Animation

- According to Webster
  - The state of being full of movement
- "Animated Cartoon"
  - A film made from many drawings. Each drawing involves a change of position in characters or objects.
  - Rapid projection makes the motion appear fluid.

1

Applications of Animation

- Entertainment
  - *Toy Story, The Last Starfighter, The Return of the King*
- Simulators
  - Military, Flight, Doom/Quake/Tribes 2
- Reenactments
  - Insurance industry, scientific visualization (Mars Exploration Rover Mission)

2

Physiological Reality

- Issue 1: Human eye needs about 15+ images/second for illusion of movement
  - Temporal integration model – want smooth transitions after considering HVS effects
- Issue 2: Human eye can easily detect flicker below about 50 images/second
  - Suggests refresh rate for CRTs, film projectors
  - CRT refresh rates: 60, 72, 75, 85, 100 Hz common
- Theater movie film: 24 fps, frame doubling
- TV: 29.97 ("30") fps, interlace by 2

3

The "Easy" Part

- Animation "projector"
  - A system which displays a sequence of pre-prepared images
- Requirements
  - Controls timing and sequencing
  - Related to hardware refresh rate

4

The "Hard" Part

- Drawing is not the "hard" part
- Describing all the motion is
- Example:
  - Suppose a scene has 6 motion parameters
  - Frame rate of 30 Hz
  - Duration of 5 seconds
  - # of motion values = 6 * 30 * 5 = 900
- A human has 200 motion parameters

5

Specifying Motion

- Procedural – each motion parameter is described as a function of time
- Representational
  - Articulation – object hierarchy
  - Deformation – soft object
- Stochastic – random variables and processes
- Behavioral – rules decide next motions

6

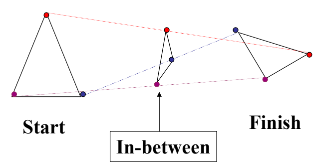
Key Framing

- Classic approach
  - Story board is developed
  - Key frames are identified and drawn
  - In-between frames are interpolated
- Modern variant
  - Morphing – transform one shape into another

7

Generating In-betweens

- Map points/vertices in one image to another
- Interpolate position in-between
  - Each point follows a line or curve



8

Interpolation Paths

- Path shape is critical to the “effect”
- Linear is most common
  - Why?
- Common curves
  - Circular/trigonometric
  - Splines
- Motion studies – describe vertex paths

9

Accelerations

- All motion so far is constant velocity
  - Vertices move the same distance between frames
  - Due to regular in-between positions
- In-betweens may use any spacing
  - Velocity curves can be used
  - Don't forget that playback is at fixed intervals

Will appear to speed up!

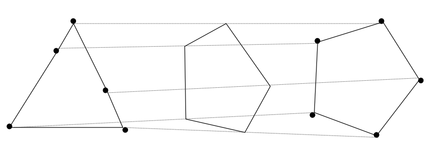
10

Morphing

- One shape is transformed into another
- Ultimate issue: Path each vertex follows
- Problem: Not all images have the same number of vertices
- Dummy vertices need to be added to one of the key frames

11

Morphing Example



Choice of "added" points and interpolation curves defines the art!

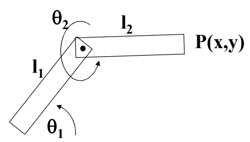
12

Articulated Motion

- Common in robotics
- Objects composed of a series of rigid links
- Joint types
  - Revolute – rotate only
  - Prismatic – slide only
- Degrees of Freedom (DOF)
  - Number of independent position variables

13

A Simple Articulated Joint



- $x = l_1\cos\theta_1 + l_2\cos(\theta_1+\theta_2)$
- $y = l_1\sin\theta_1 + l_2\sin(\theta_1+\theta_2)$
- If orientation is important too
  - Need 6 DOF! (i.e., 6 joints)

Kinematics

14

Computing Articulations (1)

- Kinematics – parameterized matrix transforming from joint parameters to position and orientation
- Inverse Kinematics – parameterized inverse matrix
- Jacobian – “First” derivative of kinematics
  - Specify incremental “motion” effects
  - I.e., given the current position, what is the local effect of each parameter on the motion?

15

Computing Articulations (2)

- A common notation
  - Denavit-Hartenberg (DH)
    - 4 DOF translation between any 2 rigid frames
    - Rotate about z, Translate along z, Translate along x, Rotate around x

16

Alternate TRTR version

Computing Articulations (2)

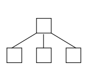
- A common notation
  - Denavit-Hartenberg (DH)
    - 4 DOF translation between any 2 rigid frames
    - Translate, rotate, translate, rotate

$$\begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{(i-1)} \\ \sin\theta_i\cos\alpha_{(i-1)} & \cos\theta_i\cos\alpha_{(i-1)} & -\sin\alpha_{(i-1)} & -\sin\alpha_{(i-1)}d_i \\ \sin\theta_i\sin\alpha_{(i-1)} & \cos\theta_i\sin\alpha_{(i-1)} & \cos\alpha_{(i-1)} & \cos\alpha_{(i-1)}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

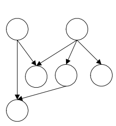
17

Hierarchical Representations (1)

- Good for articulated objects
- Structured relationship between main objects and sub-objects



  - Tree – hand is linked to arm, etc.



  - Directed acyclic graph (DAG) – allows multiple, "identical" sub-parts

18

Hierarchical Representations (2)

- Each "node" contains
  - Drawing function
    - May be parameterized
  - Transform relative to its parent
  - Children of this element
- Lends itself well to C++ class methodology

19

Soft Object Animation

- Deformations
  - Object is flexible
- Model surface using a flexible tessellation
- Each movable vertex may be a DOF
- Gets into curved surfaces
  - (optional) later lecture

20

References

- Where to begin looking for more information
- Key Framing
  - Hearn & Baker, *Computer Graphics*, Addison-Wesley, 1997, pp. 588-94.
- Hierarchical Models
  - Angel, *Interactive Computer Graphics*, Addison-Wesley, 2000, pp. 333-49.
- Motion Specification
  - Watt & Watt, *Advanced Animation and Rendering Techniques*, Addison-Wesley, 1992, Ch. 15-18.

21