CS 231

Basics of Computer Animation

Animation Techniques

Keyframing
Motion capture
Physics models
Animation systems – animate what?

Particles

Linked Particles

Rigid bodies (RB)

Linked RB

Other - ?

Keyframe animation

Highest degree of control, also difficult
Interpolation affects end result
Timing must also be considered
Constraints can help but must be set up
The keyframing process

Iterate: adjust trajectory
play back motion

Depends on:
- interpolation between frames
- speed along path
- parameters for model

The basic animation engine

Motion generation

```pseudocode
frame = 0;
Until (forever){
    set scene(frame);
    take picture(frame);
    frame++;
}
```

Offline representation
```
mymovie.mpg = make_mpeg (all frames)
```
Keyframing – 3D characters

Complete control

Quality relies on skilled animator
Keyframed control parameters:

- location/orientation
- scale/shear
- joint angles
- shape (deformations)
- materials and textures
- camera parameters
- lighting (strobe, etc)

Keyframing – 3D characters

Representation

1) Skeleton
   Origin (root)
   Joint centers/
   bones lengths

2) Keyframes
   Pos/Rot Root (x)
   Joint Angles (q)
Kinematics – study of static movement

Forward: $x = f(\alpha, \beta)$

Inverse: $\alpha, \beta = f^{-1}(x)$

easy to compute
intuitive to control
positioning hand/feet
higher-level goals

Kinematics and IK - rely on skeleton motion
the basis of all human (and many non-human) characters

Kinematics solved directly
Used for character motion based on skeleton
Forward Kinematics

\[ x = l_1 \cos(\theta_1) + l_2 \cos(\theta_2) + l_3 \cos(\theta_3) \]
\[ y = l_1 \sin(\theta_1) + l_2 \sin(\theta_2) + l_3 \sin(\theta_3) \]

Hierarchical Animation

\[ x' = T_a R_a x \]
\[ x' = T_b R_b T_a R_a x \]
\[ \ldots \]
Keyframing skeleton with constraints

Joint limits
Pos limits
IK handles
Keyframe Interpolation:

Types of interpolation:

linear
b-spline
cardinal
Problems with interpolation curves

Note, non-uniform velocity

Keyframing in practice
Keyframing to control velocity
Velocity depends on interpolation style

- Linear
- Ease In/Ease out
Keyframing Ease-in/Ease-out

Sinusoidal
  - offset cos/sin
  - half period
  - piecewise

Function
  - cubic
  - constant acceleration

Ease-in/Ease-out (EIEO) Function

\[ s = \text{ease}(t) \]

(Normalized distance and time)
**EIEO : Sinusoidal**

\[ s = ease(t) = \left( \sin(t\pi - \pi / 2) + 1 \right) / 2 \]

**EIEO : Single Cubic**

\[ s = ease(t) = -2t^3 + 3t^2 \]
EIEO : Sinusoidal Piecewise

Provide linear (constant velocity) middle segment

\[
e_{\text{ase}(t)} = \begin{cases} 
\frac{2}{\pi} \left( \sin\left(\frac{t\pi}{2k_1} - \frac{\pi}{2}\right) / f \right) \\
\left( \frac{k_1}{\pi} - t - k_1 \right) / f \\
\left( \frac{k_1}{\pi} - k_1 + (1 - k_1) \frac{2}{\pi} \sin\left(\frac{\pi(t - k_1)}{2(1 - k_1)}\right) \right) / f \\
\end{cases}
\]

\[f = k_2 \frac{2}{\pi} + k_2 - k_1 + (1 - k_1) \frac{2}{\pi}\]
EIEO: Constant Acceleration

**Graph:**
- Distance vs. Time
- Time range: 0.2 to 1
- Distance range: 0 to 1

**Comparison:**
- Piecewise Sinusoid
- Constant Acceleration
Keyframing with Ease-in/Ease out

Action Transitions

Combining actions

Keyframing with Ease-in/Ease out
Keyframing with Ease-in/Ease out

Directing/initiating actions and their “transition”

![Diagram showing state and weight over time](image-url)
Rigging

Skeleton = “bones”
Geometry = “skin”
Rigging – set up

Hierarchy

Rigging

Rigid Bind:
Points on geometry get same transforms as bones

Smooth Bind:
Average weights of bones
Weights controlled by user or based on distance from joint
Rigging

Rigid Bind:
Points on geometry get same transforms as bones

Smooth Bind:
Average weights of bones
Weights controlled by user or based on distance from joint

Rigging - advanced

Other bindings
Muscles bulge
How to set up?

Automation
Compute skeleton from model

Issues
Joint distance problems
Creases
Keyframing

Complete control

Quality relies on skilled animator