University of Pennsylvania

ScholarlyCommons

Technical Reports (CIS)

Department of Computer & Information Science

January 1991

Interactive Behaviors for Bipedal Articulated Figures

Cary B. Phillips
University of Pennsylvania

Norman I. Badler
University of Pennsylvania, badler@seas.upenn.edu

Follow this and additional works at: https://repository.upenn.edu/cis_reports

Interactive Behaviors for Bipedal Articulated Figures

Abstract

We describe techniques for interactively controlling bipedal articulated figures through kinematic constraints. These constraints model certain behavioral tendencies which capture some of the characteristics of human-like movement, and give us control over such elements as the figures' balance and stability. They operate in near real-time, so provide behavioral control for interactive manipulation. These constraints form the basis of an interactive motion-generation system that allows the active movement elements to be layered on top of the passive behavioral constraints.
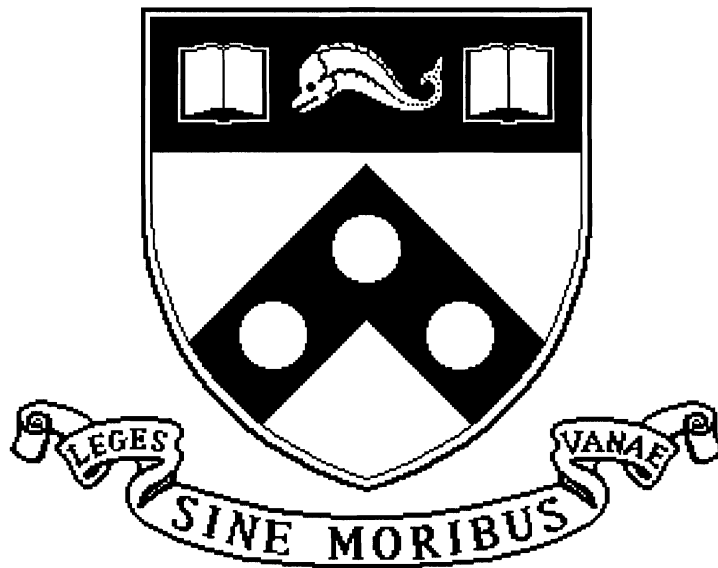
Comments

Interactive Behaviors For Bipedal Articulated Figures

MS-CIS-91-03
GRAPHICS LAB 37

Cary B. Phillips
Norman Badler

1991

Interactive Behaviors For Bipedal
Articulated Figures

MS-CIS-91-03
GRAPHICS LAB 37

Cary B. Phillips
Norman I. Badler

Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA 19104-6389

January 1991

Interactive Behaviors for Bipedal Articulated Figures

Cary B. Phillips
Norman I. Badler

Computer Graphics Research Laboratory
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, Pennsylvania 19104-6389

Abstract

We describe techniques for interactively controlling bipedal articulated figures through kinematic constraints. These constraints model certain behavioral tendencies which capture some of the characteristics of human-like movement, and give us control over such elements as the figures' balance and stability. They operate in near real-time, so provide behavioral control for interactive manipulation. These constraints form the basis of an interactive motion-generation system that allows the active movement elements to be layered on top of the passive behavioral constraints.

1   Introduction

Consider the problem of describing the motion of 3D articulated figures, and imagine how the task would be different if the figures were real human actors. Provided the actors are infinitely agreeable, experienced, and obedient, we could control their movement simply by issuing verbal commands [3].

Such a system should be successful because the motion commands can be conceptual in nature. The figures would still be able to interpret high level instructions and map them to the low level joint movements without any extra work on the user's part. In addition, the figures would automatically behave according to the physical laws of nature, but they would still able to act out the described motion. Such a system would come with the understanding that, like rehearsals in theater or film, the first attempt at a description of the action is likely to fail, either because the description is not adequate or because of misinterpretations on the part of the actors. It is in the refinement of the description that the true control takes place.

In this paper, we describe a system which provides this flavor of control over articulated figures. In terms of Zeltzer's animation taxonomy it most resembles the "guiding and functional unit" system [26]. We describe a geometric model for a human figure and a computational model which incorporates several powerful behaviors which allow us to generate a wide range of complex motions.

In particular, we address the class of motions which are not bounded by dynamics. Such motions are typically executed at slow speed where inertial or frictional effects are minimal, and include standing, shifting the weight from one foot to the other, turning around, and taking small steps to the front, back, or to the side. In short, these motions encompass the types of movement which people act out while standing and moving but not actively locomoting from one place to anther. We explored postural adjustments superficially in the past [1], but lacked the tools to implement and test a system. We believe that this type of motion is of great importance to an animator, and we show how we describe these motions kinematically. We believe this approach provides superior control to dynamics techniques, particularly since these motions do not need the full complexity of a dynamic simulation.

1.1   Overview

We show how to construct useful motion and "behavioral" primitives out of kinematic constraints. The behavioral primitives maintain and control balance and posture, while the action primitives initiate elements of

0

motion that allow complex movements to be composed out of small pieces and layers.

These primitives serve as the foundation for several higher level motion description mechanisms. Since they operate in near real time, they provide behavioral control for interactive manipulation. This allows the user to push, pull, and twist the figure interactively using our 3D direct manipulation interface, all while the figure maintains its balance. These primitives form the basis of an interactive animation system that allows the active movement elements to be layered on top of the passive behavioral constraints. Finally, these primitives provide the necessary interface to task level animation programs.

Section 2 gives background information on the animation of articulated figures, principally through goal directed techniques, to place our contribution in the proper context. Section 3 describes our geometric model for articulated figures and how to perform inverse kinematics on them. Section 4 describes our technique for using the constraints as primitive control components, and describes how we kinematically control the center of mass. It also describes how we control the torso and pelvis. Section 5 describes the real-time interaction mechanism, and Section 6 describes the motion composition techniques, providing some examples of the types of commands the system uses.

2   Background

Articulated figures are traditionally modeled with a hierarchy. The body segments form the nodes and the joints of the figure connect the levels in the hierarchy. When a joint "moves" all segments below it in the hierarchy move.

The keyframing approach to animation describes motion by interpolating between postures at specific times. This provides precise control, but requires skilled animators to provide the postures, timing, and intuition for the character's "personality."

Dynamic simulations (e.g. [11, 14, 23, 13]) are good for generating motion which consists of passive bodies reacting to external forces, such as gravity, but it is difficult to specify forces which achieve a specific motion. Adding kinematic constraints helps control, but for slow motions it is questionable whether the dynamics approach is needed at all.

Some have used the notion of constraints – desired geometric relationships – to determine motion. The term constraint has been used to mean both a goal [2, 5, 25] and a boundary condition [24, 18].

Task level animation and simulation systems have not yet progressed to the point of being suitable for general purpose animation, particularly if the animator wants very fine control over the subtle aspects of a movement, but preliminary efforts are encouraging [9, 3, 22]. The principal advantage of such systems is the reduction of effort in describing the exact timing of human actions and the symbolic references allowed to the geometry database.

Techniques for locomotion have been generally concerned with the cyclic nature of human or animal walking [8, 10, 7, 17], but considerable human movement involves more aperiodic activity such as subtle shifting of the feet to maintain balance or small steps back and forth. The "goals" in most locomotion algorithms are too specific to be applied to anything else.

Systems which provide goal directed motion have been generally used only on rather simple objects such as chains or mechanisms, and the available goals have been rather simple as well, such as point-to-nail or point-to-point constraints. Although such systems are very powerful for generating certain types of motion, they have not adequately addressed the problem of how an animator is to assemble a collection of goals which will accurately describe the intended motion [4]. Stating that such constraints will vary over time does not solve the fundamental problems of determining useful sets of constraints for human motion, negotiating their (overlapping) interactions, or organizing their timing for motion realism. In addition, they have not been successfully applied to highly articulated figures with expected behaviors. Flocking behavior-constraining functions have demonstrated particle motion within a global framework [21], but do not apply to articulated figures. Though Witkin and Kass were able to elicit a range of interesting and "human-like" behaviors from a lamp model, they used a global optimization approach that does not lend itself to interactive manipulation or large numbers of constraints. Lee et al [15] were able to use local strategies to determine motions based on a strength model of a human figure. The resulting motions could be affected interactively and appeared to correlate with expected human behaviors.

3   Articulated Figures and Inverse Kinematics

In this section, we describe the geometric model that we use for articulated figures, and we show how we perform inverse kinematics on the figures. We then describe how to use kinematic constraints as general purpose "handles" for controlling the figures.

3.1 The Geometric Model of a Human Figure

The techniques described in this paper are implemented as an extension of *Jack*™, a multifaceted system for interactively modeling, manipulating, and animating articulated (principally human) figures. *Jack* represents figures as collections of rigid segments connected by joints that may have arbitrary rotational or translational degrees of freedom. Each degree of freedom has upper and lower limits that are enforced during manipulation. The figures are described in text files through a language that allows modeling figures of arbitrary geometry and topology, not just human figures.

Although the figures modeled in *Jack* are hierarchical, they are defined independently of how they are rooted, and they may be rooted through any attachment point on the body. *Jack* maintains the hierarchy in two forms: the external form conforms to the way the user has defined the figure, in terms of the directionality of its joints; the internal form accounts for how the figure is currently rooted and enables the global placement of each segment to be determined in terms of the segment dimensions and joint displacements. This means that the user's conception of the transformation across a joint is independent of how the figure is rooted in space.

The model of the human figure that we use for the examples in this paper has 36 joints with a total of 88 degrees of freedom, excluding the hands and fingers. It has a torso consisting of 17 segments and 18 vertebral joints [16]. Each vertebral joint has three degrees of freedom, each of which has a very small range of motion. Each joint limit is based on biomedical literature. The spine has a total of 54 degrees of freedom, although for realistic-looking human motion, there is considerable coupling between the joints.

Monheit [16] has developed a computational model for describing movements of the spine in terms of total bending angles in the forward, lateral, and axial directions. The technique uses weighting factors that distribute the total bending angle to the individual vertebrae in such as way that respects the proper coupling between the joints. Different weight distributions generate bends of different flavors, such as neck curls or motions confined to the lower back.

3.2 Inverse Kinematics

Jack uses an inverse kinematics algorithm that is based on a variable-metric optimization procedure, described in detail in [27]. This method uses the gradient descent approach to minimize the potential energy described by

---

† *Jack* is a trademark of the University of Pennsylvania. "Jack" is a nonsense name, not an acronym.

a set of constraints. The constraints describe a desired geometric relationship between an *end effector*, or a reference point, on the figure and a desired *goal* position or orientation in space. The distance between the end effector and the goal is the constraint's potential energy. The constraints may have positional and/or orientational components. The positional component may be such types as point-to-point, point-to-line, point-to-plane, etc. The orientational components may describe one, two, or three degree of freedom orientational restrictions. The algorithm handles arbitrary numbers of constraints and arbitrary numbers of degrees of freedom. The constraints may overlap in the sense that a single joint may affect several constraints. The system may be over-constrained, in which case each constraint may not be able to be minimized individually. It may also be under-constrained, in which case the set of joint angles minimizing the constraint energy is not unique. In the later case, the posture generated is the one that occurred first in the algorithm's descent along the gradient of the energy function.

The algorithm is an iterative numerical procedure. At each iteration it computes the Jacobian of the input joint set, which relates the change in each joint angle to the change in total potential energy. This total potential is a weighted sum of the energy from each constraint. This determines a joint-space trajectory to follow which minimizes the energy and "solves" the constraints. The algorithm is monotonically convergent, which means that from one iteration to the next, the total energy does not increase. The algorithm terminates when the total energy is below a certain threshold, or when successive iterations fail to decrease the energy any further, in which case the algorithm is in a local minimum.

3.3 Generating Motion with Inverse Kinematics

Generating motion with inverse kinematics is somewhat different from constraint based systems that are based on dynamics. In particular, the only useful product of the inverse kinematics algorithm is the final position with the constraint energy minimized. The intermediate steps during the solution process should not be considered as "motion". To describe motion with inverse kinematics, we must select an appropriate set of *end effectors*, or reference points on the figure, and then describe the desired positions and/or orientations of these end effectors at each time increment. We then invoke the inverse kinematics algorithm at each time step to determine the set of joint angles that satisfies the desired relationships. (This is the movement generation mechanism used in [15]). Our system advances from

3

one time step to the next by examining the current location and orientation of its end effectors and deciding what direction to move them, and how fast, to get to the next time step.

For the purposes of this paper, we consider the inverse kinematics algorithm as a black box that takes as input a set of constraints and a set of joints and returns with a set of joint angles that minimize the energy described by the constraints.

The key to successfully describing motion through inverse kinematics is to choose properly the end effectors and then design sets of constraints that cause the figure to move in predictable patterns.

3.4 Making Effective Use of Inverse Kinematics

There are several pitfalls in describing postures and movements through inverse kinematics. The reason for using inverse kinematics is to be able to achieve postures for a figure through descriptions that are not very precise, without having to specify individual joint angles. But when the figures and the desired postures are complex, a seemingly simple and complete description of a posture may not be adequate. Redundancies and local minima may cause the posture to be awkward, unacceptable, or simply unattainable. This is not a deficiency in the algorithm, but a deficiency in the input.

There is a tendency to believe that animation via goals is a simple matter. And it seems at first passing that a few simple constraints should be enough to describe a fairly complex motion. However, most types of human movement involve subtle changes in many different parts of the body. Simulating this motion requires coordinating the effect of many constraints simultaneously. When there are many constraints, the complexity of controlling them all approaches that of describing motion at the joint level!

Clearly, the ability to solve constraints numerically is only part of the puzzle. The glue that holds the puzzle together must be a structure through which the constraints can be controlled.

3.5 Constraints as Handles

We use the constraints as handles by which to control parts of the figure. We can make a loose analogy between this and a marionette puppet controlled by strings, except that our strings need not hang vertically, and they can twist and push as well as pull. How do we pull on the strings to get the figure to move as we want? How many strings do we need? Where should we attach them? We choose not to shape the goal-control mechanism into highly specific motion control elements

to perform tasks like walking or running, but to design general purpose motion building blocks that stand by themselves as useful mechanisms of control.

One of the reasons that dynamic simulation has been used successfully to generate computer animations is that it makes objects obey the physical laws of nature, so certain elements of the motion come for free. There is a behavioral richness to the objects that is lacking in kinematic systems, where the animator must control every element explicitly. One key to giving kinematically based systems such richness is to have some elements of the control take care of themselves.

We are not overly concerned here with the physical laws of nature but in capturing some of the global characteristics of human-like movement. We are willing to sacrifice some degree of Newtonian realism in order to achieve greater interactive control. We believe that a large portion of these characteristics can be captured through some simple behavioral tendencies, the most important of which are balance and stability. By phrasing these tendencies as figure *behaviors*, we can view the effect of the constraints in a more intuitive light.

4 Behaviors for Articulated Figures

In this section, we describe the set of constraints which we use to control the body, both in its natural state and as we apply motion primitives to it. The basic architecture of our system lets us treat time in one of two ways. First of all, we can "freeze time" and make postural adjustments to the figure through the real-time interaction mechanism described in Section 5. In this case, we can think of each iteration of the interaction as a time step. Alternatively, we can set up a series of primitive *actions* as described in Section 6 and then start the system time running from a certain point. The primitive actions cause the motion to take place.

There are seven basic constraints through which we control the body in its natural state. These provide the handles on the figure's center of mass, pelvis, torso, and left and right heels and toes.

4.1 The Supporting Elements

We begin by recognizing the importance of the support structure of the human body, i.e. its feet and legs and how they support the body's weight. As bipedal creatures, human beings have a built-in closed loop between the feet and legs that they are very good at manipulating. Unfortunately, because we model articulated figures as a hierarchy, we must take special care in modeling the connection between the feet and the ground.

We do this by designating one foot or the other as dominant, and we the root the figure heirarchy through that foot. We hold the other foot in place by a constraint located at the ball of the foot. The foot has a toe joint which can rotate to allow the heel to come off the floor while the toes remain flat. The orientation component of the foot constraint keeps the foot flat on the floor while allowing it to twist.

Since the posture of every part of the figure is described through constraints, it theoretically doesn't matter which foot is dominant, although in practice it does give the interaction a different feel. In a sense, the dominant foot has a constraint of infinite weight, since there is no possibililty that its relationship will not be satisfied.

4.2 The Center of Mass and Balance

The center of mass of an object is one of its most important landmarks because it defines the focal point for forces and torques acting on it. The center of mass of an articulated figure is particularly significant because its location relative to the feet defines the state of balance. The *support polygon* of a figure is the convex hull of the regions of the figure in contact with the ground. As long as the center of mass of the figure is vertically above this polygon, the figure is balanced.

This is of critical importance for human figures, because so many aspects of the movement through space of a human figure are dictated by the need to maintain balance. In addition, many types of movement, such as stepping and walking, involve intentional shifts in the center of mass away from the support polygon, followed by actions of the feet and legs to restore the balance. We consider balance as one of the most significant behaviors to model in a human figure, both the ability to maintain it and the ability to deviate from it.

The center of mass of an articulated figure is a weighted sum of the centers of mass of each body segment, weighted according to the segment's fraction of the body's total mass. We can either compute the centers of mass of the body segments algorithmically, or in the case of human body figures take them from biomechanics literature [12].

4.3 Inverse Kinematics with the Center of Mass

We model balance in the figure as a constraint on the center of mass to remain vertically above a point in the support polygon. We designate a single point as the balance point rather than using the entire support polygon because we want to have control over the placement of the point within the polygon. This allows us to shift a figure's weight forward or side to side without moving its feet.

We associate the center of mass logically with the lower torso region of the figure, and we use this as the end effector of the constraint, with the ankle, knee, and hip joints of the dominant leg as the constraint variables. During the constraint satisfaction process at each time step, the center of mass is not recomputed. Since the center of mass belongs logically to the lower torso, its position relative to the torso remains fixed as the inverse kinematics algorithm positions the ankle, knee, and hip so that the previously computed center of mass point lies above the balance point. There are generally other constraints active at the same time, along with other postural adjustments, so that several parts of the figure assume different postures during the process.

After we solve the constraints, we recompute the center of mass. It will generally lie in a different location because of the postural adjustments, indicating that the figure is not balanced as it should be. Therefore, we must solve the constraints again, and repeat the process until the balance condition is satisfied. In this case the structure of the human figure helps. Most of the postural adjustments take place on the first iteration, so on subsequent iterations the changes in the center of mass relative to the rest of the body are quite minor. We measure the distance that the center of mass changes from one iteration to the next, and we accept the posture when the change is below a certain threshold. Although it is difficult to guarantee the convergence theoretically, in practice it seldom takes more than two iterations to achieve balance.

4.4 The Spine and Torso

Our inverse kinematics algorithm cannot effectively position the spine, not only because of the computational complexity of its 54 degrees of freedom, but mostly because the algorithm is not capable of respecting the proper coupling between the vertebral joints. The computational model developed by Monheit [16] is very powerful because it allows the motion to be described through the total bending angle along with a weight distribution. This function is not easy to differentiate so performing inverse kinematics is difficult.

This problem is not so severe, though, considering that the position and orientation of the neck relative to the waist are somewhat related. Although it is computationally difficult to position the neck precisely, it is considerably easier to describe its orientation and we can use changes in orientation to control the position. Seldom do humans need to position their neck at precise points in space. More common is the task of bending forward or bending to the side. This achieves a posi-

tional relationship through a change in orientation.

Biomechanics research has demonstrated that one of the most constant elements in simple human locator tasks is the global orientation of the head [6]. One theory explaining this suggests that the head is the principle sensor of stability. Therefore, we design an optional behavior that holds constant the global orientation of the upper part of the spine.

To model this type of behavior, we monitor the global orientation of the neck as the body posture changes at each time step. We measure the difference in euler angles between the current and desired neck orientation, and then apply these rotations to the spine.

4.5 The Pelvis

The pelvis connects the lower part of the spine to the upper legs. This is the general area of the center mass, so its position is governed primarily by the center of mass constraint. Therefore, the constraints on the pelvis involve only its orientation. The passive behavior of the pelvis involves holding its current orientation. Because of its central location, manipulations of the pelvis provide a powerful control over the general posture of a figure, especially when combined with the balance and torso constraints.

5 Real-time Interaction

The real-time interaction mechanism is described in [19] and [20]. Using this facility, we can interactively move and rotate the goals of constraints around in space through a 3D direct manipulation technique, which gets its input from a three button mouse. This mechanism provides a nice form of postural control, although it is not so good at choreographing complex *motions* interactively.

We allow the following types of interaction. Each of these corresponds to a *Jack* system command which allows the appropriate property to be manipulated interactively.

bend spine This follows the technique described in [16]. The center of mass constraint causes automatic postural adjustments in the legs. For example, if we bend the torso forward, the hips automatically shift backwards so that the center of mass remains over the same point. Figure 1 shows how the pelvis automatically adjusts to maintain balance.

rotate pelvis This interactively changes the orientation of the constraint on the pelvis. We can rotate the pelvis forward and backward, side to side, or

we can twist it vertically. The constraints on the feet keep them planted on the ground. For example, if we set up a constraint on the torso, and then rotate the pelvis forwards, the figure will automatically squat but keep its head up. Figure 2 shows how the torso automatically adjusts itself to remain vertical, while the hips shift backwads to maintain balance.

move center of mass To do this, we move the goal point for the center of mass constraint, which allows us to disturb the figure's balance. We can shift the center of forwards or backwards, or side to side to concentrate the weight on one foot or the other. Figure 4 shows the center of mass being moved to the side.

We can also move the center of mass by a point-to-point constraint instead of a point-to-line constraint. This allows us to move the center of mass up or down, causing the figure to squat or stand on its tip-toes.

move foot One foot is always the dominant one, and it serves as the root of the figure heirarchy. The other foot is held in place by a constraint. We can interactively move either foot. The behavior of the dominant and non-dominant feet are subtly different. Moving a foot interactively is not quite like stepping, because the center of mass does not move. However, postural adjustments in the center of mass still take place.

6 The Composition of Actions

The notion of *action* in our system is a scripted change to a constraint controlling the body. An action has three distinct parts: its beginning, its application, and its termination. Each action has a distinct starting and ending time. Each action has its own set of constraints controlling part of the body. Its parameters control the velocity of the contraint's goal and the constraint's weight as a function of time. Through a windowed interface, we can create, modify, and delete actions and get a global picture of a movement sequence. Figure 5 shows an example screen with the animation window on the left and the graphics window on the right.

We generate motion sequences by simulating the progression of time. When an action's starting time occurs, its *preaction* is performed, which usually involves activating its constraints and establishing their starting goal values. The application of the action involves changing the goal position and orientation for the constraints' end effectors. This occurs at each time step during the con-
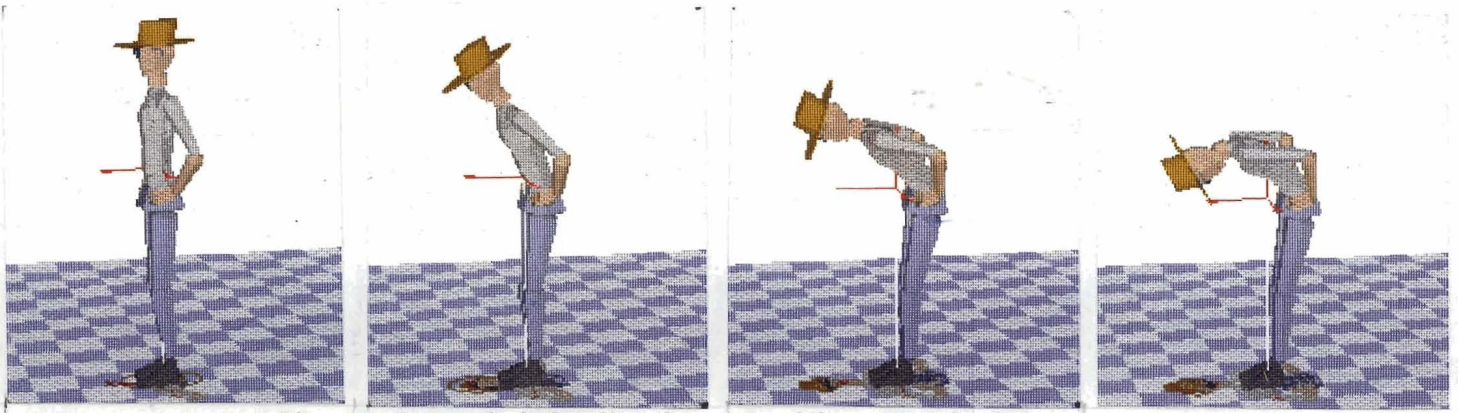
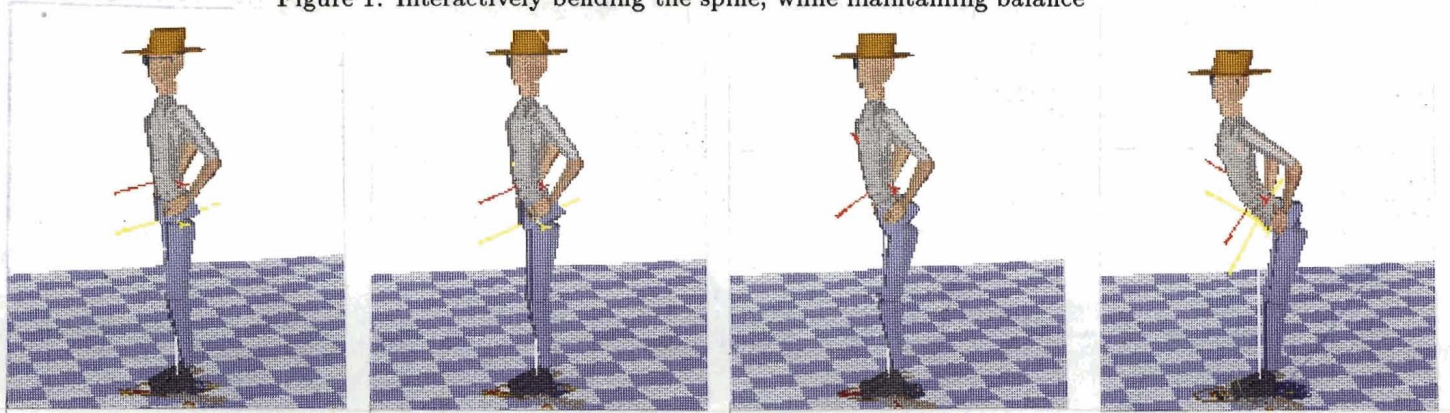Figure 1: Interactively bending the spine, while maintaining balance

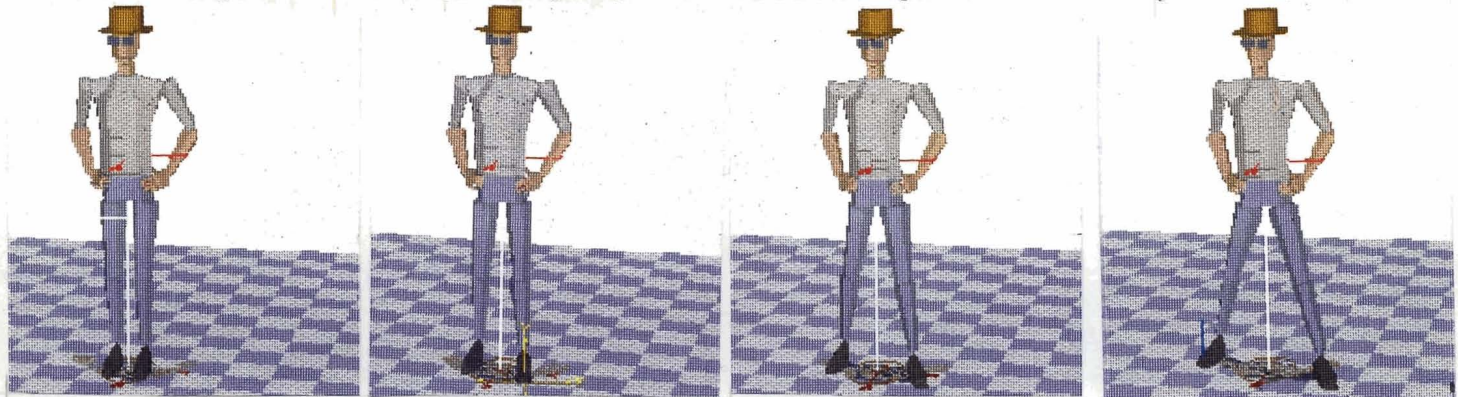Figure 2: Interactively rotating the pelvis, while maintaining balance with head up

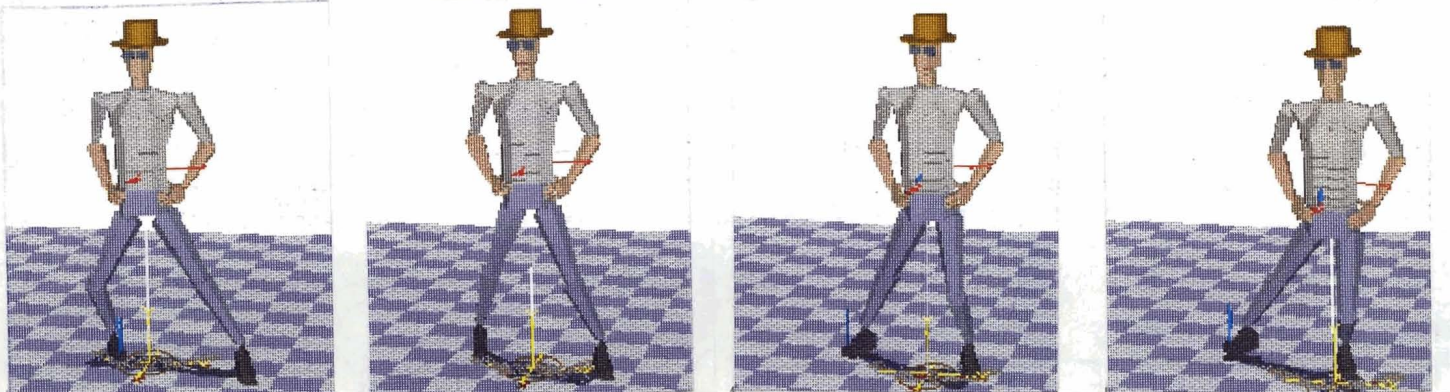Figure 3: Interactively moving a foot while maintaining balance

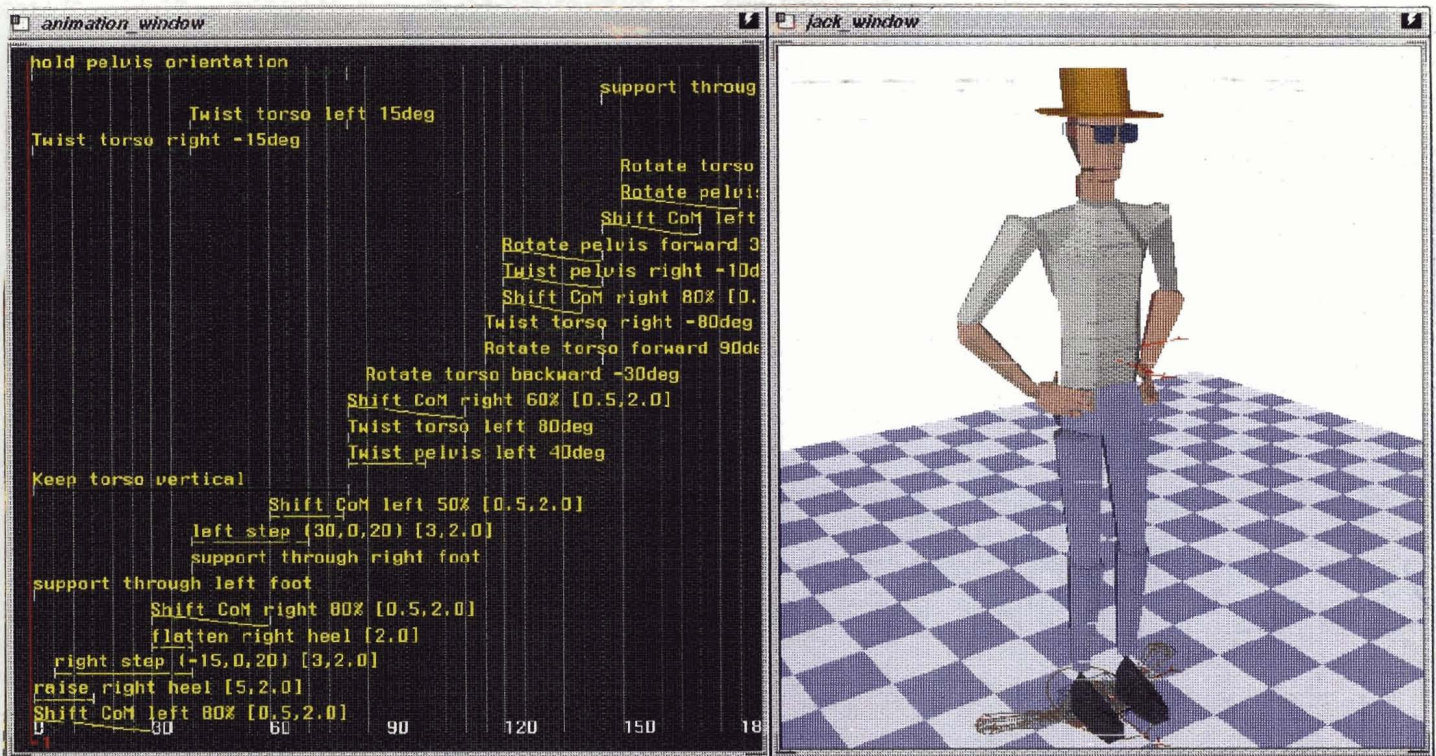Figure 4: Interactively moving the center of mass

Figure 5: Animation Script for a Movement Sequence

straint's lifetime. When an action's ending time occurs, its constraints are deactivated.

The actions we provide include:

shift center of mass The center of mass can be constrained to lie between the feet, or it can be directed to move in a specific direction. When it is constrained between the feet, it will automatically move if the feet move, say due to a stepping motion of the feet. The parameter in this case is an interpolation factor between the feet. This allows the weight to be shifted a certain percentage to one foot or the other. When the center of mass is shifted explicitly, the parameters entered describe the (x, y, z) direction in which to move.

control pelvis The pelvis can be controlled passively or actively like the torso. The passive control instructs it to keep its current orientation. The active control involves rotating forward or backward, laterally, or twisting along its vertical axis. In this case, the parameter is the twist angle. Several pelvis control actions may overlap to cause simultaneous bending in different directions. This constraint affects only the orientation of the pelvis. The center of mass constraint largely determines the position of the pelvis.

control torso The torso can be controlled passively or actively. The passive control instructs it to keep its current orientation. The active control involves bending forward, backward, side to side, or twisting along its axis. In this case, the parameter is the twist angle. Several torso control actions may overlap to cause simultaneous bending in different directions.

raise/lower heel of foot The parameters are the height to raise or lower the heel. This is useful for beginning and ending a stepping action. This involves a point-to-plane constraint on the heel. We use a point-to-plane constraint because it allows this motion to be used in conjunction with stepping motions of the foot that tend to move the heel laterally.

move foot The parameters are the (x, y, z) displacements to move the foot, and the angle to twist it. The displacements may be either global coordinates or local to the foot.

The position and orientation of the goal of an action's constraint are interpolated between a starting and ending value. The starting value is the current position or orientation of the end effector when the constraint is activated, i.e. during is preaction. We may specify the

8

ending value either in global coordinates or relative to its starting value. This is done when the action is created, but we may change the value later interactively. In either case, once the constraint is activated, its ending goal position is determined and does not change. We control the rate of movement through a velocity function. The velocity may increase, decrease, ease in and then ease out, or remain constant.

Actions may overlap in time, even ones which control the same part of the body. Since each action has its own constraints, this simply means that during the period of overlap, there will be multiple constraints on that part of the body. This is handled automatically by the inverse kinematics algorithm.

We must take special care to control the effect of constraints when this overlap occurs. If constraints die out abruptly, then their termination may cause discontinuities in the motion of the figure. This may happen if a constraint is pulling part of a figure in a certain direction opposed to another constraint. In this case, the constraint should be phased out gradually rather than terminated instantaneously.

We allow the weight factor of each constraint to be a function of time. The weighting function associated with a constraint may increase, decrease, ease in and then ease out, or remain constant over the lifetime of the action. In practice, constant weights suffices when there are not many active actions. However, actions controlling the same part of the body that overlap in time should generally have the weight of the first action decay towards the end of its lifetime instead of remaining constant. It the current implementation of our system, it is up to the user to recognize this situation and set the weight functions accordingly.

7 Examples

The videotape accompanying this paper demonstrates the real-time interaction, and it contains two example movement sequences. It shows an interactive session preforming the manipulations given in Figure 1 through Figure 4. The animation script for the movement sequence in which the figure turns to look over his shoulder appears in Figure 5.

These movement sequences illustrate the type of movements for which this system is particularly well suited. They involve slow movements of the feet, legs, pelvis, and torso. Of critical importance during these motions is the movement of the center of mass with respect to the feet. These movements are also very non-periodic.

8 Conclusions

The motion primitives in our system provide an effective means of control over articulated figures. The movement sequences which we have generated with this system would be difficult to do with either a keyframe system or a dynamics system, or with locomotion algorithms because the movement is not periodic.

We do not expect these elements alone to automatically generate realistic-looking human movement. Our purpose here is more fundamental. Our approach has been to develop a general purpose set of movement elements which have specific effects. Some effects are local, such as moving a foot or raising a heel, while others are global, like maintaining balance or keeping the torso vertical. Taken together, these elements allow us to compose movement sequences of a quite general nature.

In the future, we will consider how to automatically generate sequences of these actions to provide more "macro-like" control. Encorporation of strength and rate control models lie ahead, as well as validation experiments.

9 Acknowledgments

References

[1] Norman I. Badler, Joseph O'Rourke, and Bruce Kaufman, "Special Problems in Human Movement Mimulation," *Computer Graphics* 14(3):189–197, July 1980.

[2] Norman I. Badler, Kamran Manoochehri, and Graham Walters, "Articulated Figure Positioning by Multiple Constraints," *IEEE Computer Graphics and Applications*, 7(6):28–38, June 1987.

[3] Norman I. Badler, Bonnie L. Webber, Jugal Kalita, and Jeffrey Esakov. Animation from instructions. In *Making Them Move: Mechanics, Control, and Animation of Articulated Figures*. N. Badler, B. Barsky, and D. Zeltzer (eds.):51–93, Morgan-Kaufmann, 1990.

[4] Alan H. Barr, "Teleological Modeling," in *Making Them Move: Mechanics, Control, and Animation of Articulated Figures*. N. Badler, B. Barsky, and D. Zeltzer (eds.):315–321, Morgan-Kaufmann, 1990.

[5] Ronan Barzel and Alan Barr, "A Modeling System Based on Dynamic Constraints," *Computer Graphics* 22(4), 1988.

[6] Alain Berthoz and Thierry Pozzo, "Intermittent Head Stabilization During Postural and Locomotory Tasks in Humans," in *Posture and Gait: Development, Adaptation, and Modulation*, B. Amblard, A. Berthoz, F. Clarac (eds,):189-198, Excerpta Medica, 1988.

[7] Armin Bruderlin, and T. W. Calvert, "Goal-directed, Dynamic Animation of Human Walking," *Computer Graphics* 23(3):233–242, 1989.

[8] T. Calvert, J. Chapman, and A. Patla, "Aspects of the Kinematic Simulation of Human Movement," *IEEE Computer Graphics and Applications*, 2(9):41–50, Nov. 1982.

[9] Jeffrey Esakov, Norman I. Badler, and Moon Jung, "An Investigation of Language Input and Performance Timing for Task Animation," *Proc. Graphics Interface '89*: 86–93, Waterloo, Canada, 1989.

[10] Michael Girard, "Interactive Design of 3-D Computer Animated Legged Animal Motion," *IEEE Computer Graphics and Applications* 7(6):39–51, June 1987.

[11] Mark Green, "Using Dynamics in Computer Animation: Control and Solution Issues," in *Making Them Move: Mechanics, Control, and Animation of Articulated Figures*. N. Badler, B. Barsky, and D. Zeltzer (eds.):281–314, Morgan-Kaufmann, 1990.

[12] Marc R. Grosso, Richard D. Quach, Ernest Otani, Jianmin Zhao, Susanna Wei, Pei-Hwa Ho, Jiahe Lu and Norman I. Badler, "Anthropometry for Computer Graphics Human Figures," Technical Report MS-CIS-89-71, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, 1989.

[13] James K. Hahn, "Realistic Animation of Rigid Bodies," *Computer Graphics*, 22(4):299–308, 1988.

[14] Paul M. Isaacs and Michael F. Cohen, "Controlling Dynamic Simulation with Kinematic Constraints," *Computer Graphics*, 21(4):215–224, 1987.

[15] Philip Lee, Susanna Wei, Jianmin Zhao and Norman I. Badler, "Strength Guided Motion," *Computer Graphics*, 24(4):253–262, 1990.

[16] Gary Monheit and Norman I. Badler, "A Kinematic Model of the Human Spine and Torso," Technical Report MS-CIS-90-77, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, 1990.

[17] Michael McKenna and David Zeltzer, "Dynamic Simulation of Autonomous Legged Locomotion," *Computer Graphics* 24(4):29–38, August 1990.

[18] Greg Nelson. "Juno, A Constraint-based Graphics System," *Computer Graphics* 19(3):235–243, 1985.

[19] Cary Phillips and Norman I. Badler, "Jack: A Toolkit for Manipulating Articulated Figures," *ACM/SIGGRAPH Symposium on User Interface Software*: 221–229, Banff, Canada, 1988.

[20] Cary Phillips, Jianmin Zhao and Norman I. Badler, "Interactive Real-time Articulated Figure Manipulation Using Multiple Kinematic Constraints," *Computer Graphics* 24(2):245–250, 1990.

[21] Craig W. Reynolds, "Flocks, Herds, and Schools: A Distributed Behavioral Model," *Computer Graphics*, 21(4):25–34, 1987.

[22] Daniel Thalmann, "Motion Control: From Keyframe to Task Level Animation," in *State of the Art in Computer Animation* N. Magnenat-Thalmann and D. Thalmann (eds.):3–17 Springer-Verlag(New York), 1989.

[23] Jane Wilhelms. "Using Dynamic Analysis for Realistic Animation of Articulated Bodies," *IEEE Computer Graphics and Applications*, 7(6):12–27, June 1987.

[24] Andrew Witkin and Michael Kass, "Spacetime Constraints," *Computer Graphics*, 22(4):159–168, 1988.

[25] Andrew Witkin and William Welch, "Fast Animation and Control on Nonrigid Structures," *Computer Graphics* 24(4):243–252, August 1990.

[26] David Zeltzer, "Task-level Graphical Simulation: Abstraction, Representation, and Control," in *Making Them Move: Mechanics, Control, and Animation of Articulated Figures*. N. Badler, B. Barsky, and D. Zeltzer (eds.):3–33, Morgan-Kaufmann, 1990.

[27] Jianmin Zhao and Norman I. Badler, "Real Time Inverse Kinematics with Joint Limits and Apatial Constraints," Technical Report MS-CIS-89-09, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, 1989.