CS 362: Computer Graphics

Computer Animation

Dr. Samit Bhattacharya
Dept. of Comp. Sc. & Engg.
IIT Guwahati, Assam, India

CA

- *Animation* - derived from Latin word *anima*
  - Meaning “the act, process or result of imparting life, interest, spirit, motion or activity”
- Computer animation – uses CG techniques
- Broadly of four types
  - Keyframing
  - Motion capture
  - Physics-based
  - Procedural

Keyframing

- Idea derived from traditional animation techniques
  - Animator specifies a set of “keyframes”
  - Computer generates the “in-betweens” to complete an animation sequence
- Keyframe is a misleading term
  - No complete frame (image) is involved
  - Animator specifies a set of parameter values

Key

- Keyframe parameters
  - Center of all objects, RGB color values, scaling factor along each axis etc.
- At pre-determined time instants (say t_k) along *animation timeline*, animator sets the value (say f_k) of parameter k
  - Key = (t_k, f_k)

Key

- Issues to be considered for setting key values
  - “Principles of Traditional Animation Applied to 3D Computer Animation”, John Lasseter, ACM Computer Graphics, 21(4), 1987

Animation Principles

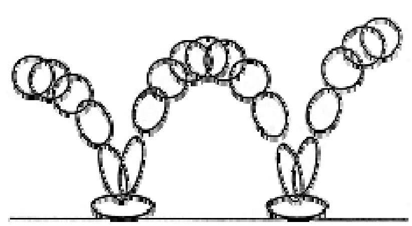
- Timing - speed of action, effects
  - Meaning of action
  - Emotional state
  - Perceived weight of objects (heavy objects move slowly)
- Sound can provide natural timing anchor
  - E.g. lip movement synchronized with speech

Principles Contd…

- Action layout: Planning of motion
  - Staging (high level planning): should lead a viewer's eyes to where the important action is currently concentrated
    - Familiarity with human perception is useful
  - Low level action planning – three stages
    - Anticipation – preparation for the action
    - The action itself
    - Follow through: termination of the action

Principles Contd…

- Animation techniques: goal is to make motion look natural
  - Squash and stretch: change object shape to give a feeling of weight, flexibility, response to pressure



Principles Contd…

- Animation techniques: goal is to make motion look natural
  - Arc: natural motion rarely happens along straight line, follow *arc* (curved path)
  - Slow in and out: no real world motion can instantly change speed, i.e. the motion should start and end gradually
- Above can be used with some reasonable amount of *exaggeration*
  - to achieve greater artistic effect/ emphasize some action

Principles Contd…

- Animation techniques: goal is to make motion look natural
  - Secondary action: motions that result from some other action
    - Needed for interest and realism
    - Shouldn't distract from primary motion
  - Appeal: attractive to the eye, strong design (avoid symmetries)

Keyframing Again

- Follow animation principles to specify keys
- Let computer generate the in-betweens – using interpolation techniques
  - Basic idea: each key is a data point on the *keyframe curve*
  - Use spline interpolation for these data points
    - Catmull-Rom spline widely used

Keyframing Again

- Most animation systems allow interactive fine editing of initial (interpolated) keyframe curve
  - TCB control – another technique

TCB Control

- TCB stands for tension (t), control (c) and bias (b)
  - The three parameters can be changed to change the keyframe curve shape

    Let T_{in}^{k} and T_{out}^{k} are the incoming and outgoing tangent valus at the k-th data point, Δt is the spacing (time gap) between data points

    Then, Catmull-Rom expression can be written as

$$T_k^{in} = T_k^{out} = \frac{1}{2\Delta t}(f_{k+1} - f_k) + \frac{1}{2\Delta t}(f_k - f_{k-1})$$

    For TCB, the corresponding expressions are,

$$T_k^{in} = \frac{(1-t)(1-c)(1+b)}{2\Delta t}(f_{k+1} - f_k) + \frac{(1-t)(1+c)(1-b)}{2\Delta t}(f_k - f_{k-1})$$

$$T_k^{out} = \frac{(1-t)(1+c)(1+b)}{2\Delta t}(f_{k+1} - f_k) + \frac{(1-t)(1-c)(1-b)}{2\Delta t}(f_k - f_{k-1})$$

Motion Control

- We want to control both the path of moving object (where it is moving) as well as the speed (how fast it is moving)
  - Simple interpolation not sufficient (only determines path, not speed)
  - Technique: determine position at time t as

$$P(t) = P(u(s(t)))$$

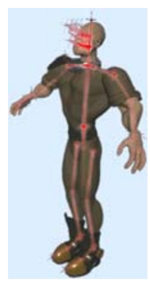
    P(u) = interpolated curve in parametric form

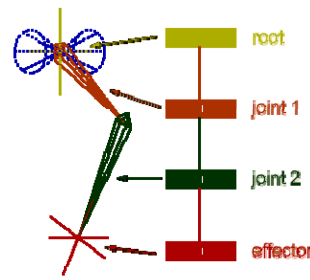
    s(t) = function to compute the spatial position w.r.t. time

    u(s) = relating spatial position to keyframe curve

(Skinned) Character Animation

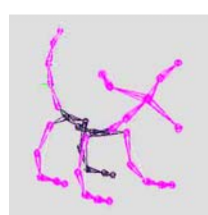
- Also called *articulated figure animation*
- Skeleton model
  - A hierarchy of "bones" – inner layer
  - Skin – outer layer



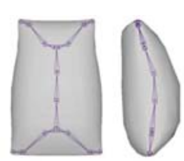


(Skinned) Character Animation

- Useful for
  - Non-humans



  - Even for non-animals



Skeleton Hierarchy

- Root represents the whole character (in world coordinate)
- Each joint acts as parent to the hierarchy below it
- Local transformation matrix (TM) to relate a joint to its parent
  - Concatenate all TMs along the path from root to a joint to get the TM to relate the joint to the world coordinate
- To evaluate whole skeleton (position and orientation of all joints)
  - Perform depth-first traversal of the complete tree

Kinematics

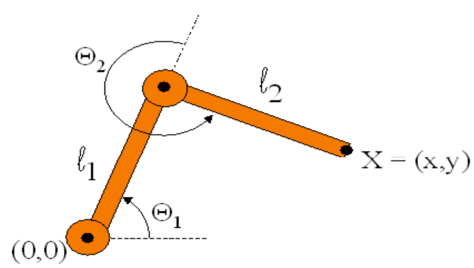
- Determined by positions, velocities, accelerations
- Forward kinematics
  - Low level approach - animator explicitly specifies all motions of every part (joint) of the animated structure (the TMs)
  - Each node inherits movement of all nodes above it

Kinematics

- Inverse kinematics
  - Requires only the position of the ends of the structure – computer does the rest
  - Functions as black box - controls detailed movement of entire structure

Forward Kinematics

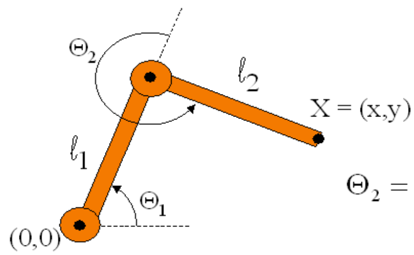
- Animator specifies joint angles
- Computer finds position of end-effector: X



$$X = (l_1 \cos\Theta_1 + l_2 \cos(\Theta_1 + \Theta_2), l_1 \sin\Theta_1 + l_2 \sin(\Theta_1 + \Theta_2))$$

Inverse Kinematics

- Animator specifies end-effector positions
- Computer finds joint angles



$$\Theta_2 = \cos^{-1}\left(\frac{x^2 + x^2 - l_1^{\,2} - l_2^{\,2}}{2 l_1 l_2}\right)$$

$$\Theta_1 = \frac{-(l_2 \sin(\Theta_2)x + (l_1 + l_2 \cos(\Theta_2))y}{(l_2 \sin(\Theta_2))y + (l_1 + l_2 \cos(\Theta_2))x}$$

Facial Animation

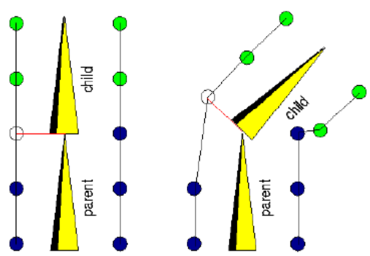
- Skeleton model not sufficient
  - Involves muscle movement, no hierarchy
- FACS – Facial Action Coding System
  - P. Ekman and W. Friesen. *Facial Action Coding System: A Technique for the Measurement of Facial Movement*. Consulting Psychologists Press, Palo Alto, 1978
  - Decomposes facial expression into a sum of elementary motions called Action Units (AUs)
  - Combine AUs to synthesize facial expressions
    - See http://www.cs.cmu.edu/~face/facs.htm for a list of AUs with description and illustration

Deformation

- To deform a model, move its control points
  - The rest is details…
- Types of deformation
  - Skeleton deformation
  - Function-based deformation
  - Free-form deformation
  - Point cluster deformers
  - Morphing

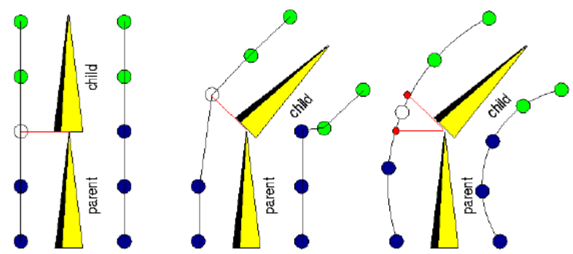
Skeleton Deformation

- Associate each point (on skin) with nearest link (of skeleton)
- When link moves, transform its points



Point Weights

- A point may gets affected by several links
  - Take weighted average
  - Adjust the weights until it looks good



Function-based Deformation

- Define a function over all space

  M: $\mathbf{R}^3 \rightarrow$ Transformation (matrix)

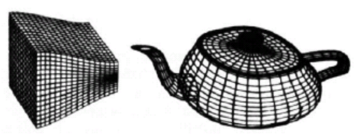
- To transform a point ***P***
  - Evaluate function **M** at ***P***
  - Transform ***P*** by the result: ***P'*** = **M*(P)*** ***P***

Examples

Taper:

$$s(x)=\begin{cases}1, & x \le x_0\\ 1-.5\dfrac{x-x_0}{x_1-x_0}, & x_0 < x < x_1\\ .5, & x_1 \le x\end{cases}$$

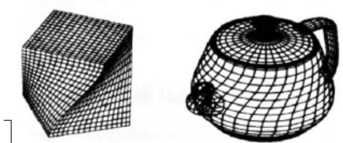
$$P'=\begin{bmatrix}1 & 0 & 0\\ 0 & s(p_x) & 0\\ 0 & 0 & s(p_x)\end{bmatrix}\begin{bmatrix}p_x\\ p_y\\ p_z\end{bmatrix}$$



Twist:

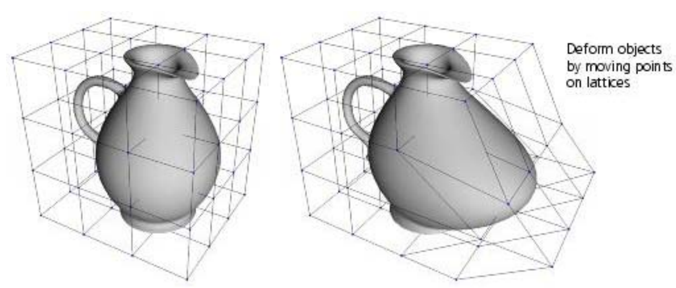
$$r(z)=\begin{cases}0, & z \le z_0\\ \dfrac{z-z_0}{z_1-z_0}\theta_{\max}, & z_0 < z < z_1\\ \theta_{\max}, & z_1 \le z\end{cases}$$

$$P'=\begin{bmatrix}\cos(r(p_z)) & -\sin(r(p_z)) & 0\\ \sin(r(p_z)) & \cos(r(p_z)) & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}p_x\\ p_y\\ p_z\end{bmatrix}$$

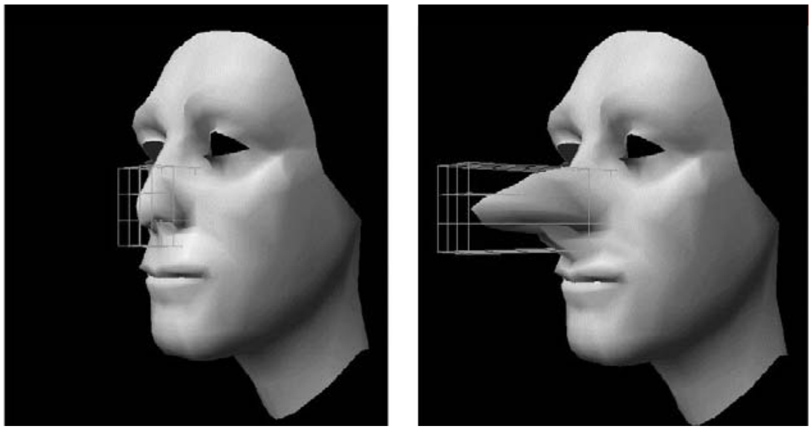


Free-Form Deformation (FFD)

- Define a lattice around the model
- Move the points of the lattice
- The model deforms with it

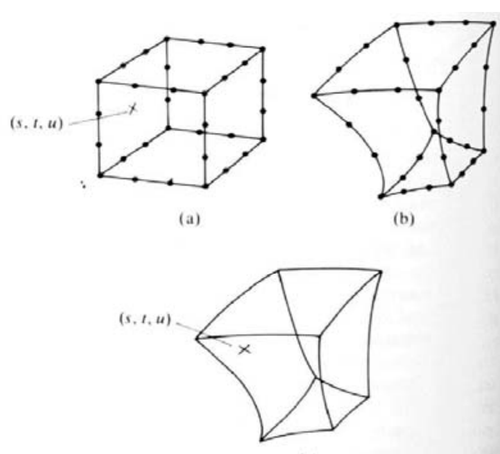


FFD Example



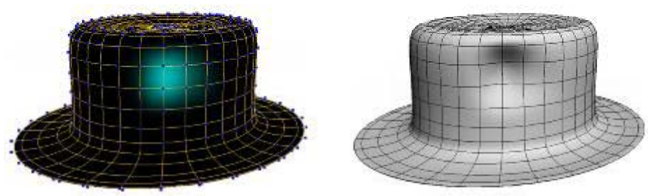
Computing FFD

- Find *(s,t,u)* coordinates of P in original grid
- Deform grid
- Interpolate new object point at (s,t,u)



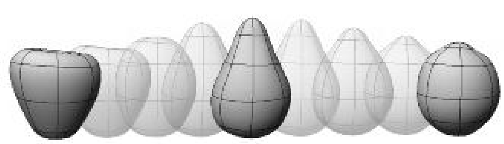
Point Cluster Deformers

- Select "cluster" of points
- Apply an operation directly to some points
- Weights often set by spatial fields



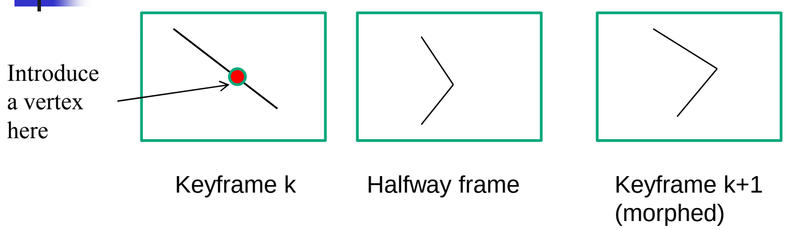
Morphing

- From "Metamorphosis" - transforming objects from one form to another



- Can be done by transforming polygon shapes through in-betweens from one keyframe to another
  - Key idea: adjust no of edges/vertices

Morphing



- General preprocessing rules for equalizing keyframes can be stated in terms of either no of edges or vertices

Morphing Rules

- Rules (edge based)
  - L_k = no of line segments in keyframe k

    L_{k+1} = no of line segments in keyframe k+1
  - The maximum and minimum no of lines to be equalized

    $L_{max} = \max(L_k, L_{k+1})$, $L_{min} = \min(L_k, L_{k+1})$
  - Next compute:

    $N_e = L_{max} \bmod L_{min}$

    $N_s = \text{int}(L_{max}/L_{min})$

Morphing Rules

- Preprocessing rules
  - Divide Ne edges of keyframe having less no of edges into Ns +1 sections
  - Divide the remaining edges of keyframe having less no of edges into Ns sections
- Rules (vertex based) - similar

$V_{max} = \max(V_k, V_{k+1})$, $V_{min} = \min(V_k, V_{k+1})$

$N_{ls} = (V_{max} - 1) \bmod (V_{min} - 1)$

$N_p = \text{int}((V_{max} - 1)/(V_{min} - 1))$

Morphing Rules

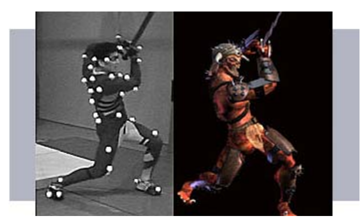
- Preprocessing rules
  - Add Np points to N_{ls} line segments of the keyframe having less no of edges
  - Add Np – 1 points to the remaining edges of the keyframe having less no of edges

Motion Capture

- Capture of motion of (human) actor
  - Whole body, upper body, face
- Technology – optical, magnetic, mechanical
  - Employs special sensors (trackers) to record motion of human performer

Motion Capture

- Problems
  - Difficult to accurately measure motion of human body – clothes may shift, etc
  - If object used to generate recorded motion and graphical object have different dimensions, animation will have noticeable flaws



Other Animation Techniques

- Physics-based
  - Actual physical laws used to generate motion (e.g. fluid motion, cloth simulation)
  - Requires solving ordinary as well as partial differential equations - time consuming
  - Use numerical methods
    - Better still - try to avoid as much as possible (use such techniques only when nothing else are available)

Other Animation Techniques

- Procedural – use procedure to generate motion
  - Physics-based animation can be approximated with procedures (e.g. generating waves on a lake surface)

Animating Groups of Objects

- Few objects – AI techniques may be used
  - Set high-level goal for each object
  - Plan motion to reach goal
  - Plan execution
  - Ex. Autonomous objects in games, agents
- Larger group - emergent behavior
  - Each object has limited intelligence
  - Each object has limited interaction with few of closest neighbors
  - Together a new behavior emerges
  - Ex. School of fish, flock of birds, group of animals …