\mathbf{p_2}$, only the first three cases (flight, single-support, and double-support) are relevant because the forces coming from the other half of the body are already represented by the spring force, indicated in Figure 5. For flight phases, the applied leg force should be zero and thus any computed leg force is considered to be infeasible and its squared magnitude is contributed towards the physics penalty.

As a specific example, we illustrate the case for double support, shown in Figure 7. Because each leg can apply a force in the direction from the foot towards the point mass, the plausible region during double support consists of the wedge-shaped region bounded by the extensions of vectors $\mathbf{L_1}$ and $\mathbf{L_2}$. The component of the leg force not in the plane, $\mathbf{F_{Ln_1}}$, is added to the optimization functional. We also penalize trajectories which have leg force components, $\mathbf{F_{Ln_2}}$, which are on the plane, but outside of the feasible region. The implausible leg force components on point $\mathbf{p}$ are thus penalized as follows:

$$(3) \quad E_p = k_{p2}(||F_L \cdot N||^2 + \max(0, F_L \cdot N_1)^2 + \max(0, F_L \cdot N_2)^2),$$

where N_i is the vector on the plane normal to L_i that points outside of the region of plausibility, N is the normal to the plane, and k_{p2} is a weighting factor for the double support penalty.

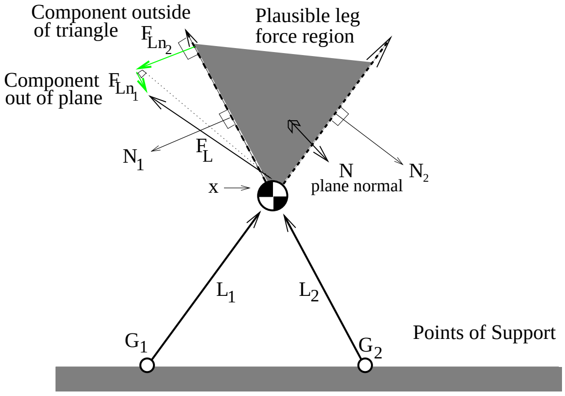


Figure 7: Double support.

The complete physics energy functional is then calculated as

$$E_{physics} = K_x E_x + K_{p1} E_{p1} + K_{p2} E_{p2},$$

where K_x, K_{p1} and K_{p2} are used to weigh the penalties of x, p_1, and p_2 respectively.

4.6 Comfort Energy

The physics energy guides the trajectory optimization to generate physically plausible motions. However, additional stylistic constraints are required to ensure that the motion is visually pleasing. To address this we employ a *comfort* term in the optimization. This term is partially motivated by previous work[10], although the definition of our comfort term is rather different. The comfort model that we use is simplistic for the time being, although its definition is open to more sophisticated uses. It encourages the legs to always maintain a comfortable length during their support phases. We use a simple set of virtual legs to evaluate this, as shown in Figure 8. A second component of the comfort model encourages the body to avoid collisions with the ground, as will be explained shortly.

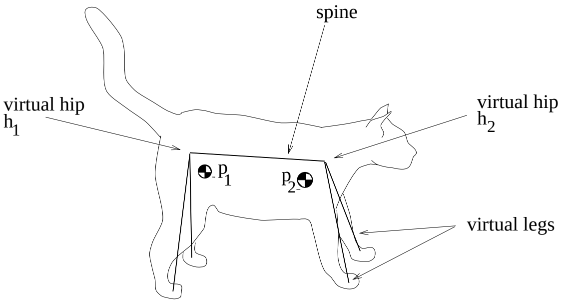


Figure 8: Comfort model.

4.6.1 Leg Length Penalty

The comfort constraints on the leg lengths keep the legs from becoming overly compressed or extended throughout the motion. We encourage this soft constraint with the term E_{lc}, which is calculated as

$$(4) \qquad E_{lc}(G_f, h, l) = K_{lc}(||G_f - h|| - l)^2,$$

where G_f is the foot position, K_{lc} is a constant used to weigh the comfort constraint, and h is the hinge that the leg attaches to. Each leg has its own leg length, l, to accomodate the creation of various walks, including limping or lopsided ones. The penalty for each leg is summed and is referred to as E_{leg}.

4.6.2 Body Collision Penalty

During ambitious ballistic motions, the body of the quadruped can potentially penetrate the ground. We therefore provide an additional term to discourage this behaviour. Our collision calculations are simple, but they are sufficient to eliminate most unwanted body collisions with the ground. We encourage each mass point to maintain a clearance distance r_i above the ground or an obstacle. If this is not the case, then we add a collision avoidance term, $E_{coll} = K_{bi}(p_{i_y} -$

$r_i - y)^2$ to the comfort model, where K_{bi} is a constant that weighs the strength of the collision term, p_{i_y} is the height of the point mass, p_i, and y is the height of the terrain below p_i.

The complete comfort energy functional is finally computed as

$$E_{comfort} = E_{leg} + E_{coll}.$$

4.7 Optimization Procedure

The energy functional used in the optimization is a numerical approximation of Equation 1:

$$E(V) \approx \sum_{i=0}^{n-1} \sum_{j=0}^{k} [E_{physics}(V_i(t_j)) + E_{comfort}(V_i(t_j))]$$

where n is the number of spline segments, k is the number of evaluations points per spline segment, $t_j = \frac{j}{k}$, and $V = \{V_1, V_2, \ldots, V_n\}$, where V_i is the set of free parameters for curve segment i. A greedy, deterministic algorithm is used to optimize the trajectories. A multi-scale approach is used to obtain fast convergence. We begin by initializing the trajectories to pass directly over the average position of the supporting footprints for each spline segment. Our implementation positions the center of mass at the average body height over the footprints, and then positions the p_1 and p_2 trajectories by assuming that the spine is parallel to the mean direction of the footprints. Then, using a coarse-to-fine approach, we perform fixed-step-size alterations to each of the free parameters in turn and reevaluate the functional for each such alteration. We retain changes which lead to lower values of the functional. When parameters can no longer be altered, or when the maximum iteration count is surpassed, the scale of the attempted alterations is halved and we optimize further. Because our splines exhibit local control, the evaluation of the optimization functional can be restricted to the locally affected areas.

5 Motion Reconstruction

The motion reconstruction stage involves building the motion of the complete skeleton from the point mass trajectories synthesized thus far. This process is broken up into steps that can be handled easily. We advocate using a hybrid application of kinematics, dynamics, and optimization techniques which best utilizes the strengths of each method. The choice of method depends on the animator's preference, the quality of animation desired, and the desired frame rate.

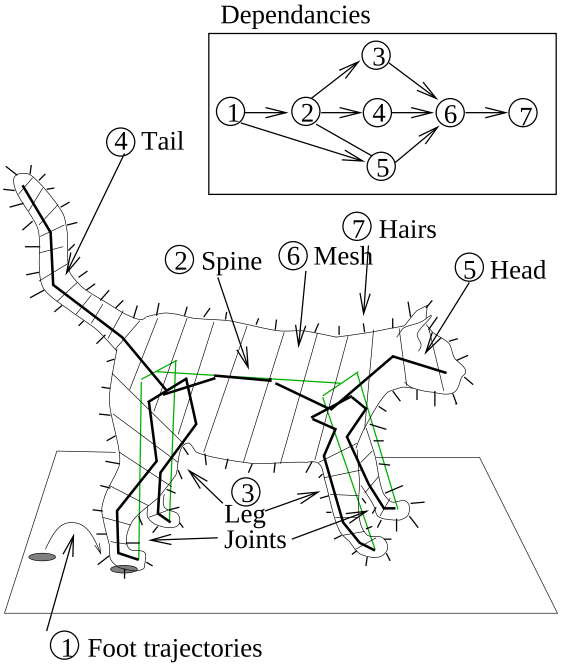


Figure 9: An example of motion reconstruction.

The reconstruction approach used in our prototype implementation is shown in Figure 9, where the numbers represent the ordering of the reconstruction steps, and the directed graph represents the dependencies between the steps. In this example, the reconstruction begins with an optimization procedure which determines the trajectories of each foot during its flight phase. The optimization provides a balance between keeping the foot at a reasonable distance from the body, minimizing added torque, and avoiding collisions of the foot. The technique used is the same as that employed for the earlier body trajectory optimization, although in this case only one spline segment is required for the trajectory of the foot from one footprint to the next[15]. Note that stylistic considerations are implicit in this step.

Next, we use inverse kinematics to determine the arch of the spine and to define the internal joint angles of the legs. More care is taken in the kinematics of the legs than that of the body because the legs are on the exterior of the body and are more visible. For greater realism and speed we employ an example-based inverse kinematics system (described in [15]), so that the inverse kinematics results closely match the joint angles of a pre-defined set of keyframes.

Procedural kinematics is used to pivot the head towards the direction of motion, while the tail is animated with passive dynamics. The use of different animation techniques for the head and tail is justified because the tail generally swings around and expresses the effects of passive dynamics, while the head usually leads the motion rather than following it. Finally, passive secondary dynamics is applied to the hairs which cover the complete animal.

6 Results and Discussion

Our experiments have provided promising results. The trajectory optimization and motion reconstruction have proved effective at generating a variety of motions, as shown in Figure 10. The footprints and optimized point-mass trajectories for a gallop are shown in (a). A frame from the corresponding animation is shown in (b).

Jumping on-to and off-of a ramp, as shown in (c) and (d) is an example which effectively illustrates both the physics and comfort constraints in action. The physics ensures for near-parabolic trajectories for the body during the jumps. The spline curves used to model the trajectories ensure smooth acceleration and deceleration for the mass points, ensuring for anticipation and follow-through. The comfort term in the functional ensures reasonable configurations for the animal throughout the motion.

Figure 10(e) shows an image from an animation illustrating autonomous planning and generation of the footprints. This type of autonomous motion plans a few steps, optimizes, displays the result, and then plans again. This happens at interactive rates, although the optimization needs to be overlapped with the motion playback to achieve an acceptably smooth real-time animation. Lastly, Figure 10(f) is included to show that we expect the technique to be generalizable to skating and sliding motions, although we have currently only implemented this for bipeds.

One potential pitfall of our technique is the presence of local minima in the optimization. This becomes problematic as the optimization problem is made more complex through additional mass points or additional spline segments. Coping with a larger number of free parameters requires more extensive searches in order to avoid local minima.

Further experimentation is needed to be able to determine the stylistic constraints which are necessary to tune the generated gaits to match specific gaits of specific animals. The physics constraints could also be embellished to take into account the magnitude of the leg forces and their stability.

In its present form, our system can be easily persuaded to produce unnatural motions in addition to the well-behaved cases described earlier. The motions produced are sensitive to the timing assigned to footprints, and thus bad input here will produce bad output. We see the need for more work on a front-end tool which generates appropriate timing information, footprint positions, and stylistic parameters as a function of more general motion descriptions, e.g., “Make this camel model do a slow camel walk along this path.”

7 Conclusions

The hybrid animation technique we have presented is unique in several ways. The use of comfort constraints in the context of a global motion trajectory yields a useful tool for shaping a motion, in conjunction with basic physics constraints. Footprints provide a convenient and readily understandable input technique which allows the animation tool to do an effective job of amplifying an animator’s intention into a complete motion. The ability to stretch physics is useful in that it allows for compromises between physics and stylistic constraints. The results show that a footprint-based system using trajectory optimization is capable of creating many varieties of quadruped locomotion.

References

[1] R. M. Alexander. How dinosaurs ran. *Scientific American*, pages 130–136, April 1991.

[2] L. Bezault, R. Boulic, D. Thalmann, and N. Magnenat-Thalmann. An interactive tool for the design of human free-walking trajectories. *Proceedings of Computer Animation ’92*, pages 87–104, 1992.

[3] R. Boulic, N. M. Thalmann, and D. Thalmann. A global human walking model with real-time kinematic personification. *The Visual Computer*, 6:344–358, 1990.

[4] A. Bruderlin and T. Calvert. Interactive animation of personalized human locomotion. *Proceedings of Graphics Interface*, pages 17–23, 1993.

[5] A. Bruderlin and T. W. Calvert. Goal-directed animation of human walking. *Proceedings of ACM SIGGRAPH*, 23(4):233–242, 1989.

[6] M. F. Cohen. Interactive spacetime control for animation. *Proceedings of SIGGRAPH ’92*, pages 293–302, 1992.

[7] M. Girard. Interactive design of computer-animated legged animal motion. *IEEE Comptuer Graphics and Applications*, 7(6):39–51, June 1987.

[8] M. Girard and S. Amkraut. Eurythmy (computer animation). 1989.

[9] E. Kokkevis, D. Metaxas, and N. I. Badler. Autonomous animation and control of four-legged animals. *Proceedings of Graphics Interface '95*, pages 10–17, 1995.

[10] P. Lee, S. Wei, J. Zhao, and N. I. Badler. Strength guided motion. *Proceedings of ACM SIGGRAPH*, pages 253–262, 1990.

[11] Z. Liu and M. F. Cohen. Keyframe motion optimization by relaxing speed and timing. *Computer Animation and Simulation '95, Proceedings of the Eurographics Workshop*, pages 144–153, 1995.

[12] M. H. Raibert and J. K. Hodgins. Animation of dynamic legged locomotion. *Proceedings of SIGGRAPH '91*, pages 349–358, 1991.

[13] C. F. Rose, B. Guenter, B. Bodenheimer, and M. F. Cohen. Efficient generation of motion transitions using spacetime constraints. *Proceedings of SIGGRAPH '96*, pages 147–154, 1996.

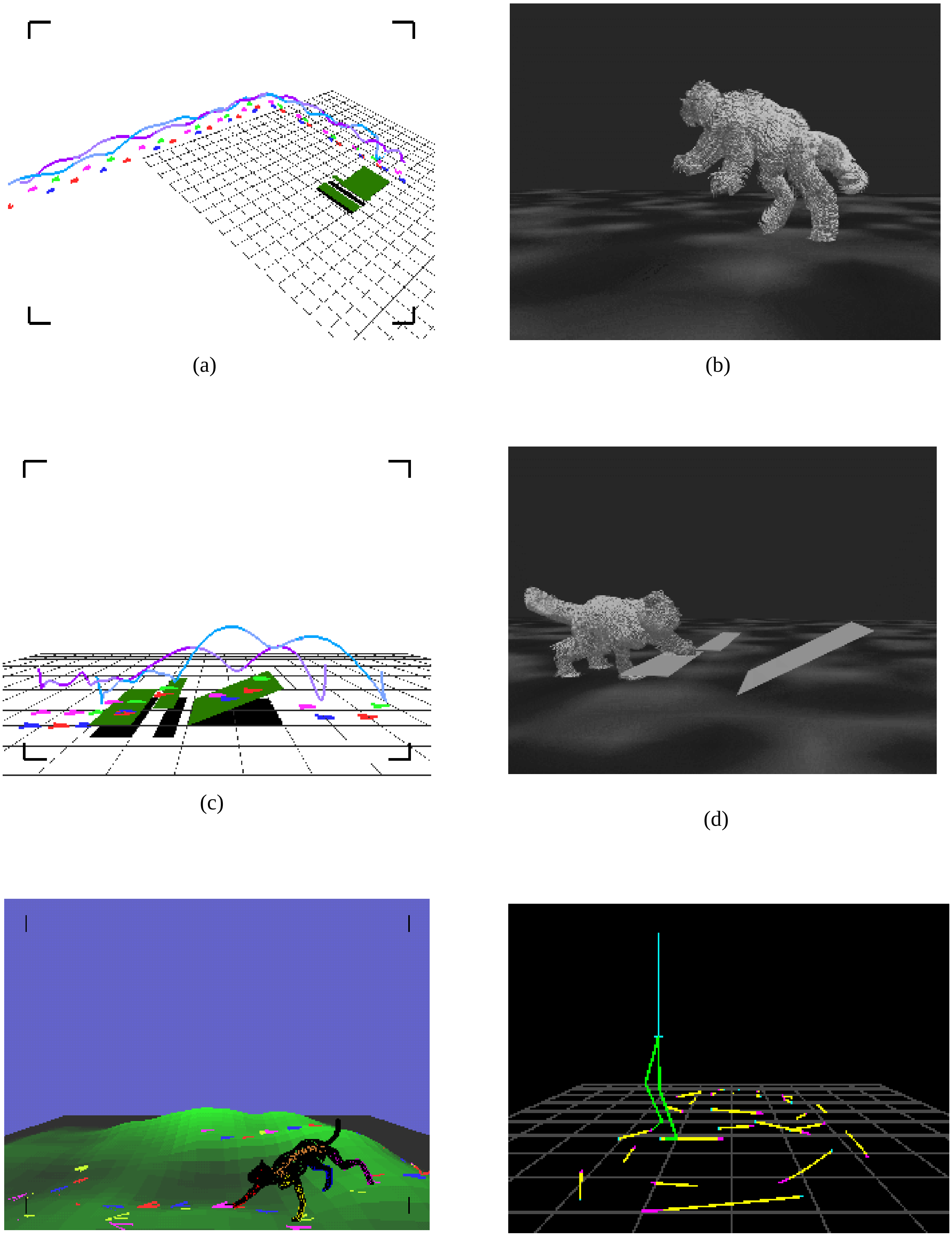
[14] Kinetix 3D Studio. "biped," part of character studio. 1995.

[15] N. Torkos. *Animating Quadrupeds with Footprints.* M.Sc. Thesis, Department of Computer Science, University of Toronto, 1997.

[16] M. van de Panne. Parameterized gait synthesis. *IEEE Computer Graphics and Applications*, 6(2):40–49, March 1996.

[17] M. van de Panne. From footprints to animation. *Computer Graphics Forum*, 16(4):211–223, 1997.

[18] A. Witkin and M. Kass. Spacetime constraints. *Proceedings of SIGGRAPH '88*, pages 159–168, 1988.



(a) (b)

(c) (d)

(e) (f)

Figure 10: Animation results