Lecture 14:

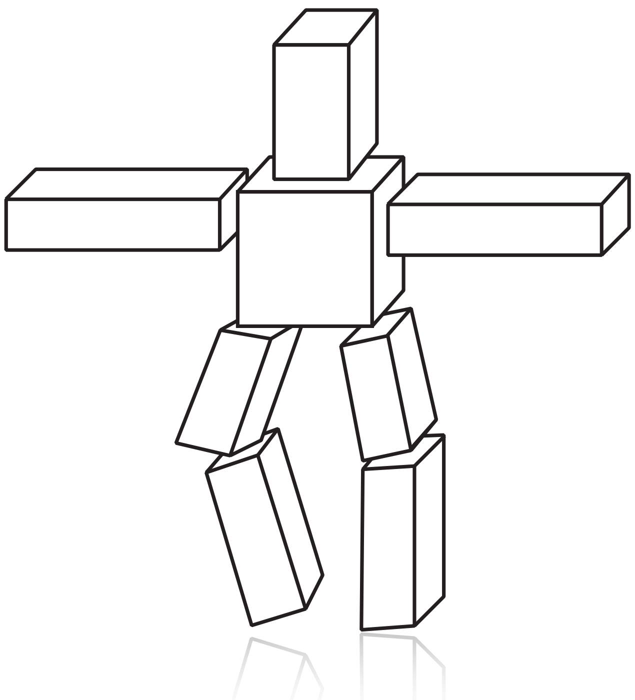
Introduction to Animation

Computer Graphics 2025

Fuzhou University - Computer Science

Increasing the complexity of our models

Trasformations



Geometry

Materials, lighting, ...

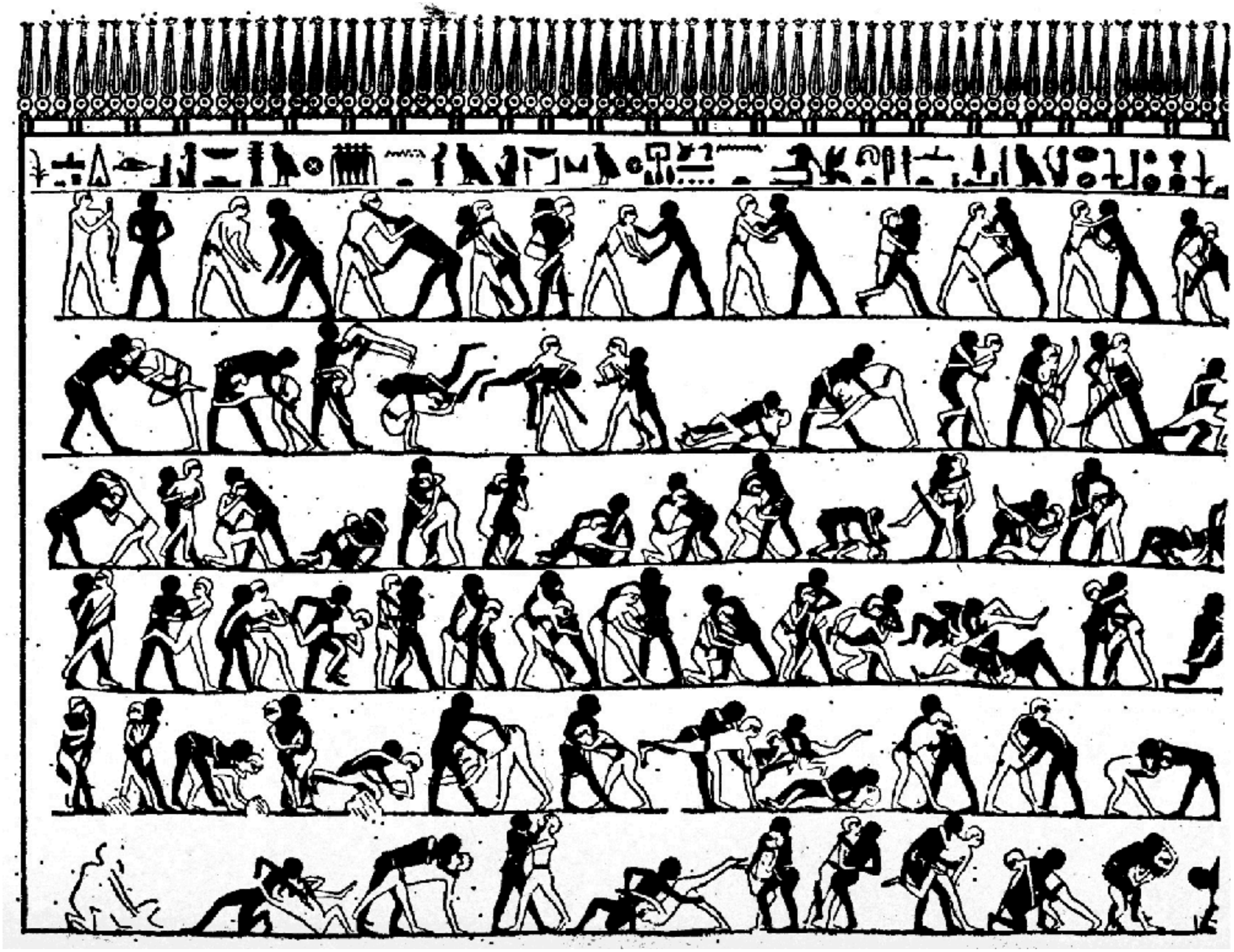
Increasing the complexity of our models

...but what about motion?

First Animation

(Shahr-e Sukhteh, Iran 3200 BCE)

History of Animation



(Tomb of Khnumhotep, Egypt 2400 BCE)

History of Animation

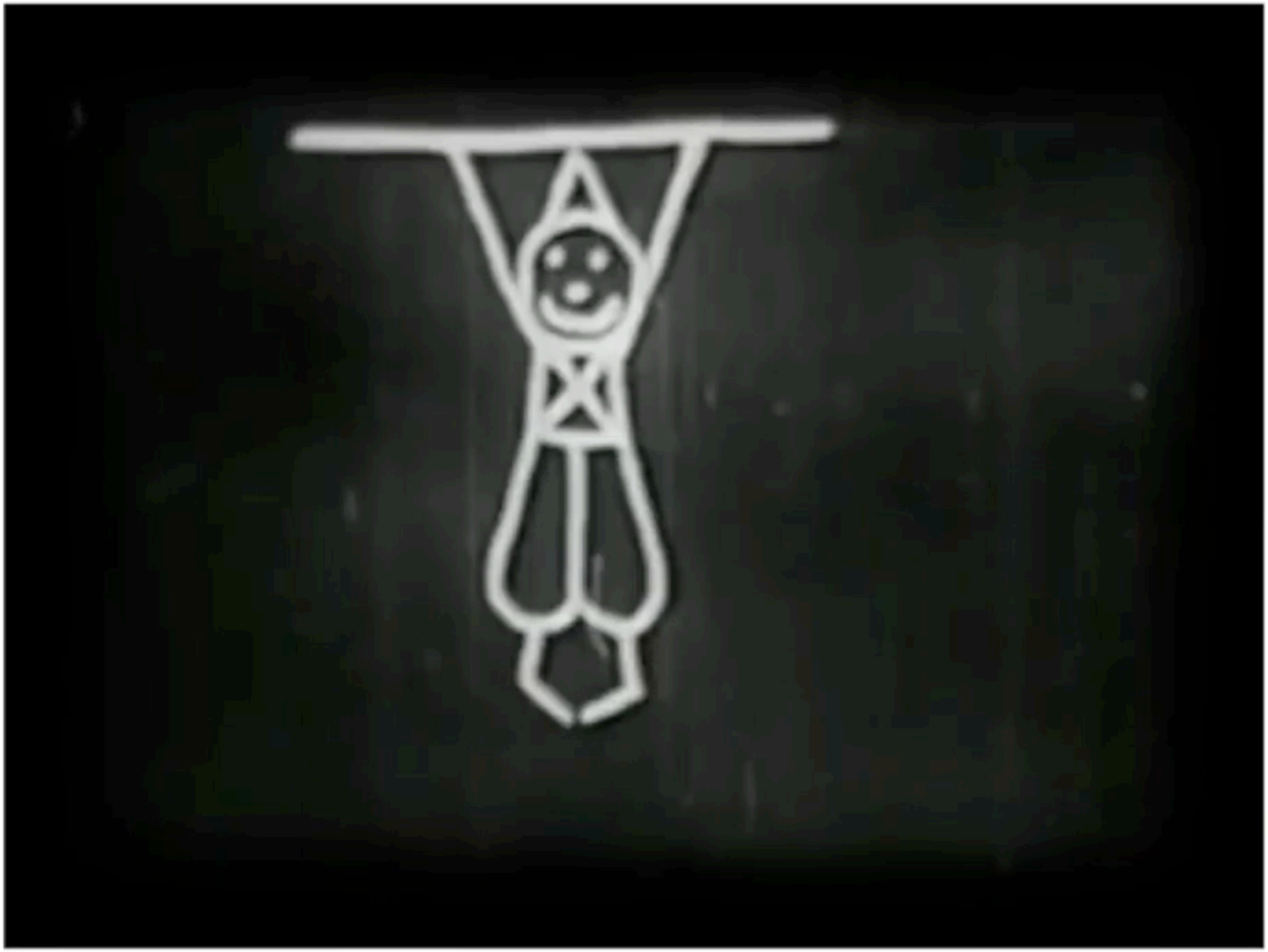
(Phenakistoscope, 1831) 动画装置

First Film

- Originally used as scientific tool rather than for entertainment
- Critical technology that accelerated development of animation

Eadweard Muybridge, "Sallie Gardner" (1878)

First Animation on Film



Emile Cohl, "Fantasmagorie" (1908) 现代动画之父

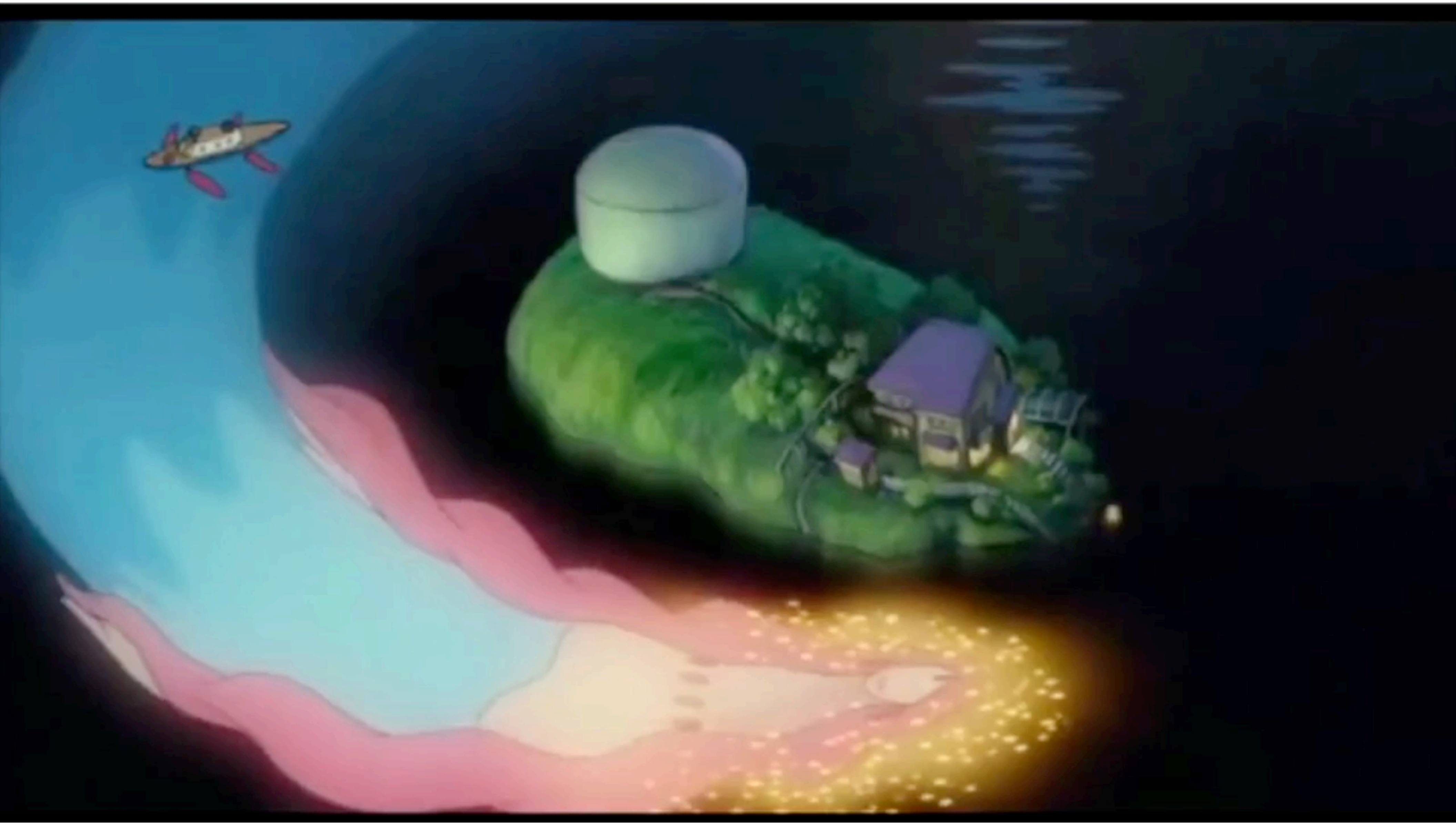
First Feature-length Animation

Lotte Reiniger, "Die Abenteuer des Prinzen Achmed" (1926)

First Hand-drawn Feature-length Animation

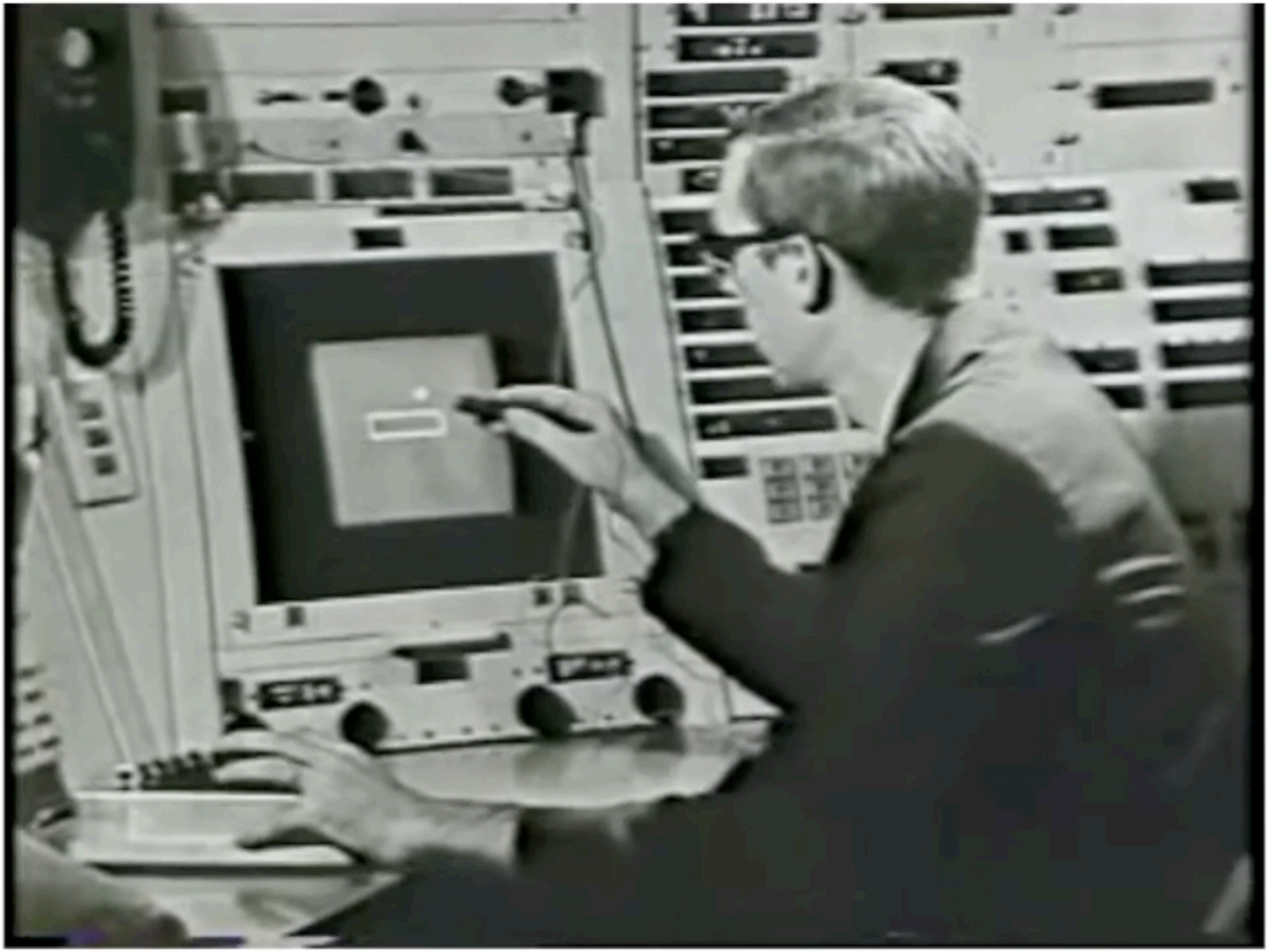
Disney, “Snow White and the Seven Dwarves” (1937)

Hand-drawn Animation - Present Day



Studio Ghibli, “Ponyo” (2008)

First Digital-Computer-Generated Animation



Ivan Sutherland, “Sketchpad” (1963)

Early Computer Animation

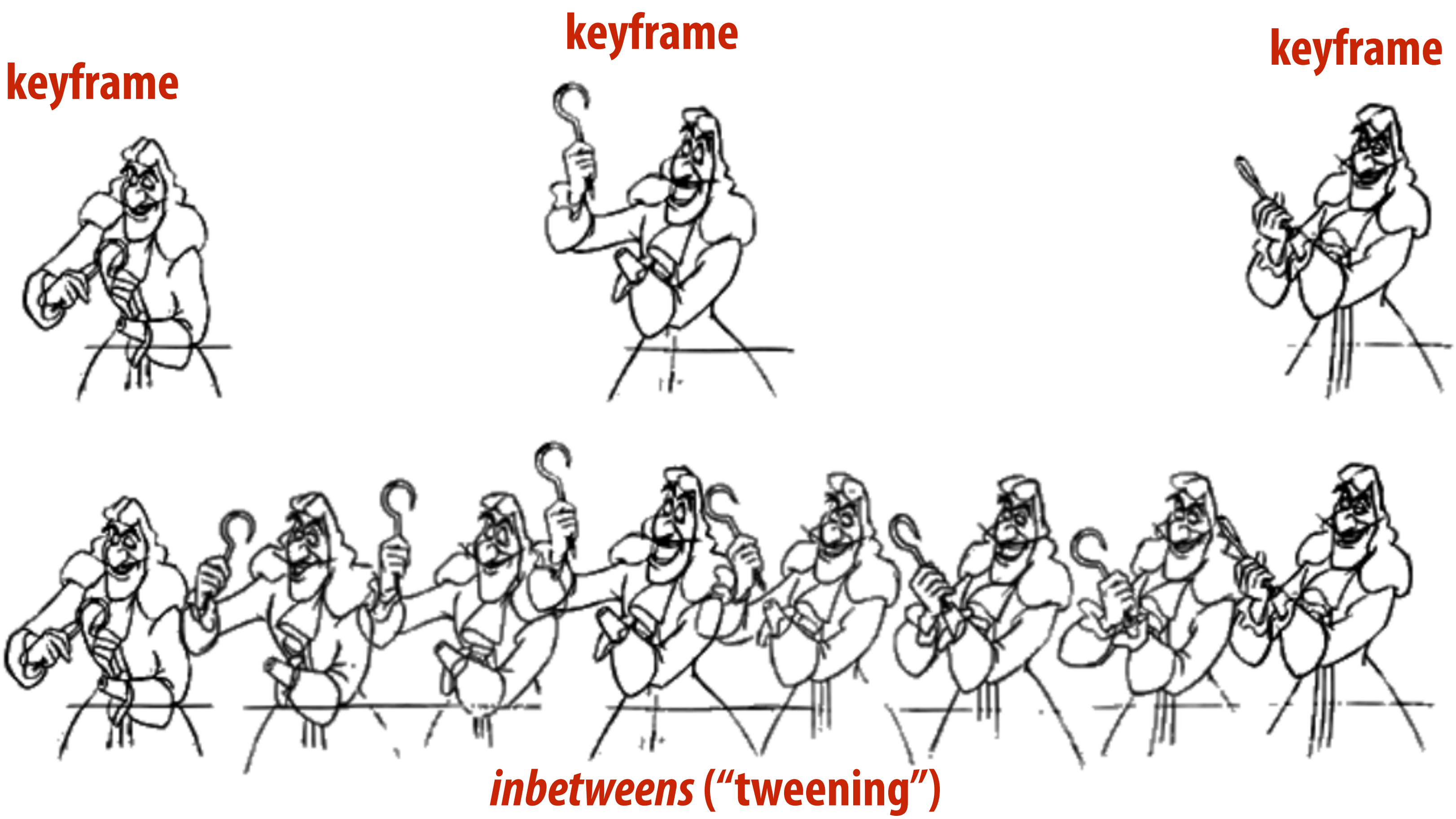
Ed Catmull & Fred Park, "Computer Animated Faces" (1972)

First CG Feature Film

Pixar, “Toy Story” (1995)

Generating Motion (Hand-Drawn)

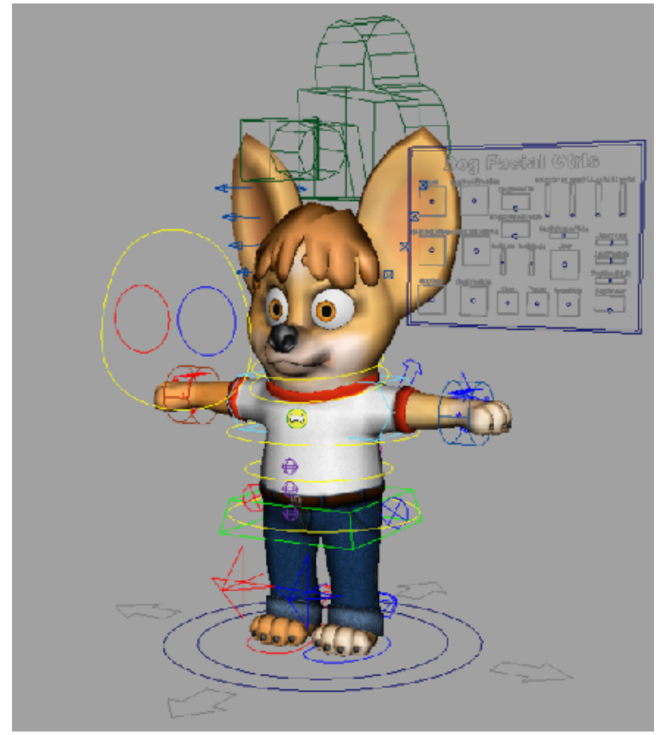
- Senior artist draws keyframes
- Assistant draws in-betweens
- Tedious / labor intensive (opportunity for technology!)

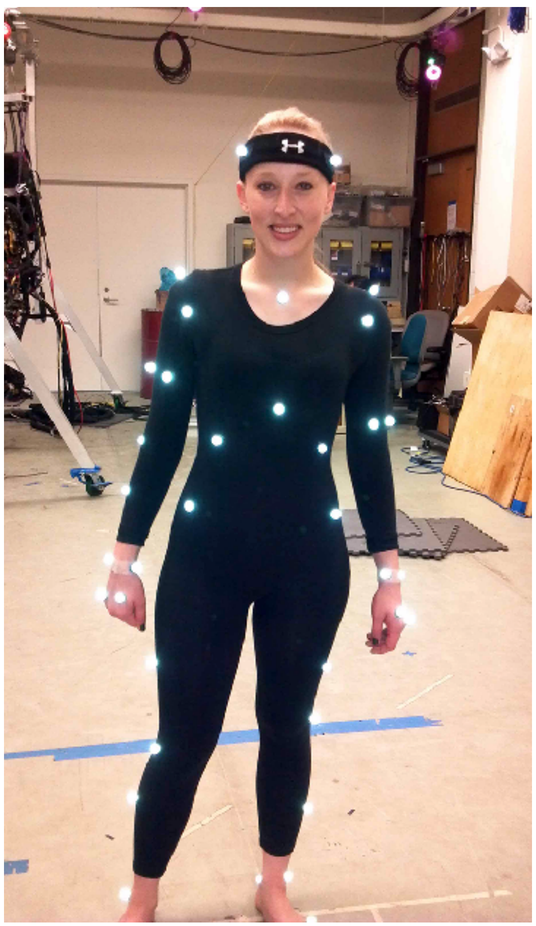


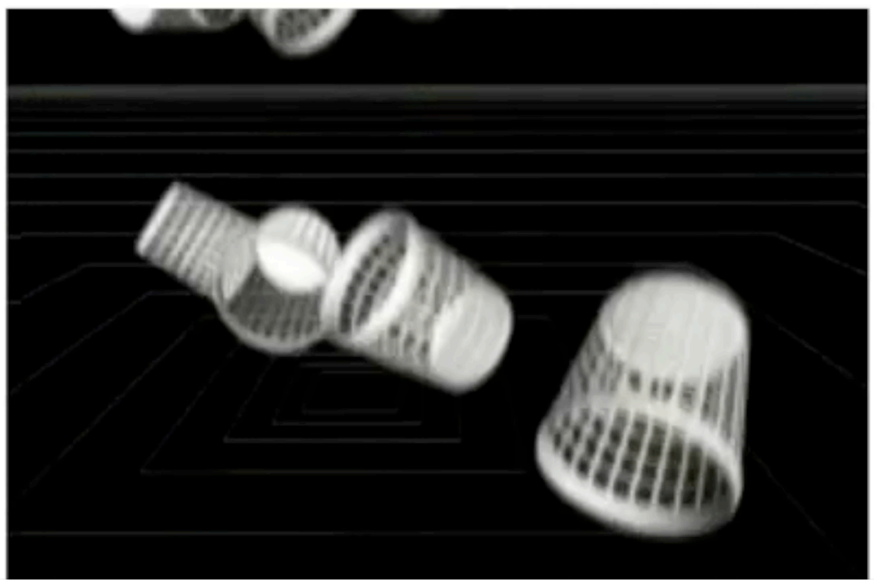
How do we describe motion on a computer?

Basic Techniques in Computer Animation

- **Artist-directed (e.g., keyframing)**
- **Data-driven (e.g., motion capture)**
- **Procedural (e.g., simulation)**

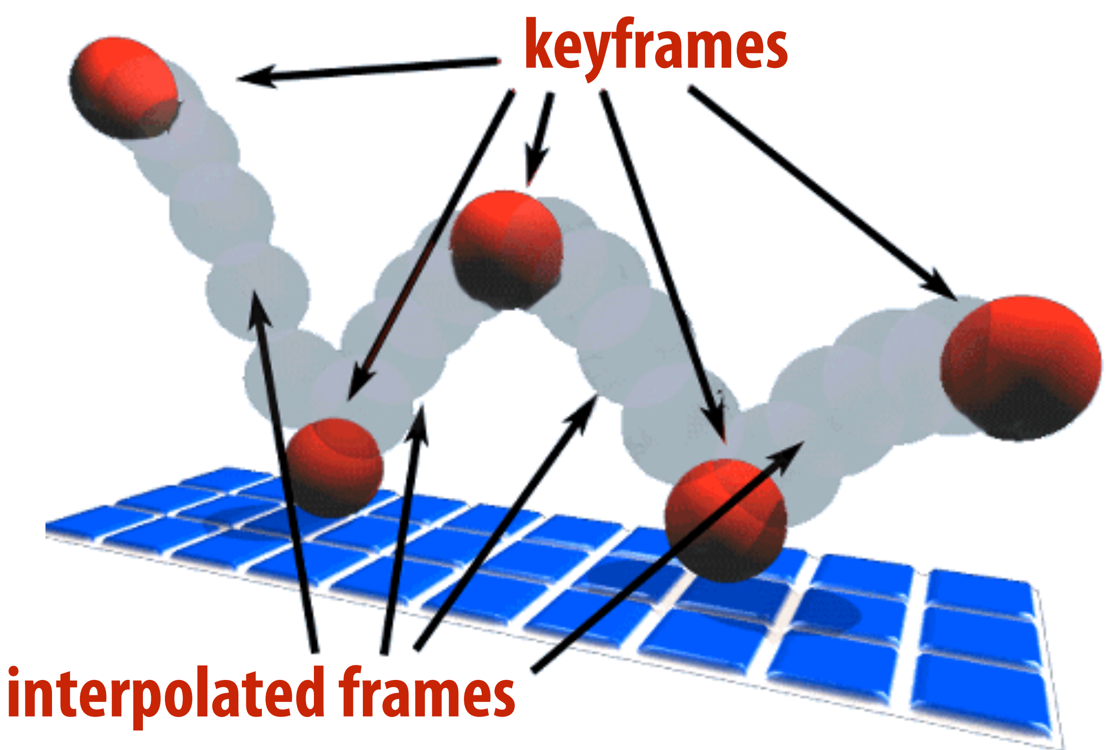






Keyframing

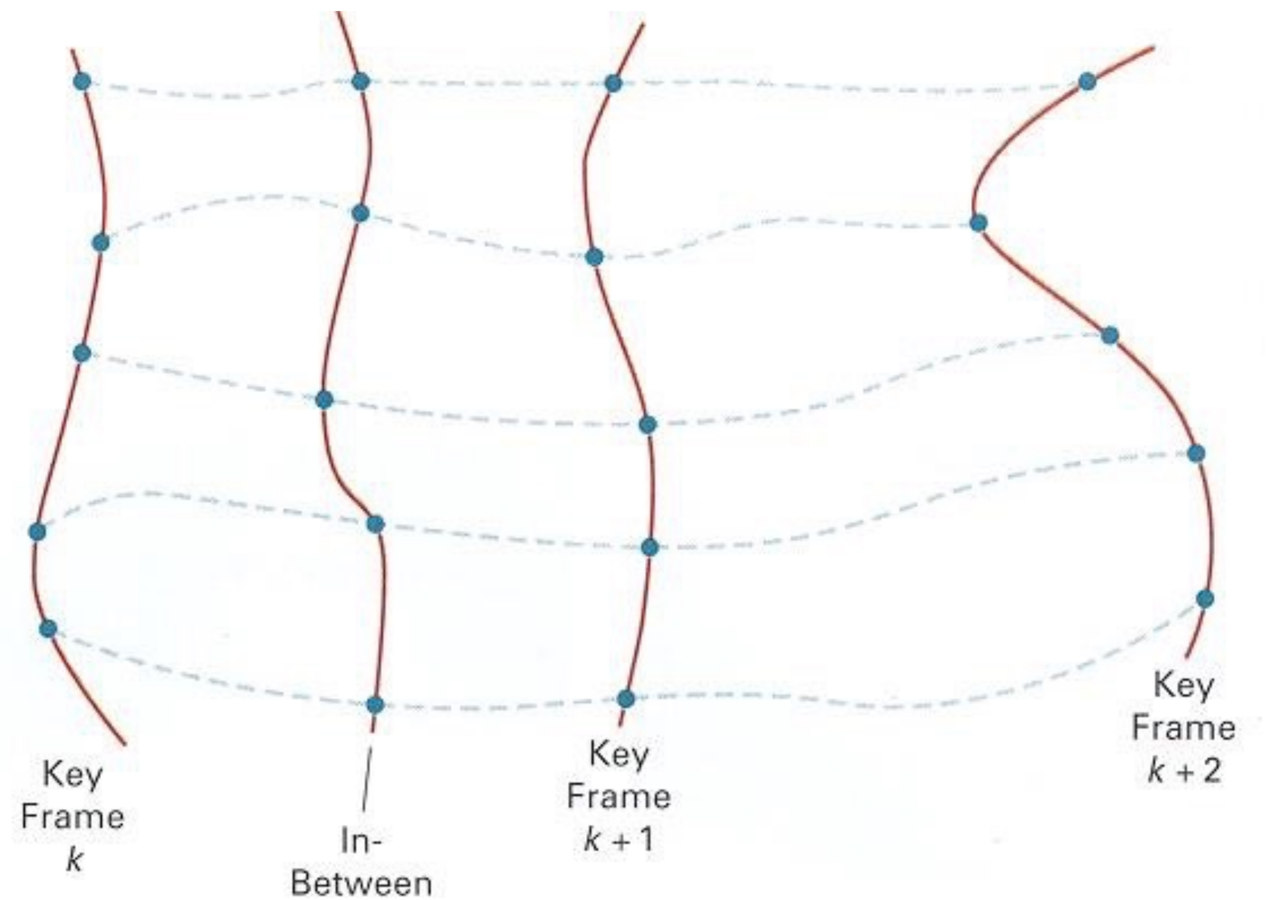
- Basic idea:
  - Specify important events only
  - Computer fills in the rest via interpolation/approximation
- "Events" don't have to be position
  - Could be color, light intensity, camera zoom,...



How do you interpolate data?

Keyframe Interpolation

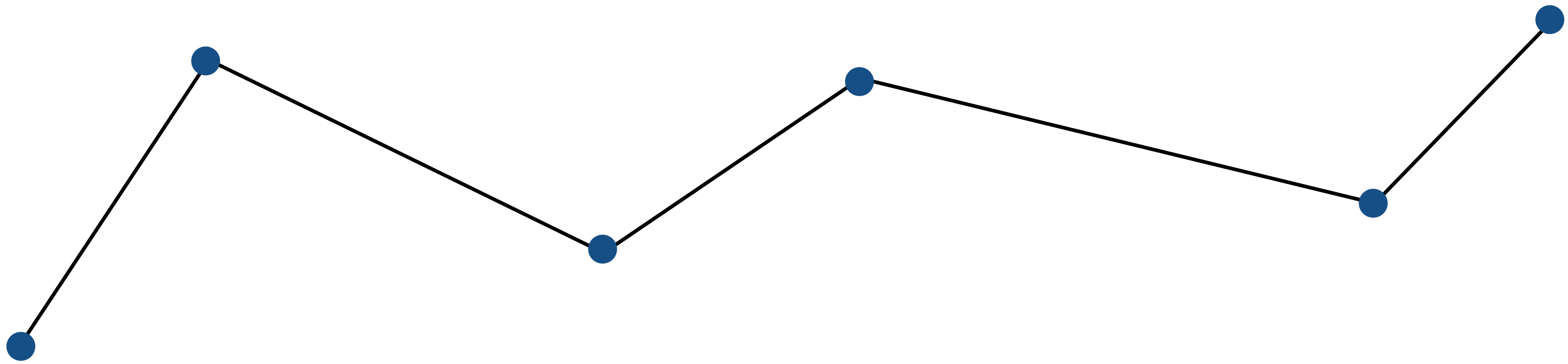
Think of each frame as a vector of parameter values



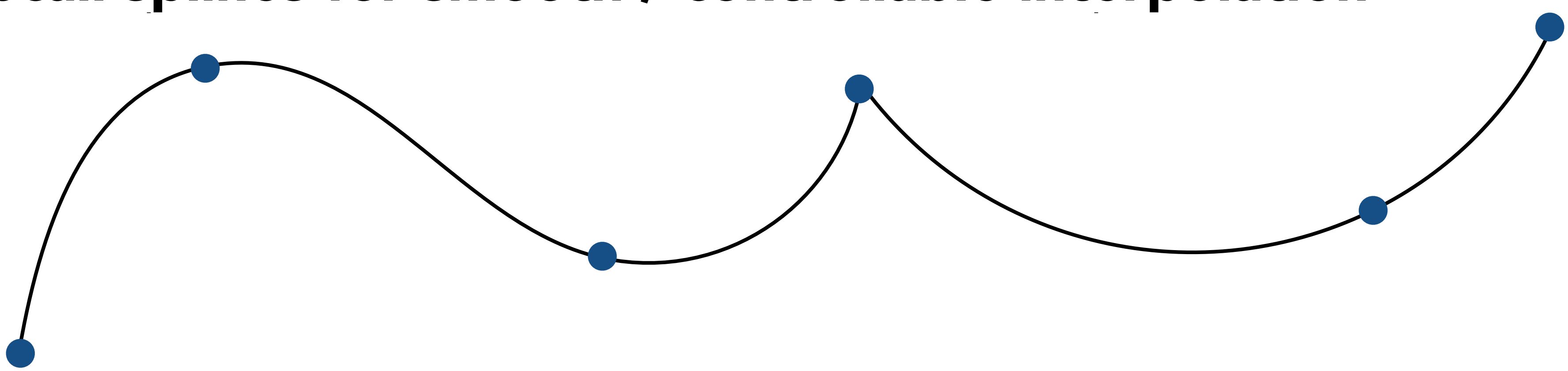
Hearn, Baker and Carithers, Figure 16.11

Keyframe Interpolation of Each Parameter

Linear interpolation usually not good enough



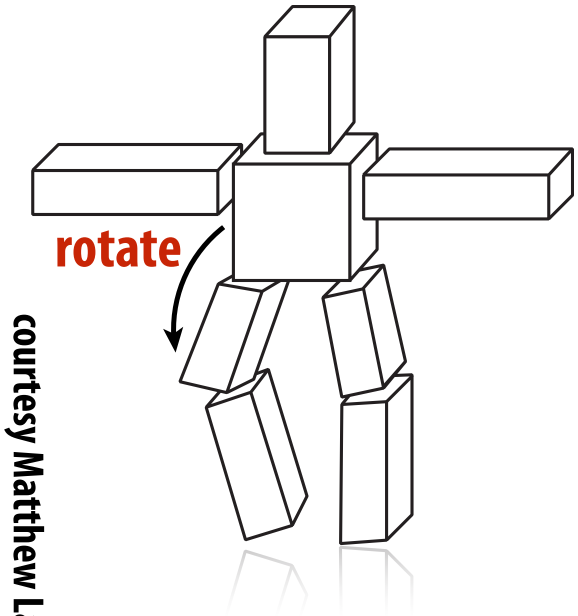
Recall splines for smooth / controllable interpolation



Character Animation

- Scene graph/kinematic chain: scene as tree of transformations
- E.g., in our "cube man," configuration of a leg might be expressed as a rotation relative to body
- Animate by interpolating transformations

courtesy Matthew Lailler



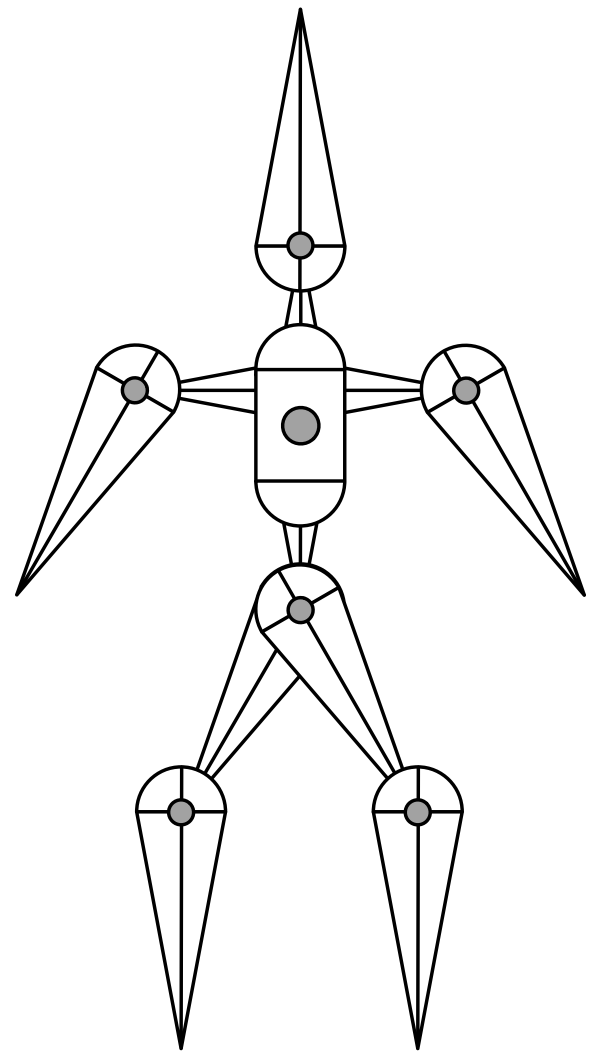
Forward Kinematics

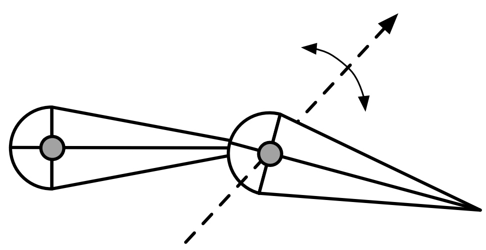
前向运动学

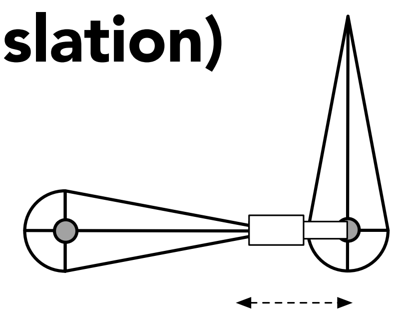
Forward Kinematics

- Articulated skeleton
  - Topology (what's connected to what)
  - Geometric relations from joints
  - Tree structure (in absence of loops)

- Joint types
  - Pin (1D rotation)
  - Ball (2D rotation)
  - Prismatic joint (translation)

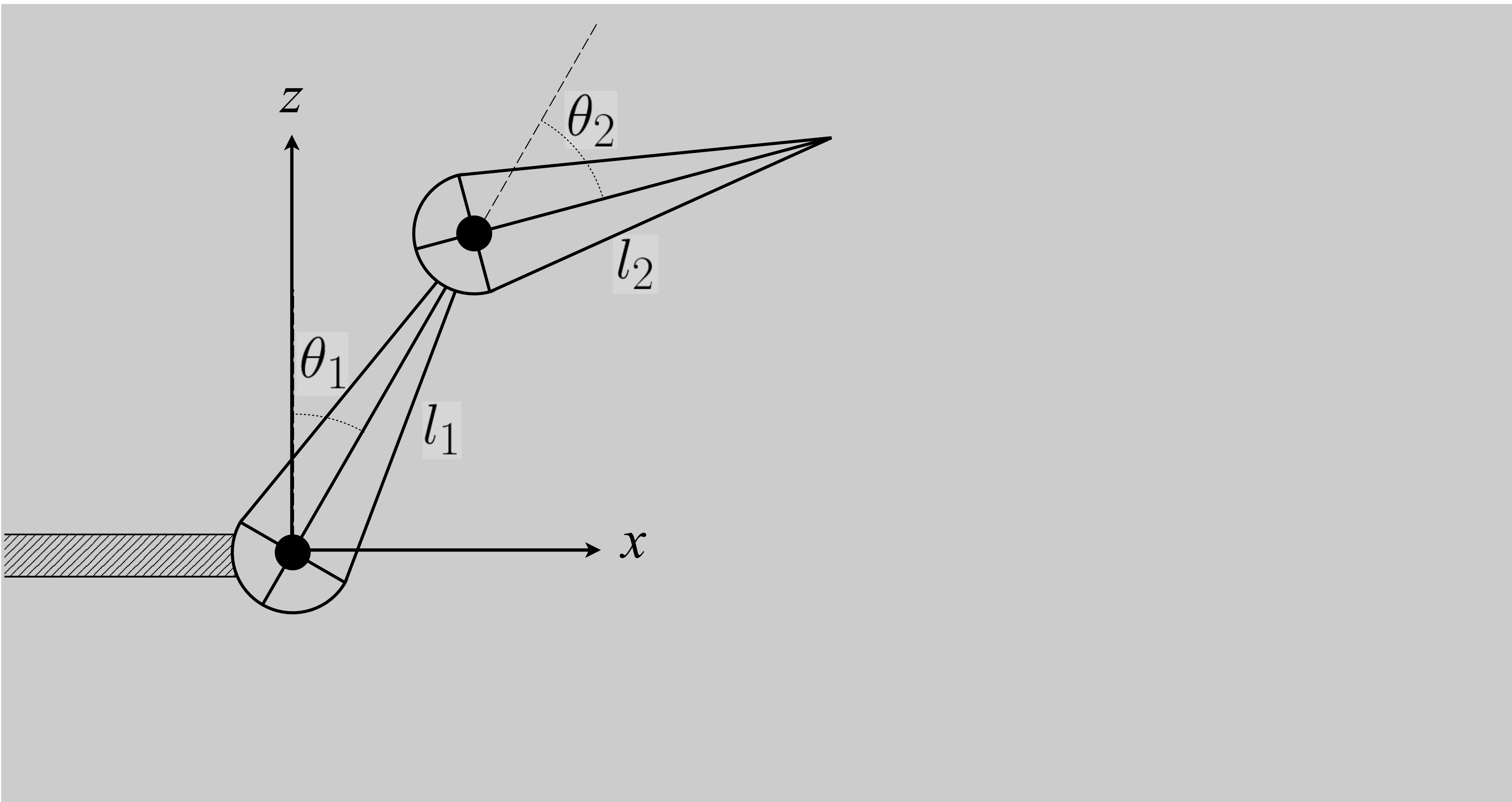






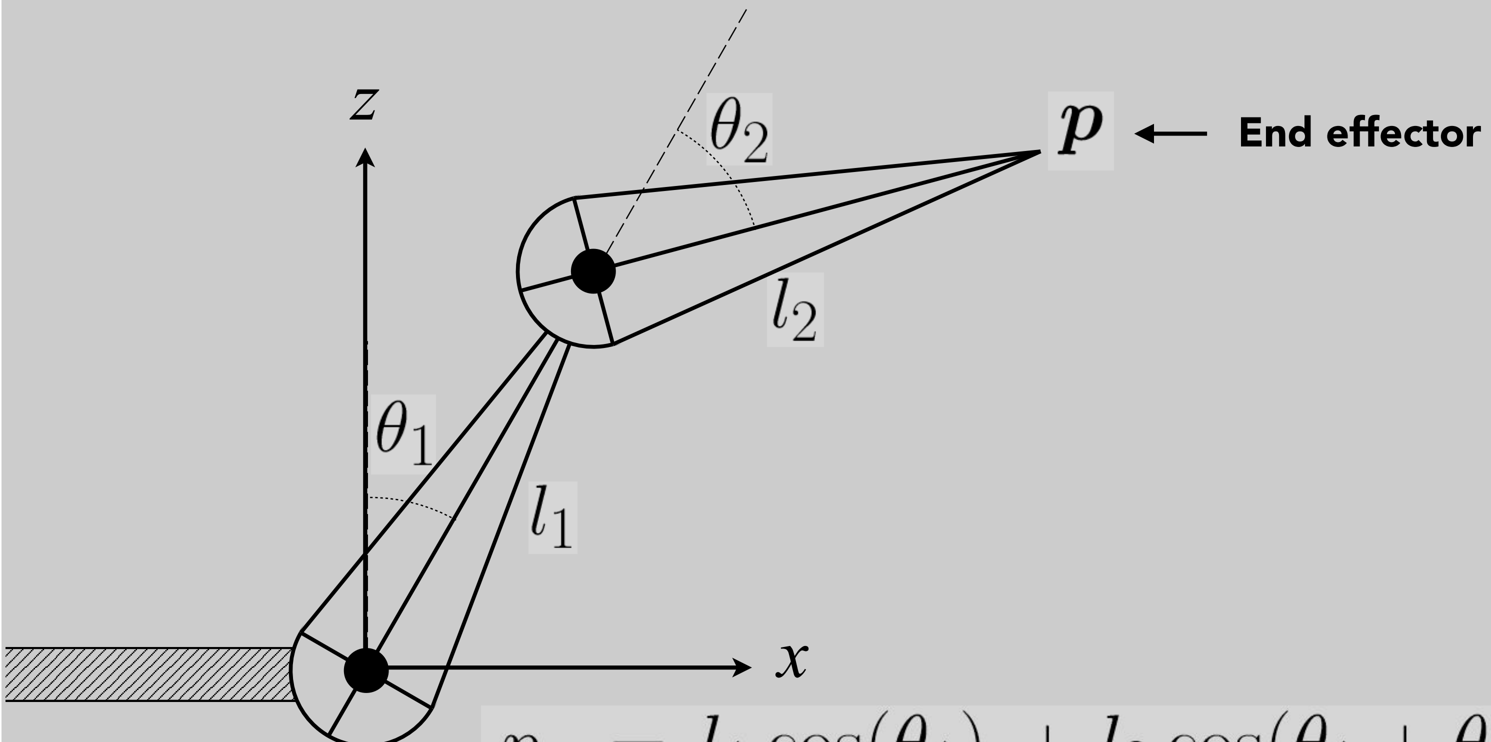
Forward Kinematics

Example: simple two segment arm in 2D



Forward Kinematics

Animator provides angles, and computer determines position p of end effector

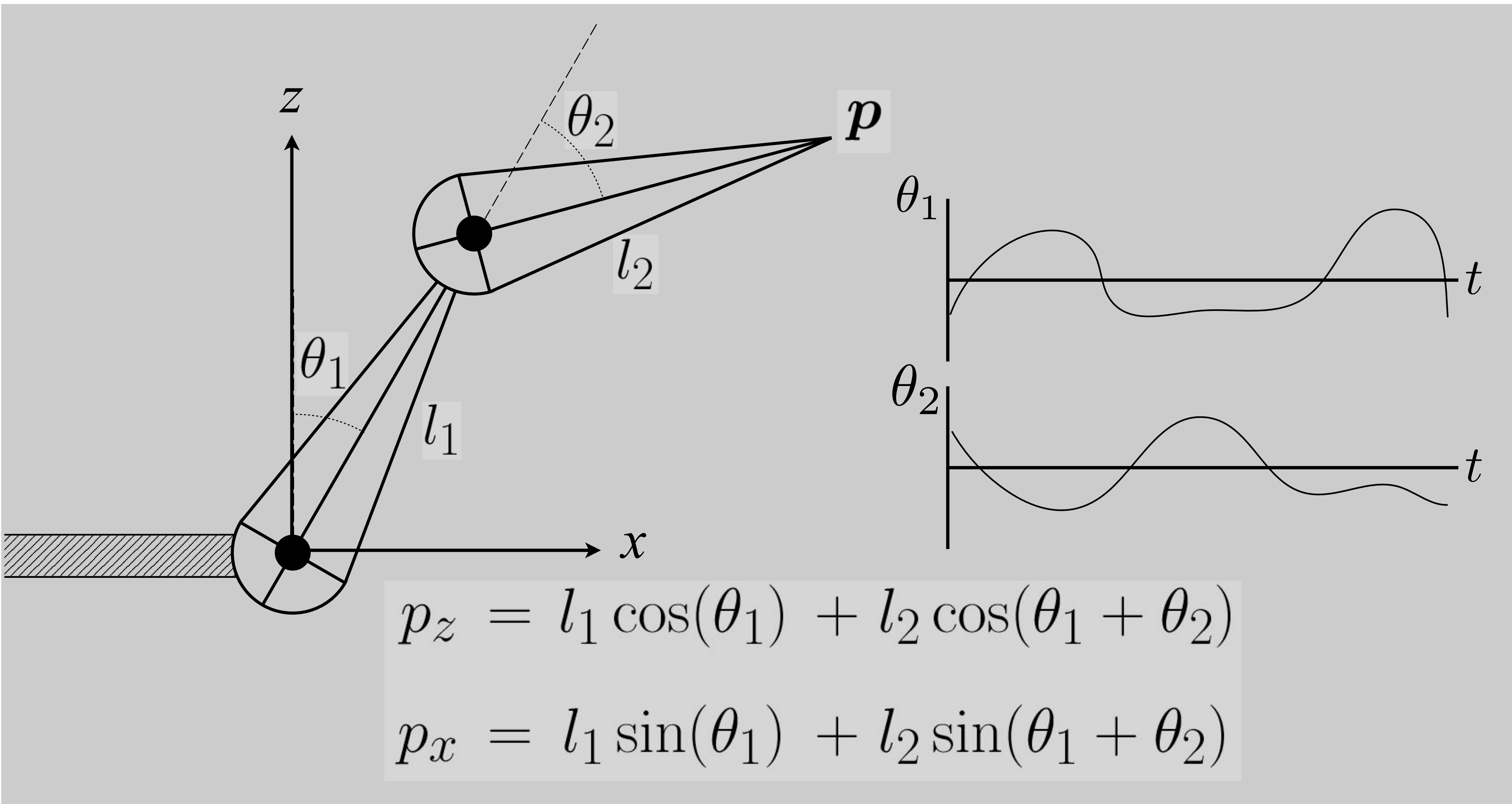


$$p_z = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2)$$

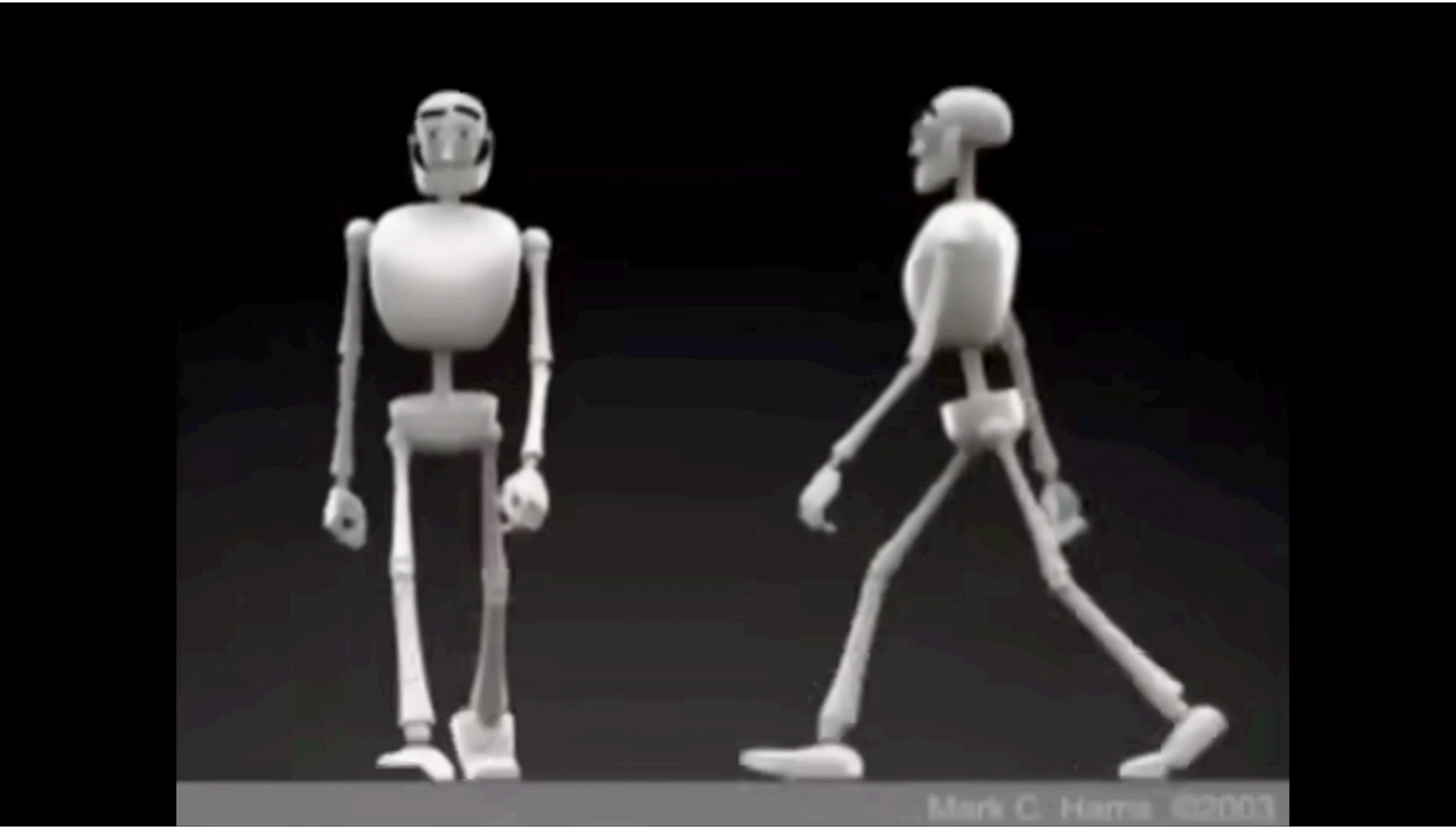
$$p_x = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2)$$

Forward Kinematics

Animator is described as angle parameter values as a function of time

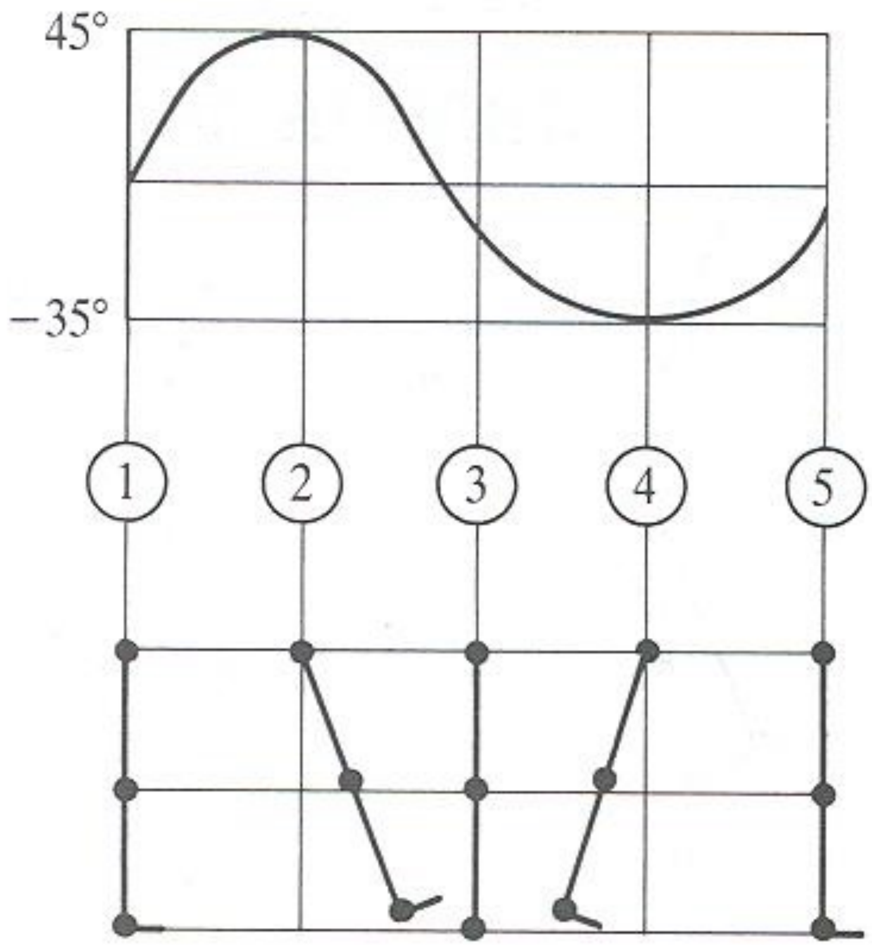


Example Walk Cycle



Example Walk Cycle

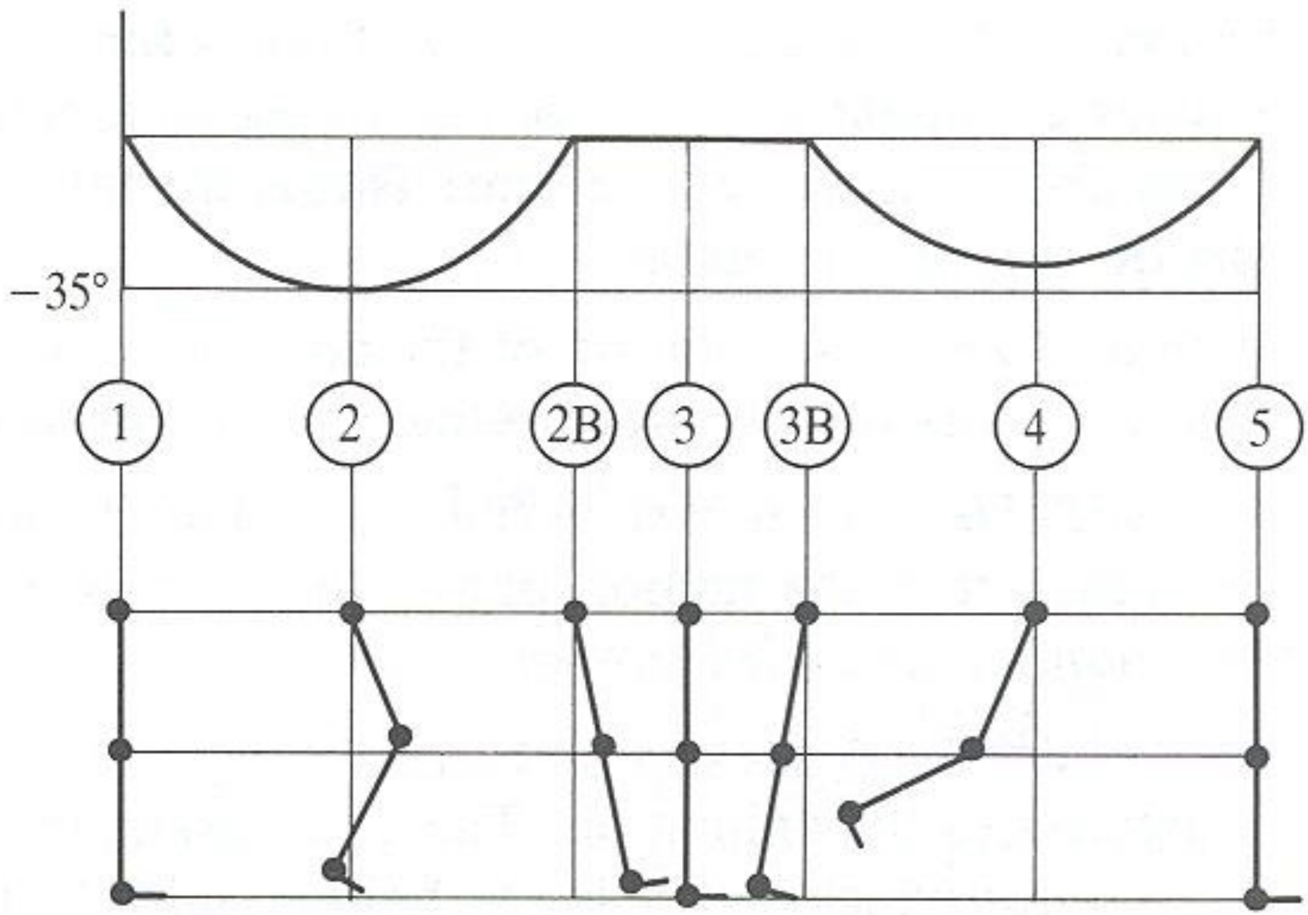
Hip joint angle



Watt & Watt

Example Walk Cycle

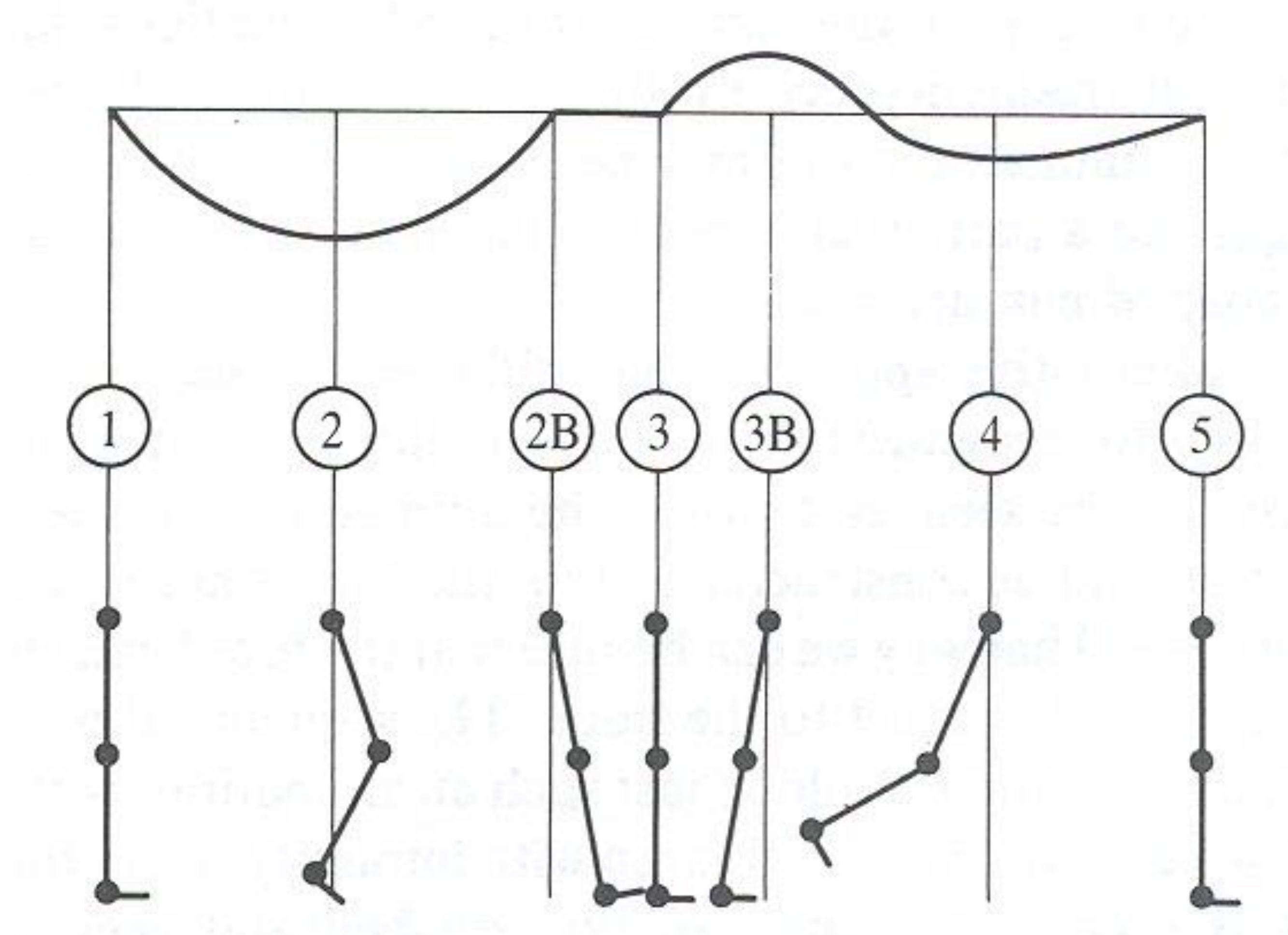
Knee joint angle



Watt & Watt

Example Walk Cycle

Ankle joint angle



Watt & Watt

Kinematics Pros and Cons

Strengths

- Direct control is convenient
- Implementation is straightforward

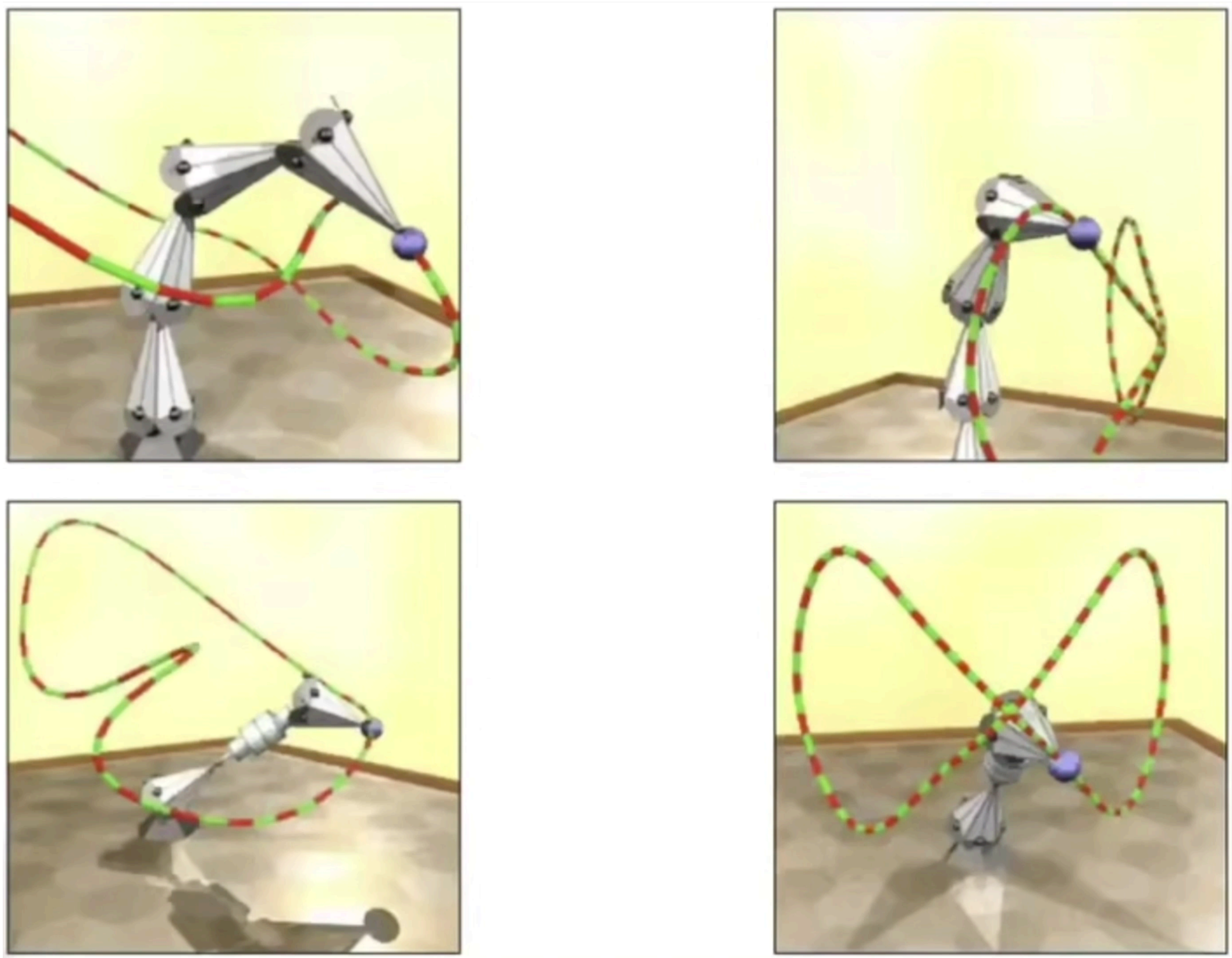
Weakness

- Animation maybe inconsistent with physics
- Time consuming for artists

Inverse Kinematics

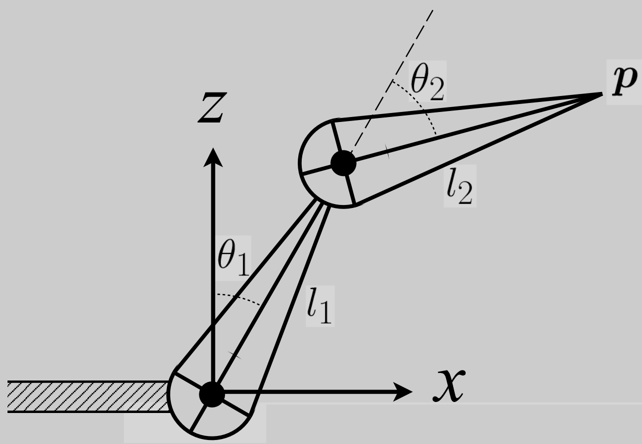
反向运动学

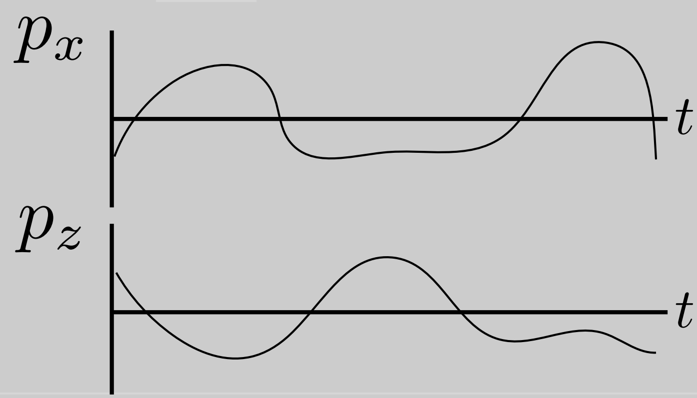
Inverse Kinematics



Inverse Kinematics

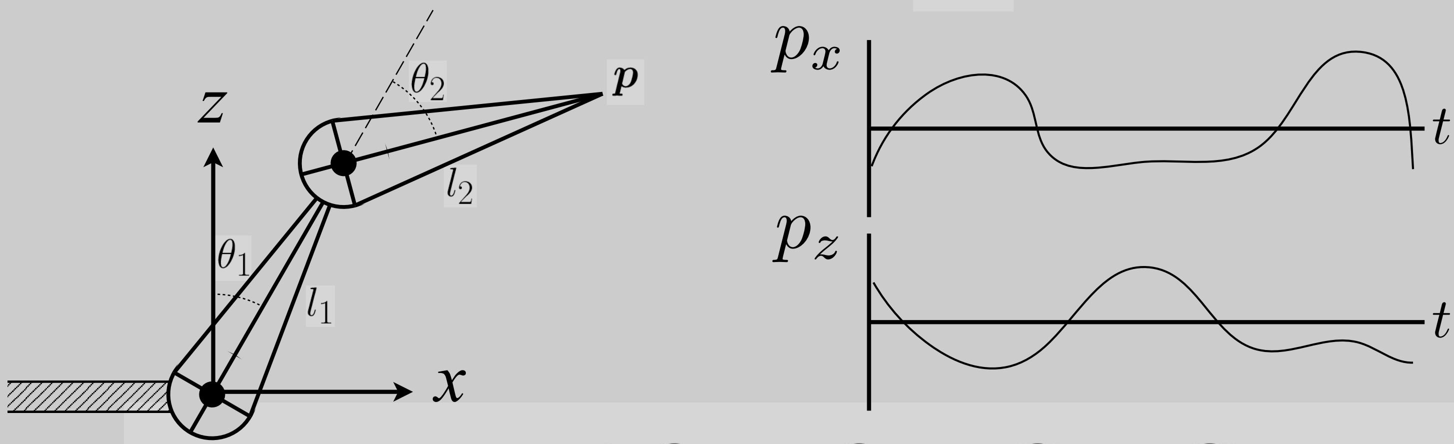
Animator provides position of end-effector, and computer must determine joint angles that satisfy constraints





Inverse Kinematics

Direct inverse kinematics: for two-segment arm, can solve for parameters analytically



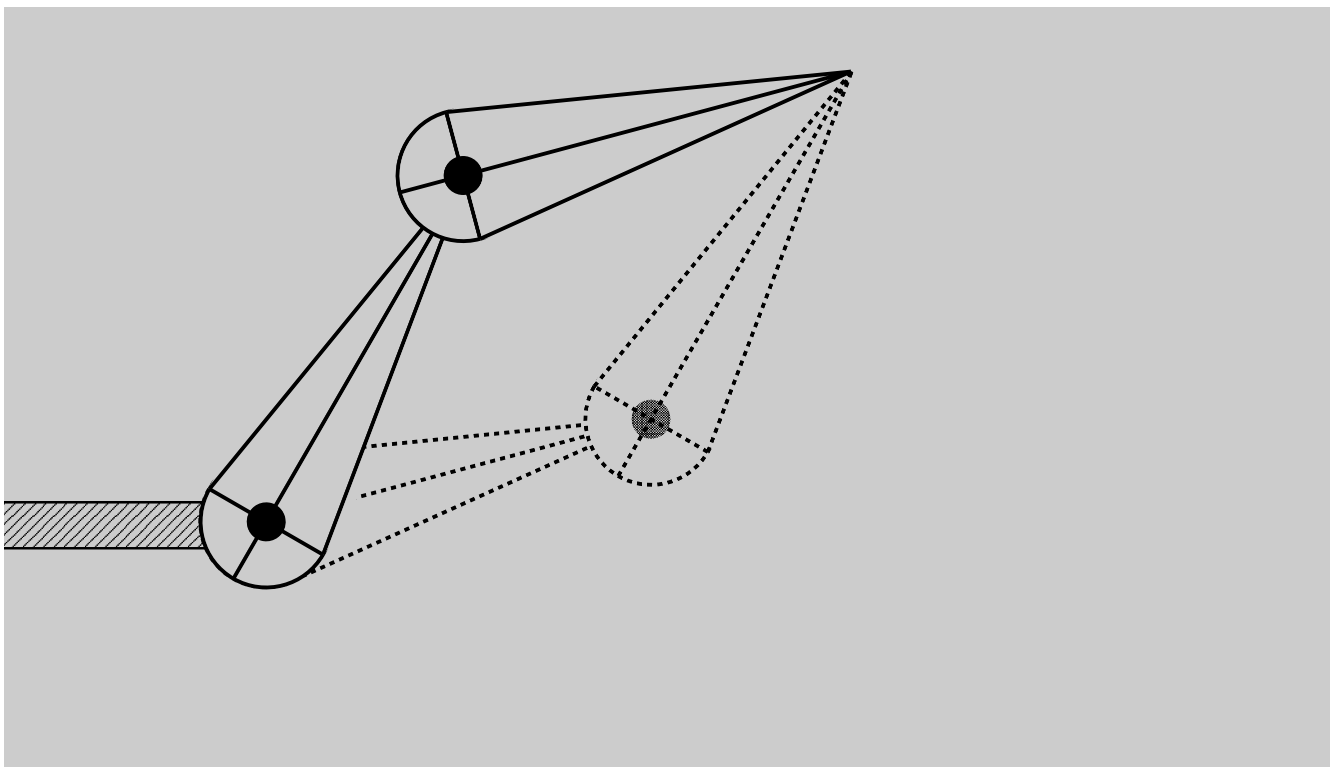
$$\theta_2 = \cos^{-1}\left(\frac{p_z^2 + p_x^2 - l_1^2 - l_2^2}{2l_1 l_2}\right)$$

$$\theta_1 = \frac{-p_z l_2 \sin(\theta_2) + p_x(l_1 + l_2\cos(\theta_2))}{p_x l_2 \sin(\theta_2) + p_z(l_1 + l_2\cos(\theta_2))}$$

Inverse Kinematics

Why is the problem hard?

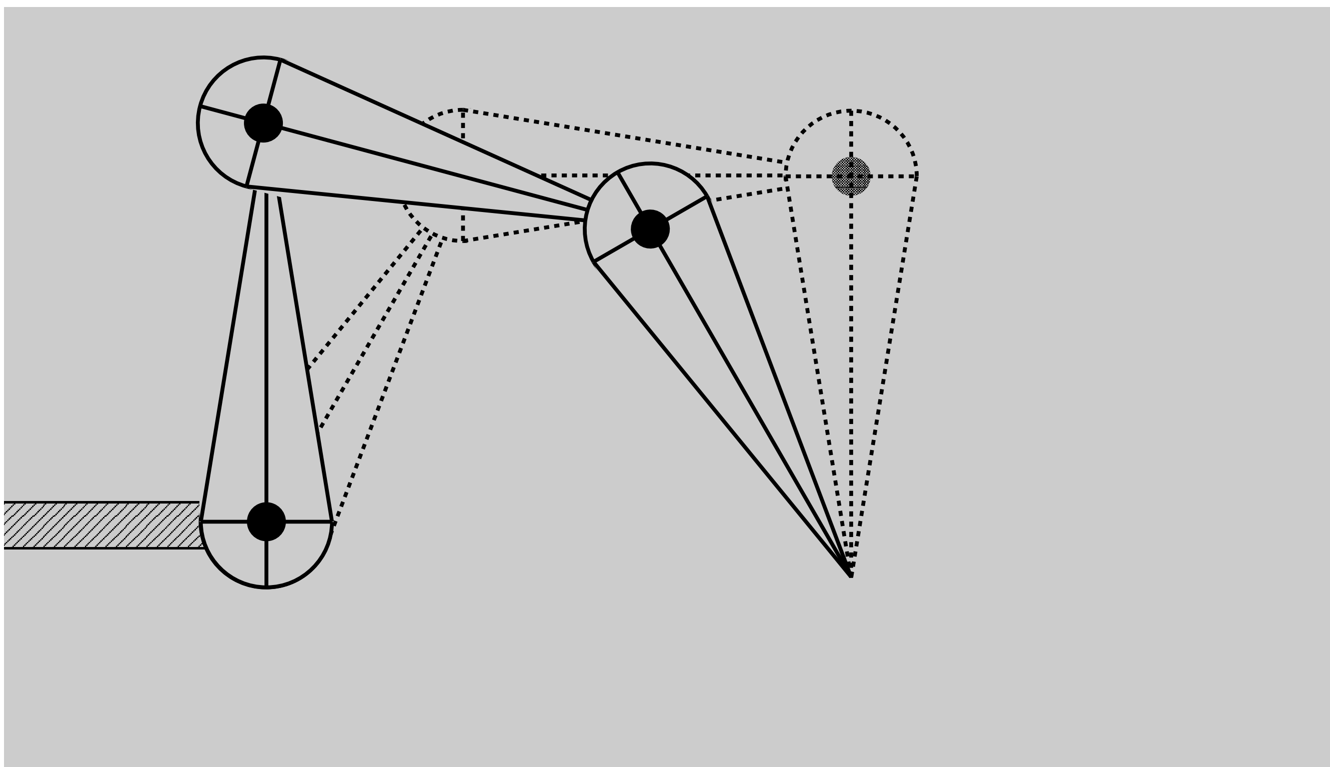
- Multiple solutions in configuration space



Inverse Kinematics

Why is the problem hard?

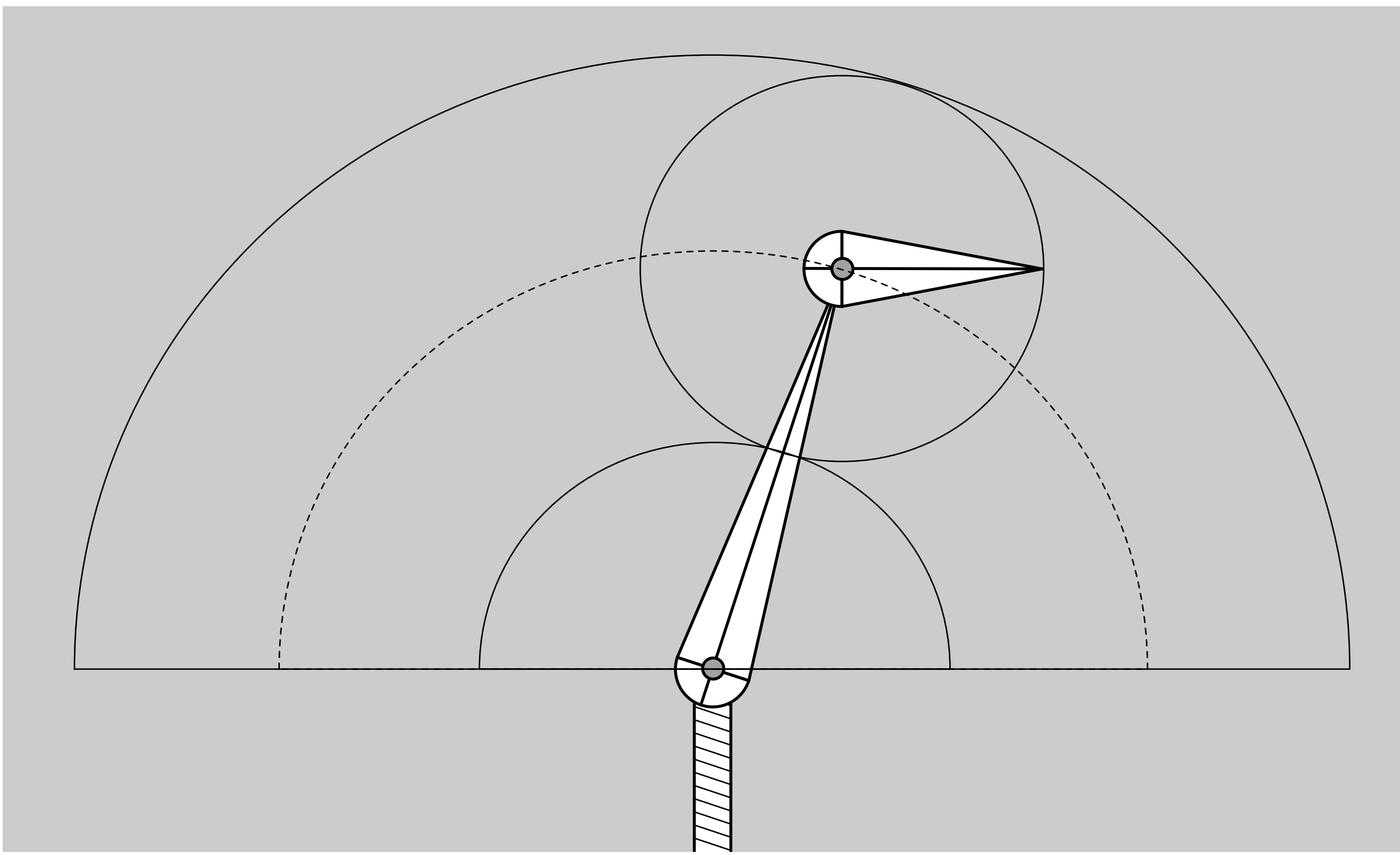
- **Multiple solutions in configuration space**



Inverse Kinematics

Why is the problem hard?

- **Solution may not always exist**



Inverse Kinematics

$$p_z = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2)$$
$$p_x = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2)$$

$$\rightarrow$$

$$\theta_2 = \cos^{-1}\left(\frac{p_z^2 + p_x^2 - l_1^2 - l_2^2}{2l_1 l_2}\right)$$

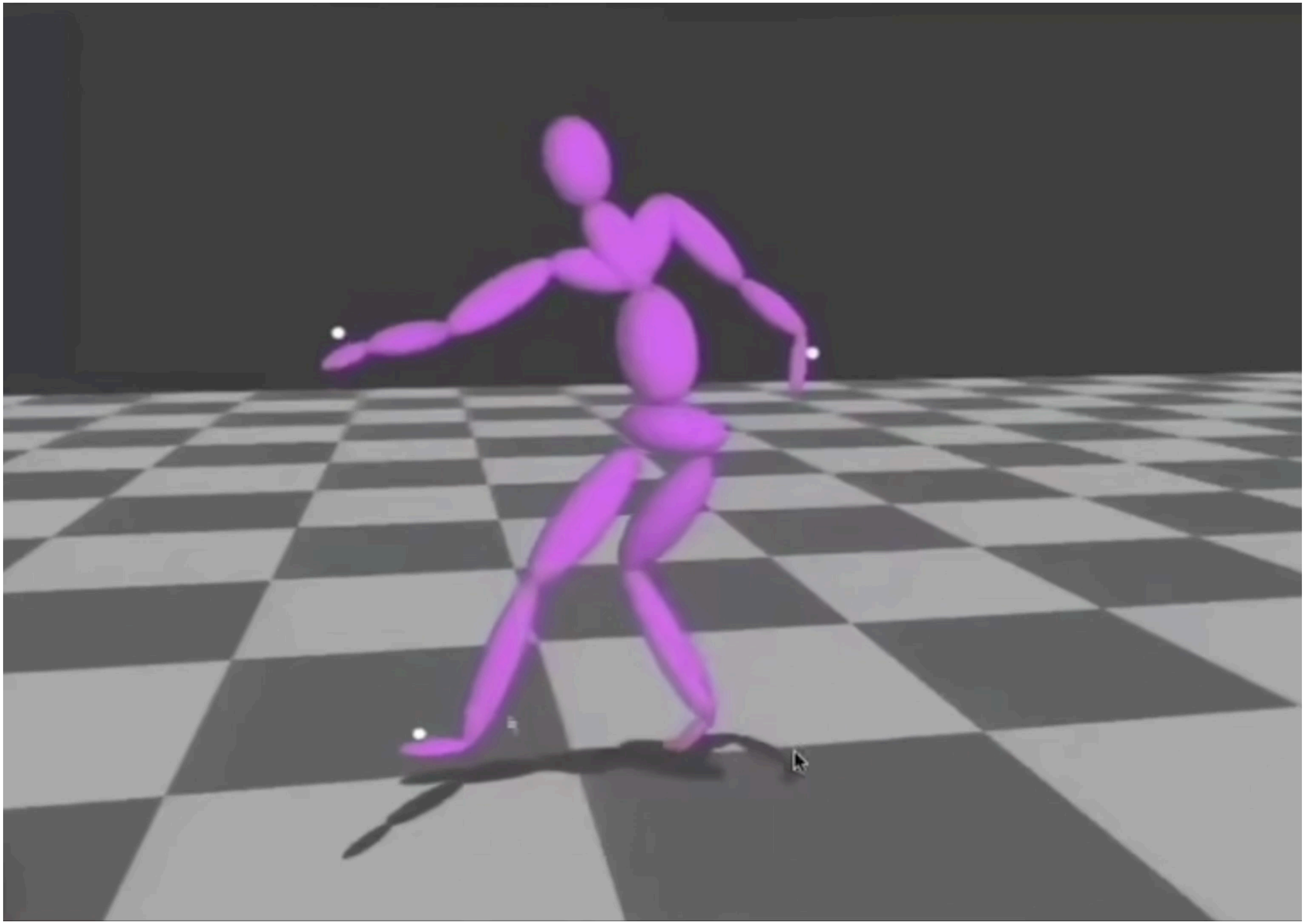
$$\theta_1 = \frac{-p_z l_2 \sin(\theta_2) + p_x(l_1 + l_2\cos(\theta_2))}{p_x l_2 \sin(\theta_2) + p_z(l_1 + l_2\cos(\theta_2))}$$

Analytical solution: 2→N?

Numerical solution to general N-link IK problem

- **Choose an initial configuration**
- **Define an error metric (e.g., square of distance between goal and current position)**
- **Computer gradient of error as function of configuration**
- **Apply gradient descent (or Newton's method, or other optimization procedure)**

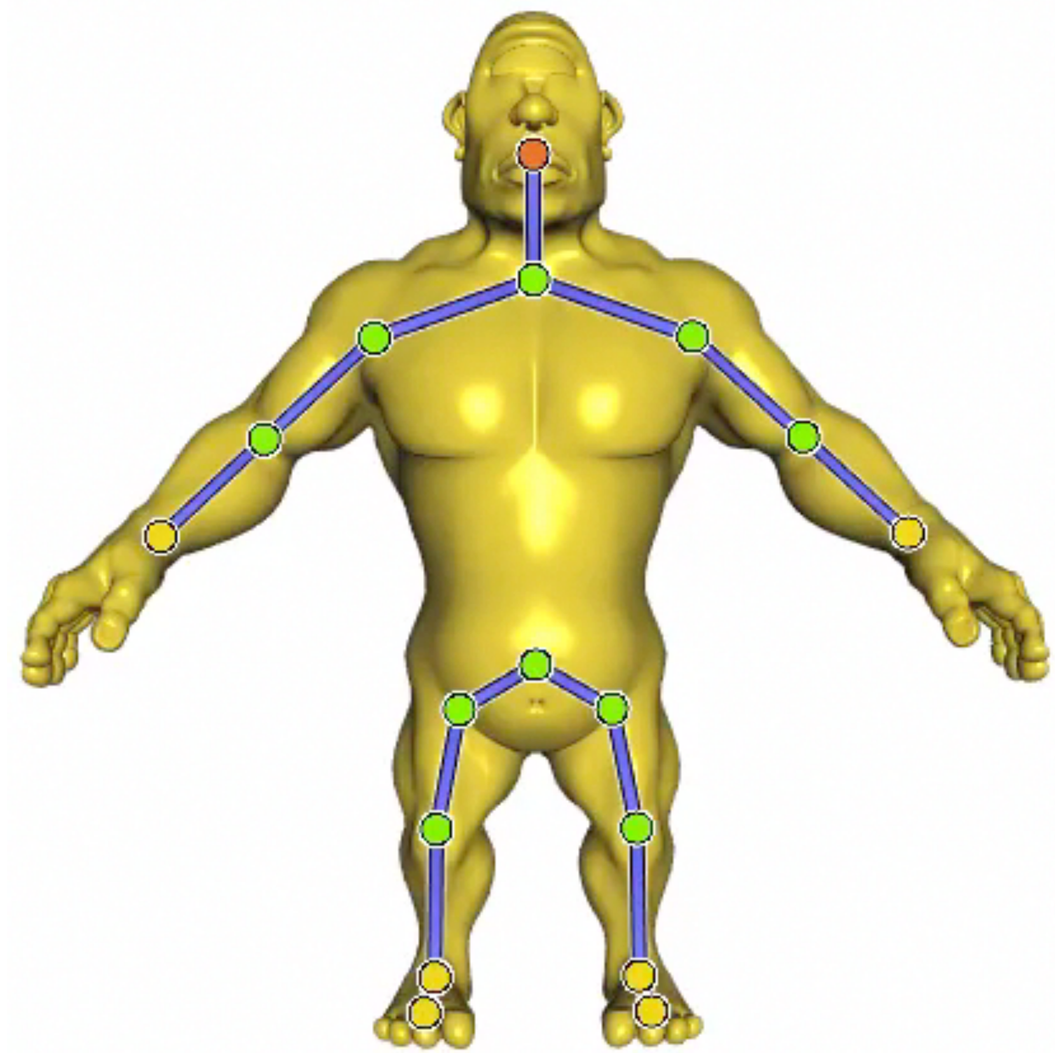
Style-Based IK



Skinning

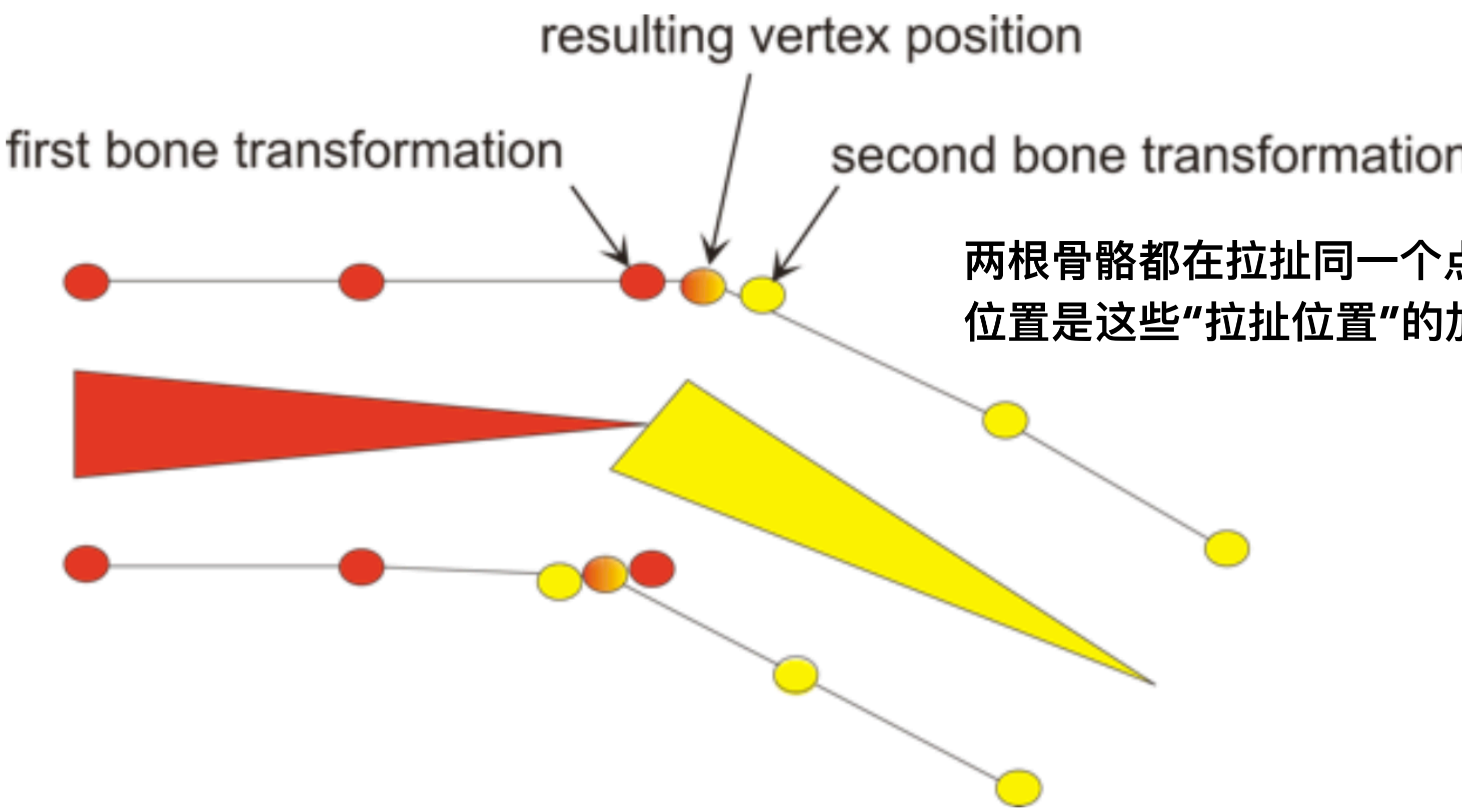
Skinning 蒙皮

Goal: move the surface along with assigned bones or "handles"



Basic Idea

1. Transform each vertex with each bone rigidly
2. Blend the results using weights, or assignments



两根骨骼都在拉扯同一个点，最终的位置是这些“拉扯位置”的加权平均。

Each vertex can be influenced by one or more bones.
Skin weighting determines the percentage of influence for each bone on each vertex.

Common Approach: Linear Blend Skinning (LBS)

Blend contribution linearly.

Super simple to implement. Great for real time.

How much influence this bone has on v (often sparse)

Bone j transformation

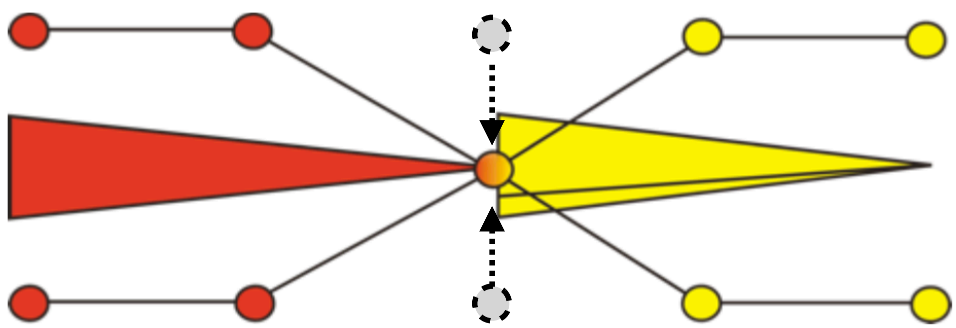
$$\mathbf{v}' = \sum_{j \in H} w_j(\mathbf{v}) \mathbf{T}_j \begin{pmatrix} \mathbf{v} \\ 1 \end{pmatrix}$$

New vertex

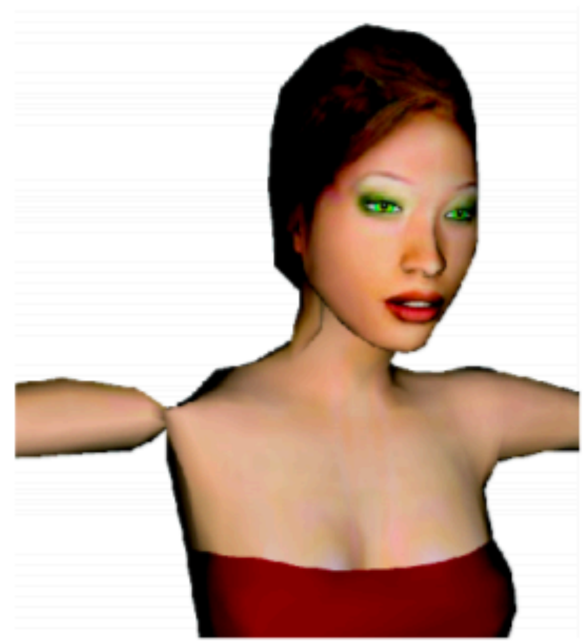
Original vertex

Problems with LBS

When joint rotates 180 degrees

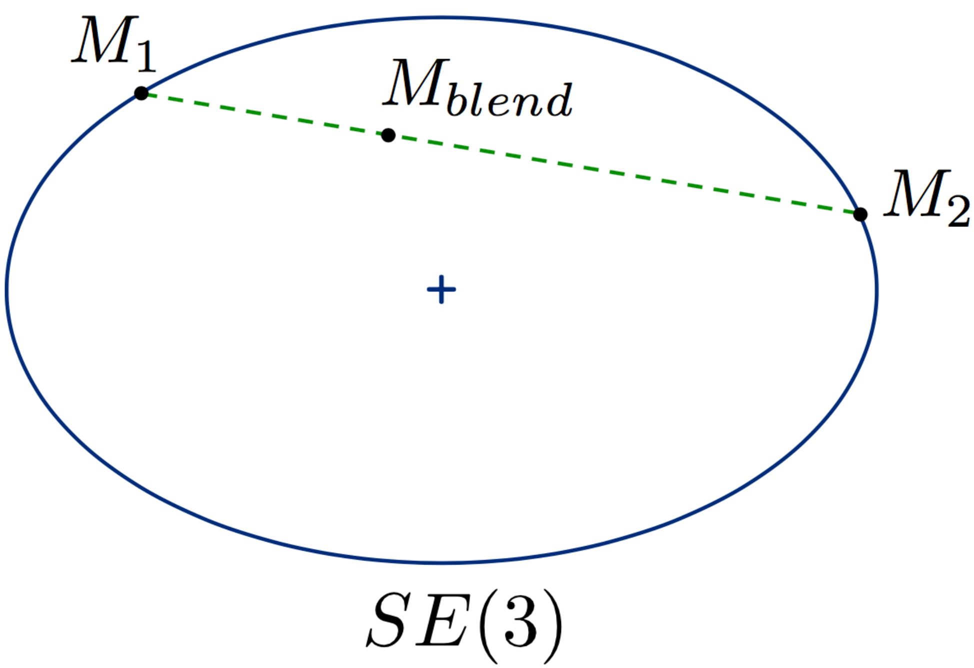


"Candy Wrapper" artifact



Why?

Can’t linearly combine rigid transformations!



旋转空间不是线性空间，线性插值不能保留旋转的性质

旋转矩阵是正交矩阵，两个正交矩阵的加权和不一定正交矩阵，所以也不是旋转矩阵。

Better methods exist like Dual Quaternion Skinning [Kavan et al. TOG 2008] （用四元组来表示旋转）

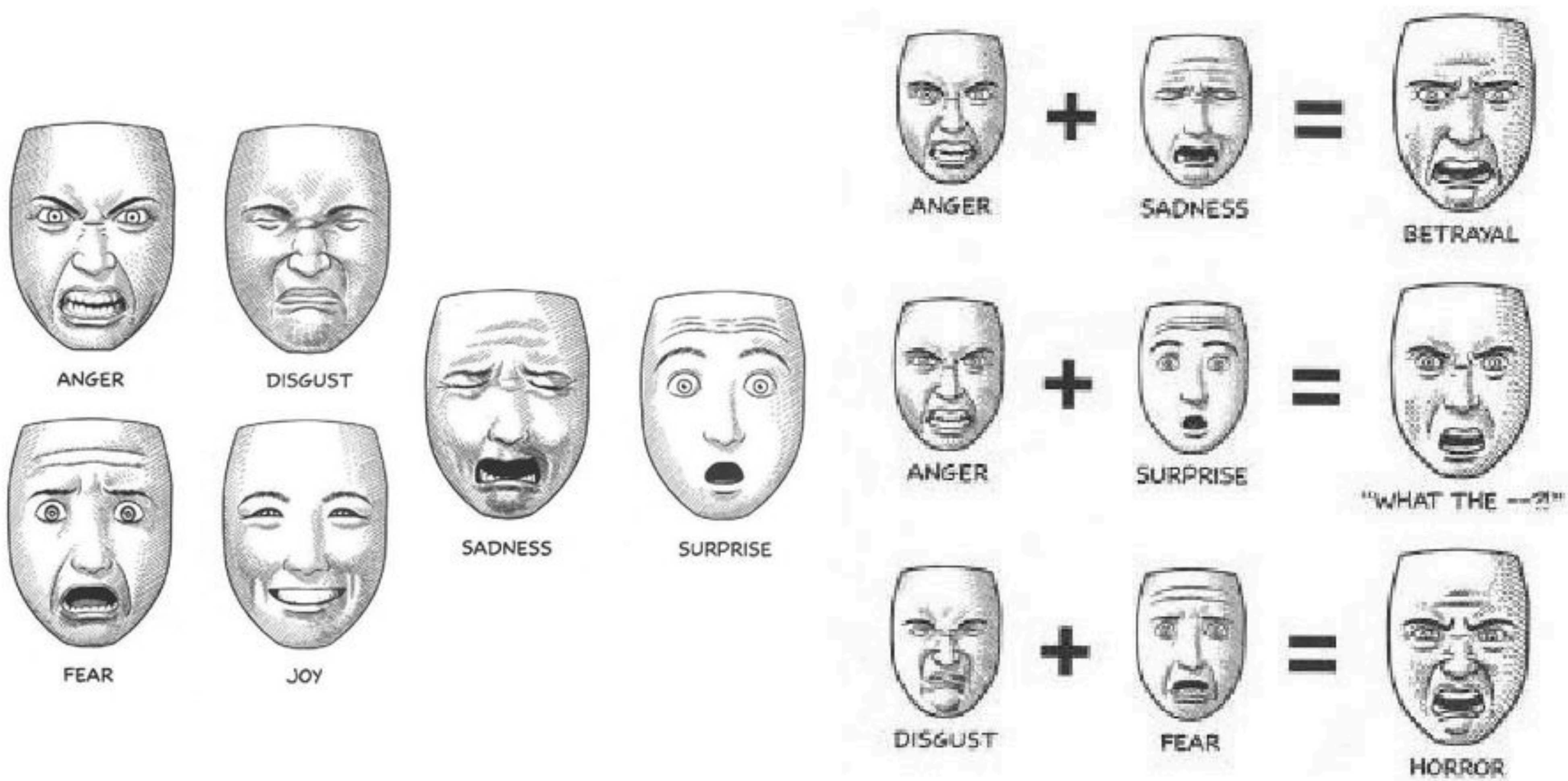
Still LBS is most popular for simplicity and is the de facto industry standard

Blend Shapes

形状插值

Blend Shapes

Not all deformation is from bones.
Interpolate surfaces between key shapes

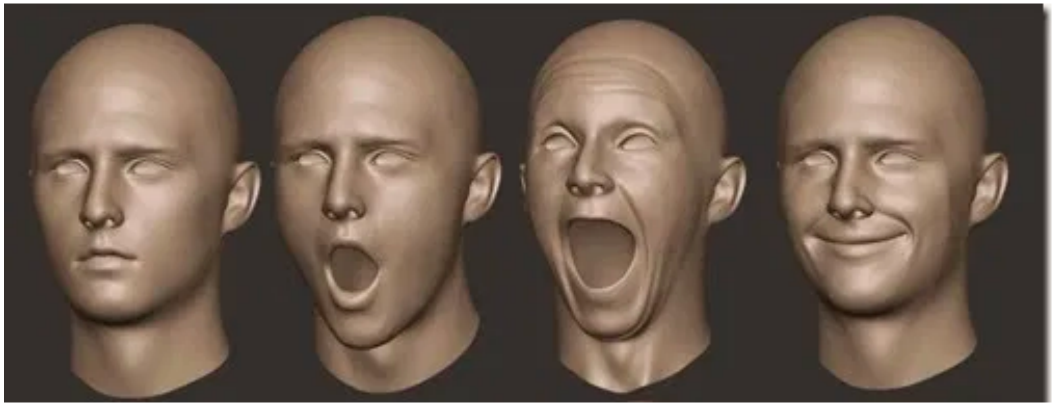


Making Comics: Storytelling Secrets Of Comics, Manga, and Graphics Novels by Scott McCloud

Blend Shapes

- A set of vertex offsets to neutral shape
- Linearly interpolate these key blend shapes for control
- Often used for expressions
- Works for deformations that are linear,
  i.e. the average of two shapes is a valid shape

$$B = \mathrm{vec}\left(\begin{bmatrix} \Delta x_1 & \Delta y_1 & \Delta z_1 \\ & \vdots & \\ & \vdots & \\ \Delta x_N & \Delta y_N & \Delta z_N \end{bmatrix}\right)$$



$$V = \sum_i \beta_i B_i$$

Blend Shapes

Courtesy Félix Ferrand

Blend Shapes

与骨骼蒙皮的关系

	控制对象	优点	常用场景
骨骼蒙皮 *(Skeletal Skinning)*	顶点由骨骼带动	整体动作	躯干、手臂
Blend Shape	顶点直接插值	精细局部表情	面部、肌肉抖动

Rigging

绑定：是连接建模和动画的桥梁

给静态模型装上“骨骼、控制器和驱动逻辑”，让它能动起来

Augment character with controls to easily change its pose, create facial expressions, bulge muscles, etc.

Rigging is like the strings on a marionette.

Capture space of meaningful deformations.

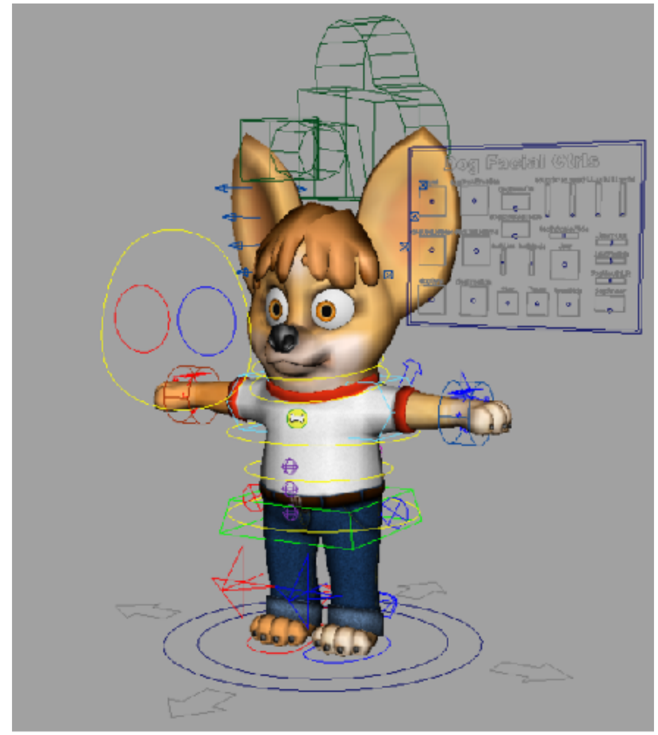
Varies from character to character.

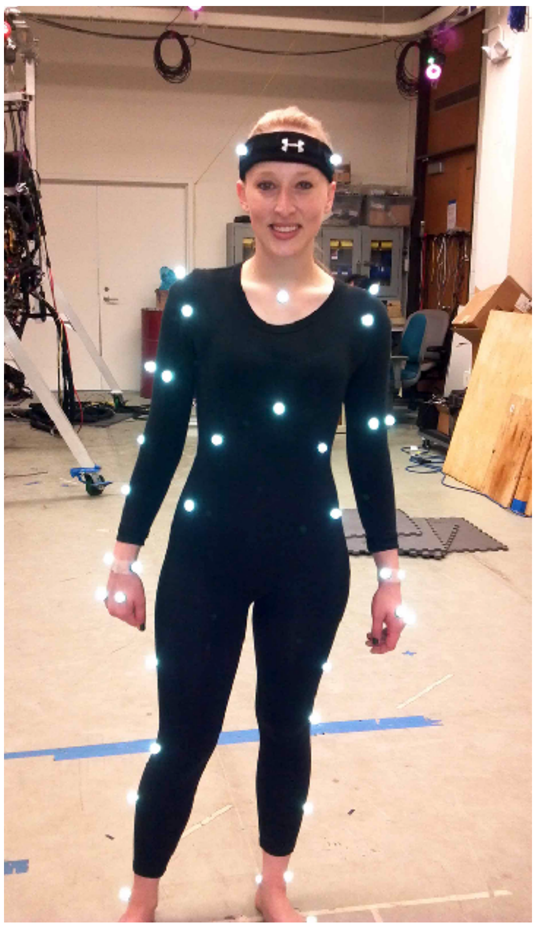
Skeleton is ONE type of rigging

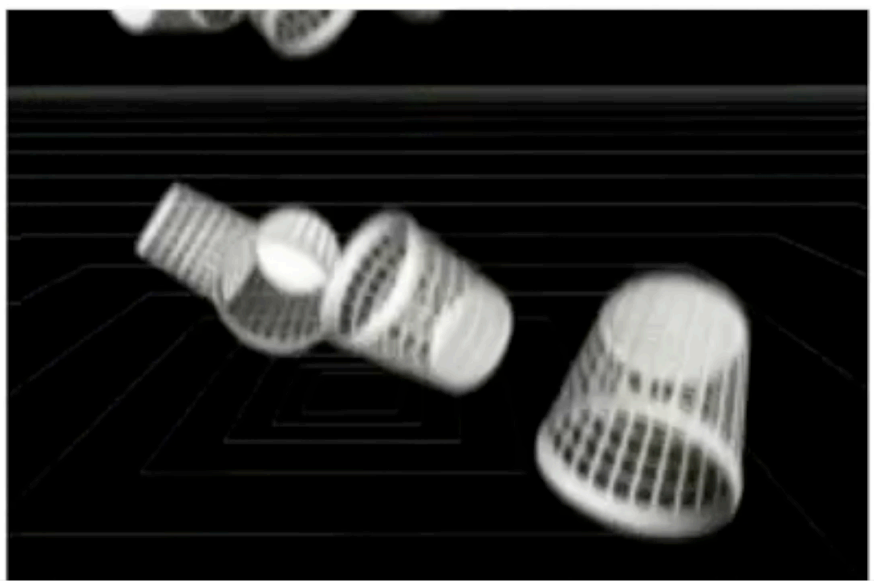
Example of A Diverse Set of Sophisticated Rigs

Basic Techniques in Computer Animation

- **Artist-directed (e.g., keyframing)**
- **Data-driven (e.g., motion capture)**
- **Procedural (e.g., simulation)**







Motion Capture

Data-driven approach to creating animation sequences

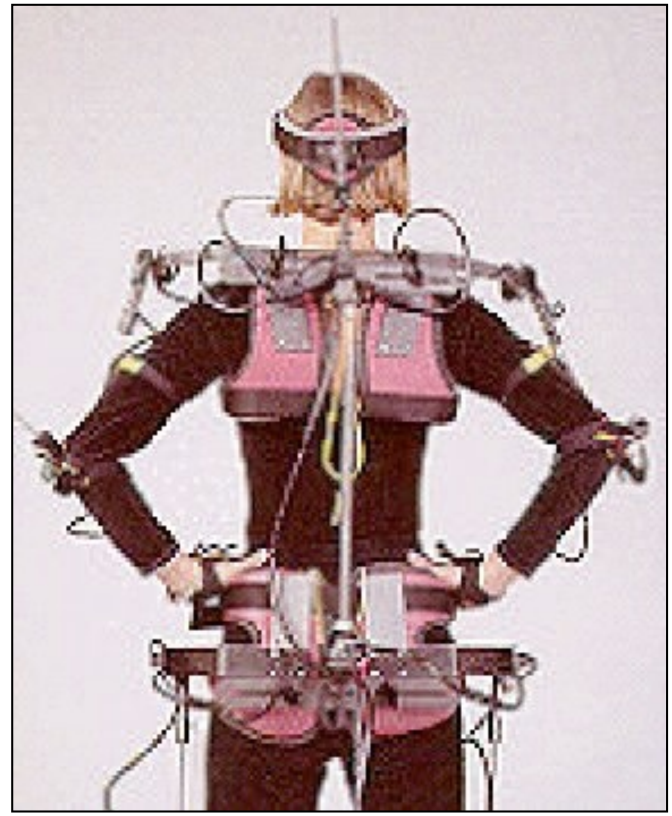
- **Record real-world performances (e.g., person executing an activity)**
- **Extract pose as a function of time from the data collected**

Motion capture room for ShaqFu

Motion Capture Equipment

Optical
(More on following slides)

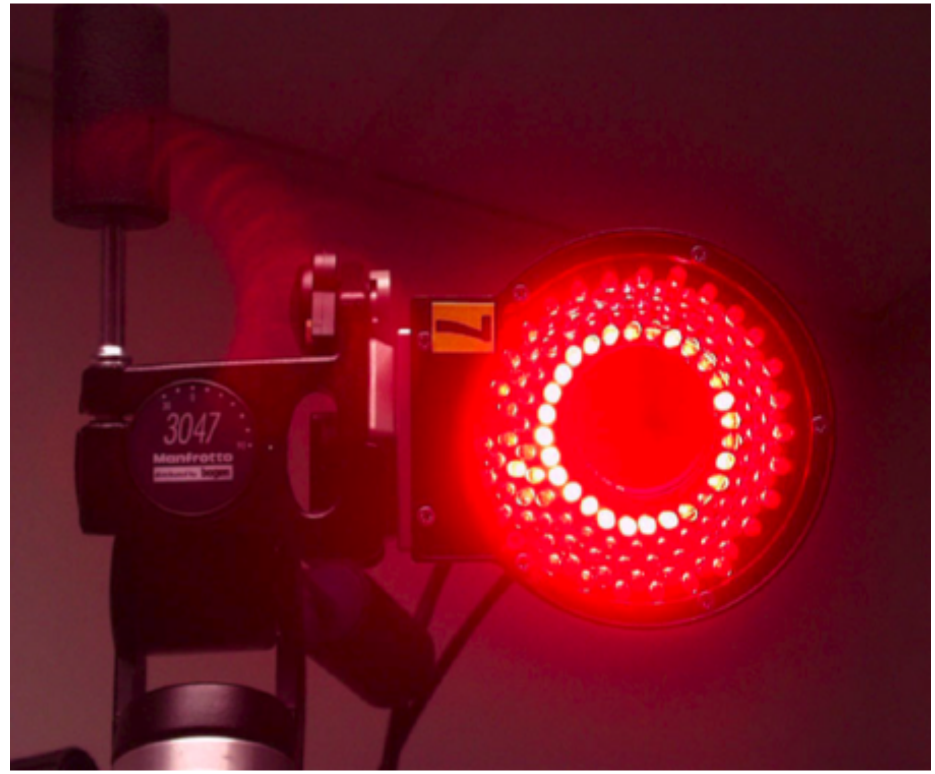
Magnetic
Sense magnetic fields to infer position / orientation. Tethered.



Mechanical
Measure joint angles directly. Restricts motion.

Optical Motion Capture

Retroflective markers attached to subject



IR illumination and cameras

- Markers on subject
- Positions by triangulation from multiple cameras
- 8+ cameras, 240 Hz, occlusions are difficult

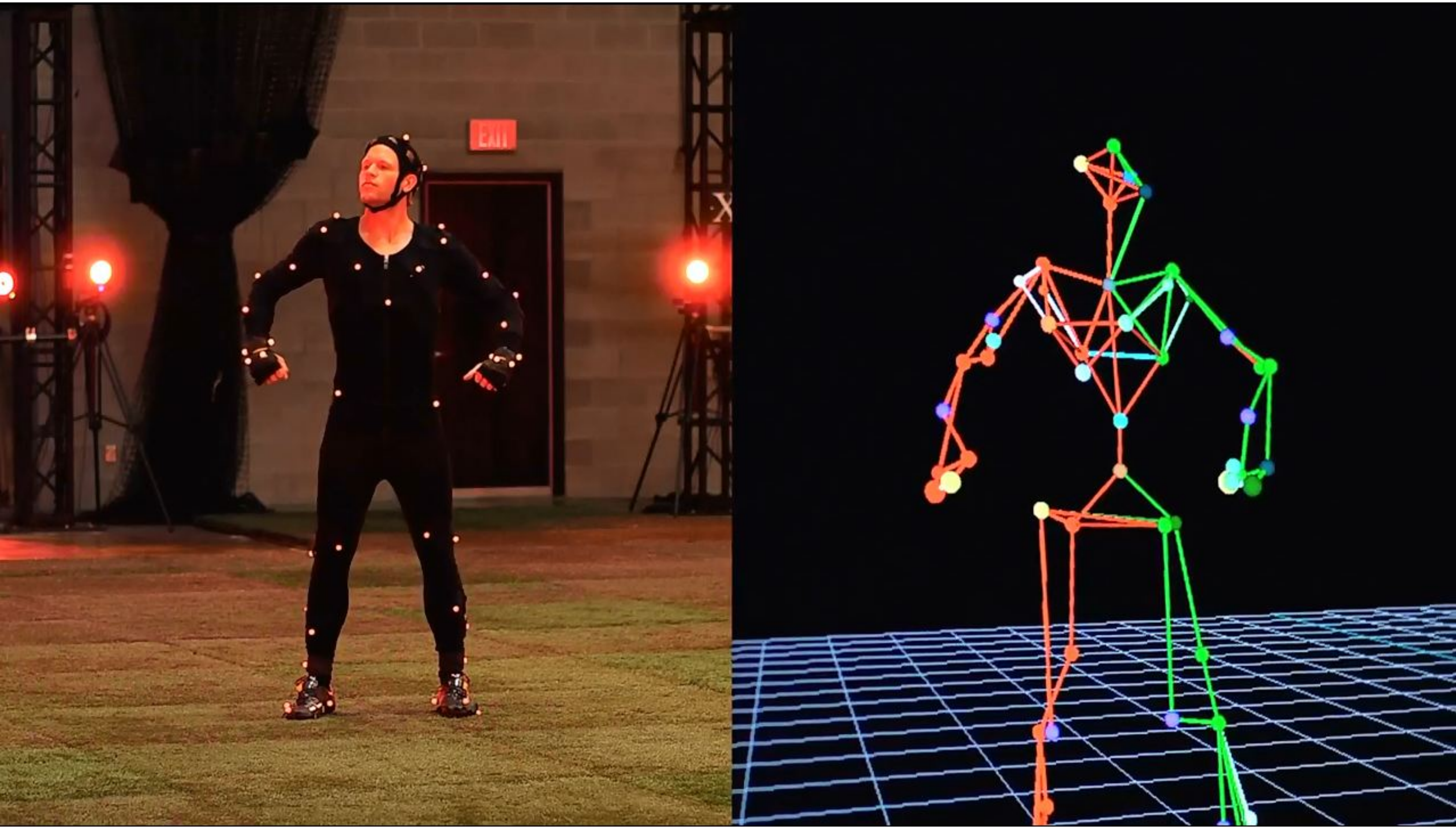
Slide credit: Prof. Steve Marschner @ Cornell

Optical Motion Capture

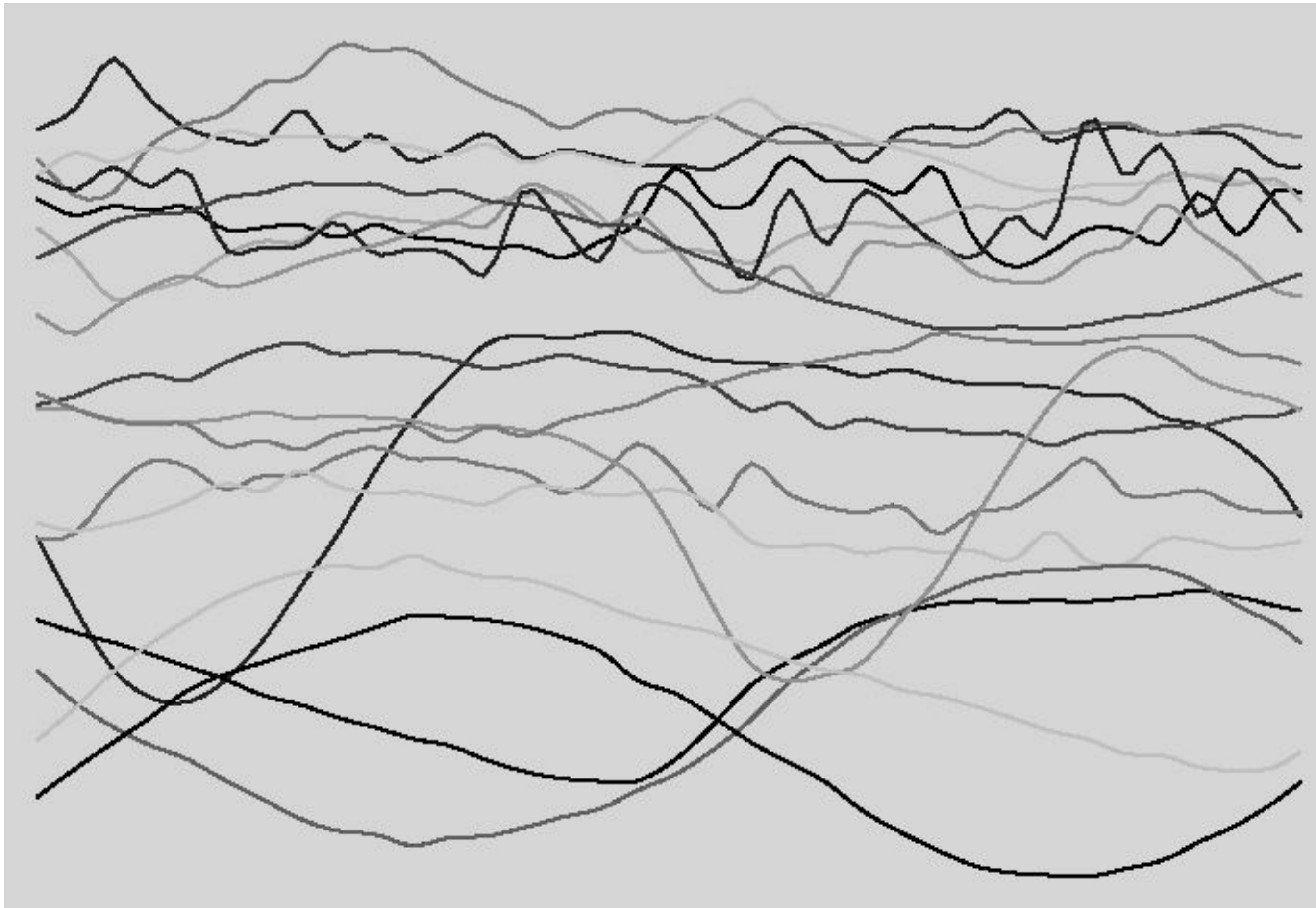
Source: http://fightland.vice.com/blog/ronda-rousey-20-the-queen-of-all-media

Ronda Roussey in Electronic Arts' motion capture studio

Motion Capture



Motion Data



Subset of motion curves from captured walking motion.

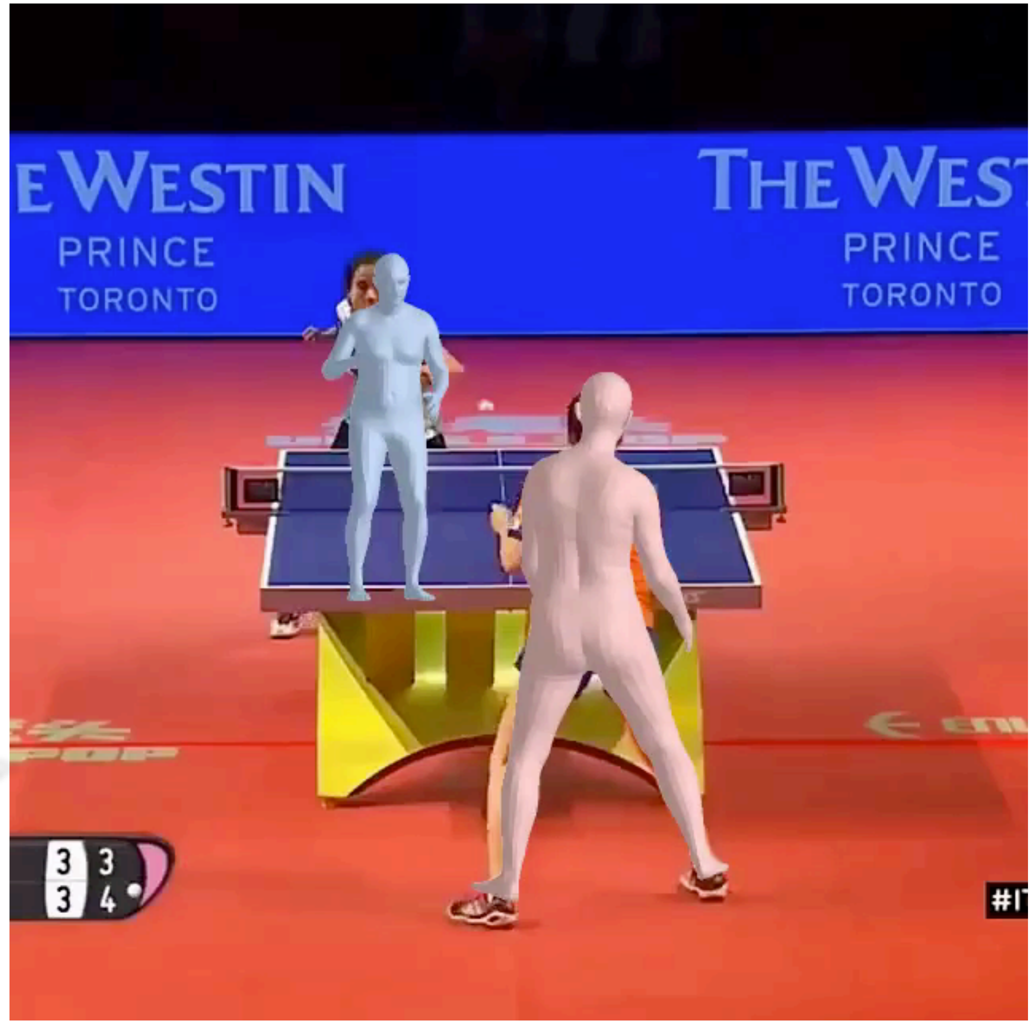
From Witkin and Popovic, 1995

Facial Motion Capture

Facial Motion Capture

Markerless Motion Capture

Kanazawa et al. 2018



Kanazawa et al. 2019

Motion Capture Pros and Cons

Strengths

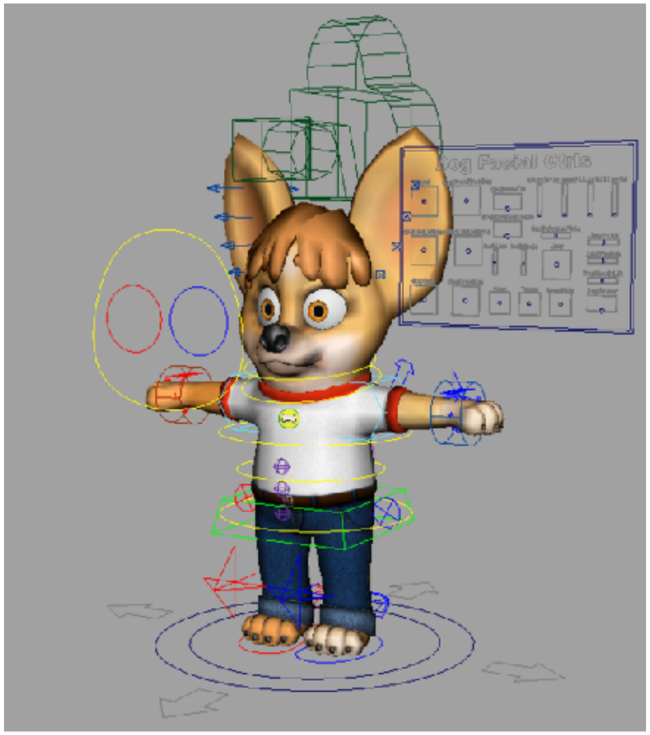
- Can capture large amounts of real data quickly
- Realism can be high

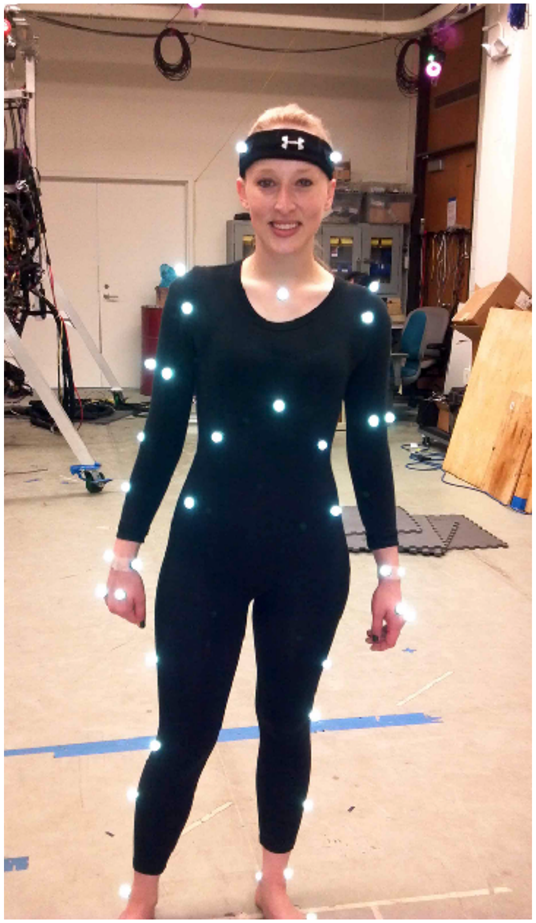
Weaknesses

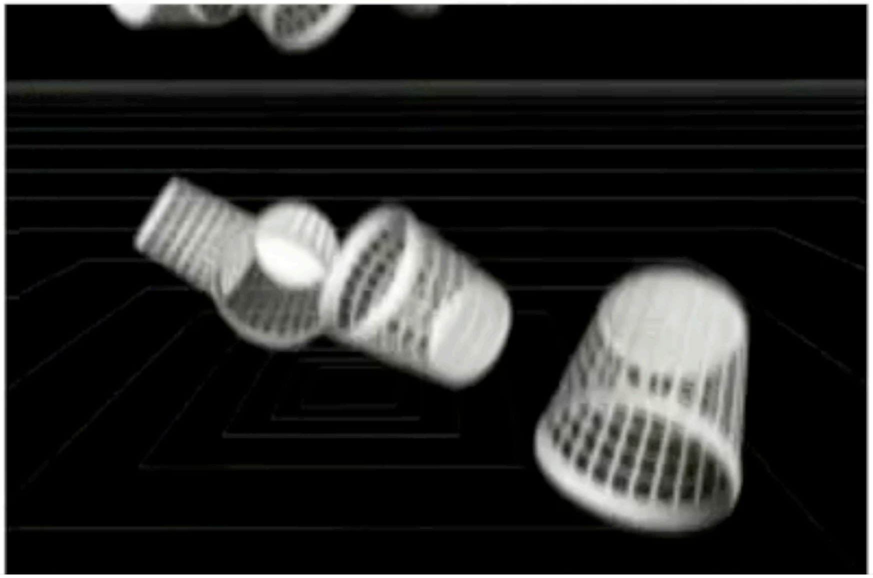
- Complex and costly set-ups
- Captured animation may not meed artistic needs, requiring alterations

Basic Techniques in Computer Animation

- **Artist-directed (e.g., keyframing)**
- **Data-driven (e.g., motion capture)**
- **Procedural (e.g., simulation)**







Dynamical Description of Motion

"A change in motion is proportional to the motive force impressed and takes place along the straight line in which that force is impressed."

— Sir Isaac Newton, 1687

"Dynamics is concerned with the study of forces and their effect on motion, as opposed to kinematics, which studies the motion of objects without reference to its causes."

— Sir Wiki Pedia, 2015

(Q: Is keyframe interpolation dynamic, or kinematic?)

The Animation Equation

- Already saw the rendering equation
- What's the animation equation?

$$\underbrace{F}_{\text{force}} = \overbrace{m}^{\text{mass}}\,\overbrace{a}^{\text{acceleration}}$$

Dot Notation for Derivatives

If $\boldsymbol{x}$ is a vector for the position of a point of interest, we will use dot notation for velocity and acceleration:

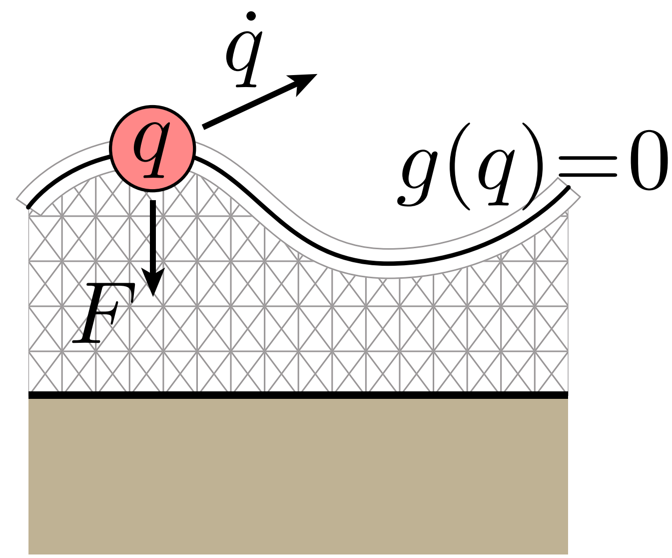
$$\boldsymbol{x}$$

$$\dot{\boldsymbol{x}} = \boldsymbol{v}$$

$$\ddot{\boldsymbol{x}} = \boldsymbol{a}$$

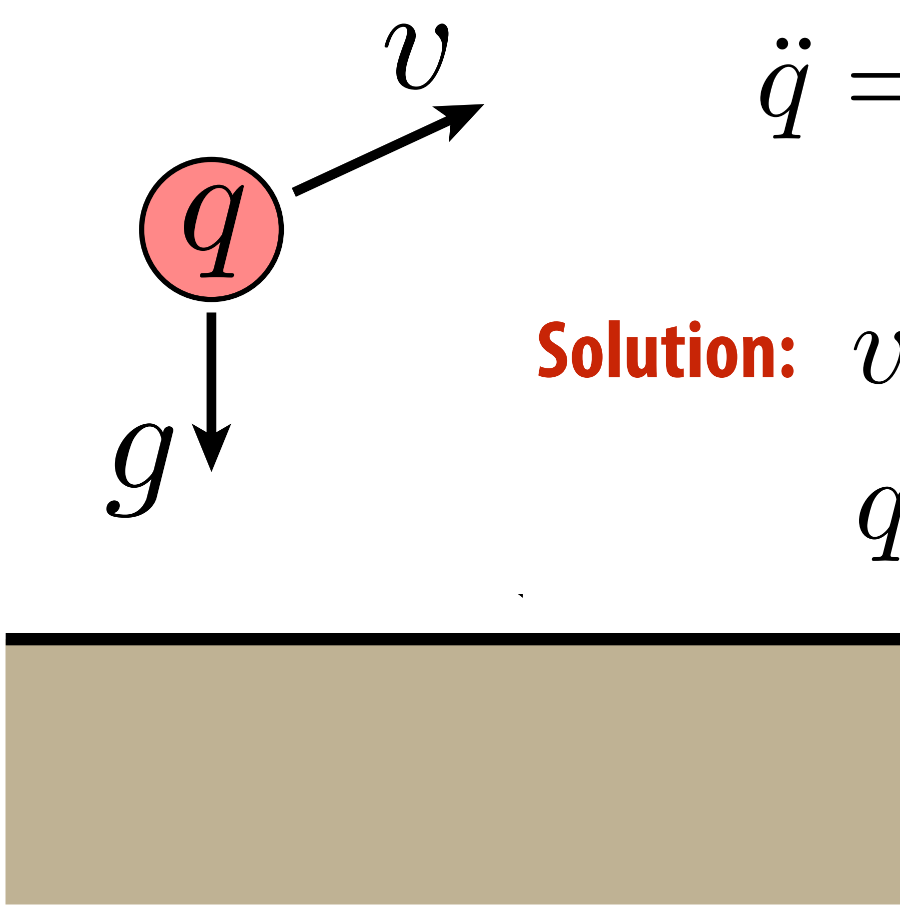
The "Animation Equation," revisited

- Well actually there are some more equations…
- Let's be more careful:
  - Any system has a configuration $q(t)$
  - It also has a velocity $\dot{q} := \frac{d}{dt} q$
  - And some kind of mass M
  - There are probably some forces F
  - And also some constraints $g(q, \dot{q}, t) = 0$
- E.g., could write Newton's 2nd law as $\ddot{q} = F/m$
- Makes two things clear:
  - Acceleration is 2nd time derivative of configuration
  - Ultimately, we want to solve for the configuration q



Simple Example: Throwing a Rock

- **Consider a rock of mass m tossed under force of gravity g**
- **Easy to write dynamical equations, since only force is gravity:**



$$\ddot{q} = g/m \quad \textbf{or} \quad \begin{aligned} \dot{q} &= v \\ \dot{v} &= g/m \end{aligned}$$

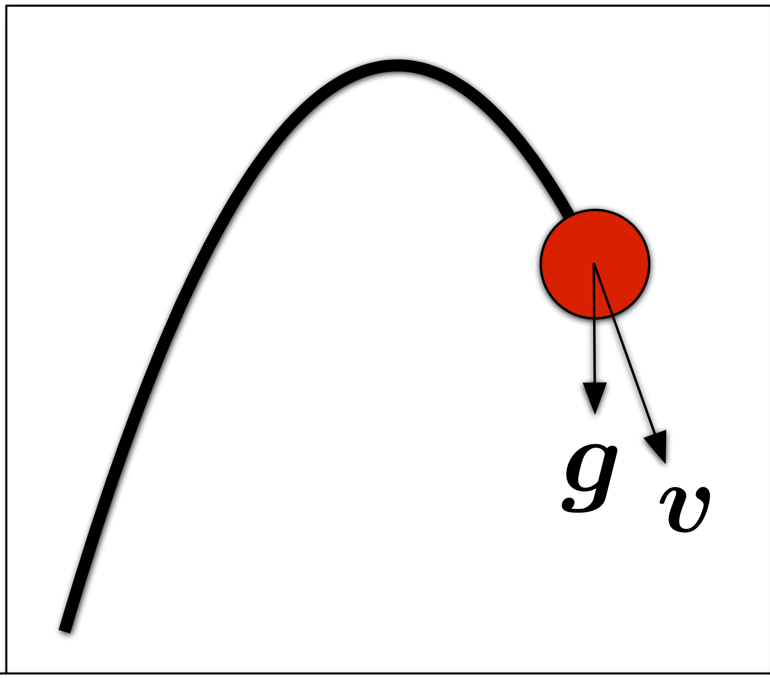
Solution:

$$v(t) = v_0 + \frac{t}{m} g$$
$$q(t) = q_0 + t v_0 + \frac{t^2}{2m} g$$

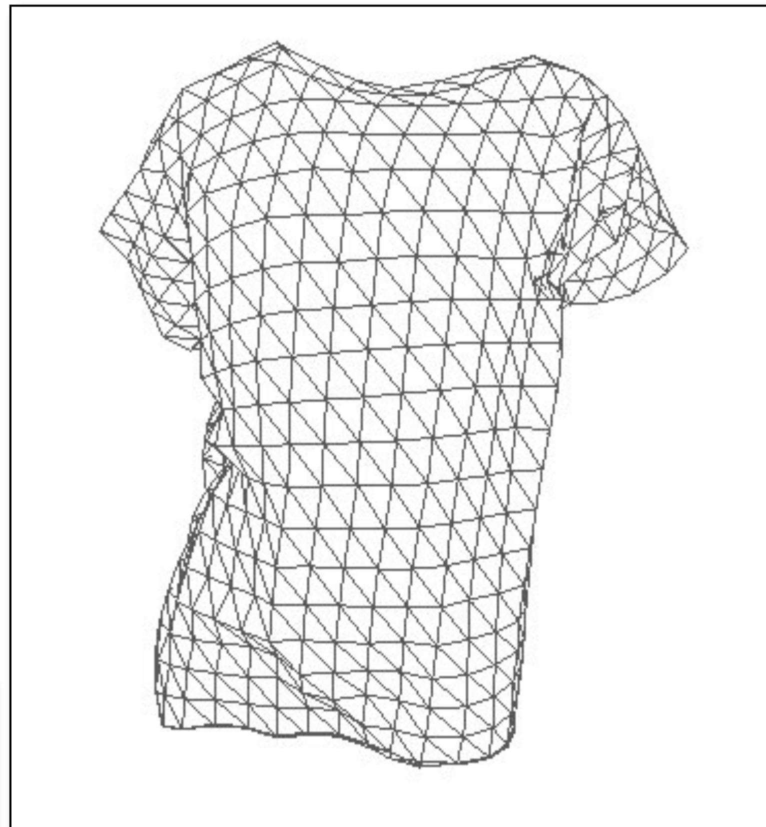
(What do we need a computer for?!)

Physically Based Animation

Generate motion of objects using numerical simulation



$$\boldsymbol{x}^{t+\Delta t} = \boldsymbol{x}^t + \Delta t \boldsymbol{v}^t + \frac{1}{2}(\Delta t)^2 \boldsymbol{a}^t$$

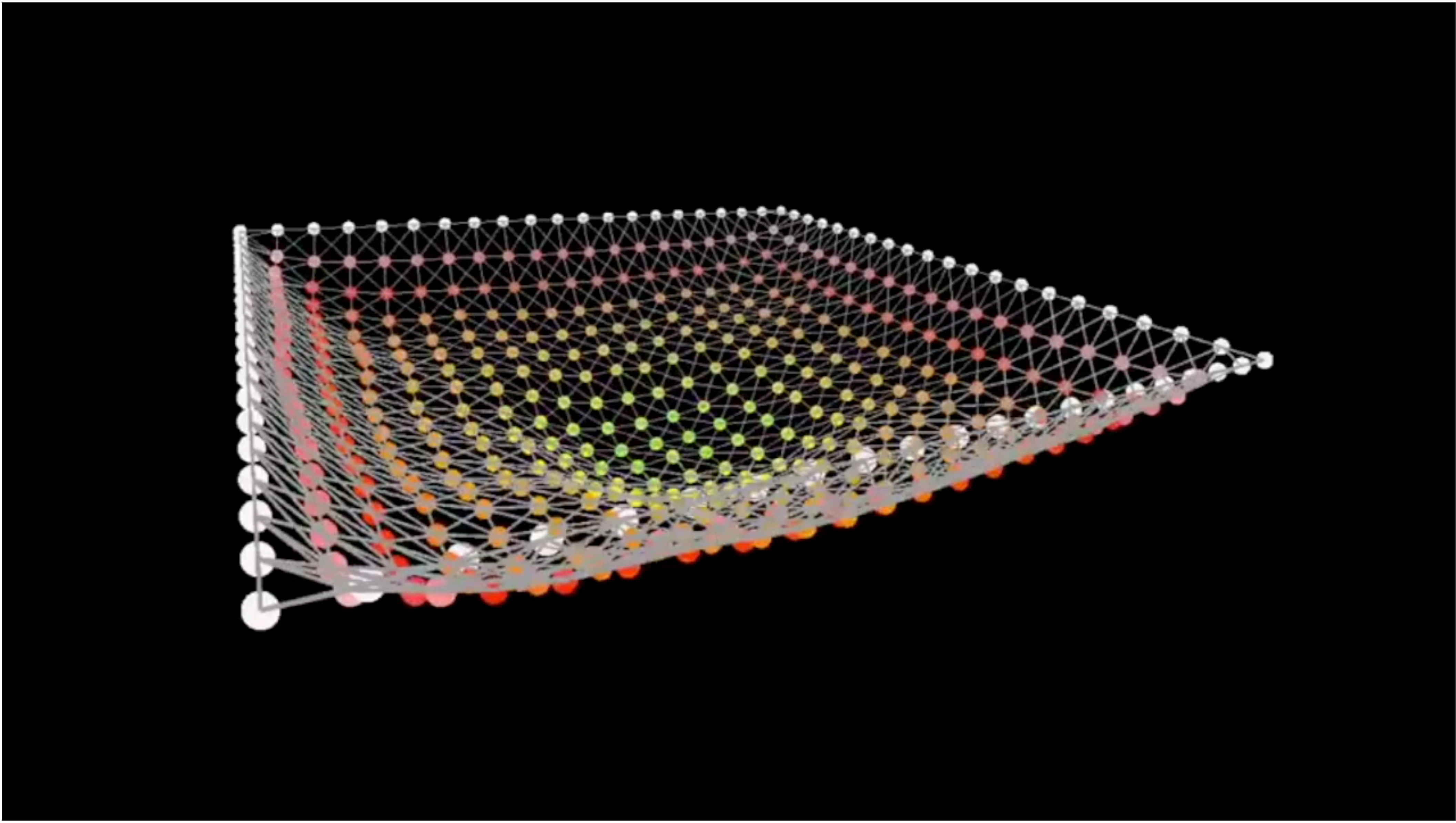


Mass Spring System:

Example of Modeling a Dynamic System

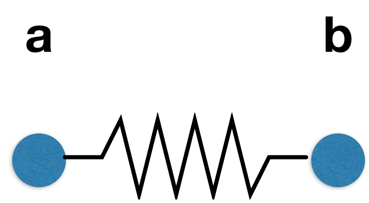
弹簧质点模型

Example: Mass Spring System



A Simple Spring

Idealized spring



$$\boldsymbol{f}_{a \to b} = k_s(\boldsymbol{b} - \boldsymbol{a})$$

$$\boldsymbol{f}_{b \to a} = -\boldsymbol{f}_{a \to b}$$

Force pulls points together

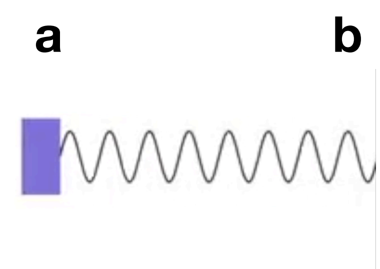
Strength proportional to displacement (Hooke's Law)

k_s is a spring coefficient: stiffness

Problem: this spring wants to have zero length

Non-Zero Length Spring

Spring with non-zero rest length



$$\boldsymbol{f}_{a\to b} = k_s \frac{\boldsymbol{b}-\boldsymbol{a}}{||\boldsymbol{b}-\boldsymbol{a}||}(||\boldsymbol{b}-\boldsymbol{a}|| - l)$$

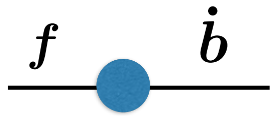
Rest length

Problem: oscillates forever

$$\boldsymbol{f}_{a\to b} = k_S \frac{\boldsymbol{b}-\boldsymbol{a}}{||\boldsymbol{b}-\boldsymbol{a}||}(||\boldsymbol{b}-\boldsymbol{a}||-l)$$

Introducing Energy Loss

Simple motion damping



$$\boldsymbol{f} = -k_d \dot{\boldsymbol{b}}$$

- Behaves like viscous drag on motion
- Slows down motion in the direction of velocity
- k_d is a damping coefficient

Problem: slows down *all* motion

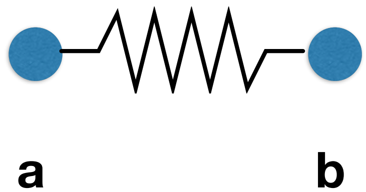
- Want a rusty spring's oscillations to slow down, but should it also fall to the ground more slowly?

Internal Damping for Spring

$$\boldsymbol{f}_{a \to b} = k_S \frac{\boldsymbol{b}-\boldsymbol{a}}{||\boldsymbol{b}-\boldsymbol{a}||}(||\boldsymbol{b}-\boldsymbol{a}|| - l)$$

$$\boldsymbol{f} = -k_d \dot{\boldsymbol{b}}$$

Damp only the internal, spring-driven motion



Relative velocity of b, assuming a is static (vector)

$$\boldsymbol{f_b} = -k_d \boxed{\frac{\boldsymbol{b}-\boldsymbol{a}}{||\boldsymbol{b}-\boldsymbol{a}||}(\dot{\boldsymbol{b}} - \dot{\boldsymbol{a}})} \cdot \frac{\boldsymbol{b}-\boldsymbol{a}}{||\boldsymbol{b}-\boldsymbol{a}||}$$

Damping force applied on b

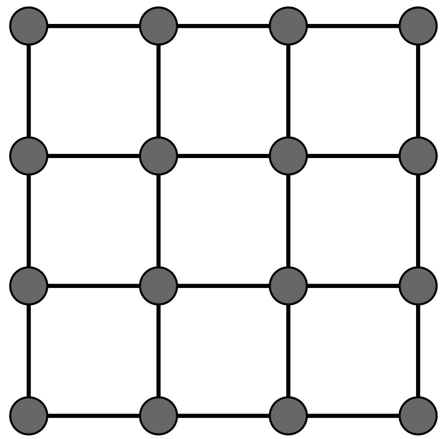
Relative velocity projected to the direction from a to b (scalar)

Direction from a to b

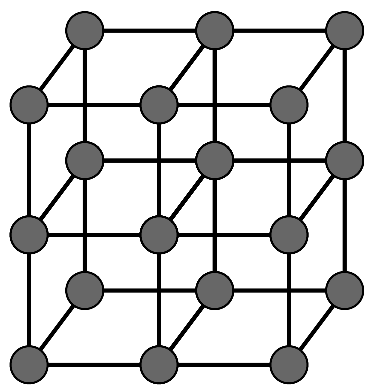
- Viscous drag only on change in spring length
  - Won't slow group motion for the spring system (e.g. global translation or rotation of the group)
- Note: This is only one specific type of damping

Structures from Springs

Sheets



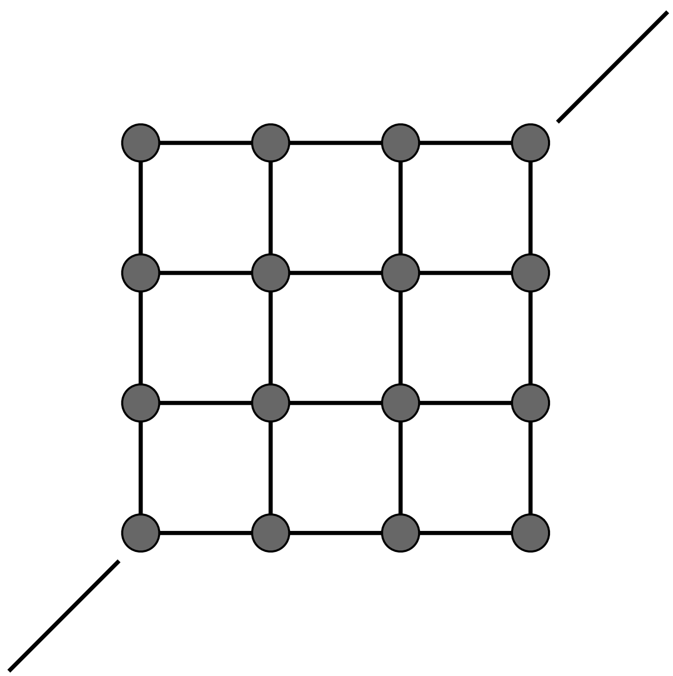
Blocks



Others

Structures from Springs

Behavior is determined by structure linkages

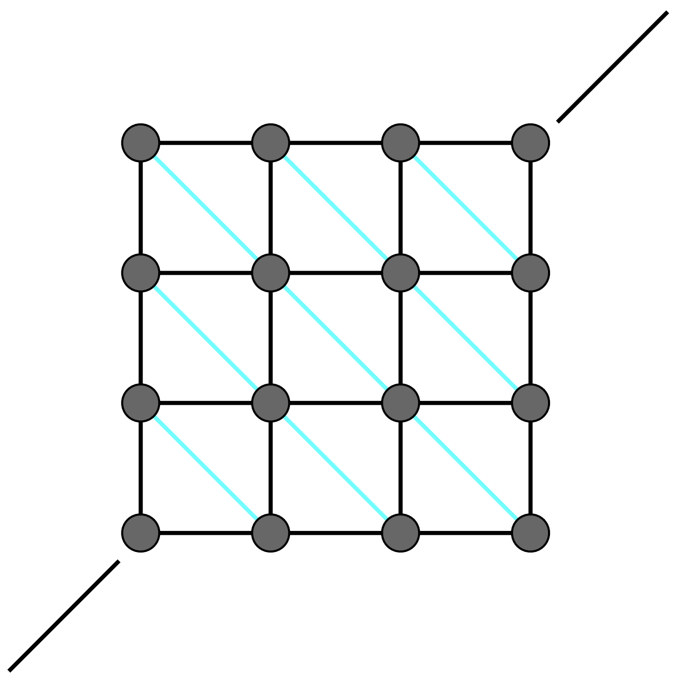


This structure will not resist shearing

This structure will not resist out-of-plane bending...

Structures from Springs

Behavior is determined by structure linkages

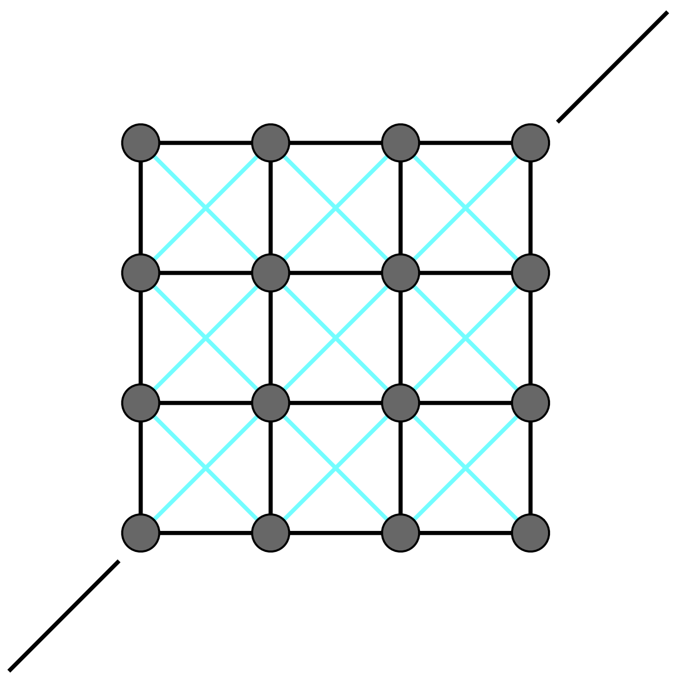


This structure will resist shearing
but has anisotropic bias

This structure will not resist out-of-plane
bending either...

Structures from Springs

Behavior is determined by structure linkages

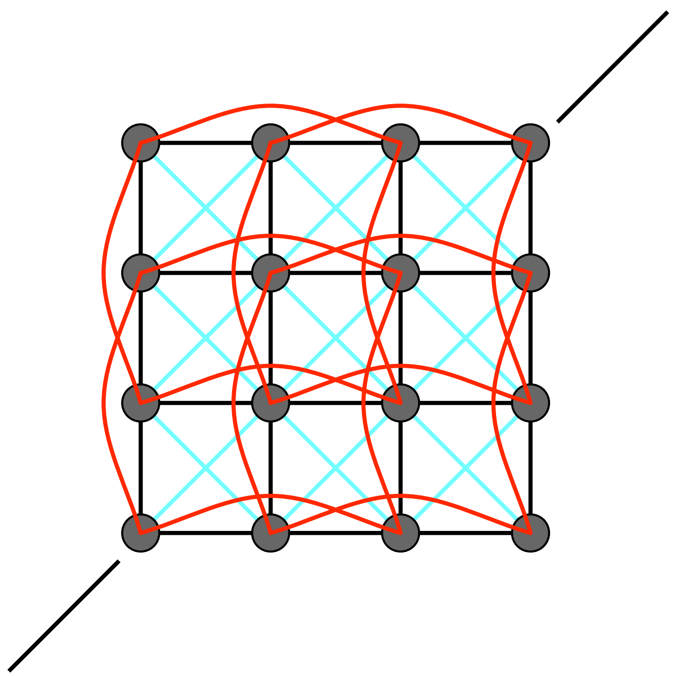


This structure will resist shearing.
Less directional bias.

This structure will not resist out-of-plane bending either...

Structures from Springs

They behave like what they are (obviously!)

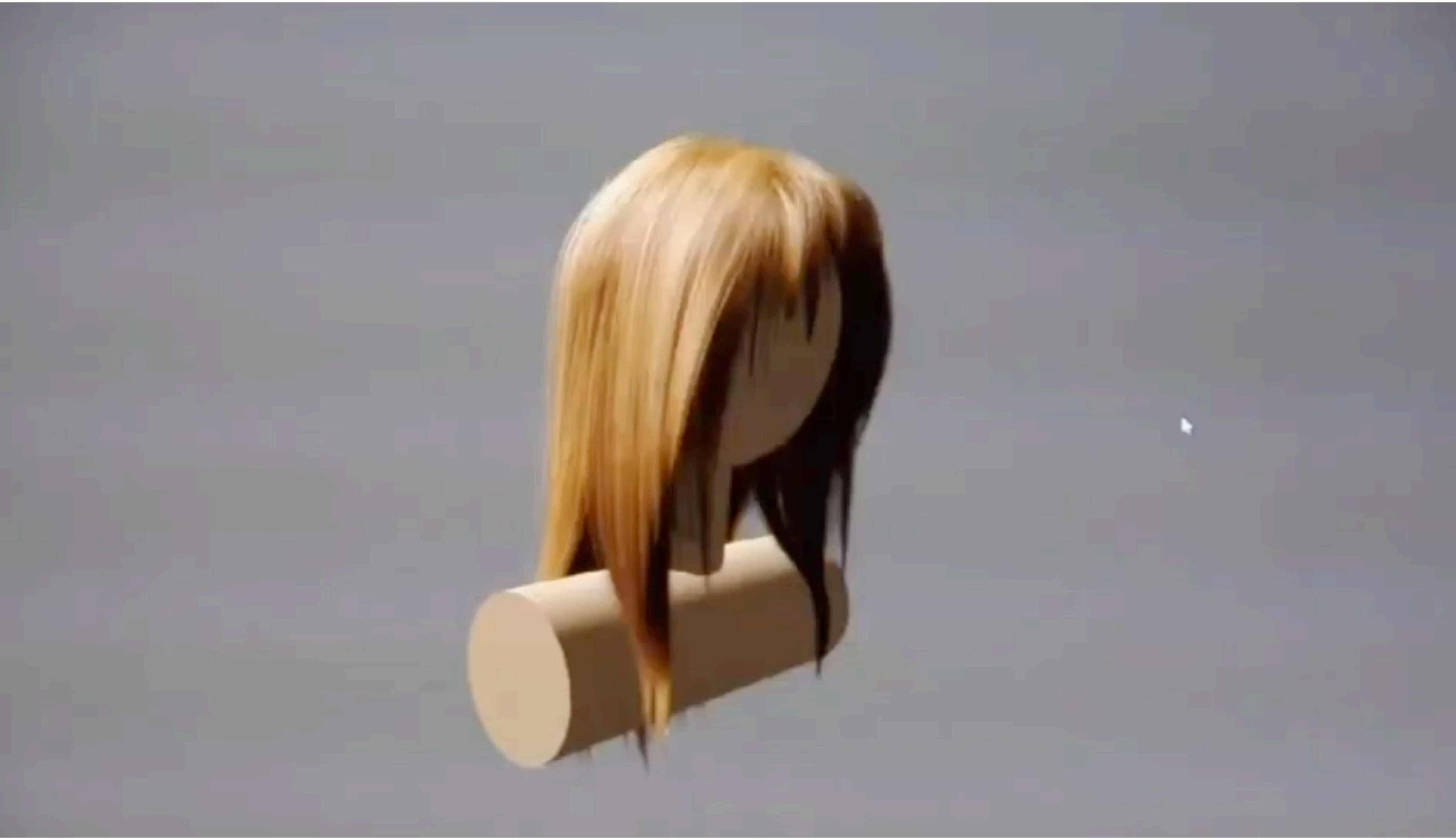


This structure will resist shearing.
Less directional bias.

This structure will resist out-of-plane bending
Red springs should be much weaker

Example: Mass Spring + Character

Example: Hair



Particle Systems

粒子系统

Example: Cosmological Simulation

Tomoaki et al - v^2GC simulation of dark matter (~1 *trillion* particles)

Particle Systems

Model dynamical systems as collections of large numbers of particles

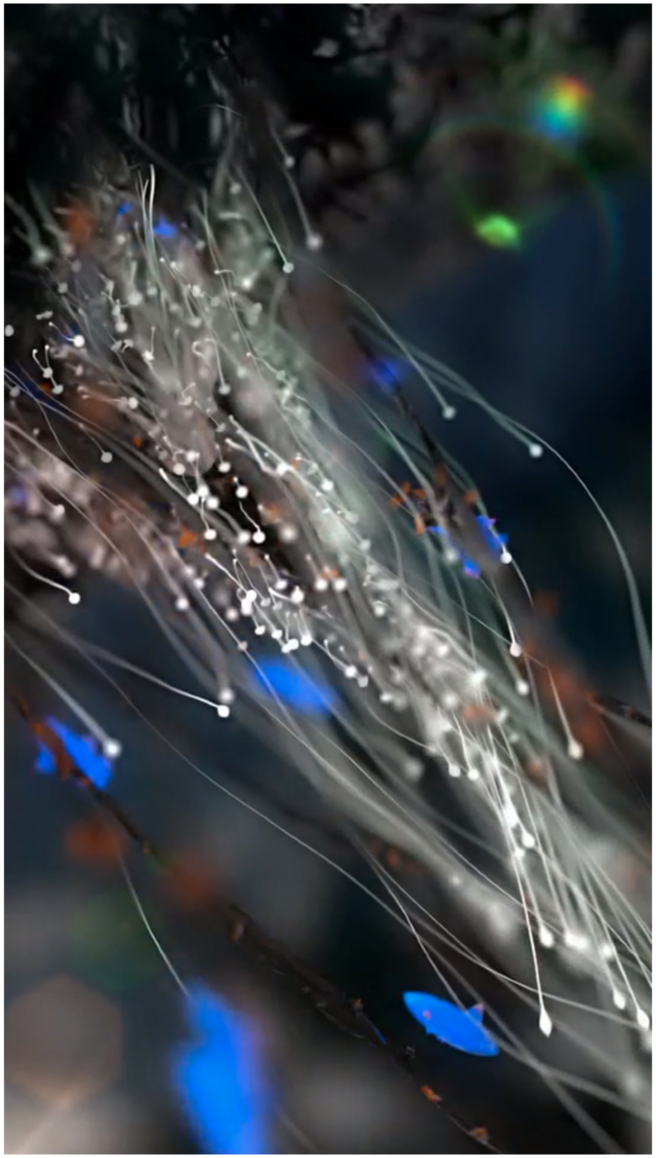
Each particle's motion is defined by a set of physical (or non-physical) forces

Popular technique in graphics and games

- Easy to understand, implement
- Scalable: fewer particles for speed, more for higher complexity

Challenges

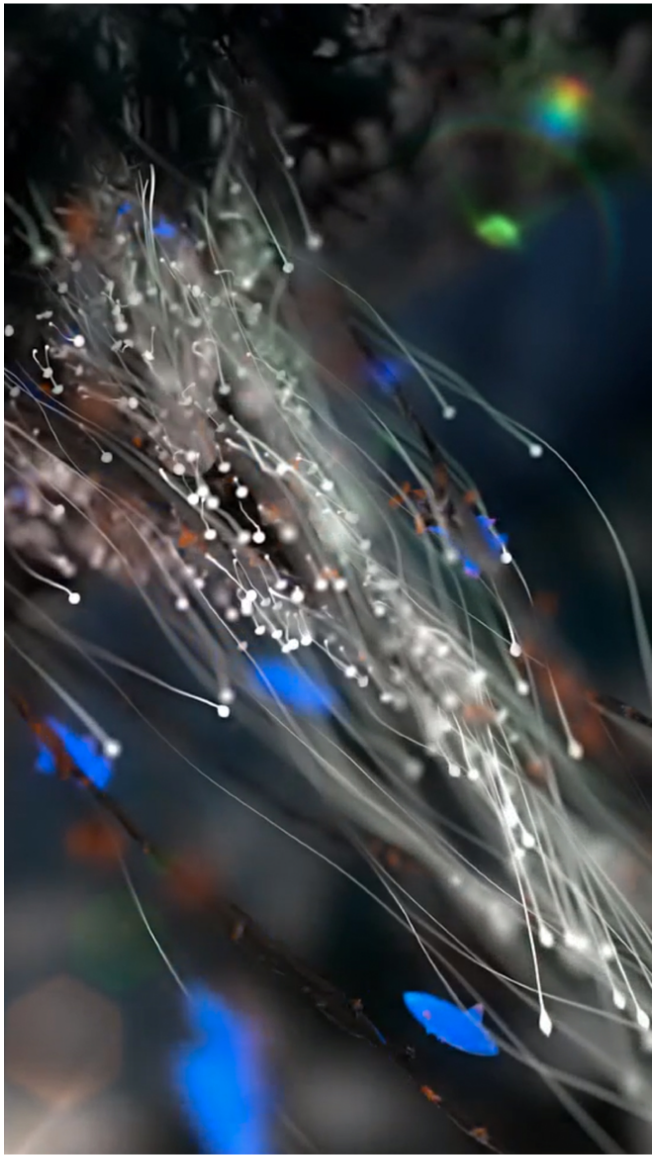
- May need *many* particles (e.g. fluids)
- May need acceleration structures (e.g. to find nearest particles for interactions)



Particle System Animations

For each frame in animation

- [If needed] Create new particles
- Calculate forces on each particle
- Update each particle's position and velocity
- [If needed] Remove dead particles
- Render particles



Particle System Forces

Attraction and repulsion forces

- Gravity, electromagnetism, …
- Springs, propulsion, …

Damping forces

- Friction, air drag, viscosity, …

Collisions

- Walls, containers, fixed objects, …
- Dynamic objects, character body parts, …

Example: Particle-Based Fluids