CS 4732: 
Computer Animation

Key Frames & Shape Changes

Robert W. Lindeman
Associate Professor
Interactive Media & Game Development
Department of Computer Science
Worcester Polytechnic Institute

gogo@wpi.edu
Key Framing

- Key frames define important poses during an animation
  - Specified by animator
  - Computer fills in ‘tweens

- How should the computer interpolate as desired by the animator?
  - For curves?
  - For shapes?
Key Framing (cont.)

- Two main problems
  - Correspondence
  - Interpolation method

- For curves
  - One could require the curves to have the same number of control points
  - Interpolate control points, regenerate curve
Key Framing (cont.)

- For shapes
  - One could require the shapes to have the same topology
  - Apply physics-based simulation
  - Or use key framing for more control

- Shape animation used a lot for facial animations
Shape Animation

- What does it mean for shapes to be different?
  - Uniform scaling?
  - Is a square the same as a rectangle?

- Pulling/pushing vertices
  - Can be tedious/time consuming
  - Can displace neighbors by some influence function
2D Shape Deformation

- One approach to deformation is to:
  - Embed the vertices of the shape within a regular grid
  - Note the relative locations of the vertices to the grid intersections
  - Apply deformations to the grid
  - Recalculate the deformed vertex positions using bilinear interpolation

- Allows for more-efficient recalculation
- Makes it easier for the animator to specify deformation
2D Shape Deformation (cont.)

\[
P_{uv} = (1 - u)P_{00} + uP_{10} \\
P_{u1} = (1 - u)P_{01} + uP_{11} \\
P_{uv} = (1 - v)P_{u0} + vP_{u1} \\
= (1 - u)(1 - v)P_{00} + (1 - u)vP_{01} + u(1 - v)P_{10} + uvP_{11}
\]
Polyline Deformation

- Similar to grid deformation
- Calculate the relative position of the vertices to a polyline
- Good for snakes, tentacles, etc.
Polyline Deformation (cont.)

\[ r = \frac{d_2}{d_1} \]

\[ d_1 \]

\[ d_2 \]

\[ d \]

\[ L_1 \]

\[ L_2 \]

\[ L_3 \]
Free Form Deformation (FFD)

- Extend 2D technique to 3D
  - Use cubic (or other) interpolation instead of bilinear
  - Embed the shape within a grid defined by three axes
  - Record the locations of the vertices within the grid
  - Deform the grid
  - Calculate the new positions
Continuity can be insured in the same way as in 2D.

FFDs can be applied in sequence or hierarchically as well.

Animation can be carried out by:
- Moving the control points over time
- Moving the shape through the distorted space
- Can define a “tool” and deform shapes with it
FFD Animation

- Control points can be deformed based on:
  - Physical simulation, e.g., a ball hitting a sponge
  - Key framing
  - Facial simulation (e.g., bones, muscles, etc.)
  - Any function you can think of!
Sets of colinear control points

Common boundary plane

Bulging

Bending

R.W. Lindeman - WPI Dept. of Computer Science
Interactive Media & Game Development
FFD Animation (cont.)

“Tools” for distortion
FFD Animation (cont.)

Object traversing the logical FFD coordinate space

Object traversing the distorted space
Bones, Joints & Muscles

- How does all this relate to modern tools for animating figures?

- Bones as reference objects
  - Link length, joints, etc.
  - Bone movement constrained by muscles, etc.

- Skin defined in space as a spring-mass model

- Deformation is a combination of FFD & springs
Bones, Joints & Muscles (cont.)

- Bones move
- FFD lattice “anchored” to bone
- Spring-mass model moves FFD control points
3D Morphing

- Smoothly change one 3D shape into another

- Two main approaches
  - Volume based
  - Surface based

- Which one to use depends on properties of the shapes, and the desired effect
Volume-Based 3D Morphing

- Represent each shape as a volume
- Morph one volume to another
- Can be expensive
- Does not take into account properties that might be important for animation
- Not used as much as surface-based morphing
Terms Used in Surface-Based

- **Object**
  - Entity that has 3D surface geometry

- **Shape**
  - Set of points in object space that make up an object’s surface

- **Model**
  - Any complete description of the shape of an object

- One 3D *object* may have several *models* that describe its *shape*
More Terms

- **Topology**
  - The number of holes an object has, and the number of bodies in the object
  - The vertex/edge/face connectivity of a polyhedron

- **Genus**
  - How many holes an object has
  - Sphere is *genus 0*
  - Doughnut is *genus 1*
Surface-Based 3D Morphing

- Two main problems to solve
  - Vertex correspondence problem
    - Finding a mapping for each vertex on one shape to a vertex on the other
  - Interpolation problem
    - Creating a set of intermediate objects that move from one to the other

- Shapes typically have different surface topologies
  - Connectivity of vertices
  - Some objects may have “holes” in them
Surface-Based 3D Morphing (cont.)

- If the topologies match
  - Just interpolate 3D vertex positions over time

- If star-shaped object
  - Find distance to point in kernel, and interpolate

R.W. Lindeman - WPI Dept. of Computer Science
Interactive Media & Game Development
Mapping Onto a Sphere

- Several approaches
  - Project all vertices, edges, etc. onto the surface of a sphere
  - Then take the union of the vertices/edges for both objects
  - Then project back
  - Then perform vertex-by-vertex interpolation

- Can lead to an explosion of new edges
- Does not attempt to match edges
- Other approaches (in book) try this
2D Morphing

- Usually an image-based post-process
- Transform a source image into a destination image

Main task
- Identify corresponding features of the two images.

Two main approaches
- User-defined coordinate grid
- Feature lines
User-defined Coordinate Grid

- Image-based approach
- User defines a curvilinear grid, where main features lie within corresponding grid squares

- Intermediate images are generated by
  - Interpolating the grid points
    - Linear, or higher-order using adjacent key frames
  - Stretching/compressing pixels from the source to the intermediate, and from the destination to the intermediate

- The two images are cross dissolved
User-defined Coordinate Grid (cont.)

- Grids

![Image A](Image A with grid points and curves defined)

![Image B](Image B with grid points and curves defined)
Feature-based Morphing

- User specifies pairs of lines on source and destination images
  - Lines should cover some features

- A mapping for each pixel to each feature line in each image is established

- Intermediate line locations are then determined using interpolation
  - Either endpoints, or center+orientation

- A weighted average is used to generate the intermediate images

- Cross-dissolve is again used
Some Examples

- http://davis.wpi.edu/~matt/courses/morph/