CHAPTER

9 Modeling and Animating Human Figures

Modeling and animating an articulated figure is one of the most formidable tasks that an animator can be faced with. It is especially challenging when the figure is meant to realistically represent a human. There are several major reasons for this. First, the human figure is a very familiar form. This familiarity makes each person a critical observer. When confronted with an animated figure, a person readily recognizes when its movement does not "feel" or "look" right. Second, the human form is very complex, with more than two hundred bones and six hundred muscles. When fully modeled with linked rigid segments, the human form is endowed with approximately two hundred degrees of freedom (DOFs). The deformable nature of the body's surface and underlying structures further complicates the modeling and animating task. Third, human-like motion is not computationally well defined. Some studies have tried to accurately describe human-like motion, but typically these descriptions apply only to certain constrained situations. Fourth, there is no one definitive motion that is humanlike. Differences resulting from genetics, culture, personality, and emotional state all can affect how a particular motion is carried out. General strategies for motion production have not been described, nor have the nuances of motion that make each of us unique and uniquely recognizable. Although the discussion in this chapter focuses primarily on the human form, many of the techniques apply to any type of articulated figure.

Table 9.1 provides the definitions of the anatomical terms used here. Particularly noteworthy for this discussion are the terms that name planes relative to the human figure: *sagittal*, *coronal*, and *transverse*.

9.1 Overview of virtual human representation

One of the most difficult challenges in computer graphics is the creation of virtual humans. Efforts to establish standard representations of articulated bodies are now emerging [16] [31]. Representing human figures successfully requires solutions to several very different problems. The visible geometry of the skin can be created by a variety of techniques, differing primarily in the skin detail and the degree of representation of underlying internal structures such as bone, muscle, and fatty tissue. Geometry for hair and clothing can be simulated with a clear trade-off between accuracy and computational complexity. The way in which light interacts with skin, clothing, and hair can also be calculated with varying degrees of correctness, depending on visual requirements and available resources.

Techniques for simulating virtual humans have been created that allow for modest visual results that can be computed in real time (i.e., 30 Hz). Other approaches may give extremely realistic results by simulating individual hairs, muscles, wrinkles, or clothing threads. These methods may consequently also require excessive amounts of computation time to generate a single frame of animation. The

Table 9.1 Selected Terms from Anatomy

Sagittal plane	Perpendicular to the ground and divides the body into right and left halves
Coronal plane	Perpendicular to the ground and divides the body into front and back halves
Transverse plane	Parallel to the ground and divides the body into top and bottom halves
Distal	Away from the attachment of the limb
Proximal	Toward the attachment of the limb
Flexion	Movement of the joint that decreases the angle between two bones
Extension	Movement of the joint that increases the angle between two bones

discussion in this section focuses solely on the representation of virtual humans. It addresses animation issues where necessary to discuss how the animation techniques affect the figure's creation.

9.1.1 Representing body geometry

Many methods have been developed for creating and representing the geometry of a virtual human's body. They vary primarily in visual quality and computational complexity. Usually these two measures are inversely proportional.

The vast majority of human figures are modeled using a boundary representation constructed from either polygons (often triangles) or patches (usually nonuniform rational B-splines; NURBS). These boundary shell models are usually modeled manually in one of the common off-the-shelf modeling packages (e.g., [1] [4] [35]). The purpose of the model being produced dictates the technique used to create it. If the figure is constructed for real-time display in a game on a low-end PC or gaming console, usually it will be assembled from a relatively low number of triangular polygons, which, while giving a chunky appearance to the model, can be rendered quickly. If the figure will be used in an animation that will be rendered off-line by a high-end rendering package, it might be modeled with NURBS patch data, to obtain smooth curved contours. Factors such as viewing distance and the importance of the figure to the scene can be used to select from various levels of detail at which to model the figure for a particular sequence of frames.

Polygonal representations

Polygonal models typically consist of a set of vertices and a set of faces. Polygonal human figures can be constructed out of multiple objects (frequently referred to as segments), or they can consist of a single polygonal mesh. When multiple objects are used, they are generally arranged in a hierarchy of joints and rigid segments. Rotating a joint rotates all of that joint's children (e.g., rotating a hip joint rotates all of the child's leg segments and joints around the hip). If a single mesh is used, then rotating a joint must deform the vertices surrounding that joint, as well as rotate the vertices in the affected limb.

Various constraints may be placed on the polygonal mesh's topology depending on the software that will be displaying the human. Many real-time rendering engines require polygonal figures to be constructed from triangles. Some modeling programs require that the object remain closed.

Polygonal representations are primarily used either when rendering speed is of the essence, as is the case in real-time systems such as games, or when topological flexibility is required. The primary problem with using polygons as a modeling primitive is that it takes far too many of them to represent a smoothly curving surface. It might require hundreds or thousands of polygons to achieve the same visual quality as could be obtained with a single NURBS patch.

Patch representations

Virtual humans constructed with an emphasis on visual quality are frequently built from a network of cubic patches, usually NURBS. The control points defining these patches are manipulated to sculpt the surfaces of the figure. Smooth continuity must be maintained at the edges of the patches, which often proves challenging. Complex topologies also can cause difficulties, given the rectangular nature of the patches. While patches can easily provide much smoother surfaces than polygons in general, it is more challenging to add localized detail to a figure without adding a great deal of more global data. Hierarchical splines provide a partial solution to this problem [21].

Other representations

Several other methods have been used for representing virtual human figures. However, they are used more infrequently because of a lack of modeling tools or because of their computational complexity.

Implicit surfaces (Chapter 12.1) can be employed as sculpting material for building virtual humans. Frequently the term "metaballs" is used for spherical implicit surfaces. Metaballs resemble clay in their ability to blend with other nearby primitives. While computationally expensive to render, they provide an excellent organic look that is perfect for representing skin stretched over underlying tissue [35] [41].

Subdivision surfaces (Chapter 12.3) combine the topological flexibility of polygonal objects with the resultant smoothness of patch data. They transform a low-resolution polygonal model into a smooth form by recursively subdividing the polygons as necessary [17] [19].

Probably the most computationally demanding representation method is volumetric modeling. While all of the previously mentioned techniques merely store information about the surface qualities of a virtual human, volumetric models store information about the entire interior space as well. Because of its extreme computational requirements, this technique is limited almost exclusively to the medical research domain, where knowledge of the interior of a virtual human is crucial.

As computers continue to become more powerful, more attempts are being made to more accurately model the interior of humans to get more realistic results on the visible surfaces. There have been several "layered" approaches, where some attempt has been made to model underlying bone and/or muscle and its effect on the skin.

Chen and Zeltzer [14] use a finite element approach to accurately model a human knee, representing each underlying muscle precisely, based on medical data. Several authors have attempted to create visually reasonable muscles attached to bones and then generate skin over the top of the muscle (e.g., [57] [66]). Thalmann's lab takes the interesting hybrid approach of modeling muscles with metaballs, producing cross sections of these metaballs along the body's segments, and then lofting polygons between the cross sections to produce the final surface geometry [10]. Chadwick et al. [13] use free form deformations (FFDs) to produce artist-driven muscle bulging and skin folding, as described in Section 9.1.5.

9.1.2 Geometry data acquisition

Geometric data can be acquired by a number of means. By far the most common method is to have an artist create the figure using interactive software tools. The quality of the data obtained by these means is of course completely dependent on the artist's skills and experience. Another method of obtaining data, digitizing real humans, is becoming more prevalent as the necessary hardware becomes more affordable. Data representing a specific individual are often captured using a laser scanner or by performing image processing on video images [28] [33] [43].

There have also been various parametric approaches to human figure data generation. Virtual human software intended for use in ergonomics simulation tends to use parameters with a strong anthropometric basis [6]. Software with an artistic or entertainment orientation allows for more free-form parametric control of the generated body data [35] [59]. A few efforts use exemplar-based models to allow for data generation by mixing known attributes [9] [43].

9.1.3 Geometry deformation

For a user to animate a virtual human figure, the figure's limbs and other parts must be able to be moved and/or deformed. The method of manipulation used is largely determined by the way the figure is being represented. Very simple figures, usually used for real-time display, are often broken into multiple rigid subparts, such as a forearm, a thigh, or a head. These parts are arranged in a hierarchy of joints and segments such that rotating a joint rotates all of the child segments and joints beneath it in the hierarchy [31]. While this method is quick, it yields suboptimal visual results at the joints, particularly if the body is textured, because the rigid parts merely overlap and occlude each other at the joint.

A single skin is more commonly used for polygonal figures. When a joint is rotated, the appropriate vertices are deformed to simulate rotation around the joint. Several different methods can be used for this, and, as with most of these techniques, they involve trade-offs of realism and speed. The simplest and fastest method is to bind each vertex to exactly one bone. When a bone rotates, the vertices move along with it [7]. Better results can be obtained, at the cost of additional computation, if the vertices surrounding a joint are weighted so that their position is affected by multiple bones [41]. While weighting the effects of bone rotations on vertices results in smoother skin around joints, severe problems can still occur with extreme joint bends. FFDs have been used in this case to simulate the skin-on-skin collision and the accompanying squashing and bulging that occur when a joint such as the elbow is fully bent [13]. The precise placement of joints within the body greatly affects the realism of the surrounding deformations. Joints must be placed strictly according to anthropometric data, or unrealistic bulges will result [27] [53].

Some joints require more complicated methods for realistic deformation. Using only a simple, single three degrees of freedom (DOF) rotation for a shoulder or vertebrae can yield very poor results. A few systems have attempted to construct more complex, anatomically based joints [21] [50]. The hands in particular can require significantly more advanced deformation techniques to animate realistically [23]. Surface deformations that would be caused by changes in underlying material, such as muscle and fat in a real human, can be produced in a virtual human by a number of means. Methods range from those that simply allow an animator to specify muscular deformations [13] to those that require complex dynamics simulations of the various tissue layers and densities [49] [57]. A great deal of muscle simulation research has been conducted for facial animation. See the Chapter 10 for more details. Finally, deformations resulting from interaction with the environment have been simulated both with traditional dynamics systems and with implicit surfaces [60].

9.1.4 Surface detail

After the geometry for a virtual figure has been constructed, its surface properties must also be specified. As with the figure's geometry, surface properties can be produced by an artist, scanned from real life, or procedurally generated. Color, as well as specular, diffuse, bump, and displacement maps may be generated. Accurately positioning the resulting textures requires the generation of texture coordinates [56]. The skin may be simulated using complex physically based simulations [25] [44]. Wrinkles may also be simulated by various means [47] [60].

9.1.5 Layered approach to human figure modeling

A common approach to animating the human figure is to construct the figure in layers consisting of skeleton, muscles, and skin. The skeletal layer is responsible for the articulation of the form. The muscle layer is responsible for deforming the shape as a result of skeletal articulation. The skin is responsible for carrying the appearance of the figure.

Chadwick et al. [13] introduced the layered approach to figure animation by incorporating an articulated skeleton, surface geometry representing the skin, and an intermediate muscle layer that ties the two together. The muscle layer is not anatomically based, and its only function is to deform the surface geometry as a function of joint articulation. The muscle layer implements a system of FFD lattices in which the surface geometry is embedded. The lattice is organized with two planes on each end that are responsible for maintaining continuity with adjoining lattices, and the interior planes are responsible for deforming the skin to simulate muscle bulging (Figure 9.1). As the joint flexes, the interior planes elongate perpendicular to the axis of the link. The elongation is usually not symmetrical about the axis and is designed by the animator. For example, the upper-arm FFD lattice elongates as the elbow flexes. Typically, the FFD for the upper arm is designed to produce the majority of the skin deformation in the region of the biceps. A pair of FFD lattices is used on either side of each joint to isolate the FFDs responsible for muscle bulging from the more rigid area around the joint. In addition, the joint FFDs are designed to maintain continuity on the outside of the joint and create the skin crease on the inside of the joint (see Figure 9.2).

Artistic anatomy can be used to guide analysis of the human form [57][58][66]. Bones, muscles, tendons, and fatty tissue are modeled in order to occupy the appropriate volumes. Scheepers [57] identifies the types of muscles sufficient for modeling the upper torso of the human figure: linear muscles,

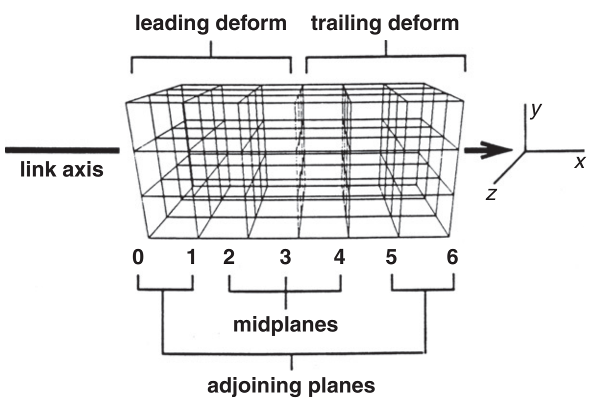

FIGURE 9.1

The basic FFD lattice [13].

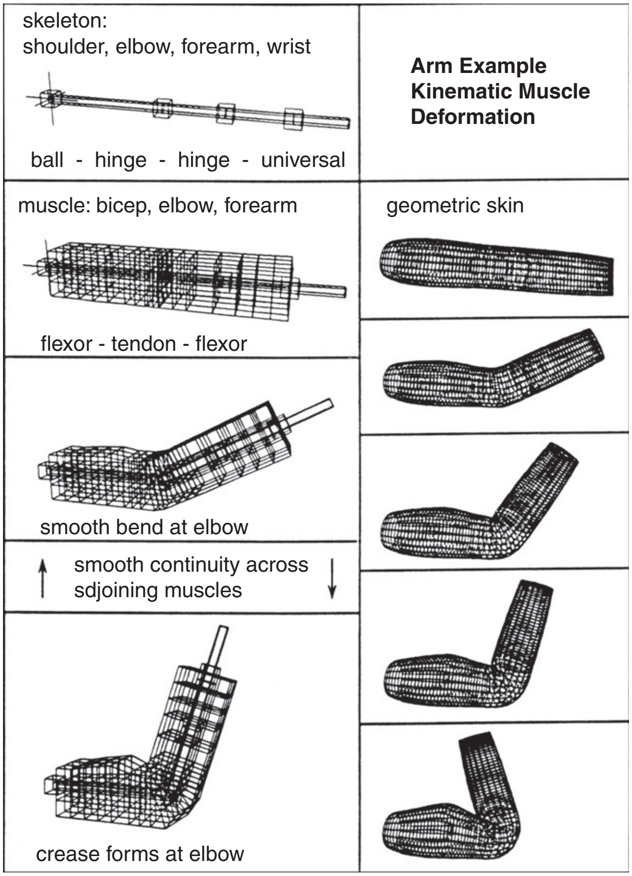

FIGURE 9.2

Deformation induced by FFDs as a result of joint articulation [13].

sheet muscles, and bendable sheet muscles. Tendons are modeled as part of the muscles and attach to the skeleton. Muscles deform according to the articulation of the skeleton. See Figure 9.3 for an example. These muscles populate the skeletal figure in the same manner that actual muscles are arranged in the human body (Figure 9.4). To deform the skin according to the underlying structure (muscles, tendons, and fatty tissue), the user defines implicit functions so that the densities occupy the volume of the corresponding anatomy. Ellipsoids are used for muscles, cylinders for tendons, and flattened ellipsoids for fat pads. The implicit primitives are summed to smooth the surface and further model underlying tissue. The skin, modeled as a B-spline surface, is defined by floating the control points of the B-spline patches to the isosurface of the summed implicit primitives. This allows the skin to distort as the underlying skeletal structure articulates and muscles deform (Figure 9.5).

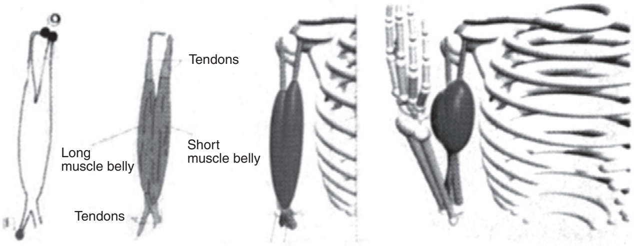

FIGURE 9.3

Linear muscle model [57].

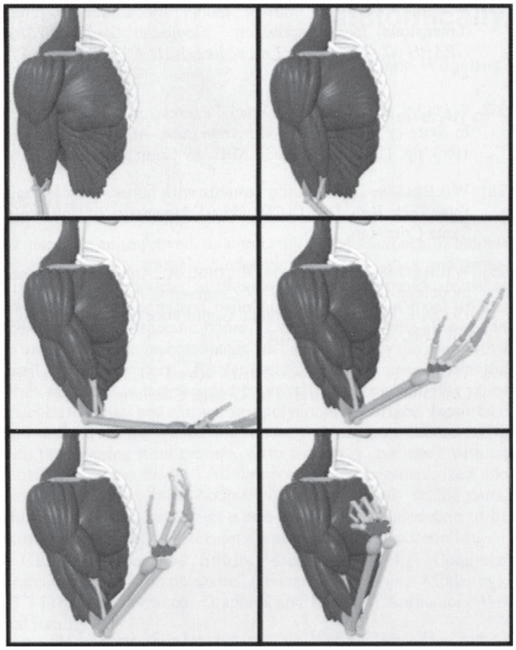

FIGURE 9.4

Muscles of upper torso [57].

Rigging

Rigging, as used here, refers to setting up interactive controls to facilitate animation of a character or other object. In the case of a character, a rig might be set up to control taking a step or waving a hand. Typically, a character rig facilitates the animation of a character by providing a high-level control that manipulates multiple DOFs in a coordinated manner. This not only makes animating the character easier, it also provides consistency for the character's movements.

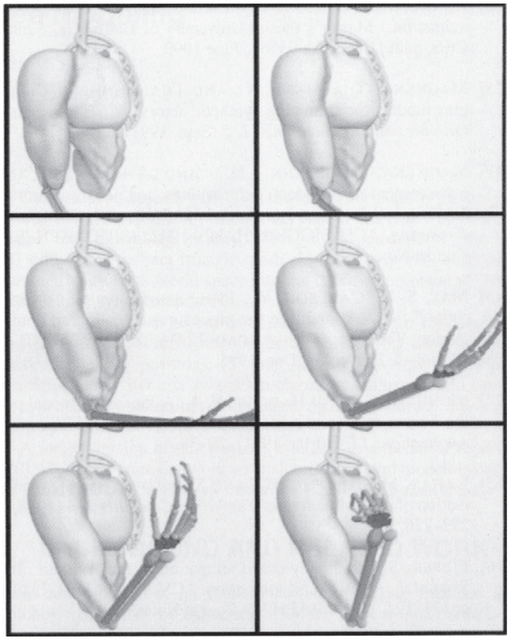

FIGURE 9.5

Skin model over muscles, tendons, and fatty tissue [57].

9.2 Reaching and grasping

One of the most common human figure animation tasks involves movement of the upper limbs. A synthetic figure may be required to reach and operate a control, raise a coffee cup from a table up to his mouth to drink, or turn a complex object over and over to examine it. It is computationally simpler to consider the arm as an appendage that moves independently of the rest of the body. In some cases, this can result in unnatural-looking motion. To produce more realistic motion, the user often involves additional joints of the body in executing the motion. In this section, the arm is considered in isolation. It is assumed that additional joints, if needed, can be added to the reaching motion in a preprocessing step that positions the figure and readies it for independently considered arm motion.

9.2.1 Modeling the arm

The basic model of the human arm, ignoring the joints of the hand for now, can be most simply represented as a seven DOFs manipulator (Figure 9.6): three DOFs are at the shoulder joint, one at the elbow, and three at the wrist. See Chapter 5 for an explanation of joint representation, forward kinematics, and inverse kinematics. A *configuration* or *pose* for the arm is a set of seven joint angles, one for each of the seven DOFs of the model.

Forearm rotation presents a problem. In Figure 9.6, the forearm rotation is associated with the wrist. However, in reality, the forearm rotation is not associated with a localized joint like most of the other

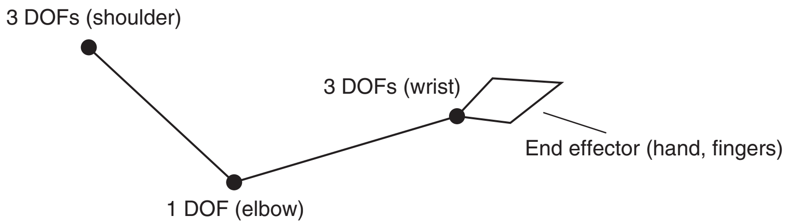

FIGURE 9.6

Basic model of the human arm.

DOFs of the human figure; rather, it is distributed along the forearm itself as the two forearm bones (radius and ulna) rotate around each other. Sometimes this rotation is associated with the elbow instead; other implementations create a "virtual" joint midway along the forearm to handle forearm rotation.

Of course, the joints of the human arm have limits. For example, the elbow can flex to approximately 20 degrees and extend to typically around 180 degrees. Allowing a figure's limbs to exceed the joint limits would certainly contribute to an unnatural look. Most joints are positioned with least strain somewhere in the middle of their range and rarely attain the boundaries of joint rotation unless forced. More subtly, joint limits may vary with the position of other joints, and further limits are imposed on joints to avoid intersection of appendages with other parts of the body. For example, if the arm is moved in a large circle parallel to and to the side of the torso, the muscle strain causes the arm to distort the circle at the back. As another example, in some individuals tendons make it more difficult to fully extend the knee when one is bending at the hip (the motion used to touch one's toes).

If joint limits are enforced, some general motions can be successfully obtained by using forward kinematics. Even if an object is to be carried by the hand, forward kinematics in conjunction with attaching the object to the end effector creates a fairly convincing motion. But if the arm/hand must operate relative to a fixed object, such as a knob, inverse kinematics is necessary. Unfortunately, the normal methods of inverse kinematics, using the pseudo-inverse of the Jacobian, are not guaranteed to give human-like motion. As explained in Chapter 5, in some orientations a singularity may exist in which a DOF is "lost" in Cartesian space. For example, the motion can be hard to control in cases in which the arm is fully extended.

According to the model shown in Figure 9.6, if only the desired end effector position is given, then the solution space is underconstrained. In this case, multiple solutions exist, and inverse kinematic methods may result in configurations that do not look natural. As noted in Chapter 5, there are methods for biasing the solution toward desired joint angles. This helps to avoid violating joint limits and produces more human-like motion but still lacks any anatomical basis for producing human-like configurations.

It is often useful to specify the goal position of the wrist instead of the fingers to better control the configurations produced. But even if the wrist is fixed (i.e., treated as the end effector) at a desired location, and the shoulder is similarly fixed, there are still a large number of positions that might be adopted that satisfy both the constraints and the joint limits. Biasing the joint angles to orientations preferable for certain tasks can reduce or eliminate multiple solutions.

To more precisely control the movement, the user can specify intermediate positions and orientations for the end effector as well as for intermediate joints. Essentially, this establishes key poses for the linkage. Inverse kinematics can then be used to step from one pose to the next so that the arm is still guided along the path. This affords some of the savings of using inverse kinematics while giving the animator more control over the final motion.

The formal inverse Jacobian approach can be replaced with a more procedural approach based on the same principles to produce more human-like motion. In human motion, the joints farther away from the end effector (the hand) have the most effect on it. The joints closer to the hand change angles in order to perform the fine orientation changes necessary for final alignment. This can be implemented procedurally by computing the effect of each DOF on the end effector by taking the cross-product of the axis of rotation, ω_1, with the vector from the joint to the end effector, V_1 (Figure 9.7). In addition, since the arm contains a one DOF angle (elbow), a plane between the shoulder, the elbow, and the wrist is formed, and the arm's preferred positions dictate a relatively limited rotation range for that plane. Once the plane is fixed, the shoulder and elbow angles are easy to calculate and can be easily adjusted on that plane (Figure 9.8). Some animation packages (e.g., Maya$^{\text{TM}}$) allow the animator to specify an inverse kinematic solution based on such a plane and to rotate the plane as desired.

Some neurological studies, notably those by Lacquaniti and Soechting [40] and Soechting and Flanders [61], suggest that the arm's posture is determined from the desired location of the end effector (roughly "fixing the wrist's orientation"), and then the final wrist orientation is tweaked for the nature

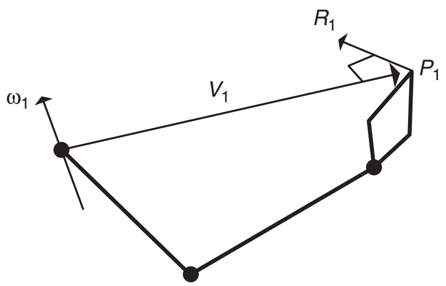

FIGURE 9.7

Effect of the first DOF on the end effector: $R_1 = \omega_1 \times V_1$.

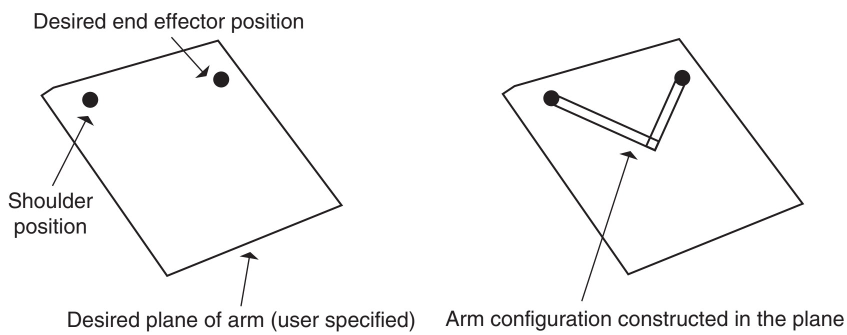

FIGURE 9.8

Constructing the arm in a user-specified plane.

of the object and the task. The model developed by Kondo [38] for this computation makes use of a spherical coordinate system. A set of angles for the shoulder and elbow is calculated from the desired hand and shoulder position and then adjusted if joint limitations are violated. Finally, a wrist orientation is calculated separately. The method is described, along with a manipulation planner for trajectories of cooperating arms, by Koga et al. [36].

9.2.2 The shoulder joint

The shoulder joint requires special consideration. It is commonly modeled as a ball joint with three coincident DOFs. The human shoulder system is actually more complex. Scheepers [57] described a more realistic model of the clavicle and scapula along with a shoulder joint, in which three separate joints with limited range provide very realistic-looking arm and shoulder motion. Scheepers also provided an approach to the forearm rotation problem using a radioulnar (mid-forearm) joint (see Figure 9.9).

9.2.3 The hand

To include a fully articulated hand in the arm model, one must introduce many more joints (and thus DOFs). A simple hand configuration may consist of a palm, four fingers, and a thumb, with joints and DOFs as shown in Figure 9.10.

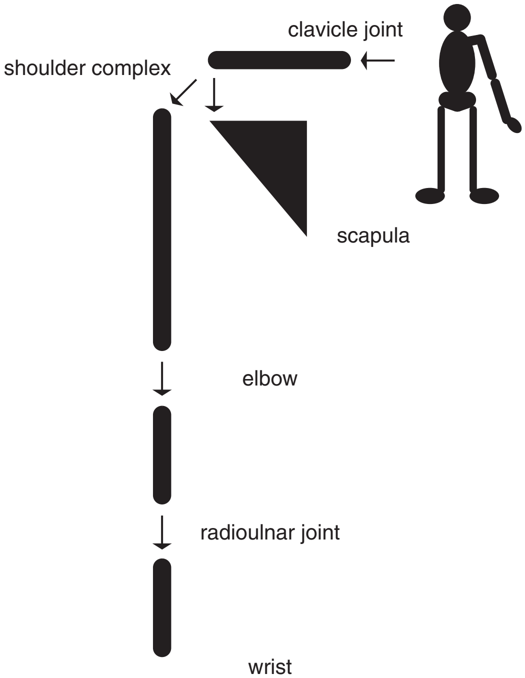

FIGURE 9.9

Conceptual model of the upper limb.

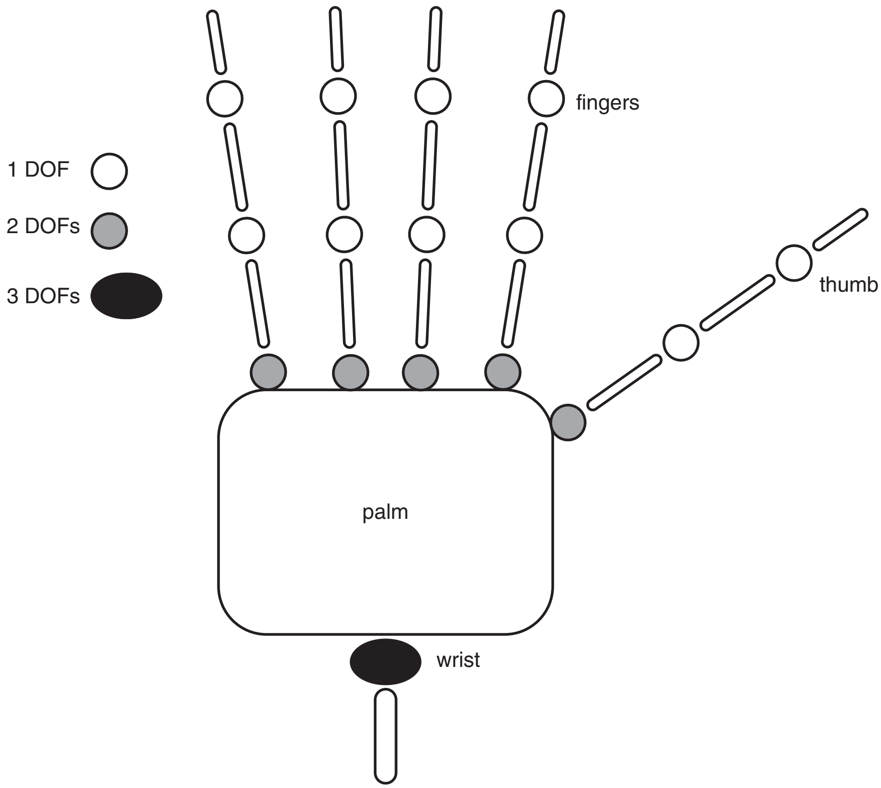

FIGURE 9.10

Simple model of hands and fingers.

A model similar to Figure 9.10 is used by Rijpkema and Girard [54] in their work on grasping. Scheepers [57] uses 27 bones, but only 16 joints are movable. Others use models with subtler joints inside the palm area in order to get human-like action.

If the hand is to be animated in detail, the designer must pay attention to types of grasp and how the grasps are to be used. The opposable thumb provides humans with great manual dexterity: the ability to point, grasp objects of many shapes, and exert force such as that needed to open a large jar of pickles or a small jewelry clasp. This requires carefully designed skeletal systems. Studies of grasping show at least 16 different categories of grasp, most involving the thumb and one or more fingers. For a given task, the problem of choosing which grasp is best (most efficient and/or most credible) adds much more complexity to the mere ability to form the grasp. Approaches to grasping are either procedural (e.g., [5] [8] [30] [46] [54]) or data driven (e.g., [18] [39] [51]).

Simpler models combine the four fingers into one surface and may eliminate the thumb (Figure 9.11). This reduces both the display complexity and the motion control complexity. Display

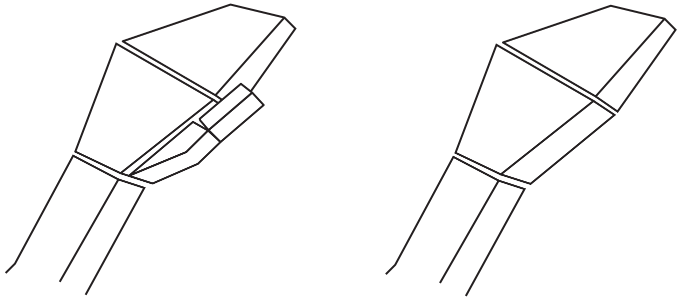

FIGURE 9.11

Simplified hands.

complexity, and therefore image quality, can be maintained by using the full-detail hand model but coordinating the movement of all the joints of the four fingers with one "grasping" parameter (Figure 9.12), even though this only approximates real grasping action.

9.2.4 Coordinated movement

Adding to the difficulties of modeling and controlling the various parts of the upper limb is the difficulty of inter-joint cooperation in a movement and assigning a specific motion to a particular joint. It is easy to demonstrate this difficulty. Stretch out your arm to the side and turn the palm of the hand so it is first facing up; then rotate the hand so it is facing down and try to continue rotating it all the way around so that the palm faces up again. Try to do this motion by involving first only the hand/wrist/forearm and

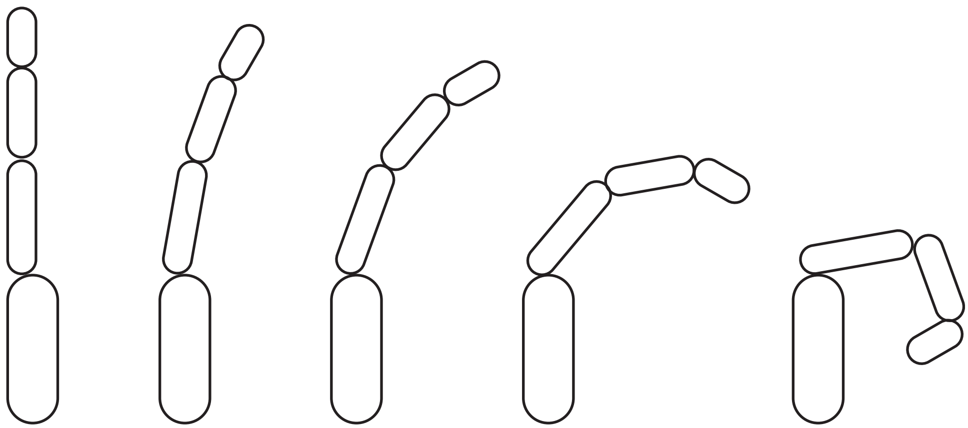

FIGURE 9.12

Finger flexion controlled by single parameter; the increase in joint angle (degrees) per joint is shown.

then the upper arm/shoulder. Adding motion of the torso including the clavicle and spine, which involves more DOFs, makes this task simpler to accomplish, but also makes the specification of realistic joint angles more complex. It is difficult to determine exactly what rotation should be assigned to which joints at what time in order to realistically model this motion.

Interaction between body parts is a concern beyond the determination of which joints to use in a particular motion. While viewing the arm and hand as a separate independent system simplifies the control strategy, its relation to the rest of the body must be taken into account for a more robust treatment of reaching. Repositioning, twisting, and bending of the torso, reactive motions by the other arm, and even counterbalancing by the legs are often part of movements that only appear to belong to a single arm. It is nearly impossible for a person reaching for an object to keep the rest of the body in a fixed position. Rather than extend joints to the edges of their limits and induce stress, other body parts may cooperate to relieve muscle strain or maintain equilibrium.

Arm manipulation is used in many different full-body movements. Even walking, which is often modeled as an activity of the legs only, involves the torso, the arms, and even the head. The arm often seems like a simple and rewarding place to begin modeling human figure animation, but it is difficult to keep even this task at a simple level.

9.2.5 Reaching around obstacles

To further complicate the specification and control of reaching motion, there may be obstacles in the environment that must be avoided. Of course, it is not enough to merely plan a collision-free path for the end effector. The entire limb sweeps out a volume of space during reach that must be completely devoid of other objects to avoid collisions. For sparse environments, simple reasoning strategies can be used to determine the best way to avoid obstacles.

As more obstacles populate the environment, more complex search strategies might be employed to determine the path. Various path-planning strategies have been proposed. For example, given an environment with obstacles, an artificial potential field can be constructed as a function of the local geometry. Obstacles impart a high potential field that attenuates based on distance. Similarly, the goal position imparts a low potential into the field. The gradient of the field suggests a direction of travel for the end effector and directs the entire linkage away from collisions (Figure 9.13). Such approaches are susceptible to local minima traps that need to be dealt with. Genetic algorithms, for example, have been used to search the space for a global minimum [48]. The genetic fitness function can be tailored to find an optimal path in terms of one of several criteria such as shortest end-effector distance traveled, minimum torque, and minimum angular acceleration.

Such optimizations, however, produce paths that would not necessarily be considered humanlike. Optimized paths will typically come as close as possible to critical objects in the path in order to minimize the fitness function. Humans seldom generate such paths in their reaching motions. The complexity of human motion is further complicated by the effect of vision on obstacle avoidance. If the figure "knows" there is an object to avoid but is not looking directly at it, then the reaching motion will incorporate more leeway in the path than it would if the obstacle were directly in the field of view. Furthermore, the cost of collision can influence the resulting path: it costs more to collide with a barbed-wire fence than a towel, and the path around these obstacles should probably be quite different.

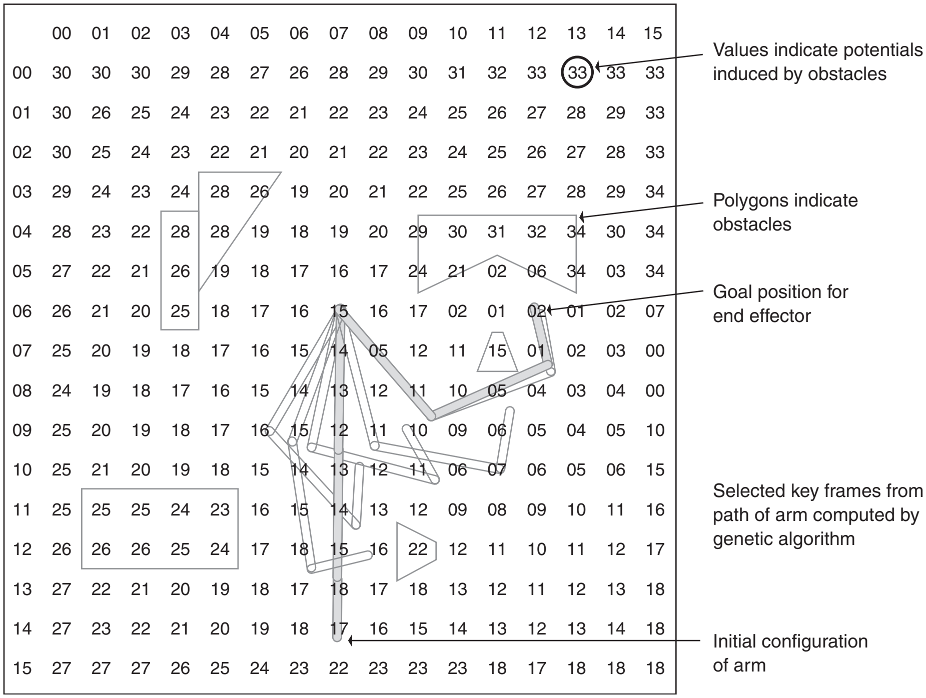

FIGURE 9.13

Path planning result [48].

9.2.6 Strength

As anyone who has ever changed a spark plug in a car knows, avoiding all the obstacles and getting a tool in the correct position is only half the battle. Once in position, the arm and hand must be in a configuration in which there is enough strength available to actually dislodge the plug. To simulate more realistic motions, strength criteria can be incorporated into the task planning [42]. As previously noted, typical reaching motion problems are usually underconstrained, allowing for multiple solutions. The solution space can be searched for a specific motion that is acceptable in terms of the amount of strain it places on the figure.

When a specific motion is proposed by a kinematic planner, it can be evaluated according to the strain it places on the body. The strain is determined by computing the torque necessary at each joint to carry out the motion and rating the torque requirements according to desirability. Given the current pose for the figure, the required joint accelerations, and any external forces, the torque required at each joint can be calculated. For each joint, the maximum possible torque for both flexion and extension is given as a function of the joint's angle as well as that of neighboring joints. A *comfort* metric can be formed as the ratio of currently requested torque and maximum possible torque. The *comfort level* for the figure is computed by finding the maximum torque ratio for the entire body. The most desirable motions are those that minimize the maximum torque ratio over the duration of the motion.

Once a motion has been determined to be unacceptable, it must be modified in order to bring its comfort level back to within acceptable ranges. This can be done by initiating one or more strategies that reduce the strain. Assume that a particular joint has been identified that exceeds the accepted comfort range. If other joints in the linkage can be identified that produce a motion in the end effector similar to that of the problem joint and that have excess torque available, then increasing the torques at these joints can compensate for reduced torque at the problem joint. It may also be possible to include more joints in the linkage, such as the spine in a reaching motion, to reformulate the inverse kinematic problem in the hope of reducing the torque at the problem joint.

9.3 Walking

Walking, along with reaching, is one of the most common activities in which the human form engages. It is a complex activity that, for humans, is learned only after an extended trial-and-error process. An aspect that differentiates walking from typical reaching motions, besides the fact that it uses the legs instead of the arms, is that it is basically cyclic. While its cyclic nature provides some uniformity, acyclic components such as turning and tripping occur periodically. In addition, walking is responsible for transporting the figure from one place to another and is simultaneously responsible for maintaining balance. Thus, dynamics plays a much more integral role in the formation of the walking motion than it does in reaching.

An aspect of walking that complicates its analysis and generation is that it is dynamically stable but not statically stable. This means that if a figure engaged in walking behavior suddenly freezes, the figure is not necessarily in a balanced state and would probably fall to the ground. For animation purposes, this means that the walking motion cannot be frozen in time and statically analyzed to determine the correct forces and torques that produce the motion. As a result, knowledge of the walking motion, in the form of either empirically gathered data [26] [32] or a set of parameters adjustable by the animator, is typically used as the global control mechanism for walking behavior. Attributes such as stride length, hip rotation, and foot placement can be used to specify what a particular walk should look like. A state transition diagram, or its equivalent, is typically used to transition from phase to phase of the gait [11] [12] [22] [29] [52]. Calculation of forces and torques can then be added, if desired, to make the nuances of the motion more physically accurate and more visually satisfying. Kinematics can be used to entirely control the legs, while the forces implied by the movement of the legs are used to affect the motion of the upper body [23] [63]. Alternatively, kinematics can be used to establish constraints on leg motion such as leg swing duration and foot placement. Then the forces and torques necessary to satisfy the constraints can be used to resolve the remaining DOF of the legs [11] [29] [52]. In some cases, forward dynamic control can be used after determining the forces and torques necessary to drive the legs from state to state [47].

9.3.1 The mechanics of locomotion

Understanding the interaction of the various joints involved in locomotion is the first step in understanding and modeling locomotion. The walking and running cycles are presented first. Then the walk cycle is broken down in more detail, showing the complex movements involved. For this discussion, walking is considered to be locomotion characterized by one or both feet touching the ground at any point in time as opposed to running where at most only one foot is on the ground at any point in time.

Walk cycle

The walk cycle can be broken down into various phases [11] based on the relation of the feet to their points of contact with the ground (see Figure 9.14). The *stride* is defined by the sequence of motions between two consecutive repetitions of a body configuration [32]. The *left stance* phase of a stride is initiated with the right foot on the ground and the left heel just starting to strike the ground. During this phase, the body is supported by both feet until the right foot pivots up and the right toe leaves the ground. The left stance phase continues as the right foot leaves the ground and starts swinging forward and as the right heel strikes the ground and both feet are once again on the ground. The left toe leaving the ground terminates the left stance phase. The *right swing phase* is the period in which the right toe leaves the ground, the right leg swings forward, and the right heel strikes the ground. Notice that the right swing phase is a subinterval of the left stance phase. The end of the right swing phase initiates the right stance phase, and analogous phases now proceed with the roles of the left leg and the right leg switched. The walking cycle is characterized by alternating periods of single and double support.

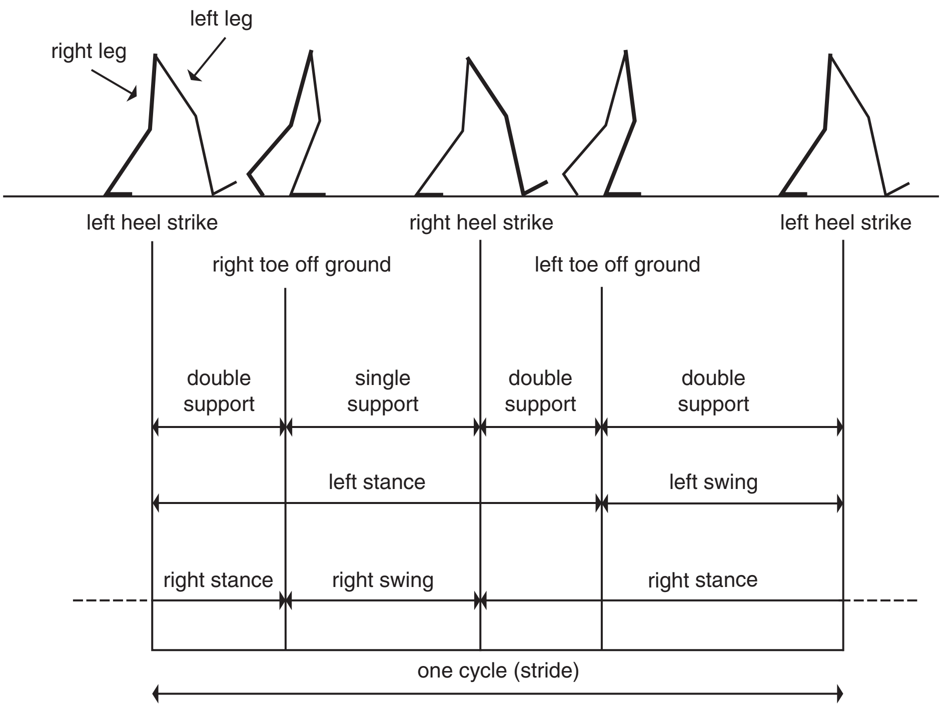

FIGURE 9.14

Walk cycle [32].

Run cycle

The run cycle can also be described as a sequence of phases. It differs from the walk cycle in that both feet are off the ground at one time and at no time are both feet on the ground. As in the walk cycle, the *stance* is the duration that a foot is on the ground. Thus, the *left stance*, defined by the left heel strike and left toe lift, has the right foot off the ground. This is followed by a period of *flight*, during which both feet are off the ground, with the right foot swinging forward. The flight is terminated by the right heel strike, which starts the *right stance* (see Figure 9.15). Notice that the left and right stances do not overlap and are separated by periods of flight.

Pelvic transport

For this discussion, let the pelvis represent the mass of the upper body being transported by the legs. Using a simplified representation for the legs, Figure 9.16 shows how the pelvis is supported by the stance leg at various points during the stance phase of the walk cycle. Figure 9.17 shows these positions superposed during a full stride and illustrates the abutting of two-dimensional circular arcs describing the basic path of the pelvis as it is transported by the legs.

Pelvic rotation

The pelvis represents the connection between the legs and the structure that separates the legs in the third dimension. Figure 9.18 shows the position of the pelvis during various points in the walking cycle, as viewed from above. The pelvis rotates about a vertical axis centered at the stance leg, helping to

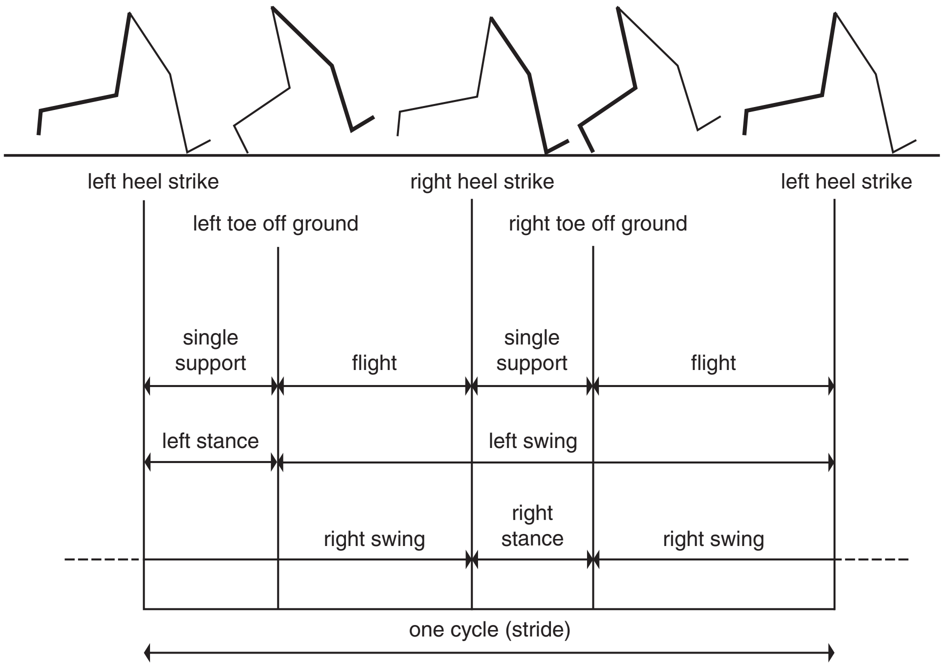

FIGURE 9.15

Run cycle [12].

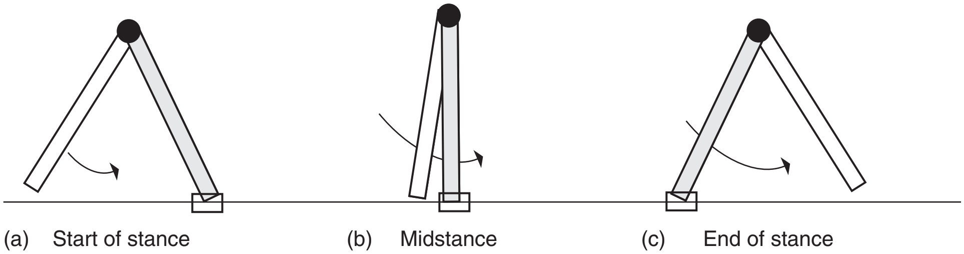

FIGURE 9.16

Position of pelvis during stance phase (sagittal plane). (a) Start of stance, (b) midstance, and (c) end of stance. Box indicates supporting contact with floor.

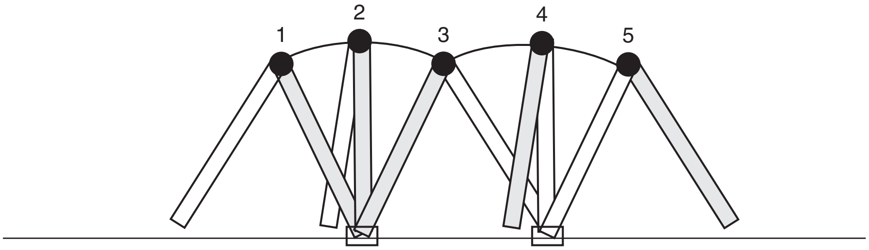

FIGURE 9.17

Transport of pelvis by intersecting circular arcs (sagittal plane).

FIGURE 9.18

Pelvic orientation during stance phase (transverse plane).

lengthen the stride as the swing leg stretches out for its new foot placement. This rotation of the pelvis above the stance leg means that the center of the pelvis follows a circular arc relative to the top of that leg. The top of the stance leg is rotating above the point of contact with the floor (Figure 9.18), so the path of the center of the pelvis resembles a sinusoidal curve (Figure 9.19).

Pelvic list

The transport of the pelvis requires the legs to lift the weight of the body as the pelvis rotates above the point of contact with the floor (Figure 9.20). To reduce the amount of lift, the pelvis lists by rotating in the coronal plane.

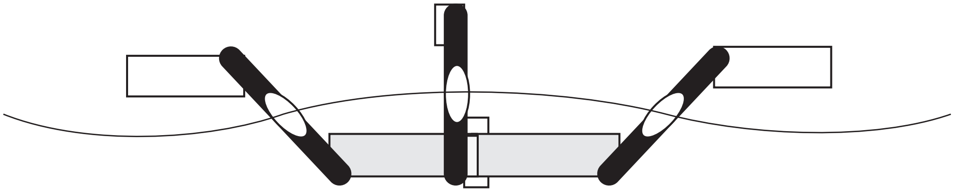

FIGURE 9.19

Path of the pelvic center from above (transverse plane), exaggerated for illustrative purposes.

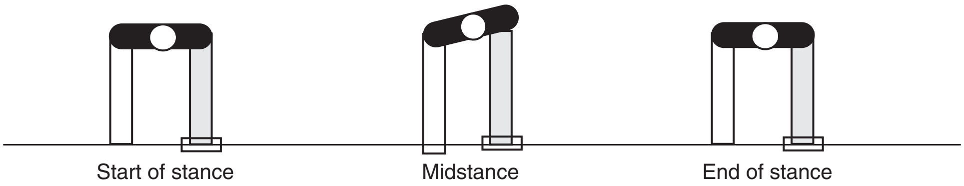

FIGURE 9.20

Pelvic list to reduce the amount of lift (coronal plane).

Knee flexion

As shown in Figure 9.20, in a pelvic list with one-piece legs, the swing leg would penetrate the floor. Bending at the knee joint (flexion) allows the swing leg to safely pass over the floor and avoid contact (Figure 9.21). Flexion at the knee of the stance leg also produces some leveling out of the pelvic arcs produced by the rotation of the pelvis over the point of contact with the floor. In addition, extension just before contact with the floor followed by flexion of the new stance leg at impact provides a degree of shock absorption.

Ankle and toe joints

The final part of the puzzle to the walking motion is the foot complex, consisting of the ankle, the toes, and the foot itself. This complex comprises several bones and many DOFs and can be simply modeled as two hinge joints per foot (Figure 9.22). The ankle and toe joints serve to further flatten out the rotation of the pelvis above the foot as well as to absorb some shock.

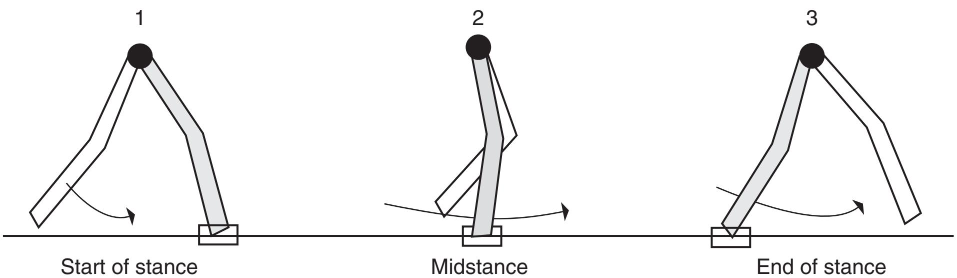

FIGURE 9.21

Knee flexion allowing for the swing leg to avoid penetrating the floor, leveling the path of the pelvis over the point of contact, and providing some shock absorption (sagittal plane).

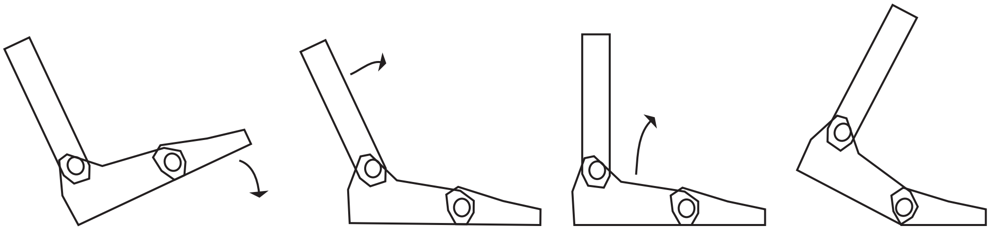

FIGURE 9.22

Rotation due to ankle-toe joints.

9.3.2 The kinematics of the walk

Animation of the leg can be performed by appropriate control of the joint angles. As previously mentioned, a leg's walk cycle is composed of a stance phase and a swing phase. The stance phase duration is the time from heel strike to toe lift. The swing phase duration is the time between contact with the ground—from toe lift to heel strike. The most basic approach to generating the walking motion is for the animator to specify a list of joint angle values for each DOF involved in the walk. There are various sources for empirical data describing the kinematics of various walks at various speeds. Figures 9.23 through 9.27, from Inman, Ralson, and Todd [32], graph the angles over time for the various joints involved in the walk cycle, as well as giving values for the lateral displacement of the pelvis.

Specifying all the joint angles, either on a frame-by-frame basis or by interpolation of values between key frames, is an onerous task for the animator. In addition, it takes a skilled artist to design values that create unique walks that deviate in any way from precisely collected clinical data. When creating new walks, the animator can specify kinematic values such as pelvic movement, foot placement, and foot trajectories. Inverse kinematics can be used to determine the angles of the intermediate joints [12]. By constructing the time-space curves traced by the pelvis and each foot, the user can determine the position of each for a given frame of the animation. Each leg can then be positioned by considering the pelvis fixed and the leg a linked appendage whose desired end-effector position is the corresponding position on the foot trajectory curve (Figure 9.28). Sensitivity to segment lengths can cause even clinical data to produce configurations that fail to keep the feet in solid contact with the floor during walking. Inverse kinematics is also useful for forcing clinical data to maintain proper foot placement.

9.3.3 Using dynamics to help produce realistic motion

Dynamic simulation can be used to map specified actions and constraints to make the movement more accurate physically. However, as Girard and Maciejewski [22] point out, an animator who wants a particular look for a behavior often wants more control over the motion than a total physical simulation provides (Girard and Maciejewski discussed this in relation to walking, but it obviously applies in many situations where physically reasonable, yet artistically controlled, motion is desired). Dynamics must be intelligently applied so that it aids the animator and does not become an obstacle that the animator must work around. In addition, to make the computations tractable, the animator almost always simplifies the dynamics. There are several common types of simplifications: (1) some dynamic effects are ignored, such as the effect of the swing leg on balance; (2) relatively small temporal variations are

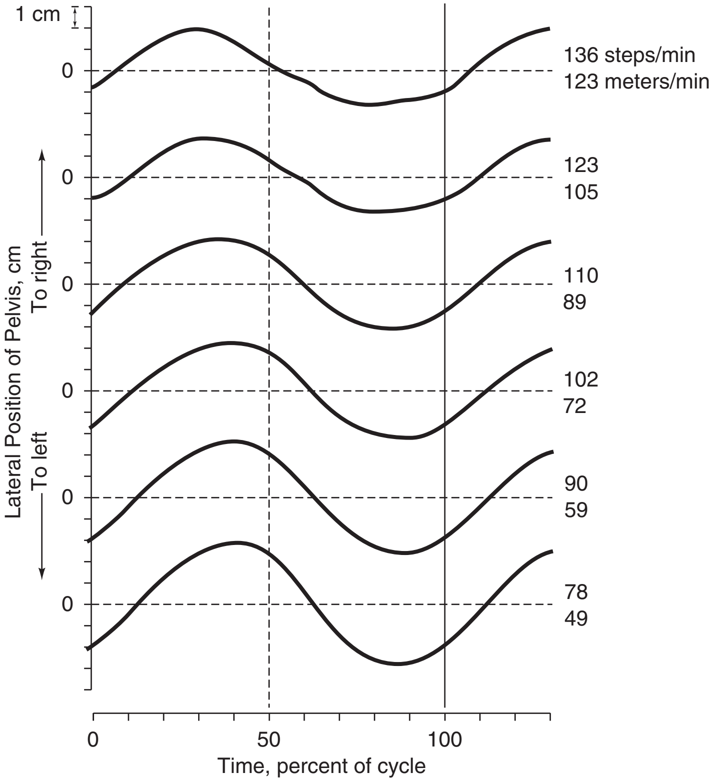

FIGURE 9.23

Lateral displacement of pelvis [32].

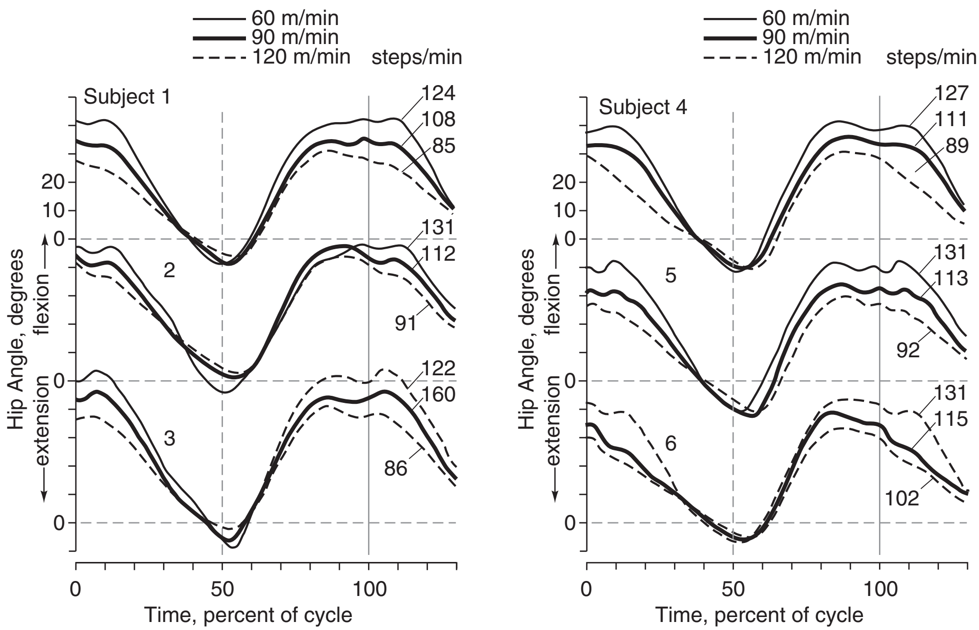

FIGURE 9.24

Hip angles [32].

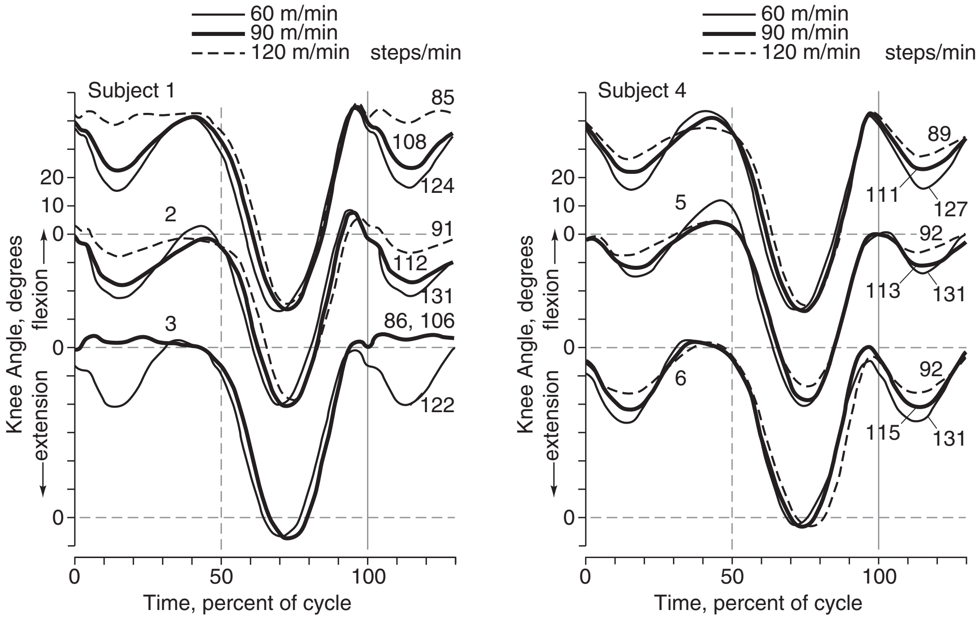

FIGURE 9.25

Knee angles [32].

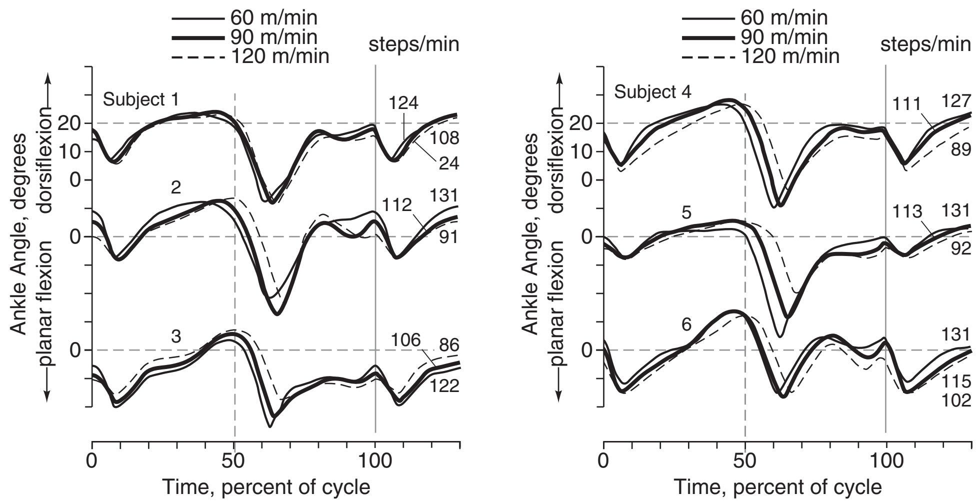

FIGURE 9.26

Ankle angles [32].

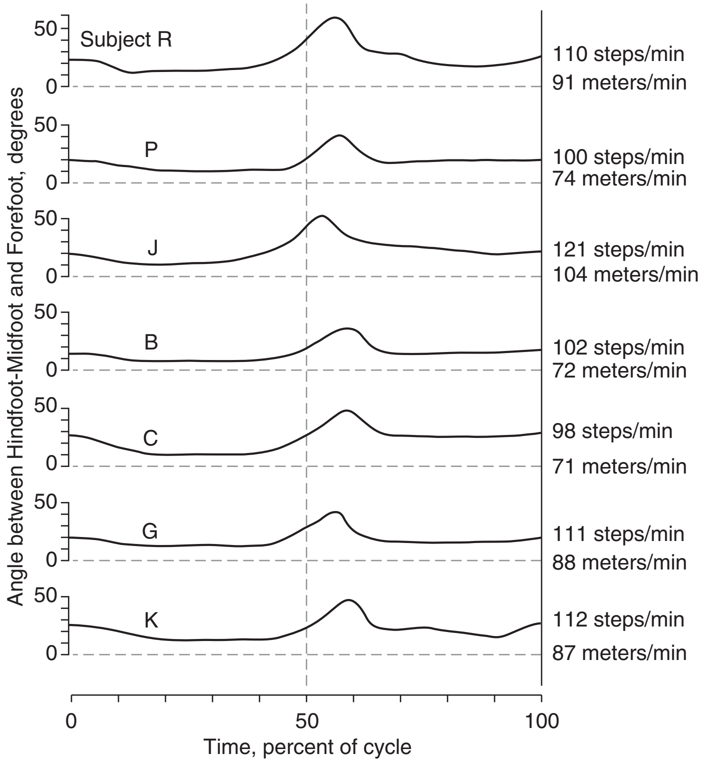

FIGURE 9.27

Toe angles [32].

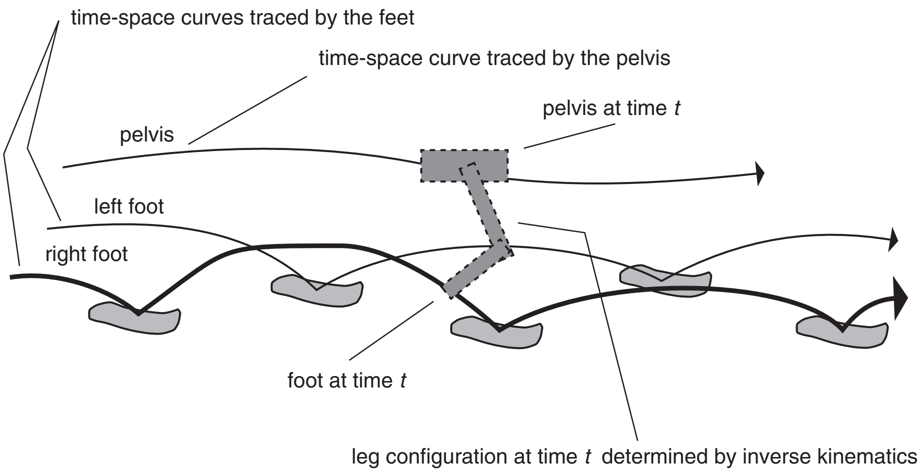

FIGURE 9.28

Pelvis and feet constraints satisfied by inverse kinematics.

ignored and a force is considered constant over some time interval, such as the upward push of the stance leg; (3) a complex structure, such as the seven DOF leg, is replaced for purposes of the dynamic computations by a simplified but somewhat dynamically equivalent structure, such as a one DOF telescoping leg; and (4) computing arbitrarily complex dynamic effects is replaced by computing decoupled dynamic components, such as separate horizontal and vertical components, which are computed independently of each other and then summed.

In achieving the proper motion of the upper body, the legs are used to impart forces to the mass of the upper body as carried by the pelvis. An upward force can be imparted by the stance leg on the mass of the upper body at the point of connection between the legs and the torso, that is, the hips [22] [63]. To achieve support of the upper body, the total upward acceleration due to the support of the legs has to cancel to total downward acceleration due to gravity. As simplifying assumptions, the effect of each leg can be considered independent of the other(s), and the upward force of a leg on the upper body during the leg's stance phase can be considered constant. As additional assumptions, horizontal acceleration can be computed independently for each leg and can be considered constant during the leg's stance phase (Figure 9.29). The horizontal force of the legs can be adjusted automatically to produce the average velocity for the body as specified by the animator, but the fluctuations in instantaneous velocity over time help to create visually more appealing motion than constant velocity alone would. The temporal variations in upward and forward acceleration at each hip due to the alternating stance phases can be used to compute pelvic rotation and pelvic list to produce even more realistic motion.

More physics can be introduced into the lower body by modeling the leg dynamics with a telescoping joint (implemented as a parallel spring-damper system) during the stance phase. The upward force of the leg during the stance phase becomes time varying as the telescoping joint compresses under the weight of the upper body and expands under the restoring forces of the leg complex (upper leg, lower leg, foot, and associated joints). The telescoping mechanism simulates the shock-absorbing effect of the knee-ankle-toe joints. The leg complex is then fit to the length of the telescoping joint by inverse kinematics (Figure 9.30). During the swing phase, the leg is typically controlled kinematically and does not enter into any dynamic considerations.

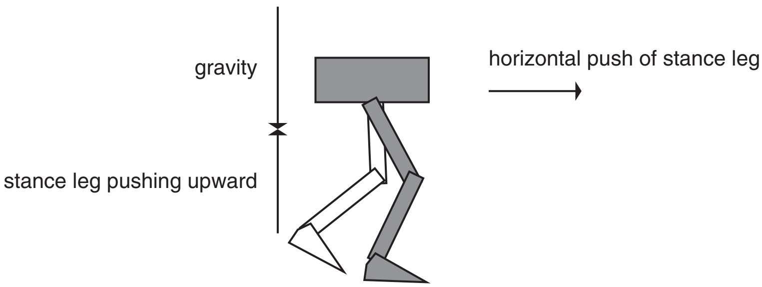

FIGURE 9.29

Horizontal and vertical dynamics of stance leg: gravity and the vertical component must cancel over the duration of the cycle. The horizontal push must account for the average forward motion of the figure over the duration of the cycle.

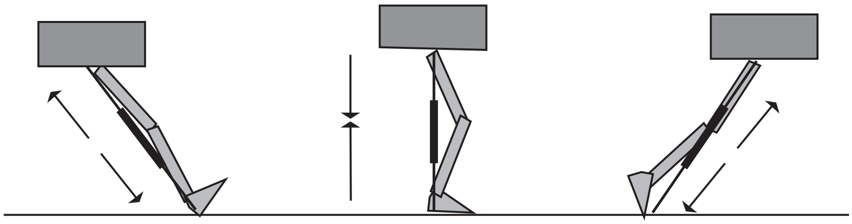

FIGURE 9.30

Telescoping joint with kinematically fit leg complex.

Incorporating more physics into the model, the kinematic control information for the figure can be used to guide the motion (as opposed to constraining it). A simple inverse dynamics computation is then used to try to match the behavior of the system with the kinematic control values. Desired joint angles are computed based on high-level animator-supplied parameters such as speed and the number of steps per unit distance. Joint torques are computed based on proportional-derivative servos (Eq. 9.1). The difference between the desired angle, denoted by the underbar, and the current angle at each joint is used to compute the torque to be applied at the next time step. The angular velocities are treated similarly. These torque values are smoothed to prevent abrupt changes in the computed motion. However, choosing good values for the gains (k_s, k_v) can be difficult and usually requires a trial-and-error approach.

$$\tau = k_s(\underline{\theta}_i - \theta_i) + k_v\left(\dot{\underline{\theta}}_i - \dot{\theta}_i\right) \tag{9.1}$$

9.3.4 Forward dynamic control

In some cases, forward dynamic control instead of kinematic control can be effectively used. Kinematics still plays a role. Certain kinematic states, such as maximum forward extension of a leg, trigger activation of forces and torques. These forces and torques move the legs to a new kinematic state, such as maximum backward extension of a leg, which triggers a different sequence of forces and torques [47]. The difficulty with this approach is in designing the forces and torques necessary to produce a reasonable walk cycle (or other movement, for that matter). In some cases it may be possible to use empirical data found in the biomechanics literature as the appropriate force and torque sequence.

9.3.5 Summary

Implementations of algorithms for procedural animation of walking are widely available in commercial graphics packages. However, none of these could be considered to completely solve the locomotion problem, and many issues remain for ongoing or future research. Walking over uneven terrain and walking around arbitrarily complex obstacles are difficult problems to solve in the most general case. The coordinated movement required for climbing is especially difficult. A recurring theme of this chapter is that developing general, robust computational models of human motion is difficult, to say the least.

9.4 Coverings

9.4.1 Clothing

It is the rare application or animation that calls for totally nude figures. Simulating clothing and the corresponding interaction of the clothing with the surfaces of the figure can be one of the most computationally intensive parts of representing virtual humans. The clothes that protect and decorate the body contribute importantly to the appearance of a human figure. For most human figures in most situations, cloth covers the majority of the body. Cloth provides important visual qualities for a figure and imparts certain attributes and characteristics to it. The way in which cloth drapes over and highlights or hides aspects of the figure can make the figure more or less attractive, more or less threatening, and/or more or less approachable. For a figure in motion, clothes can provide important visual cues that indicate the type and speed of a character's motion. For example, the swirling of a skirt or the bouncing of a shirttail indicates the pace or smoothness of a walk.

The simulation of clothing on a virtual human is a complicated task that has not been fully solved, although significant advances are being made. Real-time application, such as computer games, still often use virtual humans that sport rigid body armor that merely rotates along with whatever limb it is attached to. Some applications can get away with simply texture mapping a pattern onto a single-sheet-defined human figure to simulate tight-fitting spandex clothes. High-end off-line animation systems are starting to offer advanced clothing simulation modules that attempt to calculate the effects of gravity as well as cloth–cloth and cloth–body collisions by using mass-spring networks or energy functions. Streamlined versions of these procedures are finding their way into interactive or even real-time applications.

The animation of cloth has been discussed in Section 7.5, but clothing presents special considerations, especially for real-time situations. Collisions and surfaces in extended contact are an almost constant occurrence for a figure wearing clothes. Calculation of impulse forces that result from such collisions is costly for an animated figure. Whenever a vertex is identified that violates a face in the environment, some procedure, such as a transient spring force or enforcing positional constraints, must be invoked so that the geometry is restored to an acceptable position.

To simulate clothing, the user must incorporate the dynamic aspect of cloth into the model to produce the wrinkling and bulging that naturally occur when cloth is worn by an articulated figure. In order for the figure to affect the shape of the cloth, extensive collision detection and response must be calculated as the cloth collides with the figure and with itself almost constantly. The level of detail at which clothes must be modeled to create realistic wrinkles and bulges requires relatively small triangles. Therefore, it takes a large number of geometric elements to clothe a human figure. As a result, one must attend to the efficiency of the methods used to implement the dynamic simulation and collision handling of the clothing.

9.4.4 Hair

One of the most significant hurdles for making virtual humans that are indistinguishable from a real person is the accurate simulation of a full head of hair. Hair is extremely complex. A head of hair consists of approximately 100,000 strands, each strand a flexible cylinder [45]. Because the strands are attached to the scalp and are in close proximity to each other, the motion of a single strand is constantly affected by external forces. Collisions, friction, and even influences such as static electricity can be significant. Surface properties of the hair strands, such as wetness, cleanliness, curliness, oiliness, split ends, and cosmetic applications, affect its motion. In addition, hair can be classified into one of three main types based on the geometry of a strand: Asian, African, and Caucasian. Asian generally has smooth and regular strands with a circular cross section while African strands are irregular and rough with elliptical cross sections. The qualities of Caucasian hair are between these two extremes.

Although the hair's motion is of primary interest for the present discussion, it should also be noted that providing the user with hair style design tools and the complexities of the rendering of hair are also difficult problems. All of these are the subject of ongoing research.

Depending on the requirements of the animation, the modeling and animation of hair can take place on one or more levels of detail. The most common, visually poor but computationally inexpensive technique has been to merely construct a rigid piece of geometry in the rough shape of the volume of hair and attach it to the top of the head, like a helmet, as in Figure 9.31. Texture maps with transparency information are sometimes used to improve the appearance, providing for a wispy boundary. FFDs can be used to animate the global shape of rigidly defined hair.

To add more realism to the hair, a hairstyle feature, such as a ponytail or lock of hair, can be animated independent of the rest of the hair [59]. This provides some dynamics to the hair at minimal computational cost. The animation of cartoon hair has even received some attention in the literature, e.g., [62].

For more realistic hair, the strands, or at least groups of strands, must be able to move relative to the rest of the hair [37]. Clusters of hair strands can be animated as a unit and the hair modeled as a collection of clusters. Sheets of hair strands can be used (Figure 9.32, Color Plate 5) or a generalized cylindrical collection of strands can be modeled, depending on the type of hair modeled. The strand clusters can be animated as flexible sheets and held close together by connecting springs. Alternatively, a few select master strands can be individually animated and then intermediate strands can be interpolated in order to fill out the hair.

For the most realism, but the greatest computational cost, individual strands can be modeled and animated separately (e.g., [55]). Strands are generated using small geometric tubes [3] [47] or particle trails. Individual strands can be animated using the mass-spring-damper system or using the dynamics of rigid body linkages. Often, strand–strand interaction is ignored in order to keep the computational cost within reason.

FIGURE 9.31

Hair modeled as rigid geometry. (Image courtesy of Mike Altman.)

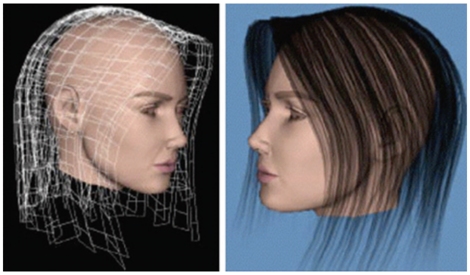

FIGURE 9.32

Hair modeled as strips of strands [37].

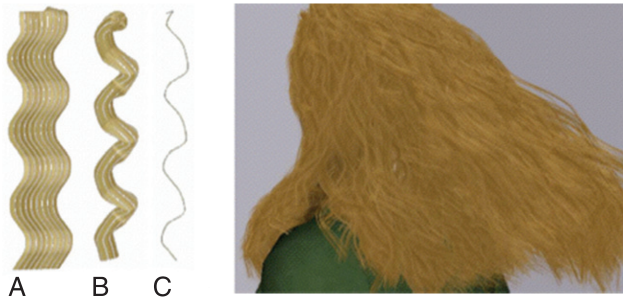

FIGURE 9.33

Hair modeled using multiple levels of detail [65].

To provide both the computational savings of strand clusters as well as the realism of animating individual strands, a multiple level of detail approach can be used (Figure 9.33, Color Plate 6) [65] and, to provide more flexibility, adaptive grouping can be used [64]. Other approaches include modeling hair as a thin shell [34] and as a continuum [24].

9.5 Chapter summary

The human figure is an interesting and complex form. It has a uniform structure but contains infinite variety. As an articulated rigid structure, it contains many DOFs, but its surface is deformable. Modeling and animating hair in any detail is also enormously complex. Moreover, the constant collision and sliding of cloth on the surface of the body represents significant computational complexity.

One of the things that makes human motion so challenging is that no single motion can be identified as correct human motion. Human motion varies from individual to individual for a number of reasons, but it is still recognizable as reasonable human motion, and slight variations can seem odd or awkward. Research in computer animation is just starting to delve into the nuances of human motion. Many of these nuances vary from person to person but, at the same time, are very consistent for a particular person, for a particular ethnic group, for a particular age group, for a particular weight group, for a particular emotional state, and so on. Computer animation is only beginning to analyze, record, and synthesize these important qualities of movement. Most of the work has focused on modeling changes in motion resulting from an individual's emotional state (e.g., [2] [15]).

Human figure animation remains a challenge for computer animators and fertile ground for graphics researchers. Developing a synthetic actor indistinguishable from a human counterpart remains the Holy Grail of researchers in computer animation.

References

[1] Alias/Wavefront. Maya, http://www.aliaswavefront.com**; 1998.**

[2] Amaya K, Bruderlin A, Calvert T. Emotion from Motion. In: Davis WA, Bartels R, editors. Graphics Interface '96. Canadian Human-Computer Communications Society; May 1996. p. 222–9. ISBN 0-9695338-5-3.

[3] Anjyo K, Usami Y, Kurihara T. A Simple Method for Extracting the Natural Beauty of Hair. In: Dill J, editor. Computer Graphics. Proceedings of SIGGRAPH 88, vol. 22(4). Atlanta, Ga.; August 1988. p. 111–20.

[4] Avid Technology, Inc. SOFTIMAGE|3D, http://www.softimage.com/.

[5] Aydin Y, Kakajiima M. Database Guided Computer Animation of Human Grasping Using Forward and Inverse Kinematics. Computers and Graphics 1999;23:145–54.

[6] Azulola F, Badler N, Hoon TK, Wei S. Sass v.2.1 Anthropometric Spreadsheet and Database for the Iris. Technical Report, Dept. of Computer and Information Science, University of Pennsylvania; 1993. MS-CIS-93-63.

[7] Baca K. Poor Man's Skinning. Game Developer, July 1998;48–51.

[8] Bekey GA, Liu H, Tomovic R, Karplus WJ. Knowledge-Based Control of Grasping in Robot Hands Using Heuristics from Human Motor Skills. IEEE Transactions on Robotics and Automation 1993;9 (6):709–22.

[9] Blanz V, Vetter T. A Morphable Model for the Synthesis of 3D Faces, In: Rockwood A, editor. Computer Graphics. Proceedings of SIGGRAPH 99, Annual Conference Series. Los Angeles, Calif,: Addison-Wesley Longman; August 1999. p. 187–94. ISBN 0-20148-560-5.

[10] Boulic R, Capin T, Huang Z, Kalra P, Linterrnann B, Magnenat-Thalmann N, et al. The HUMANOID Environment for Interactive Animation of Multiple Deformable Human Characters. In: Post F, Göbel M, editors. Computer Graphics Forum, vol. 14(3). Blackwell Publishers; August 1995. p. 337–48. ISSN 1067-7055.

[11] Bruderlin A, Calvert T. Goal-Directed, Dynamic Animation of Human Walking. In: Lane J, editor. Computer Graphics. Proceedings of SIGGRAPH 89, vol. 23(3). Boston, Mass.; July 1989. p. 233–42.

[12] Bruderlin A, Calvert T. Knowledge-Driven, Interactive Animation of Human Running. In: Davis WA, Bartels R, editors. Graphics Interface '96. Canadian Human-Computer Communications Society; May 1996. p. 213–21. ISBN 0-9695338-5-3.

[13] Chadwick J, Haumann D, Parent R. Layered Construction for Deformable Animated Characters. In: Lane J, editor. Computer Graphics. Proceedings of SIGGRAPH 89, vol. 23(3). Boston, Mass.; July 1989. p. 243–52.

[14] Chen D, Zeltzer D. Pump It Up: Computer Animation of a Biomechanically Based Model of Muscle Using the Finite Element Method. In: Catmull EE, editor. Computer Graphics. Computer Graphics. Proceedings of SIGGRAPH 92, vol. 26(2). Chicago, Ill.; July 1992. p. 89–98. ISBN 0-201-51585-7.

[15] Chi D, Costa M, Zhao L, Badler N. The EMOTE Model for Effort and Shape, In: Akeley K, editor. Computer Graphics. Proceedings of SIGGRAPH 2000, Annual Conference Series. ACM Press/ACM SIGGRAPH/ Addison-Wesley Longman; July 2000. p. 173–82. 1-58113-208-5.

[16] COVEN, http://coven.lancs.ac.uk/mpeg4/index.html, January 2.

[17] DeRose T, Kass M, Truong T. Subdivision Surfaces for Character Animation, In: Cohen M, editor. Computer Graphics. Proceedings of SIGGRAPH 98, Annual Conference Series. Orlando, Fla.: Addison-Wesley; July 1998. p. 85–94 ISBN 0-89791-999-8.

[18] ElKoura G, Singh K. Handrix: Animating the Human Hand, In: Symposium on Computer Animation; 2003.

[19] Elson M. Displacement Animation: Development and Application. In: Course #10: Character Animation by Computer, SIGGRAPH 1990. Dallas, Tex.; August 1990.

[20] Engin A, Peindl R. On the Biomechanics of Human Shoulder Complex—I: Kinematics for Determination of the Shoulder Complex Sinus. J Biomech 1987;20(2):103–17.

[21] Forsey D, Bartels R. Hierarchical B-Spline Refinement. In: Dill J, editor. Computer Graphics. Proceedings of SIGGRAPH 88, vol. 22(4). Atlanta, Ga.; August 1988. p. 205–12.

[22] Girard M, Maciejewski A. Computational Modeling for the Computer Animation of Legged Figures. In: Barsky BA, editor. Computer Graphics. Proceedings of SIGGRAPH 85, vol. 19(3). San Francisco, Calif.; August 1985. p. 263–70.

[23] Gourret J-P, Magnenat-Thalmann N, Thalmann D. Simulation of Object and Human Skin Deformations in a Grasping Task. In: Lane J, editor. Computer Graphics. Proceedings of SIGGRAPH 89, vol. 23(3). Boston, Mass.; July 1989. p. 21–30.

[24] Hadap S, Magnenat-Thalmann N. Modeling Dynamic Hair as a Continuum. Computer Graphics Forum 2001;20(3):329–38.

[25] Hanrahan P, Wolfgang K. Reflection from Layered Surfaces due to Subsurface Scattering. In: Kajiya JT, editor. Computer Graphics. Proceedings of SIGGRAPH 93, Annual Conference Series. Anaheim, Calif.; August 1993. p. 165–74. ISBN 0-201-58889-7.

[26] Harris G, Smith P, editors. Human Motion Analysis. Piscataway, New Jersey: IEEE Press; 1996.

[27] Henry-Biskup S. Anatomically Correct Character Modeling. Gamasutra November 13 1998;2(45). http://www.gamasutra.com/features/visual_arts/19981113/charmod_01.htm.

[28] Hilton A, Beresford D, Gentils T, Smith R, Sun W. Virtual People: Capturing Human Models to Populate Virtual Worlds, http://www.ee.surrey.ac.uk/Research/VSSP/3DVision/VirtualPeople/.

[29] Hodgins J, Wooten W, Brogan D, O'Brien J. Animating Human Athletics, In: Cook R, editor. Computer Graphics. Proceedings of SIGGRAPH 95, Annual Conference Series. Los Angeles, Calif: Addison-Wesley; August 1995. p. 71–8. ISBN 0-201-84776-0.

[30] Huang Z, Boulic R, Magnenat-Thalmann N, Thalmann D. A Multi-Sensor Approach Fograsping and 3D Interaction. Proc Computer Graphics International 1995.

[31] Humanoid Animation Working Group . Specification for a Standard Humanoid, http://ece.uwaterloo.ca/~h-anim/spec1.1/; **1999.**

[32] Inman V, Ralston H, Todd F. Human Walking. Baltimore, Maryland: Williams & Wilkins; 1981.

[33] Kakadiaris I, Metaxas D. 3D Human Body Model Acquisition from Multiple Views. In: Proceedings of the Fifth International Conference on Computer Vision. Boston, Mass.; June 20–23, 1995.

[34] Kim T-Y, Neumann U. A Thin Shell Volume for Modeling Human Hair. In: Computer Animation- 2000. IEEE Computer Society; 2000. p. 121–8.

[35] Kinetix. 3D Studio MAX, http://www.ktx.com**; February 2007.**

[36] Koga Y, Kondo K, Kuffner J, Latombe J-C. Planning Motions with Intentions, In: Glassner A, editor. Computer Graphics. Proceedings of SIGGRAPH 94, Annual Conference Series. Orlando, Fla.: ACM Press; July 1994. p. 395–408. ISBN 0-89791-667-0.

[37] Koh C, Huang Z. Real-Time Animation of Human Hair Modeled in Strips. In: Computer Animation and Simulation '00. Sept. 2000. p. 101–12.

[38] Kondo K. Inverse Kinematics of a Human Arm. Technical Report STAN-CS-TR-94-1508, Stanford University; 1994.

[39] Kry P, Pai D. Interaction Capture and Synthesis. Transactions on Graphics 2006;25(3).

[40] Lacquaniti F, Soechting JF. Coordination of Arm and Wrist Motion during a Reaching Task. Journal of Neuroscience 1982;2(2):399–408.

[41] Lander J. On Creating Cool Real-Time 3D, Gamasutra October 17, 1997;1(8). http://www.gamasutra.com/features/visual_arts/101797/rt3d_01.htm.

[42] Lee P, Wei S, Zhao J, Badler N. Strength Guided Motion. In: Baskett F, editor. Computer Graphics. Proceedings of SIGGRAPH 90, vol. 24(4). Dallas, Tex.; August 1990. p. 253–62. ISBN 0-201-50933-4.
[43] Lee W-S, Magnenat-Thalmann N. From Real Faces to Virtual Faces: Problems and Solutions. In: Proc. 3IA'98. Limoges (France); 1998.
[44] Lewis JP, Cordner M, Fong N. Pose Space Deformations: A Unified Approach to Shape Interpolation and Skeleton-Driven Deformation, In: Akeley K, editor. Computer Graphics. Proceedings of SIGGRAPH 2000, Annual Conference Series. ACM Press/ACM SIGGRAPH/Addison-Wesley Longman; July 2000. p. 165–72. ISBN 1-58113-208-5.
[45] Magnenat-Thalmann N, Hadap S. State of the Art in Hair Simulation. In: International Workshop on Human Modeling and Animation. Korea Computer Graphics Society; June 2000. p. 3–9.
[46] Mas Sanso R, Thalmann D. A Hand Control and Automatic Grasping System for Synthetic Actors. In: Eurographics '94. 1994.
[47] McKenna M, Zeltzer D. Dynamic Simulation of Autonomous Legged Locomotion. In: Baskett F, editor. Computer Graphics. Proceedings of SIGGRAPH 90, vol. 24(4). Dallas, Tex.; August 1990. p. 29–38 ISBN: 0-201-50933-4.
[48] Miller D. The Generation of Human-Like Reaching Motion for an Arm in an Obstacle-Filled 3-D Static Environment. Ph.D. dissertation, Ohio State University; 1993.
[49] Nedel L, Thalmann D. Modeling and Deformation of the Human Body Using an Anatomically-Based Approach. In: Computer Animation '98. Philadelphia, Pa: IEEE Computer Society; June 1998.
[50] Pandzic I, Capin T, Magnenat-Thalmann N, Thalmann D. Developing Simulation Techniques for an Interactive Clothing System. In: Proceedings of VSMM '97. Geneva, Switzerland; 1997. p. 109–18.
[51] Pollard NS, Zordan VB. Physically Based Grasping Control from Example. In: Symposium on Computer Animation; 2005.
[52] Raibert M, Hodgins J. Animation of Dynamic Legged Locomotion. In: Sederberg TW, editor. Computer Graphics. Proceedings of SIGGRAPH 91, vol. 25(4). Las Vegas, Nev.; July 1991. p. 349–58. ISBN 0-201-56291-X.
[53] REM Infografica. MetaReyes and ClothReyes, http://www.infografica.com/.
[54] Rijpkema H, Girard M. Computer Animation of Knowledge-Based Human Grasping. In: Sederberg TW, editor. Computer Graphics, Proceedings of SIGGRAPH 91, vol. 25(4). Las Vegas, Nev.; July 1991. p. 339–48. ISBN 0-201-56291-X.
[55] Rosenblum R, Carlson W, Tripp E. Simulating the Structure and Dynamics of Human Hair: Modeling, Rendering, and Animation. The Journal of Visualization and Computer Animation 1991;2(4):141–8.
[56] Sannier G, Magnenat-Thalmann N. A User-Friendly Texture-Fitting Methodology for Virtual Humans. In: Computer Graphics International 1997. Hasselt/Diepenbeek, Belgium: IEEE Computer Society; June 1997.
[57] Scheepers F. Anatomy-Based Surface Generation for Articulated Models of Human Figures. Ph.D. dissertation, Ohio State University; 1996.
[58] Scheepers F, Parent R, Carlson W, May S. Anatomy-Based Modeling of the Human Musculature. In: Whitted T, editor. Computer Graphics. Proceedings of SIGGRAPH 97, Annual Conference Series. p. 163–72. ISBN 0-89791-896-7.
[59] Sega. Virtual Fighter 3, http://www.sega.com/games/games_vf3.html.
[60] Singh K. Realistic Human Figure Synthesis and Animation for VR Applications. Ph.D. dissertation, Ohio State University; 1995.
[61] Soechting J, Flanders M. Errors in Pointing Are Due to Approximations in Sensorimotor Transformations. J Neurophysiol August 1989;62(2):595–608.
[62] Sugisaki E, Yu Y, Anjyo K, Morishima S. Simulation-Based Cartoon Hair Animation, In: 13th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG). Plzen, Czech Republic; January, 2005.
[63] Torkos N, van de Panne M. Footprint-Based Quadruped Motion Synthesis. In: Booth K, Fournier A, editors. Graphics Interface '98. June 1998. p. 151–60. ISBN 0-9695338-6-1.

[64] Ward K, Lin M. Adaptive Grouping and Subdivision for Simulating Hair Dynamics. In: Pacific Graphics Conference on Computer Graphics and Applications. October 2003. p. 234–43.

[65] Ward K, Lin M, Lee J, Fisher S, Macri D. Modeling Hair Using Level-of-Detail Representations. In: International Conference on Computer Animation and Social Agents. May 2003. p. 41–7.

[66] Wilhelms J, van Gelder A. Anatomically Based Modeling, In: Whitted T, editor. Computer Graphics. Proceedings of SIGGRAPH 97, Annual Conference SeriesLos Angeles, Calif.: Addison-Wesley; August 1997. p. 173–80. ISBN 0-89791-896-7.