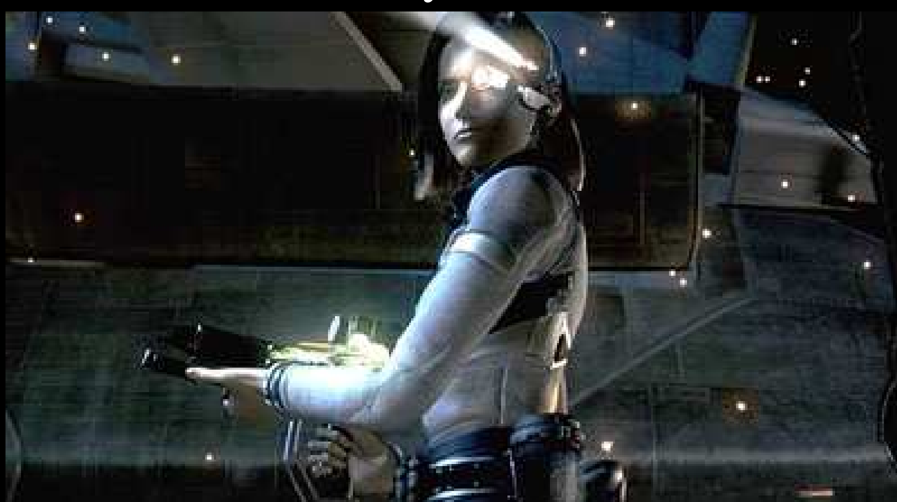
Animation

1999 Star Wars: Phantom Menace

2002 LOTR Two Towers

2001 Final Fantasy



Computer Animation

- Models have parameters
  - Polygon positions, normals, spline control points, joint angles, camera parameters, lights, color
  - *n* parameters define an n-dimensional *state space*
  - Values of *n* parameters = point in *state space*
- Animation defined by path through state space
  - To produce animation:
    - » 1. start at beginning of state space path
    - » 2. set the parameters of your model
    - » 3. render the image
    - » 4. move to next point along state space path, repeat 2
  - Path usually defined by a set of motion curves
    - » one for each parameter
- Every animation technique reduces to specifying the state space trajectory—the state space will change with the technique

Traditional Cel Animation

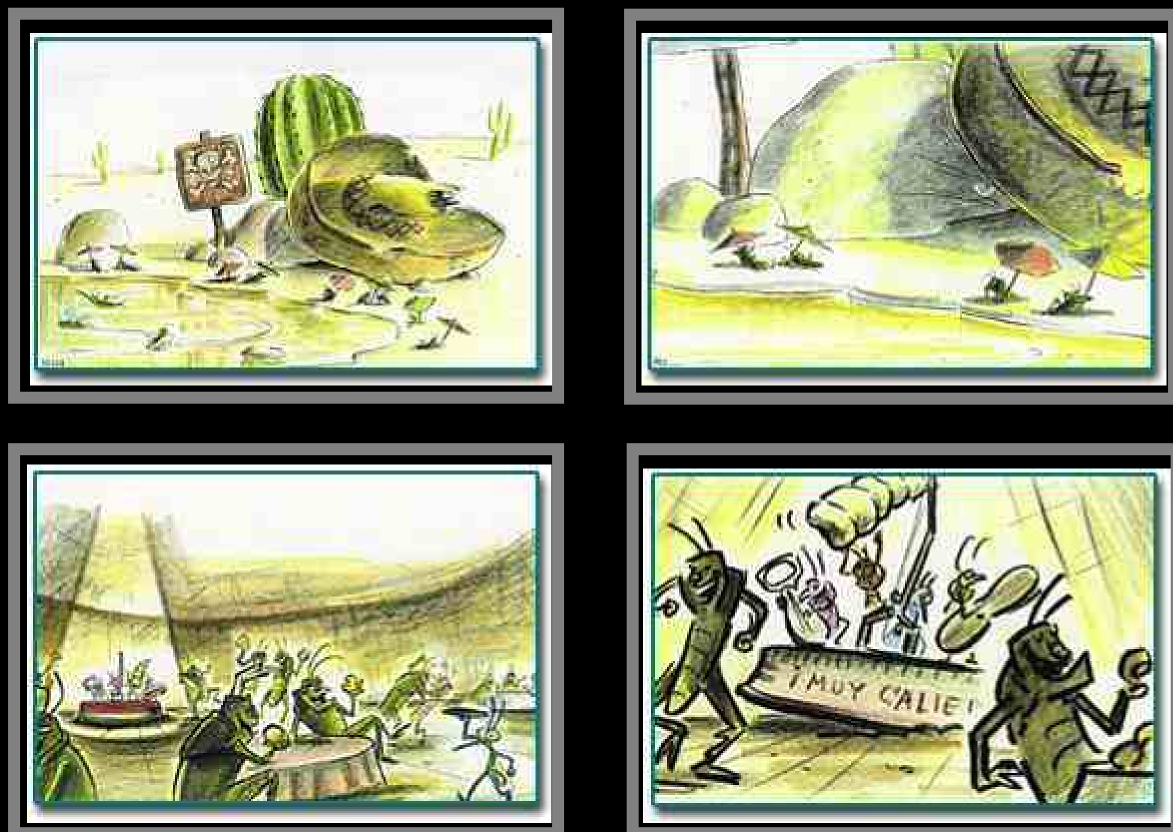
- Film runs at 24 frames per second (fps)
  - That's 1440 pictures to draw per minute
  - 1800 fpm for video (30fps)
- Productions issues:
  - Need to stay organized for efficiency and cost reasons
  - Need to render the frames systematically (render farms)
- Artistic issues:
  - How to create the desired look and mood while conveying story?
  - Artistic vision has to be converted into a sequence of still frames
  - Not enough to get the stills right--must look right at full speed
    - » Hard to "see" the motion given the stills
    - » Hard to "see" the motion at the wrong frame rate

Traditional Animation: The Process

- Story board
  - Sequence of drawings with descriptions
  - Story-based description
- Key Frames
  - Draw a few important frames as line drawings
    - For example, beginning of stride, end of stride
- Inbetweens
  - Draw the rest of the frames
- Painting
  - Redraw onto acetate *Cels*, color them in

Gertie

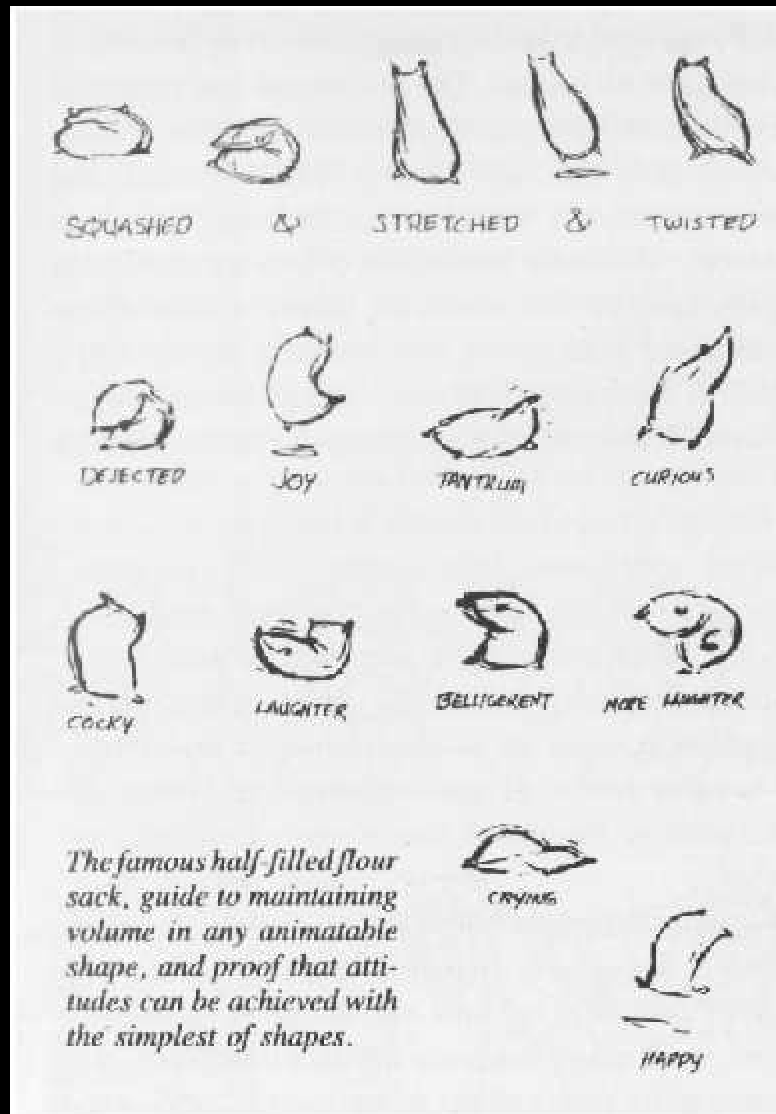
Story Boarding (from “A Bug’s Life”)



Principles of Traditional Animation
[Lasseter, SIGGRAPH 1987]

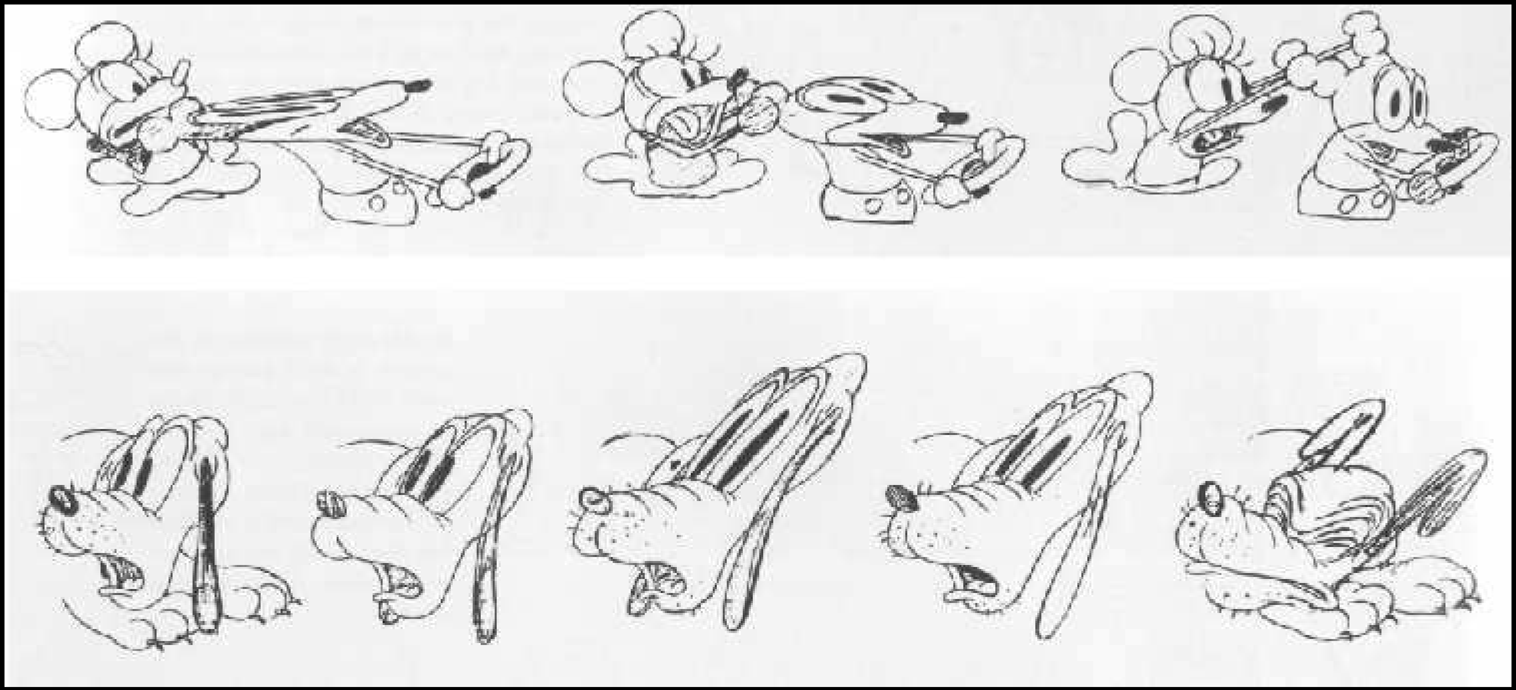
- Stylistic conventions followed by Disney’s animators and others *(but this is not the only interesting style, of course)*
- From experience built up over many years
  - Squash and stretch -- use distortions to convey flexibility
  - Timing -- speed conveys mass, personality
  - Anticipation -- prepare the audience for an action
  - Followthrough and overlapping action -- continuity with next action
  - Slow in and out -- speed of transitions conveys subtleties
  - Arcs -- motion is usually curved
  - Exaggeration -- emphasize emotional content
  - Secondary Action -- motion occurring as a consequence
  - Appeal -- audience must enjoy watching it

Principles of Traditional Animation

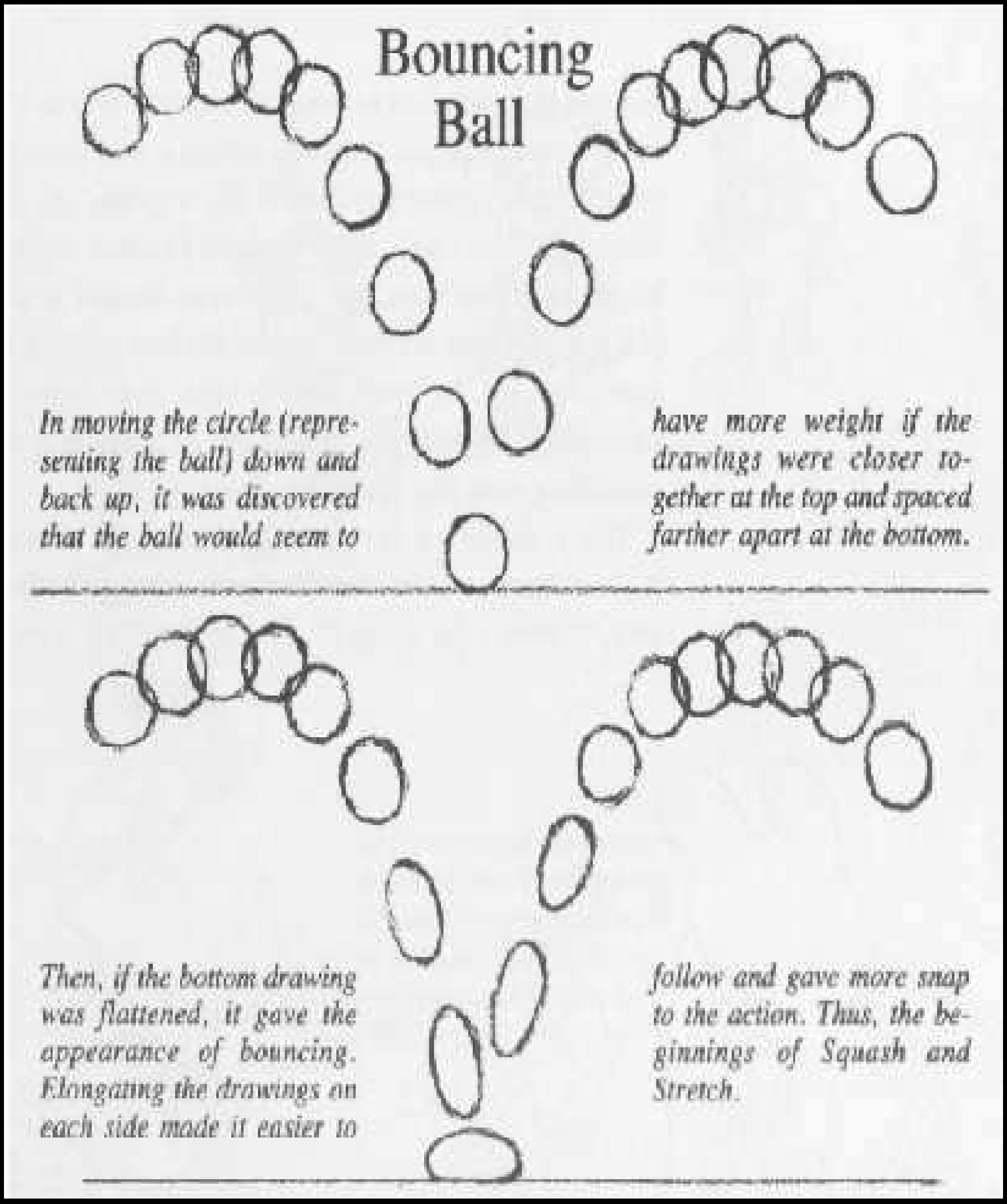


The famous half-filled flour sack, guide to maintaining volume in any animatable shape, and proof that attitudes can be achieved with the simplest of shapes.

Squash and Stretch

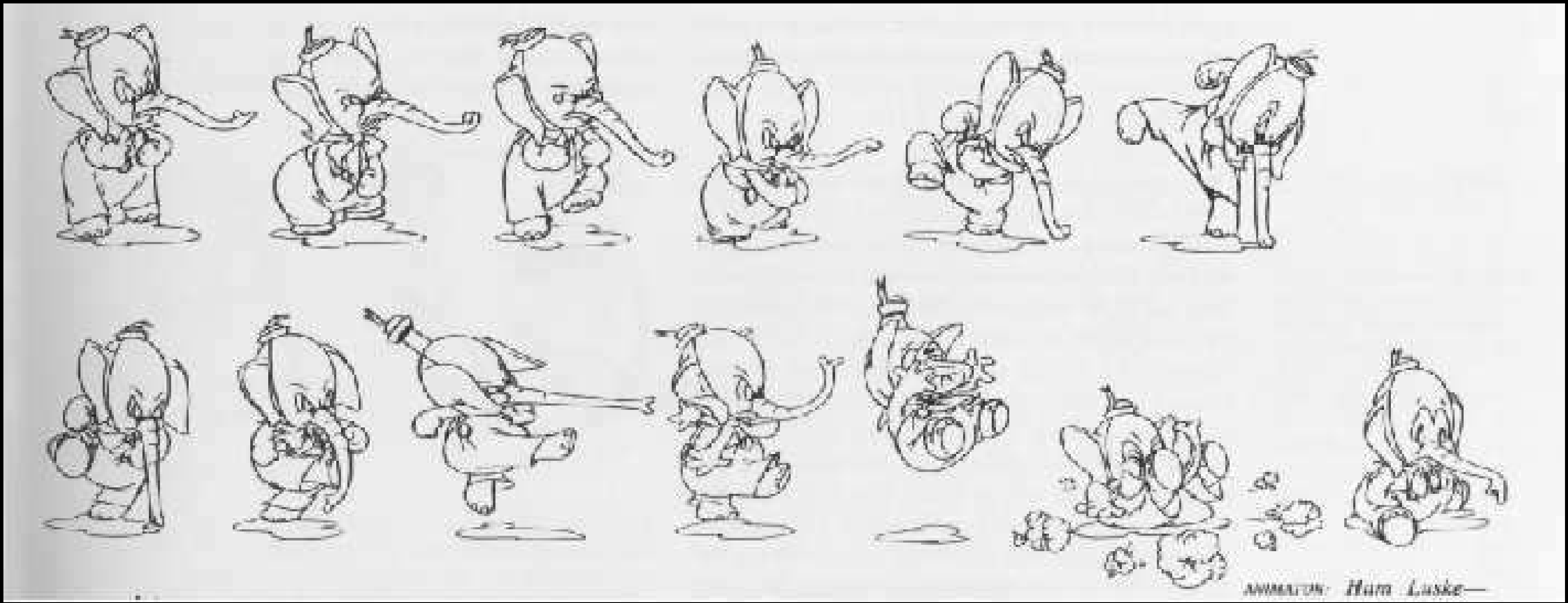


Squash and Stretch

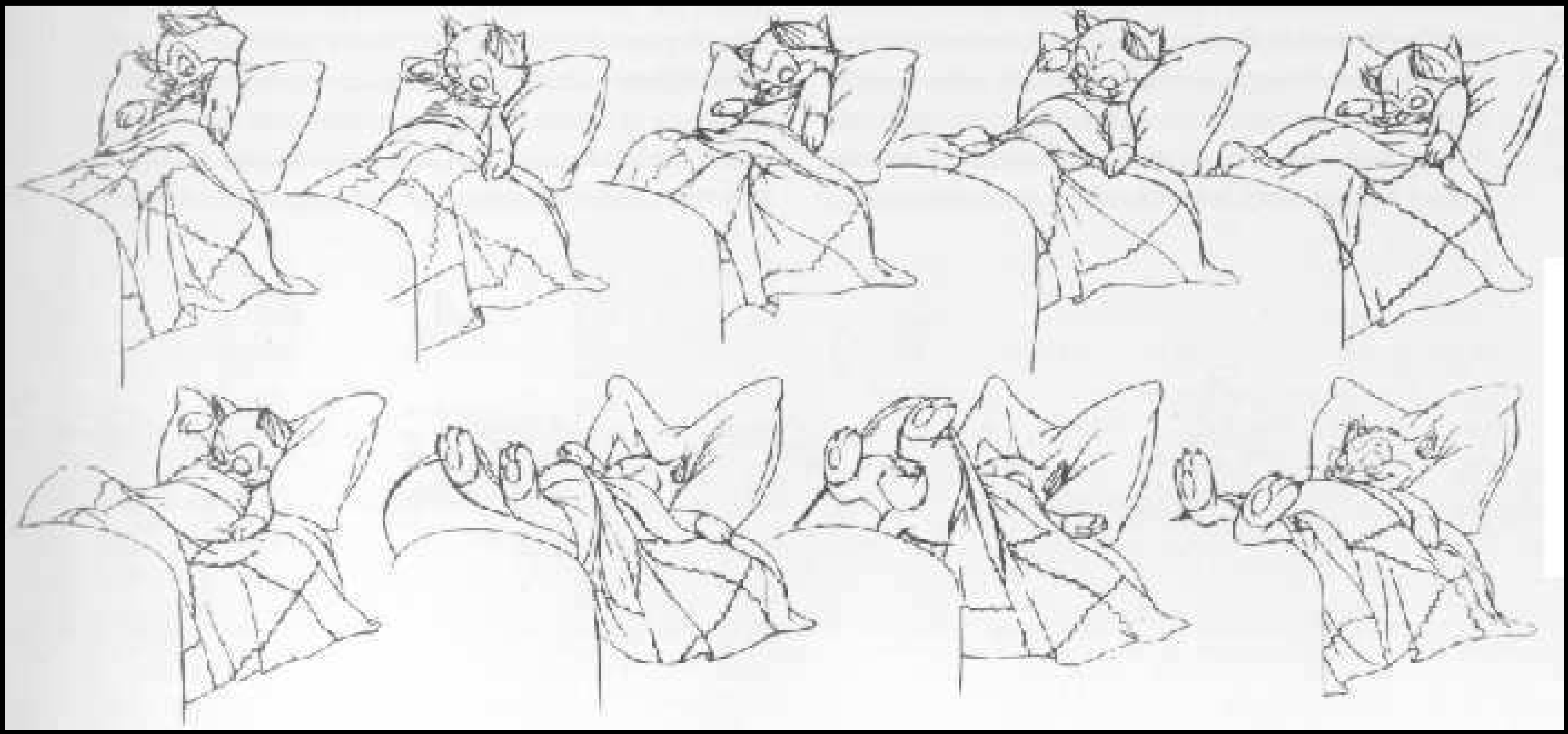


Anticipation

Follow Through



Secondary Action



Computer Assisted Animation

- Computerized Cel painting
  - Digitize the line drawing, color it using seed fill
  - Eliminates cel painters (low rung on totem pole)
  - Now part of the production process (little hand painting any more)
  - e.g. *Lion King*
- Cartoon Inbetweening
  - Automatically interpolate between two drawings to produce inbetweens (*a la* morphing)
  - Hard to get right
    - » inbetweens often don't look natural
    - » what are the parameters to interpolate? Not clear...
    - » not used very often

Hunger

3D Computer Animation

- Generate the images by rendering a 3-D model
- Vary the parameters to produce the animation
- Brute force
  - Manually set the parameters for each and every frame
  - For an *n* parameter model: *1440n* values per minute
- Traditional keyframing
  - Lead animators draw the important frames
  - Underpaid drones draw the inbetweens
- Computer keyframing
  - Lead animators create the important frames with 3-D computer models
  - Unpaid computers draw the inbetweens
  - This is easier in 3D than with 2D line drawings!
  - The dominant production method

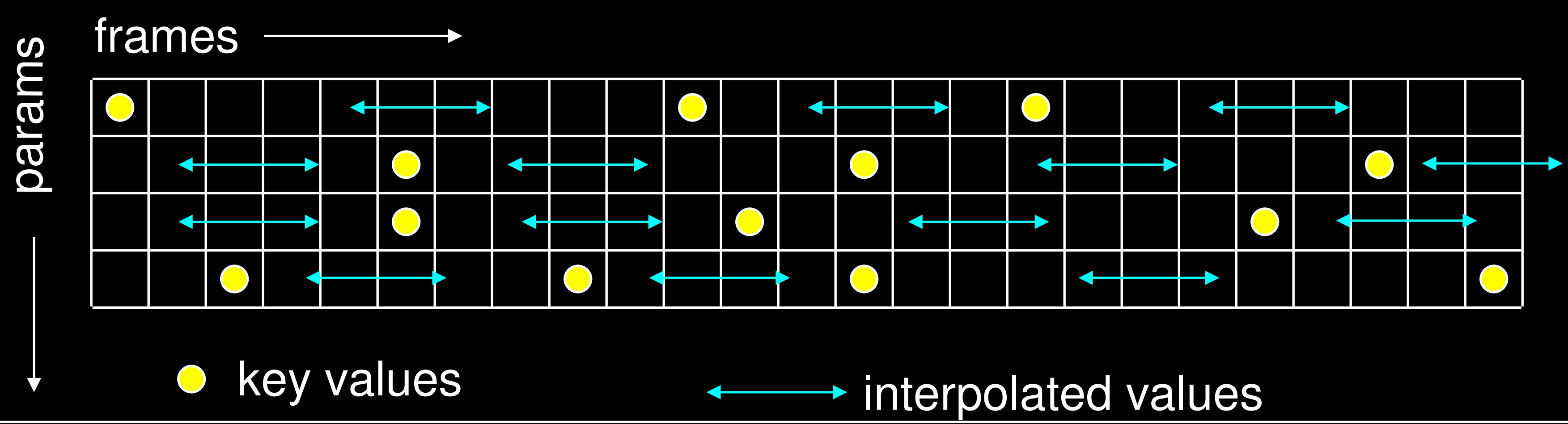
Geri's Game

Interpolation

- Hard to interpolate hand-drawn keyframes
  – Computers don’t help much
- The situation is different in 3D computer animation:
  – Each keyframe is a defined by a bunch of parameters (state)
  – Sequence of keyframes = points in high-dimensional state space
- Computer inbetweening interpolates these points
- How? You guessed it: splines

Keyframing Basics

- Despite the name, there aren't really keyframes, *per se.*
- For each variable, specify its value at the "important" frames. Not all variables need agree about which frames are important.
- Hence, *key values* rather than key frames
- Create path for each parameter by interpolating key values



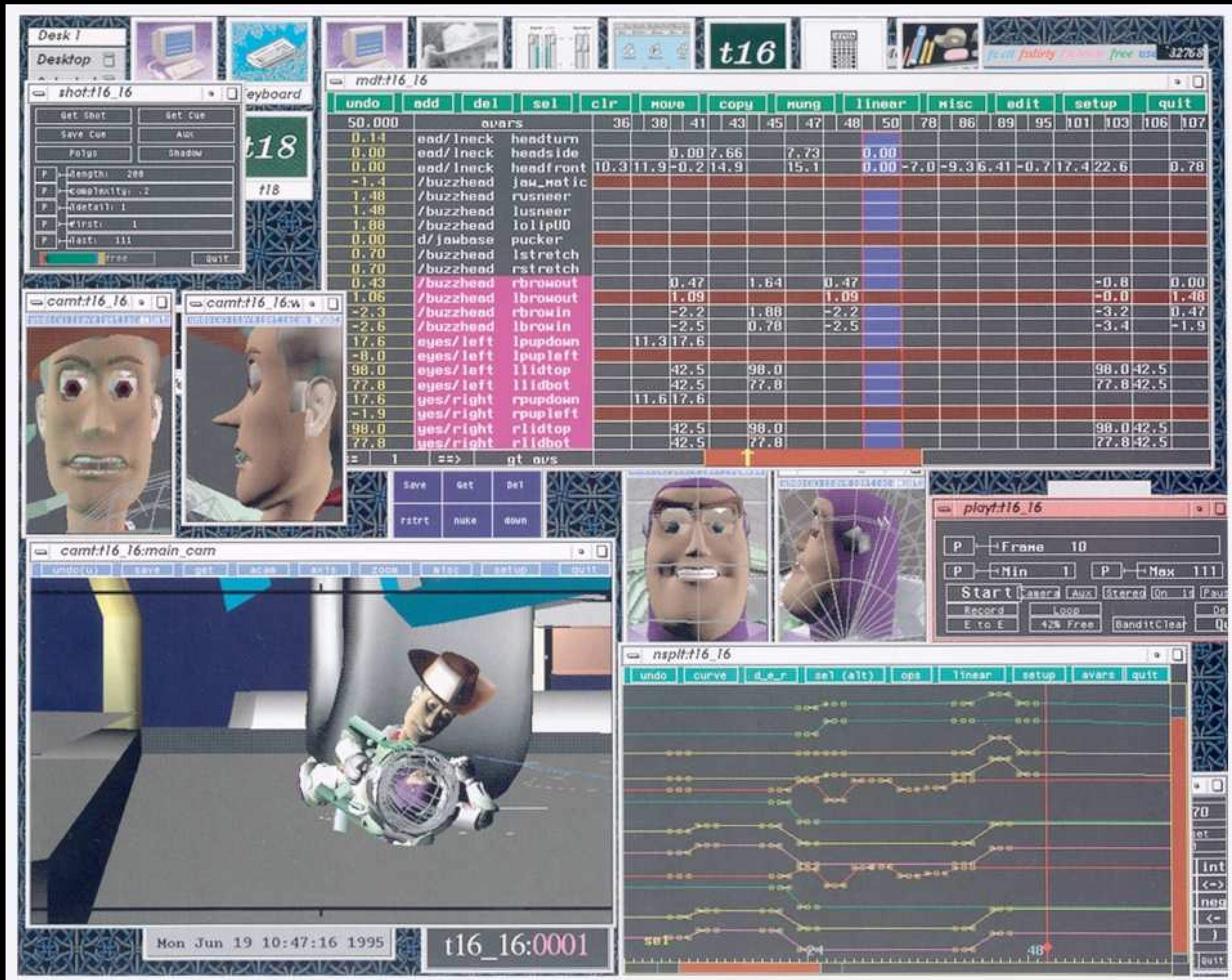
Keyframing: Issues

- What should the key values be?
- When should the key values occur?
- How can the key values be specified?
- How are the key values interpolated?
- What kinds of BAD THINGS can occur from interpolation?
  - Invalid configurations (pass through objects)
  - Unnatural motions (painful twists/bends)
  - Jerky motion

Keyframe Animation: Production Issues

- How to learn the craft
  - apprentice to an animator
  - practice, practice, practice
  - Read Cinefex, …
- Pixar starts with animators, teaches them computers and starts with computer folks and teaches them some art

From the Making of Toy Story

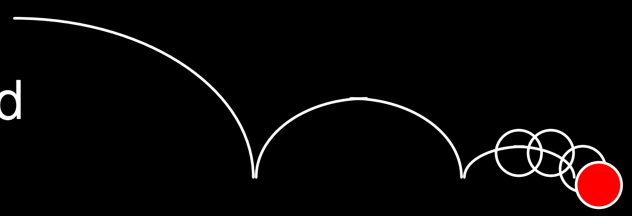


Scene from Toy Story II

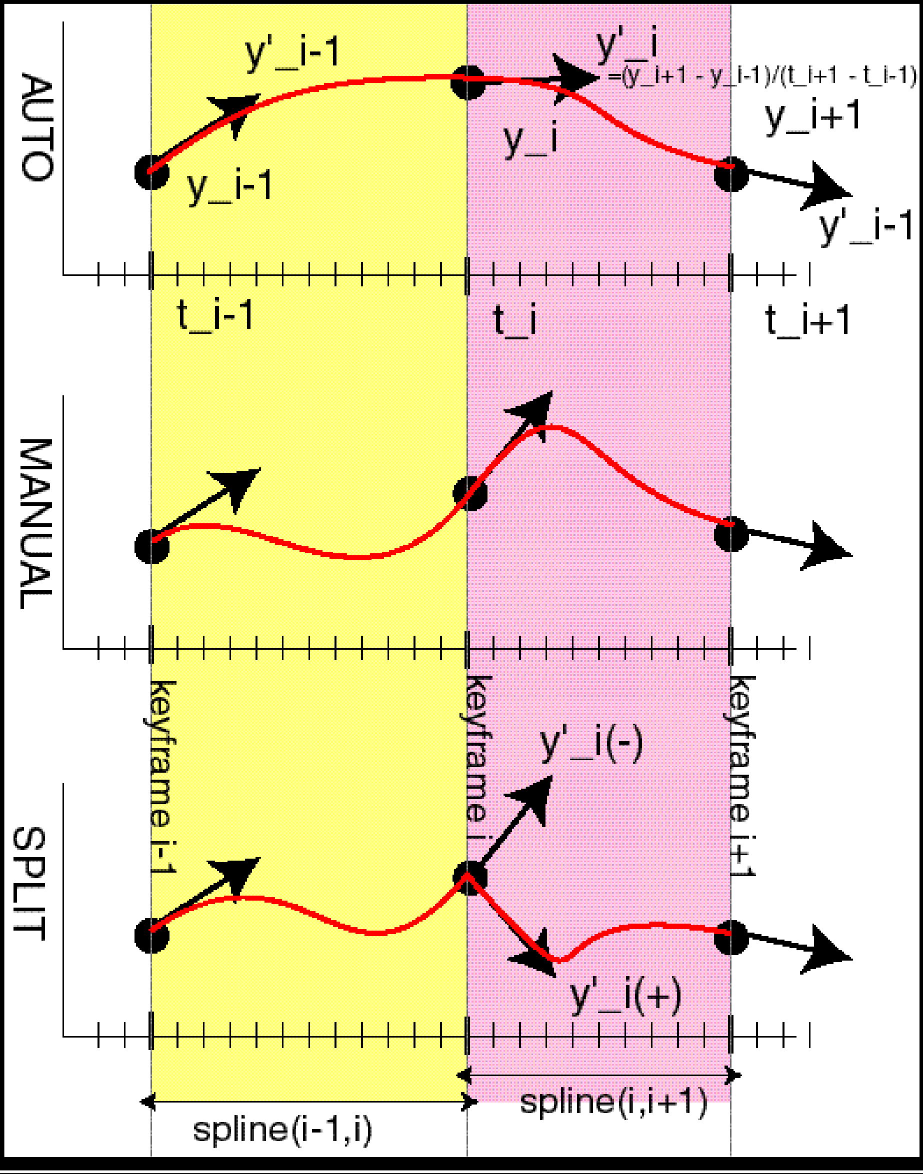
How Do You Interpolate Between Keys?

- Splines: non-uniform, C^1 is pretty good
- Velocity control is needed at the keyframes
- Classic example - a ball bouncing under gravity
  - zero vertical velocity at start
  - high downward velocity just before impact
  - lower upward velocity after
  - motion produced by fitting a smooth spline looks unnatural
- What kind of spline might we want to use?

Hermite is good



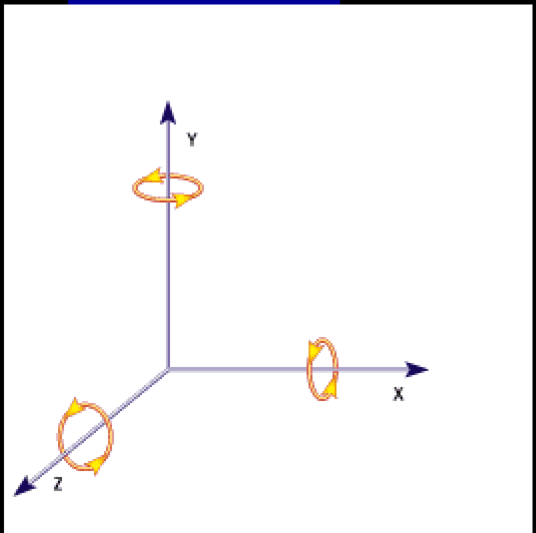
How Do You Interpolate Between Keys?



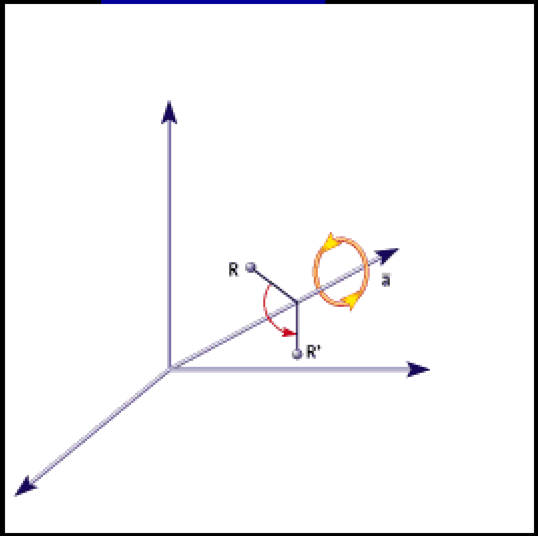
Problems with Interpolation

- Splines don't always do the right thing
- Classic problems
  - –Important constraints may break between keyframes
    - » feet sink through the floor
    - » hands pass through walls
  - –3D rotations
    - » Euler angles don't always interpolate in a natural way
- Classic solutions:
  - –More keyframes…
  - –Quaternions help fix rotation problems

Interpolating Rotations



Euler angles



Axis-angle

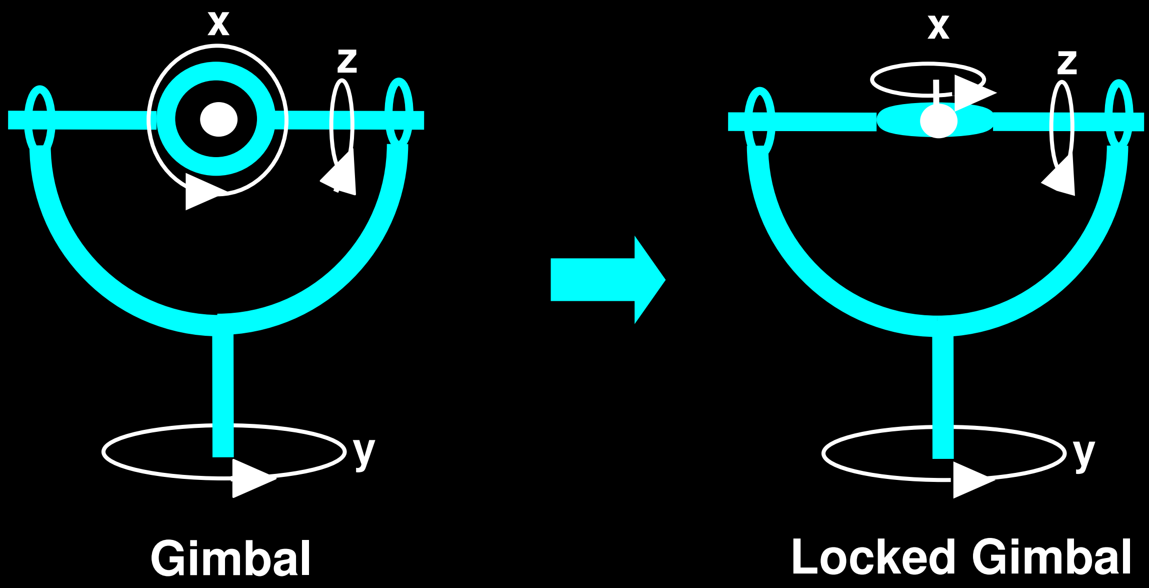
Q: What kind of compound rotation do you get by successively turning about each of the 3 axes at a constant rate?

A: Not the one you want

Euler Angles

- Good for single-axis rotations
- Awkward for other rotations

Gimbal Lock

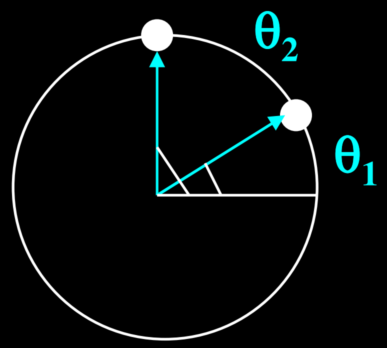


A *Gimbal* is a hardware implementation of Euler angles (used for mounting gyroscopes, expensive globes)

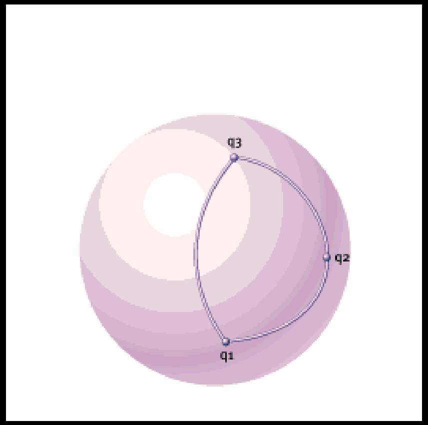
Gimbal lock is a basic problem with representing 3-D rotations using Euler angles

Quaternion Rotation

- We can think of rotations as lying on an n-D unit sphere



1-angle (θ) rotation
(unit circle)



2-angle (θ-ϕ) rotation
(unit sphere)

- Interpolating rotations means moving on n-D sphere
  - Can encode position on sphere by unit vector
  - SLERP: *Spherical Linear Interpolation*
    - » take shortest path between two points on unit sphere
  - How about 3-angle rotations?

Quaternion Rotation

- A quaternion is a 4-D unit vector q = [x y z w]
  - It lies on the unit hypersphere $x^2+y^2+z^2+w^2=1$
- For rotation about (unit) axis *v* by angle θ
  - vector part (sin θ/2) *v* = [x y z]
  - scalar part cos θ/2 = w
- The rotation matrix corresponding to a quaternion is

$$\begin{pmatrix} 1-2(Y^2+Z^2) & 2(XY-ZW) & 2(XZ+YW) \\ 2(XY+ZW) & 1-2(X^2+Z^2) & 2(YZ-XW) \\ 2(XZ-YW) & 2(YZ+XW) & 1-2(X^2+Y^2) \end{pmatrix}$$

Kinematics & Inverse Kinematics

- We need help in positioning joints
- Kinematics
  - gives motions in terms of joint angles, velocities, and positions
  - used by most keyframing and procedural animation systems
- *Inverse* kinematics
  - determine joint angles from positions
  - e.g. "calculate the shoulder, elbow, and wrist rotation parameters in order to put the hand here"
  - better for interaction
  - sometimes underdetermined (i.e. many combinations of joint angles to achieve a given end result)

UTPoser

Procedural Animation

- Define the motion using formulas
  - Hand-crafted
  - Physically based
- The animator must be a programmer
- Keyframing starts to become procedural as expressions are added
- At some level of complexity it becomes easier/more efficient than keyframing.

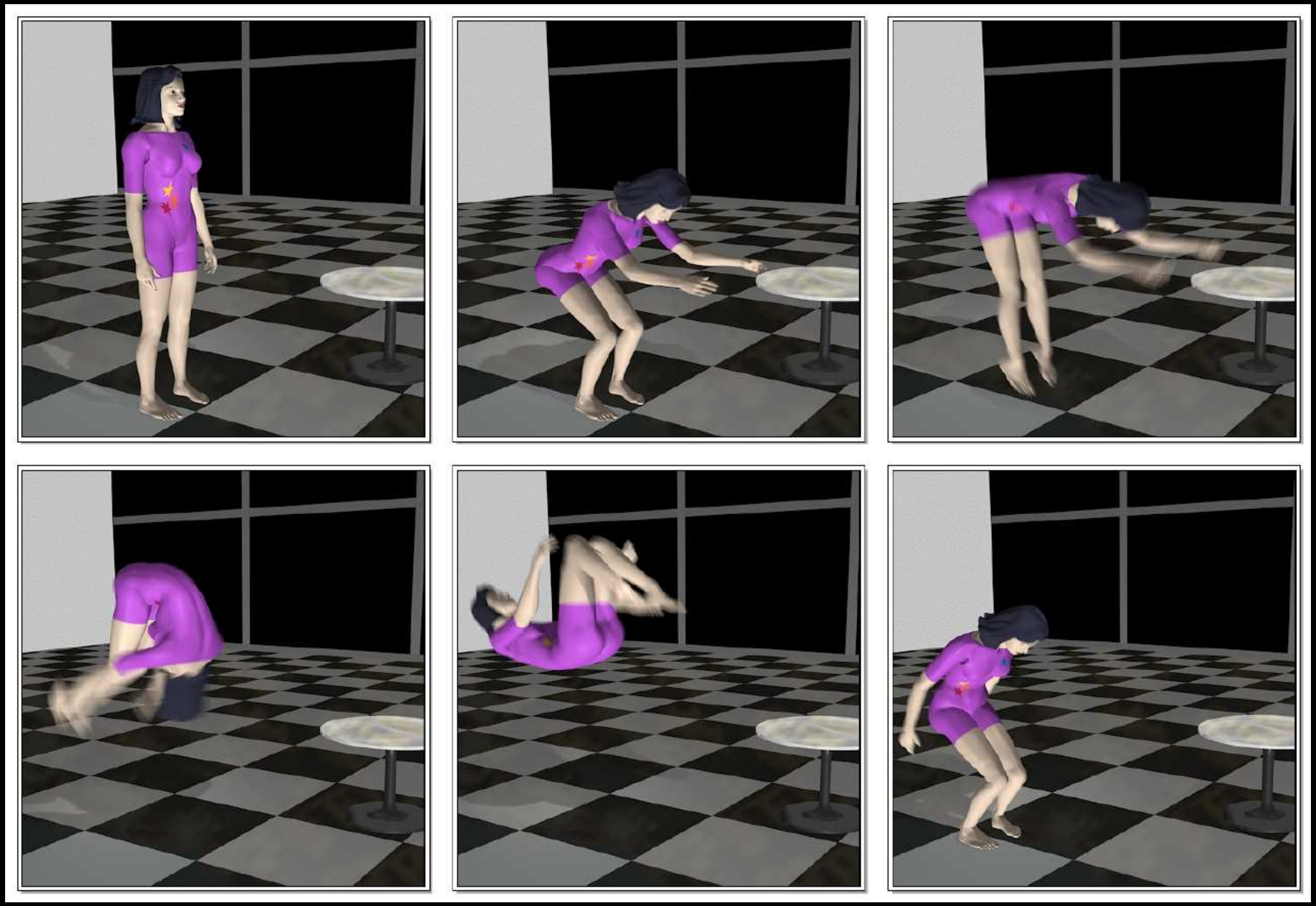
Procedural Animation

Battle of Helm's Deep, LOTR

Dynamics

- Generate motion by specifying mass and force, apply physical laws (e.g., Newton's laws)
- Simulates physical phenomena
  - –gravity
  - –momentum (inertia)
  - –collisions
  - –friction
  - –fluid flow (drag, turbulence, ...)
  - –deformation
  - –fracture

Active Simulations



Luxo

Alla LowD

Wayne Wooten and Jessica Hodgins

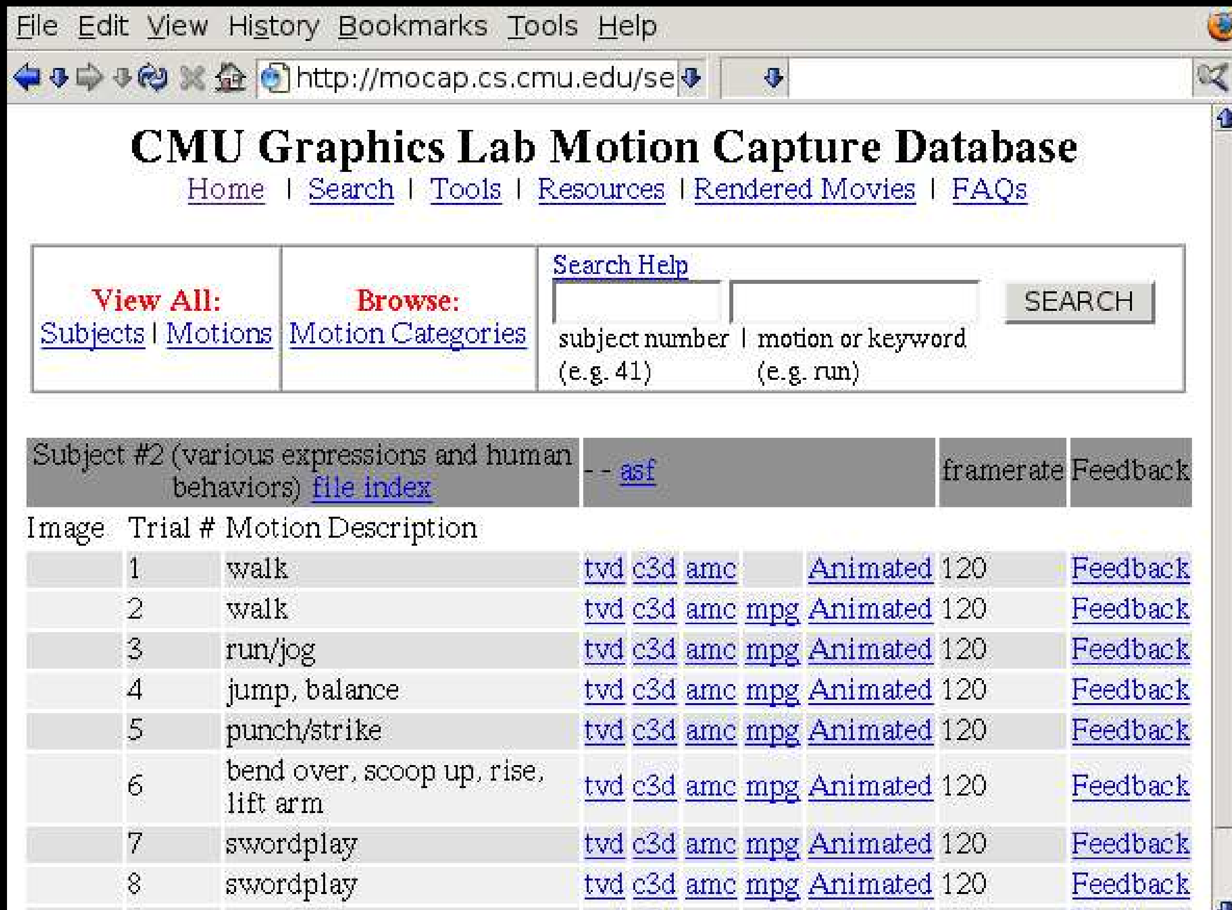
Passive Simulations

Performance-based Animation (Motion Capture)

- Record the animation from live action
  - simplest method - rotoscope (trace) over video of real motions
- Real time input devices
  - electronic puppeteering
- Motion capture
  - track motion of reference points
    - » body or face or hands
  - magnetic
  - optical
  - exoskeletons
  - convert to joint angles (not always straightforward)
  - use these angles to drive an articulated 3-D model
  - These motion paths can be *warped*


NEW LINE CINEMA
An AOL Time Warner Company

Motion capture databases



File Edit View History Bookmarks Tools Help

http://mocap.cs.cmu.edu/se

CMU Graphics Lab Motion Capture Database

Home | Search | Tools | Resources | Rendered Movies | FAQs

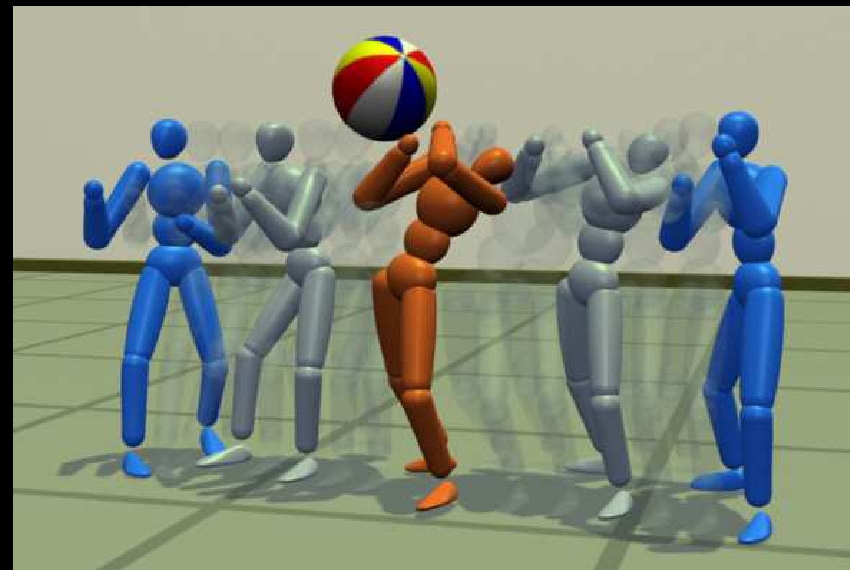
View All: Subjects | Motions	Browse: Motion Categories	Search Help — subject number (e.g. 41) | motion or keyword (e.g. run) — SEARCH

Subject #2 (various expressions and human behaviors) file index			- - asf					framerate	Feedback
Image	Trial #	Motion Description							
	1	walk	tvd	c3d	amc		Animated	120	Feedback
	2	walk	tvd	c3d	amc	mpg	Animated	120	Feedback
	3	run/jog	tvd	c3d	amc	mpg	Animated	120	Feedback
	4	jump, balance	tvd	c3d	amc	mpg	Animated	120	Feedback
	5	punch/strike	tvd	c3d	amc	mpg	Animated	120	Feedback
	6	bend over, scoop up, rise, lift arm	tvd	c3d	amc	mpg	Animated	120	Feedback
	7	swordplay	tvd	c3d	amc	mpg	Animated	120	Feedback
	8	swordplay	tvd	c3d	amc	mpg	Animated	120	Feedback

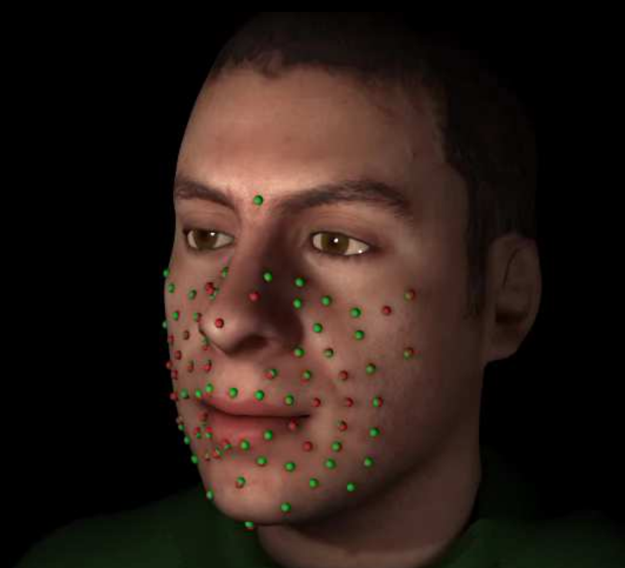
Kovar

Interpolated

Combining Motion Capture and Simulation

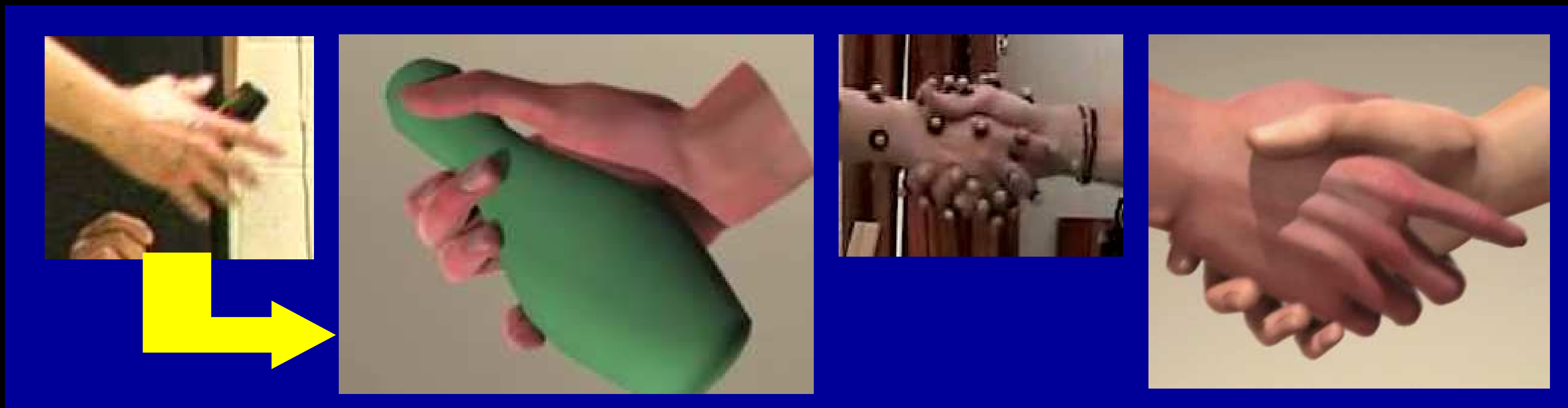


Zordan

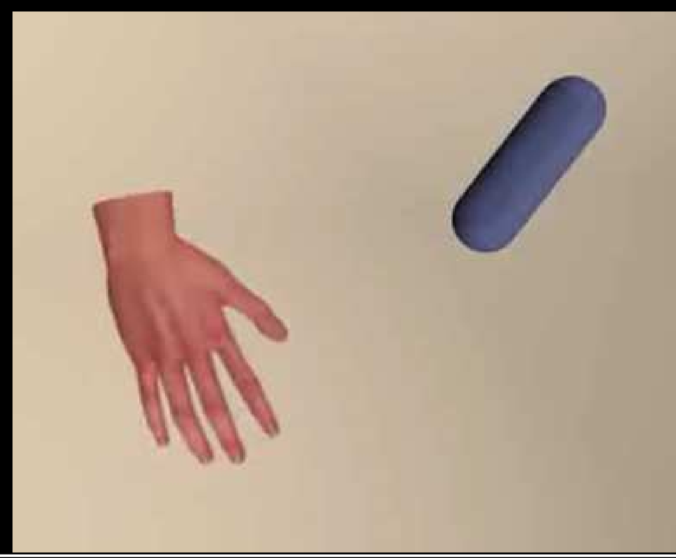


Sifakis

Combining Motion Capture and Simulation



Pollard



Improving Motion Capture

Hodgins