# Unit 7
# Computer Animation and Visualization

# Introduction

- Some typical applications of computer-generated animation are entertainment (motion pictures and cartoons), Adventism, scientific and engineering studies, and training and education.
- Although we tend to think of animation as implying object motions, 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.
- Advertising animations often transition one object shape into another: for example, transforming a can of motor oil into an automobile engine.
- Computer animations can also be generated by changing camera parameters, such as position, orientation, and focal length.
- And we can produce computer animations by charging lighting effects or other parameters and procedures associated with illumination and rendering.

- Many applications of computer animation require realistic displays.
- An accurate representation of the shape of a thunderstorm or other natural phenomena described with a numerical model is important for evaluating the reliability of the model.
- Also, simulators for training aircraft pilots and heavy-equipment operators must produce reasonably accurate representations of the environment.
- Entertainment and advertising applications, on the other hand, are sometimes more interested in visual effects.
- Thus, scenes may be displayed with exaggerated shapes and unrealistic motions and transformations.
- There are many entertainment and advertising applications that do require accurate representations for computer-generated scenes.
- And in some scientific and engineering studies, realism is not a goal. For example, physical quantities are often displayed with pseudo-colors or abstract shapes that change over time to help the researcher understand the nature of the physical process.

# 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 ap plications as well, although there are many special applications that do not follow this sequence.
- Real-time computer animations produced by Bight simulators, for instance, display motion sequences in response to setting 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.

- 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 or it could be a list of the basic ideas for the motion.
- 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.
- 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.

- ***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 1-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.
- There are several other tasks that may be required, depending on the application.
- They include motion verification, editing, and production and synchronization of a soundtrack.
- Many of the functions needed to produce general animations are now computer-generated

# GENERAL COMPUTER-ANIMATION FUNCTIONS

- Some steps in the development of an animation sequence are well-suited to computers solution.
- These include  object manipulations and rendering, camera motions, and the generation of in-betweens.
- Animation packages, such as Wavefront, for example, provide special functions for designing the animation and processing individual objects.
- 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.
- Standard functions can then be applied to identify visible surfaces and apply the rendering algorithms.
- Another typical function simulates camera movements.
- Standard motions are zooming, panning, and tilting.
- Finally, given the specification for the key frames, the in-between can be automatically generated

# RASTER ANIMATIONS

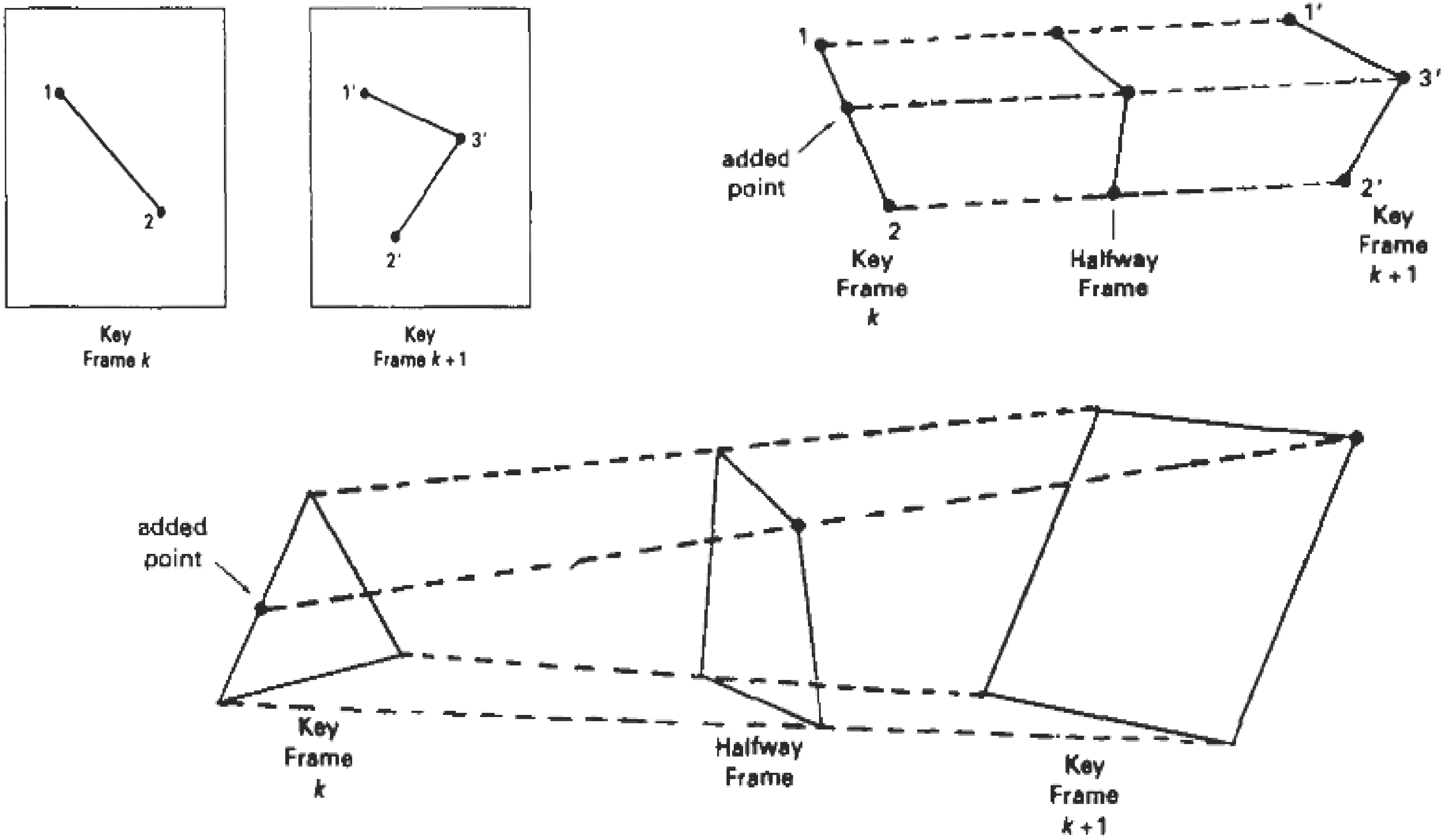
- We can generate real-time animation in limited applications using *raster operations.*
- *A* simple method for translation in the **xy** plane is to transfer a rectangular block of pixel values from one location to another.
- Two dimensional rotations in multiples of ***90º*** are also simple to perform, 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.*
- Here we predefine the object at successive positions along the motion path, and set the successive blocks of pixel values to color-table entries.
- We set the pixels at the first position of the object to "on" values, and we set the pixels at the other object positions to the background color.
- 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

# KEY - FRAME SYSTEMS

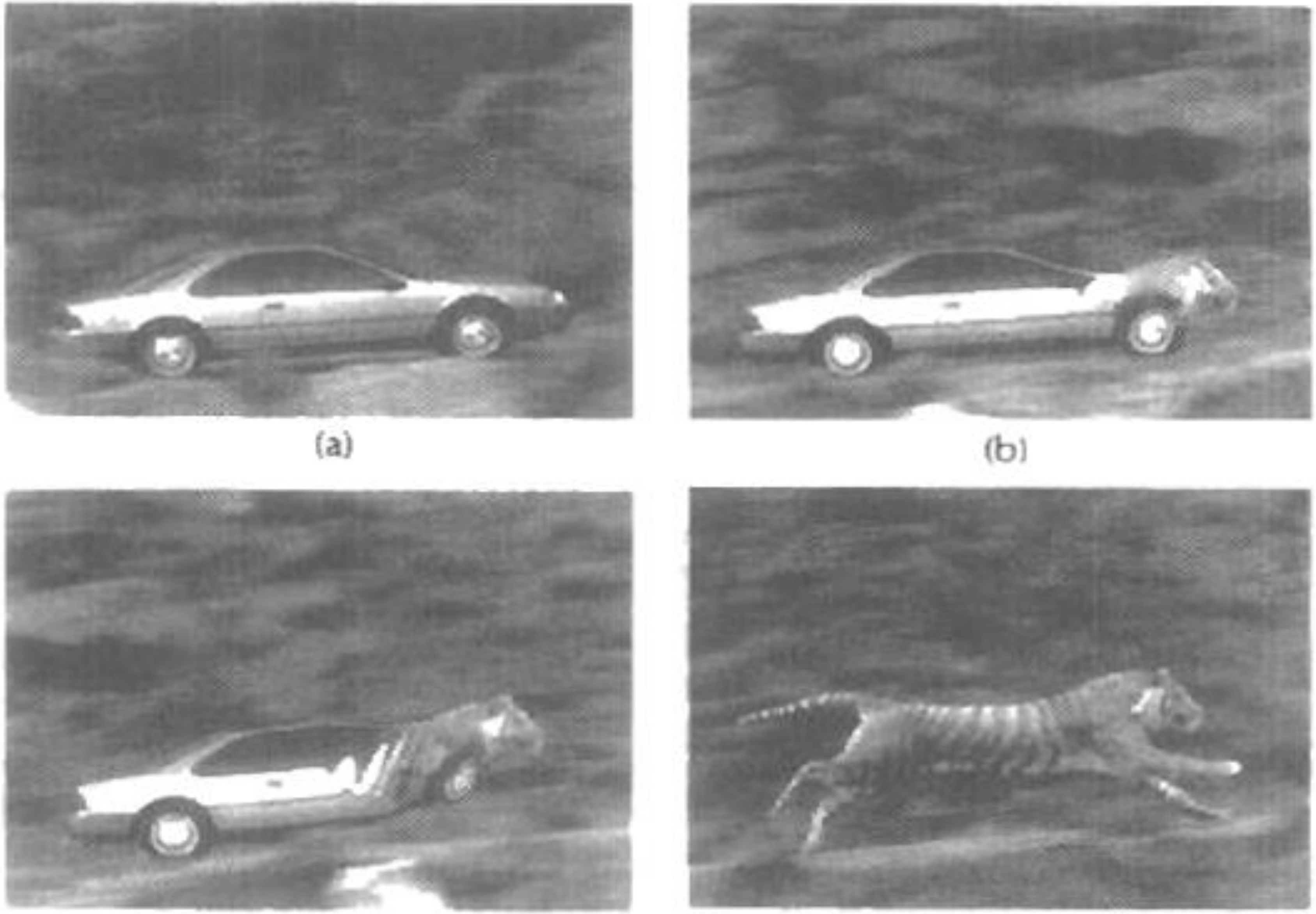
- We generate each set of in-betweens from the specification of two (or more) key frames.
- Motion paths can be given with a *kinematic descriptions as* a set of spline curves, or the motions can be *physically based* by specifying the forces acting on the objects to be animated.
- For complex scenes,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

# 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 Figure.



1
2
Key
Frame k
1'
3'
2'
Key
Frame k + 1
1
added
point
2
Key
Frame
k
Halfway
Frame
1'
3'
2'
Key
Frame
k + 1
added
point
Key
Frame
k
Halfway
Frame
Key
Frame
k + 1



(a) (b) (c) (d)

# 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.

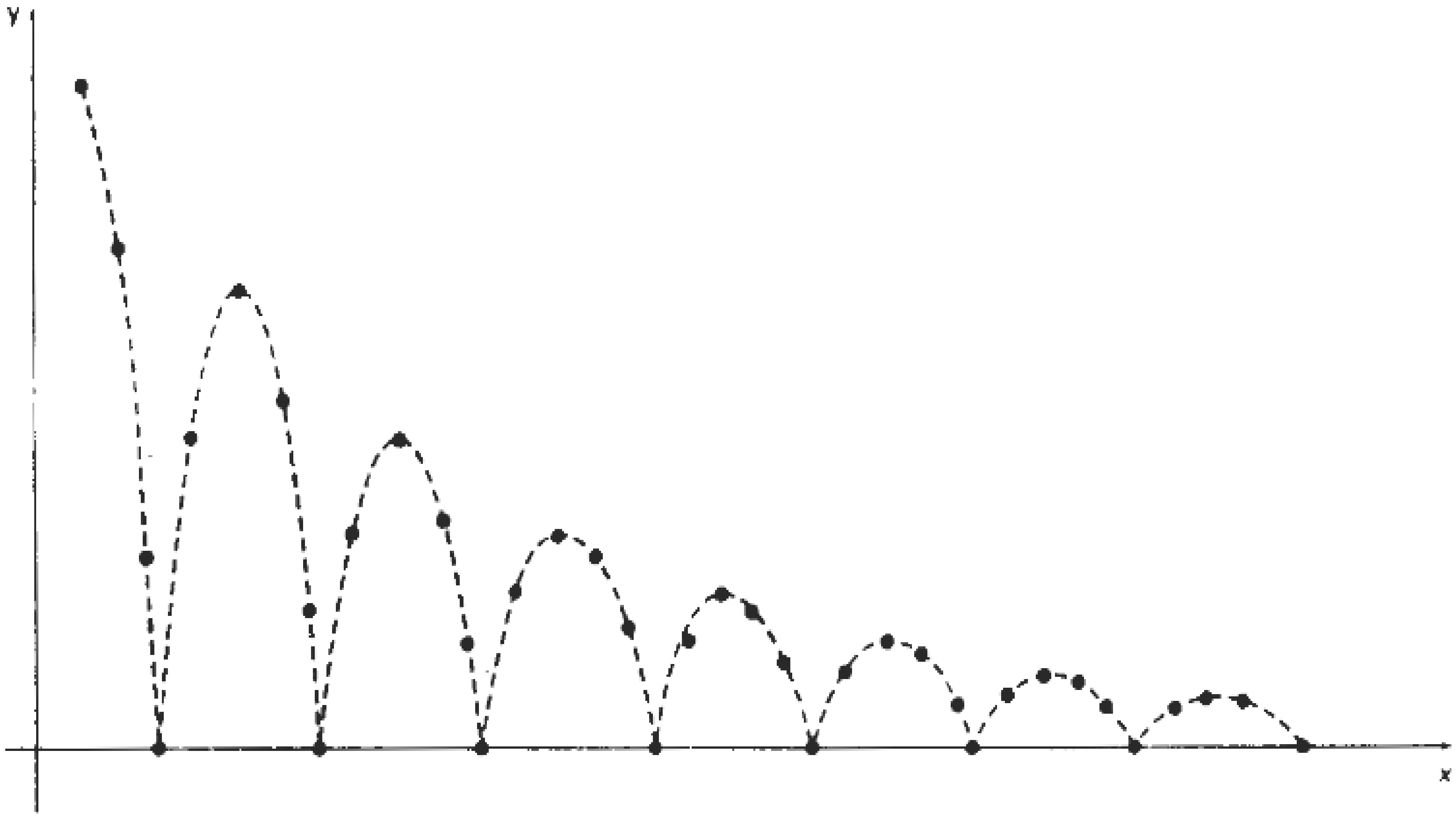
**1. Direct Motion Specification**

- 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 equation to specify certain kinds of motions.
- We can approximate the path of a bouncing ball, for instance, with a damped, rectified, ***sine*** curve (Figure***):***

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

where ***A*** is the initial amplitude, ***w*** is the angular frequence, **θo,** is the phase angle, and **k** is the damping constant.

- These methods can be used for simple user-programmed animation sequences.



y
x

# 2. 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.

# 3. 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 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 object.
- As an example, if a velocity is specified as **(3,0, - 4 )** km/sec, then this vector gives the direction for the straight-line motion path and the speed (magnitude of velocity) is **5 km/sec**.
- 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.
- The system then determines the motion parameters of other nodes to accomplish the desired motion.
- 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 electromagnetic, 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, Euler or Navier-Stokes equations describing fluid flow, and Maxwell's equations for electromagnetic forces.
- For example, the general form of Newton's second law for a particle pf mass ***m*** is

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

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.
- Otherwise, mass is a function of time, as in relativistic motions or the motions of space vehicles that consume measurable amounts of fuel per unit time.
- We can also use inverse dynamics to obtain the forces, given the initial and final positions of objects and the type of motion.

# Design of Animation Sequences

- The design of an animation sequence involves multiple stages, from conceptualization to final rendering. A well-structured animation sequence ensures smooth, realistic motion and engaging visuals. The key steps in designing an animation sequence include:

**1. Storyboarding**

- A sequence of hand-drawn or digital sketches representing key moments of the animation.
- Helps visualize the flow, timing, and composition of each scene.
- Includes camera angles, character movements, and scene transitions.

**2. Layout Design**

- Establishes the background, scene composition, and positioning of characters.
- Defines the camera angles, lighting, and perspective for each shot.
- Helps animators plan character interactions within the environment.

**3. Keyframe Animation**

- **Keyframes** define important positions or poses of characters and objects at specific time intervals.
- **In-between Frames (Tweening):** Automatically generated frames between key poses for smooth motion.
- **Easing and Interpolation:** Controls acceleration and deceleration for natural movement.

## 4. Motion Path and Timing

- Objects and characters follow predefined motion paths to ensure realistic movement.
- Timing charts help define the speed and pacing of actions.
- Motion curves (Bezier or spline) refine acceleration and deceleration.

## 5. Physics-Based Animation (If Required)

- Uses kinematics and dynamics to simulate real-world physics.
- Includes gravity, collisions, fluid simulations, and cloth physics.
- Enhances realism in animations such as character movements, water, and fire effects.

**6. Rendering and Lighting**

- Texturing and shading are applied to enhance visual appeal.
- Lighting effects (global illumination, ray tracing) create realistic shadows and reflections.
- Rendering converts 3D models into final animation frames.

**7. Post-Processing and Compositing**

- Adds visual effects (VFX), color grading, and motion blur.
- Compositing integrates multiple layers (background, characters, effects) into a seamless scene.
- Enhances the overall look with depth of field, glow, and particle effects.

**8. Exporting and Final Output**

- The final animation sequence is exported in the required format (MP4, AVI, GIF, etc.).
- Frame rate (FPS) is adjusted for smooth playback (24 FPS for cinematic, 30/60 FPS for games).
- The animation is reviewed and refined based on feedback before release.