Animating Human Walking

Shang-Lin Chen

May 17, 2003

1 Summary

Simulating realistic human locomotion on a computer is one of the challenges of computer graphics. Even walking, one of the most basic of human motions, is complex and difficult to simulate. When we walk, we hardly ever consider angles, coordinates, or partial derivatives. Yet once we move onto a computer, walking becomes a complicated process that involves both kinematics and dynamics, encompassing angles, coordinates, partial derivatives, matrices, and much more.

This project simulates a basic model of human walking using a combination of inverse kinematics and forward dynamics. We do not intend to simulate a model of "real" human walking with all the muscles and joints. Instead, we simulate an abstracted version that preserves the essence of walking and produces a reasonably realistic animation.

2 Abstraction

The essence of walking is the movement of one leg while the entire body leans forward. To clearly illustrate these basic mechanisms of walking, we can use two sticks and two hinges to represent the legs. The upper leg (a stick) is attached to the body by a hinge called the "hip joint" and the lower leg (another stick) is attached to the upper leg by a hinge called the "knee joint." The hip joint allows the upper leg to rotate forward and backward around its vertical position, but the knee joint only allows the lower leg to rotate backward with regard to the upper leg.

The basic model of two sticks and two hinges is implemented in the first phase of the project. In this phase we focus solely on the legs. After the motion of the legs has been animated correctly, a stick-like torso and head are added to the legs, producing a walking stick person. Finally, the torso is converted to three dimensions for more realism.

The resulting person has legs, knees, arms, a rectangular torso, and a spherical head. The motion of the person is limited to walking along a straight line on a solid flat surface; other motions, such as running and jumping, have not been implemented. Other aspects that affect gait, such as differing body builds, emotions, and walking surfaces, have also been omitted.

3 Mathematical Representation

The process of walking can be simplified by dividing it into states and implementing a finite state machine [4, 2]. In this project the motion of walking is

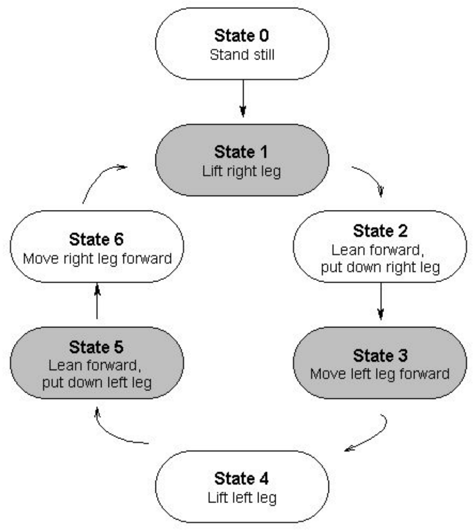

Figure 1: State machine.

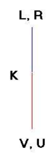

Figure 2: Side view of State 0.

divided into seven states (Fig. 1). Within each state, inverse kinematics are used to calculate the positions of the legs and joints of the animated person. This means that the final position in each state is calculated first. Then the intermediate positions are obtained for each animation frame. In States 2 and 5 forward dynamics are also used. Dynamics is the application of physical forces; the "forward" means that the final position is not calculated beforehand.

State 0 In State 0, the starting state, the person is at rest. Both legs are straight and stationary (Fig. 2). Once the person starts walking, State 0 only occurs again when the person stops.

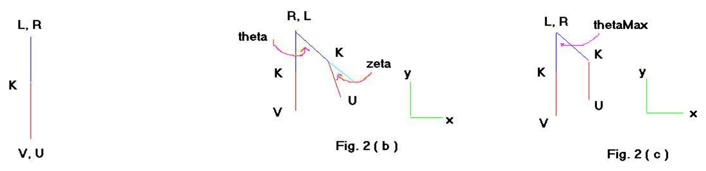

Figure 3: State 1. (a) Starting position. (b) Intermediate position. (c) Final position.

State 1 Fig. 3 depicts State 1, the lifting of the right leg. The right leg is lifted until the lower right leg hangs down vertically and the upper right leg forms an angle θ_{max} with the vertical. The left leg remains straight and stationary.

The initial position of the tip of the right leg, point U (x_0, y_0) in Fig. 3(a), is obviously

$$x_0 = 0,\ y_0 = 0 \tag{1}$$

but in principle can be any point. The final position of the tip of the right leg, point U (x_f, y_f) in Fig. 3(c), is given by

$$x_f = u \sin\theta_{max},\ y_f = u - u\cos\theta_{max}, \tag{2}$$

where θ_{max} is an input parameter that specifies how high the upper right leg should be lifted, u is the length of the upper leg, and L is the length of the lower leg.

At any time during this state, the upper right leg forms an angle θ with the vertical and an angle ζ with the lower right leg. The position of the tip of the right leg, point U (x, y) in Fig. 3(b), can be expressed in terms of angles θ and ζ as:

$$\begin{aligned} x &= u\sin\theta + L\sin(\theta - \zeta), \\ y &= (u + L) - u\cos\theta - L\cos(\theta - \zeta). \end{aligned} \tag{3}$$

After finding the partial derivatives of (3), we obtain the transformation:

$$\begin{pmatrix} \triangle x \\ \triangle y \end{pmatrix} = J \begin{pmatrix} \triangle\theta \\ \triangle\zeta \end{pmatrix}, \tag{4}$$

where J, the Jacobian of the transformation, is a 2 by 2 matrix of four elements:

$$\begin{pmatrix} \frac{\partial x}{\partial\theta} & \frac{\partial x}{\partial\zeta} \\ \frac{\partial y}{\partial\theta} & \frac{\partial y}{\partial\zeta} \end{pmatrix}.$$

We can solve for $\triangle\theta$ and $\triangle\zeta$ in terms of $\triangle x$ and $\triangle y$ using the inverse Jacobian matrix, or J^{-1}.

$$\begin{pmatrix} \triangle\theta \\ \triangle\zeta \end{pmatrix} = J^{-1} \begin{pmatrix} \triangle x \\ \triangle y \end{pmatrix} \tag{5}$$

The inverse Jacobian can be determined in terms of the determinant of the Jacobian:

$$J^{-1} = \frac{1}{\det J} \begin{pmatrix} \frac{\partial y}{\partial\zeta} & -\frac{\partial x}{\partial\zeta} \\ \frac{\partial y}{\partial\theta} & \frac{\partial x}{\partial\theta} \end{pmatrix}.$$

$\det J = \frac{\partial x}{\partial\theta} \cdot \frac{\partial y}{\partial\zeta} - \frac{\partial x}{\partial\zeta} \cdot \frac{\partial y}{\partial\theta}$ is calculated explicitly as (from (3)):

$$\det J = u \cdot L \cdot \sin\zeta. \tag{6}$$

To calculate $\triangle\theta$ and $\triangle\zeta$ we still need the values of $\triangle x$ and $\triangle y$. In this state the motion of the tip of the right leg is mostly linear. We already know both the initial and the final coordinates of the tip of the right leg. Thus, by setting $\triangle x = (x_f - x_0)/N$ and $\triangle y = (y_f - y_0)/N$, at each given value of θ and ζ, where N is an input integer to specify how fine the animation between one frame to

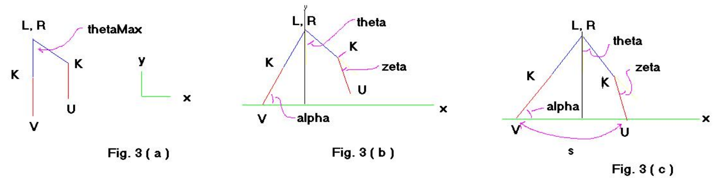

Figure 4: State 2. (a) Initial position. (b) Stepping and leaning forward. (c) Final position.

the next should be, we can calculate $\triangle\theta$ and $\triangle\zeta$ from (5). The new values of the angles θ and ζ are:

$$\begin{aligned}\theta &= \theta + \triangle\theta,\\ \zeta &= \zeta + \triangle\zeta.\end{aligned} \tag{7}$$

Once the angles θ and ζ are known, we can determine the coordinates of all the critical points of the leg at any time within State 1.

However, this direct method can lead to erroneous results. When the determinant of J is zero, the inverse Jacobian becomes singular, and directly solving for $\triangle\theta$ and $\triangle\zeta$ produces unacceptable results. This problem occurs when $\sin\zeta = 0$ — that is, when the entire leg forms a straight line at the beginning of the state. To circumvent singularities, small predetermined values of $\triangle\zeta$ and $\triangle\theta$ are used instead of the values of $\triangle\theta$ and $\triangle\zeta$ from (5). When the lengths of the upper leg and lower leg are both set to unit length, $|\sin\zeta| < 0.1$ turns out to be an adequate criterion to consider J^{-1} singular. Throughout this project, singularities can be calculated explicitly from the characteristics of each state, so other algorithms for calculating pseudo-inverses [4, 2] are not required.

State 2 In State 2, the person steps forward with the right leg and leans forward with the left leg. The initial position in this state is the same as the final position of State 1 (Fig. 4a). The tip of the right leg, point U, is initially at the position $(x_1,\ y_1)$ as given by (2). The final position of point U (x_2, y_2) as in Fig. 4(c) is given as

$$x_2 = s,\ y_2 = 0, \tag{8}$$

where s is a given parameter to specify how long a step should be. The angle α in Fig. 4(b) describes the rate with which the animated person's body leans forward. α can be calculated dynamically using the torque caused by gravity and Newton's equations for an extended rigid body with the moment of inertia. In general it takes the form

$$\alpha = c_2t^2 + c_1t + c_0, \tag{9}$$

where c_2, c_1, c_0 are input parameters equivalent to specifying the weight and the size of the whole body, and t is time but can be considered just as a parameter. We can cut t into small increments $\triangle t$ (a given input), and compute $t = t + \triangle t$ starting from $t = 0$ for each frame. Thus each frame will have a given angle α.

The position of the tip of the right leg, point U (x, y), in Fig. 4(b) is

$$\begin{aligned}x &= (u + L)\cos\alpha + u\sin\theta + L\sin(\theta - \zeta)\\ y &= (u + L)\sin\alpha - u\cos\alpha - L\sin(\theta - \zeta).\end{aligned} \tag{10}$$

Again using inverse kinetics, we can obtain

$$\begin{pmatrix} \triangle\theta \\ \triangle\zeta \end{pmatrix}$$

as

$$\begin{pmatrix} \triangle\theta \\ \triangle\zeta \end{pmatrix} = J^{-1} \begin{pmatrix} \triangle x \\ \triangle y \end{pmatrix} \tag{11}$$

where $\triangle x$ and $\triangle y$ need to be specified frame by frame as will be discussed later, a very different situation from State 1.

The Jacobian J is again a 2 by 2 matrix of four elements

$$\begin{pmatrix} \frac{\partial x}{\partial\theta} & \frac{\partial x}{\partial\zeta} \\ \frac{\partial y}{\partial\theta} & \frac{\partial y}{\partial\zeta} \end{pmatrix}.$$

The four elements of partial differentiation, of course, must be calculated from (10).

As in State 1, singularities occur when $\sin\zeta$ is close to 0. To handle singularities, $\triangle\theta$ and $\triangle\zeta$ are calculated explicitly using the approximation $\cos\zeta \approx 1$ when $\zeta \to 0$:

$$\begin{aligned} \sin\zeta\triangle\theta &= (u+L)\cdot(-\sin\theta\cdot\triangle x + \cos\theta\cdot\triangle y) \\ &\quad + L\ \cdot(\cos\theta\cdot\triangle x + \sin\theta\cdot\triangle)\cdot\sin\zeta \\ \triangle\zeta &= \triangle\theta. \end{aligned} \tag{12}$$

When J^{-1} becomes singular, we just choose $\triangle x$ and $\triangle y$ such that

$$-\sin\theta\cdot\Delta x + \cos\theta\cdot\Delta y = 0. \tag{13}$$

Then the singularity of J^{-1} is resolved, yielding

$$\triangle\theta = \triangle\zeta = L(\cos\theta\cdot\triangle x + \sin\theta\cdot\triangle y). \tag{14}$$

Again $|\sin\zeta| \leq 0.1$ can be considered to be the singular region of J^{-1} such that equation (14) applies.

To implement this method of removing singularities, one of $(\triangle x, \triangle y)$ can be chosen rather arbitrarily. For example, say

$$\triangle x = \frac{final_x - current_x}{n}$$

where n is an appropriately chose interval-generating integer. Then $\triangle y$ is computed according to (13). Outside the singular region, both $\triangle x$ and $\triangle y$ can be chosen at will; we use

$$\begin{aligned} \triangle x &= \frac{final_x - current_x}{n} \\ \triangle y &= \frac{final_y - current_y}{n}. \end{aligned}$$

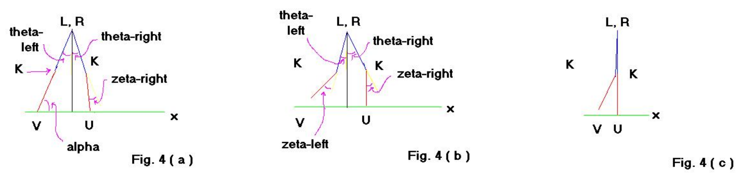

Figure 5: State 3. (a) Initial position. (b) The person pulls forward left leg and straightens right leg. (c) The right leg is straightened.

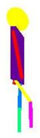

Figure 6: Complete person with torso, arms, and head.

State 3 In State 3 the right leg is straightened while the left leg is pulled forward (Fig. 5b). The critical moving point in this state is the hip joint, whose coordinates are denoted as (x, y). The initial position of the right hip joint in this state is the same as the final position in the previous state, State 2. The final position of the right hip joint is:

$$final_x = Rtip_x, \text{ and } final_y = Rtip_y,$$

where $Rtip_x$ and $Rtip_y$ are the x and y coordinates of the tip of the right leg. Throughout this state the right tip serves as a pivot and does not move.

The x and y coordinates of the right hip joint at any time during this state can be expressed in terms of angles θ and ζ as:

$$\begin{aligned} x &= Rtip_y - L \cdot \sin(\theta - \zeta) - u \cdot \sin\theta, \\ y &= Rtip_y + L \cdot \cos(\theta - \zeta) + u \cdot \cos\theta. \end{aligned} \tag{15}$$

The inverse Jacobian derived from the partial derivatives of the above equations turns out to be in the same form as the Jacobian in State 2, so we can use the same method of avoiding singularities.

During this state, the left leg is pulled forward with a predetermined $\triangle\theta_L$ and $\triangle\zeta_L$ in each frame. At the end of the state, the left leg is not required to be straightened, thus producing a more natural-looking gait.

States 4-6 States 4 to 6 are the mirror images of States 1 to 3. State 4, lifting the left leg, has very similar mathematics to State 1. Similarly, State 5 corresponds to State 2 and State 6 corresponds to State 3.

Once the states of the legs have been calculated, arms, a torso, and a head can be added (Fig. 6). The torso and head are translated along with the legs, while the arms swing synchronously.

4 Algorithms

This project is written in C++ with OpenGL for rendering and GLUT for user interactivity. The structure of the resulting program is a straight-forward state machine. A state controller class tracks the current state and performs the necessary calculations. Each state performs the calculations discussed in the previous sections to find the coordinates of the critical points of the animated person's legs.

The state machine is initially in State 0. Once the animation starts, the state machine loops through States 1-6. The machine does not return to State 0 unless the animated person stops walking to change direction.

5 Results

The implementation of this project displays a person walking along the x-axis. The default structure of the person is as described in the Abstraction section. Pressing F1 changes the display to show a stick person. Pressing F2 changes the display to show only the legs. The left and right arrow keys cause the person to walk left or right, respectively. When the arrow key opposite to the current direction is pressed, the person returns to State 0, appears to turn 180 degrees, and begins walking again, starting from State 0.

An interesting phenomenon occurs whenever the tip of a leg reaches the ground. Upon reaching the x-axis, the tip of the leg shifts backward slightly. The reason for this is still unclear.

6 Conclusions

The method used in this project, inverse kinematics with some forward dynamics, produces an acceptably realistic simulation of walking using fairly simple calculations. However, since this method does not involve much dynamics, the realism of the animated person is limited. For example, if θ_{max} and the step size are changed to values that cause a real person to fall, the animated person will go into violent, unpredictable motion instead of falling. This occurs because physical forces are not taken into account.

One unexpected difficulty encountered in the project is changing direction. Right now, a camera trick is used so that the animated person appears to change direction by stopping and turning 180 degrees. However, changing direction while walking without changing the camera position involves much more complex calculations and was not implemented successfully. Another difficulty, as mentioned in the previous section, occurs whenever the tip of a leg touches the ground and shifts backward slightly.

Related projects that may be of interest include adding more realism to the upper body [1], using more dynamics, implementing motions other than walking, and walking on uneven terrain.

7 Appendix

7.1 Other Methods

The method used in this project to animate human walking is only one of many possible methods. Current researchers favor using dynamics instead of kinematics [3]. The dynamics method relies on applying physical forces to the parts of the body for a realistic animation. The results of using dynamics can be more realistic than kinematics, but the process is more complex, both theoretically and computationally [2]. Applying physics requires knowledge of biomechanics – for example, the mass distribution of the human body and the mechanics of the muscles involved in walking. Kinematics, on the other hand, does not require knowledge of biomechanics beyond basic restrictions on joints' angles of rotation.

Yet another method uses motion capture. In this method, movement data is recorded from sensors attached to the body of a subject, either a person or an animal. Realistic-looking motion is then animated from the collected data. Motion capture can also be accomplished using a constructed physical model instead of a live subject. Many movies and video games use motion capture to create animated characters — for example, the Gollum character in the recent *Lord of the Rings* films.

7.2 Screen Shots

The following pages contain screen shots of the project implementation.

References

[1] Koji Ashida and Seung-Joo Lee et al. Pedestrians: Creating agent behaviors through statistical analysis of observation data. *Computer Animation*, 2001.

[2] Franck Multon et al. Computer animation of human walking: a survey. Technical report, INRIA, June 1998.

[3] Hyeongseok Ko and Norman I. Badler. Animating human locomotion with inverse dynamics. *Computer Graphics and Applications*, 16:50–59, March 1996.

[4] Michael Girard and A. A. Maciejewski. Computational modeling for the computer animation of legged figures. *SIGGRAPH*, 19:263–270, July 1985.

[5] Daniel Thalmann. *Dynamic Simulation as a Tool for Three-Dimensional Animation*, pages 257–272. John Wiley and Sons, Chichester, UK, 1991.