Interactive Animation of Personalized Human Locomotion

Armin Bruderlin, Tom Calvert

School of Computing Science
Simon Fraser University
Burnaby, B.C. Canada V5A 1S6
(armin@cs.sfu.ca)

ABSTRACT

This paper describes recent advances in the procedural animation of human locomotion. Our original KLAW (Keyframe-Less Animation of Walking) system [3] has been substantially modified; locomotion parameters such as "velocity" or "step length" as well as locomotion attributes like "bounciness" or "pelvic list" can now be changed on the fly, and since they become immediately active, their effect can be observed in the motion of a human figure on the screen. By providing interactive and real-time control, the system has been shown to be quite useful in the rapid prototyping of personalized human locomotion. Sequences like a the walk of an old man or a marching soldier can be quite readily obtained by changing the values for parameters and attributes via sliders. Whereas the original system relied on a simple dynamic model to produce realistic locomotion cycles, our new version is completely kinematic. Cubic spline and linear interpolation between step constraints replaces the dynamic calculations with little trade-off in realism, but a great improvement in performance.

KEYWORDS: Computer animation, human figure animation, motion control.

1. INTRODUCTION

In recent years, computer animation has played an increasing role in such areas as advertising, entertainment, education, scientific visualization and simulation. However, while animating simple, rigid objects like "flying logos" has become common practice, expressing human movement with a computer is still in its infancy. One of the problems is that the human body possesses over 200 degrees of freedom and is capable of very complex movements. Another challenge in animating human movement is the fact that humans are very sensitive observers of each others motion, in the sense that we can easily detect erroneous movement (it simply doesn't look right), although we often find it much more difficult to isolate the factor which causes the movement to look incorrect.

Typically, a body is represented by a hierarchical structure of rotational joints where each joint has up to three degrees of freedom. Even for very simplified models of a human figure with as few as 22 body segments [6], on the order of 70 parameters (joint angles and a reference point for the body) have to be specified in each frame of an animation; for a 1 minute animation at 30 frames/sec, this means that 126,000 numbers are involved to determine the motion of the model. The animation of a realistic human model with "flesh" requires many more parameters; problems such as facial expressions, clothing and the adjustment of tissue around the joints have to be resolved [6].

Because of this, much of the research in motion control for articulated bodies has been devoted to ways of reducing the amount of specification necessary to achieve a desired movement, that is to develop higher level controls which relieve the animator from having to specify tedious detail explicitly [2,5,7,12,13,14,18]. As shown in Figure 1, human figure animation can be looked at as a hierarchical process.

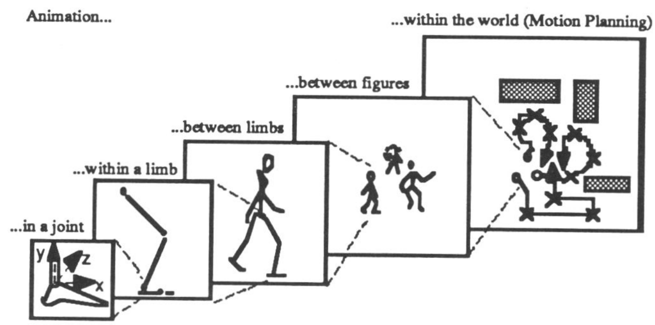

Figure 1: Levels of motion control.

The traditional keyframing technique [16] provides motion control at the lowest level where joint angles are specified over time. As we move higher up in the hierarchy, the system relies increasingly on internal knowledge about particular movements in order to automate the movement generation. Approaches that incorporate information about movements are denoted as *procedural* systems; an algorithm generates a specific movement based on some "high-level" specification of the animator (e.g. "walk at speed x"). At the top level, motion is specified in terms of a script like "Frank walks to the door while Sally is watching him..." from which the system derives all the motion. The reality today is that we are quite far from such a general, high-level system, and most commercially available animation systems still rely on low-level keyframing, which provides the most detailed control, but can be tedious to use.

In order to produce realistic looking motion of articulated figures, dynamic analysis has been applied as a control technique [1,8,10,15,17]. Although motions generated by these approaches look convincingly real, the animator usually has to experiment to get the desired result. Generally, dynamics provides less flexible control and involves numerical integration of equations of motion which makes real-time interaction difficult.

In this paper, we describe a procedural technique to animate human locomotion. We think that procedural approaches are well suited for cyclic or structured movements such as walking, running, or grasping. These movements are well studied, making it easier to define algorithms that produce realistic animations. In a procedural system, a desired motion is conveniently specified by a set of parameters (rather than joint angles in keyframing, or forces and torques in dynamics). These parameters have to capture the essence of a movement and allow for different instances of a movement. Therefore, the naturalness and usability of such a system depends much on the choice of these parameters. In our system, up to three *locomotion parameters* — step length, step frequency and velocity — can be specified to define the basic locomotion stride. Furthermore, any of 15 *locomotion attributes* can be set to individualize the locomotion; for example, the amount of pelvic rotation and list, the bounciness or the stride width can be altered. With control of the attributes the animator can reflect the personal characteristics of the figure being animated such as the walk of a young girl or an old man, of a happy or a sad person.

Unlike the original KLAW (Keyframe-Less Animation of Walking) system [3] which is a hybrid procedural system based on dynamics and task-level animation, the new system is completely kinematic. Cubic spline and linear interpolation between step constraints replace the dynamic calculations with little trade-off in realism. The gains from this are quite significant: the algorithm is now fully interactive and runs in real-time, and thus has become a much more useful tool for animators.

In the next section, the basic control mechanisms of the procedural method are outlined. Section 3 explains the locomotion parameters and attributes in the context of the real-time interface. A discussion on the usage of the system and a comparison with the dynamic system is given in section 4.

2. PROCEDURAL CONTROL OF LOCOMOTION

2.1 LOCOMOTION CYCLE

Human locomotion describes an intricate activity where body translation results from rotational movements in the lower limbs. However, locomotion is a cyclic activity of recurring patterns with a basic unit of one locomotion stride or cycle consisting of two symmetric steps (left and right) as shown in Figure 2, so we can limit our discussion to one locomotion step.

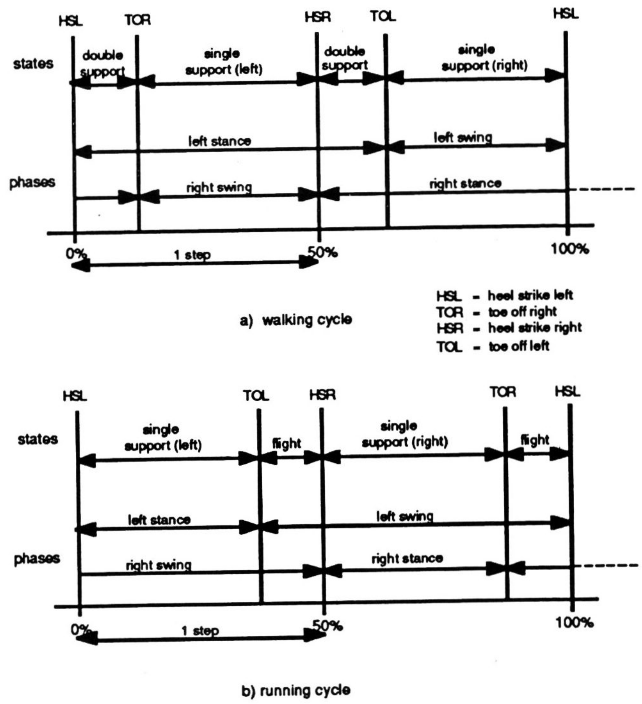

Figure 2: Locomotion cycles for walking and running.

The underlying idea of our walking algorithm is twofold: it is hierarchical and step-oriented. "Hierarchical" implies that each locomotion step is determined by two independent parameters (expressed in terms of the step unit): step length and step frequency. Together with their product, which is the speed of the locomotion, they form the three *locomotion parameters* that specify a desired gait pattern as a high level task.

A step consists of a double support state, where both feet are on the ground, and a single support state, where one foot is off the ground; a running step consists of a single support state plus a flight state where both feet are off the ground. In terms of the individual legs, each state is made up from a stance and a swing phase, shifted in time. This holds for walking as well as for running. In fact, it is just the amount of "overlap" of these phases that determines whether a walking or running gait is present. In walking, the stance phases of the two legs overlap; as the step frequency increases, the duration of this overlap becomes smaller. When the duration of the double support vanishes completely, then a running gait results, in which the swing phases start to overlap.

The algorithm is also "step-oriented" which means that the appearance of a walk – determined by the locomotion parameters and attributes (see section 3) – can be changed with the granularity of one locomotion step. For example, the step length, pelvic rotation or amount of knee bend can be changed from one step to the next.

2.2 ALGORITHM

For each locomotion step, we need to calculate three *step constraints* from the high-level specification of parameters and attributes: the duration for the leg phases, the leg angles at the end of each step, and the control points which determine the movement of the hip of the stance leg during a step. Then the intermediate values from the previous step to the current step can be interpolated. The duration for the leg phases as well as the leg angles at the end of each step are calculated based on the current step frequency and step length, respectively (as explained in more detail in [3]).

The computation of the control points for the hip of the stance leg is done in two parts, a vertical (y) and a horizontal (x) component. In each case, four control points per step are determined, as indicated by t1,...,t4 in Figure 3; the first and last point in x and y are easily derived given the current step length and the leg angles at the beginning/end of each step [3] (at times t1 and t4 in Figure 3). The second and third control points are computed as follows: based on research on human locomotion [9] it is known that the vertical displacement is lowest around the middle of double support (t2) and highest around the middle of the swing phase (t3), whereas the horizontal displacement reaches a maximum (ahead of average position) around the middle of double support (t2) and a minimum (behind average position) around the middle of the swing phase (t3). These observations are in direct correlation with energy expenditure during a locomotion stride as shown in Figure 4. The potential energy curve of the motion of the upper body is essentially identical with the vertical displacement, whereas kinetic energy changes with horizontal velocity. Knowing the durations of the current step (t_{step}), the double support (t_{ds}) and the swing phase (t_{swing}) [3], as well as the current step length (s_l), the second and third control points are

$$
\begin{aligned}
x_{mid\text{-}ds} &= 0.5 * t_{ds} * s_l * x\_factor \,/\, t_{step};\\
x_{mid\text{-}swing} &= (\, t_{ds} + 0.5 * t_{swing}\,) * s_l \,/\, (\, t_{step} * x\_factor\,);\\
y_{mid\text{-}swing} &= f(knee\_bend);\\
y_{mid\text{-}ds} &= y_{impact} - bounce\text{-}factor * \\
&\qquad\qquad (\, y_{mid\text{-}swing} - y_{impact}\,) \,/\, 5;
\end{aligned}
$$

where x_faxtor is the change in horizontal velocity (a value of 1.1 has given good results); f() is a trigonometric function of the locomotion attribute for maximum knee extension during stance (with a default value for knee_bend of 8 degrees), and bounce_factor is a locomotion attribute for the degree of bounciness of the locomotion (with a default value of 1); y_{impact} is calculated based on the leg angles at the end of the step.

Interpolating splines [11] are now fitted through the two sets of four control points to generate the position of the hip for the stance leg during a step. Given the position of the toe which is stationary during stance, the *virtual leg* principle is applied to calculate the angles for the stance leg during the current step as explained elsewhere [4].

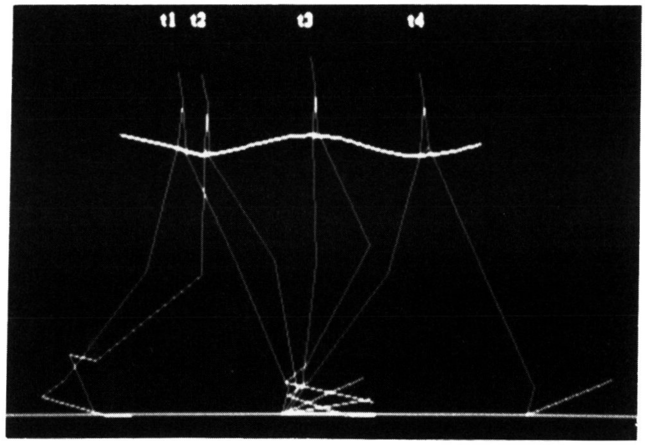

Figure 3: Trajectory of the stance leg hip during a step.

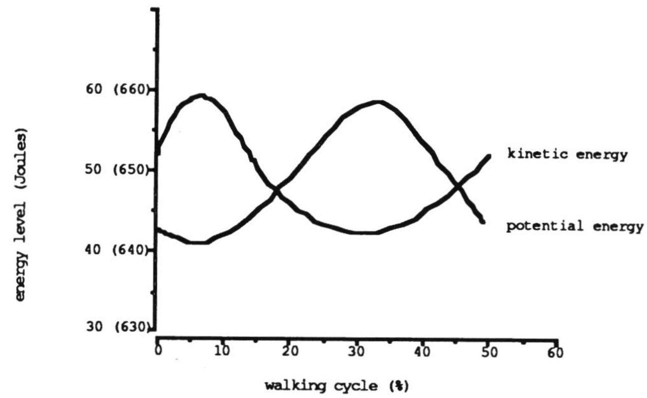

Figure 4: Approximations of kinetic and potential energies of the upper body for an average walking step (adapted from [9]). The ordinate values in parentheses are for potential energy.

Once the motion of the stance leg is determined, a pelvis is induced to produce three determinants of gait: pelvic rotation, pelvic list and lateral displacement of the body. As shown in Figure 5, pelvic rotation is a maximum at heel-strike (t1 and t3) and a minimum at mid-stance (t2; both hips are aligned horizontally). Figure 6 illustrates pelvic list, which is a minimum at heel-strike (t1 and t3; hips are aligned vertically) and a maximum at the end of double support (t2; the hip of the swing leg drops below the stance leg hip). Lateral displacement of the body is caused by the fact that the body always shifts slightly over the weight-bearing leg (see arrows in Figure 5). The displacement is a minimum at heel-strike (t1 and t3) and a maximum halfway through the stance (t2). From the extreme values of these determinants, the position of the hip for the swing leg is determined by linear interpolation. Both the default values for rotation and list of the pelvis can be changed interactively as locomotion attributes. Lateral displacement is a function of stride width and velocity (greater stride width means more and faster locomotion means less displacement).

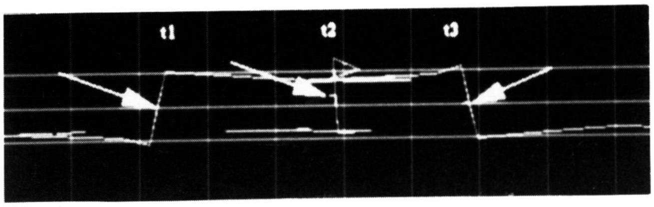

Figure 5: Top view: pelvic rotation in transverse plane and lateral displacement of body (white arrow).

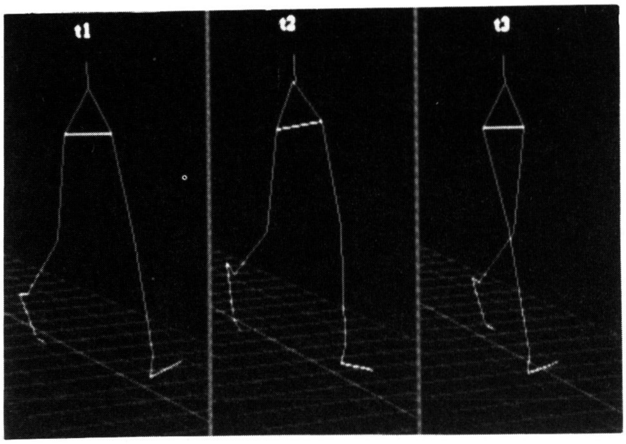

Figure 6: Pelvic list in coronal plane.

The motion of the swing leg is divided into three subphases [3], in which the leg angles are obtained by linear interpolation. The upper body motion is expressed as functions of the lower body. For example, the arm swings forward with the opposite leg, and the shoulder rotation is a function of the pelvic rotation. Both arm swing and shoulder rotation are defined as attributes and can be adjusted as desired.

3. INTERFACE

One of the major advantages of a procedural or high-level motion control system is that it does not require the animator to meticulously specify the low-level detail; in fact, producing a movement like bipedal locomotion with traditional keyframing would require enormous skills in order to get the timing and coordination of all the body parts to look right. Of course, such a procedural approach only becomes useful in practice if it is not completely hard-coded, i.e. it allows the user to flexibly choose different instances of a particular motion. The choice of the parameters to specify a desired motion is therefore crucial. It is also important to provide interactive and real-time control, so that the animator can quickly create and shape a movement idea.

In our system, three locomotion parameters can be set for the current step to achieve a specific stride: step length, velocity and step frequency, as shown in Figure 7. In *normal* mode, if one of the parameters is changed via a slider, the other two are automatically adjusted to maintain a "natural" gait according to the normalizing formulae [3]. In *locked* mode, where one of the three parameters is locked at the current setting, the other two can be adjusted via sliders. For instance, to generate a slow walk at a large step length, the step length would be locked at a large value and then the velocity slider would be set to a slow speed; this is shown in Figure 7, where step length was locked at 0.75 m and the velocity was then reduced from 5 km/h to 3 km/h.

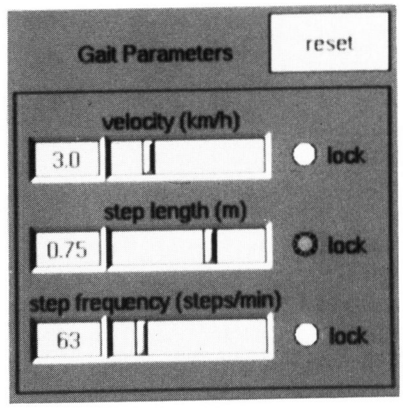

Figure 7: Locomotion parameter panel.

In addition to the locomotion parameters, 15 locomotion attributes are also provided to individualize walks, that is to produce walks at the same step length, step frequency and velocity, but with different characteristics such as upper body tilt or leg bounciness. These attributes are illustrated in Figure 8; there are 5 attributes for varying the movements of the arms: shoulder rotation, arm swing (sagittal plane), arm out (coronal plane), minimum elbow flexion and maximum elbow flexion. There are 2 attributes for the torso: forward tilt and sway; 2 for the pelvis: rotation (transverse plane) and list (coronal plane). Finally, there are 6 attributes for the movement of the legs: bounciness, minimum knee flexion during stance, knee flexion at impact, minimum toe clearance during swing, foot angle and stride width.

All of these attributes as well as the parameters are initially set to default values and can be adjusted interactively via sliders while the motion of a human figure is displayed on the screen, as shown in Figure 9. Of course, many more parameters are conceivable to further personalize locomotion; however, experience has shown that too many variables lead to confusion and make it difficult to predict the outcome. One solution is to provide overlays, that is to allow the user to specify the parts of the body for which the movement should be generated by the procedural algorithm, and the parts for which movement should be determined by other sources (e.g. keyframing, rotoscoping). We have experimented with this idea in our human animation system [5].

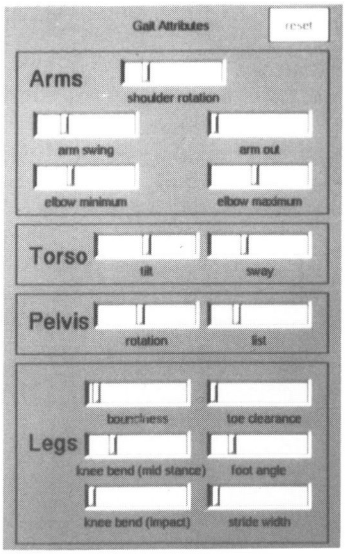

Figure 8: Locomotion attribute panel.

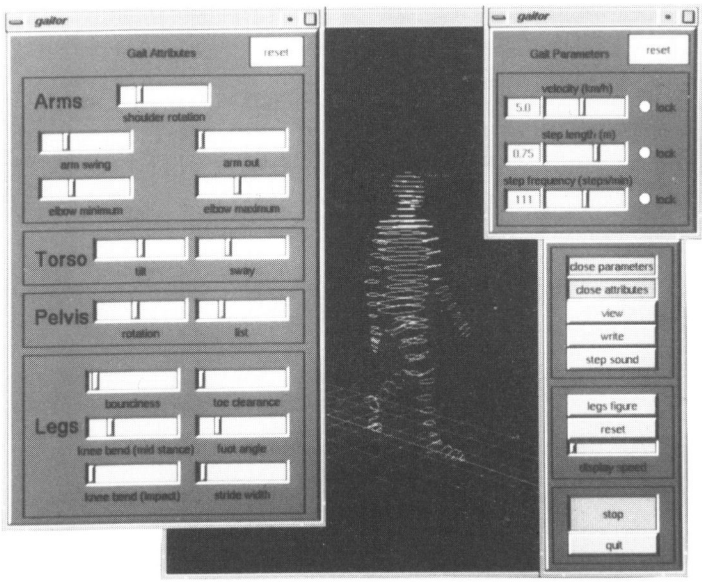

Figure 9: Interface of locomotion system.

4. DISCUSSION

The system computes a total of 56 angles for 37 joints of the default body model (24 of these joints are between vertebrae in the spine) plus a position vector in space for each time step. Different sizes and shapes of bodies can be accounted for (Figure 10). Since the algorithm is step-oriented, changes in the locomotion parameters and attributes over time become active with the granularity of one step. This allows for acceleration and deceleration in the locomotion, including starting and stopping. The program is implemented as a producer-consumer, double-buffer problem synchronized with semaphores. One process calculates all the joint angles for one locomotion step based on the current parameters and attributes writing into one buffer, while the other process handles the display and interactions reading from the other buffer. As long as the "producer" process can compute a step faster than the "consumer" process is able display the previous step, the user can adjust parameters and attributes as the figure is walking without noticeable delays. On a Silicon Graphics R3000 Indigo workstation, this real-time feedback is achieved when our contour line-drawing human figure is displayed.

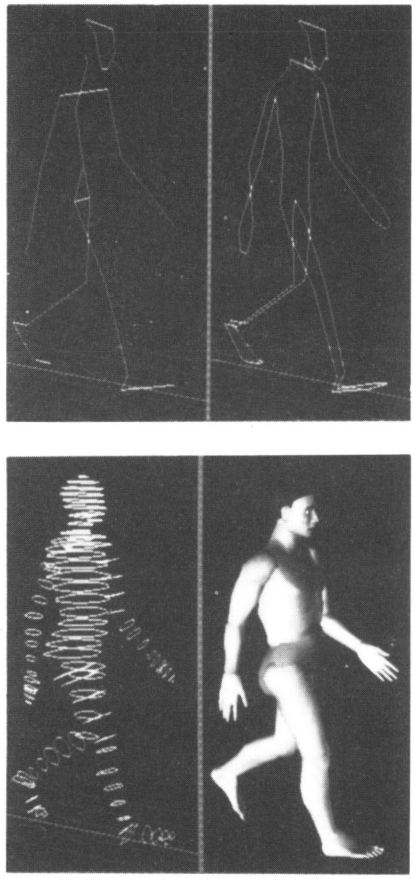

Figure 10: Different body models.

Figure 11 illustrates a variety of walks expressing different personalities and moods which were obtained by altering the default values of some parameters and attributes; a normal, proud, muscle-man, marching, bouncy, loose, happy, and tired walk are displayed (left-to-right, top-to-bottom). The "normal" walk was generated from the default values for all parameters and attributes. For the "proud" walk, the velocity was set to 4 km/h, then it was locked and the step length was increased from its default value of 0.65 m to 0.75 m; the values for foot angle, arm swing, arm out and shoulder rotation were then increased to almost their maximum values. As another example, the "tired" walk was produced by reducing the velocity to 3 km/h and the arm swing to almost a minimum, while increasing the values for torso tilt, sway and bounciness slightly.

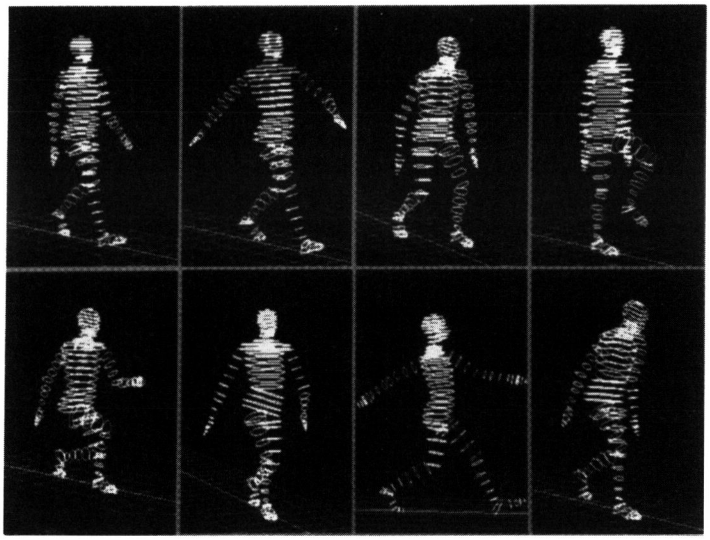

Figure 11: Variety of walks.

Compared to the original KLAW system [3], which did not perform in real-time but which utilized an underlying dynamic model to produce realistic looking walking cycles driven by forces and torques, the system introduced here trades off dynamic realism with real-time feedback. In fact, the difference in the quality of motion is surprisingly small as shown in Figure 12 which compares an instance of the same sequence computed once by the dynamic and once by the new system. The interactive feature of the new system is well suited to rapid prototyping of personalized human locomotion, and once all parameters for a desired walk are adjusted, the original KLAW system could then be used off-line to generate a sequence based on these parameters at slightly better quality. Whereas the original dynamically based system provided 28 locomotion attributes, the new system uses only 15 as described above. This is because some of the attributes only controlled the dynamic generator, or were found to have a very small visual impact. In spite of this, the new system can generate a larger variety of sequences; for example a very "bouncy" walk, which wasn't possible in the dynamic system, because it would have caused numerical instabilities due to a loose spring in the leg model.

5. CONCLUSIONS

This paper has shown that procedural techniques are possible and useful in human figure animation. By adopting a hierarchical approach to the control of locomotion, the load on the animator can be minimized in producing realistic animation of a wide variety of personalized human walks. Compared to keyframing, a higher level of control is provided by specifying movements through parameters rather than joint-angles. This approach is well suited for scripted or even task-oriented animation, where synthetic actors perform autonomously based on a script. We are currently developing a scripting language interface for human locomotion and other human motions. We are also in the process of extending the concepts in this paper to human running, as well as to locomotion along arbitrary paths.

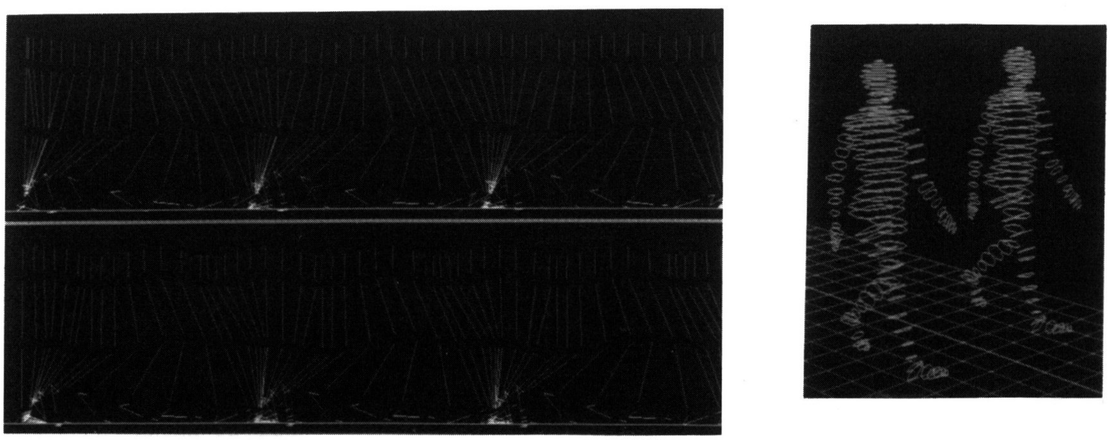

Figure 12: Comparison of dynamic (top and right) and kinematic walk (bottom and left).

6. REFERENCES

1. W. W. Armstrong, M. Green, "The Dynamics of Articulated Rigid Bodies for the Purposes of Animation", *Graphics Interface'85, Proceedings*, 1985, pp. 407-415.

2. N. Badler, "Animating Human Figures: Perspectives and Directions", *Graphics Interface'86, Proceedings*, 1986, pp. 115-120.

3. A. Bruderlin, T.W. Calvert, "Goal-Directed, Dynamic Animation of Human Walking", *Computer Graphics (ACM SIGGRAPH'89), Proceedings*, vol. 23, July 1989, pp. 233-242.

4. A. Bruderlin, T.W. Calvert, "Animation of Human Gait", *in Adaptability of Human Gait, Advances in Psychology Series, Elsevier Science Publishers B.V.*, North Holland, 1991, pp. 305-330.

5. T.W. Calvert, C. Welman, S. Gaudet, C. Lee, "Composition of Multiple Figure Sequences for Dance and Animation", *CG International'89, New Advances in Computer Graphics, Proceedings*, June 1989, pp. 245-255.

6. T.W. Calvert, "The Challenge of Human Figure Animation", *Graphics Interface'88, Proceedings*, 1988, pp. 203-210.

7. M. Girard, A. Maciejweski, "Computational Modeling for the Computer Animation of Legged Figures", *Computer Graphics (ACM SIGGRAPH'85), Proceedings*, vol. 19, July 1985, pp. 263-270.

8. J. Hodgins, P. Sweeney, D. Lawrence, "Generating Natural-Looking Motion For Computer Animation", *Graphics Interface'92, Proceedings*, 1992, pp. 265-272.

9. V. T. Inman, H. J. Ralston, F. Todd, "Human Walking", *Williams & Wilkins*, Baltimore, 1981.

10. P. Isaacs, M. Cohen, "Controlling Dynamic Simulation with Kinematic Constraints, Behavior Functions and Inverse Kinematics", *Computer Graphics (ACM SIGGRAPH'87), Proceedings*, vol. 21, July 1987, pp. 215-224.

11. D. Kochanek, R. Bartels, "Interpolating Splines with Local Tension, Continuity and Bias Control", *Computer Graphics (ACM SIGGRAPH'84), Proceedings*, vol. 18, !984, pp. 33-41.

12. P. Lee, S. Wei, J. Zhao, N. Badler, "Strength Guided Motion", *Computer Graphics (ACM SIGGRAPH'90), Proceedings*, vol. 24, August 1990, pp. 253-262.

13. N. Magnenat-Thalman, D. Thalman, "The Use of High-Level 3-D Graphical Types in the MIRA Animation System", *IEEE Computer Graphics and Applications*, 3, 9, December 1983, pp. 9-16.

14. C. Phillips, N. Badler, "Interactive Behaviors for Bipedal Articulated Figures", *Computer Graphics (ACM SIGGRAPH'91), Proceedings*, vol. 25, July 1991, pp. 359-362.

15. M. Raibert, J. Hodgins, "Animation of Dynamic Legged Locomotion", *Computer Graphics (ACM SIGGRAPH'91), Proceedings*, vol. 25, July 1991, pp. 349-358.

16 D. Sturman, "Interactive Keyframe Animation of 3-D Articulated Models", *Graphics Interface'86, Tutorial on Computer Animation*, 1986.

17. J. Wilhelms, "Using Dynamic Analysis to Animate Articulate Bodies such as Humans and Robots", *Graphics Interface'86, Proceedings*, 1985, pp. 97-104.

18. D. Zeltzer, "Towards an Integrated View of 3-D Computer Character Animation", *Graphics Interface' 85, Proceedings*, 1985, pp. 105-115.