

Guided Optimization for Balanced Locomotion
Michiel van de Panne and Alexis Lamouret
Sixth Eurographics Workshop on Animation and Simulation

Published in:

D. Terzopoulos and D. Thalmann (eds.)
Computer Animation and Simulation
Eurographics Series

Copyright Springer-Verlag

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than the publisher must be honoured and acknowledged.

Guided Optimization for Balanced Locomotion

Michiel van de Panne
Department of Computer Science
University of Toronto

Alexis Lamouret
iMAGIS/IMAG-INRIA
Grenoble

Abstract. Teaching simulated creatures how to walk and run is a challenging problem. As with a baby learning to walk, however, the task of synthesizing the necessary muscle control is simplified if an external hand to assist in maintaining balance is provided. A method of using guiding forces to allow progressive learning of control actions for balanced locomotion is presented. The process has three stages. Stage one involves using a “hand of God” to facilitate balance while the basic actions of a desired motion are learned. Stage two reduces the dependence on external guidance, yielding a more balanced motion. Where possible, a third stage removes the external guidance completely to produce a free, balanced motion. The method is applied to obtain walking motions for a simple biped and a bird-like mechanical creature, as well as walking, running, and skipping motions for a human model of realistic proportions.

1 Introduction

The synthesis of realistic walking and running gaits is a difficult problem that has been the focus of a large body of research in computer animation, biomechanics, and robotics. Solutions to the low-level control of walking and running are key elements in the construction of “high-level” locomotion behaviours that deal with direction and speed as opposed to actual joint motion. Unfortunately, existing solutions to the low-level control problem for the physics-based animation of walking and running are few and have many limitations. This impedes the construction of high-level behaviours, which must know the capabilities and limitations of the constituent low-level control primitives. In an attempt to remedy this deficit, we present a new computational method for making synthetic actors of arbitrary design progressively acquire the necessary control for balanced locomotion.

The logical sources of inspiration to draw upon when designing a sensory-motor control mechanism for synthetic actors (both human, animal, and imaginary) are the solutions evolved by nature. Unfortunately, present efforts in this direction often lead to an immense respect for the finely-tuned design of these systems rather than a complete understanding of the complex mechanisms involved. At the same time, it is reasonable

to suspect that biological sensory-motor systems are highly tuned to the bodies they live in, the external environment, and the required motions. As such, they may provide hints towards designing similar control systems for synthetic actors, but they will in no case yield a complete solution[1].

We propose a multi-step process for the automated synthesis of balanced gaits. This process draws upon our everyday experience by making use of *motion guidance* to simplify the learning problem. Consider a parent helping a child learn how to walk or skate by providing a helping hand. Just as this hand serves to stabilize the motion being learned, we shall introduce a similar ‘hand of God’ to the aid of learning synthetic actors. Another good example is that of the training wheels of a bicycle, which offer external support to simplify the learning problem. Rehabilitation techniques for pathological human gaits also employ guiding mechanisms[8].

The guided optimization technique we present uses an external torque to enforce a balanced upright posture during walking or running. We hereafter refer to this guiding external torque as the “hand of God”, or HOG for short. With the HOG in effect, the problem of balance is temporarily (and artificially) resolved, simplifying the process of finding the appropriate control actions to produce a desired motion. Once a basic control strategy for a gait has been synthesized, the external support can subsequently be reduced or eliminated through further optimization. First, the initial control strategy is refined so that the HOG performs less work in helping to maintain balance. This typically produces a motion with a small residual external guiding torque. Second, any remaining HOG work can be removed in sequential pieces.

It is worthwhile elaborating that while the synthesis of motion control for synthetic actors is difficult, it is to some extent a side issue in computer animation, where only the final motion and not its underlying complexity is an issue. It is necessary to hide the underlying complexity in order to produce simple-to-use animation tools. This was a strong consideration for our choice of a synthesis tool whose effects on the final motion are obvious and direct.

The remainder of this paper is divided into four sections. In section 2 we present related work. Section 3 discusses the HOG and its use in the automated optimization process. Section 4 presents results for newly-invented synthetic animals as well as a human model. Lastly, section 5 concludes.

2 Previous Work

The study of walking and running motions has long been a subject of fascination. The easiest way to animate such motions for humans is to capture and store the motion of a real human being. The data can be gathered from film or video (rotoscoping) or by pasting electronically-trackable sensors directly on live actors (motion capture). While no technique can hope to do better than the direct capture of the real phenomena, motion capture has several limitations. Besides requiring special hardware, it is best suited towards animating humans, given the difficulty of finding the extremely rare live Robo-bird and Simpleton creatures discussed later in this paper. Furthermore, once a motion is captured it is difficult to change or reuse in multiple situations. In contrast to

relying on instances of existing acted motion, we shall rely on existing tools for motor learning, namely the principle of guiding a motion.

The specific challenge of creating tools for human animation has been taken on by few. The work of Badler et al.[2], Magnenat-Thalmann and Thalmann[17], and Boulic et al.[4] illustrate the present state of the art in modeling humans using predominantly kinematic methods. The method presented in [3] includes the use of kinematically-guided corrections to motions. A method of generalizing existing rotoscope data while preserving original motion characteristics is presented in [14].

Beyond kinematic methods, there have been several proposals to produce animated walking motions using physics-based models. Girard[9] uses a mix of kinematic and dynamic methods to achieve a variety of biped and quadruped motions. Bruderlin and Calvert[5] use a similar mix of techniques to generate realistic parameterized walking motions for a kinematically complex human model, and later show that parameterized walks can also be achieved using a purely kinematic model[6]. The gaits are constructed using important features extracted from experimental gait data.

The work of Raibert[23] and Raibert and Hodgins[24] demonstrates an elegant and robust control solution for balanced hopping and running creatures having 1, 2, or 4 legs. Hodgins[10] uses a similar solution to control a running motion for a realistic planar human model. As well, Hodgins et al.[11] have developed a variety of control algorithms for other tasks. Stewart and Cremer[26] use changing sets of desired constraints to control the walking motion of a human-like model. McGeer[18] shows that stable passive walks can be achieved down modest inclines. McKenna and Zeltzer[19] show how to synthesize a variety of gaits for a fully-dynamic hexapodal model. There is a significant body of robotics research concerning the control of bipedal walking motions, as well as in biomechanics for simulating human motion. While specific control solutions abound, there has been relatively little work on the learning of gaits in a general setting, i.e., for creatures of arbitrary design.

The challenge of *learning* motor-control functions must thus be addressed if we hope to use physics-based simulations as a tool for creating animations. The space-time constraints technique of Witkin and Kass[29] and the subsequent work of Cohen[7] and Liu et al[16] show that optimization can be used as an effective tool for this purpose. Van de Panne et al.[27] propose a general method for the synthesis of closed-loop control, but this method is restricted to simple systems. The work of Pandy et al.[22] represents a significant advance, showing how to optimize parameterized control histories to obtain desired jumping motions for a human model. The work of van de Panne and Fiume[28], Ngo and Marks[21], and Sims[25] further shows how parameter optimization techniques can be applied with the goal of animation in mind, using a variety of different control representations and optimization techniques. The guided optimization technique presented here proceeds onwards from these optimization techniques by giving additional thought to the optimization process and drawing upon some tricks humans use to help themselves learn. We note that the notion of adding external forces to help achieve a desired motion exists implicitly in the work on space-time constraints, and has also been previously proposed by Lamouret and Gascuel[15].

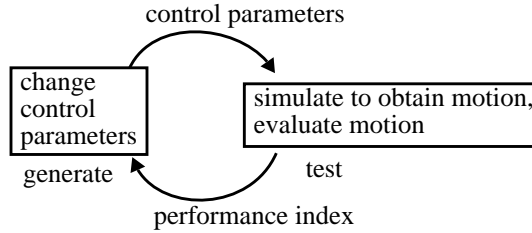


Fig. 1. Nonlinear parameter optimization.

3 Guided Optimization

The HOG exists as a tool to simplify the kind of optimization problems encountered in animation. This need arises from the large search spaces that must be explored by methods based on parameter optimization. While the synthesis procedure we propose relies on parameter optimization, the use of the HOG leads to a simplified optimization problem that can be solved in several stages. Following a brief review of parameter-optimization methods, the implementation of the HOG and its use in learning is described.

3.1 Parameter Optimization for Animation

The general structure of parameter optimization techniques is shown in Figure 1. Typically the control parameters are desired positions for the joints or muscles of the animated figure, which we shall collectively refer to as Ω_d . For articulated figures, these are translated into torques using proportional-derivative servomechanisms defined by $\tau = k_p(\theta_d - \theta) - k_d\dot{\theta}$, where τ is the torque computed for the joint, k_p is a proportional gain constant, θ_d is the desired joint angle, θ is the joint position, k_d is a damping constant, and $\dot{\theta}$ is the joint velocity. These internal torques are subsequently translated into an actual movement by carrying out a mechanical (physical) simulation. The goal is thus to determine how to manipulate the control functions $\Omega_d(t)$ in order to achieve a desired motion.

The first step in solving this problem is to choose a finite-dimensional representation for $\Omega_d(t)$. This can be done using any number of basis functions, ranging from simple discrete samples to more continuous basis functions such as splines, sinusoids, or wavelets. Using a finite set of basis functions to express $\Omega_d(t)$ over a finite time T yields a finite set of numbers which serve to define the applied control. If motions can be evaluated with a scalar performance index that measures their ‘desireability’, then the problem of generating a desired motion is converted into one of parameter optimization, as expressed in Figure 1. An alternative to making Ω_d a function of time is to allow it to be a function of sensory information[21][25][28]. This can also easily be posed as a parameter optimization problem. Regardless of the choice of representation, there are many numerical techniques that can be applied to the resulting parameter-optimization problems. However, they must all take into account that the parameter space is high-dimensional and possibly replete with local minima. In this work, we choose to make Ω_d a piece-wise constant function of time, although the same representation also allows for a piece-wise linear representation, as illustrated later in Figure 5. The controlling torques are calculated at every time step of the simulation.

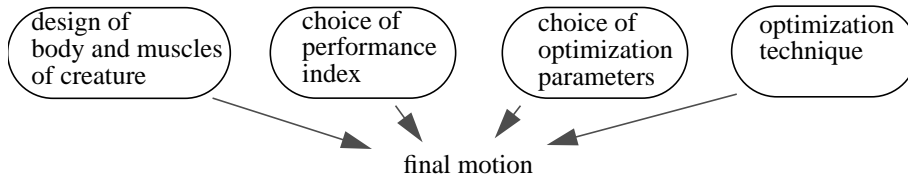


Fig. 2. Using optimization to generate animation.

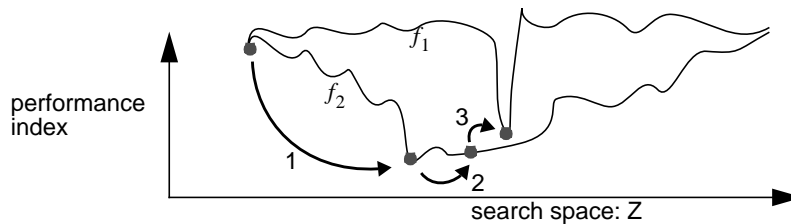


Fig. 3. The effect of shaping an optimization problem. The desired balanced motion is represented by the global minimum of f_1 . Guiding forces change the equations of motion and allow for partial success to be rewarded, thus yielding a simpler optimization problem, as shown by f_2 . The effect of the three stages of the optimization process is shown schematically. Stage 1 finds a good solution for the simpler problem, stage 2 finds a point on f_2 which minimizes the guiding forces, and stage 3 returns to the original equations of motion.

There are ultimately several factors that influence the motion produced using parameter optimization techniques. These are summarized in Figure 2. Two important characteristics can make a parameter optimization problem difficult. The first is dealing with the size or dimensionality of the search space. Methods such as genetic algorithms and simulated annealing are typically used in an attempt to cope with this. A second often-overlooked factor is the fraction of the search space occupied by useful solutions and similarly, the ability of the performance index to guide the parameter search towards this subset of parameter space. It is this factor which makes balanced locomotion a challenge to synthesize. The situation is illustrated in Figure 3.

For a biped, balanced motions exist for a small part of the parameter search space, making it a difficult problem to solve. This difficulty can be overcome in two ways. The first is to change the performance index to reward partial progress. This redesign of the performance index unfortunately has the animator learning how to construct appropriate performance metrics instead of keeping to our original goal of having the virtual actor do the ‘learning’. The second method is to modify the system being controlled in order to work with a simpler projection of the problem, and then revert back to the original projection when a solution to the simpler projection has been obtained. One example of this method might be that of a baby learning to walk with large, stable feet and then gradually having the feet shrink in size. The technique we propose avoids altering the body design, instead using an external HOG to take care of the balance requirements, which can then progressively be reduced or removed.



Fig. 4. The hand of God (HOG). The HOG operates by applying an external torque vector τ to ensure that the up-vector b , fixed to the torso, never deviates significantly from its ideal direction given by u .

3.2 The Hand of God

The hand of God (HOG) is an externally-applied torque vector which serves to keep the body or torso of an animated figure in an upright posture, as illustrated in Figure 4.

It is up to the animator to specify the desired ‘upright’ posture using the vector b , which is defined in the coordinate frame of the torso. This allows walks where the torso is leaning forward or to the side, for example. The HOG ensures that the upright posture is maintained by applying a torque $\tau = k_p (b \times u) - k_d \omega_b$, where k_p and k_d are spring and damper constants, u defines the desired up direction, and the vector ω_b is the angular velocity of the torso. Note that although the HOG causes an upright orientation to be maintained, it is the legs that must at all times bear the weight of the torso.

3.3 Stages of Optimization

The optimization process using the HOG is divided into three stages, as shown in Figure 5. The goal is to first learn the basics of the motion with external assistance, and then to reduce or remove this assistance as learning proceeds. Our particular application of locomotion benefits from being a cyclic motion, so we can use a cyclic representation of the control function $\Omega_d(t)$ for the first two stages of the learning process. Our choice of control representation consists of sampling the cycle period T at n uniform intervals, and constructing an interpolant of order k , $k \leq n$ through the control samples. The resulting parameter set Z has nq elements, where q is the number of actuators to be controlled. As part of the design of the animated figure, the animator is responsible for defining Z_{min} and Z_{max} , which serve to bound the parameter space to be searched. These correspond to joint-limits when animating articulated figures. Our choice of optimization technique is a greedy gradient-descent algorithm, although a variety of other parameter-optimization algorithms can equally well be substituted. In practice, a gradient-descent algorithm seems to work well with the projections of the optimization problem obtained with the HOG.

As a result of the HOG, an automated parameter search to produce a basic stepping motion is a simple task. Performance indices useful for this stage of the optimization are typically related to the distance travelled over several periods of the cyclic motion. In the case of human motion, it can also be convenient to construct a performance index which measures the similarity to existing human gait data. A simple index we propose for this is given by $J = \int \sum w_i (\theta_i - \hat{\theta}_i)^2 dt$, where $\hat{\theta}_i$ is the experimental data

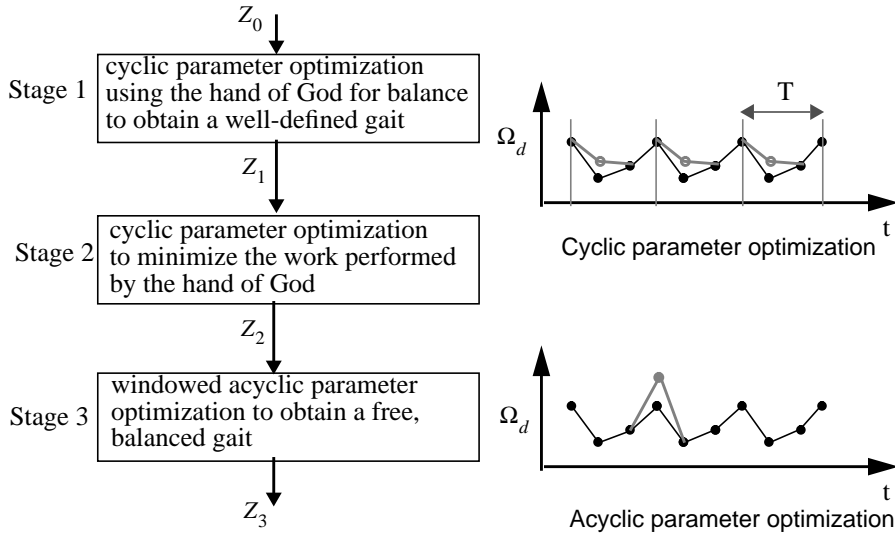


Fig. 5. Stages for a guided optimization. Each stage ensures that the next stage has a suitable starting point for parameter optimization.

for joint i , and w_i defines the relative importance of each joint in the performance index. If a periodic swaying of the torso is desired, the definition of the desired upright vector u can be changed cyclically to accommodate this.

The second stage of the optimization continues to use the HOG, but minimizes the reliance of the motion upon it. Beginning with the parameter set produced by the first phase, Z_1 , the motion is optimized to minimize the work performed by the HOG, defined here as $J = \int (\tau \cdot \tau) dt$, where τ is the moment applied by the HOG as previously defined. The resulting parameter set Z_2 yields a motion with the HOG playing a greatly reduced role, but not completely eliminated. This is typically the result of the synthetic actor being unable to sustain the desired torso position using muscle action alone, which manifests itself as a torso motion which is excessively rigid.

The last stage of optimization is first described in terms of its real-life analogy. Consider a baby taking a first free step into his parents arms. If this step is successful, the parent will have to perform little or no work in helping to restore the child's balance. After it has been determined how to take a successful first step, the parent retreats another step, allowing for two free steps before helping out. The third stage of the optimization algorithm operates according to this description until a true free walk has been generated.

The process is illustrated in Figure 6. A first point to note is that the cyclic constraint on the parameter set is released, thus allowing for small perturbations to the predominantly cyclic control in order to take the corrective actions necessary to maintain balance. Two adjacent windows of fixed size serve to define the duration of the simulation trials: the *optimization window* and the *evaluation window*. The optimization window defines the parameter subset Z_w to undergo optimization. These are the control param-

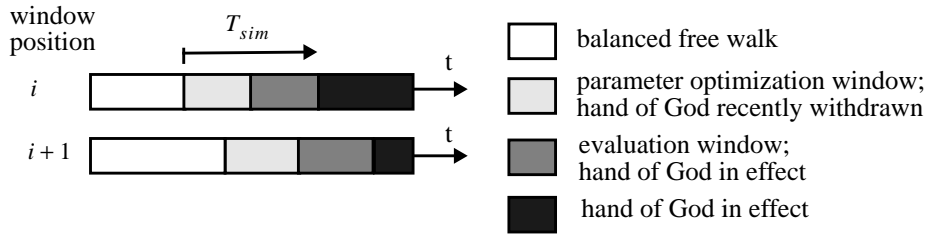


Fig. 6. Stage 3: progressive removal of the hand of God

eters affecting the motion immediately before the restoring action of the HOG is applied. The evaluation window measures the amount of work, J , necessary to restore a balanced position at the end of the free-portion of the motion. The optimization itself consists of minimizing J over the parameter subset Z_w . The windows are moved forward in time when J has been appropriately minimized for the current window position.

4 Results

Our synthetic actors are articulated figures having 5, 12, and 15 links, having 6, 11, and 16 actuators respectively. The equations of motion are obtained using the mechanical simulation package described in [12]. Ground contact is modelled using a stiff spring-damper system which is modified to allow for the simulation of friction and slippage. The simulations require from 3 to 10 seconds of compute time for each second of simulation time. Four second simulation trials were used for most of the motions. Although the optimization process has been defined as having three stages, we have noted that in many cases visually-satisfactory motion is obtained after the application of only stages one and two.

4.1 Simpleton

One of the goals of this work is to show that difficult problems can be tackled effectively using this technique. Figure 7 shows Simpleton, a run-of-the-mill biped. Because the biped has point feet, it is non-trivial to arrive at a balanced walking motion.

For our experiments we used a cycle discretization of $n = 4$ in our control representation. Given six actuated joints, this produces a parameter set of size 24. The initial conditions for the parameter set can be chosen at random if it is desired that Simpleton learn how to walk without the use of a priori information. We often found it useful to specify the initial conditions for the parameter set Z_0 directly because an initial parameter set that already crudely represents the motion is a simply way to save on some otherwise random searching. For the case of a walking gait, this is equivalent to specifying two keyframes: one having the right leg straight and the left leg bent, and the other vice-versa. Distance-travelled over the simulation interval remains the most useful optimization metric we have found. The intermediate results of the optimization

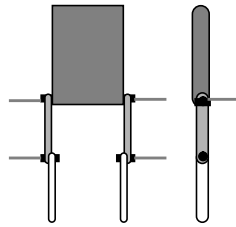


Fig. 7. Simpleton. This three-dimensional figure has pin hinges at the knees and two degree-of-freedom hinges at the hips.

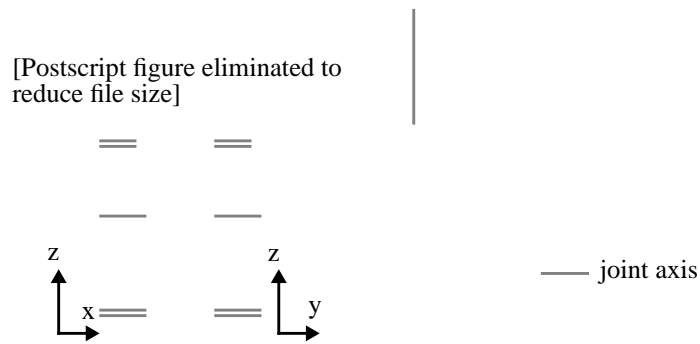


Fig. 8. Robo-bird. Each leg has 5 actuated, aligned pin-hinges. The 'neck' has an additional pin-hinge allowing it to turn from side to side.

can be saved in order to be able to retrieve gaits of any desired speed, ranging from a slow trundle to a loping run.

The effect of the HOG can sometimes be easily recognized after stage 1 due to a clearly off-balance movement that nevertheless remains upright. In an experiment where 'upright' was defined as leaning forward, this had the effect of looking like it was pulling something heavy. The stage 2 optimization removes such artifacts, resulting in changed foot placements to yield an improved balanced motions. Stages 1 and 2 typically require 50 to 200 trials to achieve reasonable results. Stage 3 is more expensive to apply, requiring approximately 500 trials to produce 5 seconds of free, balanced walking. An optimization window size of one-half stride with an evaluation window of one stride works well in practice. Plate A shows the result of a stage-two walking motion over variable terrain.

4.2 Robo-bird

The imaginary robotic bird shown in Figure 8 provides a good example of a more realistic figure one might choose to animate. In order to deal with the large number of degrees of freedom on each leg (five), a set of left-right symmetry constraints were imposed on the parameter set, in addition to the cycle constraint. The image in Plate B shows the creature walking using the control automatically synthesized after stages 1 and 2. A cycle discretization of $n = 4$ was used. Interestingly enough, the animated walk rotates from side to side, even though all the joints in the leg are coaxial. This is a

[Postscript figure eliminated
to reduce file size]

<u>joint</u>	<u>axis of rotation</u>
neck	x
waist	x
LR hip	x
LR hip	y
LR knee	x
LR ankle	x
LR toes	x
LR shoulder	x
LR elbow	x



Fig. 9. A human model with 16 actuated internal degrees of freedom. The model can walk, run, and skip.

good example of the type of secondary motion we can expect to emerge from physics-based animations.

Another point of interest is that stage 3 for Robo-bird cannot find an entirely HOG-free balancing gait. This makes us aware of the common possibility that an animator can easily design a creature which is simply not capable of performing the desired motion, although we cannot guarantee that this was the case here. Using the technique of guided optimization, desired motions can be approximated, even though the actual desired motion might be physically unfeasible.

4.3 Human Model

Last, we present a three-dimensional human model of realistic dimensions and physical parameters[20]. In keeping with our goal of having synthetic actors learn balanced locomotion, we present the human model as a challenging test case. The model we use is shown in Figure 9. While the physical model and the controllers we synthesize have 16 actuated degrees of freedom, 8 of these can be treated in a largely passive fashion, thus leaving 8 significant joints to be controlled. Symmetry conditions are employed along with the cyclic representation of the parameter set is used in order to halve the effective size of the parameter set. A cyclic discretization of $n = 8$ is used.

Using stages 1 and 2, walking, running, and skipping gaits were synthesized for the model. The motions were distinguished by using different initial parameter sets during stage 1, as well as being given different performance indices. The initial parameter set for the walking motion consisted of four keyframes defining a crude alternating stepping motion for the legs. These initial keyframes make the model walk in place when the HOG is first applied. The running motion was initiated with a more energetic set of keyframes, using the ankles to push the body into the air, making for an alternating-leg, in-place hopping motion. The initial parameter set for the skipping motion is chosen to allow for an in-place skipping motion. For all three motions, the motions were then

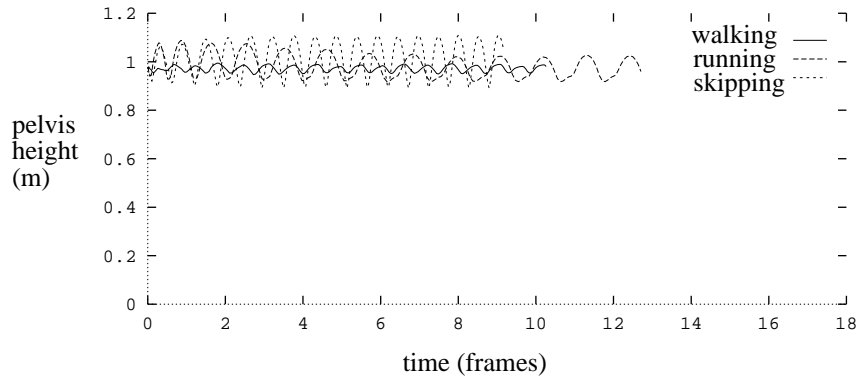


Fig. 10. Average height of pelvis during walking, running, and skipping motions.

optimized for speed (stage 1), similarity to known human motions[13] (stage 1), and for balance (stage 2). The trajectory of the midpoint on the pelvis is shown for the various motions in Figure 10.

The resulting human motions are convincing when a simple line-based visual representation is used, but the motion still retains unconvincing qualities when detailed geometrical models are used to animate the result. The image in Plate C illustrates our 'dressed' geometric model in a running pose. One of the general advantages of parameter-optimization based techniques over manually-designed control systems for the various gaits is that they offer a uniform framework for treating different gaits, such as walking, running, and skipping. As well, the technique proposed here is intended to be general in order to be applicable to the animation of arbitrary types of articulated figures.

5 Conclusions

It is difficult to imagine that we could do better at teaching virtual actors how to walk and run than we would do at teaching real human beings. As such, it is worthwhile looking at the techniques that are employed for teaching motor-control skills to children, patients with pathological walks, or athletes learning a new manoeuvre for the first time. The guidance techniques commonly employed in these situations can be used effectively in the context of physics-based animation as well. The hand-of-God concept is the first step in trying to borrow from human motor-learning techniques and applying them in a virtual setting.

Guided optimization seeks to address the problem of motions which are difficult to synthesize directly using parameter optimization techniques. In these cases, the desired motion typically corresponds to an infinitesimal region of the parameter space to be searched. While it might be possible to construct modified performance indices which reward partial progress, just how to accomplish this is not clear. Using the hand of God has a predictable effect and solves the same problem in a more straightforward fashion. Guided optimization also provides a means of synthesizing an acceptably-close result

to a desired motion even though the desired motion might prove to be impossible, a situation which easily arises when the physical design of a creature is left to a non-expert.

While the use of *sensor-based* motor control has not been directly addressed, we believe that the use of guiding is perhaps even more important for motions to be controlled using sensory information. An initial guided version of the motion serves to determine which sensory inputs are meaningful and which are not. This is likely an important step in the construction of a sensory-motor control system. We feel that the learning of other types of motion can also benefit from guiding techniques. Because the use of harnesses can aid in the learning of many athletic manoeuvres, we foresee virtual guiding harnesses as being a flexible and powerful tool.

References

- [1] N. I. Badler, B. Barsky, and D. Zeltzer. *Making Them Move*. Morgan Kaufmann Publishers Inc., 1991.
- [2] N. I. Badler, C. B. Phillips, and B. L. Webber. *Simulating Humans*. Oxford University Press, 1993.
- [3] R. Boulic, D. Thalmann. Combined Direct and Inverse Kinematic Control for Articulated Figures Motion Editing. *Computer Graphics Forum*, 2 (4), 1992, 189-202.
- [4] R. Boulic, N. M. Thalmann, D. Thalmann. A global human walking model with real-time kinematic personification. *The Visual Computer*, 1990, 6, 344-358.
- [5] A. Bruderlin and T. W. Calvert. Goal-Directed Dynamic Animation of Human Walking. *Proceedings of SIGGRAPH '89*, In *ACM Computer Graphics*, vol. 23, July 1989, 233-242.
- [6] A. Bruderlin and T. W. Calvert. Interactive Animation of Personalized Human Locomotion. In *Proceedings of Graphics Interface '93*, 1993, 17-23.
- [7] M. F. Cohen. Interactive Spacetime Control for Animation. *Proceedings of SIGGRAPH '92*. In *ACM Computer Graphics*, 26, 2 (July 1992), 293-302.
- [8] R. Dickstein, Z. Smolinski, and T. Pillar. Self-propelled weight-relieving walker for gait rehabilitation. *Journal of Biomedical Engineering*, 1992, vol. 14, July, 351-355.
- [9] M. Girard. Interactive design of 3-D computer-animated legged animal motion. *IEEE Computer Graphics and Applications*, June 1987, 39-51.
- [10] J. K. Hodgins. Simulation of Human Running. IEEE Conference on Robotics and Automation, 1994, 1320-1325.
- [11] J. K. Hodgins, P. K. Sweeney, and D. G. Lawrence. Generating Natural-looking Motion for Computer Animation. *Proceedings of Graphics Interface '92*, 265-272, May 1992.
- [12] M. G. Hollars, D. E. Rosenthal, and M. A. Sherman. *SD/FAST User's Manual*. Symbolic Dynamics Inc., 1991.
- [13] V. T. Inman. *Human Walking*. Williams and Wilkins, 1981.
- [14] H. Ko and N. I. Badler. Straight Line Walking Animation Based on Kinematic Generalization that Preserves Original Characteristics. *Proceedings of Graphics Interface '93*, May 1993, 9-16.

- [15] A. Lamouret, M.-P. Gascuel. An approach for guiding colliding physically-based models. *4th Eurographics Workshop on Animation and Simulation*, Barcelona, 1993.
- [16] Z. Liu, S. J. Gortler, M. Cohen. *Hierarchical Spacetime Control*, *Proceedings of SIGGRAPH '94*. In *ACM Computer Graphics Proceedings*, 1994, 35-42.
- [17] N. Magnenat-Thalmann and D. Thalmann. *Computer Animation: Theory and Practice*. Springer-Verlag, New York, 1990.
- [18] T. McGeer. Passive Dynamic Walking. *The International Journal of Robotics Research*, 9, 2, 1990, 62-82.
- [19] M. McKenna and D. Zeltzer. Dynamic Simulation of Autonomous Legged Locomotion. *Proceedings of SIGGRAPH '90*. In *ACM Computer Graphics*, 22, 4 (August 1990), 29-38.
- [20] NASA. *The Anthropometry Source Book*. NASA reference publication 1024, Johnson Space Center, Houston, 1978.
- [21] J. T. Ngo and J. Marks. Spacetime Constraints Revisited. *Proceedings of SIGGRAPH '93*. In *ACM Computer Graphics*, 27 (August 1993).
- [22] M. G. Pandy, F. C. Anderson, and D. G. Hull. A Parameter Optimization Approach for the Optimal Control of Large-Scale Musculoskeletal Systems. *Journal of Biomechanical Engineering*, 114 (November 1992), 450-460.
- [23] M. H. Raibert. *Legged Robots that Balance*. MIT Press, Cambridge, 1985.
- [24] M. H. Raibert and J. K. Hodgins. Animation of dynamic legged locomotion. *Proceedings of SIGGRAPH '91*, In *ACM Computer Graphics*, 25, 4 (July 1991), 349-358.
- [25] K. Sims. Evolving Virtual Creatures. *Proceedings of SIGGRAPH '94*, In *ACM Computer Graphics proceedings*, 1994, 15-22.
- [26] A. J. Stewart and J. F. Cremer. Beyond Keyframing: An Algorithmic Approach to Animation. In *Proceedings of Graphics Interface '92*, 1992, 273-281.
- [27] M. van de Panne, E. Fiume, and Z. Vranesic. Reusable Motion Synthesis Using State-Space Controllers. *Proceedings of SIGGRAPH '90*, In *ACM Computer Graphics*, 1990, 24, 4, 225-234.
- [28] M. van de Panne and E. Fiume. Sensor-Actuator Networks. *Proceedings of SIGGRAPH '93*, In *ACM Computer Graphics*, August 1993, 335-342.
- [29] A. Witkin and M. Kass. Spacetime Constraints. *Proceedings of SIGGRAPH '88*. In *ACM Computer Graphics*, 22, 4 (August 1988), 159-168

Plate A: Simpleton

Plate B: Robo-bird

Plate C: A human model