Introduction to Computer Graphics

GAMES101, Lingqi Yan, UC Santa Barbara

Lecture 21:
Animation

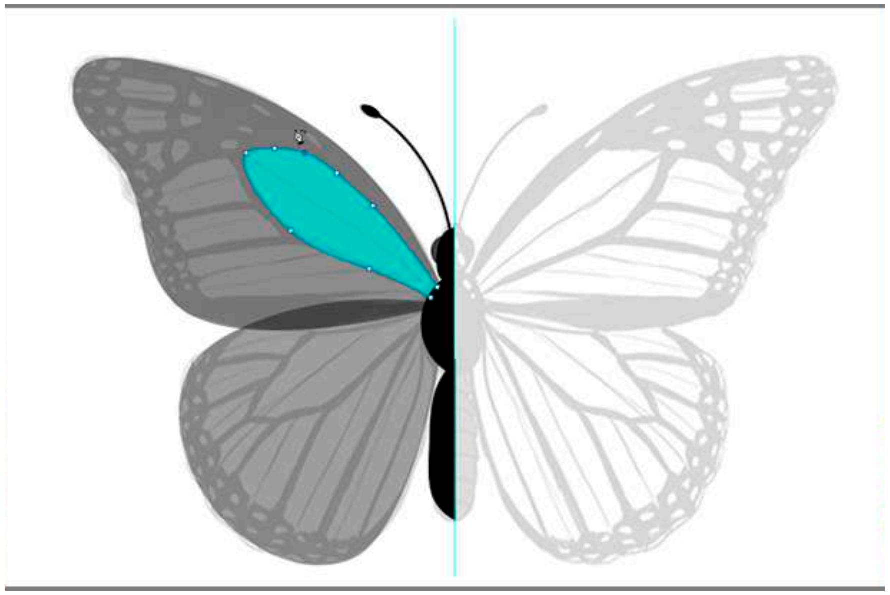
http://www.cs.ucsb.edu/~lingqi/teaching/games101.html

Announcements

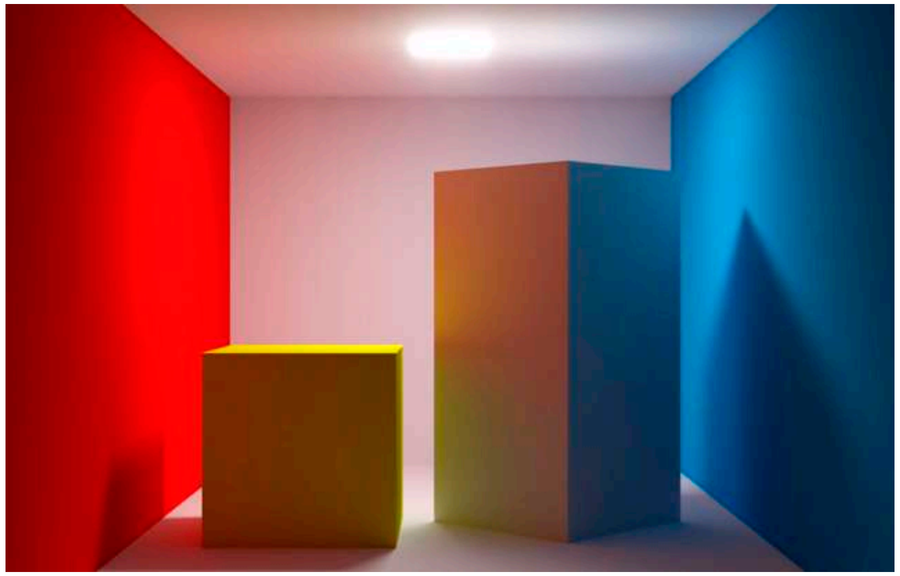
- Homework 7: 95 submissions so far
- Final project ideas: 18 submissions so far (expected more)
- My personal bad habit
  - Misuse of conjunctions
  - “OK”, “so”, etc. in English
  - “这个”, “然后”, etc. in Chinese
  - Can’t really control when I’m thinking, but will try my best to avoid them

Course Roadmap

Rasterization



Geometry



Light Transport

Animation / simulation

Today

Introduction to Computer Animation

- History
- Keyframe animation
- Physical simulation
- Kinematics
- Rigging

Animation

"Bring things to life"

- Communication tool
- Aesthetic issues often dominate technical issues

An extension of modeling

- Represent scene models as a function of time

Output: sequence of images that when viewed sequentially provide a sense of motion

- Film: 24 frames per second
- Video (in general): 30 fps
- Virtual reality: 90 fps

Historical Points in Animation

(slides courtesy of Prof. Keenan Crane @ CMU)

First Animation

(Shahr-e Sukhteh, Iran 3200 BCE)

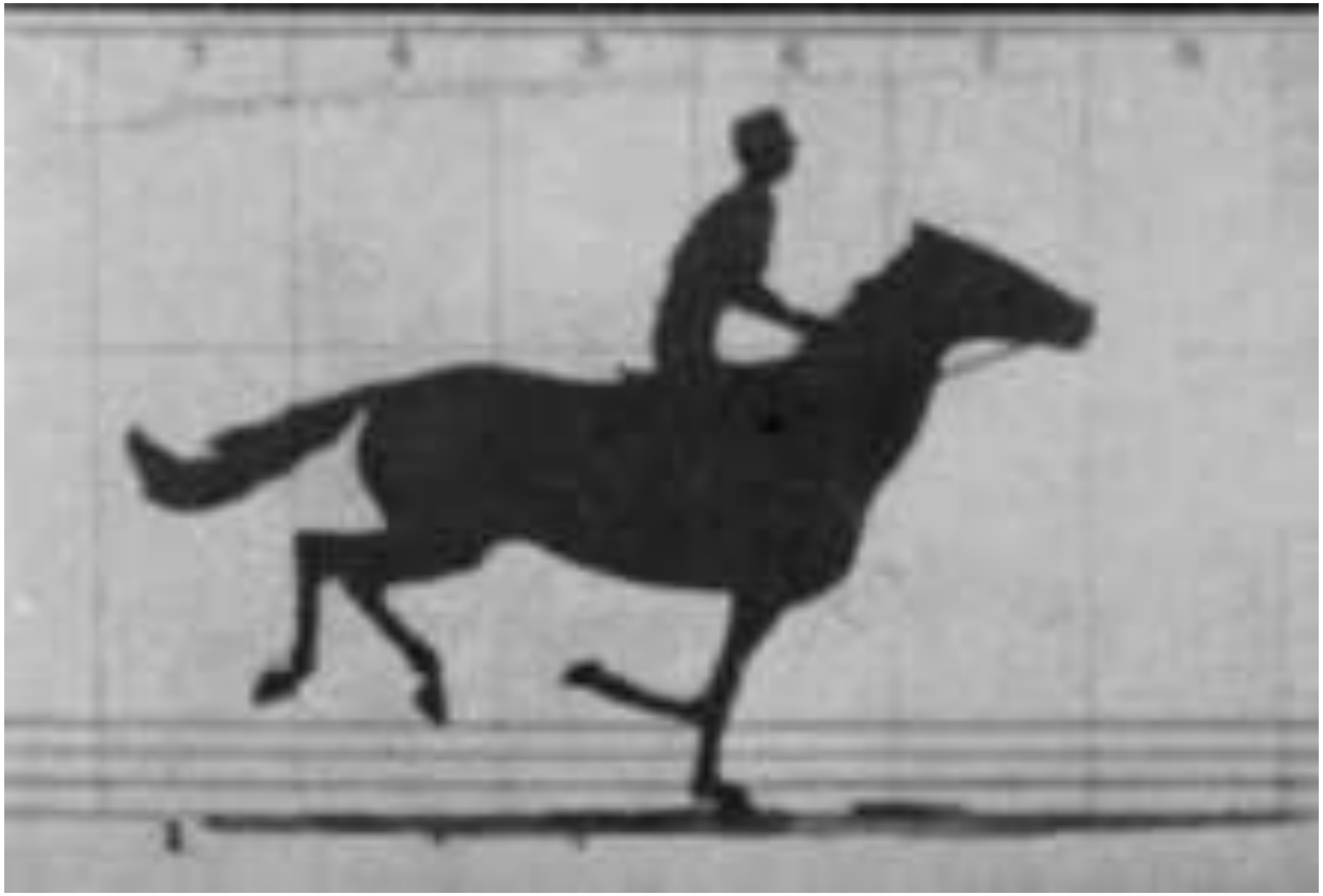
History of Animation

(Phenakistoscope, 1831)

First Film

Originally used as scientific tool rather than for entertainment

Critical technology that accelerated development of animation



Edward Muybridge, *"Sallie Gardner"* (1878)

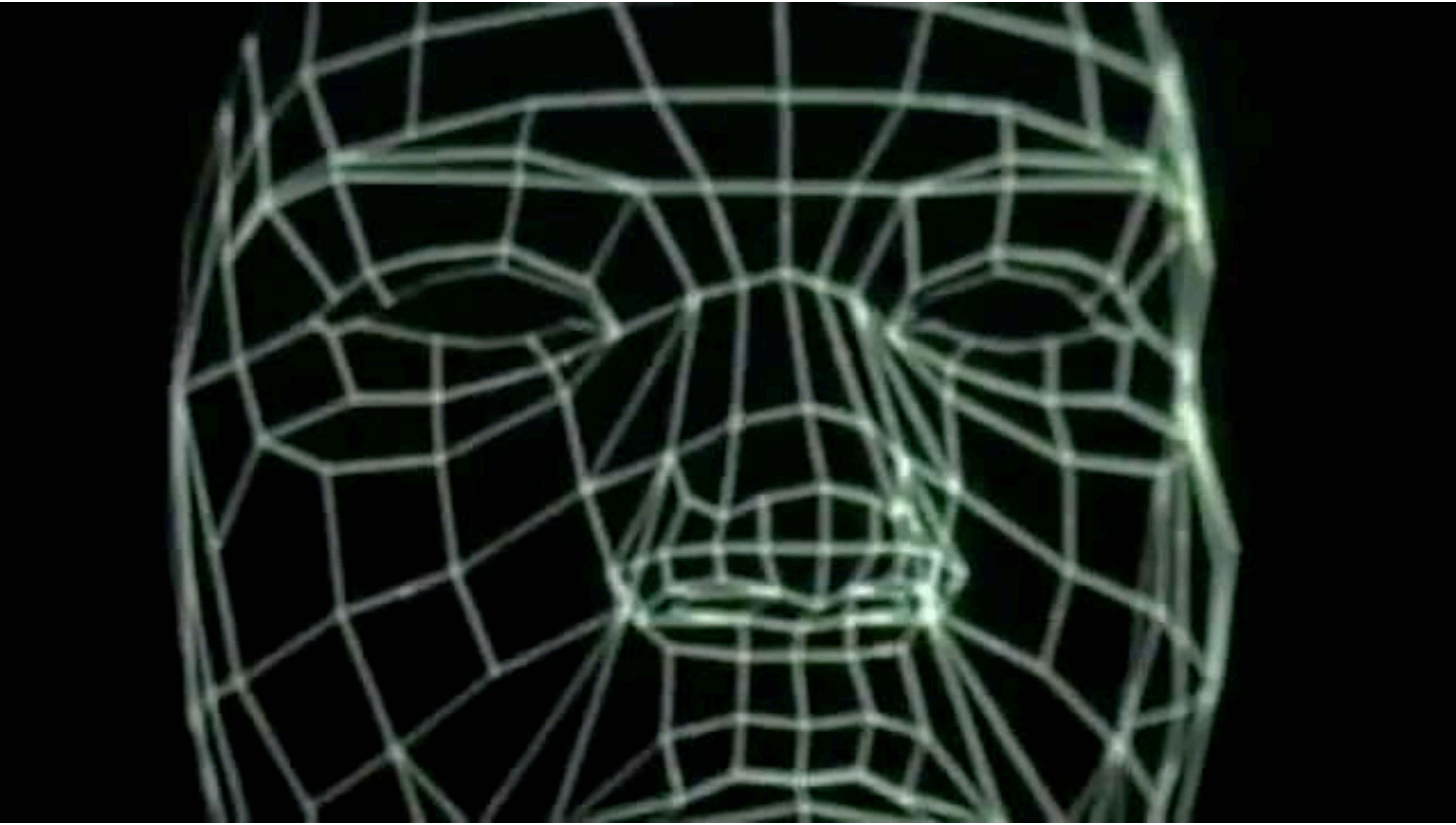
First Hand-Drawn Feature-Length (>40 mins) Animation

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

First Digital-Computer-Generated Animation

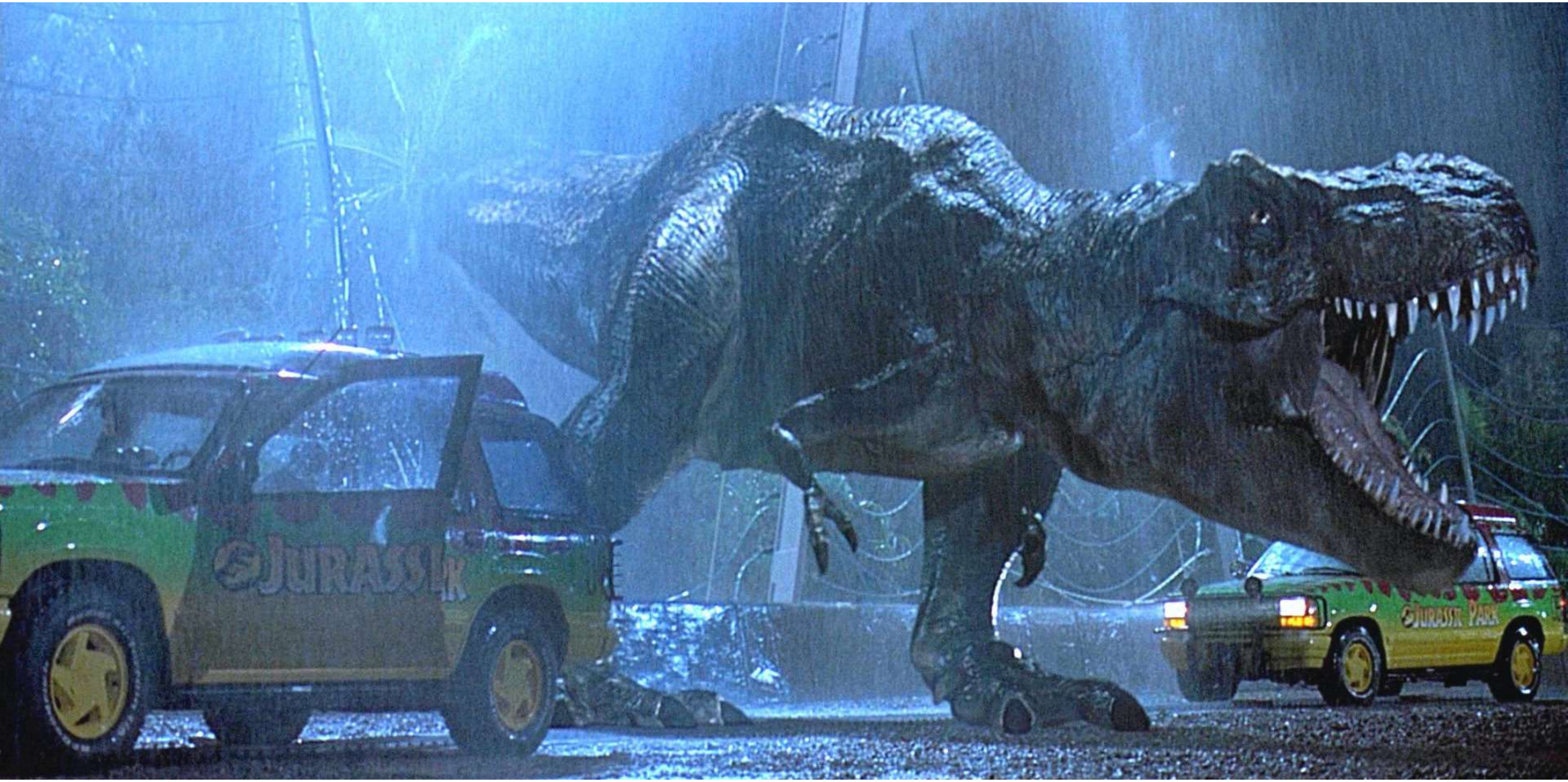
Ivan Sutherland, “Sketchpad” (1963) – Light pen, vector display

Early Computer Animation



Ed Catmull & Frederick Parke, “Computer Animated Faces” (1972)

Digital Dinosaurs!

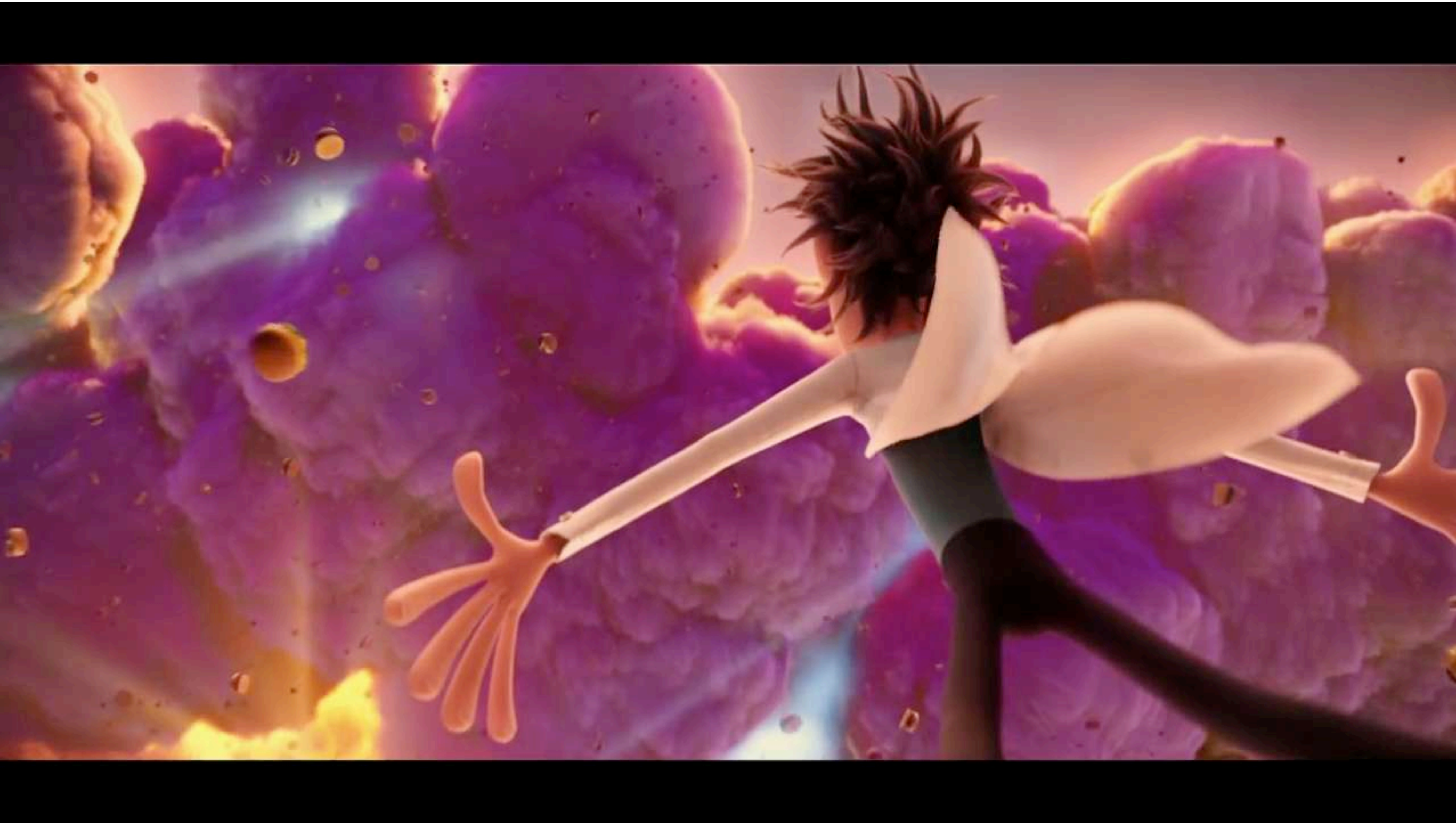


Jurassic Park (1993)

First CG Feature-Length Film

Pixar, “Toy Story” (1995)

Computer Animation - 10 years ago



Sony Pictures Animation, “Cloudy With a Chance of Meatballs” (2009)

Computer Animation - last year

Walt Disney Animation Studios, “Frozen 2” (2019)

Keyframe Animation

Keyframe Animation

Keyframes

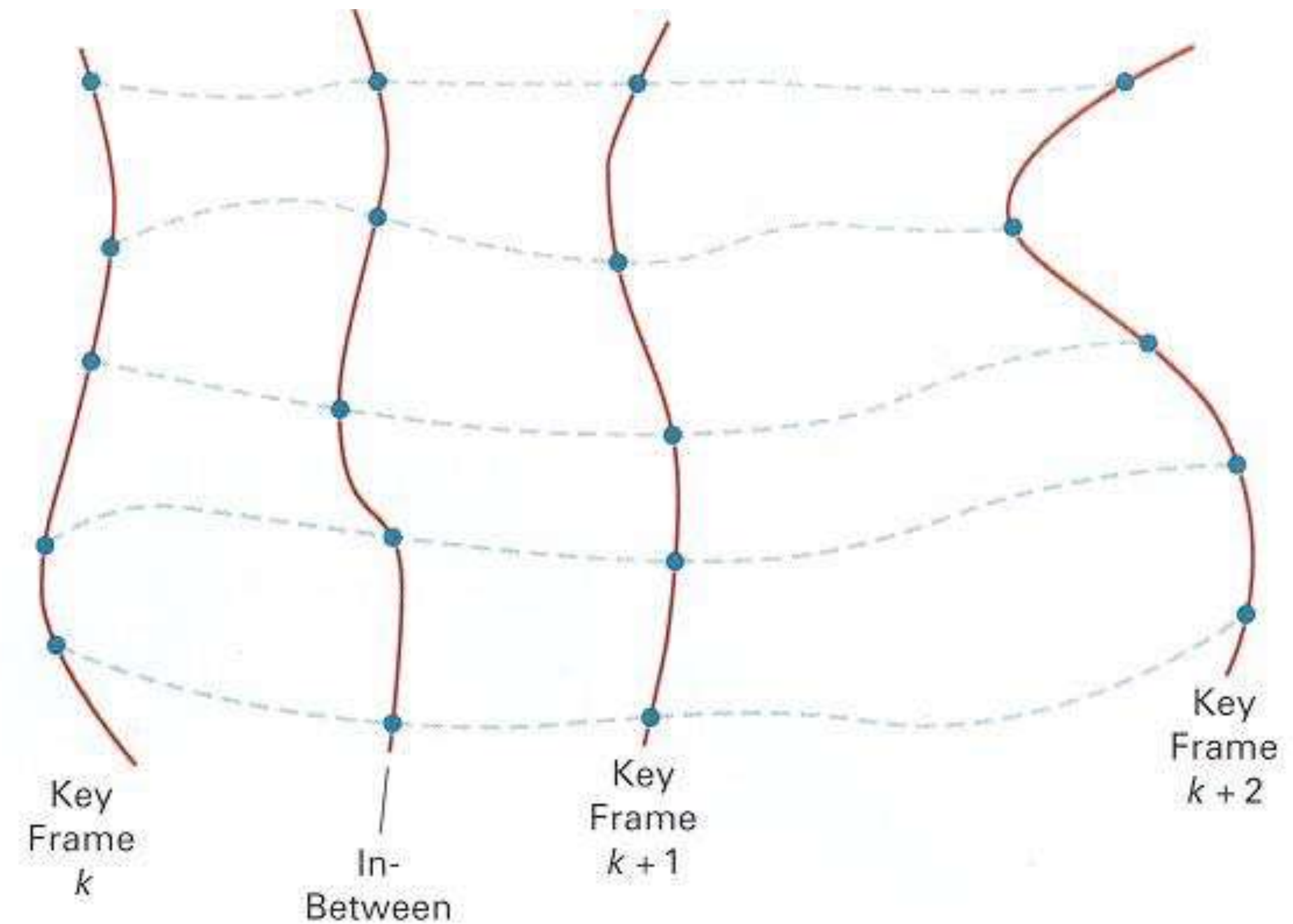
"Tweens"

Animator (e.g. lead animator) creates keyframes

Assistant (person or computer) creates in-between frames ("tweening")

Keyframe Interpolation

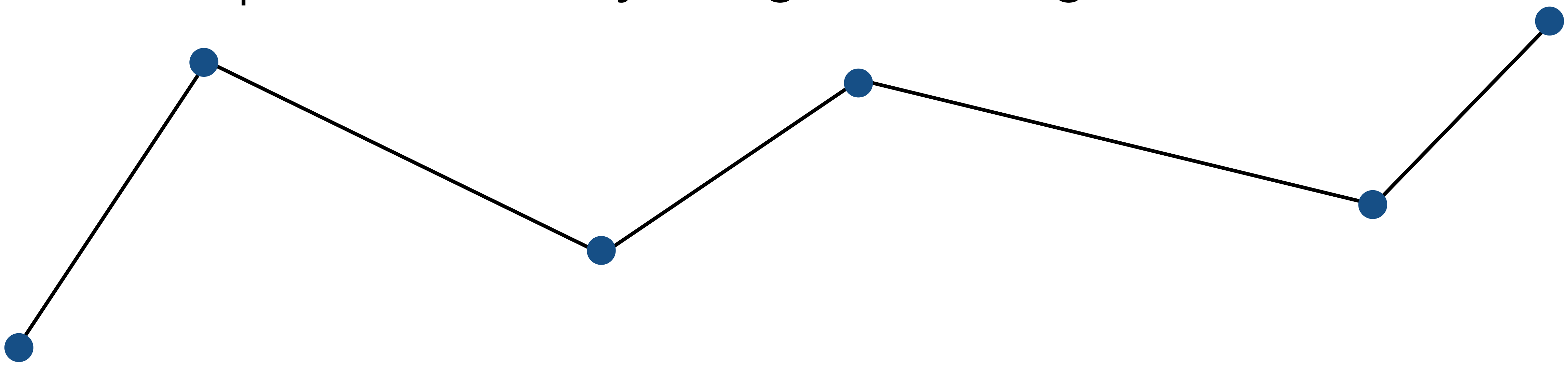
Think of each frame as a vector of parameter values



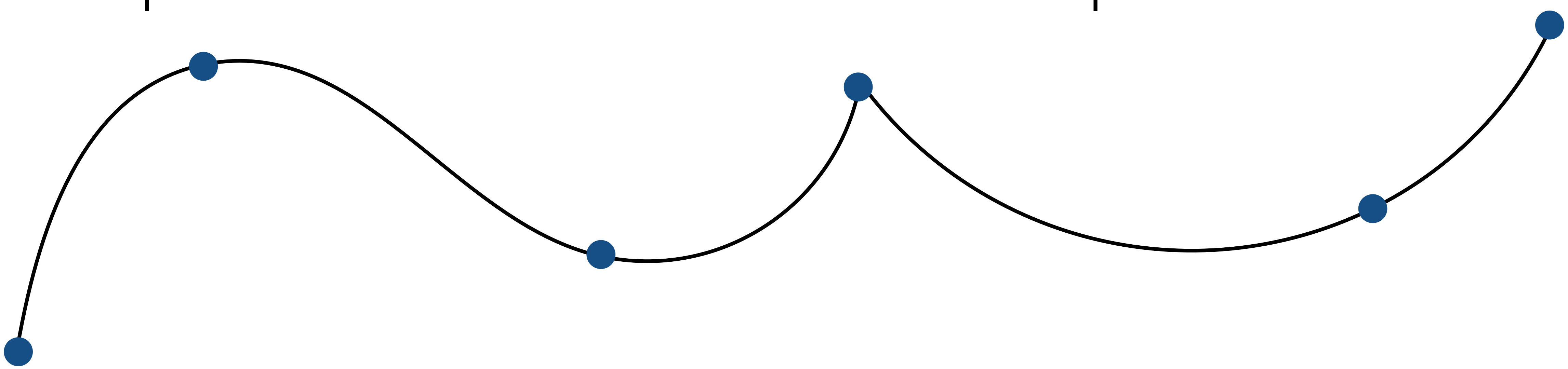
Hearn, Baker and Carithers, Figure 16.11

Keyframe Interpolation of Each Parameter

Linear interpolation usually not good enough



Recall splines for smooth / controllable interpolation



Physical Simulation

Newton's Law

$$F = ma$$

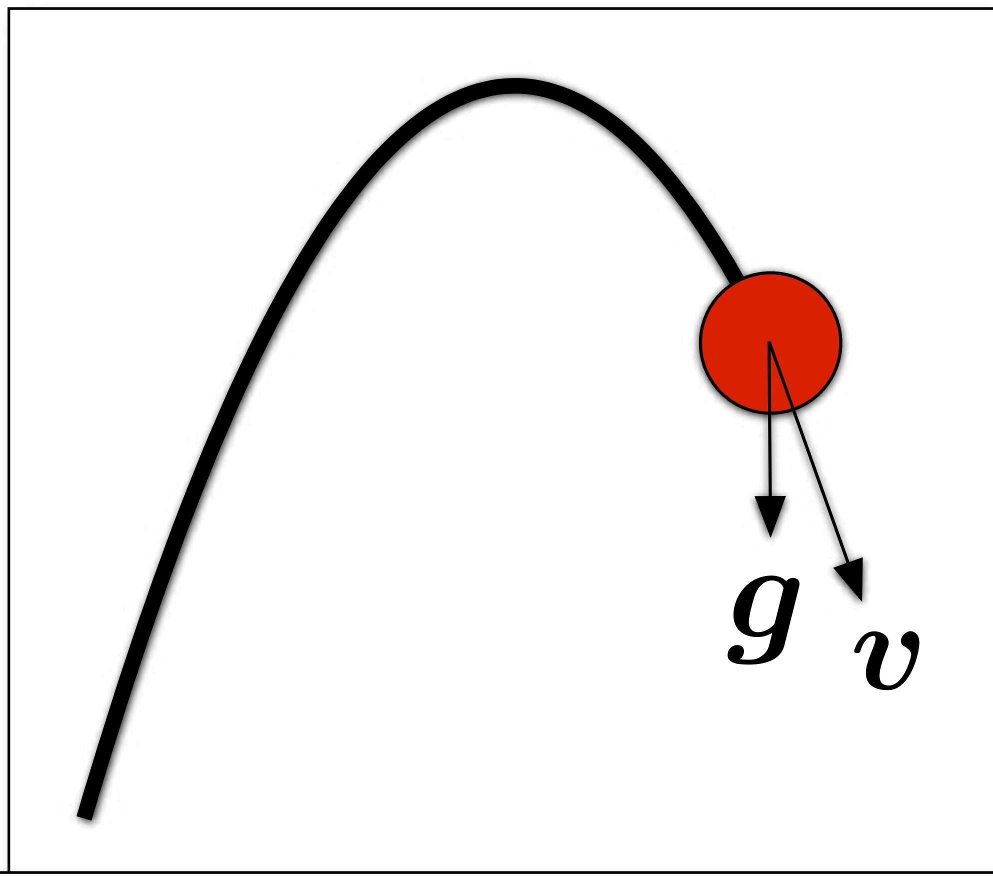
Force

Mass

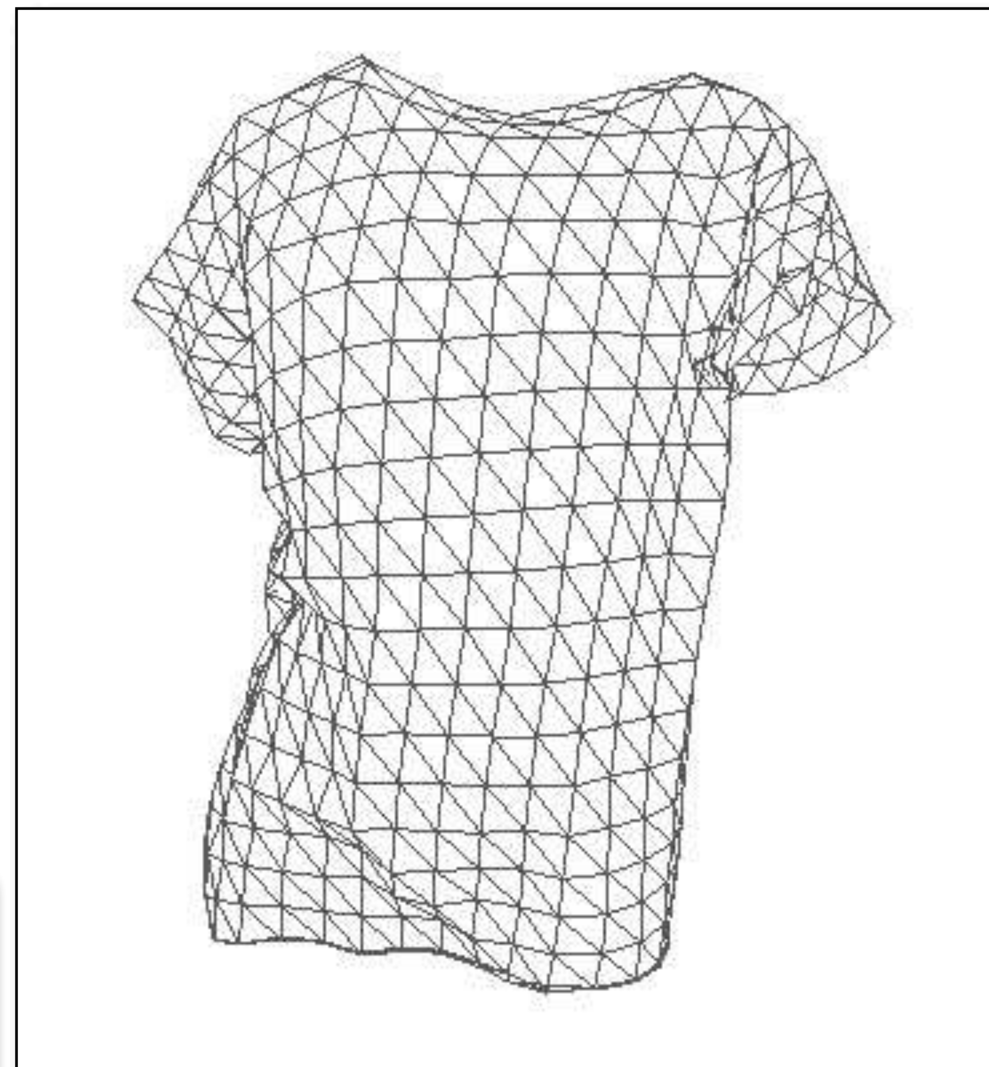
Acceleration

Physically Based Animation

Generate motion of objects using numerical simulation



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



Example: Cloth Simulation

Example: Fluids

Macklin and Müller, Position Based Fluids

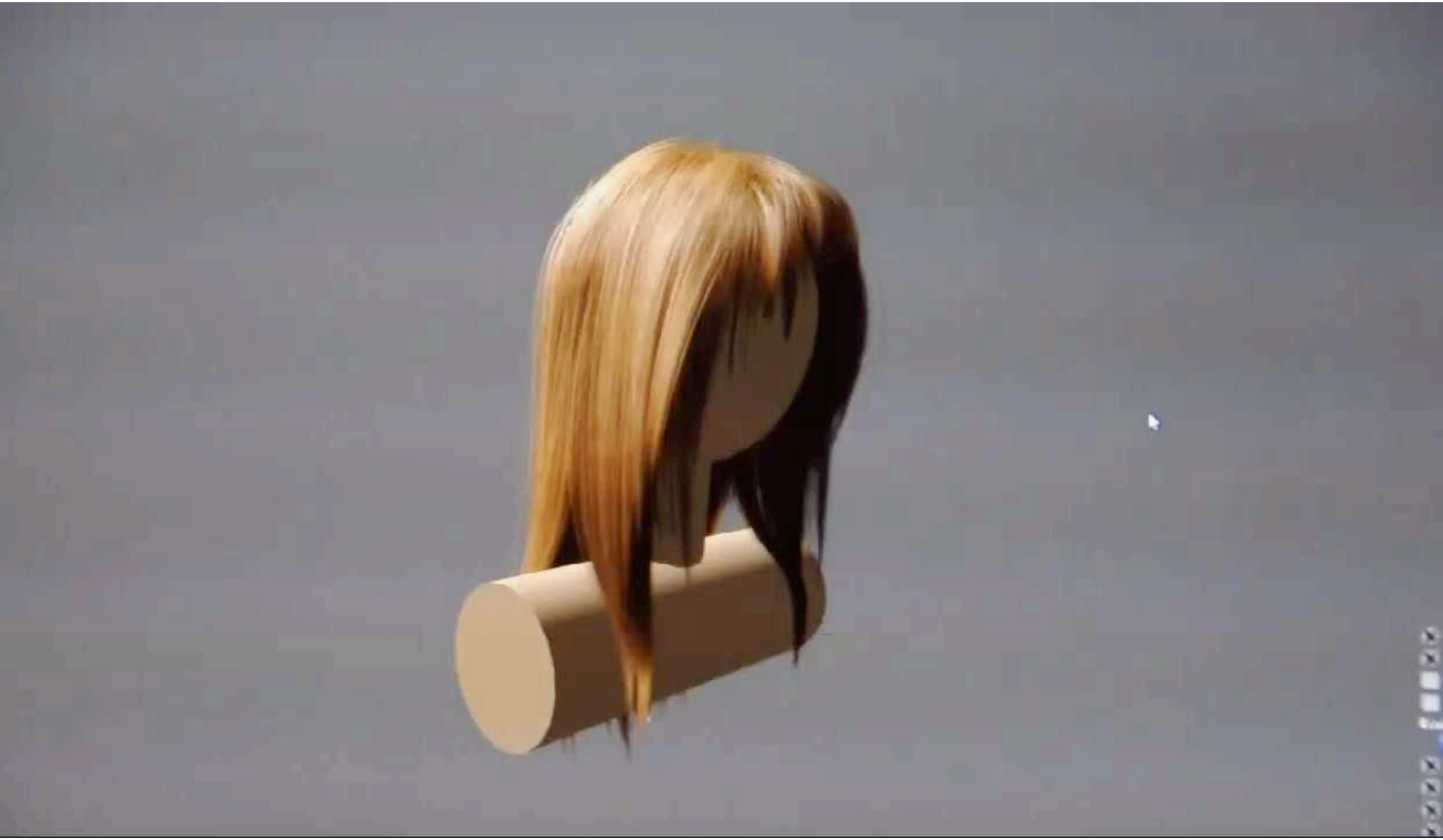
Mass Spring System:

Example of Modeling a Dynamic System

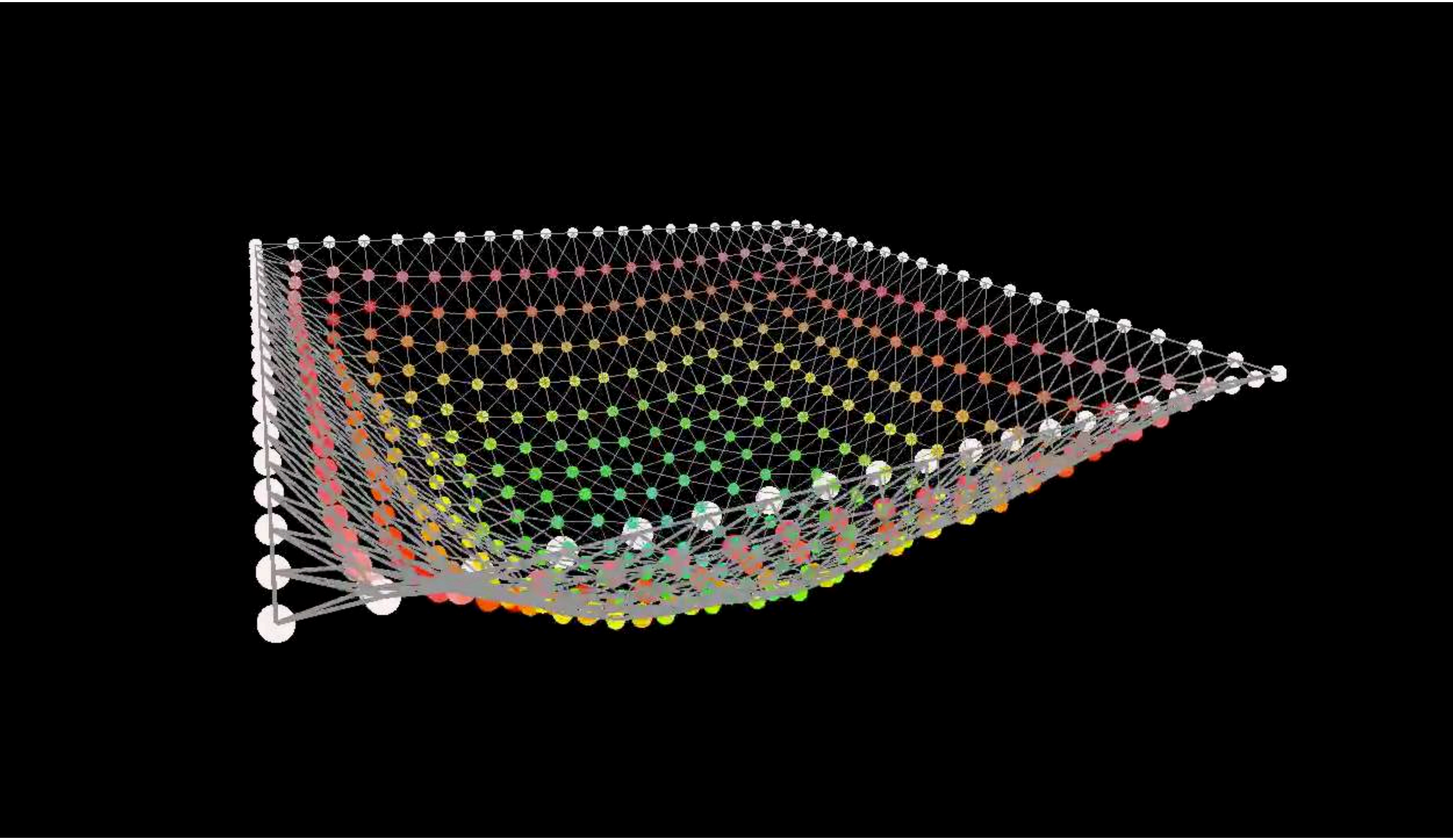
Example: Mass Spring Rope

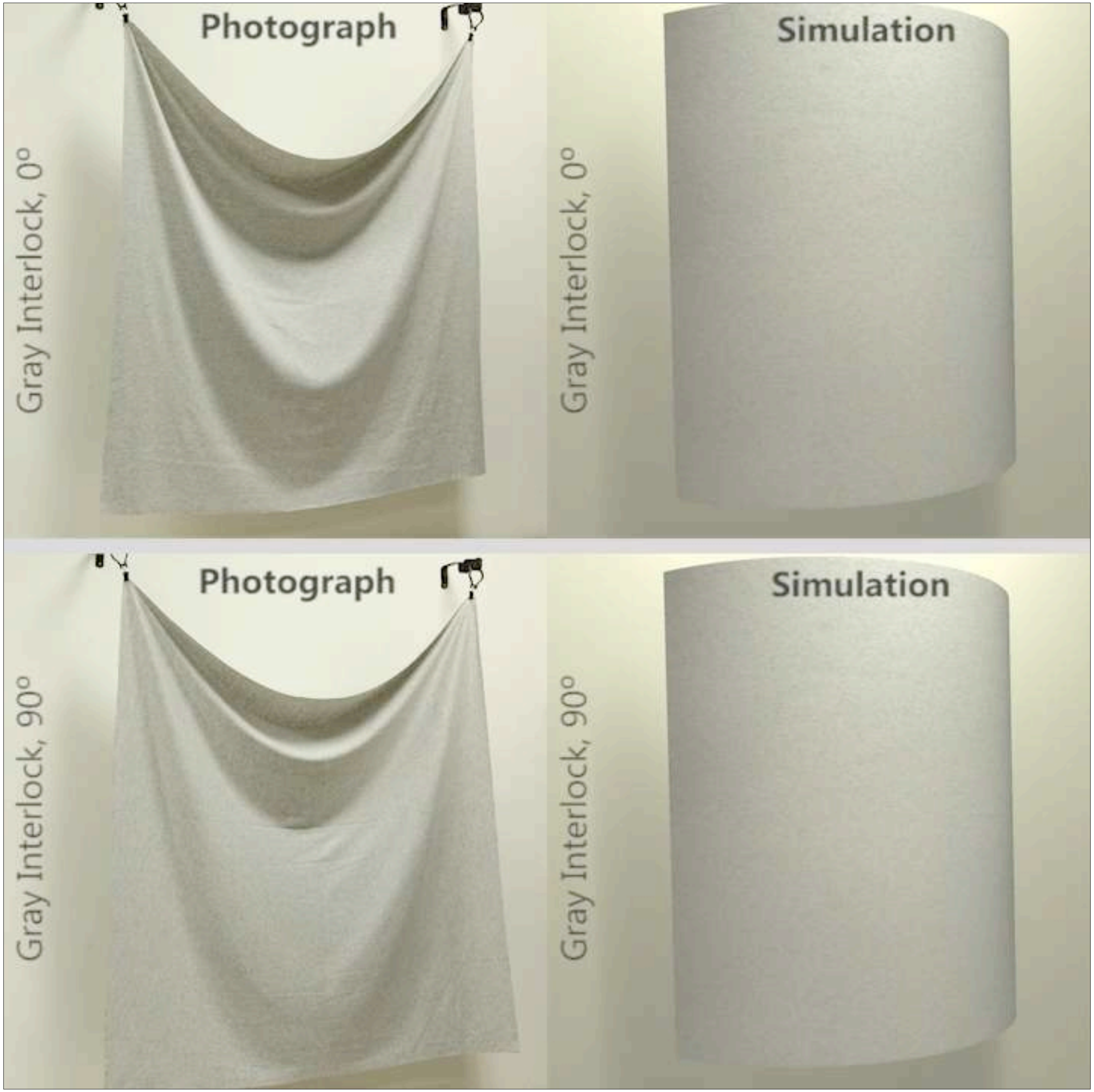
https://youtu.be/Co8enp8CH34

Example: Hair



Example: Mass Spring Mesh

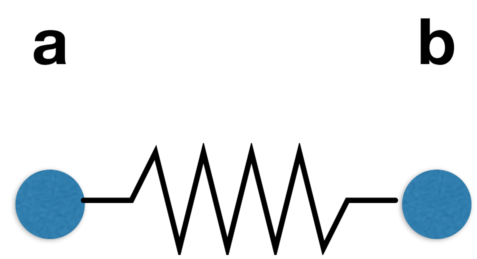




Huamin Wang, Ravi Ramamoorthi, and James F. O'Brien. "Data-Driven Elastic Models for Cloth: Modeling and Measurement". *ACM Transactions on Graphics*, 30(4):71:1–11, July 2011. Proceedings of ACM SIGGRAPH 2011, Vancouver, BC Canada.

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

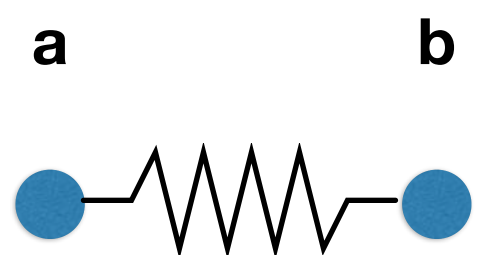
Strength proportional to displacement (Hooke's Law)

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



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

Rest length

Problem: oscillates forever

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:

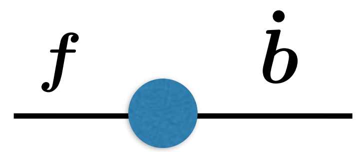
$$\boldsymbol{x}$$

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

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

Introducing Energy Loss

Simple motion damping



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

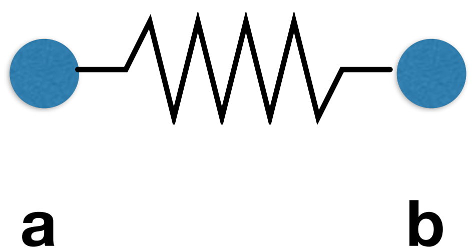
- Behaves like viscous drag on motion
- Slows down motion in the direction of velocity
- k_d is a damping coefficient

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

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

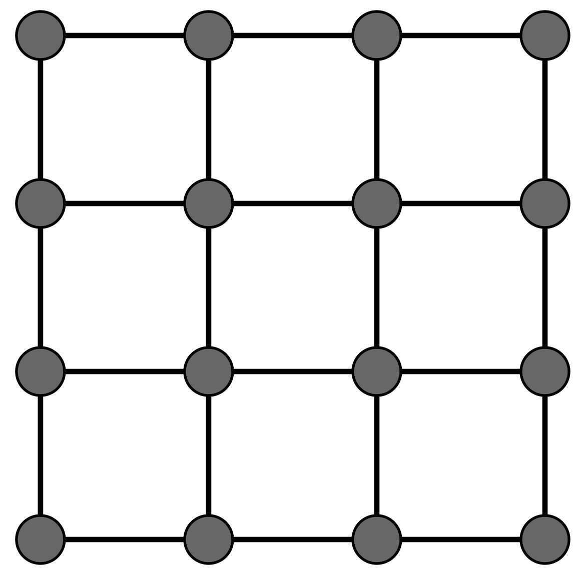
Relative velocity projected to the direction from a to b (scalar)

Direction from a to b

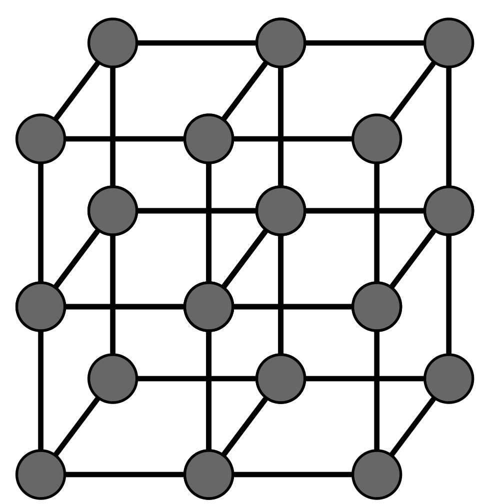
- Viscous drag only on change in spring length
  - Won't slow group motion for the spring system (e.g. global translation or rotation of the group)
- Note: This is only one specific type of damping

Structures from Springs

Sheets



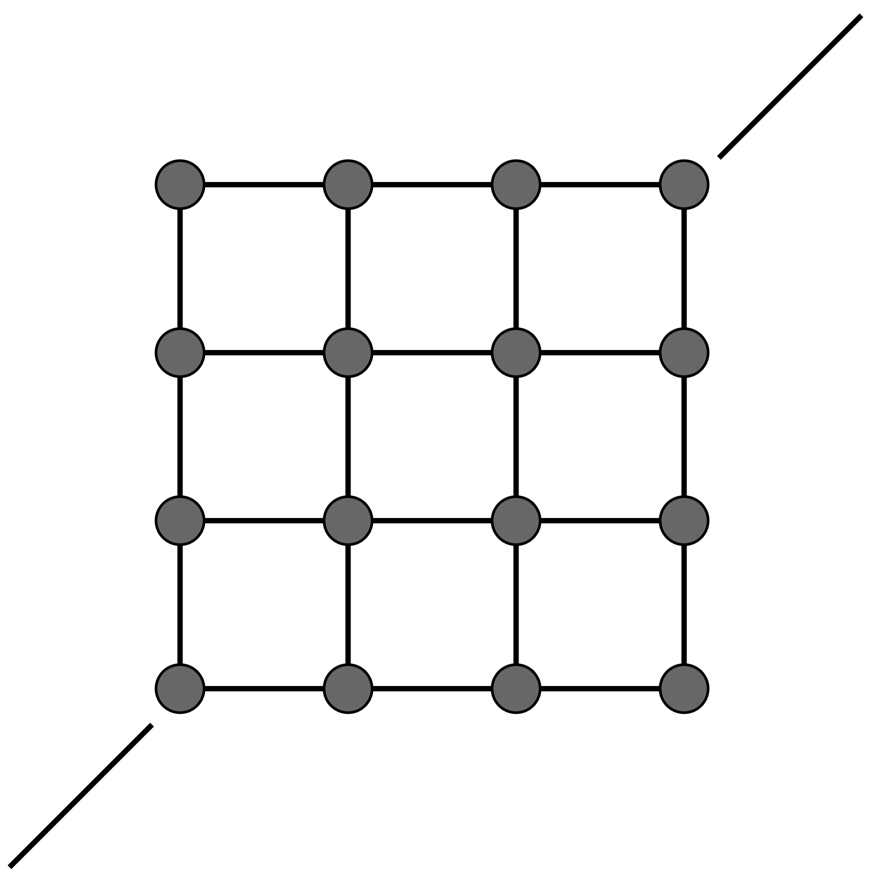
Blocks



Others

Structures from Springs

Behavior is determined by structure linkages

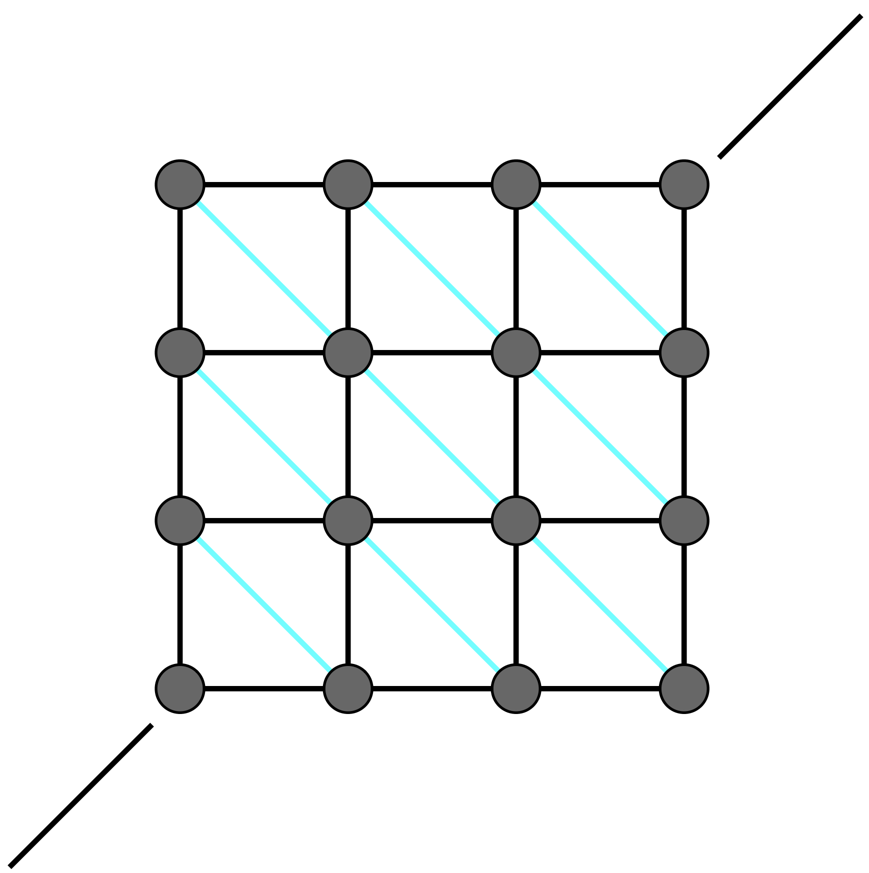


This structure will not resist shearing

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

Structures from Springs

Behavior is determined by structure linkages

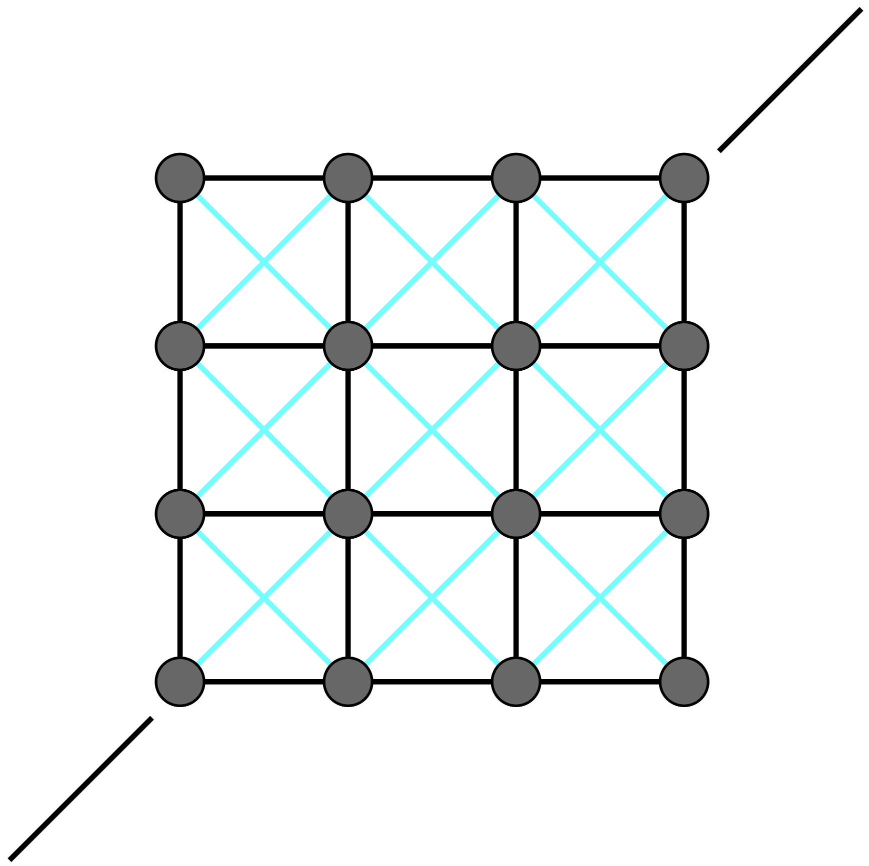


This structure will resist shearing but has anisotropic bias

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

Structures from Springs

Behavior is determined by structure linkages

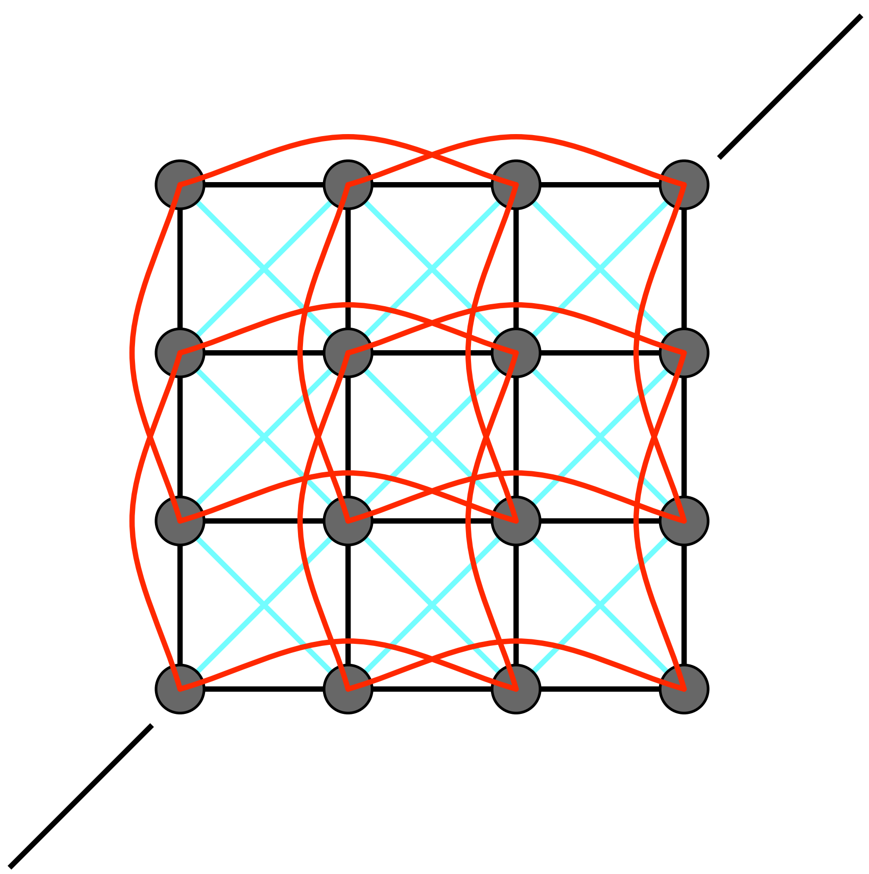


This structure will resist shearing.
Less directional bias.

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 Dress + Character

Aside: FEM (Finite Element Method) Instead of Springs

Particle Systems

Particle Systems

Model dynamical systems as collections of large numbers of particles

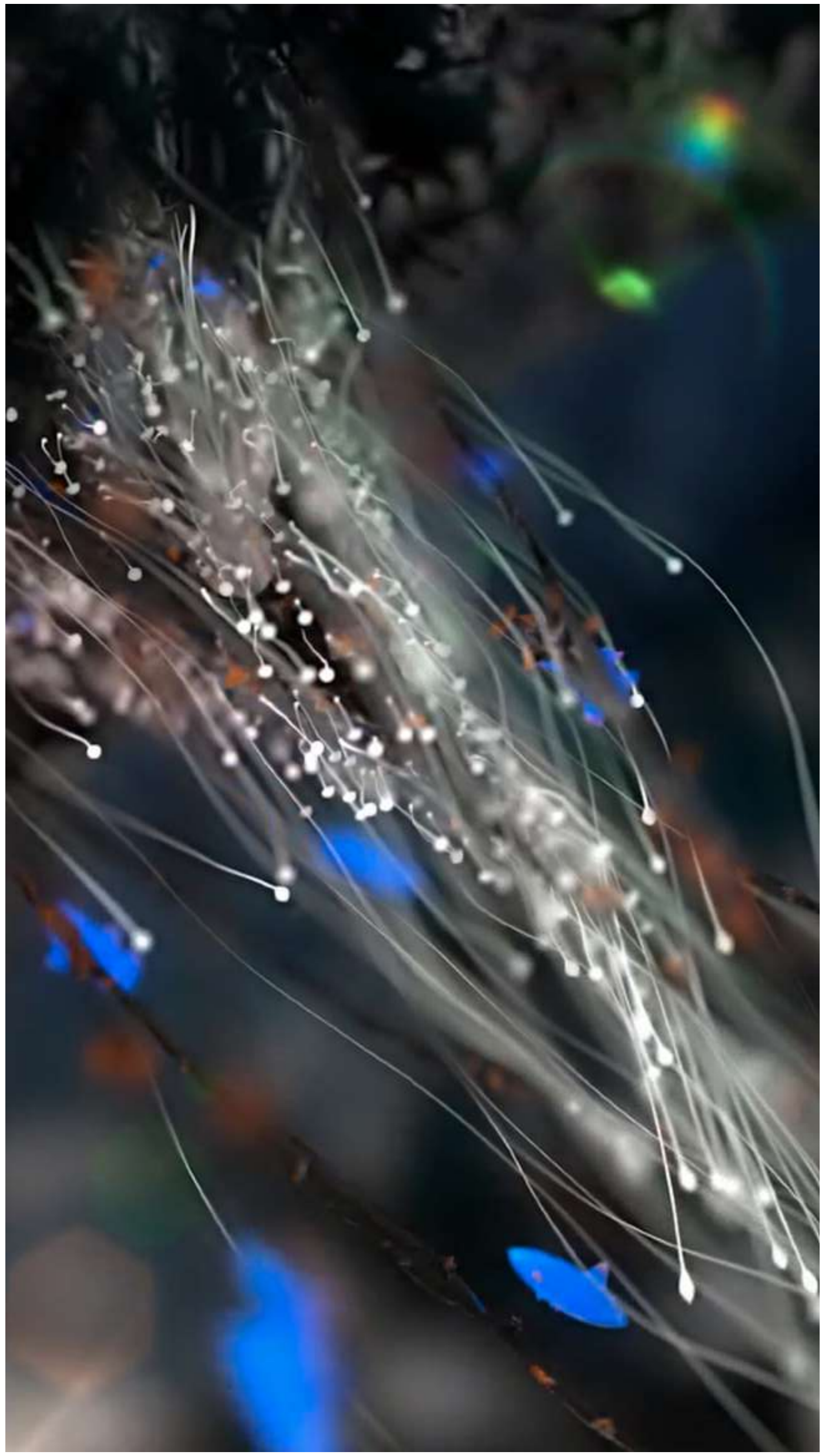
Each particle's motion is defined by a set of physical (or non-physical) forces

Popular technique in graphics and games

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

Challenges

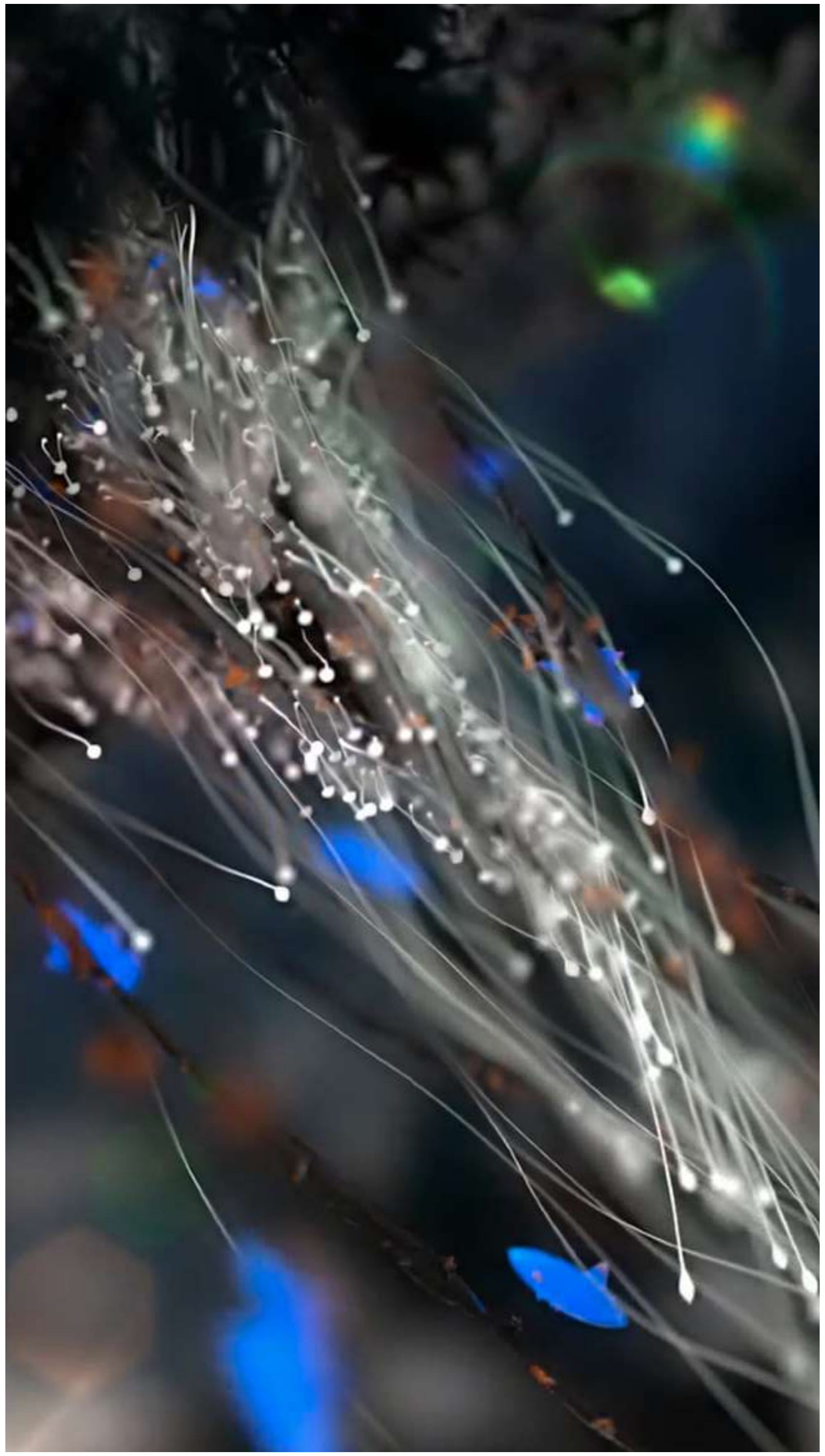
- May need *many* particles (e.g. fluids)
- May need acceleration structures (e.g. to find nearest particles for interactions)



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, …

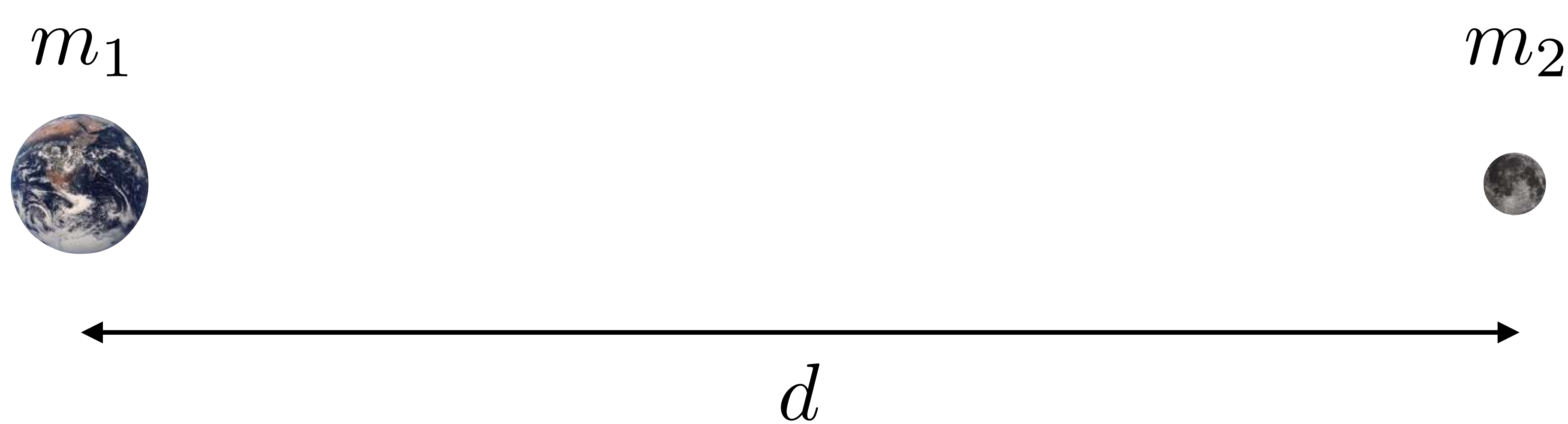
Gravitational Attraction

Newton's universal law of gravitation

- Gravitational pull between particles

$$F_g = G\frac{m_1 m_2}{d^2}$$

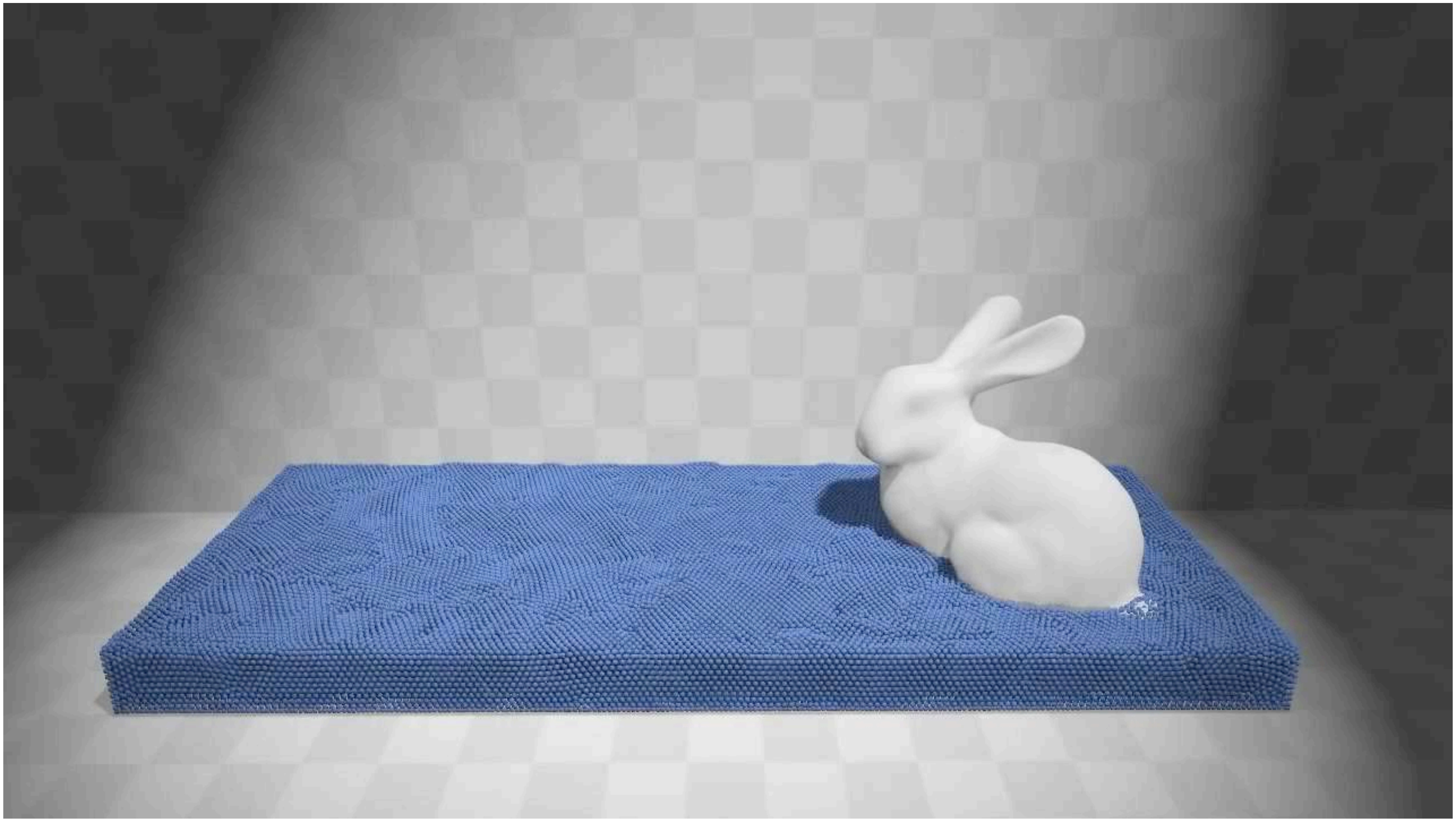
$$G = 6.67428 \times 10^{-11}\,\mathrm{Nm}^2\mathrm{kg}^{-2}$$



Example: Galaxy Simulation

Disk galaxy simulation, NASA Goddard

Example: Particle-Based Fluids



Macklin and Müller, Position Based Fluids

Simulated Flocking as an ODE

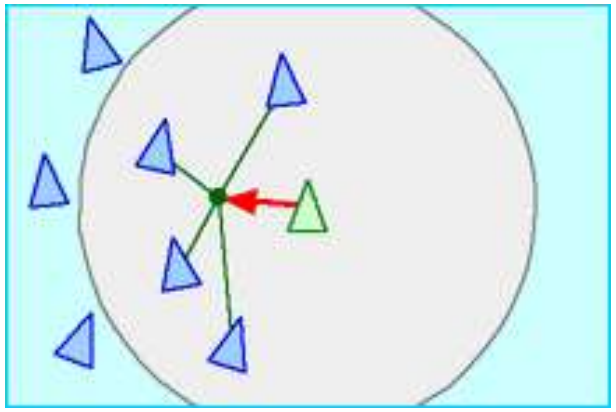
Model each bird as a particle

Subject to very simple forces:

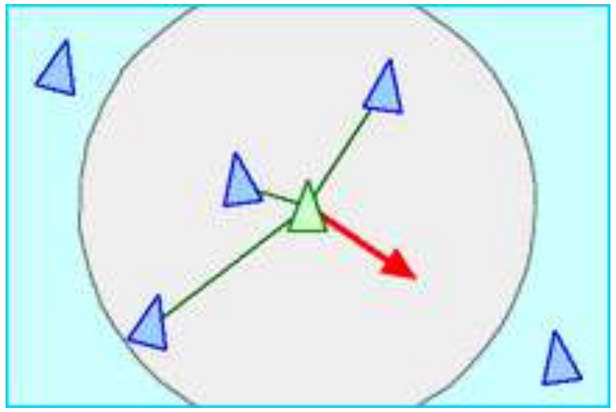
- attraction to center of neighbors
- repulsion from individual neighbors
- alignment toward average trajectory of neighbors

Simulate evolution of large particle system numerically

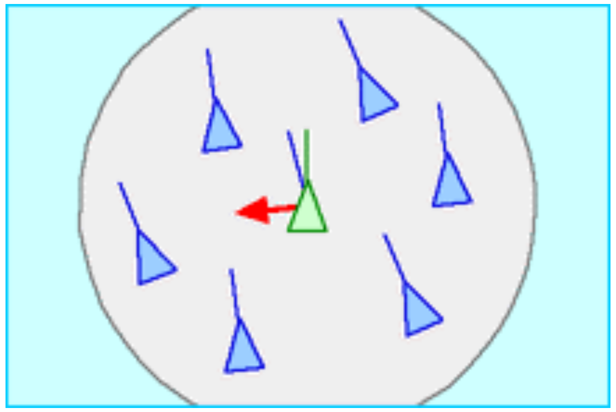
Emergent complex behavior (also seen in fish, bees, ...)



attraction



repulsion

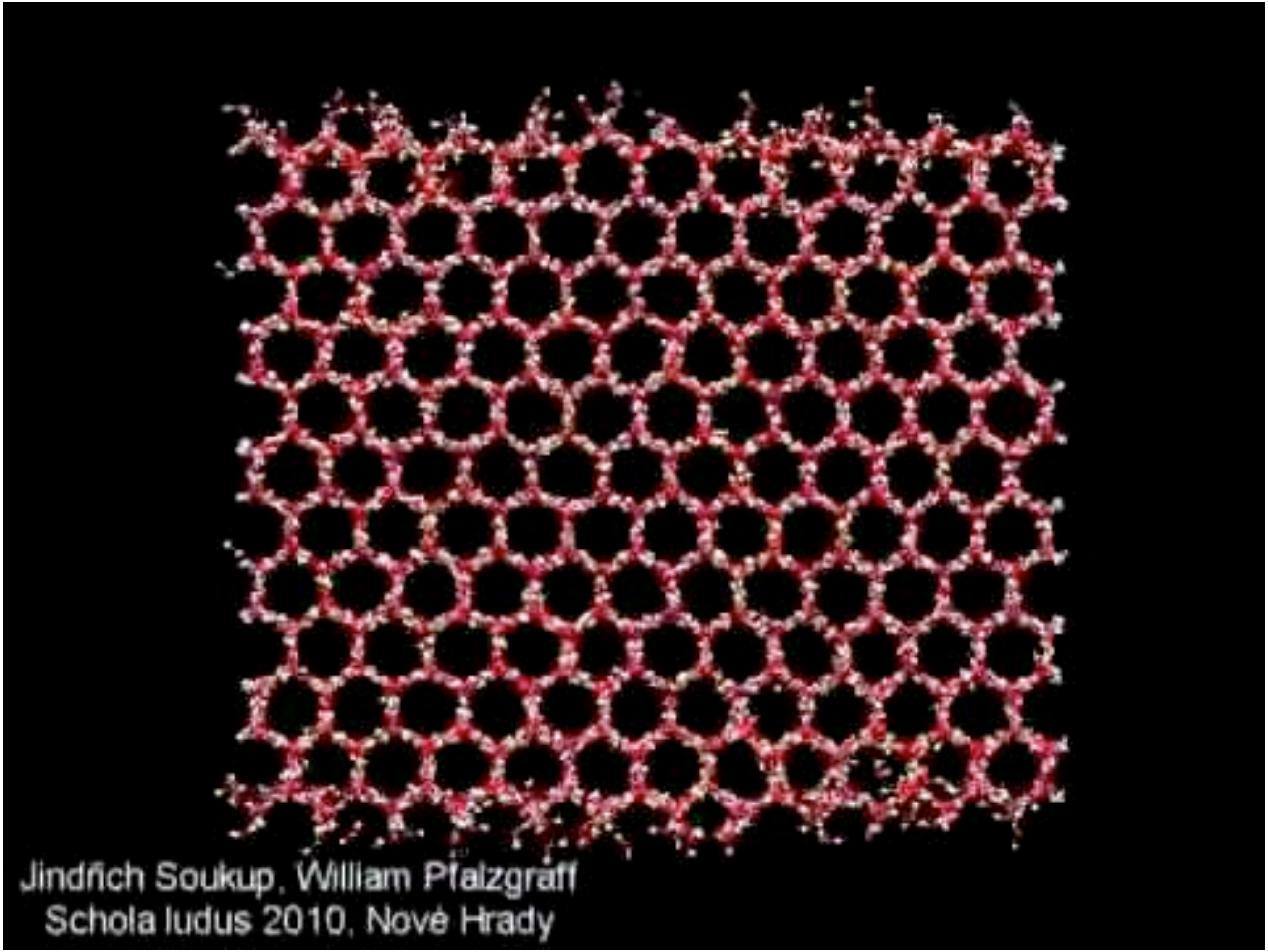


alignment

Credit: Craig Reynolds (see http://www.red3d.com/cwr/boids/)

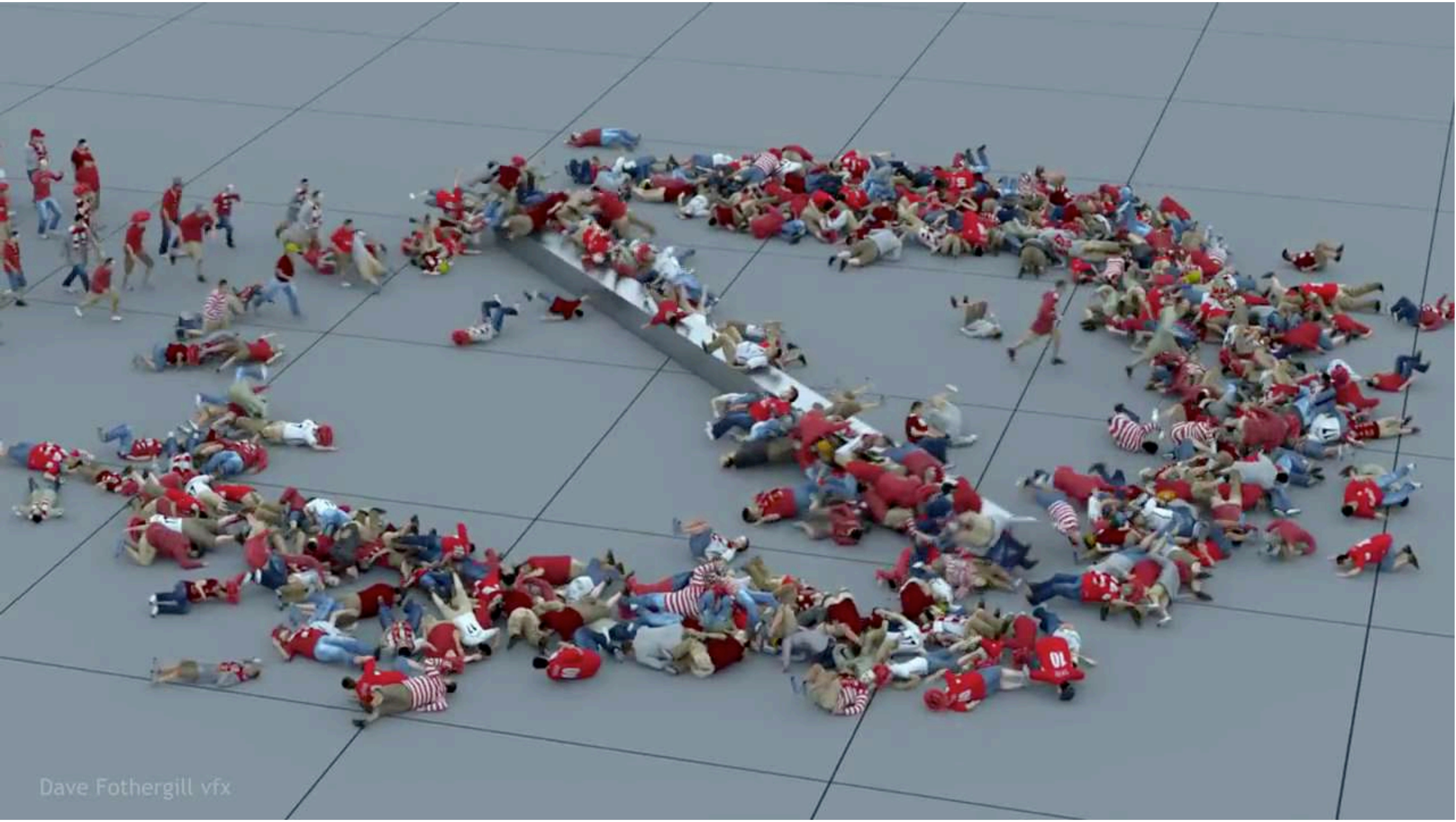
Slide credit: Keenan Crane

Example: Molecular Dynamics



(model of melting ice crystal)

Example: Crowds + "Rock" Dynamics



Forward Kinematics

(Slides by Prof. James O'Brien)

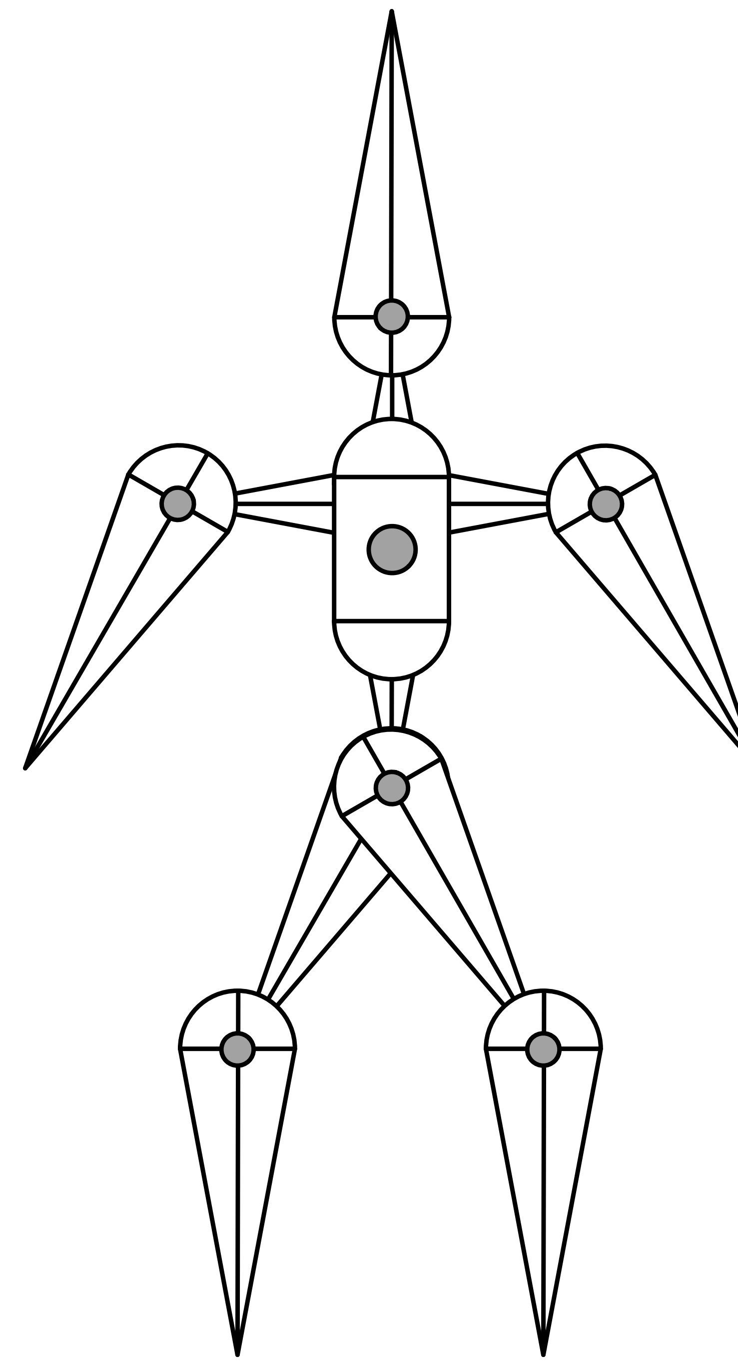
Forward Kinematics

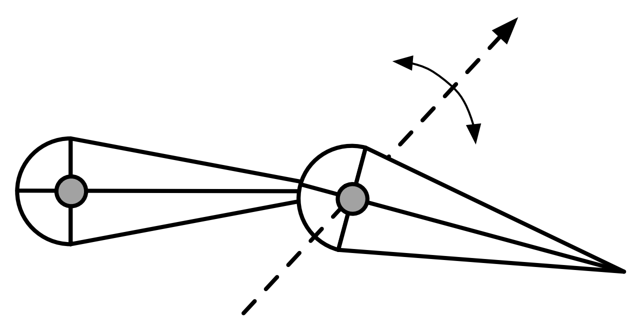
Articulated skeleton

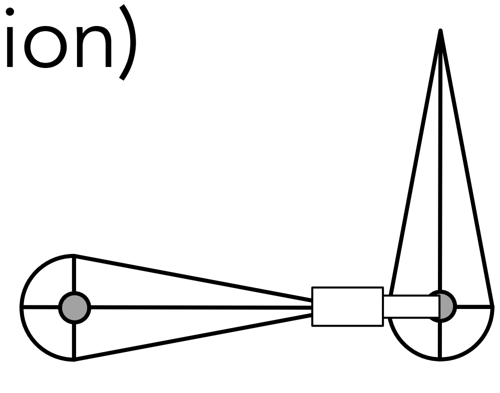
- Topology (what's connected to what)
- Geometric relations from joints
- Tree structure (in absence of loops)

Joint types

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

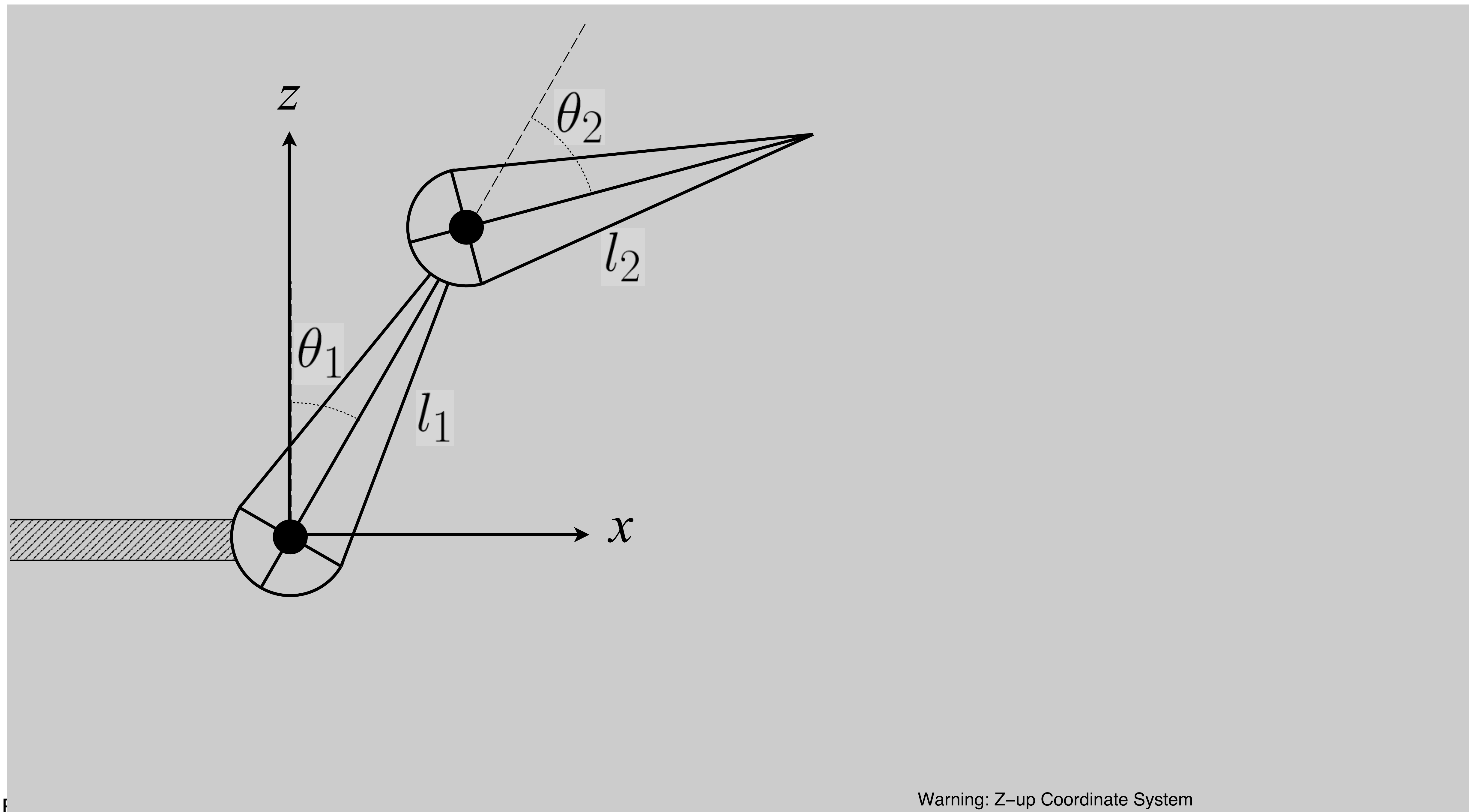






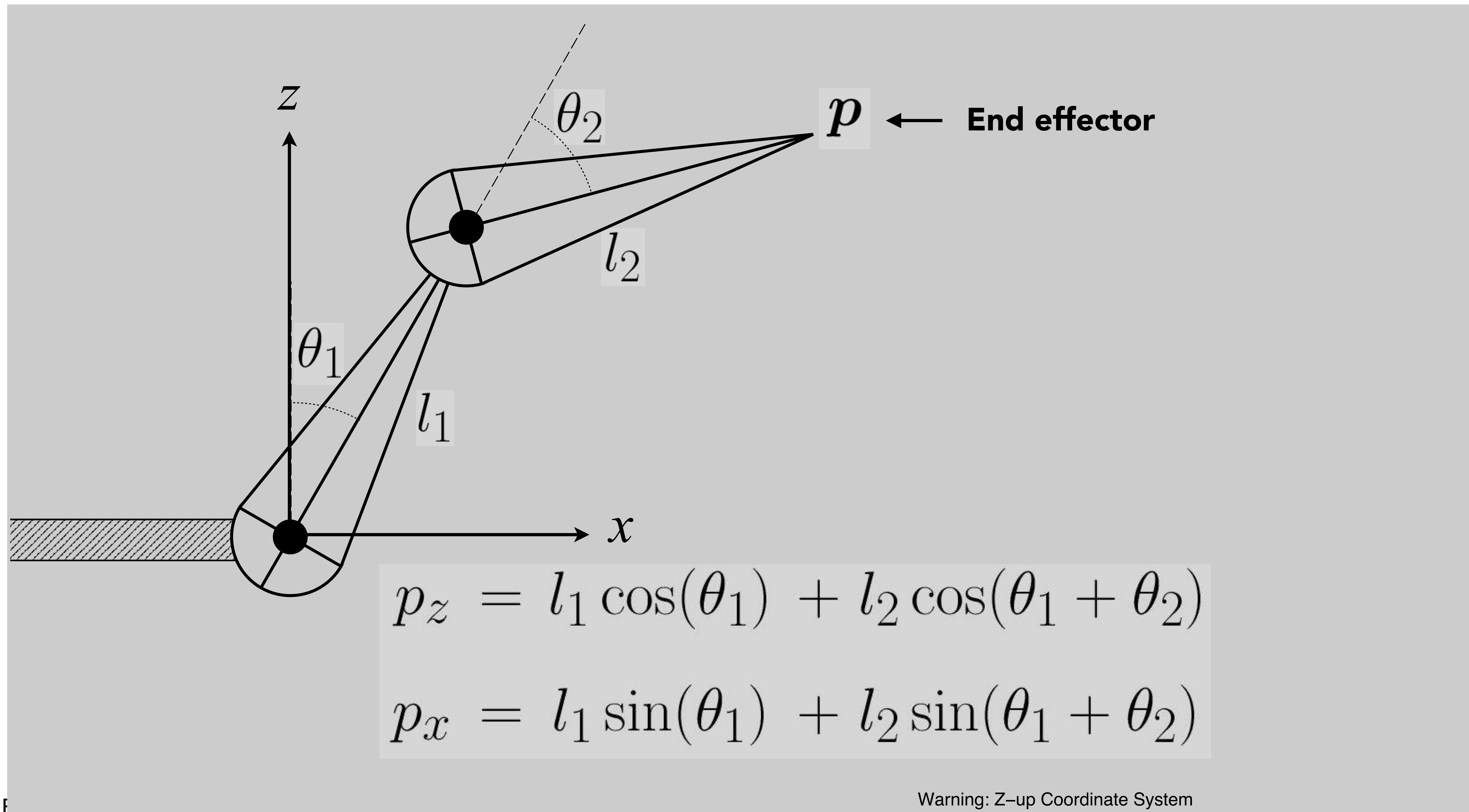
Forward Kinematics

Example: simple two segment arm in 2D



Forward Kinematics

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

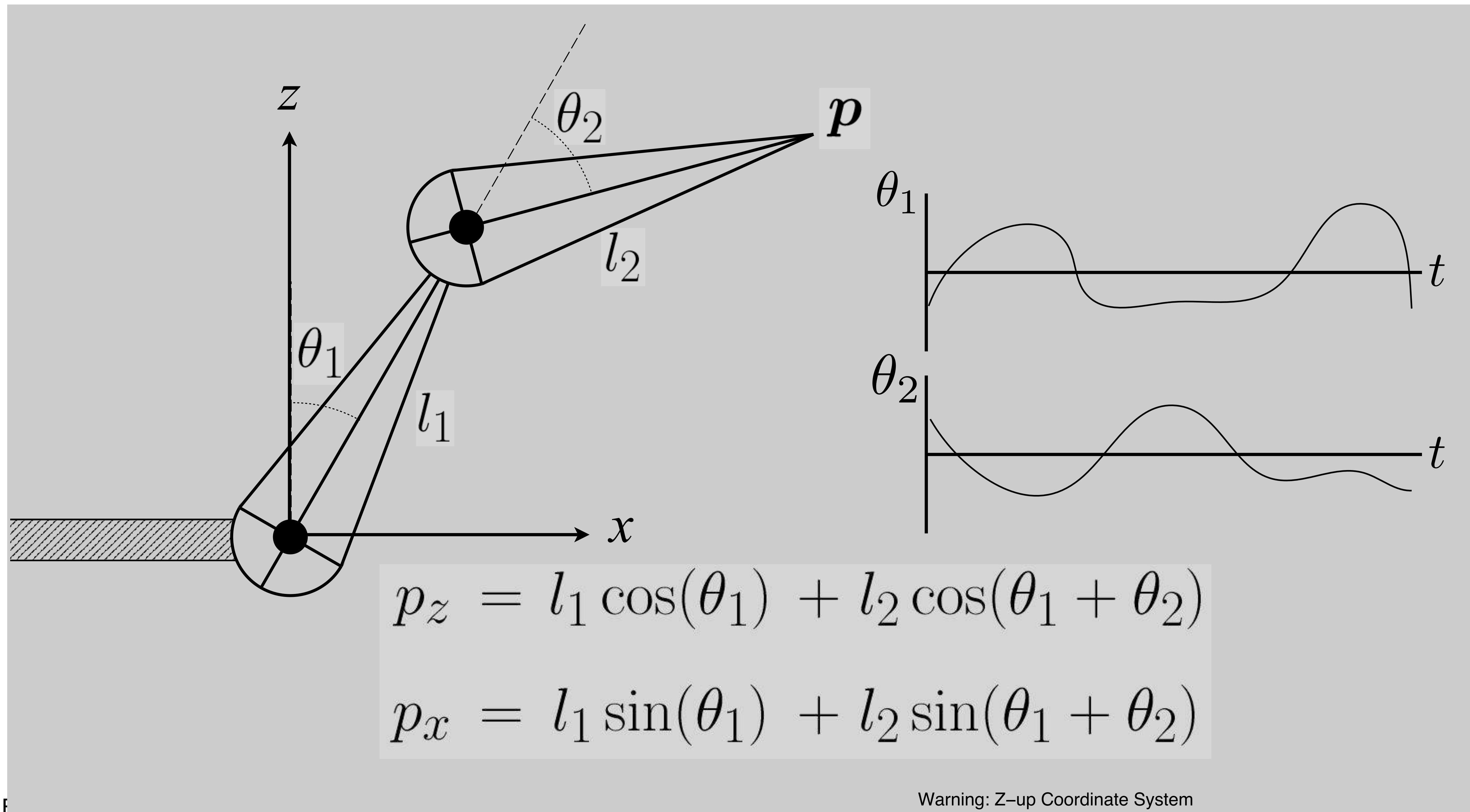


$$p_z = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2)$$

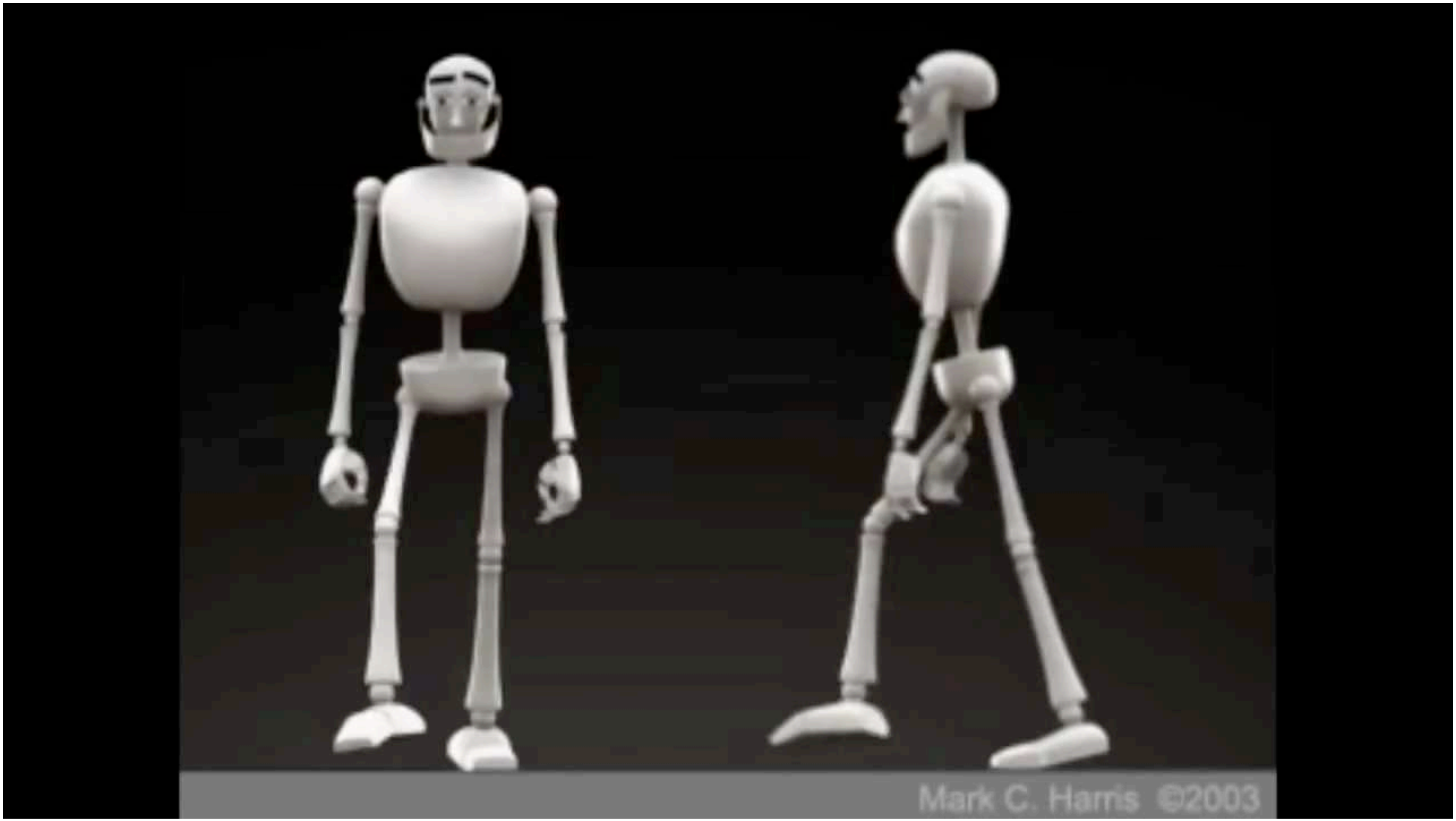
$$p_x = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2)$$

Forward Kinematics

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



Example Walk Cycle



Kinematics Pros and Cons

Strengths

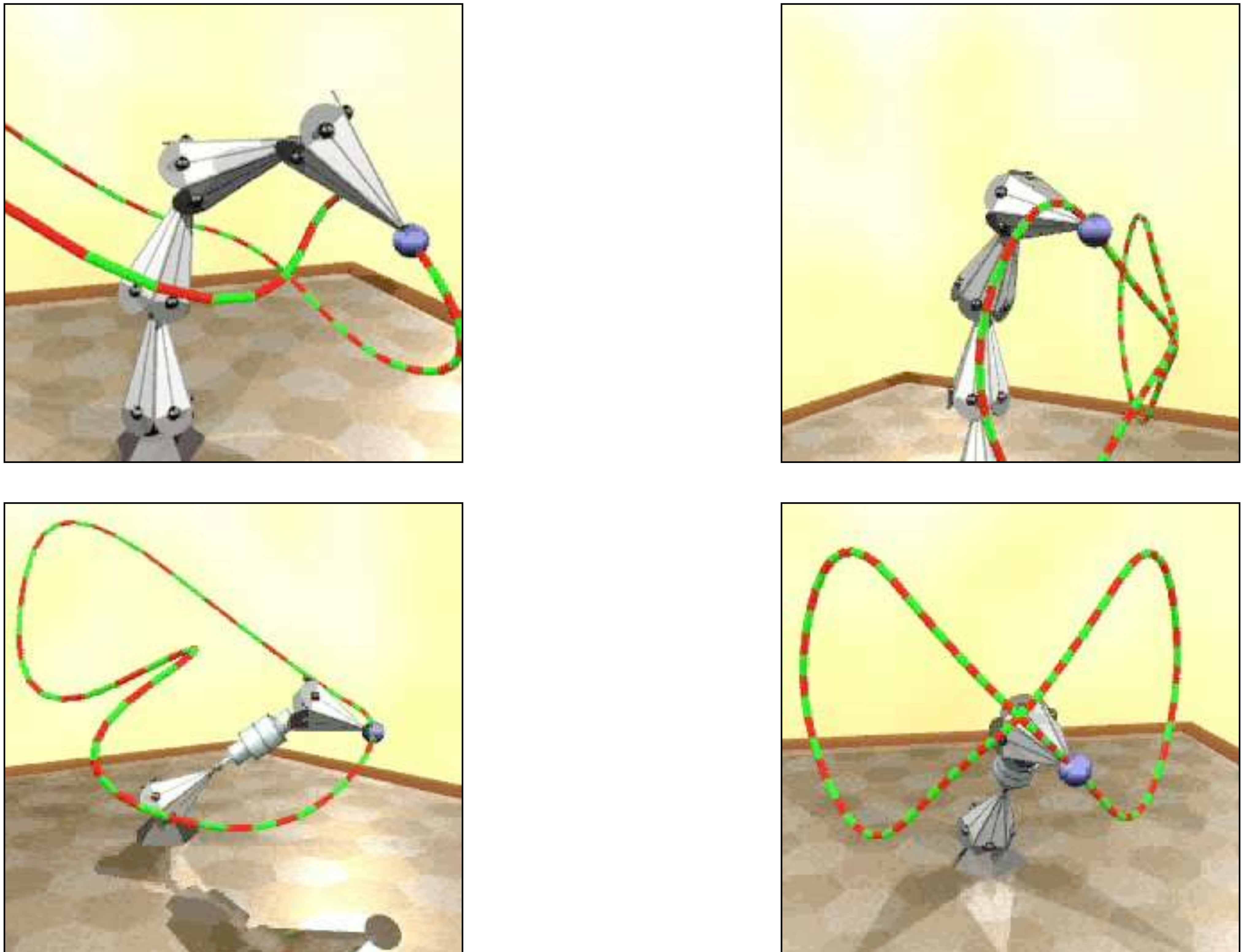
- Direct control is convenient
- Implementation is straightforward

Weaknesses

- Animation may be inconsistent with physics
- Time consuming for artists

Inverse Kinematics

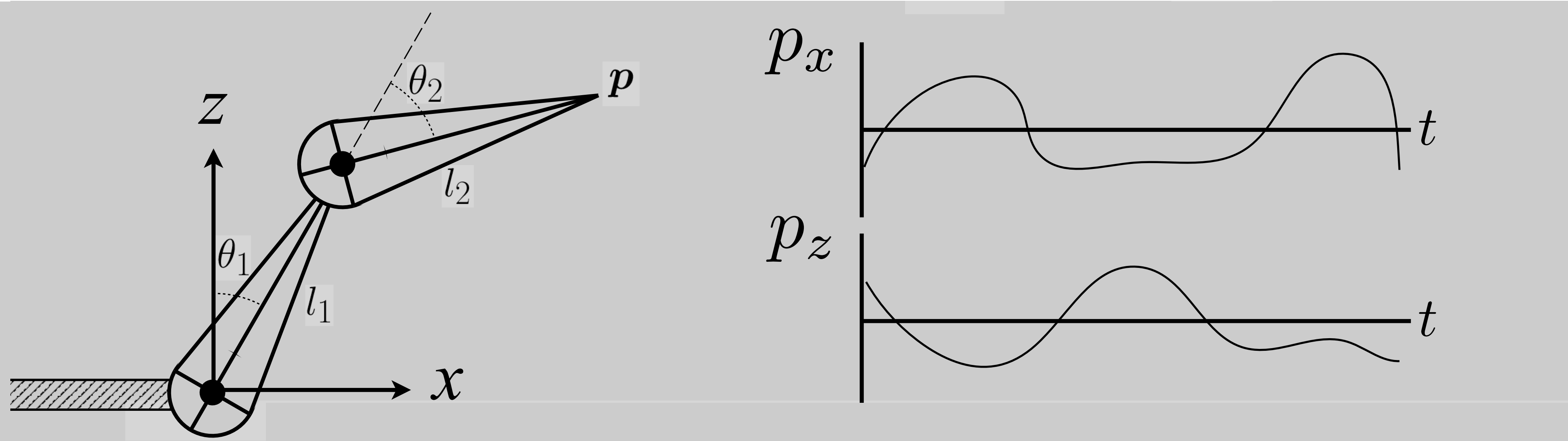
Inverse Kinematics



Egon Pasztor

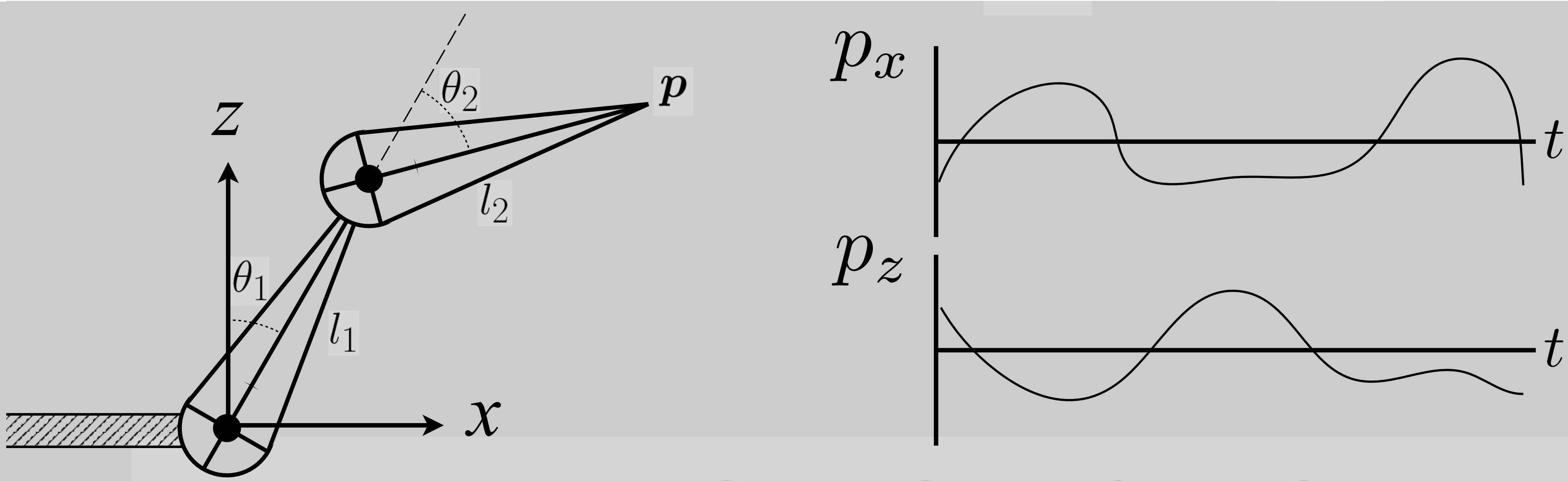
Inverse Kinematics

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



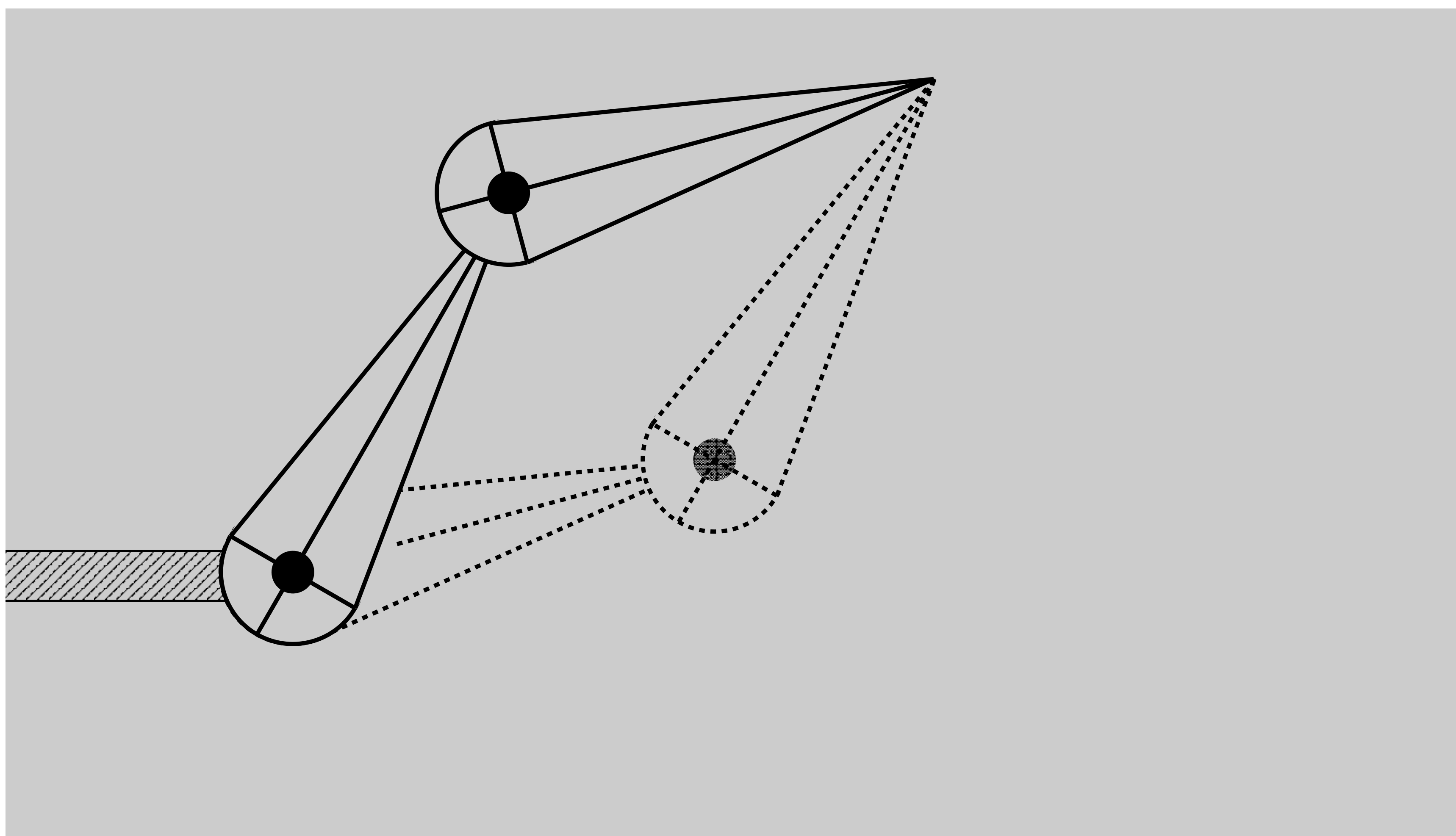
$$\theta_2 = \cos^{-1}\left(\frac{p_z^2 + p_x^2 - l_1^2 - l_2^2}{2 l_1 l_2}\right)$$

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

Inverse Kinematics

Why is the problem hard?

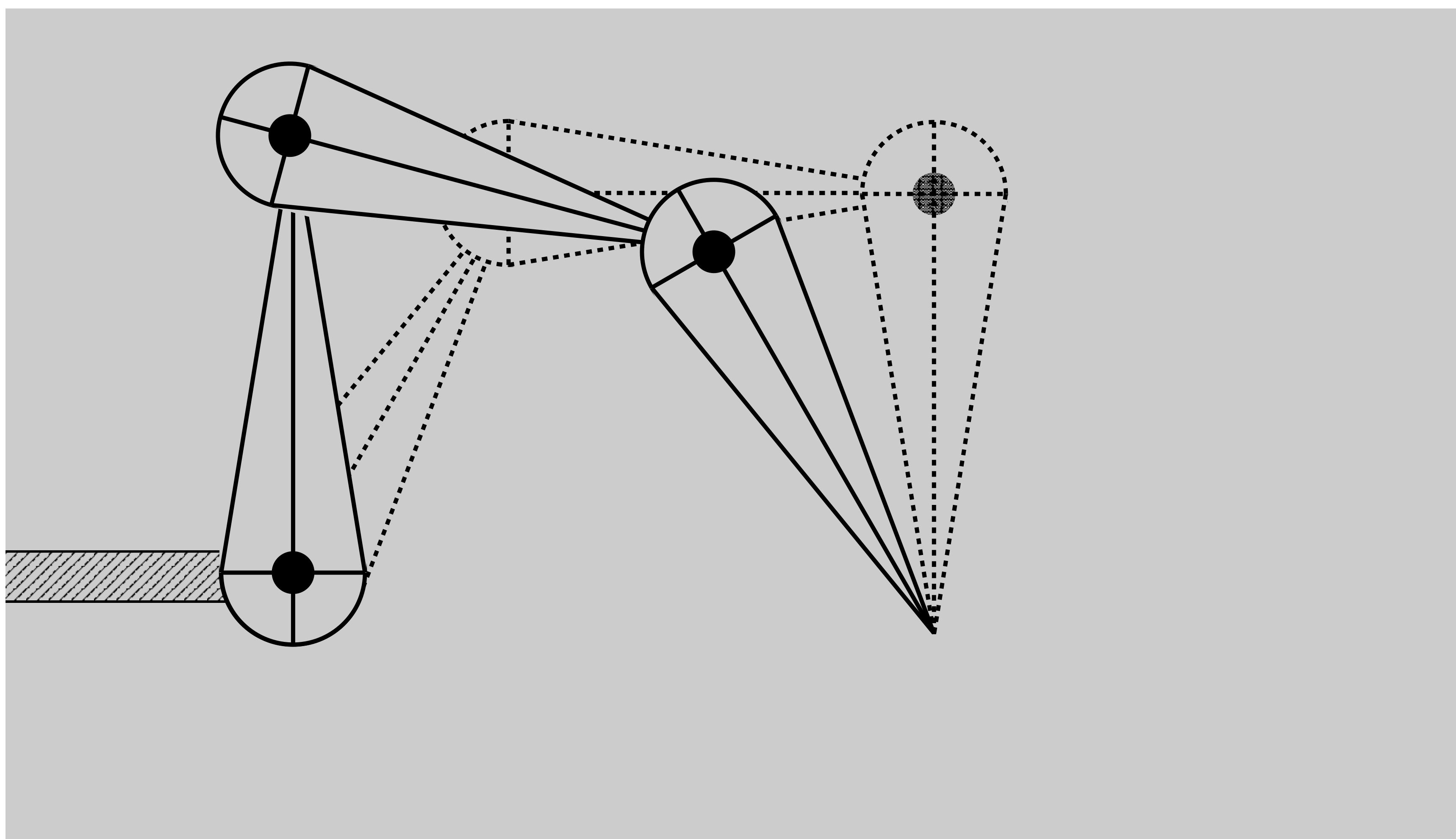
- Multiple solutions in configuration space



Inverse Kinematics

Why is the problem hard?

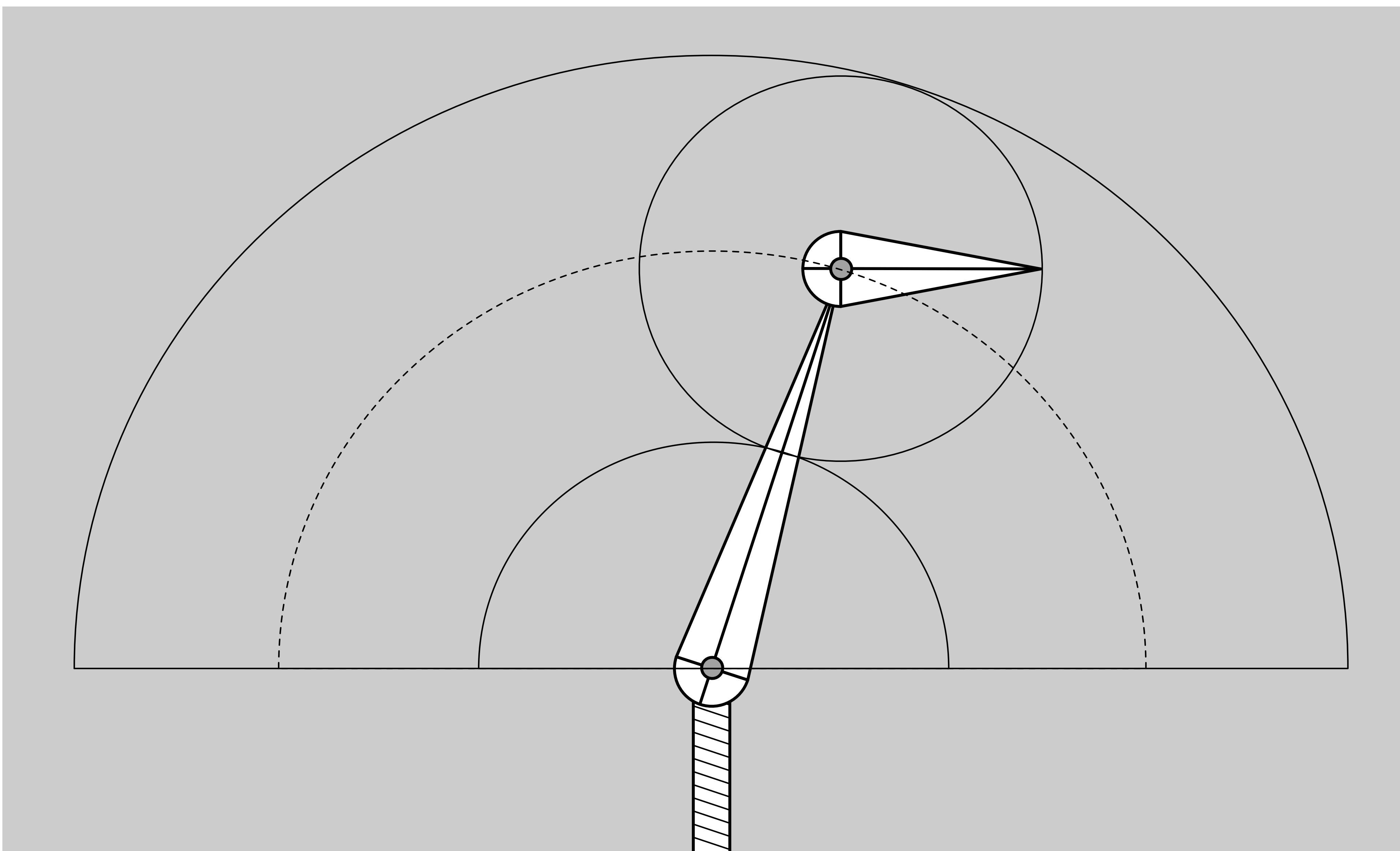
- Multiple solutions in configuration space



Inverse Kinematics

Why is the problem hard?

- Solutions may not always exist

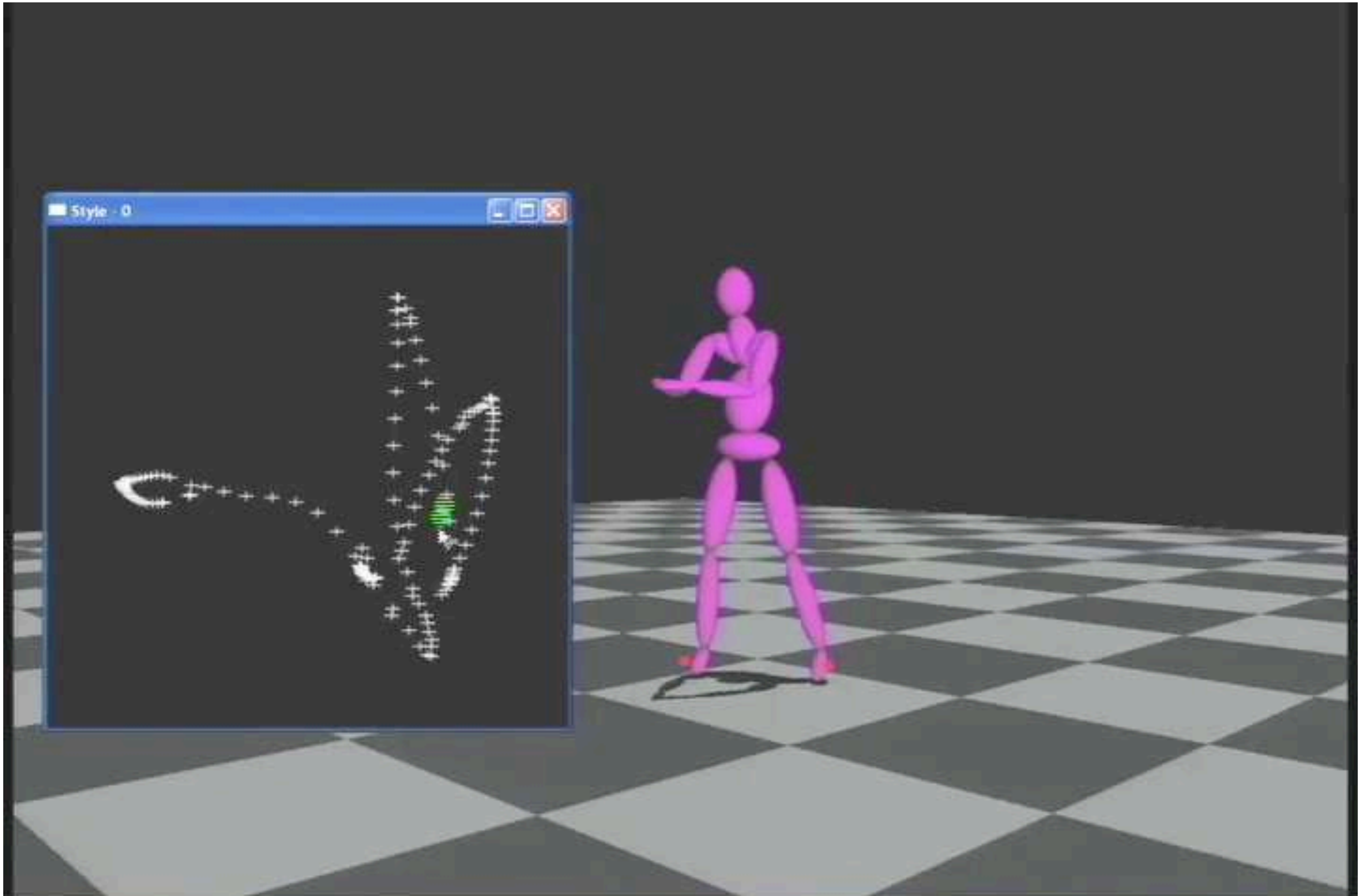


Inverse Kinematics

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)
- Compute gradient of error as function of configuration
- Apply gradient descent (or Newton's method, or other optimization procedure)

Style-Based IK



Grochow et al., Style Based Inverse Kinematics

Rigging

Rigging

Rigging is a set of higher level controls on a character that allow more rapid & intuitive modification of pose, deformations, expression, etc.

Important

- Like strings on a puppet
- Captures all meaningful character changes
- Varies from character to character

Expensive to create

- Manual effort
- Requires both artistic and technical training

Rigging Example

Courtesy Matthew Lailler via Keenan Crane

Blend Shapes

Instead of skeleton, interpolate directly between surfaces

E.g., model a collection of facial expressions:

Simplest scheme: take linear combination of vertex positions

Spline used to control choice of weights over time

Courtesy Félix Ferrand

Blend Shapes

Courtesy Félix Ferrand

Motion Capture

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 Pros and Cons

Strengths

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

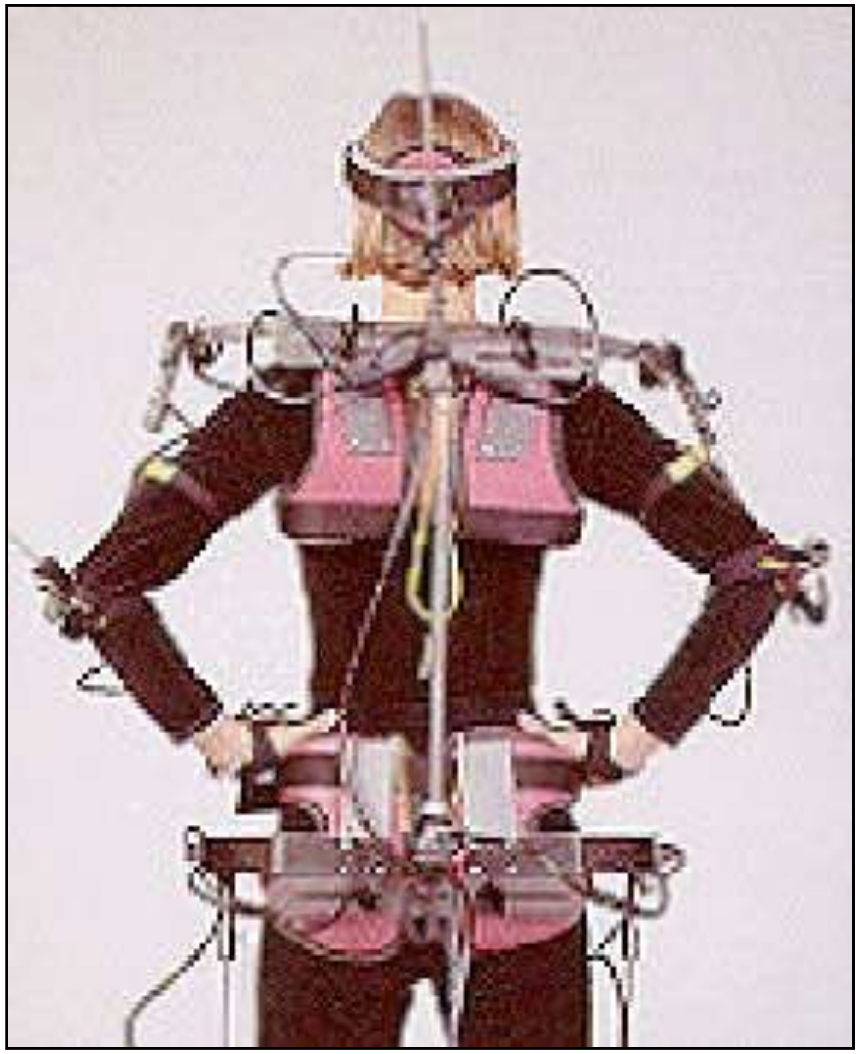
Weaknesses

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

Motion Capture Equipment

Optical
(More on following slides)

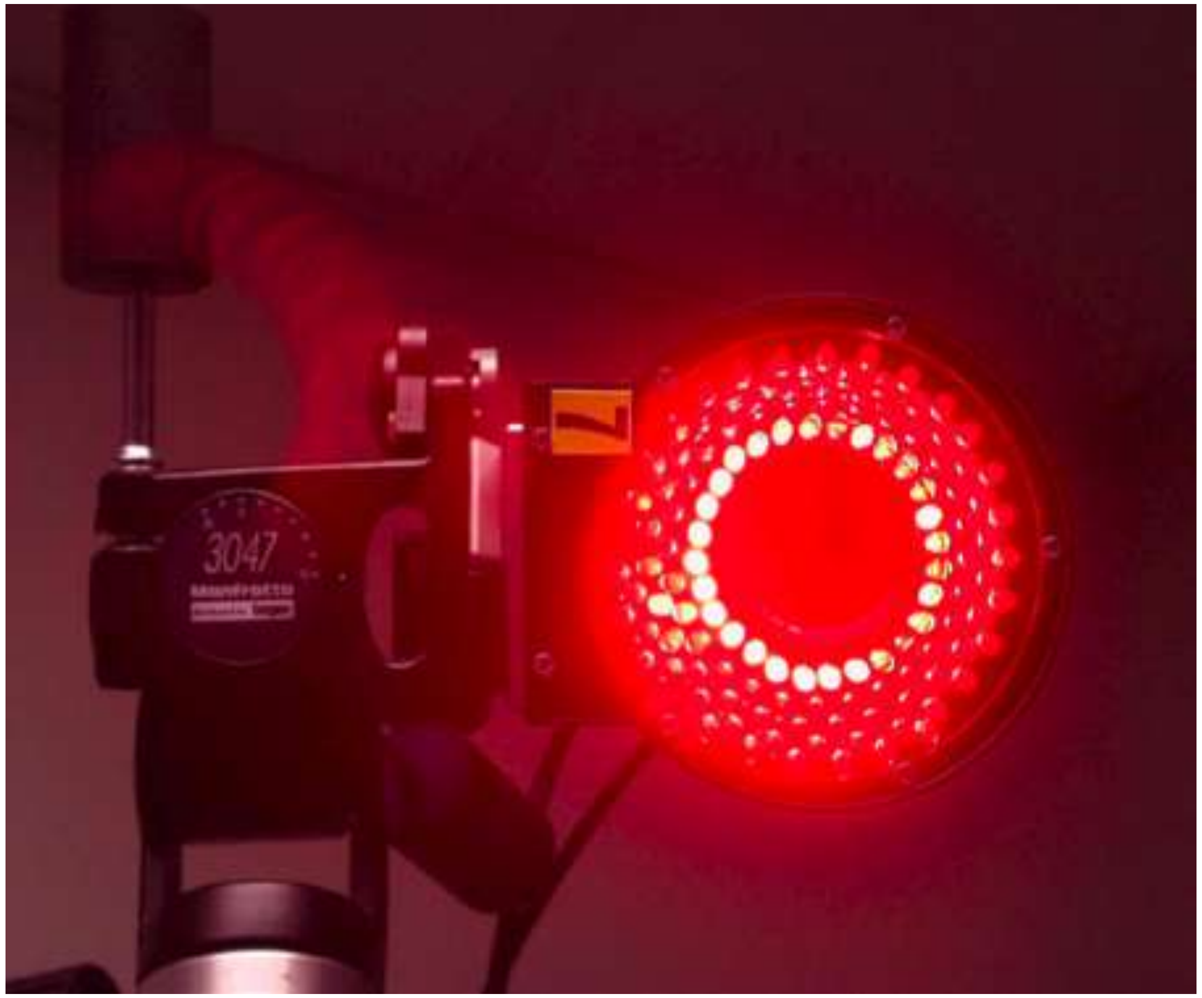
Magnetic
Sense magnetic fields to infer position / orientation. Tethered.



Mechanical
Measure joint angles directly. Restricts motion.

Optical Motion Capture

Retroflective markers attached to subject

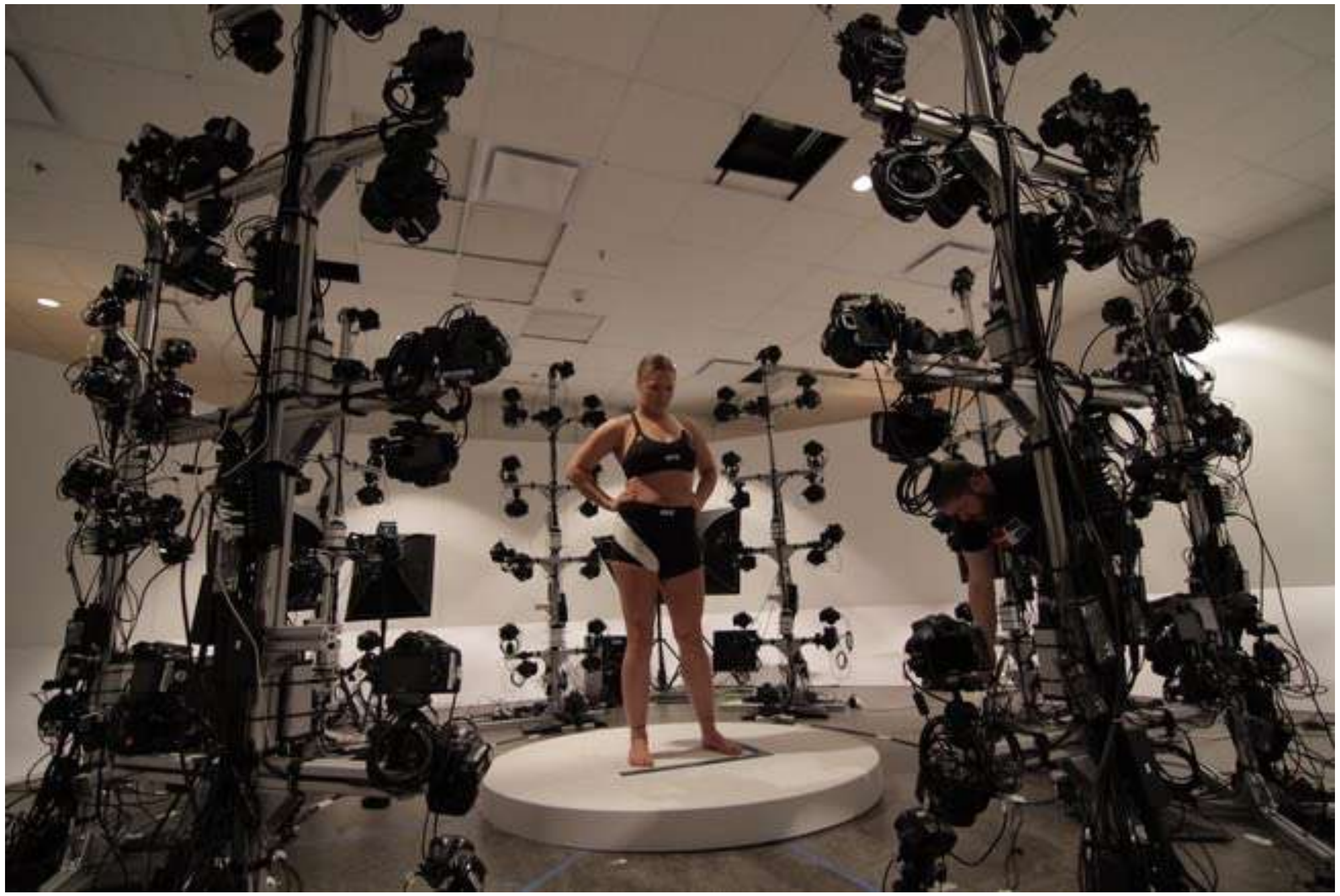


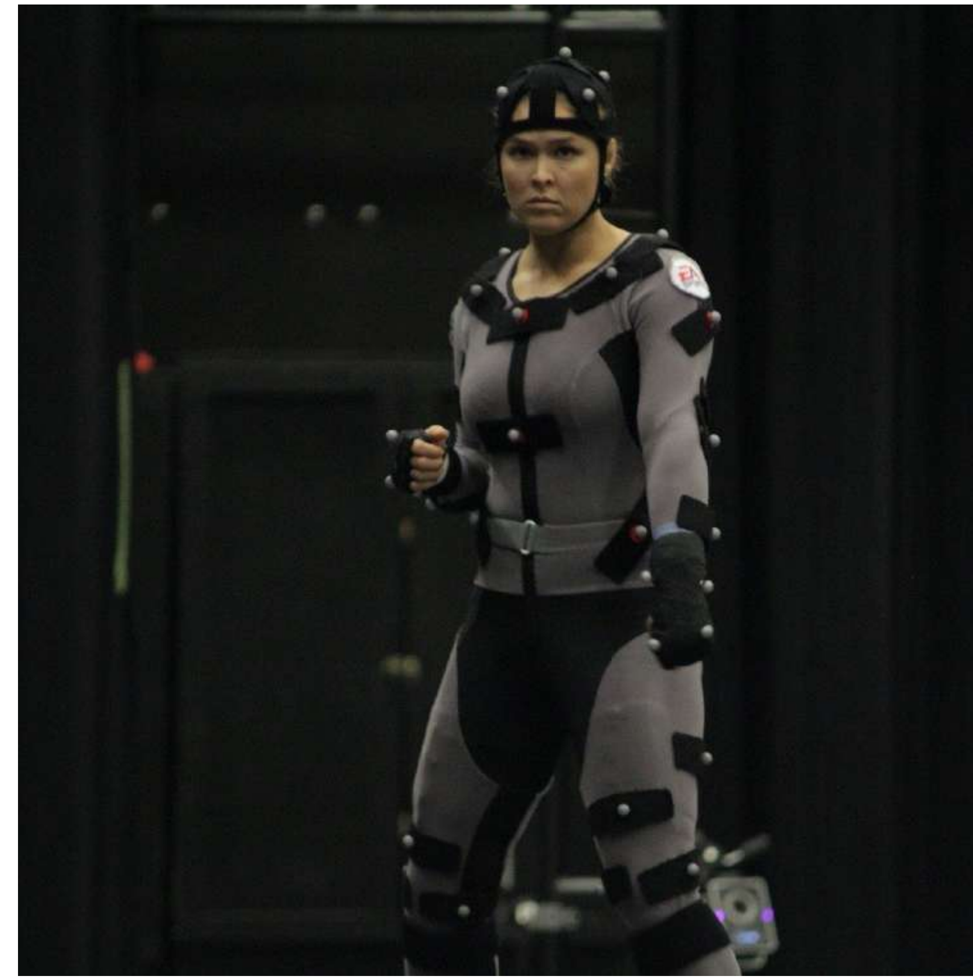
IR illumination and cameras

- 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

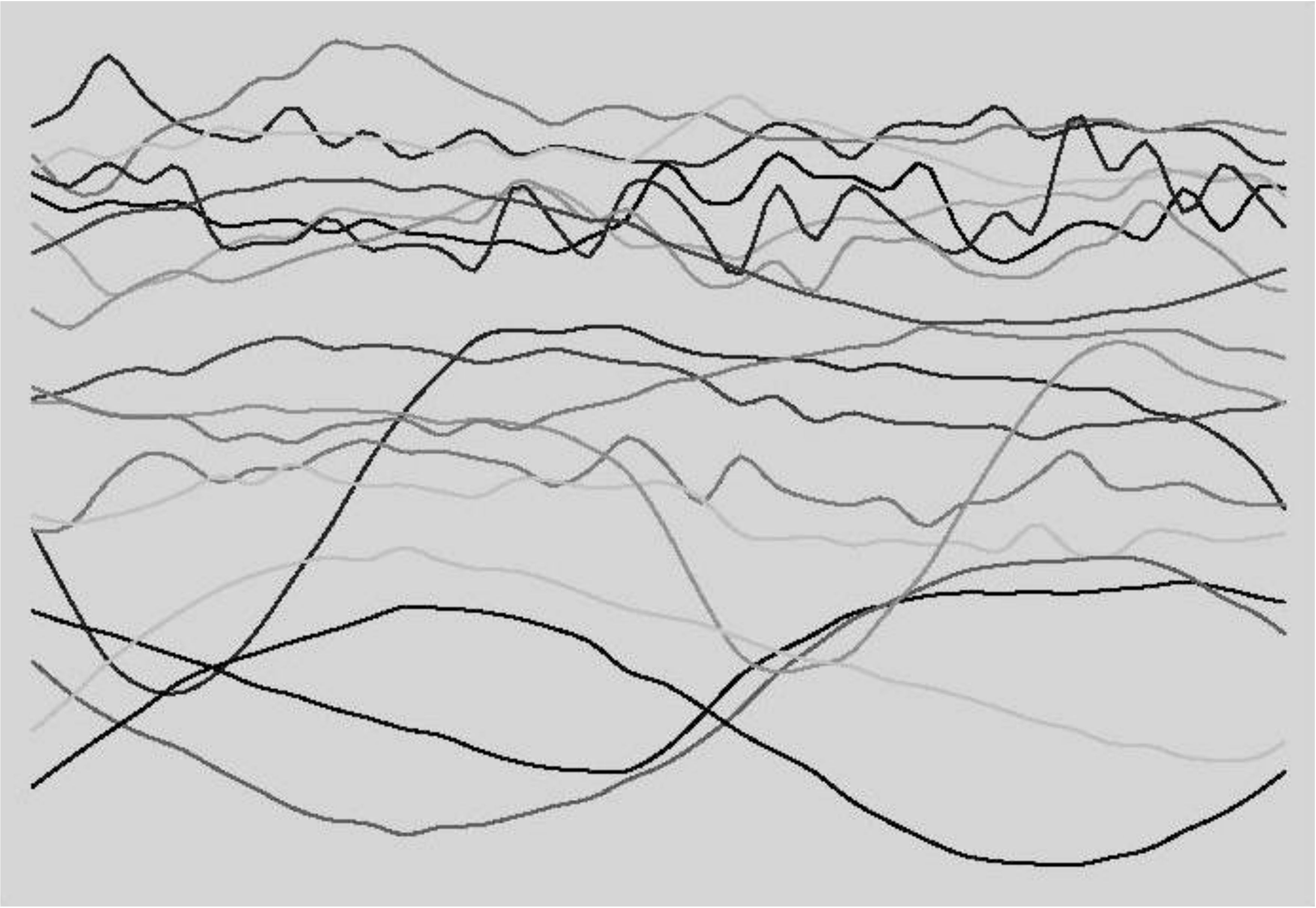




Source: http://fightland.vice.com/blog/ronda-rousey-20-the-queen-of-all-media

Ronda Roussey in Electronic Arts' motion capture studio

Motion Data



Subset of motion curves from captured walking motion.

From Witkin and Popovic, 1995

Challenges of Facial Animation

Uncanny valley (恐怖谷效应)

- In robotics and graphics
- As artificial character appearance approaches human realism, our emotional response goes negative, until it achieves a sufficiently convincing level of realism in expression

Cartoon.
Brave, Pixar

Semi-realistic. Polar Express, Warner Bros.

Facial Motion Capture

Discovery, “Avatar: Motion Capture Mirrors Emotions”, https://youtu.be/1wK1Ixr-UmM

The Production Pipeline

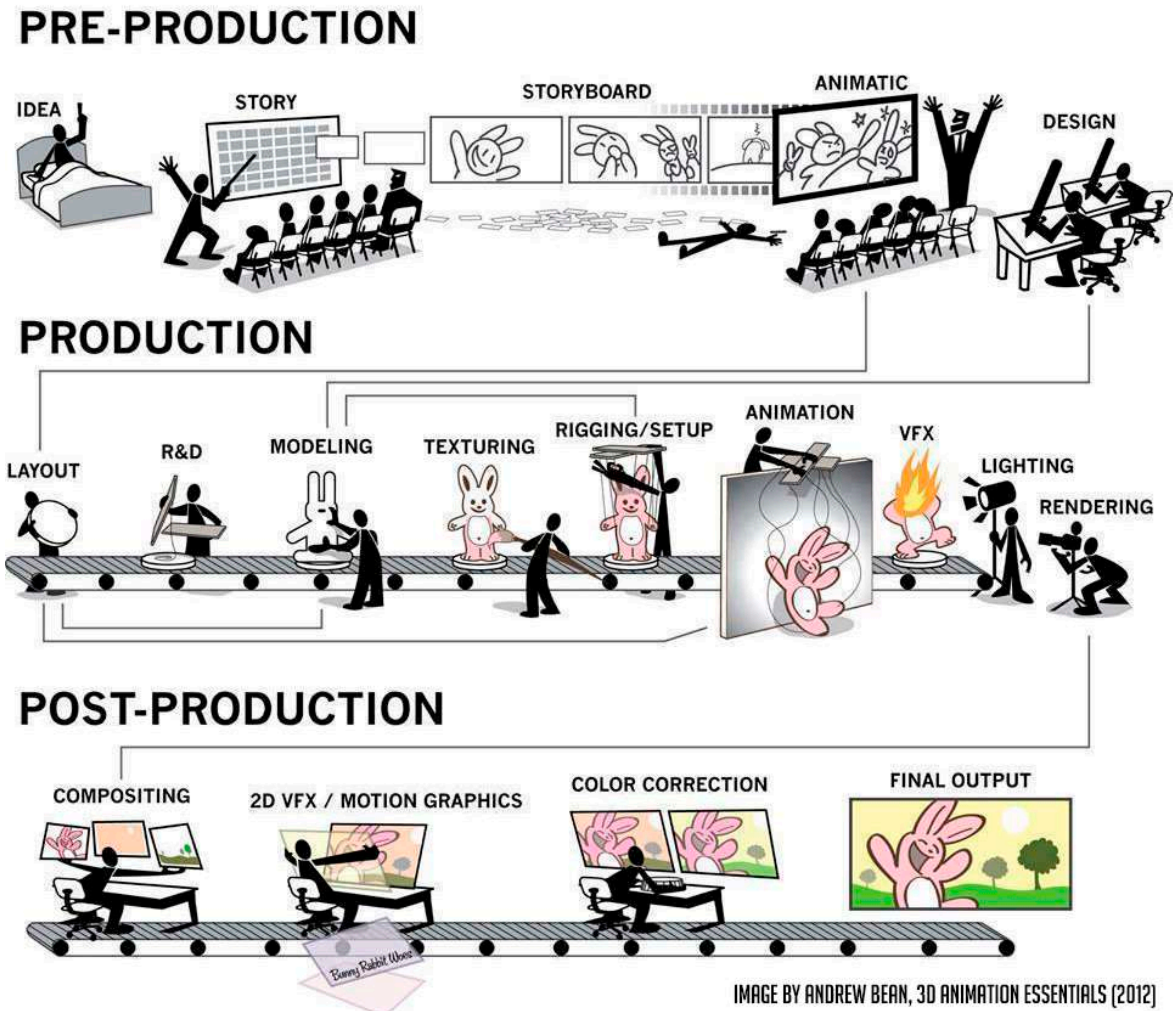


IMAGE BY ANDREW BEAN, 3D ANIMATION ESSENTIALS (2012)

Next (Final) Lecture

Given the forces / physics / theory, how to simulate actual movements

Hint: what would he say in a fight?

Credit: JoJo's Bizarre Adventure

Thank you!

(And thank Prof. Ren Ng for many of the slides!)