Computer Animation

Jessica K. Hodgins and James F. O'Brien
College of Computing and Graphics, Visualization, and Usability Center
Georgia Institute of Technology
Atlanta, GA 30332-0280
[jkh|obrienj]@cc.gatech.edu

Animated images are almost magical in their ability to capture our imagination. By telling a compelling story, astounding with special effects, or mesmerizing with abstract motion, animation can infuse a sequence of inert images with the illusion of motion and life. Creating this illusion, either by hand or with the assistance of computer software, is not easy. Each individual image, or *frame*, in the animated sequence must blend seamlessly with the other images to create smooth and continuous motion that flows through time.

Traditionally, animation has been created by drawing images at certain key points in the action. These images, known as *keyframes*, outline the motion for the sequence. Later, the images between the keyframes are filled in to complete the sequence, in a process called *in-betweening*. For example, to keyframe hitting a baseball, the animator would draw several key moments in the sequence such as the impact of the bat on the ball and the follow through of the swing. The remaining images would be filled in later, perhaps by a different animator.

The most basic computer animation tools assist the process of hand animation by automatically interpolating between the keyframes of images or models. Animation tools have also been developed to composite together multiple layers of an animated scene in much the same way that layers of *cels* are used in traditional animation. Other more powerful techniques make use of algorithms that render the images from geometric descriptions of the scene. These techniques change the task from drawing sequences of images by hand to using computer tools to effectively specify how images should change over time.

A wide variety of techniques are used in the process of creating a complex computer animation such as Disney and Pixar's *Toy Story* (*www.toystory.com*). These techniques can be grouped into two main classes: two dimensional (2D) and three dimensional (3D). Although there is some overlap between the two classes, 2D techniques tend to focus on image manipulation while 3D techniques usually build virtual worlds in which characters and objects move and interact.

To appear in the *Encyclopedia of Computer Science*, from International Thomson Computer Press, Fourth Edition.

1 Two-dimensional Animation

The impact of 2D techniques can be as spectacular as the addition of the film character ET to a shot of the moon or as subtle as the removal of the guide wires used to suspend Superman. These techniques contribute a great deal to computer animation by providing the tools used for blending or morphing between images, embedding graphical objects in video footage, or creating abstract patterns from mathematical equations.

Morphing refers to animations where an image or model of one object is metamorphosed into another. In Michael Jackson's music video *Black or White*, the animators at Pacific Data Images (*www.pdi.com*) created morphs between people with strikingly different facial characteristics. This work is remarkable because the faces in the intermediate images appear natural and humanlike. Unfortunately, morphing is labor intensive because the key elements of each image must be specified by hand.

Embedding graphical objects into an existing image allows new elements to be added to a scene. For example, the ghosts in *Casper* and many of the dinosaurs in *Jurassic Park* were computer generated and then composited into existing footage (*www.lost-world.com*). Objects can also be removed from a scene. The bus in *Speed* flies over a gap in a partially constructed bridge The gap was created by digitally removing a span from footage of an intact bridge. Both the processes of embedding and of removing objects are made more difficult if the camera is moving because the alteration must be consistent with the changing viewpoint.

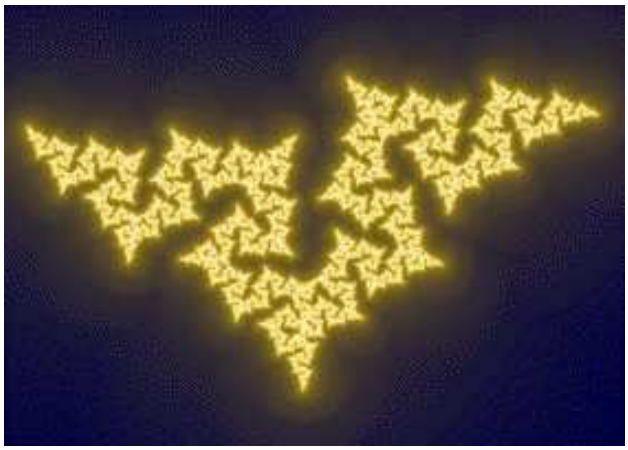

Figure 1: **Image created from fractal equations.** The abstract image in this figure was created by using a type of complex mathematical equation known as a fractal. Image from the *The 1996 Fractals Calendar* courtesy of Daryl H. Hepting.

Mathematical equations are often used to create abstract motion sequences. When the values of the mathematical functions are mapped to color values and varied with time, the motion of the underlying structures can be quite beautiful. Fractals, such as the one shown in Figure 1, are a well-known example of functions that create attractive patterns.

Morphing and the generation of abstract images from mathematical equations can be generalized for use in 3D. All of these 2D techniques can be used either on their own to create an animation or as a post-processing step to enhance images generated using other 2D or 3D techniques.

2 Three-dimensional Animation

When using 3D techniques, the animator constructs a virtual world in which characters and objects move and interact. Using a virtual 3D world to generate images for an animation involves three steps: modeling, animating, and rendering. Briefly stated, modeling involves setting up the elements in a scene and describing each of those elements. Rendering converts the description of the objects and their motion into images. Techniques for modeling and rendering are, for the most part, independent of their role in the animation process. Some modifications, however, must be made to the modeling or rendering procedures if they are to be used for animation, and those are described below.

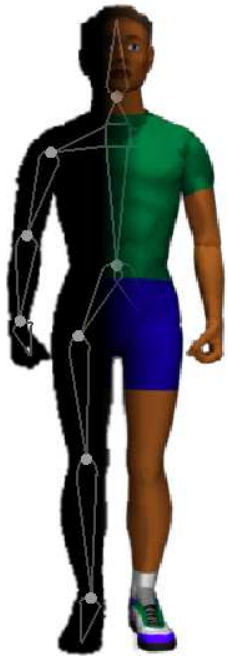

Figure 2: **An articulated model of a human male:** The structure of the joint hierarchy is shown on the left. The graphical model used for rendering is shown on the right.

2.1 Modeling Requirements

To animate motion, the user needs both a static description of an object and information about how that object moves. One common way to specify this additional information is to use an *articulated model* such as the one shown in Figure 2. An articulated model is a collection of objects connected together by joints in a hierarchical, tree-like structure. The location of an object is determined by the location of the objects above it in the hierarchy. For example, the motion of the elbow joint in a human model will affect not only the position of the lower arm but also the position of the hand and fingers. The object at the top of the hierarchy, or the root of the tree, can be moved arbitrarily, affecting the position and orientation of the entire model.

A second type of model used in animation is a *particle system* or collection of points. The motion of the particles through space is determined by a set of rules. The laws of physics often provide a basis for the motion so that the particles fall under gravity and collide with other objects in the environment. Systems that are well modeled by particle systems include water spray, smoke, and even flocks of birds.

Deformable objects are a third type of model and in-

Figure 3: **A diver entering pool:** These images show the combined use of an articulated model, a deformable model, and a particle system. Images courtesy of the Graphics, Visualization and Usability Center, Georgia Institute of Technology.

clude objects that do not have well-defined articulated joints but nevertheless have too much structure to be easily represented with a particle system. Because of the broad nature of this class, there are several fundamentally different ways to represent deformable objects, including spring-mass lattices, volumetric models, and surface representations. Water, hair, clothing, and fish are among the systems that have been successfully modeled as deformable objects.

While each of these model types can be used to describe a wide variety of objects, complex systems often require *hybrid models* that combine two or more types. This approach allows each part of the system to be modeled by the most appropriate technique. The image in Figure 3 shows a diver entering a swimming pool. The diver is an articulated model, the pool is a deformable model, and the spray is a particle system.

2.2 Rendering Requirements

Motion blur is a rendering technique that is required for animation but not for most still images. Without motion blur, rapid motion of an object in a series of frames creates unpleasant strobing effects, causing objects, such as wheels, to appear to move in the wrong direction. To solve this problem, a fast moving object can be rendered in several of the positions it had during the period of time represented by each frame. This technique creates a blurred representation of the object and mimics the exposure of film by a camera shutter that is open for short period of time. Motion blur is actually a form of *anti-aliasing* that occurs over time instead of space and is often called *temporal anti-aliasing*.

3 Motion Generation

The task of specifying the motion of an animated object to the computer is surprisingly difficult and even animating a simple object like a bouncing ball can present problems. In part, this task is difficult because humans are very skilled at observing motion and quickly detect motion that is unnatural or implausible. In many situations, the animator must be able to specify subtle details of the motion in order to convey the personality of a character or the mood of an animation in a compelling fashion.

A number of techniques have been developed for specifying motion, but all the available tools require a tradeoff between automation and control. *Keyframing* allows fine control but does little to automatically insure the naturalness of the result. *Procedural methods* and *motion capture* generate motion in a fairly automatic fashion but offer little control over fine details.

3.1 Keyframing

Borrowing its name from the traditional hand animation technique, keyframing requires the animator to specify key positions for the objects being animated. The computer then interpolates to determine the positions for the in-between frames. The interpolation algorithm is an important factor in the appearance of the final motion. The simplest form of interpolation, *linear interpolation*, often results in motion that appears jerky because the velocities of the moving objects are discontinuous. To correct this problem, better interpolation techniques, such as *splines*, are used to produce smoothly interpolated curves.

The specification of keyframes can be made easier with techniques such as *inverse kinematics*, which aid in the placement of articulated models by allowing the animator to specify the position of one object and have the positions of the objects above it in the articulated hierarchy computed automatically. For example, if the hand and torso of an animated character must be in particular locations, inverse kinematics allows the computer to calculate the angle at the elbow and shoulder. Commercial animation packages include inverse kinematics and interpolation routines designed

Figure 4: **Runner in a park:** All the objects in this image were animated using dynamic simulation. The runner and the child on the swing are active simulations governed by control systems. The clothing is a passive system that has been coupled to an active system. Image courtesy of the Graphics, Visualization and Usability Center, Georgia Institute of Technology.

specifically for animating human figures. These tools take into consideration such factors as maintaining balance, joint angle limitations, and collisions between the limbs and the body.

Although these techniques make animation easier, keyframed animation nevertheless requires that the animator understand in intimate detail how the animated object should behave and have the talent to express that behavior in keyframes.

3.2 Procedural Methods

Current technology is not capable of generating motion automatically for arbitrary objects; however, algorithms for specific types of motion can be built. These techniques are called *procedural methods* because a computer procedurally follows the steps in an algorithm to generate the motion. Procedural methods have two main advantages over keyframing techniques: they make it easy to generate a family of similar motions, and they can be used for systems that would be too complex to animate by hand, such as particle systems or flexible surfaces.

Physically based simulation refers to a class of procedural methods that makes use of the laws of physics, or an approximation to those laws, to generate motion. Simulated motion is inherently realistic and for many applications that is an advantage. Unfortunately, building a new simulation is sometimes a difficult process requiring an in-depth understanding of the relevant physical laws. Once a simulation has been designed, however, an animator may use it without knowing how the internals of the simulation function.

Simulations can be divided into two categories: *passive* and *active*. Passive systems have no internal energy source and move only when an external force acts on them. Passive systems are well suited to physically based simulation because once the physical laws have been encoded correctly and the initial conditions of the animation have been specified, the method is ready for use. Pools of water, clothing, hair, and leaves have been animated using passive simulations.

Active systems do have an internal source of energy and can move of their own volition. People, animals, and robots are examples of active systems. These systems are more difficult to model because in addition to implementing the physical laws, the behavior of the simulated muscles or motors must be specified. An additional algorithm, a *control system*, must be designed to allow the model to walk, run, or perform other actions. For example, a control system for standing must contain laws that specify how the hips and knees should move to keep the figure balanced when one arm is extended out to the side. Control systems can be designed by hand for figures with the complexity of a 3D model of a human figure. For slightly simpler systems, they can be designed automatically using optimization techniques. Although the design of the control system is not a fully understood problem, after a particular control system has been built, an animator can use it without understanding the internal details. Figure 4 shows a human figure running. The runner is an active simulation and a control system generates the running motion. The runner's clothes are a passive cloth simulation.

Procedural methods can also be used to generate motion for groups of objects that move together. Flocks of birds, schools of fish, herds of animals, or crowds of people are all situations where algorithms for *group behaviors* can be used. In Walt Disney's animated version of *The Hunchback of Notre Dame*, most of the crowd scenes were computer animated using procedural models. This animated film is particularly impressive because computer and hand animation are seamlessly combined to create very detailed scenes.

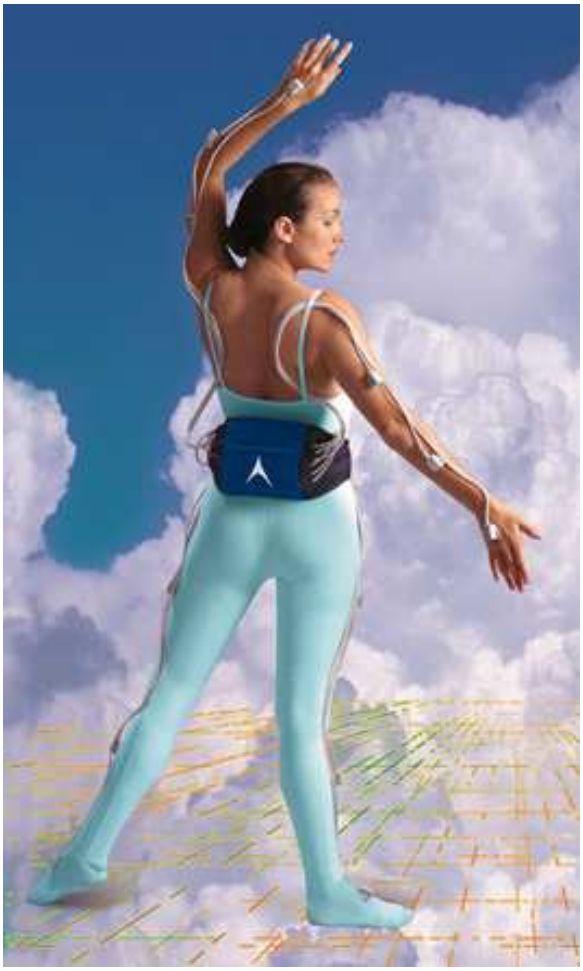

Figure 5: **A performer wearing a motion capture apparatus.** The device shown is a full body magnetic tracking system with a wireless interface. Image courtesy of Ascension Technology Corporation.

4 Motion Capture

A third technique for generating motion, *motion capture*, employs special sensors, called *trackers*, to record the motion of a human performer (Figure 5). The recorded data is then used to generate the motion for an animation. Alternatively, special puppets with joint angle sensors can be used in place of a human performer.

Motion capture is a very popular technique because of the relative ease with which many human motions can be recorded. However, a number of problems prevent it from becoming an ideal solution for all applications. First, accurately measuring the motion of the human body is tricky because trackers attached to skin or clothing shift as the performer moves, creating errors in the recorded data. Furthermore, if the object used to generate the recorded motion and the graphical object have different dimensions, the animation may have noticeable flaws. For example, if the actor were resting her arms on a real table, the arms of the graphical actor might appear to be suspended in the air or sunk into the table.

The technology used for motion capture makes it difficult to capture some motions. One class of sensors are magnetic, and metal in the environment creates noise in the data. Another class of sensors requires that the actor be connected to the computer by an umbilical cord thereby restricting the actor's motion. All sensing technologies have a relatively small field of view, which limits the kinds of actions that can be captured.

In spite of these difficulties, motion capture is widely used. Much of the motion found in commercial animation is generated by using captured data and "tweaking" the results by hand.

5 Suggested Reading

For more information about the implementation and theory behind these techniques, we recommend the following texts.

References

[1] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics, Principles and Practice, Second Edition.* Addison-Wesley, Reading, MA, 1990.

[2] Isaac Victor Kerlow. *The Art of 3-D Computer Animation and Imaging.* Van Nostrand Reinhold, New York, NY, 1996.

[3] Richard Taylor. *The Encyclopedia of Animation Techniques.* Quarto Inc., London, England, 1996.

[4] Frank Thomas and Ollie Johnston. *Disney Animation: The Illusion of Life.* Abbeville Press, New York, NY, 1984.

[5] Alan Watt and Mark Watt. *Advanced Rendering and Animation Techniques: Theory and Practice.* Addison-Wesley, Reading, MA, 1991.