Computer Animation

Jessica K. Hodgins, James F. O'Brien, and Robert E. Bodenheimer, Jr.

College of Computing and Graphics, Visualization, and Usability Center

Georgia Institute of Technology

Atlanta, GA 30332-0280

[jkh|obrienj|bobbyb]@cc.gatech.edu

Animation is the production of consecutive images, which, when displayed, convey a feeling of motion. 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 was created by drawing images of the characters for each frame in the action. At the start of the production, the animator is given *storyboards*, which are sketches depicting the sequence of major actions and illustrating the expressions of the characters. The animator also works from a finished soundtrack, which determines the timing of the piece. In older animations, the background scenery was often stationary and the characters were painted on *cels*, pieces of clear celluloid that could be stacked on top of the background. Most hand animation is created with *keyframing* where a lead animator creates the key, or most important frames, and a second animator creates the in between frames. Regardless of the medium, the challenge for the animator is to create images that impart expressiveness and life to the character.

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

In addition to providing tools that give the animator new capabilities, the computer also creates new applications for animation. Computer animations can be generated in real-time for use in video games and other

---

An earlier version of this article was published in the *Encyclopedia of Computer Science*, Thomson Press.

interactive media. Combining puppeteering with computer animation allows a human operator to control an interactive character in a live performance. Realistic rendering and animation techniques enable the creation of digital actors that can be seamlessly blended with real world footage.

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.

1 Two-dimensional Animation

Two-dimensional (2D) animation techniques contribute a great deal to computer animation by providing the tools used for sprite-based animation, blending or morphing between images, embedding graphical objects in video footage, or creating abstract patterns from mathematical equations. The impact of 2D techniques can be as spectacular as the addition of the film character E.T. to a shot of the moon or as subtle as the erasing of lines around matte boxes for the TIE Fighters in *Star Wars*.

The most common form of 2D animation is *sprite* animation. A sprite is a bitmap image or set of images that are composited over a background, producing the illusion of motion. They are usually small with respect to the size of the screen. For example, to animate a rabbit hopping across a meadow, the animator would create a sequence of images showing poses of the rabbit hopping. This sequence of images would then be composited one image per frame onto a background image of the meadow. Sprite-based animation can be done extremely quickly with current graphics hardware, and thus many elements of the scene can be moving simultaneously. The disadvantage of this technique is that the sprites come from a fixed library and subtle changes in lighting and depth cannot be reproduced. Consequently, sprite animation is most often used in interactive media where rendering speed is more important than realism.

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. Morphing is remarkable because it provides a startling yet convincing transformation of one image into another. Unfortunately, morphing is labor intensive because the key elements of each image must be specified by hand, although automatic feature detection is an area of active research.

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.

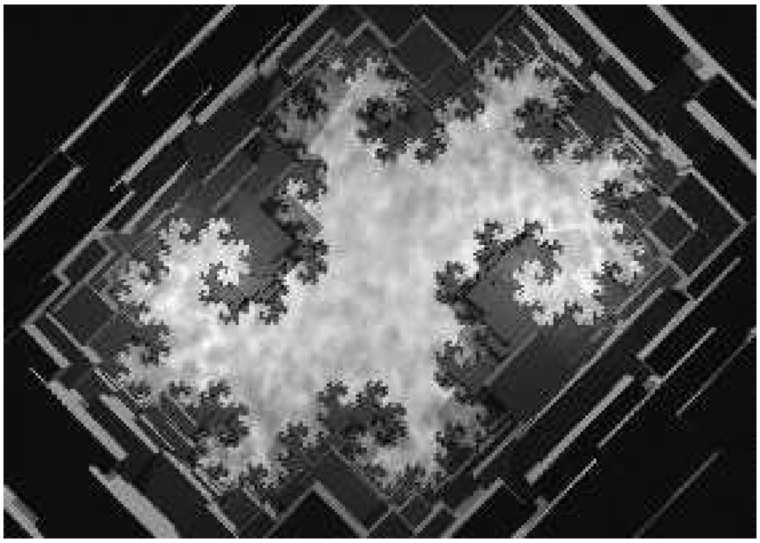

Figure 1: **Image created from fractal equations.** The abstract image in this figure was created by using a type of complex mathematical equation known as a fractal. Image courtesy of Daryl H. Hepting.

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 techniques.

2 Three-dimensional Animation

Three-dimensional animation involves constructing a virtual world in which characters and objects move and interact. The animator must model, animate, and render the 3D scene. Briefly stated, modeling involves describing the elements of a scene and placing them appropriately. Animating specifies how the objects should move in the 3D world. Rendering converts the description of the objects and their motion into images. Modeling and rendering are, for the most part, independent of their role in the animation process but a few necessary modifications are described below.

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

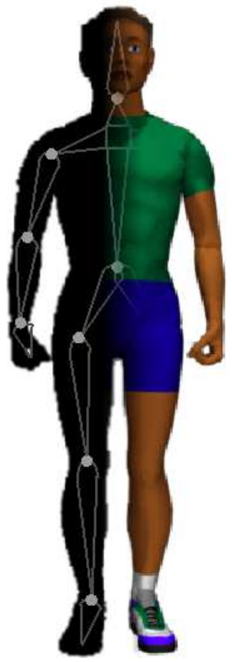

Figure 2: **An articulated model of a human male:** The structure of the joint hierarchy is shown on the left. The graphical model used for rendering is shown on the right. Image courtesy of the Graphics, Visualization and Usability Center, Georgia Institute of Technology.

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 to control 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 modeled well by particle systems include water spray, smoke, and even flocks of birds.

Deformable objects are a third type of model and include 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 water in the pool is a deformable model, and the spray is a particle system.

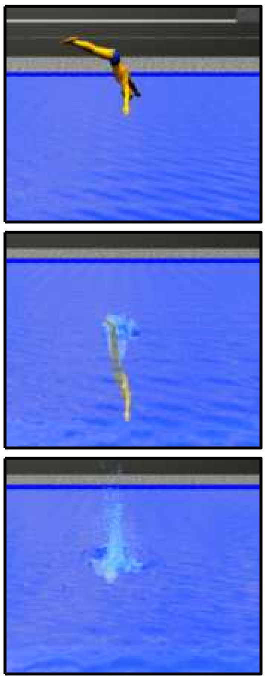

Figure 3: **A diver entering pool:** These images show the combined use of an articulated model, a deformable model, and a particle system. Images courtesy of the Graphics, Visualization and Usability Center, Georgia Institute of Technology.

2.2 Rendering Requirements

Motion blur is a rendering technique that is required for animation but not for most still images. Animations usually display images at 24 or 30 frames per second, and thus a continuous motion is being sampled. This sampling process causes the rapid motion of an object to create unpleasant strobing effects, because high frequencies are masquerading as low frequencies. For example, objects such as wheels may appear to move in the wrong direction. This phenomenon, called *aliasing*, is a well-known problem in signal processing. 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 a frame. This technique creates a blurred representation of the object. While it may seem strange to think that quality can be improved by blurring, humans perceive a motion-blurred animation as more realistic.

3 Motion Generation

The task of specifying the motion of an animated object to the computer is surprisingly difficult. 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. The animator must be able to specify subtle details of the motion 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 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 outline the motion by specifying key positions for the objects being animated. In a process known as *in-betweening*, the computer interpolates to determine the positions for the intermediate frames. For example, to keyframe hitting a baseball, the animator would pose a batter at several key moments in the sequence, such as the batter's initial stance, the contact with the ball, and the follow through. The remaining images would be filled in by the computer. 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*. This technique aids 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, an inverse kinematics algorithm could determine the elbow and shoulder angles. Commercial animation packages include inverse kinematics and interpolation routines designed 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 intimately understand 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; nevertheless, algorithms for specific types of motion can be built. These techniques are called *procedural methods* because a computer 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 will be realistic if the model captures the salient physical characteristics of the situation. For many applications, this realism 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, the animator may use it without understanding the internals of the simulation.

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 the motion is determined by the physical laws and the initial conditions of the system. Pools of water, clothing, hair, and leaves have been animated using passive simulations.

Active systems 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 contains 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. For slightly simpler systems, they can be designed automatically using optimization techniques. After a particular control system has been built, an animator can use it by giving high-level commands such as stand, walk fast, or jump without understanding its internal details. Figure 4 shows a simulation of a running human. To compute the running motion, the animator specifies the desired velocity and a control system generates the 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.

The main advantage procedural methods have over other techniques is the potential for generating interactive behaviors that respond precisely to the actions of the user. In a video game, for example, predicting the behavior of the game player in every situation is impossible, but the characters should appear to be reacting to the actions of the player. Procedural methods allow this capability by computing a response in real-time. While methods using keyframing can also respond to the player, they can only do so by picking from a fixed library of responses. Although procedural methods are currently computationally too expensive to generate motion in real-time for complicated scenes, advances in computer technology may render this possible.

The automatic nature of simulation has a cost in that the animator is not able to control the fine details of the motion. As a result, characters often lack expressiveness or individuality in their motions. Creating tools to allow the animator to control these aspects of a character is a topic of current research.

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 his 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. Some sensors require that the actor be connected to the computer by an umbilical cord, thereby restricting the actor's motion. Another class of sensors are optical and occlusions caused by props and other body parts create confusion in the data. 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 because it automatically captures the subtleties of human motion. An animated figure generated from motion capture of Michael Jackson dancing, for example, will be recognizable as Michael Jackson. The subtle mannerisms that motion capture contains are currently beyond the reach of procedural techniques, and much of the motion found in commercial animation is generated by using captured data and "tweaking" the results by hand.

5 The Future

In the future, the three techniques of keyframing, procedural approaches and motion capture will be merged to produce systems with the advantages of each approach. For example, motion capture data can be used a source of information in the construction of a simulation. Short simulations can be used to blend between motion capture segments. Keyframing techniques can be applied to motion that was generated either procedurally or with motion capture.

The demand for animated motion is already high because of the popularity of computer-generated characters in video games and special effects in film. The list of applications that require animation will grow as improvements in graphic accelerators make animated user interfaces, graphical software for education, and virtual environments for training and entertainment feasible.

6 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.

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.

Figure 5: **A performer wearing a motion capture apparatus.** The device shown is a full body magnetic tracking system. Image courtesy Bobby Bodenheimer