UNIT 5

Animation: Conventional and Computer-Assisted Animation, Animation Languages, Methods of Controlling Animation, Basic Rules of Animation, Problems Peculiar to Animation.

Animating the Virtual Environment: The dynamics of numbers, Linear interpolation, Non-linear interpolation, parametric interpolation.

The animation of objects: Linear translation, Non-linear translation, Linear and Non-linear angular rotation. Shape, object parametric line/surface patch Inbetweening. Free form deformation, Particle systems. Physics based modeling and simulation.

Animation

Conventional animation

Conventional animation, also called key-frame animation because of the use of key frames and inbetweening. This animation is created using a fixed sequence called a storyboard. A storyboard is an outline of an animation.

- It contains a high-level sequence of sketches showing structure and ideas of the animation, soundtracks which are recorded etc.
- A detailed layout which is mostly drawn, is then correlated with the soundtrack. Next, keyframes containing the characters being animated are drawn, the characters are at the extreme or characteristic positions, from which intermediate positions are inferred and can be filled in(inbetweening), forming a trial film(pencil test).
- The pencil test frames are the transferred to cels(acetate films) either by hand copying or photocopying.

In multiplane animation, multiple layers of cels are used, some for background which remain almost constant and some for foreground which change over time.

- The entire process of producing an animation is supposed to be sequential, but is often (especially when done with computers) iterative: the available sound effects may cause the storyboard to be modified slightly, the eventual look of the animation may require that some sequences be expanded, in turn requiring new sound-track segments, and so on.

The people producing the animation have quite distinct roles:

- some design the sequence
- some draw key frames
- some are strictly inbetweeners
- some work only on painting the final cels

Two examples of films using computer-assisted animation are *Beauty and the Beast* and *Antz*.

Computer Assisted animation

Computer assisted animation is basically using new technologies to cut down the time scale that traditional animation could take, but still having the elements of traditional drawings of characters or objects.

Many stages of conventional animation seem ideally suited to computer assistance, especially inbetweening and coloring, which can be done using the seed-fill techniques. Before the computer can be used, however, the drawings must be digitized. This can be done by:

- using optical scanning
- by tracing the drawings with a data tablet
- by producing the original drawings with a drawing program in the first place.

The drawings may need to be post processed (e.g., filtered) to clean up any glitches arising from the input process (especially optical scanning), and to smooth the contours somewhat. The composition stage, in which foreground and background figures are combined to generate the individual frames for the final animation, can be done with the image-composition techniques. Three examples of computer-generated animation movies are *Toy Story*, *The Incredibles* and *Shrek*.

DIFFERENCE BETWEEN CONVENTIONAL AND COMPUTER ASSISTED ANIMATION

Conventional animation

1. Conventional Assisted animation is usually classed as two-dimensional (2D) animation.
2. Drawings are either hand drawn (pencil to paper) or interactively drawn(drawn on the computer) using different assisting appliances and are positioned into specific software packages.
3. Not limited by available technology and computing techniques
4. Still faster than computer animation for high quality
5. Complete control of drawing and motion, limited only by artistic ability (and patience)
6. Draw each frame of animation with great control but is tedious.

Computer Animation

1. Computer Assisted animation is known as 3-dimensional (3D) animation. Creators will design an object or character with an X,Y and Z axis.
2. Unlike the traditional way of animation no pencil to paper drawings create the way computer generated animation works.
3. Many frames can be calculated instead of drawn
4. Many scenarios/variations can be quickly tried out
5. Complex 3D models: don't have to draw different views
6. Fewer tedious steps

Animation Languages:

1. Linear List Notations: It is the specially animation supporting language. Each event in the animation is described by start and ending frame number and an action that is to take place (event).
2. General Purpose Language: Another way to describe animation is to embed animation capability within a general purpose programming language. The high level computer languages which are developed for the normal application software development also have the animation supporting features along with graphics drawing.
3. Graphical Languages: Graphical animation languages describe animation in a more visual way. These languages are used for expressing, editing, and comprehending the simultaneous changes taking place in an animation.

Methods of Controlling Animation:

1. Full explicit or explicitly declared control: This is the simplest type of control. In this, the animator describes all the events which could occur in an animation. All events in an animation can be done at the following levels:
    a. Object level by specifying simple transformations (translations, rotations, scaling) to objects.
    b. Frame level by specifying key frames and methods for interpolating between them.
2. Procedural control: In the Procedural method, a set of rules are used to animate the objects. Animator defines or specifies the initial rules, procedure to process and later runs simulations. Rules are often based on physical rules of the real world expressed by mathematical equations
3. Constraint based control: This method specifies an animation sequence which needs to be followed. For instance, many objects move in straight lines but there could be many other objects whose movement is determined by the objects with which they come in contact whose compound movement may or may not be linear, these objects act as constraints that control the movement of the former object.
4. Control by tracking live action: Control is achieved by examining the motions of objects in the real world wherein live action can be tracked to generate trajectories of objects in the course of an animation. Rotoscoping is a technique where animators trace live action movement, frame by frame, for use in animated films. Pre-recorded live-film images were projected onto a frosted glass panel and redrawn by an animator.
5. Actors: The use of actors is a high-level form of procedural control. An actor in an animation is a small program invoked once per frame to determine the characteristics of some object in the animation.
6. Kinematics and Dynamics control: Kinematics refer to the position and velocity of points. Dynamics takes into account the physical laws that govern kinematics. Unlike other methods, this method makes use of the laws of physics to generate motion of pictures and other objects.
7. Physically Based Animation: Dynamics are examples of physically based animations. Animation of motion of physically based models of cloth, plasticity, and rigid-body are described using simulations of the evolution of physical systems. Various formulations of classical mechanical behavior have been developed, they all represent the evolution of a physical system as a solution to a system of partial differential equations.

Basic Rules of Animation: There are 12 basic principles of animation, they are:-

1. Squash and stretch: This is the most important principle of animation, it gives the sense of weight and volume to draw an object. Squashing and stretching are the two most basic animated reactions a drawn object can exhibit and they applied to everything like a simple bouncing ball. In more realistic animations, squash and stretch are linked, meaning that if an object is stretched vertically, it's squashed horizontally.

2. Anticipation: In this principle animator will create a starting scene like that it shows that something will happen, almost nothing happens suddenly. In 2D animation, the anticipation of a movement is just as important as the movement itself.

3. Flow through and overlapping action: Two object's action have different speed in any scene can easily describe this principle. Follow through means that a part of the object will keep moving for a while even if the base or centre of mass of object has stopped..

4. Slow-out and Slow-in: The smoothness of an animation is governed mostly by how many frames the animation contains. More frames means smoother, slower animation, fewer frames will speed up the animation. Slow-Ins and Slow-Outs make use of this relationship by adding more frames at the beginning or end of an action respectively

5 Arcs: Natural movement tends to follow the basic trajectory of an arc, as such, it is important for animators to consider implied "arcs" for each movement. The faster an object is moving, the more subtle it's arc. Making the proper use of arcs absolutely imperative for expressing speed.

7. Secondary action: The addition of secondary action can make a scene more interesting to the viewer and further help enforce the illusion of reality. Typically used to convey strong emotions, secondary action is an integral part of creating engaging animations that are more than just one moving part.

8. Timing: Timing, or the number of frames in a given scene or action, dictates the overall speed of the animation. Having the right amount of frames in an action is integral to creating a scene that looks like everything is happening at the same time.

9. Exaggeration: This principle creates extra reality in the scene by developing a proper animation style. Exaggeration remains true to reality but presented in a wilder and more extreme form.

10. Solid Drawing: Solid drawing is a principle that applies to objects drawn in three a three dimensional space. Animation that takes place on a 2D plane is often insufficient to realize the full motion of an action..

.

11. <u>Appeal:</u> Appeal in a cartoon character corresponds to what would be called charm in an actor. A character who is appealing is not necessarily sympathetic. Villains or monsters can also be appealing but keeping them interesting is key.

Problems peculiar to Animation:

There are many variations and problems in animation fields. What may seem like a good approach frequently requires some preparation or fix up. In order to evaluate any approach we must introduce the concept of "touch-time." Touch-time is the amount of time that some operator or artist must spend working on a figure in some process. The total touch-time would then be the total of the times spent on all steps of the pipeline. An analysis then requires that we find the average total touch-time per frame. If the computer can perform in between then the average total touch-time goes down. If any method requires some touch up, we must include the touch-up time. The processes included are tracing or figure entry, scanning, coloring or painting, error correction, etc.

Solutions to the various problems of computer assisted character animation should include analyses of the ramifications of that solution. The full extent of any ramifications cannot be fully understood until the method is tried in the environment for which it is intended. Extrapolation of results will yield incorrect conclusions.

Another part of any analysis should be a statement regarding the class of characters to be handled. The ease with which any algorithm can be used is dependent on the complexity of characters and the quality of movement. It is necessary to understand any restrictions on characters before one can evaluate the approach used to animate them.

New problems and challenges arise with the addition of a new dimension(like, 2D to 3D). One such problem is temporal aliasing. Just as aliasing problems in 2D and 3D graphics are partially solved by increasing the screen resolution, the temporal aliasing problems in animation can be partially solved by increasing temporal resolution.

High temporal resolution (many frames per second) may seem unnecessary. After all, video motion seems smooth, and it is achieved at only 30 fps. Movies, however, at 24 fps, often have a jerkiness about them, especially when large objects are moving fast close to the viewer, as sometimes happens in a panning action.Another problem in 4D rendering is the requirement that we render many very similar images. This problem is a lot like that of rendering multiple scan lines in a 2D image: each scan line, on the average, looks a lot like the one above it.

The dynamics of numbers: As a virtual environment is nothing more than a complex numerical database. Animating any one of the attributes either physical or abstract involves changing one number into another. The numerical envelope of the change is important so too are the derivatives at any point in the operation. Physical based motion can be emulated by evaluating the relevant equations but very often realistic motion can be simulated by numerical interpolation. Example - To simulate a door opening or closing the angle of rotation has to be computed at various instances of time

Linear interpolation: Linear interpolation is a method of curve fitting using linear polynomials to construct new data points within the range of a discrete set of known data points.Linear interpolation on a set of data points $(x_0, y_0), (x_1, y_1), ..., (x_n, y_n)$ is defined as the concatenation of linear interpolants between each pair of data points. This results in a continuous curve, with a discontinuous derivative (in general), Linear interpolation is useful when looking for a value between given data points. The strategy for linear interpolation is to use a straight line to connect the known data points on either side of the unknown point.

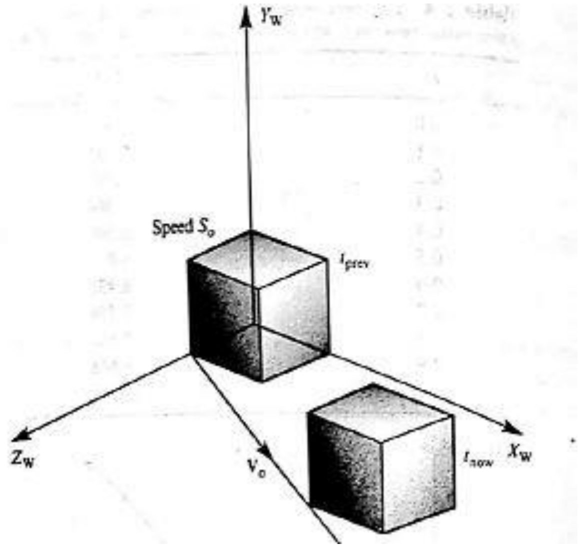
Non- Linear Interpolation: Non-Linear interpolation is used to make a smooth curve between two or points, though it is slightly more complicated. Functions must be found to represent a smooth curve

Parametric Interpolation: Parametric interpolation provides a wide range of controllable functions without requiring excessive amounts of computation. They involve the basis functions used for Bezier space curves.

1. **Quadratic interpolation:** A pair of numbers v1 and v2 can be linearly interpolated as follows: v(t) = (1-t)v1 + tv2   for 0 <= t <= 1. This can be extended into quadratic interpolation by introducing a control value vc that determines the function’s symmetry. The quadratic form is given as: v(t) = (1-t)^2v1 + 2t(1-t)vc + t^2v2 for 0<=t<=1
2. **Cubic interpolation:** Interpolation can be done between two numbers v1 and v2 using two control values as follows: v(t) = (1-t)^3 + 3t(1-t)^2vc + 3t^2(1-t)vd + t^3v2 for 0<=t<=1 where vc and vd are control values. The same expression is used for computing cubic Bezier space curves and surface patches. If vc=2 and vd=8 the interpolation is cubic and symmetric about t=0.5..
3. **Hermite interpolation:** Hermite interpolation is useful when specific rates of change are required at the start end of a transition. The algorithm requires the start and finish values together with their derivatives at these points.

Linear Translation:

- Frames per second is an important measure in any animation. If a system has an update rate of F Hz, then frames are separated by 1/F time.
- To show a virtual object moving at a speed S, it must be displaced a linear distance St between each update. But we cannot rely upon a consistent F, a real-time system clock must be used to determine the actual time that has elapsed since the last update. t must be derived by subtracting the previous time, and the actual time.



The object must be modified by the corresponding components of the vector Vo. The velocity Vo can be defined as the scalar. To contain the object within specified boundaries, we can automatically compute the object's new velocity after bouncing off the boundary. We can see that eventually the point will collide with the right-hand boundary at P.

Non Linear Translation:

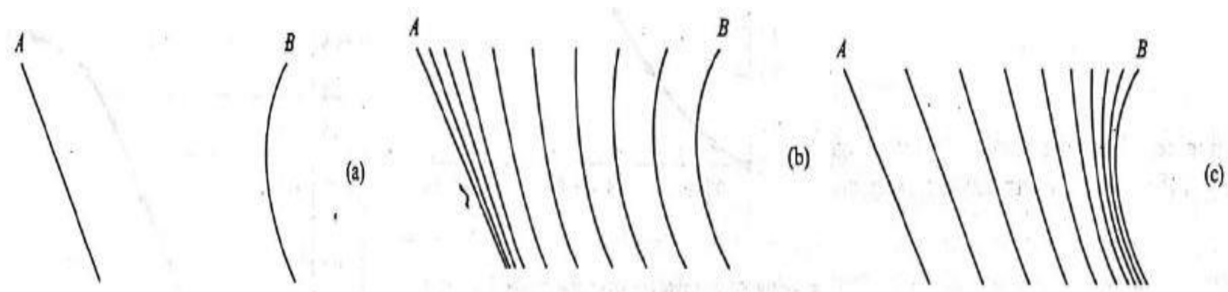
We now deal with cubic interpolation. Say we had to move an object along the X axis to a point, make it stay there for 1 second, and move it back to the origin after 2 seconds, we would need to formulate this as a function of time. t = T-T1 and t = (T-T1-1)/2 would hence be the two interpolations. Two control values Ca and Cb are used to control non-linearity in the forward motion, and another two to control the backward. These values are responsible to make certain of a smooth acceleration and deceleration at the start and end of each movement. To define the X translations, interpolants must be used.

Linear Angular rotation

XYZ fixed angles achieve a compound rotation by applying 3 fixed rotations about the fixed axes; whereas XYZ euler angles about an axial system that rotates with the object. If an object needs to be rotated about its local Y axis, and twice about the x axis, then a raw and pitch rotation can be shown.

Shape inbetweening

One of the techniques used to speed up the process of creating artwork in the cartoon industry is that of inbetweening. This is a process where 'inbetween' drawings are derived from two key images drawn by a skilled animator. These key images may show an object or character in two different positions, called key frames, which require extra inbetween images if they are to animate smoothly. Rather than give this task to the animator, the extra drawings are produced by an 'inbetweener' who prepares them by interpolating between the two key drawings. The number and positions of these interpolated images determine the dynamics of the final animation.



.

Object inbetweening

By inbetweening the z-coordinate, the above technique can be applied to 3D contours. However, 3D object descriptions are not arbitrary collections of contours: they are constructed from a coherent set of surface elements, and we cannot expect to interpolate between two different complex objects and expect that the inbetween objects are geometrically consistent.

Parametric line inbetweening

Although the technique of Shape inbetweening has an immediate application to explicit shapes and objects, there is no reason why it should not be applied to the control points of implicit parametric lines. For instance, in the case of a quadratic Bézier curve, the two end points and associated control vertex could be inbetweened with a similar set belonging to a second curve. More complex shapes can be processed by employing piecewise curves, and although curves with identical numbers of line segments are easier to process, dissimilar curves can be processed by introducing phantom segments of zero length.

Parametric surface patch inbetweening

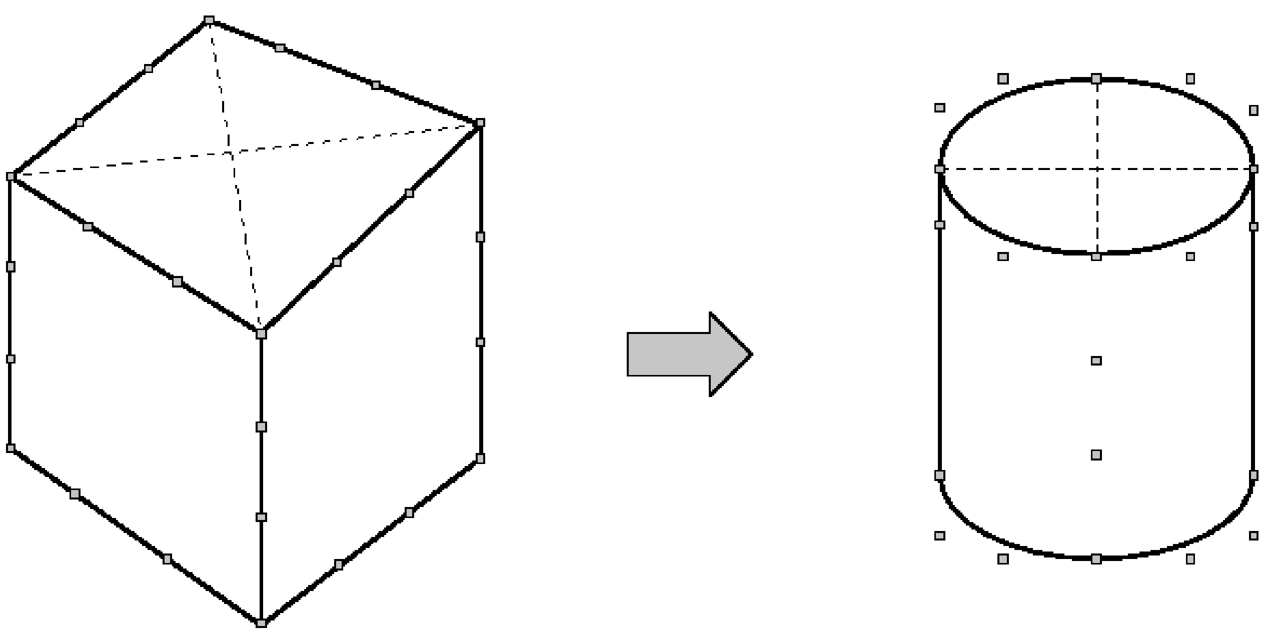
The elegance of parametric techniques enables line-based procedures to be extended to incorporate surface patch descriptions. In the case of Bézier surface patches, two sets of control meshes are inbetweened, which involves two sets of nine control points for quadratic patches, or sixteen control points for cubic patches. Even multi-patched objects can be inbetweened, but a consistent level of patch correspondence must be maintained between the two key objects.

Free Form Deformation: Sederberg and Parry introduced the concept of a free-form deformation in []. This technique defines a free-form deformation of space by specifying a trivariate Bézier solid, which acts on a parallelpiped region of space.

In general, an object can be deformed by deforming all the points that are contained in the object. Usually this is an impossible task and thus we normally strive to determine a finite set of points that can be deformed (the B-rep model) and then attempt to reconstruct the deformed B-rep from the finite set of deformed points. In general, the accepted method to utilize Sederberg/Parry deformations is to take an object and tesselate it into small polygons. The vertices of these polygons are then deformed and the resulting polygons are drawn on a graphics device.

The Sederberg-Parry deformation method has had wide use in the computer graphics/modeling/animation fields (see []). The simplicity of the method is that it utilizes parallelepipedidal regions of 3-dimensional space, resulting in a simple calculation for the (*u*,*v*,*w*) coordinate of any point and a direct substitution into a trivariate Bézier function.

For example, a cylindrical lattice can be formed by expanding the control points of a standard lattice representing a cube to approximate the sides by Bézier representations of a circle. The figure below illustrates this process



Particle Systems: The term particle system is loosely defined in computer graphics. It has been used to describe modeling techniques, rendering techniques, and even types of animation. In fact, the definition of a particle system seems to depend on the application that it is being used for. The criteria that hold true for all particle systems are the following:

- **Collection of particles** - A particle system is composed of one or more individual particles. Each of these particles has attributes that directly or indirectly effect the behavior of the particle or ultimately how and where the particle is rendered. Often, particles are graphical primitives such as points or lines, but they are not limited to this. Particle systems have also been used to represent complex group dynamics such as flocking birds.
- **Stochastically defined attributes** - The other common characteristic of all particle systems is the introduction of some type of random element. This random element can be used to control the particle attributes such as position, velocity and color. Usually the random element is controlled by some type of predefined stochastic limits, such as bounds, variance, or type of distribution.

Each object in Reeves particle system had the following attributes:

- **Position**
- **Velocity (speed and direction)**
- **Color**
- **Lifetime**
- **Age**
- **Shape**
- **Size**
- **Transparency**

Particle Life Cycle

Each particle goes through three distinct phases in the particle system: generation, dynamics, and death. These phases are described in more detail here:

- **Generation** - Particles in the system are generated randomly within a predetermined location of the fuzzy object. This space is termed the generation shape of the fuzzy object, and this generation shape may change over time. Each of the above mentioned attribute is given an initial value. These initial values may be fixed or may be determined by a stochastic process.
- **Particle Dynamics** - The attributes of each of the particles may vary over time. For example, the color of a particle in an explosion may get darker as it gets further from the center of the explosion, indicating that it is cooling off. In general, each of the particle attributes can be specified by a parametric equation with time as the parameter

Extinction - Each particle has two attributes dealing with length of existence: age and lifetime. Age is the time that the particle has been alive (measured in frames), this value is always initialized to 0 when the particle is created. Lifetime is the maximum amount of time that the particle can live (measured in frames).

Rendering: When rendering this system of thousands of particles, some assumptions have to be made to simplify the process. First, each particle is rendered to a small graphical primitive (blob). Particles that map to the same pixels in the image are additive - the color of a pixel is simply the sum of the color values of all the particles that map to it.

Physics based modelling and simulation

Physics-based modeling is a cross-disciplinary field, including elements of applied mathematics, numerical analysis, computational physics, computer graphics, computer vision, and software engineering. It has different ends than its parent, physics and applied mathematics, and it is somewhat different from its sibling fields, such as computational physics and classical computer graphics. Its long term goal is development methods enabling us to specify, design, build and control computational models of heterogeneous physical systems of objects. The term physics-based modeling has become a catch-all term for a variety of techniques that all share the approach of defining physical principles of behavior of their models. A physics-based model is a mathematical representation of an object (or its behavior) which incorporates physical characteristics such as forces,torques and energies into the model, allowing numerical simulation of its behavior. In computer graphics, common elements are classical dynamics with rigid or flexible bodies, inner-body interaction and constrained-based control.

An application of physics based modeling can be facial animation. Realistic facial animation is important in many graphics applications, like animated feature films and computer games, to enrich human computer interaction. In this paper, we propose a physically-based facial animation approach employing knowledge from the anatomy and biomechanics of human facial muscles. First, the 3D face mesh is generated automatically by commercial software and the facial skin is represented by a nonlinear mass-spring system which simulates realistic elastic dynamics of human dermis. Then, a structure skull is attached to fit the face mesh. A set of anatomically consistent facial muscles are incorporated to model the forces deforming the face mesh. Finally, we extend Waters' muscle model to improve the combination of multiple muscle actions and to generate realistic expression.