Advanced Animation Techniques

Modern Animation

Example: Player package from Death Stranding

https://www.youtube.com/watch?v=55W5ZKNrpz4?t=152

Modern Animation Concerns

- Motion capture
- Easy and flexible animation controls
- Automatic retargeting
- Handling soft bodies
- Physically based animations

Motion Capture

g-actors-logan-never-knew-149013.html

https://youtu.be/BH58puh-Olg?t=76

Motion Capture Overview

- Detect and track markers associated with joints
  - Variety of techniques and equipment
  - Closely related to computer vision
- Markers provide data about changes to position and rotation
- Data cleaned then used for retargeting meshes

Automatic Retargeting

Practice of taking existing skeleton and remapping it to a new mesh to apply animations

Requires both meshes to share the same skeleton asset

- Same structure, different proportions
- Skeleton joints translated to correct position on the mesh

Animation handles joint rotations during sequence

Animation Controls

Articulated Models

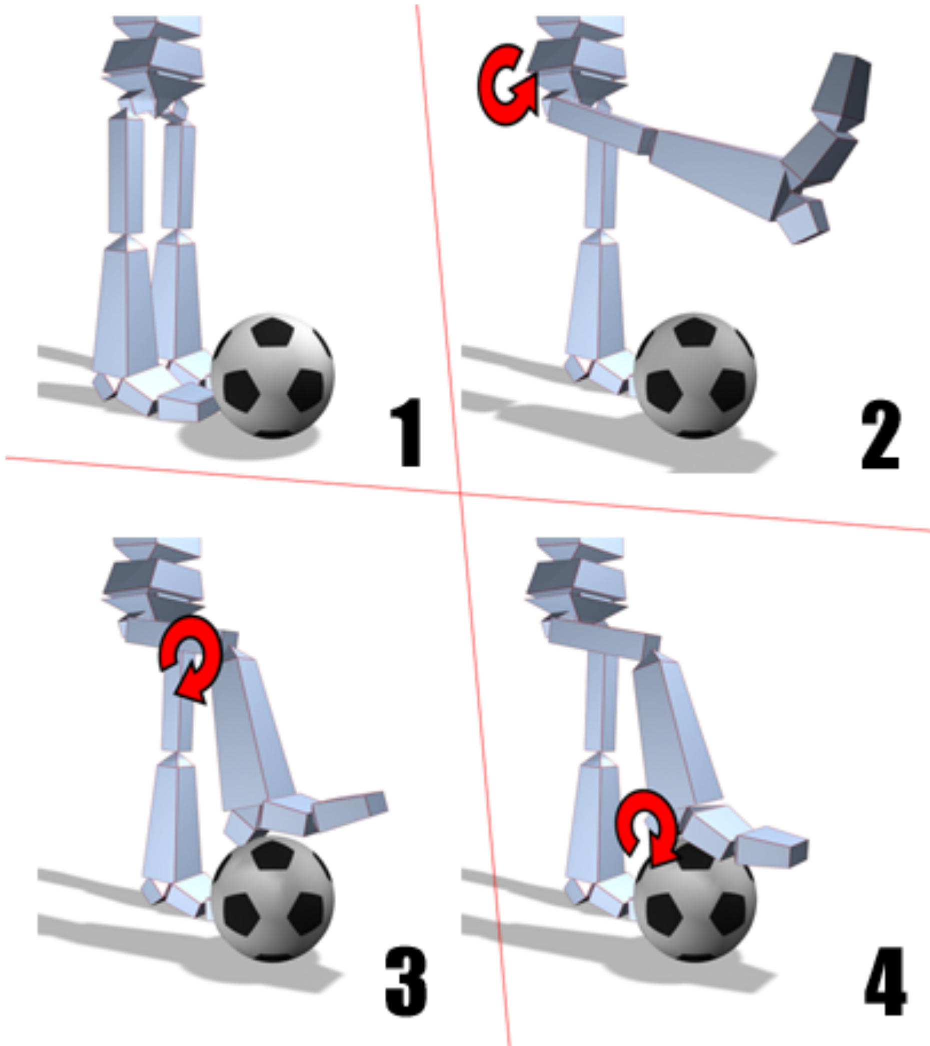
The previous examples are all **articulated models**:

- Rigid parts (bones)
- Connected by joints

They can be animated by specifying the joint angles (or other display parameters) as functions of time.

- Direct control of joints
- Inverse kinematics (IK)

Forward Kinematics



Inverse Kinematics

Inverse kinematics take the **target pose** and compute all necessary joints along the chain to reach that pose

Forward kinematics are easier to compute but harder to reason about

Inverse kinematics are more natural to reason about but harder to compute

Defining Target Poses

Skeleton must include **kinematic chains** along parts of hierarchy with joint dependencies

- e.g. Dependency extends from shoulder to wrist

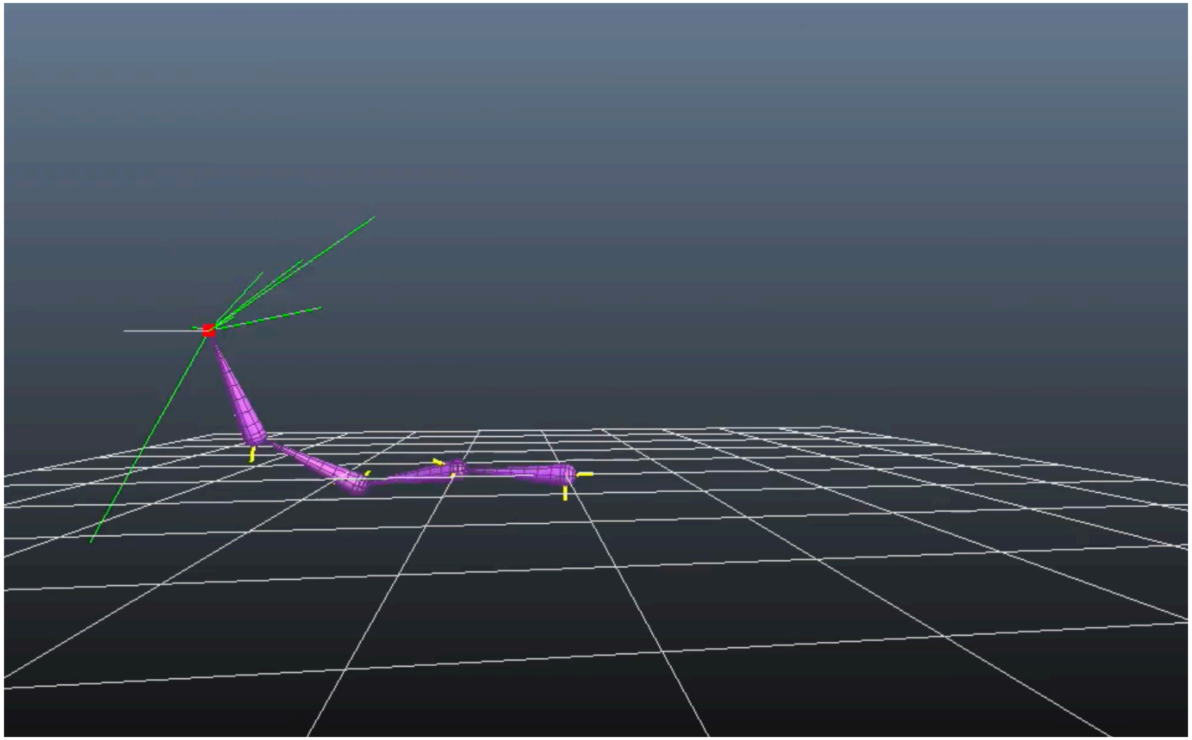
Each kinematic chain has an **end effector** to target different positions (or orientations)

- New position of end effector updates all joints along kinematic chain

May be possible to solve analytically but usually solved approximately and iteratively

- Minimization problem: get end effector as close to target position as possible

Inverse Kinematics Example



https://vimeo.com/114626019

Jacobian Technique

A Jacobian matrix is a matrix of first-order partial derivatives of system

- Describes changes in end effector position based on changes in joint angle

Jacobian inverse allows computation of joint angles from changes in end effectors

Goal is to perform small, iterative changes to joint angles using Jacobian to reach target position

Must calculate Jacobian

- Numerically or analytically

Must approximate inverse

- Pseudo-inverse or transpose

Cyclic Coordinate Descent

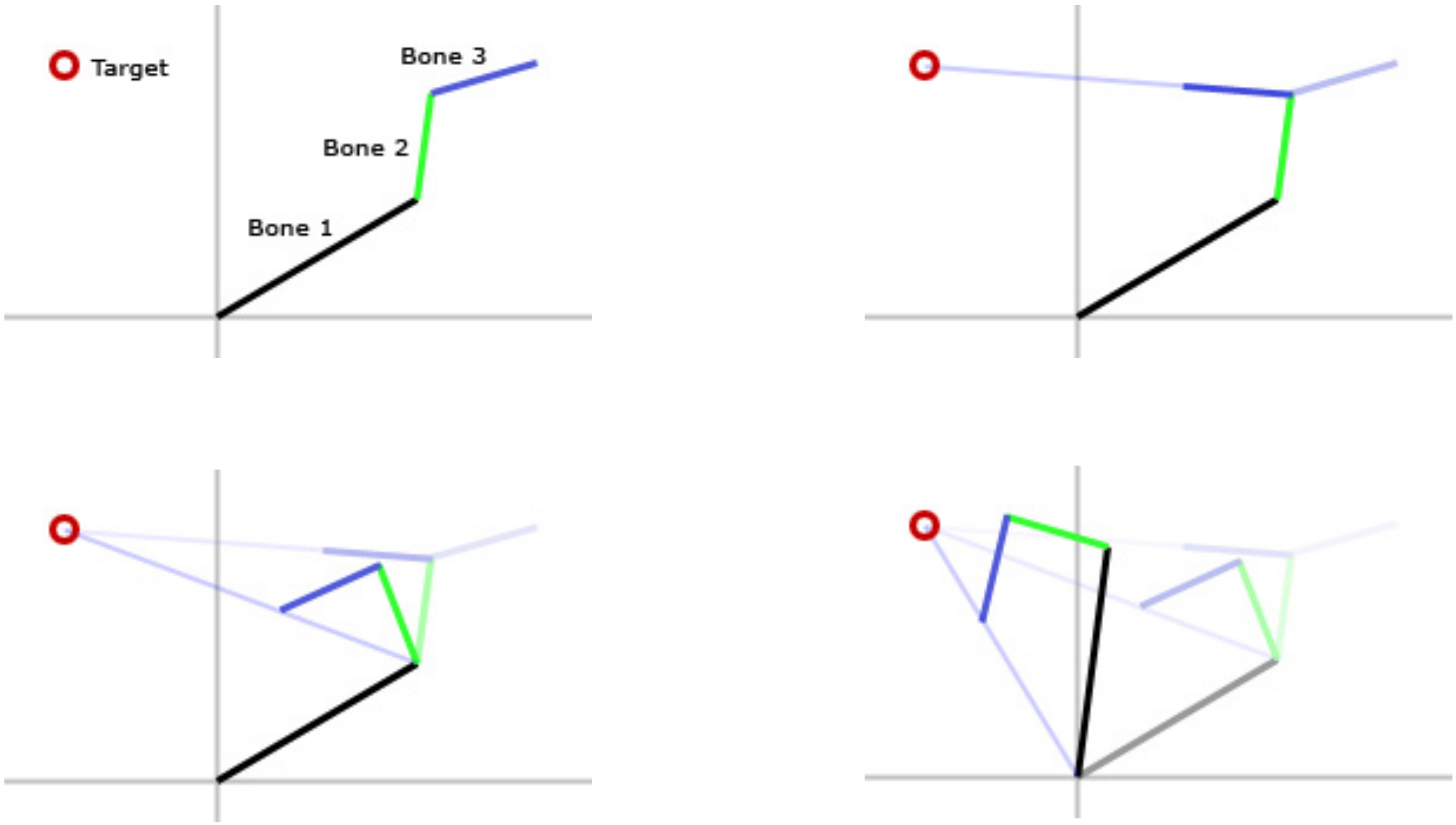
Iterative optimization algorithm used to reduce error or find minimum of function

- Minimize distance between end effector and target position

Individually adjusts joint angles starting at last link and working backward

Determine joint-to-end-effector vector and target-to-end effector vectors

- Can determine angle between them (or amount to rotate) using dot product
- Can determine direction to rotate using cross product


Target
Bone 3
Bone 2
Bone 1

Local Minima

Issue with all optimization algorithms

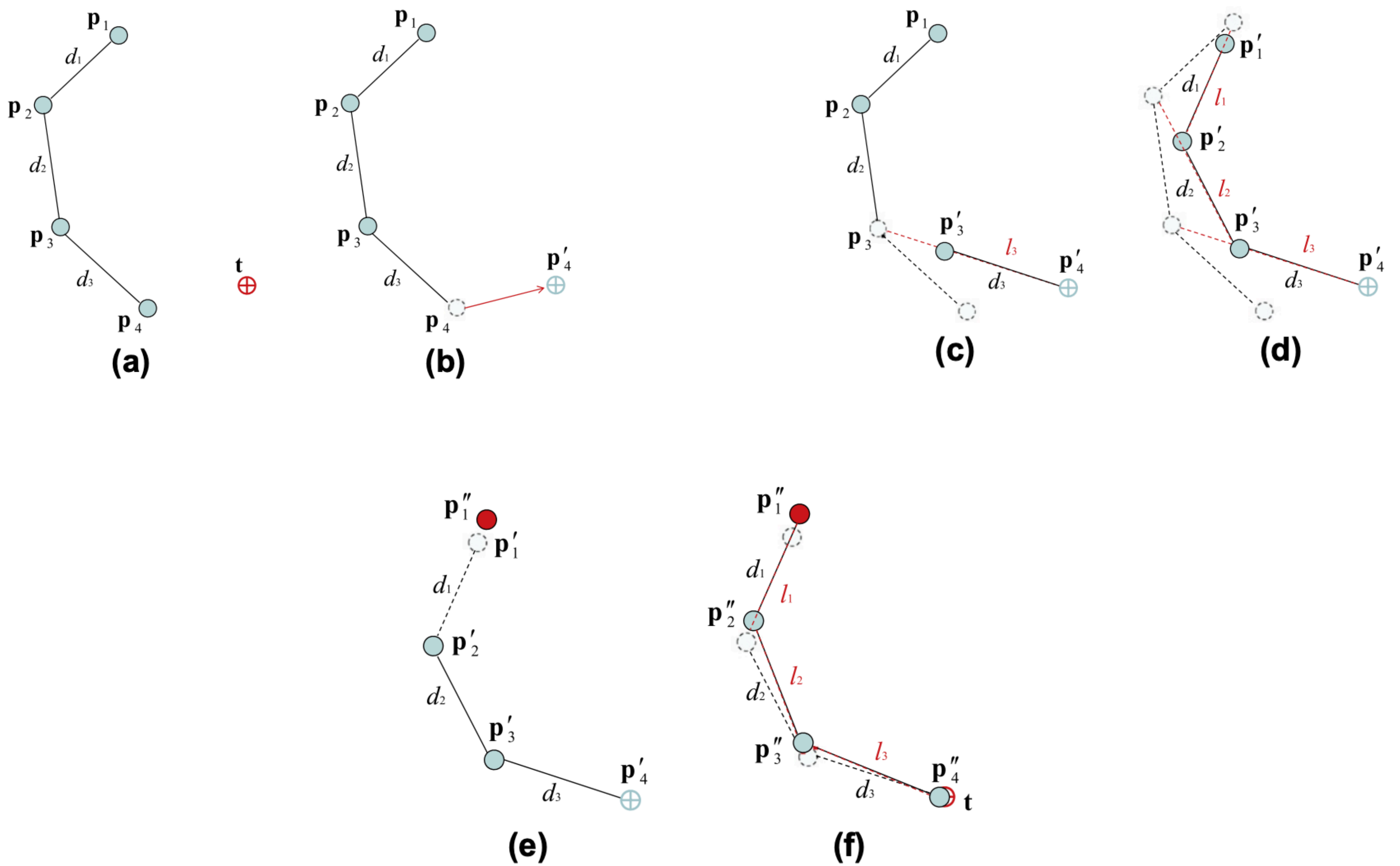
Solution finds a local (but incorrect) minima and is unable to continue descent toward global minima

In CCD, bones do not consider other bones — only distance to optimal position

- Leads to “tangling” that places bones in optimal positions but does not respect chain
- Placing rotational constraints on joints reduces these errors but cannot fully correct for it

FABRIK

- Forward and Backward Reaching Inverse Kinematics
- Iterative method to find joint position on a line rather than considering joint angles
- Determine if target is reachable (i.e. distance from root to target is less than total length of chain)
- Move end effector to target position (within tolerance) and recalculate positions of previous joints
- Move root joint back to its initial position and recalculate positions of all child joints



p1
d1
p2
d2
p3
d3
p4
t
(a)
p1
d1
p2
d2
p3
d3
p4
p′4
(b)
p1
d1
p2
d2
p3
p′3
l3
d3
p′4
(c)
p′1
d1
l1
p′2
d2
l2
p′3
l3
d3
p′4
(d)
p″1
p′1
d1
p′2
d2
p′3
d3
p′4
(e)
p″1
d1
l1
p″2
l2
d2
p″3
l3
d3
p″4
t
(f)

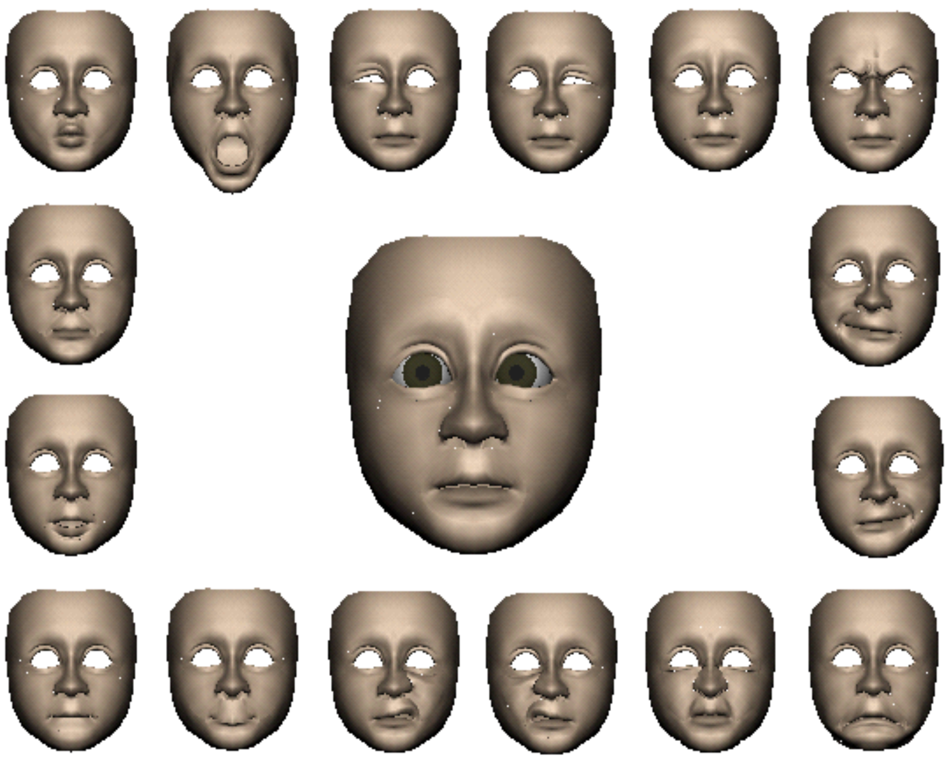
IK in Games

Emphasis on stability and speed

- Inverse Jacobian common in film but too slow for games
- CCD has unwanted pathologies
- FABRIK is efficient, simple to implement, and looks good so preferred in games
  - Can also apply FABRIK with joint constraints and to hierarchies with multiple end effectors

Other 3D Animation Techniques

Faces are hard to rig: many muscles

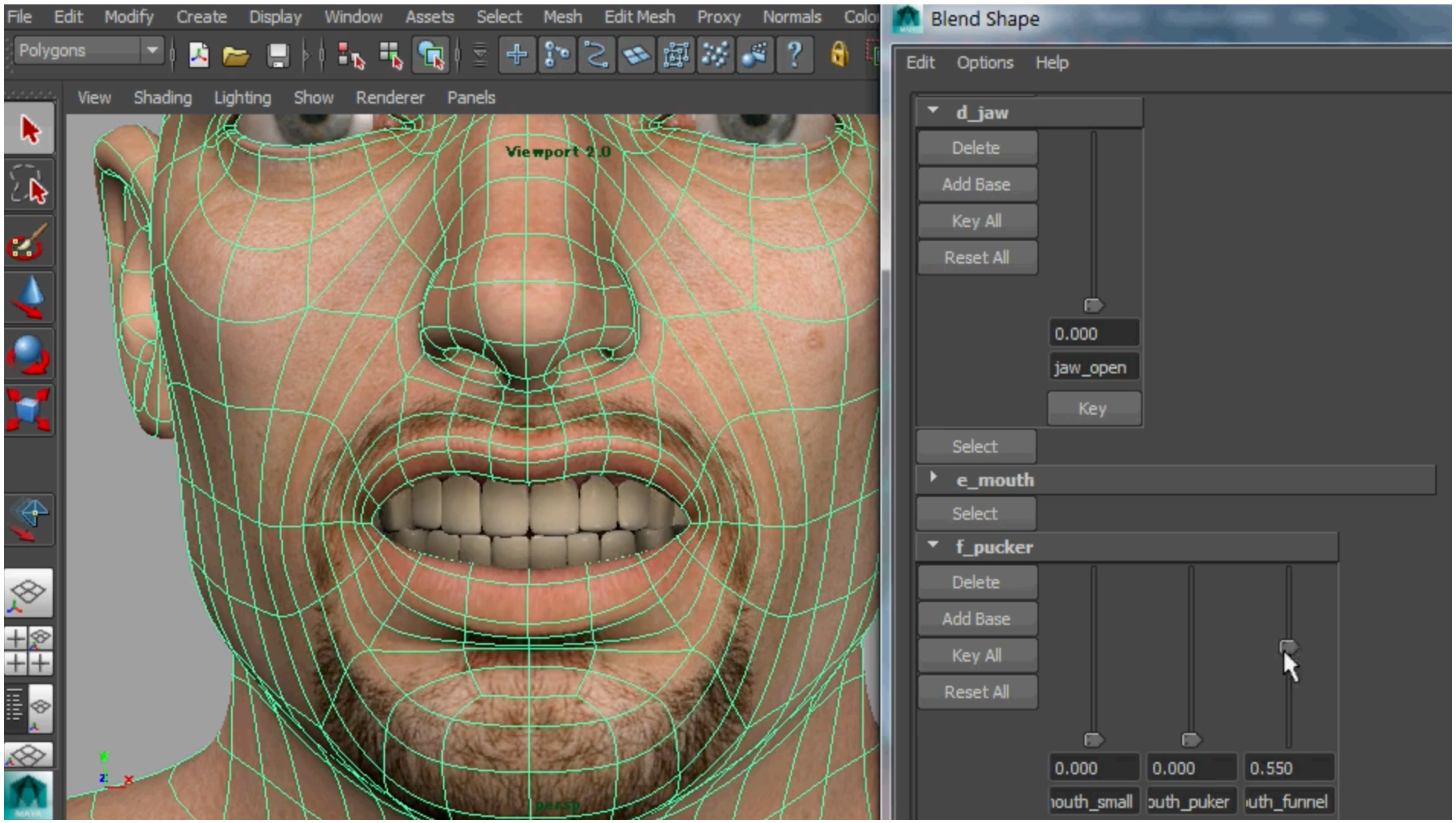


instead, precompute a small set of **basis deformations**

blend between them (**blendshapes**)

knobs not always intuitive

Blend Shape Demo



https://vimeo.com/115836284

Soft Bodies

Treats objects as deformable

- Shape of object can change

Accurate simulation is more computationally intensive

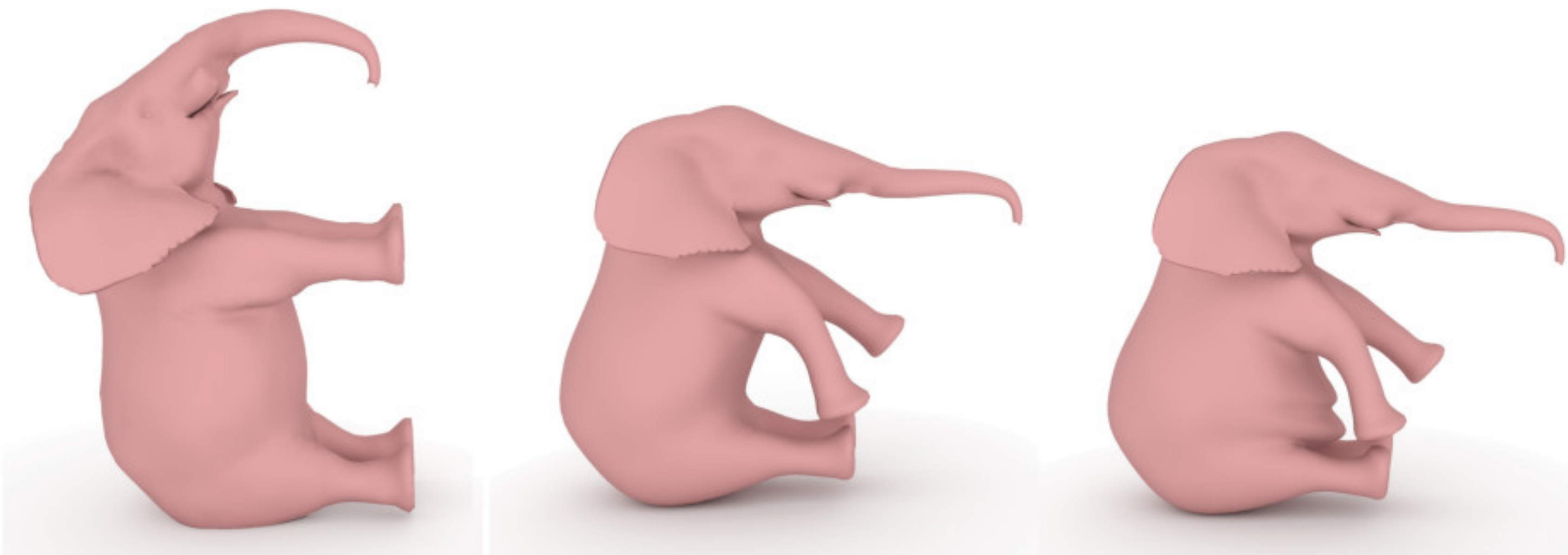
- **Finite element simulation** breaks system into smaller, solvable subdomains that can be reassembled

Mostly faked in games

- Rigid bodies in a lattice can fake slime, cloth, etc

Soft Body Uses

Volume Conservation



Deformation and Fracture in Games

https://www.youtube.com/watch?v=ly64-Bn7i4k

Other 3D Animation Techniques

Anatomy-based animation

- simulate the tendons and muscles

Most correct motion

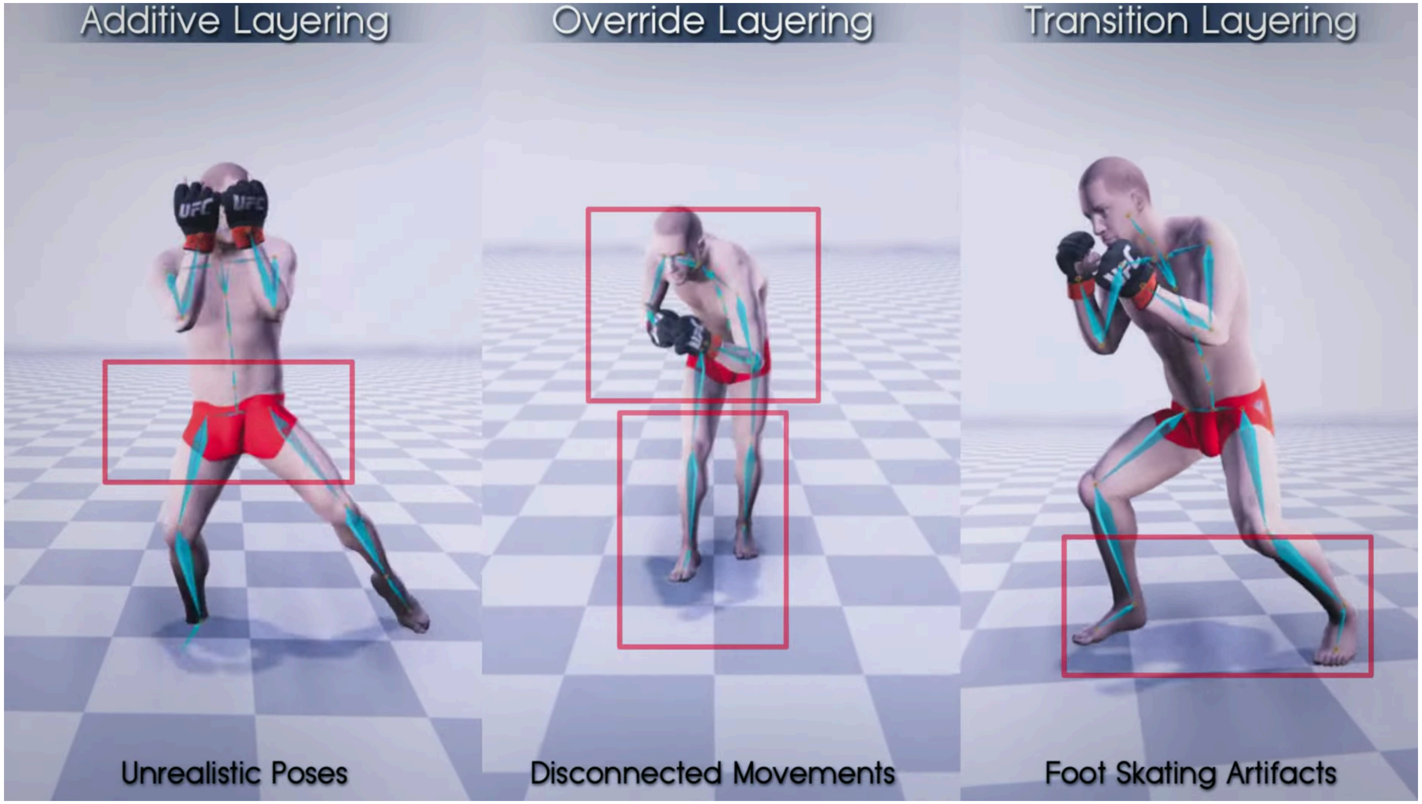
Restricted to “real” animals

Slow

Machine Learning and Motion Synthesis

- Machine learning allows for good-looking animations
  - https://www.youtube.com/watch?v=urf-AAIwNYk
  - Can lower cost of artistic pipeline
- Downside is that learned function is mostly unmodifiable/incomprehensible
  - Difficult to correct in pathological cases
- Also realism is not necessarily as appealing as artistic interpretation

ML in Modern Games



https://www.youtube.com/watch?v=SkJNxLYNwN0