Chapter 16: Computer Animation

INTRODUCTION

- The term ***computer animation*** generally refers to any time sequence of visual changes in a scene.
- In addition to changing object position with translations or rotations, a computer-generated animation could display time variations in object size, color, transparency, or surface texture.
- Computer animations can be generated by changing camera parameters, such as position, orientation, and focal length.
- Computer animations can also be produced by changing lighting effects.

DESIGN OF ANIMATION SEQUENCES

An animation sequence is designed with the following steps:

- Storyboard layout
- Object definitions
- Key-frame specifications
- Generation of in-between frames

STORYBOARD LAYOUT

- It 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 or it could be a list of the basis ideas for the motion

OBJECT DEFINATION

- An object definition is given for each participant in the action.
- Objects can be defined in terms of basic shapes, such as polygons or splines.
- Along with the shape, the associated movements for each object are specified.

KEY FRAME SPECIFICATION

- A key frame is a detailed drawing of the scene at a certain time in the animation sequence.
- Within each key frame, each object 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.

IN-BETWEEN FRAMES

- It is a process of generating intermediate frames between two images to give appearance that the first image evolves smoothly into the second image. In–betweens are the drawing between the key frames which help to create the illusion of motion.

- In-between are the intermediate frames between the key frames
- The number of in-betweens needed is determined by the media to be used to display the animation. Film requires 24 frames per second, and graphics terminals are refreshed at the rate of 30 to 60 frames per second
- Time intervals for the motion are set up so that there are from three to five in-betweens for each pair of key frames.

There are several other tasks that may be required, depending on the application. They include:

- Motion verification
- Editing
- Production and synchronization of a soundtrack

GENERAL COMPUTER ANIMATION FUNCTIONS

- Some steps included the development of an animation sequence are:
- Object manipulations and rendering
- Camera motions
- Generation of in-betweens

- Animation packages, ***such as Wave-front***, provide special functions for designing the animation and processing individual objects.

- In animation packages, one function is provided ***to store and manage the object database***. Object shapes are stored and updated in the database.

- Other object functions include those ***for motion generation and object rendering***. Motions can be generated according to specified constraints using two-dimensional or three-dimensional transformations.

- Another function ***simulates camera movements***. Standard motions are zooming, panning, and tilting.

GENERAL COMPUTER-ANIMATION FUNCTIONS

- Animation packages ,such as wave front , provide special functions for designing the animation and processing individual objects.

- Some steps included in the development of animation sequence are-
  - Object manipulation and rendering
  - Camera motions .
  - Generation of in-betweens .

▪ONE FUNCTION AVAILABLE IN ANIMATION PACKAGES IS PROVIDED TO STORE AND MANAGE THE OBJECT DATABASE . (OBJECT SHAPES AND ASSOCIATED PARAMETERS ARE STORED AND UPDATED IN THE DATABASE).

- Other object functions include:-
  - Object motion generation
    (2-D or 3-D transformations)
  - Object rendering .
- One function to stimulate Camera Movements:-
  - Zooming
  - Panning
    (rotating horizontally or vertically)
  - Tilting .

RASTER ANIMATIONS

On Raster systems ,we generate real-time animation in limited application using raster operation.

Such as 2-D or 3-D transformations.

We can also animate objects along 2D motion paths using the color-table transformations.

The pixel value at successive positions along the motion path of an object are stored in color-table and the pixel at 1st pixel is set on , we set the pixel at the other object positions to the background color.

KEY-FRAME SYSTEMS

A key-frame in animation is a drawing that defines the starting and ending points of any smooth transition. The drawings are called "frames" because their position in time is measured in frames on a strip of film. A sequence of key frames defines which movement the viewer will see, whereas the position of the key frames on the film, video or animation defines the timing of the movement. Because only two or three key frames over the span of a second do not create the illusion of movement, the remaining frames are filled with in-betweens.

With complex object transformations, the shapes of objects may change over time. Examples are clothes, facial features, magnified detail, evolving shapes, exploding or disintegrating objects, and transforming one object into another object.

If all surfaces are described with polygon meshes, then the number of edges per polygon can change from one frame to the next. Thus, the total number of line segments can be different in different frames.

MORPHING

Transformation of object shapes from one form to the other is termed as morphing as short form of metamorphism. This method can be applied to any of motion or transition relating a change in shape.

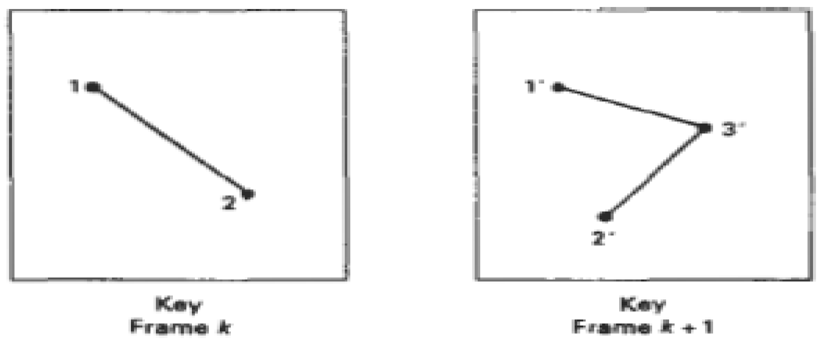
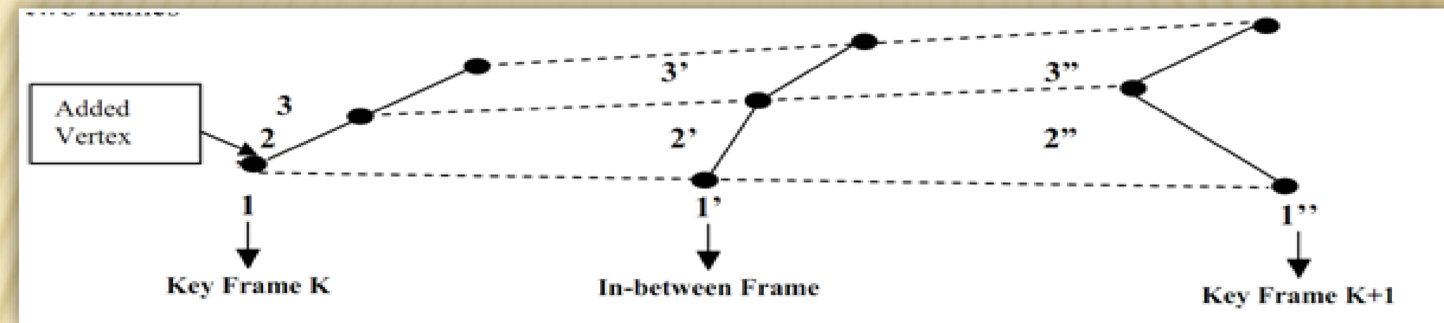


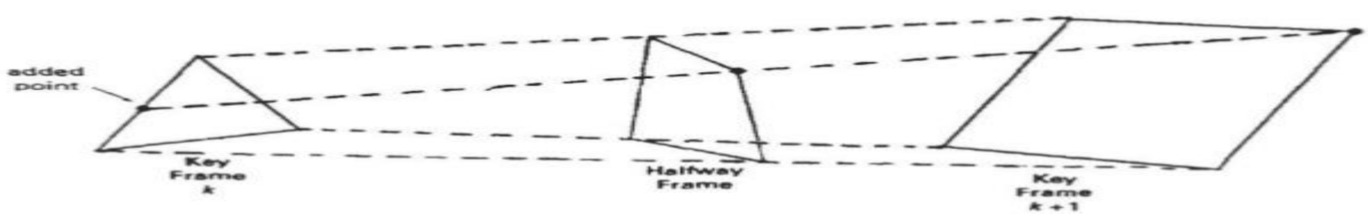
Figure 16–6
An edge with vertex positions 1 and 2 in key frame k evolves into two connected edges in key frame $k + 1$.

Given two key frames for an object transformation, we first adjust the object
specification in one of the frames so that the number of polygon edges (or the number of vertices) is the same for the two frames

A straight-line segment in key frame k is transformed into
two line segments in key frame k +1. Since key frame k + 1 has an extra vertex,
we add a vertex between vertices 1 and 2 in key frame k to balance the number of vertices (and edges) In the two key frames. Using linear interpolation to generate the in-betweens, we transition the added vertex in key frame k into vertex 3‘ along the straight-line path shown in Fig.



Morphing



An example of a triangle linearly expanding into a quadrilateral given In Fig

The general pre processing rules for equalizing key frames in terms of either the number of vertices to be added to a key frame.

Suppose we equalize the **edge count** and parameters L_k and L_{k+1} denote the number of line segments in two consecutive frames. We define,

$L_{max} = \max(L_k, L_{k+1})$
$L_{min} = \min(L_k, L_{k+1})$
$N_e = L_{max} \bmod L_{min}$
$N_s = \text{int}(L_{max}/L_{min})$

The pre processing is accomplished by

•Dividing N_e edges of key $frame_{min}$ into N_s+1 section.

•Dividing the remaining lines of key $frame_{min}$ into N_s sections.

For example, if $L_k = 15$ and $L_{k+1} = 11$, we divide 4 lines of key-$frame_{k+1}$ into 2 sections each. The remaining lines of $keyframe_{k+1}$ are left intact.

If we equalize the **vertices counts** then the parameters Vk and Vk+1 are used to denote the number of vertices in the two consecutive frames. In this case we define:

$V_{max} = \max(V_k, V_{k+1})$
$V_{min} = \min(V_k, V_{k+1})$ and
$N_{ls} = (V_{max} - 1) \bmod (V_{min} - 1)$
$N_p = \mathrm{int}((V_{max} - 1)/(V_{min} - 1))$

Preprocessing using vertex count is performed by:

• Adding N_p points to N_{ls} line section of key-frame$_{min}$.

• Adding N_p-1 points to the remaining edges of key-frame$_{min}$.

For the triangle-to quadrilateral example, $V_k = 3$ and $V_{k+1} = 4$.
Both N_{ls} and N_p are 1, so we would add one point to one edge of key-frame$_k$
No points would be added to the remaining lines of keyframe$_{k+1}$.

SIMULATING ACCELERATIONS

Curve-fitting techniques are often used to specify the animation paths between
key frames. Given the vertex positions at the key frames, we can fit the positions
with linear or nonlinear paths.
This figure illustrates a nonlinear fit of key-frame positions. To simulate accelerations, we can adjust the time spacing for the in-betweens.

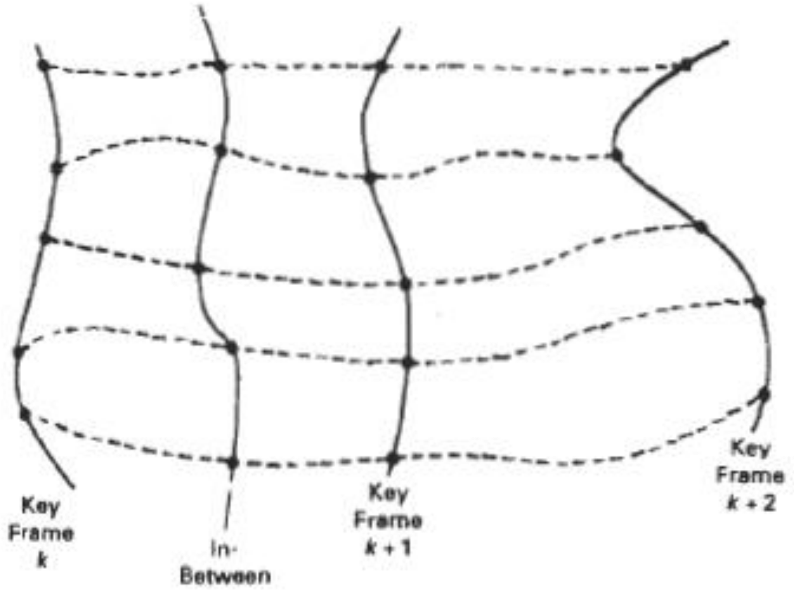


Figure 16-11
Fitting key-frame vertex positions with nonlinear splines.

For constant speed (zero acceleration), we use equal-interval time spacing for the in-betweens. Suppose we want ***n*** in-betweens for key frames at times ***t1*** and ***t2*** . The time interval between key frames is then divided into **n +1** subintervals, yielding an in-between spacing of

$$\Delta t = \frac{t_2 - t_1}{n + 1}$$

We can calculate the time for any in-between as

$$tB_j = t_1 + j\,\Delta t, \qquad j = 1, 2, \ldots, n$$

Nonzero accelerations are used to produce realistic displays of speed changes, particularly at the beginning and end of a motion sequence. We can model the start-up and slowdown portions of an animation path with spline or trigonometric functions. Parabolic and cubic time functions have been applied to acceleration modelling, but trigonometric functions are more commonly used in animation packages.

To model increasing speed (positive acceleration), we want the time spacing between frames to increase so that greater changes in position occur as the object moves faster. We can obtain an increasing interval size with the function

$$1 - \cos\theta, \qquad 0 < \theta < \pi/2$$

For n in-betweens, the time for the **jth** in-between would then be calculated as:
where Δt the time difference between the two key frames

$$tB_j = t_1 + \Delta t\left[1 - \cos\frac{j\pi}{2(n+1)}\right], \qquad j = 1, 2, \ldots, n$$

This figure gives a plot of the trigonometric acceleration function and the in-between spacing for n= 5.

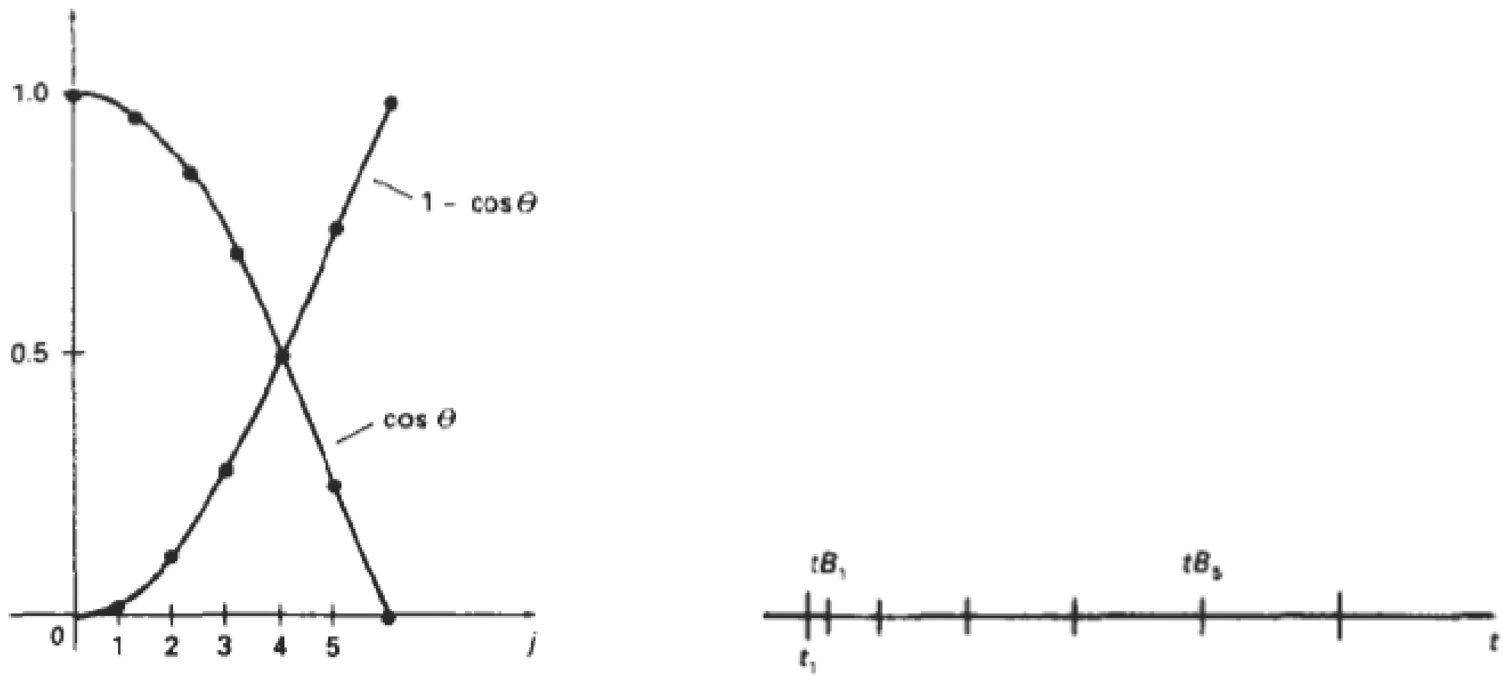


Figure 16-13
A trigonometric acceleration function and the corresponding in-between spacing for $n = 5$ and $\theta = j\pi/12$ in Eq. 16-7, producing increased coordinate changes as the object moves through each time interval.

Often, motions contain both speed-ups and slow-downs. We can model a combination of increasing-decreasing speed by first increasing the in-between time spacing, then we decrease this sparing. A function to accomplish these time changes is

$$\frac{1}{2}(1 - \cos\theta), \qquad 0 < \theta < \pi/2$$

The time for the jth in-between is now calculated as:

$$tB_j = t_1 + \Delta t\left\{\frac{1 - \cos[j\pi/(n + 1)]}{2}\right\}, \qquad j = 1, 2, \ldots, n$$

with Δ ***t*** denoting the time difference for the two key frames. Time intervals for the moving object first increase, then the time ntervals decrease, as shown in next figure:

We can model decreasing speed (deceleration) with $\sin\theta$ in the range $0<\theta<\pi/2$. The time position of an in-between is now defined as:

$$tB_j = t_1 + \Delta t \sin \frac{j\pi}{2(n+1)}, \qquad j = 1, 2, \ldots, n$$

A plot of this function and the decreasing size of the time intervals is shown in the next figure for five in-betweens

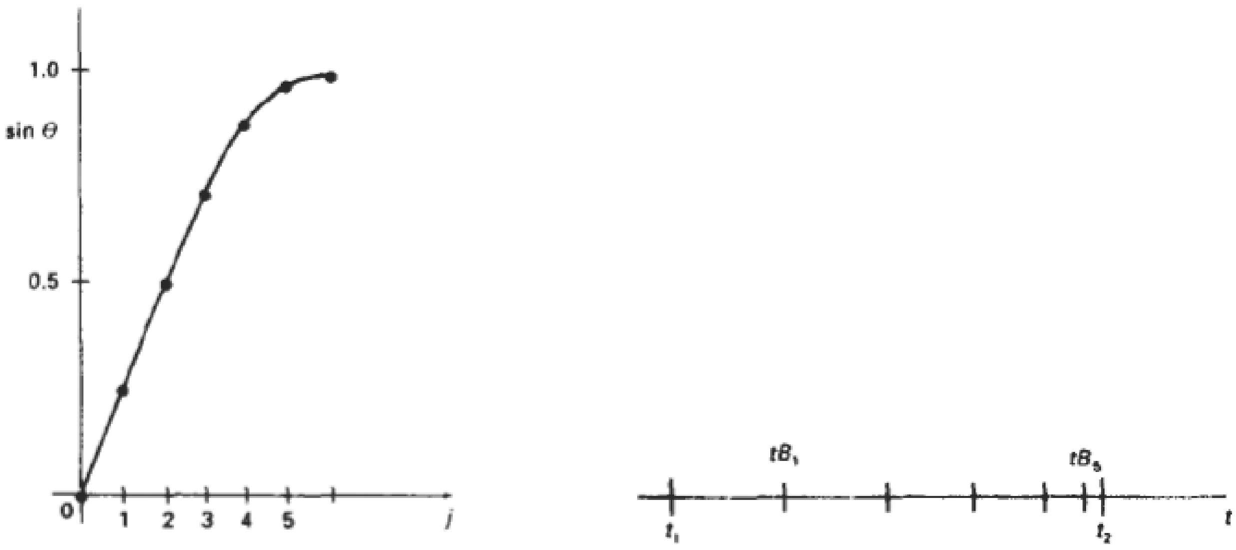


Figure 16-14
A trigonometric deceleration function and the corresponding in-between spacing for $n = 5$ and $\theta = j\pi/12$ in Eq. 16-8, producing decreased coordinate changes as the object moves through each time interval.

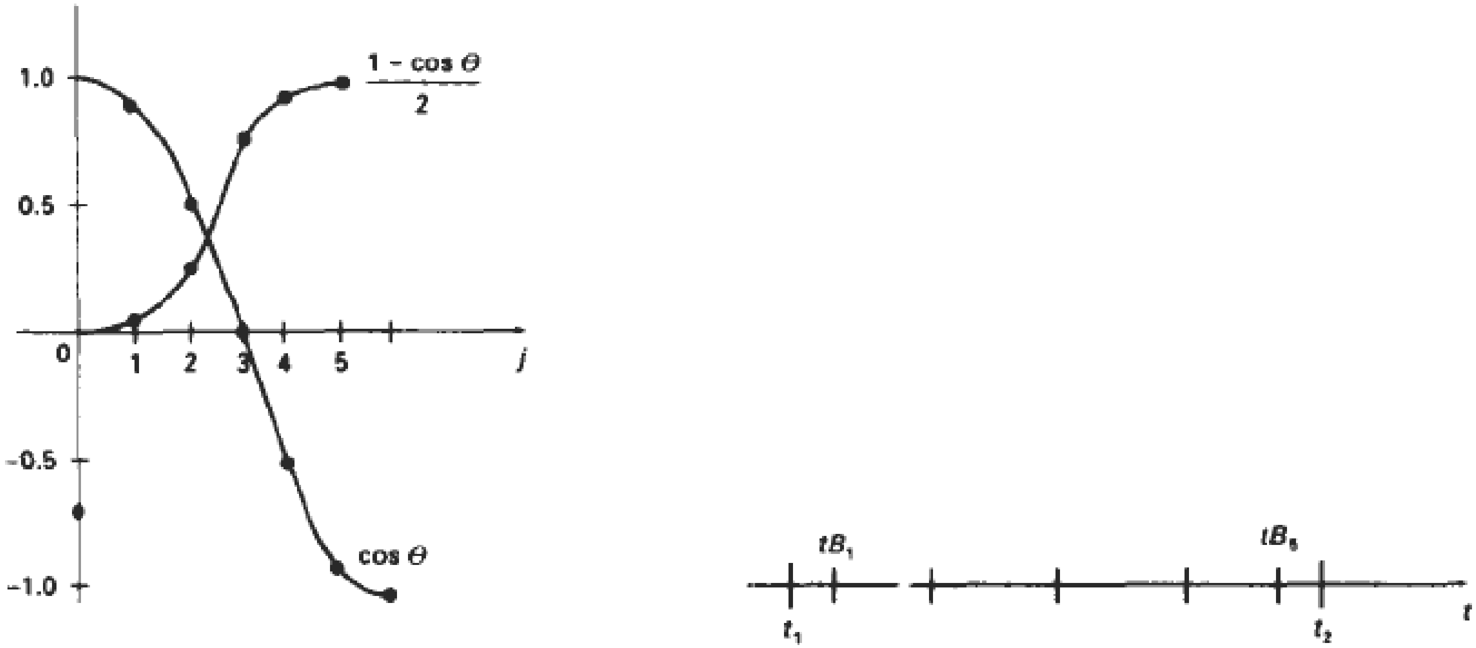


Figure 16-15
A trigonometric accelerate–decelerate function and the corresponding in-between spacing for $n = 5$ in Eq. 16-9.

Motion Specifications

- There are several ways in which the motions of objects can be specified in an animation system.
- Direct Motion Specification:
- * We explicitly give the rotation angles and translation vectors. The geometric transformations are applied to transform coordinate positions.
- * We could use an approximating equation to specify certain kinds of motions like bouncing ball, with sine curve.

- **Goal – Directed Systems:**

- We can specify the motions that are to take place in general terms that abstractly describe the actions.

--> Example: We want an object to walk or to run to a particular destination.

- --> We want an object to pick-up some other specified object.

Kinematics and Dynamics:

- * We can construct animation sequences using kinematic or dynamic descriptions. We specify animation by giving motion parameters like position, velocity and acceleration parameters.
- Inverse Kinematics and dynamics:
- * We can specify the initial and final positions of the object and calculations are done by the computer