MODULE 4

THREE-DIMENSIONAL DISPLAY METHODS

- It is field of computer graphics that deals with generating and displaying 3D objects in 2D space.
- In addition to color and brightness a 3D pixels add a depth property that indicates where the point lies on the imaginary Z axis.

3D Display methods

1. Parallel Projection

- One method for generating a view of a solid object is to project points on the object surface along parallel lines onto the display plane.
- By selecting different viewing positions, we can project visible points on the object onto the display plane to obtain different two-dimensional views of the object .
- This technique is used in engineering and architectural drawings to represent an object with a set of views that maintain relative proportions of the object.
- The appearance of the solid object can then be reconstructed from the major views.

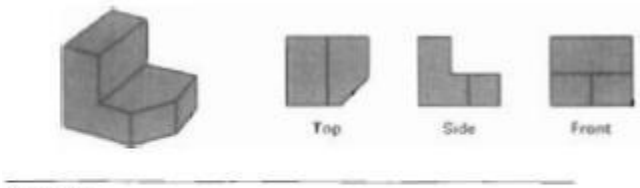


Figure 9-3
Three parallel-projection views of an object, showing relative proportions from different viewing positions.

2. Perspective Projection

- Another method for generating a view of a three-dimensional scene is to project Methods points to the display plane along converging paths.
- This causes objects farther from the viewing position to be displayed smaller than objects of the same size that are nearer to the viewing position.
- In a perspective projection, parallel lines in a scene that are not parallel to the display plane are projected into converging lines.
- Scenes displayed using perspective projections appear more realistic, since this is the way that our eyes and a camera lens form images.
- In the perspective projection view shown in Fig., parallel lines appear to converge to a distant point in the background, and distant objects appear smaller than objects closer to the viewing position

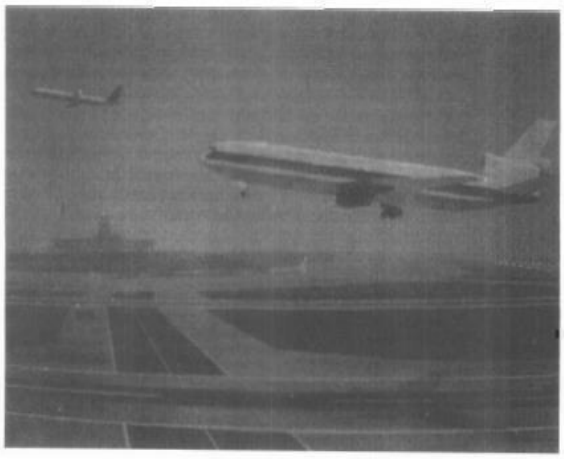


Figure 9-4
A perspective-projection view of an airport scene. (*Courtesy of Evans & Sutherland.*)

3. Depth Cueing

- With few exceptions, depth information is important so that we can easily identify, for a particular viewing direction, which is the front and which is the back of displayed objects.
- Figure 9-5 illustrates the ambiguity that can result when a wireframe object is displayed without depth information. T
- Here are several ways in which we can include depth information in the two- dimensional representation of solid objects.
- A simple method for indicating depth with wireframe displays is to vary the intensity of objects according to their distance from the viewing position. Figure 9-6 shows a wireframe object displayed with depth cueing.

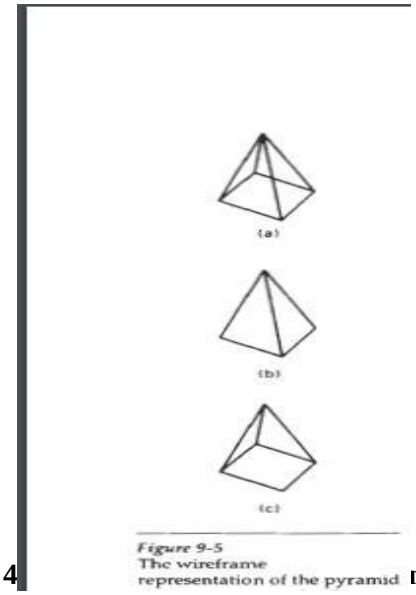


Figure 9-5
The wireframe representation of the pyramid

4 ntification

- We can also clarify depth relationships in a wireframe display by identifying visible lines in some way.
- The simplest method is to highlight the visible lines or to display them in a different color.

- Another technique, commonly used for engineering drawings, is to display the nonvisible lines as dashed lines.
- Another approach is to simply remove the nonvisible lines, as in Figs. 9-5(b) and 9-5(c).
- But removing the hidden lines also removes information about the shape of the back surfaces of an object.
- These visible-line methods also identify the visible surfaces of objects. When objects are to be displayed with color or shaded surfaces, we apply surface- rendering procedures to the visible surfaces so that the hidden surfaces are obscured.

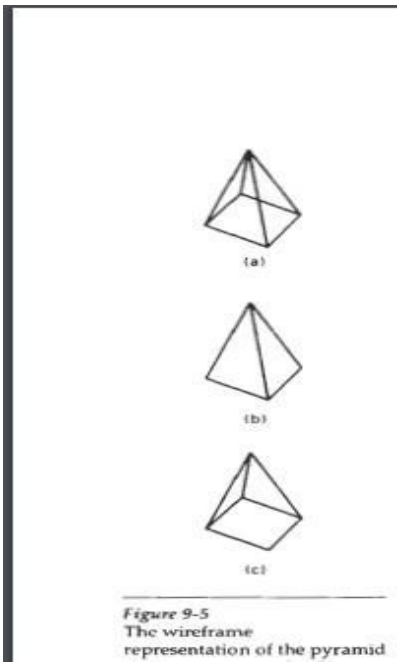


Figure 9-5
The wireframe representation of the pyramid

Surface Rendering

- Added realism is attained in displays by setting the surface intensity of objects according to the lighting conditions in the scene and according to assigned surface characteristics.
- Lightning specifications include the intensity and positions of light sources and the general background illumination required for a scene.
- Surface properties of objects include degree of transparency and how rough or smooth the surfaces are to be.
- Procedures can then be applied to generate the correct illumination and shadow regions for the scene.

Figure 9-7
A realistic room display achieved with stochastic ray-tracing methods that apply a perspective projection, surface-texture mapping, and illumination models. (*Courtesy of John Snyder, Jed Lengyel, Devendra Kalra, and Al Barr, California Institute of Technology.* Copyright © 1992 Caltech.)

5. Exploded and Cutaway Views

- Many graphics packages allow objects to be defined as hierarchical structures, so that internal details can be stored.
- Exploded and cutaway views of such objects can then be used to show the internal structure and relationship of the object parts. Figure shows several kinds of exploded displays for a mechanical design.

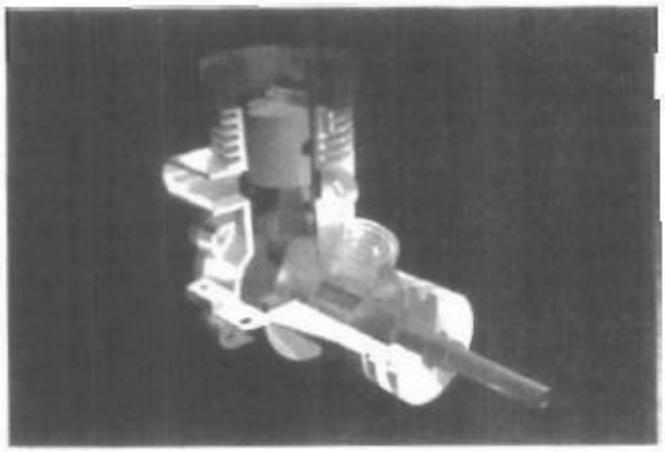


Figure 9-9
Color-coded cutaway view of a lawn mower engine showing the structure and relationship of internal components. (*Courtesy of Autodesk, Inc.*)

6. Three-Dimensional and Stereoscopic Views

- Another method for adding a sense of realism to a computer-generated scene is to display objects using either three-dimensional or stereoscopic views.
- raster image from a vibrating flexible mirror. The vibrations of the mirror are synchronized with the display of the scene on the CRT.
- As the mirror vibrates, the focal length varies so that each point in the scene is projected to a position corresponding to its depth.
- Stereoscopic devices present two views of a scene: one for the left eye and the other for the right eye. The two views are generated by selecting viewing positions that correspond to the two eye positions of a single viewer.
- \These two views then can be displayed on alternate refresh cycles of a raster monitor, and viewed through glasses that alternately darken first one.
- lens then the other in synchronization with the monitor refresh cycles.

BLOBBY OBJECTS

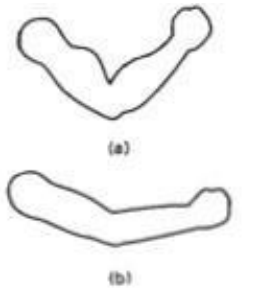


Figure 10-15
Blobby muscle shapes in a human arm.

- Some objects do not maintain a fixed shape, but change their surface characteristics in certain motions or when in proximity to other objects.
- Examples in this class of objects include molecular structures, water droplets and other liquid effects, melting objects, and muscle shapes in the human body.
- These objects can be described as exhibiting "blobbiness" and are often simply referred to as blobby objects, since their shapes show a certain degree of fluidity.

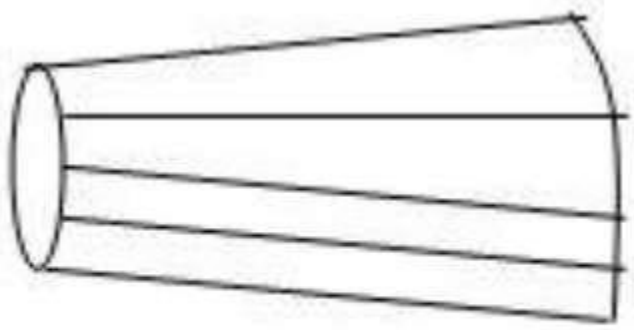
Polygon Surfaces

Objects are represented as a collection of surfaces. 3D object representation is divided into two categories.

- **Boundary Representations** B−reps − It describes a 3D object as a set of surfaces that separates the object interior from the environment.
- **Space–partitioning representations** − It is used to describe interior properties, by partitioning the spatial region containing an object into a set of small, non-overlapping, contiguous solids usually cubes.

The most commonly used boundary representation for a 3D graphics object is a set of surface polygons that enclose the object interior. Many graphics system use this method. Set of polygons are stored for object description. This simplifies and speeds up the surface rendering and display of object since all surfaces can be described with linear equations.

The polygon surfaces are common in design and solid-modeling applications, since their **wireframe display** can be done quickly to give general indication of surface structure. Then realistic scenes are produced by interpolating shading patterns across polygon surface to illuminate.

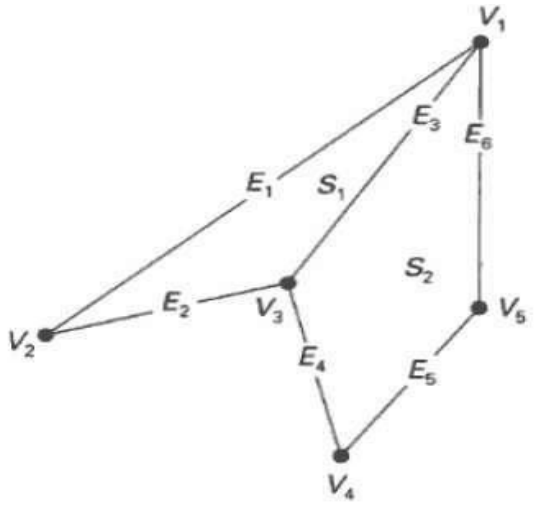


A 3D object represented by polygons

Polygon Tables

In this method, the surface is specified by the set of vertex coordinates and associated attributes. As shown in the following figure, there are five vertices, from v_1 to v_5.

- Each vertex stores x, y, and z coordinate information which is represented in the table as v_1: x_1, y_1, z_1.
- The Edge table is used to store the edge information of polygon. In the following figure, edge E_1 lies between vertex v_1 and v_2 which is represented in the table as E_1: v_1, v_2.
- Polygon surface table stores the number of surfaces present in the polygon. From the following figure, surface S_1 is covered by edges E_1, E_2 and E_3 which can be represented in the polygon surface table as S_1: E_1, E_2, and E_3.



VERTEX TABLE	
V_1:	x_1, y_1, z_1
V_2:	x_2, y_2, z_2
V_3:	x_3, y_3, z_3
V_4:	x_4, y_4, z_4
V_5:	x_5, y_5, z_5

EDGE TABLE	
E_1:	V_1, V_2
E_2:	V_2, V_3
E_3:	V_3, V_1
E_4:	V_3, V_4
E_5:	V_4, V_5
E_6:	V_5, V_1

POLYGON-SURFACE TABLE	
S_1:	E_1, E_2, E_3
S_2:	E_3, E_4, E_5, E_6

SPLINE REPRESENTATIONS

- A spline is a flexible strip used to produce a smooth curve through a designated set of points. Several small weights are distributed along the length of the strip to hold it in position on the drafting table as the curve is drawn. The term spline curve originally referred to a curve drawn in this manner.
- Splines are used in graphics applications to design curve and surface shapes, to digitize drawings for computer storage, and to specify animation paths for the objects or the camera in a scene. Typical CAD applications for splines include the automobile bodies, aircraft and spacecraft surfaces.

Figure 10-19
A set of six control points interpolated with piecewise continuous polynomial sections.

SWEEP REPRESENTATIONS

- Solid-modeling packages often provide a number of construction techniques. Sweep representations are useful for constructing three-dimensional objects that possess translational, rotational, or other symmetries.
- We can represent such objects by specifying a two dimensional shape and a sweep that moves the shape through a region of space.
- A set of two-dimensional primitives, such as circles and rectangles, can be provided for sweep representations as menu options.
- Other methods for obtaining two-dimensional figures include closed spline curve constructions and cross-sectional slices of solid objects

Figure 10-53
Constructing a solid with a translational sweep. Translating the control points of the periodic spline curve in (a) generates the solid shown in (b), whose surface can be described with point function $\mathbf{P}(u,v)$.

Figure illustrates a translational sweep.

The periodic spline curve in Fig. 10-53(a) defines the object cross section.

- We then perform a translational sweep by moving the control points p, through p3 a set distance along a straight line path perpendicular to the plane of the cross section.
- At intervals along this we replicate the cross-sectional shape and draw a set of connecting lines in the direction of the sweep to obtain the wireframe representation shown in Fig. 10-53(b).

- An example of object design using a rotational sweep is given in Fig. 10-54.

- This time, the periodic spline cross section is rotated about an axis of rotation specified in the plane of the cross section to produce the wireframe representation shown in F&. 10-54(b).

- Any axis can be chosen for a rotational sweep. If we use a rotation axis perpendicular to the plane of the spline cross section in Fig. 10-54(a), we generate a two-dimensional shape.

- But if the cross section shown in this figure has depth, then we are using one three-dimensional object to generate another.

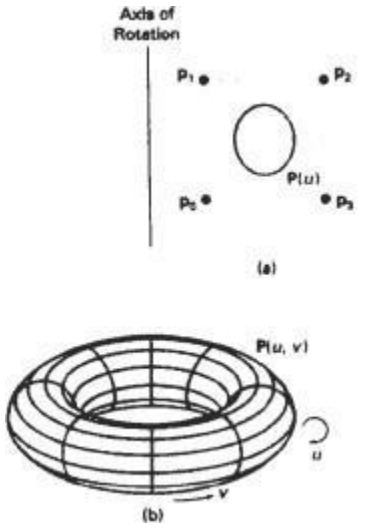


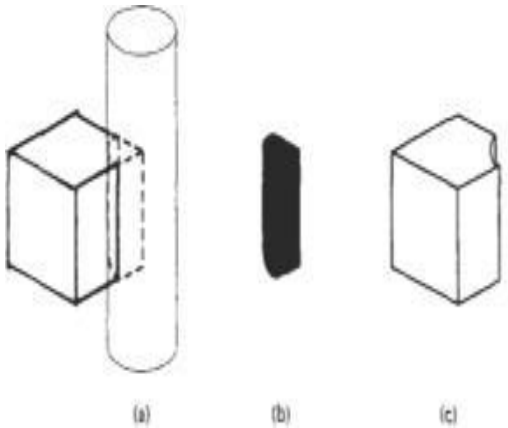
Figure 10-54
Constructing a solid with a rotational sweep Rotating the control points of the periodic spline curve in (a) about the given rotation axis generates the solid shown in (b), whose surface can be described with point function $P(u,v)$.

CONSTRUCTIVE SOLID-GEOMETRY METHODS

- Another technique for solid modeling is to combine the volumes occupied by overlapping three-dimensional objects using set operations.
- This modeling method, called constructive solid geometry (CSG), creates a new volume by applying the union, intersection, or difference operation to two specified volumes.
- A CSG application-starts with an initial set of three-dimensional objects (primitives), such as blocks, pyramids, cylinders, cones, spheres, and closed spline surfaces.
- The primitives can be provided by the CSG package as menu selections, or the primitives themselves could be formed using sweep methods, spline constructions, or other modeling procedures.
- To create a new three-dimensional shape using CSG methods, we-first select two primitives and drag them into position in some region of space.
- Then we select an operation (union, intersection, or difference) for combining the volumes of the two primitives.
- Now we have a new object, in addition to the primitives, that we can use to form other objects. We continue to construct new shapes, using combinations of primitives and the objects created at each step, until we have the final shape

Ray-casting

- Ray-casting methods are commonly used to implement constructive solid geometry operations when objects are described with boundary representations.
- we apply ray casting by constructing composite objects in world coordinates with the xy plane corresponding to the pixeI plane of a video monitor.
- This plane is then referred to as the "firing plane" since we fire a ray from each pixel position through the objects that are to be combined (Fig. 10- 58).
- We then determine surface intersections along each ray path, and sort the intersection points according to the distance from the firing The surface limits for the composite object are then determined by the specified set operation.



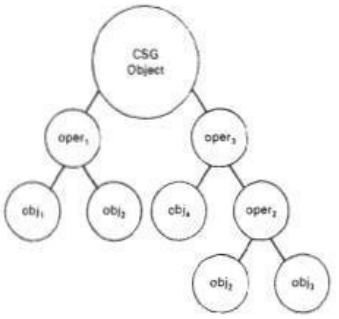


Figure 10-57
A CSG tree representation for an object.

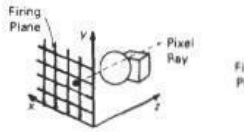
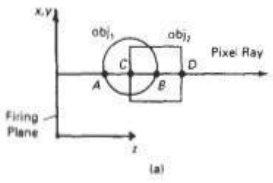


Figure 10-58
Implementing CSG operations using ray casting.



Operation	Surface Limits
Union	A, D
Intersection	C, B
Difference ($obj_2 - obj_1$)	B, D

(b)

Figure 10-59
Determining surface limits along a pixel ray.

OCTREES

- Hierarchical tree structures, called octrees, are used to represent solid objects in some graphics systems. Medical imaging and other applications that require displays of object cross sections commonly use octree representations.
- The tree structure is organized so that each node corresponds to a region of three-dimensional space. This representation for solids takes advantage of spatial coherence to reduce storage requirements for three-dimensional objects.
- It also provides a convenient representation for storing information about object interiors
- An octree encoding scheme divides regions of three-dimensional space (usually cubes) into octants and stores eight data elements in each node of the tree (Fig. 10-64).
- Individual elements of a three-dimensional space are called volume elements, or voxels. When all voxels in an octant are of the same type,the value is stored in the corresponding data element of the node.
- Empty regions of space are represented by voxel type "void." Any heterogeneous octant is subdivided into octants, and the corresponding data element in the node points to the next node in the octree.
- Procedures for generating octrees are similar to those for quadtrees.

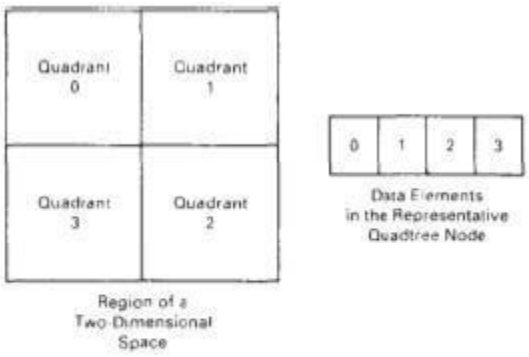


Figure 10-61
Region of a two-dimensional space divided into numbered quadrants and the associated quadtree node with four data elements.

Quadtrees

- The octree encoding procedure for a three-dimensional space is an extension of an encoding scheme for two-dimensional space, called quadtree encoding.
- Quadtrees are generated by successively dividing a two-dimensional region (usually a square) into quadrants.
- Each node in the quadtree has four data elements, one for each of the quadrants in the region (Fig. 10-61). If all pixels within a quadrant have the same color (a homogeneous quadrant), the corresponding data element in the node stores that color.
- In addition, a flag is set in the data element to indicate that the quadrant is homogeneous. Suppose all pixels in quadrant 2 of Fig. 10-61 are found to be red.

- The color code for red is then placed in data element 2 of the node.
- Otherwise, the quadrant is said to be heterogeneous, and that quadrant is itself divided into quadrants (Fig. 10-62).
- The corresponding data element in the node now flags the quadrant as heterogeneous and stores the pointer to the next node in the quadtrees.

ume elements, or **voxels**. When all voxels in an octant are of the same type, this

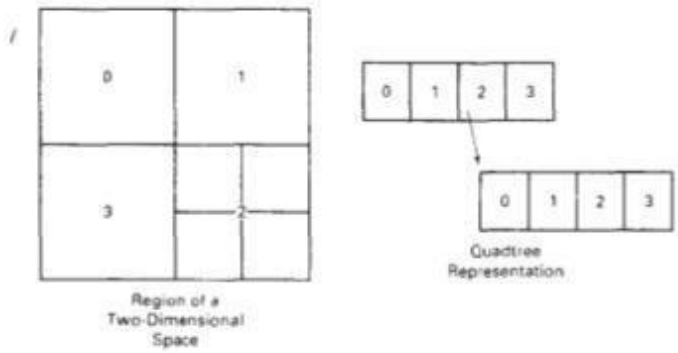


Figure 10-62
Region of a two-dimensional space with two levels of quadrant divisions and the associated quadtree representation

UNIT-5

COMPUTER GRAPHICS

DESIGN OF ANIMATION SEQUENCES

In general, an animation sequence is designed with the Following steps:

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

This standard approach for animated cartoons is applied to other animation applications as well, although there are many special applications that do not follow this sequence. Real-time computer animations produced by flight simulators, for instance, display motion sequences in response to settings on the aircraft controls .And visualization applications are generated by the solutions of the numerical models. For frame-by-frame animation, each frame of the scene is separately generated and stored. Later, the frames can be recoded on film or they can be consecutively displayed in "real-time playback" mode..

Storyboard layout

The storyboard is an outline of the action. It defines the motion sequence as 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 basic ideas for the motion.

Object definitions

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. In addition, the associated movements for each object **are** specified along with the shape

Key-frame specifications

A keyframe 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. More key frames are specified for intricate motions than for simple, slowly varying motions.

Generation of in-between frames

In-betweens 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. 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 can be duplicated. For a I-minute film sequence with no duplication, we would need 1440 frames. With five in-betweens for each pair of key frames, we would need 288 key frames. If the motion is not too complicated, we could space the key frames a little farther apart.

GENERAL COMPUTER-ANIMATION FUNCTIONS

Some steps in the development of an animation sequence are well-suited to computer solution. These include object manipulations and rendering, camera motions, and the generation of in-betweens. Animation packages, such as Wave front,

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

RASTER ANIMATIONS

On raster systems, we can generate real-time animation in limited applications using raster operations. Sequences of raster operations can be executed to produce real-time animation of either two-dimensional or three-dimensional objects, as long as we restrict the animation to motions in the projection plane. Then no viewing or visible- surface algorithms need be invoked. The animation is then accomplished by changing the color-table values so that the object is "on" at successively positions along the animation path as the preceding position is set-to the background intensity

On raster systems, we can generate real-time animation in limited applications using raster operations. As we have seen in Section 5-8, a simple method for translation in the xy plane is to transfer a rectangular block of pixel values from one location to another. Two dimensional rotations in multiples of 90" are also simple to perform, although we can rotate rectangular blocks of pixels through arbitrary angles using antialiasing procedures. To rotate a block of pixels, we need to determine the percent of area coverage for those pixels that overlap the rotated block. Sequences of raster operations can be executed to produce real-time animation of either two-dimensional or three-dimensional objects, as long as we restrict the animation to motions in the projection plane. Then no viewing or visible-surface algorithms need be invoked .We can also animate objects along two-dimensional motion paths using the color-table transformations.

COMPUTER-ANIMATION LANGUAGES

Design and control of animation sequences are handled with a set of animation Routines. **A** general-purpose language, such as C, Lisp, Pascal, or FORTRAN, is often used to program the animation functions, but several specialized animation languages have been developed. Animation functions include a graphics editor, a key-frame generator, an in-between generator, and standard graphics routines.

A typical task in an animation specification is ***scene description.*** This includes the positioning of objects and light sources, defining the photometric parameters (light-source intensities and surface-illumination properties), and setting the camera parameters (position, orientation, and lens characteristics). Another standard function is ***action specification.*** This involves the layout of motion paths for the objects and camera. And we need the usual graphics routines: viewing and perspective transformations, geometric transformations to generate object movements as a function of accelerations or kinematic path specifications, visible-surface identification, and the surface-rendering operations.

Key-frame systems :-are specialized animation languages designed simply to generate the in-betweens from the user-specified key frames. Usually, each object in the scene is defined as a set of rigid bodies connected

at the joints and with a limited number of degrees of freedom. As an example, the single-arr.1 robot in Fig. 16-4 has six degrees of freedom, which are called arm sweep, shoulder swivel, elbow extension, pitch, yaw, and roll. We can extend the number of degrees of freedom for this robot arm to nine by allowing three-dimensional translations for the base (Fig. 16-51. If we also allow base rotations, the robot arm can have a total of 12 degrees of freedom. The human body, in comparison, has over200 degrees of freedom.

Parameterized systems allow object-motion characteristics to be specified as part of the object definitions. The adjustable parameters control such object characteristics as degrees of freedom, motion limitations, and allowable shape changes.

Figure 16-4
Degrees of freedom for a stationary, single-arm robot

Scripting systems allow object specifications and animation sequences to be defined with a user input *script*. From the script, a library of various objects and motions can be constructed.

KEY-FRAME SYSTEMS

We generate each set **of** in-betweens from the specification of two (or more) key frames. Motion paths can be given with a ***kinematic description*** as a set of spline curves, or the motions can be ***physically based*** by specifying the forces acting on the objects to be animated. For complex scenes, we can separate the frames into individual componentsor objects called *cels (*celluloid transparencies), an acronym from cartoon animation. Given the animation paths, we can interpolate the positions of individual objects between any two times. 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.

Kinematics-The branch of mechanics concerned with the motion of objects without reference to the forces which cause the motion.

Kinematic description- With a kinematic description, we specify the animation by giving motion parameters position, velocity, and acceleration) without reference to the forces that cause the motion. for constant velocity (zero acceleration), we designate the motions of rigid bodies in a scene by giving an initial position and velocity vector for each objects.

MORPHING

Transformation of object shapes from one form to another is called morphing, which is a shortened Form of metamorphosis. Morphing methods can he applied to any motion or transition involving a change in shape.

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. This preprocessing step is illustrated in Fig. 16-6. 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 n 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. 16-7. An example of a triangle linearly expanding into a quadrilaterals given In Fig. 16-8. Figures 16-9 and 16-10 show examples of morphing in television advertising.

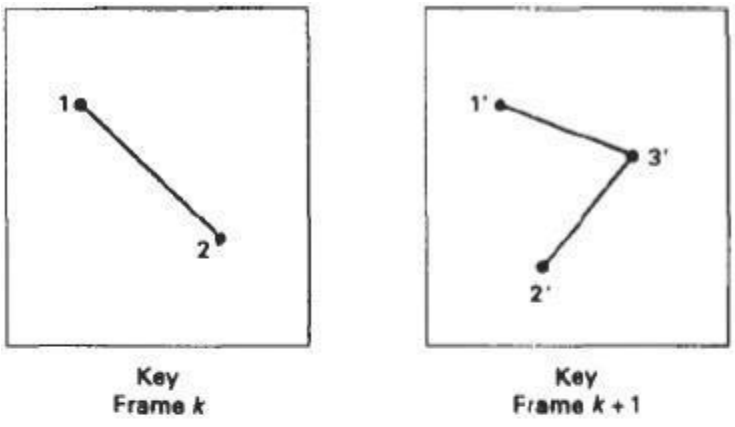
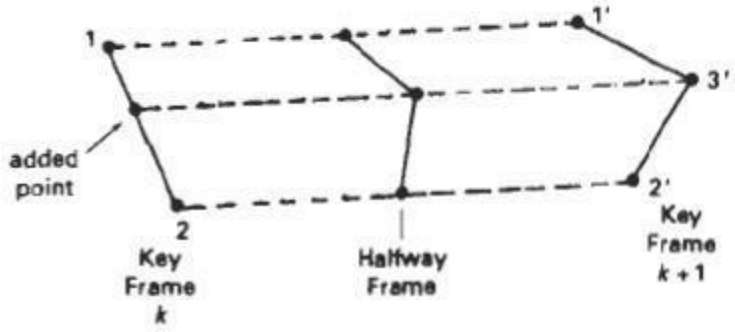


Figure 16-6
An edge with vertex positions 1 and 2 in key frame k evolves into two connected edges in key frame $k + 1$.



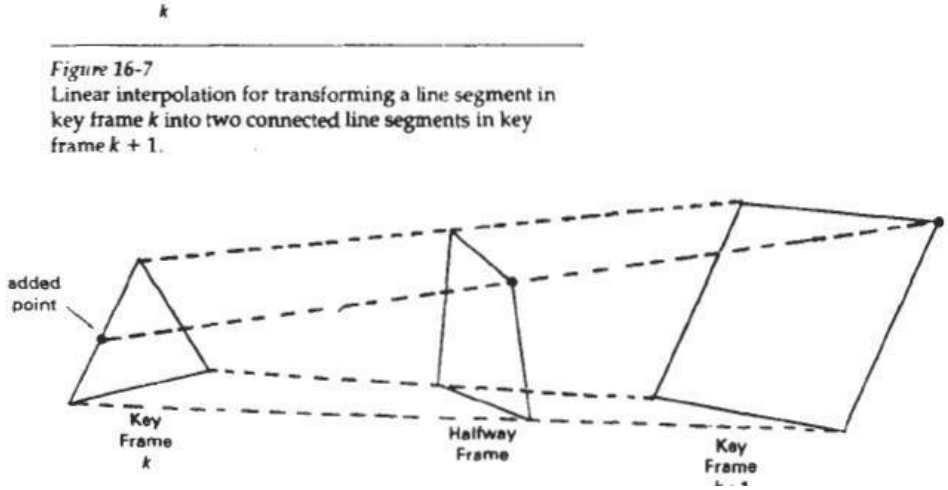


Figure 16-7
Linear interpolation for transforming a line segment in key frame k into two connected line segments in key frame $k + 1$.

Figure 16-8
Linear interpolation for transforming a triangle into a quadrilateral.

MOTION SPECIFICATIONS

There are several ways in which the motions of objects can be specified in an animation system. We can define motions in very explicit terms, or we can use more abstract or more general approaches.

Direct Motion Specification

The most straightforward method for defining a motion sequence is ***direct*** specification of the motion parameters. Here, we explicitly give the rotation angles and translation vectors. Then the geometric transformation matrices are applied to transform coordinate positions. Alternatively, we could use an approximating

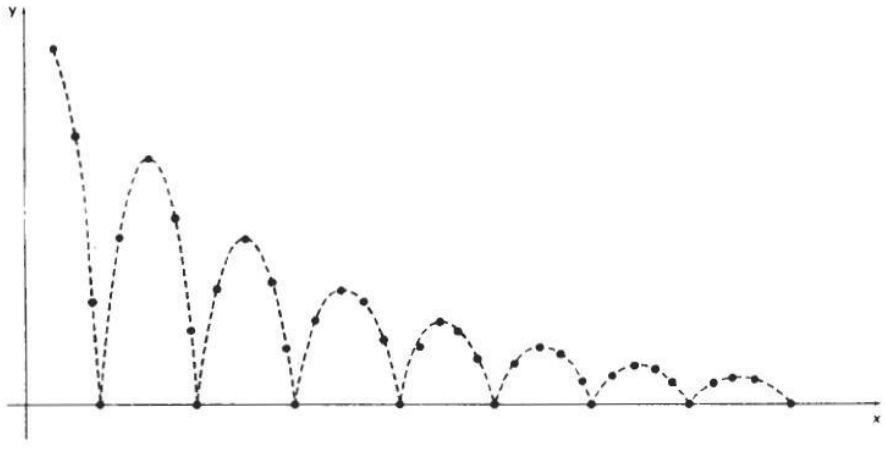


Figure 16-16
Approximating the motion of a bouncing ball with a damped *sine* function (Eq. 16-10).

equation to specify certain kinds of motions. We can approximate the path of a bouncing ball, for instance, with a damped, rectified, ***since*** curve

$$y(x) = A\,|\sin(\omega x + \theta_0)|\,e^{-kx}$$

:where ***A*** is the initial amplitude, ***w*** is the angular frequence , θ_0 **,** is the phase angle ,and k is the damping constant. These methods can be used for simple user-programmed animation sequences.

Goal-Directed Systems

At the opposite extreme, we can specify the motions that are to take place in general terms that abstractly describe the actions. These systems are referred to as ***goal directed*** because they determine specific motion parameters given the goals of the animation. For example, we could specify that we want an object to "walk" or to "run" to a particular destination. Or we could state that we want an object to "pick up" some other specified object. The input directives are then interpreted in terms of component motions that will accomplish the selected task.

Human motions, for instance, can be defined as a hierarchical structure of sub motions for the torso, limbs, and so forth.

Kinematics and **Dynamics**

We can also construct animation sequences using ***kinematic*** or **dynamic** descriptions .With a kinematic description, we specify the animation by giving motion parameters (position, velocity, and acceleration) without reference to the forcesthat cause the motion. For constant velocity (zero acceleration), we designate the motions of rigid bodies in a scene by giving an initial position and velocity vector. If we also specify accelerations (rate of change of velocity), we can generate speed-ups, slowdowns, and curved motion paths. Kinematic specification of a motion can also be given by simply describing the motion path. This is often done using spline curves.

An alternate approach is to use inverse kinematics Here, we specify the initial and final positions of objects at specified times and the motion parameters are computed by the system. For example, assuming zero accelerations, we can determine the constant velocity that will accomplish the movement of an object from the initial position to the final position. This method is often used with complex objects by giving the positions and orientations of an end node of an object, such as a hand or a foot.

Dynamic descriptions on the other hand, require the specification of the Forces that produce the velocities and accelerations. Descriptions of object behavior under the influence of forces are generally referred to as a physically based modeling .Examples of forces affecting object motion include electro magnetic, gravitational, friction, and other mechanical forces.

Object motions are obtained from the force equations describing physical Laws , such as Newton's laws of motion for gravitational and friction processes,

$$\mathbf{F} = \frac{d}{dt}(m\mathbf{v})$$

with F as the force vector, and v as the velocity vector If mass is constant, we solve the equation F = ma, where a is the acceleration vector.