

Chapter 6- Computer Animation

Computer Animation

1. Raster Methods for Computer Animation
2. Design of Animation Sequences
3. Traditional Animation Techniques
4. Computer-Animation Languages
5. OpenGL Animation Procedures

Computer Animation

- Computer-graphics methods are now commonly used to produce animations for a variety of applications, including **entertainment** (motion pictures and cartoons), **advertising**, **scientific** and **engineering studies**, and **training** and **education**.
- The term **computer animation** generally refers to any time sequence of visual changes in a picture.
- In addition to changing object positions using translations or rotations, a computer-generated animation could display time variations in object size, color, transparency, or surface texture.

Computer Animation

- Two basic methods for constructing a motion sequence are **real-time animation** and **frame-by-frame animation**.
- In a **real-time** computer-animation, each stage of the sequence is viewed as it is created.
- Thus the animation must be generated at a rate that is compatible with the constraints of the refresh rate.
- For a **frame-by-frame** animation, each frame of the motion is separately generated and stored.
- Later, the frames can be recorded on film, or they can be displayed consecutively on a video monitor in “real-time playback” mode.

- Simple animation displays are generally produced in real time, while more complex animations are constructed more slowly, frame by frame.

Raster Methods for Computer Animation

- Most of the time, we can create simple animation sequences in our programs using real-time methods.
- In general, though, we can produce an animation sequence on a raster-scan system one frame at a time, so that each completed frame could be saved in a file for later viewing.
- The animation can then be viewed by cycling through the completed frame sequence, or the frames could be transferred to film.

Raster Methods for Computer Animation

- If we want to generate an animation in real time, however, we need to produce the motion frames quickly enough so that a continuous motion sequence is displayed.
- For a complex scene, one frame of the animation could take most of the refresh cycle time to construct.
- In that case, objects generated first would be displayed for most of the frame refresh time, but objects generated toward the end of the refresh cycle would disappear almost as soon as they were displayed.

Raster Methods for Computer Animation

- For very complex animations, the frame construction time could be greater than the time to refresh the screen, which can lead to erratic motion and fractured frame displays.
- Because the screen display is generated from successively modified pixel values in the refresh buffer, we can take advantage of some of the characteristics of the raster screen-refresh process to produce motion sequences quickly.

Double Buffering

- One method for producing a real-time animation with a raster system is to employ two refresh buffers.
- Initially, we create a frame for the animation in one of the buffers. Then, while the screen is being refreshed from that buffer, we construct the next frame in the other buffer.
- When that frame is complete, we switch the roles of the two buffers so that the refresh routines use the second buffer during the process of creating the next frame in the first buffer.

Double Buffering

- This alternating buffer process continues throughout the animation.
- Graphics libraries that permit such operations typically have one function for activating the double buffering routines and another function for interchanging the roles of the two buffers.
- The most straight forward implementation is to switch the two buffers at the end of the current refresh cycle, during the vertical retrace of the electron beam.
- If a program can complete the construction of a frame within the time of a refresh cycle, say $\frac{1}{60}$ of a second, each motion sequence is displayed in synchronization with the screen refresh rate.

Double Buffering

- However, if the time to construct a frame is longer than the refresh time, the current frame is displayed for two or more refresh cycles while the next animation frame is being generated.
- For example, if the screen refresh rate is 60 frames per second and it takes of a second to construct an animation frame, each frame is displayed on the screen twice and the animation rate is only 30 frames each second.
- Similarly, if the frame construction time is $\frac{2}{3}$ of a second, the animation frame rate is reduced to 20 frames per second because each frame is displayed three times.

Double Buffering

- Irregular animation frame rates can occur with double buffering when the frame construction time is very nearly equal to an integer multiple of the screen refresh time.
- A san example of this, if the screen refresh rate is 60 frames per second, then an erratic animation frame rate is possible when the frame construction time is very close to $\frac{1}{60}$ of a second, or $\frac{2}{60}$ of a second, or $\frac{3}{60}$ of a second, and so forth.

Double Buffering

- Because of slight variations in the implementation time for the routines that generate the primitives and their attributes, some frames could take a little more time to construct and some a little less time.
- Thus, the animation frame rate can change abruptly and erratically.
- One way to compensate for this effect is to add a small time delay to the program.
- Another possibility is to alter the motion or scene description to shorten the frame construction time.

Operations

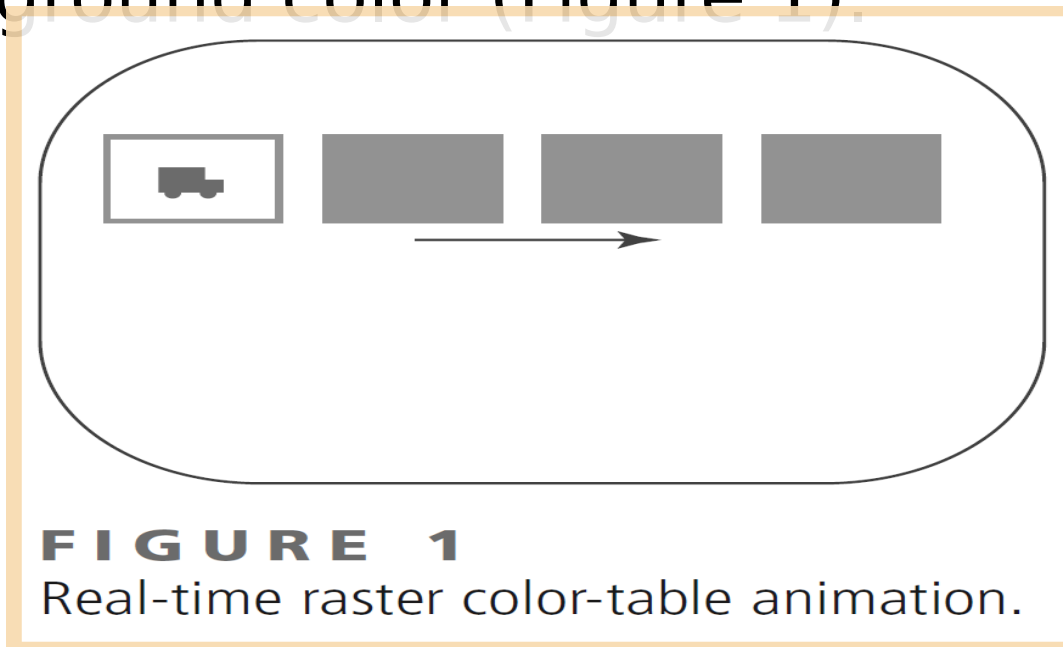
- We can also generate real-time raster animations for limited applications using block transfers of a rectangular array of pixel values.
- This animation technique is often used in game-playing programs.
- A simple method for translating an object from one location to another in the xy plane is to transfer the group of pixel values that define the shape of the object to the new location.
- Two-dimensional rotations in multiples of 90 are also simple to perform.

Operations

- Sequences of raster operations can be executed to produce realtime animation for either two-dimensional or three-dimensional objects.
- We can also animate objects along two-dimensional motion paths using **colortable transformations**.
- Here we predefine the object at successive positions along the motion path and set the successive blocks of pixel values to color-table entries.
- The pixels at the first position of the object are set to a foreground color, and the pixels at the other object positions are set to the background color.

Operations

- The animation is then accomplished by changing the color-table values so that the object color at successive positions along the animation path becomes the foreground color as the preceding position is set to the background color (Figure 1).



Design of Animation Sequences

- Constructing an animation sequence can be a complicated task, particularly when it involves a story line and multiple objects, each of which can move in a different way.
- A basic approach is to design such animation sequences using the following development stages:
 - Storyboard layout
 - Object definitions
 - Key-frame specifications
 - Generation of in-between frames

Design of Animation Sequences

- The **storyboard** is an outline of the action. It defines the motion sequence as a set of basic events that are to take place.
- Depending on the type of animation to be produced, the storyboard could consist of a set of rough sketches, along with a brief description of the movements, or it could just be a list of the basic ideas for the action.
- Originally, the set of motion sketches was attached to a large board that was used to present an overall view of the animation project. Hence, the name “storyboard.”

Design of Animation Sequences

- An **object definition** is given for each participant in the action.
- Objects can be defined in terms of basic shapes, such as polygons or spline surfaces.
- In addition, a description is often given of the movements that are to be performed by each character or object in the story.

Design of Animation Sequences

- A **key frame** is a detailed drawing of the scene at a certain time in the animation sequence.
- Within each key frame, each object (or character) is positioned according to the time for that frame.
- Some key frames are chosen at extreme positions in the action; others are spaced so that the time interval between key frames is not too great.
- More key frames are specified for intricate motions than for simple, slowly varying motions.
- Development of the key frames is generally the responsibility of the senior animators, and often a separate animator is assigned to each character in the animation.

Design of Animation Sequences

- **In-betweens** are the intermediate frames between the key frames.
- The total number of frames, and hence the total number of in-betweens, needed for an animation is determined by the display media that is to be used.
- Film requires 24 frames per second, and graphics terminals are refreshed at the rate of 60 or more frames per second.
- Typically, time intervals for the motion are set up so that there are from three to five in-betweens for each pair of key frames.
- Depending on the speed specified for the motion, some key frames could be duplicated.
- As an example, a 1-minute film sequence with no duplication requires a total of 1,440 frames. If five in-betweens are required for each pair of key frames, then 288 key frames would need to be developed.

Design of Animation Sequences

- There are several other tasks that may be required, depending on the application.
- These additional tasks include motion verification, editing, and the production and synchronization of a soundtrack.
- Many of the functions needed to produce general animations are now computer-generated.

Traditional Animation Techniques

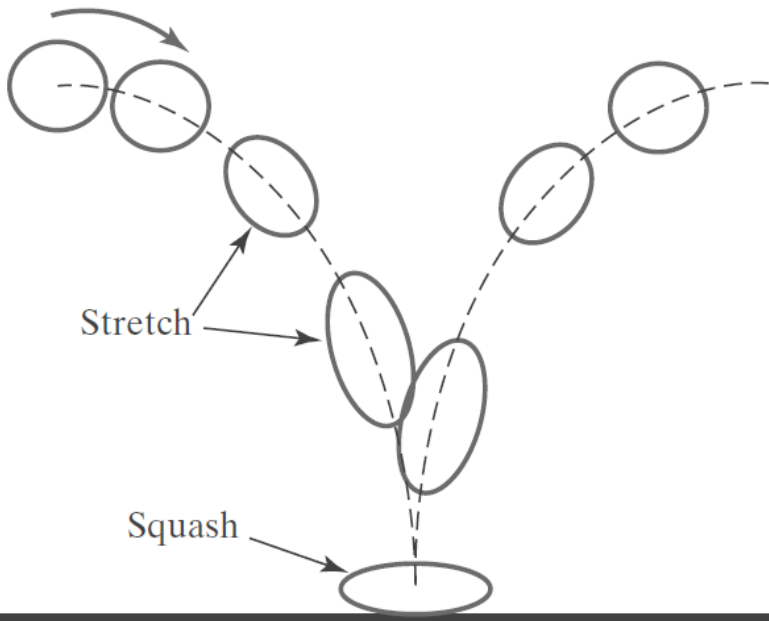


FIGURE 4

A bouncing-ball illustration of the "squash and stretch" technique for emphasizing object acceleration.

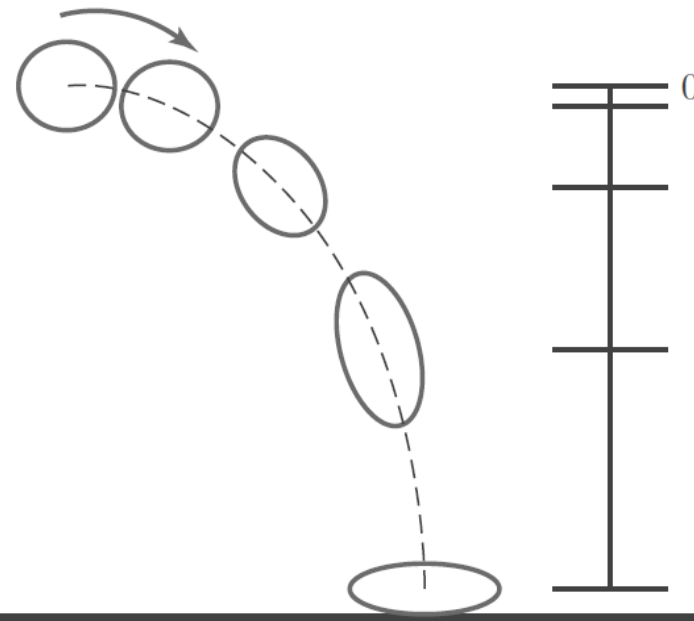


FIGURE 5

The position changes between motion frames for a bouncing ball increase as the speed of the ball increases.

Computer-Animation Languages

- We can develop routines to design and control animation sequences within a general-purpose programming language, such as C, C++, Lisp, or Fortran, but several specialized animation languages have been developed.
- These languages typically include a graphics editor, a key-frame generator, an in-between generator, and standard graphics routines.
- The graphics editor allows an animator to design and modify object shapes, using spline surfaces, constructive solid geometry methods, or other

OpenGL Animation Procedures

- Double buffering operations can be specified