*Volume 16 (1997), number 4 pp. 211–223* COMPUTER GRAPHICS *forum*

# From Footprints to Animation

Michiel van de Panne

Department of Computer Science, The University of Toronto, Toronto, Ontario, Canada

**Abstract**

*A method of using footprints as a basis for generating animated locomotion is proposed. An optimization process is used to maximize the physical plausibility and perceived comfort of a motion which is constrained to match given footprint and timing information. The proposed method creates plausible walking, turning, leaping, and running motions for bipedal figures at interactive rates. Autonomous motions are generated using a planning algorithm which dynamically generates new footprints and the associated new motions.*

## 1. Introduction

Imagine walking through a forest and coming across a set of footprints in the mud. Such tell-tale traces of passage can reveal much information, such as the type and size of the animal, as well as hints about its pattern of movement. A key example of this has been the use of footprint analysis in estimating the gait characteristics of dinosaurs[1]. In this paper, we further explore the relationship between footprints and motion, with the goal of being able to synthesize complete motions from a set of footprints.

The animation technique we propose is formulated for bipedal characters and uses some timing information in addition to the footprint locations and orientations. Figure 1 shows the input data for a typical motion. The footprints in this case specify two small steps, followed by a hop (from footprint 3 to 4), two more small steps, a short time spent standing still on two feet (footprints 6 and 7), and a last small step.

The timing diagram shown in Figure 1(b) plays an important role in conjunction with the footprints. The timing information specifies when a given foot should be in contact with its assigned footprint. This is indispensible information, as we might want a character to remain standing in a particular place, at a particular time, and for a particular duration. The timing information is also important for disambiguating large walking steps and a running gait, both of which could conceivably fit the same set of footprints.

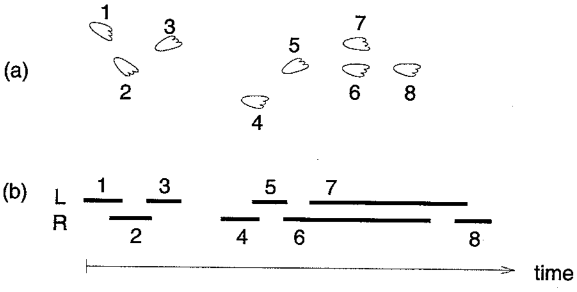

**Figure 1:** *Specification of a Walking Motion (a) plan view of footprints (b) timing diagram.*

The combined footprint-plus-timing motion specification is particularly useful when elements in the script or factors in the environment require synchronization. Explicit control over coordination and synchronization is a practical requirement for choreographing repeatable motions in computer animation[12]. Our subjective experiments to date suggest that footprints are an intuitive means for animators to specify a motion. A more objective result is that footprints form a useful intermediate representation of locomotion for path planning algorithms.

### 1.1. Previous Work

Both analysis and synthesis have been used in the past to help understand and characterize the motion of humans and animals. Motion analysis has led to the determination of general motion properties, an example of which is the *Froude number*[1]. It has also led to specific notations for human movement, such as *Labanotation*[4]. In contrast, generating animation is a synthesis problem, which has a more recent history.

Effective tools for generating walking and running motions have long been of interest in animation. The applied techniques can be loosely classified as being kinematic, dynamic, or partly-dynamic in nature. What follows is a brief overview of previous work in each of these three categories. The interested reader is also referred to more general introductions to the issues of motion control in animation[2, 3].

Kinematic animation techniques attempt to directly capture the structure of motion using a set of rules which encode knowledge about the motion. *A priori* knowledge about motions can come from many possible sources: motion capture data, emperical observations, knowledge of the skeletal geometry, etc. The motions are typically expressed in terms of parameters directly useful to the animator, namely walking speed, hip tilt, turning speed, etc. Kinematic models have proved particularly adept at reproducing specific cyclic classes of motion such as human walking and running gaits with a good degree of fidelity[3, 7, 8, 9]. Kinematic techniques for finding statically-stable poses using knowledge of the points of support have also been proposed[5]. Another line of research examines how real motion data can be manipulated and seamlessly blended together in order to create desired motions[10, 17, 26, 31]. However, it has not yet been shown how to build general and flexible models of locomotion using motion capture data.

An alternative to directly modeling motion is to reconstruct a motion from its constituent components, namely sets of muscles which exert forces on a physical skeleton. The motion of the skeleton is subsequently dictated by physics. Attempts to control physically-based simulations or real walking robots have met with some success[16, 19, 22, 23, 28, 29], although the control techniques do not yet provide the general solutions and animator control which are necessary for them to be practical for wide-spread use.

Our motion model makes use of dynamics in a limited way, making the assumption that most of the mass of the animated character is in the body and relatively little in the legs. Similar assumptions are the basis of well-studied algorithms which can be used to control hopping and running[23]. Recent extensions show that this assumption can be weakened to allow for the successful simulation of fully dynamic human running[15, 16]. The relative disadvantage of forward-simulation approaches is that they do not allow animators to place tight constraints on the motion, such 'place your foot here' or 'be here at this time'.

A variety of interesting compromises can be struck between kinematic and simulation-based techniques. The essence of all such compromises is to add dynamic considerations back into a kinematic representation of the motion. One means of doing this is to optimize all aspects of a motion trajectory subject to user-specified criteria and a user-specified set of constraints[30, 11, 21, 24]. The technique we propose similarly relies on the optimization of a motion trajectory, but in a setting specific to bipedal walking and running.

Our work makes use of some insights first proposed by Girard[13, 14], which has been extended to allow the specification of a motion through the use of footprints[25]. This previous work makes use of a multi-pass process for determining a body motion which best fits a set of footprints. The vertical body motion is determined using ballistic motion during the flight phase and a fixed family of functions during support phases. The horizontal motion is computed independently using a velocity-error feedback loop. Turning and banking behavior are added separately once the translational motion is known.

Although our formulation also makes use of footprint position and timing information, it is different from Girard's work in several significant ways. First, our formulation collapses the multitude of constraints and rules into a single compact optimization problem. Second, our formulation performs a global optimization of the center-of-mass trajectory. This allows a footprint which is two or three steps into the future to influence the motion of the current step, allowing for extensive anticipation. Third, we explicitly allow for the laws of physics to be compromised while still generating an otherwise plausible motion.

Our work also implements a high-level planner which dynamically plans the placement of footprints. Similar planning algorithms have been proposed in the past[3, 6]. The purpose of our implementation is to show that our motion synthesis algorithm can also automatically generate motions at interactive rates.

### 1.2. The Proposed Method

The foundation of the proposed method is to represent the motion solely in terms of a center of mass (COM) trajectory which itself is synthesized from the footprint information. The trajectory of the center of mass must satisfy several properties: (a) the COM trajectory varies smoothly; (b) the COM only accelerates as allowed by the supporting leg(s) and the laws of physics; and (c) the animated character must be in 'comfortable' configurations throughout the motion. Objective (a) is satisfied implicitly by modeling the COM trajectory using a series of spline curves. Objectives (b) and (c) are satisfied explicitly by optimizing the COM trajectory.

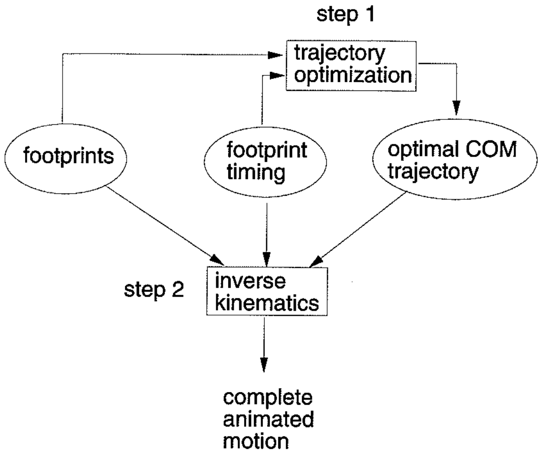

**Figure 2:** *Block diagram for motion synthesis from footprints.*

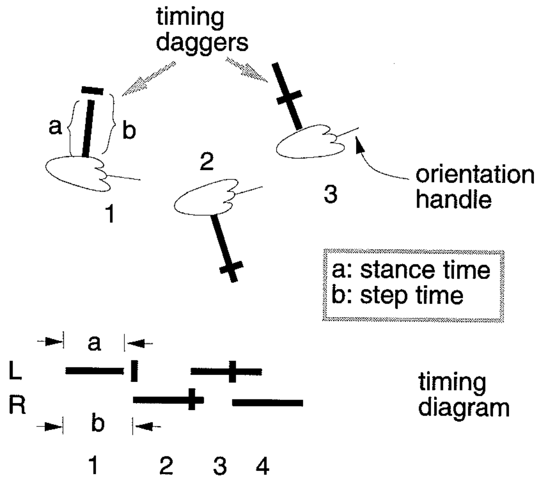

**Figure 3:** *Footprints with timing information and manipulation handles.*

Two factors which often preclude the use of dynamics for animation are the cost of dynamic simulation and, more importantly, the difficulty of generating appropriate control for simulated muscles. The technique we propose employs the features of both kinematics and dynamics. The use of simplified COM dynamics provides for correct timing and balance of all motions. Simultaneously, inverse kinematics is used to ensure natural shapes for the body and legs as they intermediate between the environment and the body. The various elements of the synthesis technique work together as shown in Figure 2.

An explicit feature of our method is that it allows compromises to be struck between physics and visual realism. An animator can ‘stretch’ the underlying physics of a situation in order to yield visually convincing results for otherwise impossible motions. The notion of explicitly enforcing the visual plausibility of a motion is key to our work. We shall use the words *plausibility* and *comfort* interchangeably for the remainder of the paper. The notion of motions being plausible but perhaps non-physical was partially motivated by the rules used to construct exaggerated cartoon motions.

The remainder of the paper is divided into four sections. Section 2 gives the formulation of the problem and describes its solution. Section 3 presents results for several animated characters and their desired footprints. Section 4 describes a high-level planning algorithm which dynamically generates footprints and their timing. It is shown that responsive, interactive locomotion can be produced by using the planning algorithm and the motion-from-footprints algorithm. Section 5 provides conclusions on the use of this technique.

## 2. Problem Formulation and Solution

Creating an animation from footprints requires the specification of footprints and their timing, a definition of the figure to be animated, and a representation for solving the problem. These are now discussed in order.

### 2.1. Footprint and Timing Specification

The basic input provided by the animator is a track of footprints, as shown in Figure 1(a). Also required is timing information, as shown in the timing diagram in Figure 1(b). The use of timing diagrams as shown in this figure is a traditional means for viewing gait data. Unfortunately, it is not always an ideal representation for animation because of the lack of a visible association between the footprints and the timing information. Figure 3 illustrates an alternative representation we choose to use in which timing information can be directly associated with individual footprints.

The *timing daggers* shown in Figure 3 associate two timing parameters with each step. The *stance time* is shown graphically using a bar projecting from the side of the foot, whose length $a$ is proportional to the stance time for the given foot. The *step time* is the time until the next footfall and is indicated by a crossbar at a distance $b$ from the foot. A walk is indicated by the presence of a double stance period, as shown by the transition from footprint 2 to 3. In contrast, footprint 1 specifies a flight phase and is thus a running step. The user interface allows for all five degrees of freedom of a footprint to be interactively manipulated: $x,y$ location, orientation, stance time, and step time.

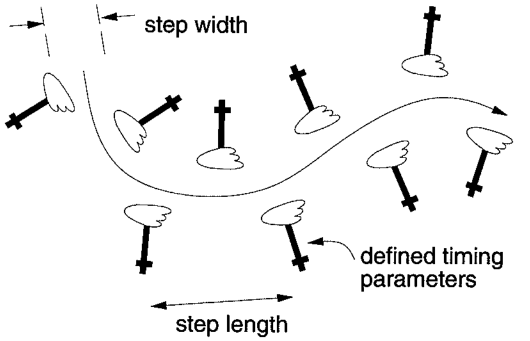

**Figure 4:** *Automatic footprint generation from a path.*

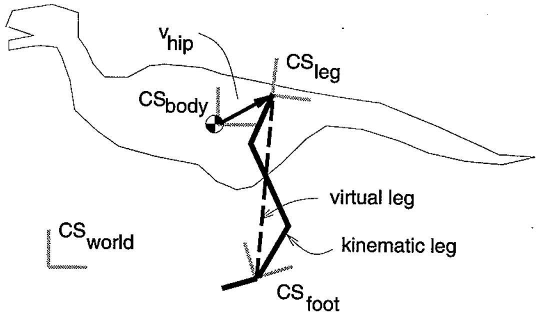

**Figure 5:** *The generic biped model.*

Traditional timing diagrams are likely to be more useful than timing daggers in situations where synchronization is required with other elements of animation. Using an absolute time scale allows an animator to ensure that a character arrives at a given footprint at a given time. It thus remains useful to have the timing dagger representation coexist with a timing diagram representation.

An animator can choose to place individual footprints, or draw a desired path and have the footprints be automatically placed according to parameters controlling the current step length, step width, and timing parameters. Figure 4 illustrates a desired path and the set of automatically generated footprints. The desired path is modeled using a $C^1$–continuous Hermite spline curve. The spline curve of the desired path can be interactively manipulated to give a footprint path the desired shape.

Footprints and footprint paths are placed using a plan view of the terrain. When individual footprints are placed (either manually or automatically), they take on the slope and height of the given point in the terrain. In this manner, specifying a walk on variable terrain is no more difficult than specifying a straight-line walk on flat terrain.

### 2.2. Model Specification

The construction and dimensions of the biped model have a considerable influence on how a stepping motion is carried out. For example, a character with short legs might have to stretch to reach desired footprints. A figure with narrow hips will need to splay its legs in order to straddle widely-spaced footprints.

The generic construction of the biped model is shown in Figure 5. There are a total of five coordinate frames associated with the biped: one for the body, one for each leg, and one for each foot. The body coordinate frame has its origin at the designated location of the center of mass (COM). The coordinate systems for the legs have their origin at the hips. The location of the hips are specified with respect to the COM using vectors defined in the body coordinate frame. The foot coordinate frame has its origin at the ankle and is aligned with the plane of the foot.

The difficulties of dealing with complex leg models for characters are averted by introducing a simpler *virtual* leg onto which the real leg can be mapped[9]. The virtual legs associated with the model are straight, extensible links connecting the hip to the ankle. Inverse kinematics is used to map other kinematic leg models onto the virtual leg. Associated with the virtual leg are a maximal extensible length, $\ell_{max}$ and a nominal length, $\ell_{nom}$, where $0 < \ell_{nom} < \ell_{max}$. The nominal leg length serves to define an extension length that is 'comfortable' for the leg to maintain. This parameter plays an important role in the optimization procedure, as will be seen shortly. Using the general biped model of Figure 5, it is possible to animate a variety of different types of figures, including the bird, dinosaur, and human models to be used in our examples.

The hips and ankles in the model are considered to each have three rotational degrees of freedom. This allows the foot to assume a position flush with the slope of the ground, regardless of the position and orientation of the body.

### 2.3. Centre-of-Mass Splines

The character motion will be represented by a set of parametric curves, $P_i(t)$, representing the trajectory of the center-of-mass in $\Re^3$ as a function of time. One Hermite curve is used per walking step and $C^1$ continuity is ensured between successive curves. A Hermite curve segment begins at the point in time coincident with foot contact and ends upon the next foot contact time. For steps involving a flight phase, a second Hermite curve segment is used to model the trajectory for the duration

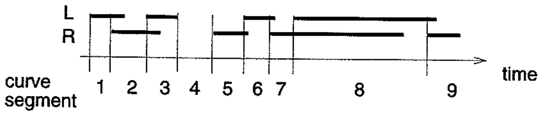

**Figure 6:** *Construction of the center-of-mass spline.*

of flight. Figure 6 shows the Hermite curve segments assigned for a given set of steps.

For a series of $n$ walking steps, there are $n$ Hermite curve segments, each defined by 2 control points and 2 tangent vectors. Parametric continuity conditions constrain the position and tangents of a total of $(n-1)$ of these, thus leaving a total of $2n-(n-1)=n+1$ free control points and tangents to define the trajectory. For Hermite control points and tangent vectors in 3-space, this represents $6n+6$ free variables to specify the COM trajectory over $n$ steps. We shall denote these by the vector $\mathbf{v}$. The values of these free variables will be optimized in order to produce a physically-plausible and visually-plausible motion.

### 2.4. Optimization

The optimization process attempts to minimize the following two-term functional,

$$E=\int_0^T (e_{physics}+e_{comfort})dt$$

where $e_{physics}$ and $e_{comfort}$ are penalty terms for non-physical and unnatural motions, respectively, and $T$ is the duration of the complete motion. The free variables in the optimization are $\mathbf{v}$, namely the parameters which describe the center-of-mass trajectory. The optimization process thus attempts to minimize $E(\mathbf{v})$. We now discuss the constituent terms of the optimization function in more detail.

#### 2.4.1. The Physics Penalty Term

One of the keys to our motion algorithm is a simple and efficient method for computing a physics penalty term, $e_{physics}$, for the motion at any point in time. This physics penalty is based solely on the position of the COM, its acceleration, and the current points of support. The legs are assumed to be massless. Given a COM trajectory and a set of footprints, all of these quantities are known at every point in time. The physics penalty term effectively calculates the imbalance of Newton's second law, $F=ma$, as applied to the center of mass. Specifically, $e_{physics}$ can be thought of as the external force that would need to be applied to achieve the given acceleration.

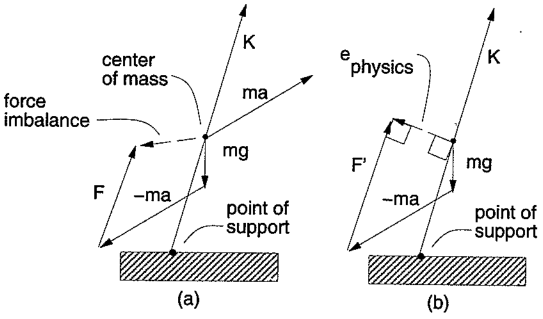

**Figure 7:** *Free-body diagram for single-stance.*

As a simple example, a body having its COM directly above the point of support can only accelerate the COM upwards using leg forces. In such a case, any horizontal component of the acceleration is non-physical, and contributes to the physics penalty term.

We define $e_{physics}$ as follows:

$$e_{physics}=\|F+mg-ma\|$$

where $F$ is the (unknown) net result of the applied muscle forces on the COM, $m$ is the body mass, $a$ is the COM acceleration, and $g$ represents gravitational pull. The acceleration, $a$, is computed as the second time derivative of the COM trajectory. The force $F$ represents the net result of the force of the virtual leg, shown in Figure 5, *as well as* the effect of ankle and hip torques. Assuming no body rotation, the combined effect of these forces and torques must lie on a vector passing from the point of support to the COM.

The net result of the applied muscle forces, $F$, plays an active role in determining the size of the force imbalance. Fortunately, there are several constraints which help determine what value $F$ should take. The simplest situation occurs during a flight phase where there is no support. In this case we evidently have $F=0$.

The situation for single stance, where one leg supports the body, is more interesting. Figure 7 graphically illustrates the previous vector equation. The key constraint placed upon the force $F$ exerted by the stance leg is that its line of action must be parallel to $K$, namely along the direction from the point of support to the center of mass.

It is clear from Figure 7(b) that the force imbalance is minimized when a leg force of $F'$ is applied, where

$$F'=-(mg-ma)\cdot K$$

and $K$ is a unit vector defining the line of action from the point of support through the COM. A negative value of $F'$ implies the leg is pulling the body towards the

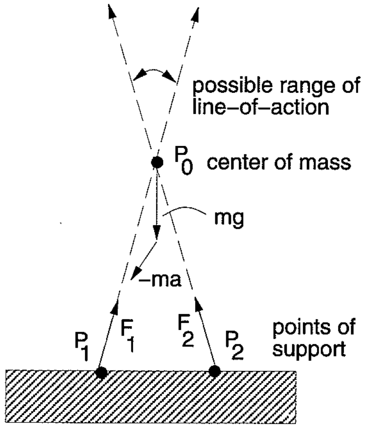

**Figure 8:** *Free-body diagram for double-stance.*

ground and introduces the additional constraint $F' > 0'$. We thus obtain:

$$F = \| \max(0, -(mg - ma) \cdot K \|$$

The actual location of the point of support (POS) is varied within the footprint, depending on the state of the foot. If inverse kinematics determines the leg to be 'on its toes', the POS is taken to be the toe of the footprint. Otherwise, it is taken to be the middle of the footprint. This is an approximation to the more realistic scenario where the POS smoothly moves from the back of the foot towards the front during a stance phase.

For double stance, the situation is slightly more complex, as illustrated in Figure 8. The point of support becomes a line of support, which in effect approximates the convex support polygons which have been commonly employed in static balance algorithms[3]. A plane which embeds the line of action of the legs is defined by $P_0P_1P_2$. From this, one can determine that any component of $mg - ma$ that does not lie in this plane represents an acceleration which requires the use of an external force. For the remaining planar projection of $mg - ma$, any projected component lying within the triangle $\triangle P_0P_1P_2$ can be achieved using an approriate mix of leg forces $F_1$ and $F_2$ and thus the physics can be considered 'real'. For projected components of $mg - ma$ which lie outside this triangle, a similar calculation to the single-stance phase yields the amount of resulting force imbalance.

### 2.4.2. The Comfort Term

Although terms such as control energy have been used in the past for motion optimization, they have typically not been a true reflection of the effort required to perform a given motion. An exception to this is previous work on *strength guided motion*[18, 20]. Our comfort term is intended to reflect a measure of effort or awkwardness. We employ a deceptively simple but effective measure of the comfort at any particular time instant. The measure used is

$$e_{comfort} = k(\ell - \ell_{nom})^2$$

where $k$ is a constant. This penalizes the length of the virtual leg when it is longer or shorter than its desired nominal length $\ell_{nom}$.

The comfort term strongly influences the COM trajectory. It pulls the COM trajectory to lie within a comfortable distance of the appropriate contact footprints and eliminates unrealistic crouching and excessive leg extension. This has a great influence on the realism of a motion and ensures that the inverse kinematics algorithms used to map sophisticated leg models onto the virtual leg always have reasonable configurations to work with. The nominal leg length parameter can also be used by the animator to effect control over crouched gaits, or to alter the leg length as a function of speed if this is desired.

More complex evaluations can also be substituted for the comfort term. In particular, the inverse kinematics solution and the magnitude of the applied leg force could both be used as factors to evaluate the comfort of a given situation. The compromise made in choosing a more sophisticated evaluation of the comfort of a given character position is one of compute time. It is more efficient to postpone the complete motion reconstruction using inverse kinematics until after the COM trajectory has been computed. Performing inverse kinematics during the optimization in order to evaluate the comfort implies an extra constant-but-large computational cost. Our present comfort optimization is calculated directly from the COM trajectory and footprint information.

### 2.4.3. The Optimization Method

In preparation for optimization, the trajectory is initialized as shown in Figure 9 to provide a reasonable point of departure for the optimization process. The Hermite curve control points are initialized to lie directly above their corresponding footprints at a height equal to the nominal leg length. The tangents at the control points are initialized to zero.

While the problem formulation described thus far makes simplifications in a variety of ways, the resulting optimization problem is still of high dimension, making it potentially troublesome to solve. This is in part because the entire motion trajectory is optimized at once. We have applied numerical techniques to the problem because determining analytical derivatives of the terms comprising the energy functional is excessively difficult. Appropriately-tuned numerical techniques can provide solutions at interactive rates.

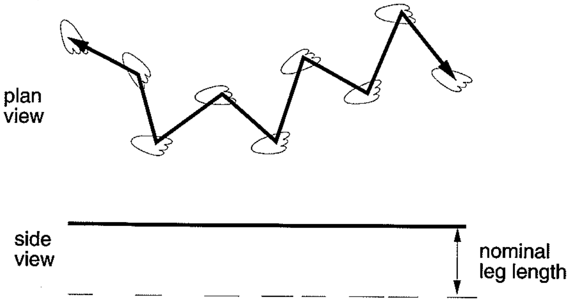

**Figure 9:** *Initial trajectory for optimization.*

```
initialize v
Ebest = E(v)
repeat
  for each element i of v
    Enew = E(v + deltav_i)
    if (Enew<Ebest) then
      v = v + deltav_i
      Ebest = Enew
    else
      Enew = E(v - deltav_i)
      if (Enew<Ebest then
        v = v + deltav_i
        Ebest = Enew
      endif
    endif
  endfor
until converged
```

**Figure 10:** *Pseudocode for optimization algorithm.*

A simple, greedy algorithm is sufficient to provide speedy and effective solutions for minimizing $E(\mathbf{v})$ and hence optimizing the COM trajectory. The optimization algorithm works by iteratively making alterations to each element of $\mathbf{v}$ and evaluating $E(\mathbf{v} \pm \delta v_i)$. If a decrease in cost is observed, the alteration is accepted and otherwise it is rejected. This is repeated until convergence. The pseudocode for the optimization process is shown in Figure 10.

In order to evaluate $E(\mathbf{v})$, the trajectory is discretized into 10 points per Hermite curve segment, yielding the following sum to replace the original integral:

$$E = \sum_i e_{physics}(t_i)\delta t + e_{comfort}(t_i)\delta t$$

Because the curve segments offer local control over the COM trajectory, only the curve segments sharing an altered control point or control tangent need to be re-evaluated when a Hermite control point or tangent (i.e., an element $v_i$) is perturbed.

Several further optimizations are implemented to maximize the speed of convergence. These are described below rather than in the pseudocode for purposes of clarity. A multi-scale approach is used for the alterations, with convergence being achieved at coarse scales before progressively proceeding to finer scales. We use a two-scale search search, corresponding to control-point alterations of *6cm* and *2cm*, respectively. Our results show that these parameter values work well for a wide range of characters and motions.

If an alteration is accepted, further alterations to the same parameter in the same direction are attempted before proceeding on. An optimization takes on the order of 2 seconds for a motion covering 12 steps, as evaluated on a 166 MHz Pentium PC. From our experience to date, the optimization is quite insensitive to the choice of initial condition and thus consistently finds the same solution. There is, however, no guarantee that the solutions found are a global optimum.

### 2.5. Leg Inverse Kinematics

Once the COM trajectory has been determined through the optimization, the animated motion of the legs is created using inverse kinematics. The location of the hips can be computed at all points in time from the COM trajectory and the body orientation. During stance, the desired location of the foot is given by the footprint. During the flight phase, the foot orientation is interpolated between its orientation upon departure from the previous footprint and its orientation upon arrival at the next footstep. The interpolation is carried out by treating the change in orientation as a rotation about a single axis. Legs are retracted to a specified length during the flight phase in a procedural fashion.

Other specific elements of the motion also necessitate a kinematic solution, or can alternatively be modelled dynamically through treatment as secondary motions. For example, the motion of the swing leg for a bipedal character hopping on one leg alone is an example of a case requiring special attention. Other motions, such as swinging arms, are relatively simple to add using either kinematics or as secondary dynamical systems. Many aspects of the motion can be driven as a function of the current phase of a walk cycle. Proposed techniques for the extraction and reuse of medium-to-high frequency motion characteristics obtained from motion capture data[10, 26] should also applicable to further augment the realism of the motions generated using our technique, although we have not yet incorporated this into our implementation.

### 2.6. Body Orientation

While the COM trajectory defines the position of the body throughout a motion, nothing has yet been said about its orientation. The orientation of the body is determined by having it follow the mean orientation of the feet, as determined over a moving window of fixed size. The feet also follow the mean orientation of the footprints. In addition, we have implemented procedural sinusoidal oscillations of the body pitch and roll, which are driven by the current phase of the walking or running motion. The net effect is one of making the body motion more lively in response to the stepping motion.

The orientation can be scripted to operate independently of the orientation of the feet if desired. In such as a case, the COM optimization will take the planned body orientation into account, ensuring that the resulting motion remains both physically-plausible and comfortable. However, it does presume that the orientation of the body is itself not a function of the COM trajectory.

## 3. Results

This section summarizes the type of results that can be expected from the proposed technique using a variety of motions, terrains, and characters. The animator input for the motions discussed in this section consists of the footprints and their timing information. The animations for the following examples can also be found online[27].

All our results make use of a unit mass because the mass $m$ merely acts a constant scaling factor for the $e_{physics}$ term in the optimization functional. This reflects the fact that the optimization attempts to optimize for a *physically-feasible* trajectory, rather than directly being concerned with the magnitude of the supporting forces over time. Note that the optimization functional can be set up to further take the magnitude of the supporting forces into account, if this is desired.

A motion begins with the interactive specification of the footprints, as shown in Figure 15(a). The footprints take on a shape appropriate to the character, in this case the web-shaped foot of a bird. Footprints come with an orientation handle, shown in green, and a timing dagger, shown in red. These handles are directly manipulable. As footprints are added or manipulated, they automatically conform to the underlying terrain.

Figure 15(b) shows the optimized COM trajectory which results from the specification. The phases of the walking motion are indicated by the colour of the trajectory: red represents right stance, magenta represents left stance, and dark blue represents a double stance phase. The green tangent lines on the COM trajectory are the Hermite tangent vectors for the curves. The motion reconstructed from the COM trajectory curve by using inverse kinematics is shown in Figure 15(c). The motion of the bird can easily be influenced through a variety of character parameters, including hip spacing, nominal leg length, the COM position, and the chosen amount of pitch and sway.

The example shown in Figure 16 involves more interesting terrain, namely a set of stairs. As before, the motion specification consists of placing footprints in the plan view. The figures show the motion specification, the optimized trajectory, parts of the animated motion, and a rendered version of the model. Fluid climbing and turning motions are generated as the result of optimizing both the physics and comfort of the movement. The rendered dinosaur model consists of two legs and a rigid body. Animating the shape of the body itself would clearly be an important step in striving for additional realism.

Having a character perform a leap involves placing footsteps an appropriate distance apart and including a flight phase in the timing specification. Figure 17 shows the result of a walking turn, followed by a large leap and a small leap. The parabolic shape of the trajectories during flight are a direct consequence of the physics term in the optimization function. The bends in the trajectory immediately before and after the leaps provide for anticipation and recovery for the motion. These are a direct result of the continuity constraint imposed on the COM trajectory and the comfort term in the optimization.

Running gaits are generated in exactly the same way as walking gaits, but use different footprint distances and timing characteristics. In Figure 18, a human-like figure is made to run across variable terrain. The flight phases in this particular example are of short duration and thus form flat parabolic arcs. The inverse kinematics can be tuned in order to vary the time at which the heel lifts off the ground as the toe performs its work in pushing off.

While motions produced using the algoritm are predictable and typically well-behaved, several shortcomings still remain. The interpolation technique used for the swing leg can potentially cause the legs to pass through each other. The motion is quite sensitive to the footprint timing information. This is especially true for flight phases because the height of a jump varies with the square of the flight time. In most situations, however, the motion is quite forgiving to the input data. In a situation where a walking step is substituted for a hop-step (one with a flight phase) or vice-versa, the motion produced remains reasonably convincing. This depends in part on the intelligent implementation of the inverse-kinematics, which should make a leg reach towards its support footprint, even though the COM trajectory may preclude a leg actually reaching its footprint in extreme situations. Automatically generating the timing is one

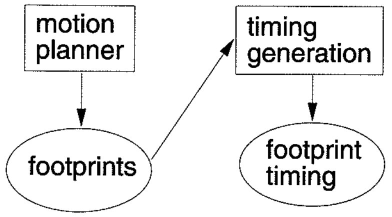

**Figure 11:** *Block diagram for motion planning.*

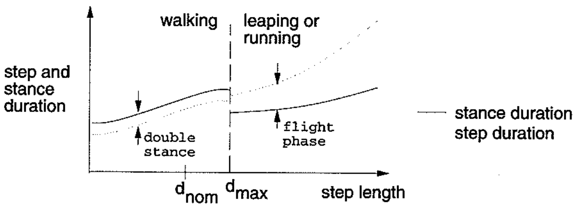

**Figure 12:** *Stance duration and step time as a function of step distance.*

possible method to address these types of problems, as will be discussed shortly.

## 4. Motion Planning

Footprints are also a useful intermediate representation for higher-level motion planning and behaviours. A high-level motion-planning algorithm need only produce footprints as output and can safely delegate the remaining details of generating a motion to the animation algorithm we have described. This process involves solving two problems, as shown in Figure 11. The first is the motion planning process which must choose locations and orientations for footprints based upon the environment and the goals and behaviours of the character. The second step involves synthesizing the timing information from the footprints themselves. We first discuss this latter problem.

### 4.1. Automatic Timing

Given a set of footprints alone, there are many variations of timing which might be used to execute the gait. The method we choose to use to generate the timing information relies on a function which maps the distance between successive footprints directly to stance-time and step-time. An example function is shown in Figure 12. A flight phase (i.e., a leap) is created whenever a step size exceeds a predefined distance. This simplistic criterion is sufficient to induce appropriate leaping and running behaviour from the footprints. An animator is free to make further adjustments to this timing information if the default behaviour is not what is desired in certain instances.

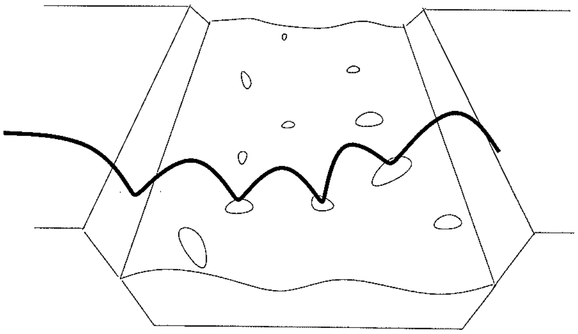

**Figure 13:** *The crossing-the-river problem.*

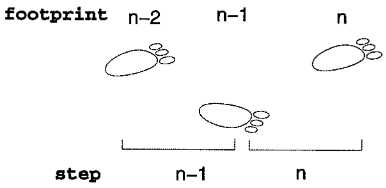

**Figure 14:** *Numbering of steps and footprints.*

Figure 19 illustrates the timing automatically synthesized from a set of bird tracks. In this case, two small leaps are generated, as can be seen from the synthesized timing daggers and the flight phases of the resulting trajectory, as shown in pale blue.

### 4.2. Interactive Planning

A high-level motion planner can make effective use of footprints as a compact and flexible representation for motions. Footprints are a particularly useful represenation for situations in which there are few legitimate footholds, such as crossing a river by stepping from rock to rock, as shown in Figure 13. In these situation, the motion planner need only decide on what would be a reasonable sequence of rocks to step on in order to cross safely. The footprint-to-animation synthesis technique is one of few motion representations which is well suited to generating this type of constrained motion.

It is also possible to make the motion planning process interactive by dynamically generating new footprints as the motion proceeds. Each time a new footprint $n$ is generated, timing information for step $n$ is generated, and the COM trajectory for step $n-1$ is optimized and animated. Figure 14 illustrates the situation. Note that step $n$ is not yet optimized or executed, as this would not allow for proper anticipation of the next

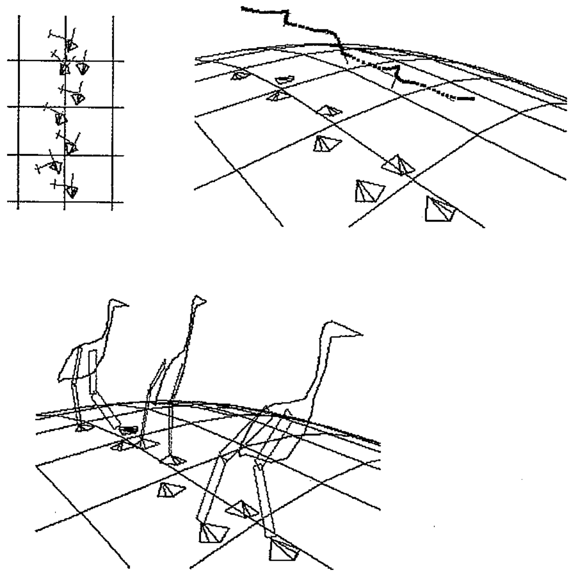

**Figure 15:** *Walking bird. (a) footprints and timing (b) optimized trajectory (c) animation.*

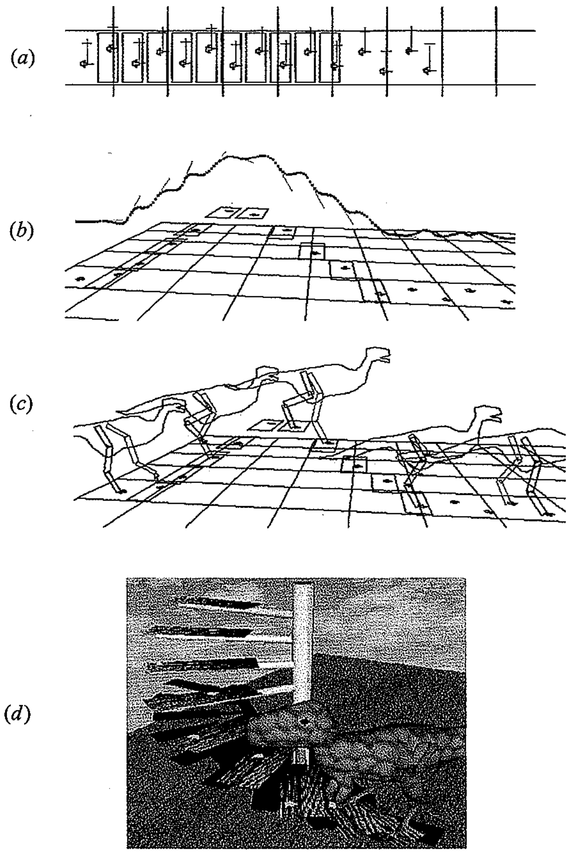

**Figure 16:** *Dinosaur on stairs. (a) footprints and timing (b) optimized trajectory (c) animation (d) rendered dino climbing spiral staircase.*

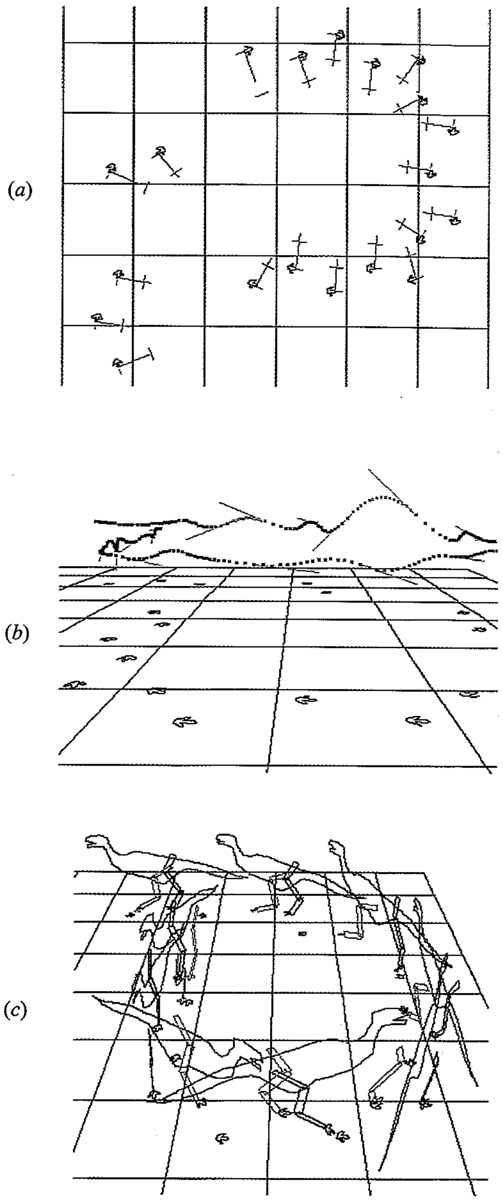

**Figure 17:** *Dinosaur turning and leaping. (a) footprints and timing (b) optimized trajectory (c) animation.*

movement. The optimization which is carried out is local to step $n-1$, although the spline continuity conditions ensure that steps $n-2$ and $n$ also have an appropriate influence on the motion. The limited local optimization requires only a fraction of a second to perform, allowing for a character to interactively plan and execute its locomotion in a dynamic environment.

Our prototype motion planner implements a random wandering behaviour for our bird character, as shown in

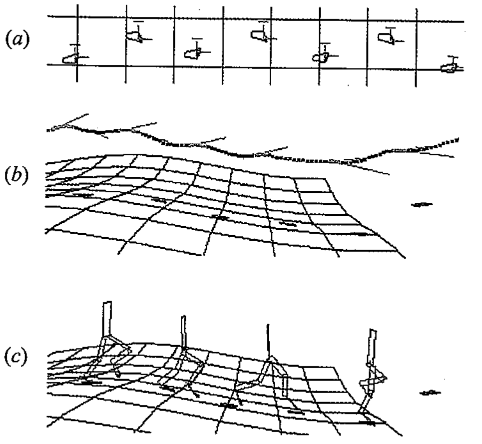

**Figure 18:** *Human running. (a) footprints and timing (b) optimized trajectory (c) animation.*

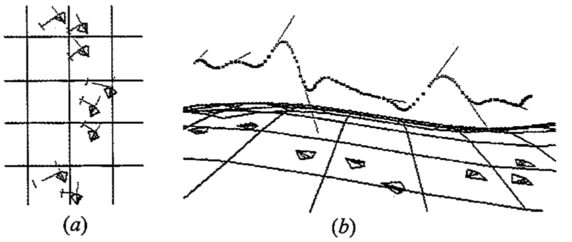

**Figure 19:** *Automatic generation of timing from footprints. (a) footprints and resulting timing (b) optimized trajectory.*

Figure 20. The character's goal in this case is to wander randomly while avoiding trees and staying within the confines of its finite square world. Figure 20 shows results from a session of interactive motion planning and optimization. The motion planner avoids obstacles by making stochastic changes to the walking direction if it detects that the current direction will bring it too close to an obstacle, as defined by a fixed tolerance.

## 5. Conclusions

A technique for synthesizing animated motions from a series of footprints and timing information has been presented. The motion has a physics-based origin and, where necessary, strikes a balance between physical correctness and natural motion. By carrying out a global optimization over a trajectory, motions exhibit anticipation and recovery, which is otherwise difficult to obtain in an automated way.

Footprints are an intermediate-level of motion specification, letting an animator plan the motion on a gross scale, while the low level motion details are synthesized automatically. Consecutive jumps or stepping motions can be smoothly chained together because each instant of the motion takes into account the preceeding and following portions of the motion trajectory, as a result of the global path optimization. The proposed technique does not make exclusive use of kinematics, path planning, or dynamics. Instead, it utilizes the strong features of all three of these approaches to produce a single effective animation technique based on footprint specification.

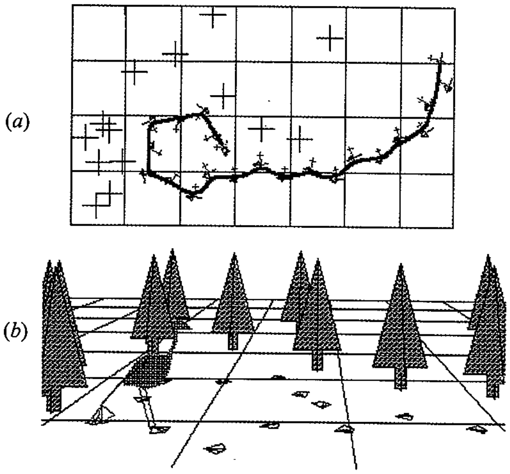

**Figure 20:** *Interactive motion planning. (a) Resulting footprints, timing, and trajectory (b) animation.*

We believe a variety of interesting future work is possible with motion-from-footprint techniques. To properly animate complex human motions, the comfort functions and inverse kinematics require more sophistication than those used in our prototype animation system. The use of complex procedural secondary motions which are tied to the locomotion cycle can potentially lend additional realism to the walking characters. The application of footprint techniques and optimization algorithms to quadrupedal locomotion remains an interesting and challenging direction. The scope of the optimization could also be expanded in other ways. For example, the body orientation could be treated as a further set of trajectories to be optimized. Such a reformulation would have improved body motion at the expense of additional optimization. Further optimized trajectories could also be employed, although the additionaly complexity would likely suffer from diminishing returns.

We have only begun to explore the interesting higher-level behaviours that could exploit the footprint motion specification. Motion planning should include reactions to the movements and behaviours of other characters in the scene. Terrain planning should incorporate more knowledge than the simple avoidance behaviours we have implemented. As with real humans and animals, the terrain planning should be good, but still have sufficient imperfections to be realistic. Motion-from-footprint techniques should provide a solid point of departure for all of these future variations.

## References

1. R. M. Alexander, "How dinosaurs ran", *Scientific American*, pp. 130–136, April 1991.
2. N. I. Badler, B. Barsky, and D. Zeltzer, *Making Them Move.* Morgan Kaufmann Publishers Inc., 1991.
3. N. I. Badler, C. B. Phillips, and B. L. Webber, *Simulating Humans.* Oxford University Press, (1993).
4. N. I. Badler and S. Smoliar, "Digital representations of human movement", *Computing Surveys*, XI/1:19–38, March 1979.
5. R. Boulic, R. Mas and D. Thalmann, "Position control of the center of mass for articulated figures in multiple support", *Computer Animation and Simulation '95*, pp. 130–143 (1995).
6. R. Boulic, H. Noser and D. Thalmann, "Automatic derivation of curved human walking trajectories from synthetic vision", *Computer Animation '94, IEEE Press*, pp. 93–103 (1994).
7. 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).
8. A. Bruderlin and T. Calvert, "Interactive animation of personalized human locomotion", *Proceedings of Graphics Interface*, pp. 17–23 (1993).
9. A. Bruderlin and T. W. Calvert, "Goal-directed animation of human walking", *Proceedings of ACM SIGGRAPH*, **23**(4) pp. 233–242 (1989).
10. A. Bruderlin and L. Williams, "Motion signal processing", *Proceedings of Siggaph '95, ACM Computer Graphics*, pp. 97–104 (1995).
11. M. F. Cohen, "Interactive spacetime control for animation", *Proceedings of SIGGRAPH '92*, pp. 293–302 (1992).
12. T. W. Calvert et al. "Composition of multiple figure sequences for dance and animation", *Proceedings of Computer Graphics International*, pp. 245–255 (1989).
13. M. Girard, "Interactive design of computer-animated legged animal motion", *IEEE Comptuer Graphics and Applications*, **7**(6), pp. 39–51, June 1987.
14. M. Girard and S. Amkraut, Eurythmy, (1989).
15. J. K. Hodgins, "Simulation of human running", *Proceedings, IEEE International Conference on Robotics and Automation*, pp. 1320–1325 (1994).
16. J. K. Hodgins, W. L. Wooten, D. C. Brogan and J. F. O'Brien, "Animating human athletics", *Proceedings of SIGGRAPH '95, ACM Computer Graphics*, pp. 71–78 (1995).
17. H. Ko and N. I. Badler, "Straight line walking animation based on kinematic generalization that preserves the original characteristics", *Proceedings of Graphics Interface '93*, pp. 9–16 (1993).
18. H. Ko and N. I. Badler, "Animating human locomotion with inverse dynamics", *IEEE Computer Graphics and Applications*, pp. 50–59, March 1996.
19. J. F. Laszlo, M. van de Panne and E. Fiume, "Limit cycle control and its application to the animation of balancing and walking", *Proceedings of ACM SIGGRAPH*, pp. 155–162 (1996).
20. P. Lee, S. Wei, J. Zhao and N. I. Badler, "Strength guided motion", *Proceedings of ACM SIGGRAPH*, pp. 253–262 (1990).
21. Z. Liu and M. F. Cohen, "Keyframe motion optimization by relaxing speed and timing", *Computer Animation and Simulation '95, Proceedings of the Eurographics Workshop*, pp. 144–153 (1995).
22. M. H. Raibert, *Legged Robots that Balance.* MIT Press, (1986).
23. M. H. Raibert and J. K. Hodgins, "Animation of dynamic legged locomotion", *Proceedings of SIGGRAPH '91*, pp. 349–358 (1991).
24. C. F. Rose, B. Guenter, B. Bodenheimer and M. F. Cohen, "Efficient generation of motion transitions using spacetime constraints", *Proceedings of SIGGRAPH '96*, pp. 147–154 (1996).
25. Kinetix 3D Studio. "biped," part of character studio. (1995).
26. M. Unuma, K. Anjyo and R. Takeuchi, "Fourier principles for emotion-based human figure animation", *Proceedings of SIGGRAPH '95, ACM Computer Graphics*, pp. 91–96 (1995).
27. M. van de Panne, "From footprints to animation: Online animations", *http:// www.dgp.utoronto.ca /people/van/ani.html*, (1996).
28. M. van de Panne, "Parameterized gait synthesis", *IEEE Computer Graphics and Applications*, pp. 40–49 (1996).

29. C. W. A. M. van Overveld and H. Ko, "Small steps for mankind: Towards a kinematically driven dynamic simulation of curved path walking", *The Journal of Visualization and Computer Animation*, **5** pp. 143–165 (1994).

30. A. Witkin and M. Kass, "Spacetime constraints", *Proceedings of SIGGRAPH '88*, pp. 159–168 (1988).

31. A. Witkin and Z. Popovi'c, "Motion warping", *Proceedings of SIGGRAPH '95*, pp. 105–107 (1995).