TEL AVIV UNIVERSITY

3D Animation

0368-3236, Spring 2021

Tel-Aviv University

Guy Tevet

Classic Animation

Moving image:

$$I \in \mathbb{R}^{H \times W \times C} \longrightarrow [I_1, \ldots, I_T] \in \mathbb{R}^{H \times W \times C \times T}$$

Disney's Snow White (1937)

3D Animation

CG model:

$S \in \mathbb{R}^d \longrightarrow [S_1, \ldots, S_T] \in \mathbb{R}^{d \times T}$

3D Mesh

3D Primitives

Light

Camera

Toy Story (1995)

Applications

- Animated films

Inside Out (2015)

Applications

- Animated films
- Gaming

Fortnite

Applications

- Animated films
- Gaming
- Visual Effects

Game of Thrones

Applications

- Animated films
- Gaming
- Visual Effects
- Modeling
  - Mechanical engineering
  - Medical
  - Physical simulation

4:32

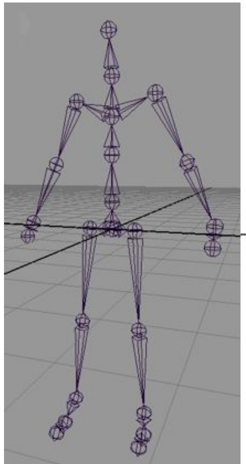
Today

- Model
  - Skeleton Rigging
  - Blendshapes

- Animation
  - Key-Frames
  - Motion capture

Skeleton Rigging

Input: 3D mesh

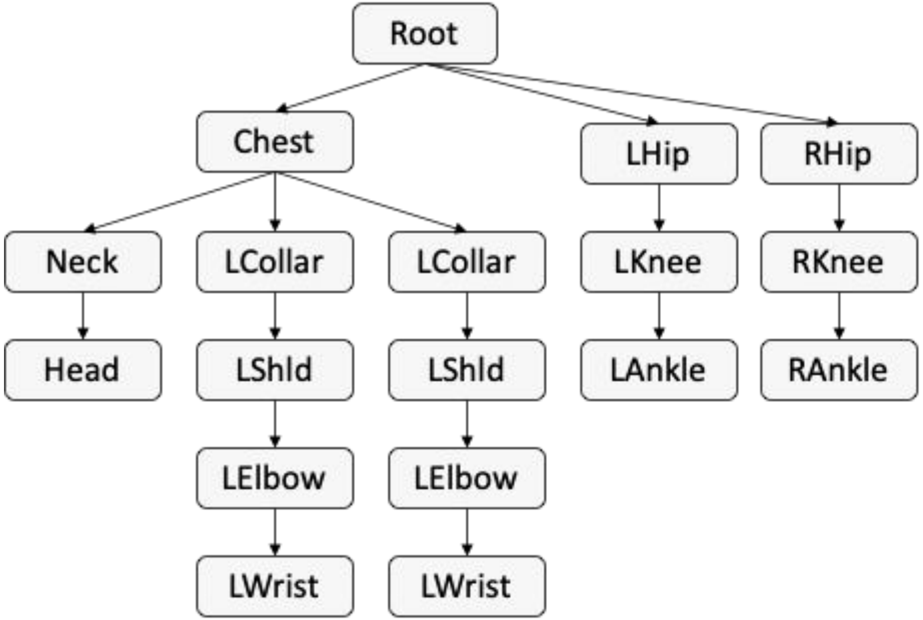


1. Define a skeleton

2. Skinning

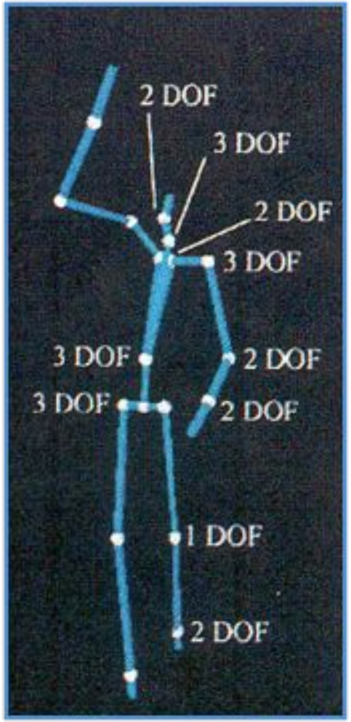
Output: 3D *rigged* mesh

Define a skeleton



Zoom in

- Bone length
- Angle

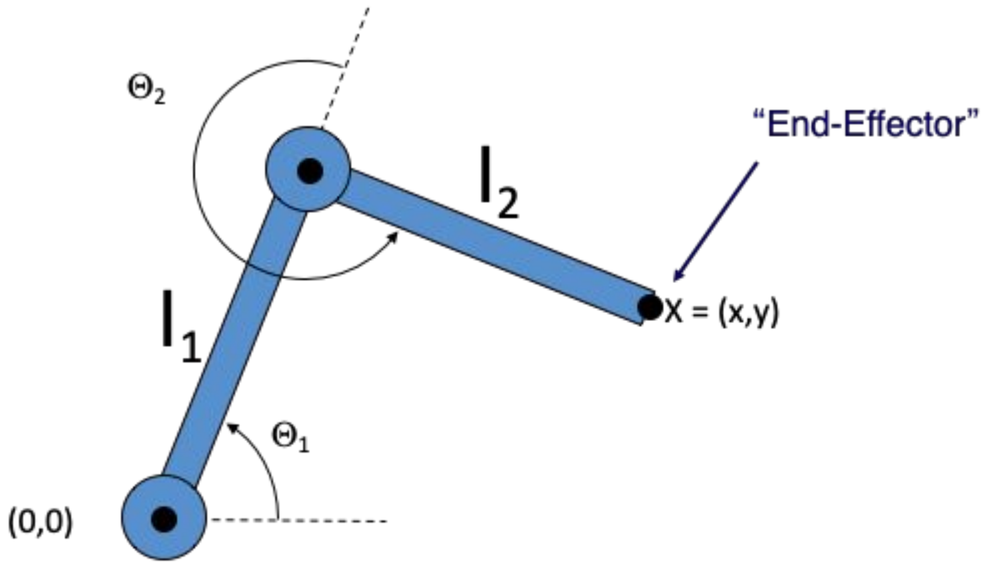


Rose et al. `96

Forward Kinematics

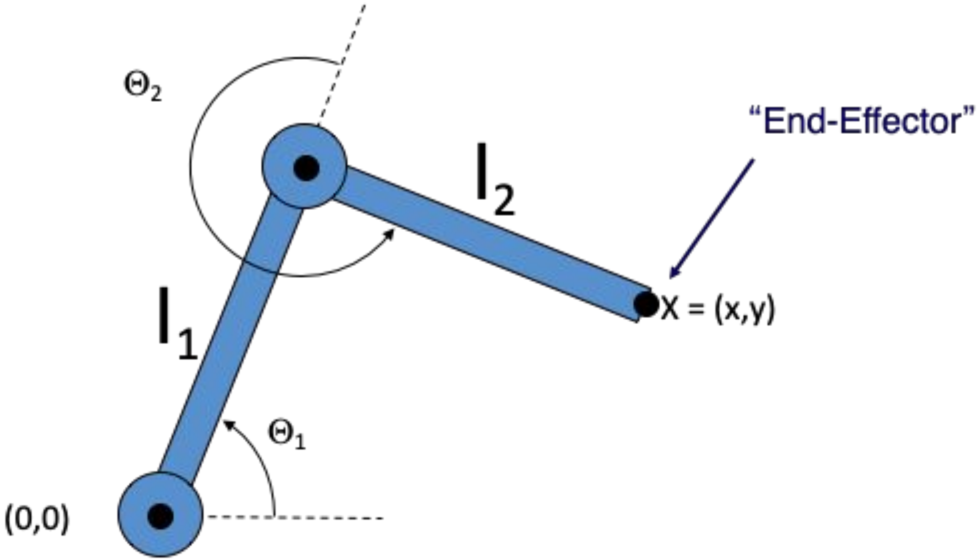
Toy example - robotic arm

- Animator specifies joint angles: Θ_1 and Θ_2
- Computer finds positions of end-effector: X



Forward Kinematics

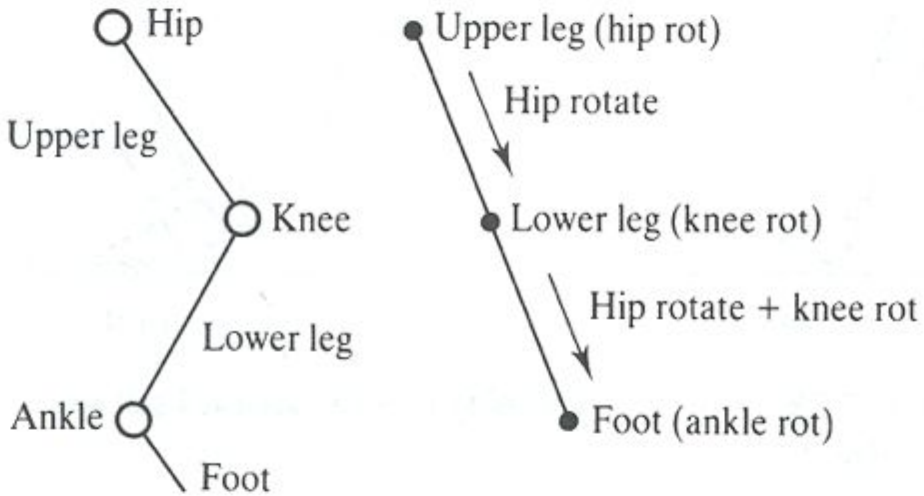
Toy example - robotic arm



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

Forward Kinematics

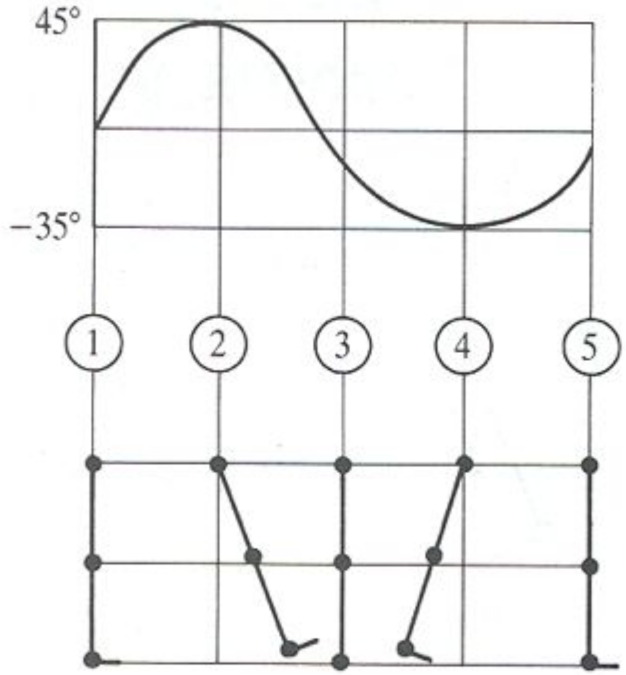
Human leg - walking cycle



Forward Kinematics

Human leg - walking cycle

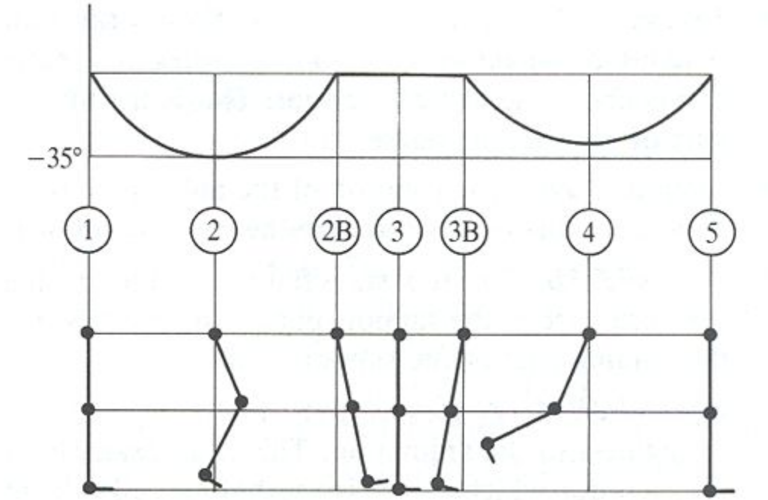
- Hip joint orientation



Forward Kinematics

Human leg - walking cycle

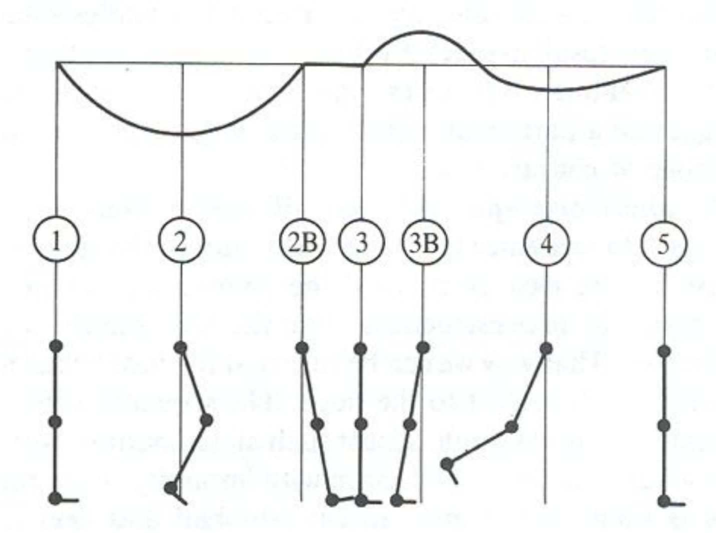
- Knee joint orientation



Forward Kinematics

Human leg - walking cycle

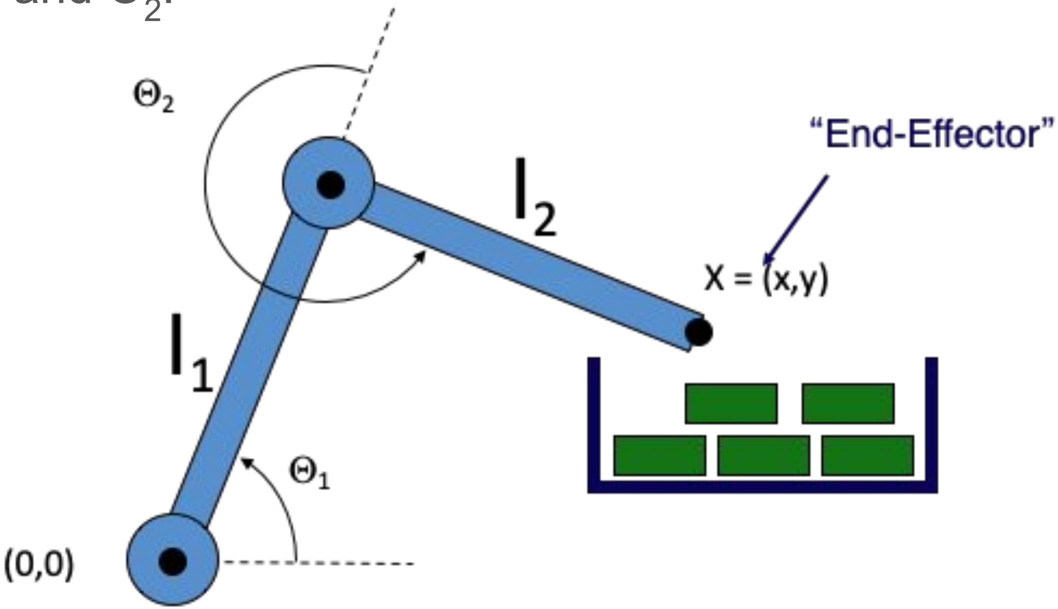
- Ankle joint orientation



Inverse Kinematics

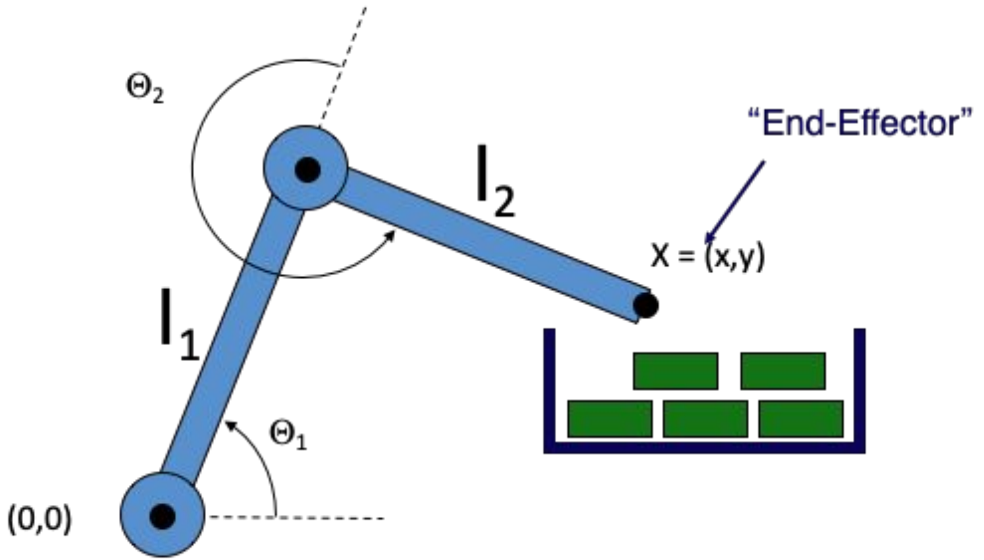
Toy example - robotic arm

- Animator specifies end-effector positions: X
- Computer finds joint angles: Θ_1 and Θ_2:



Inverse Kinematics

Toy example - robotic arm



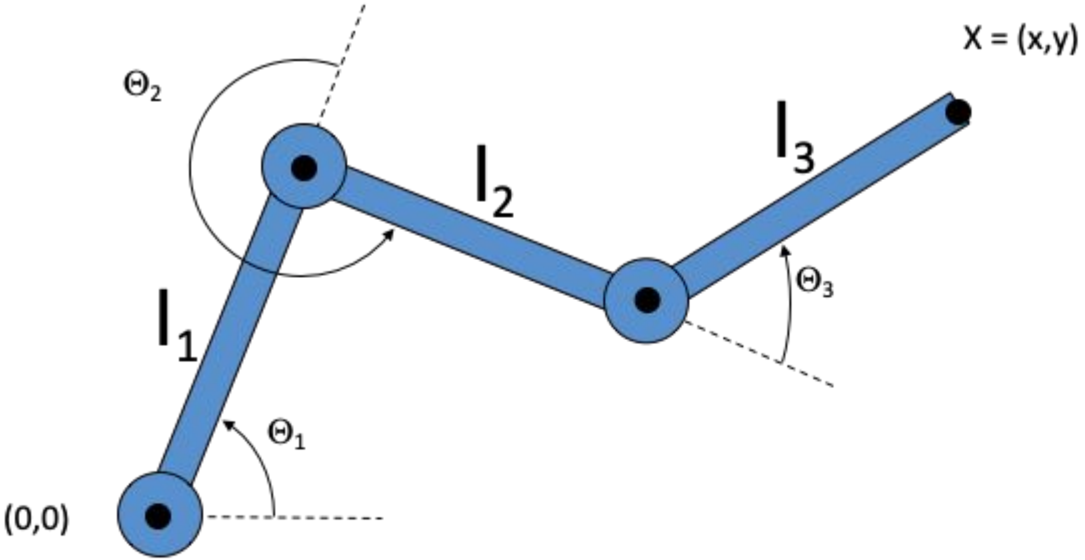
Two unknowns: Θ_1, Θ_2
Two equations: x, y

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

Inverse Kinematics

Toy example - robotic arm

Three unknowns: $\Theta_1, \Theta_2, \Theta_3$
Two equations: x, y



Problem:

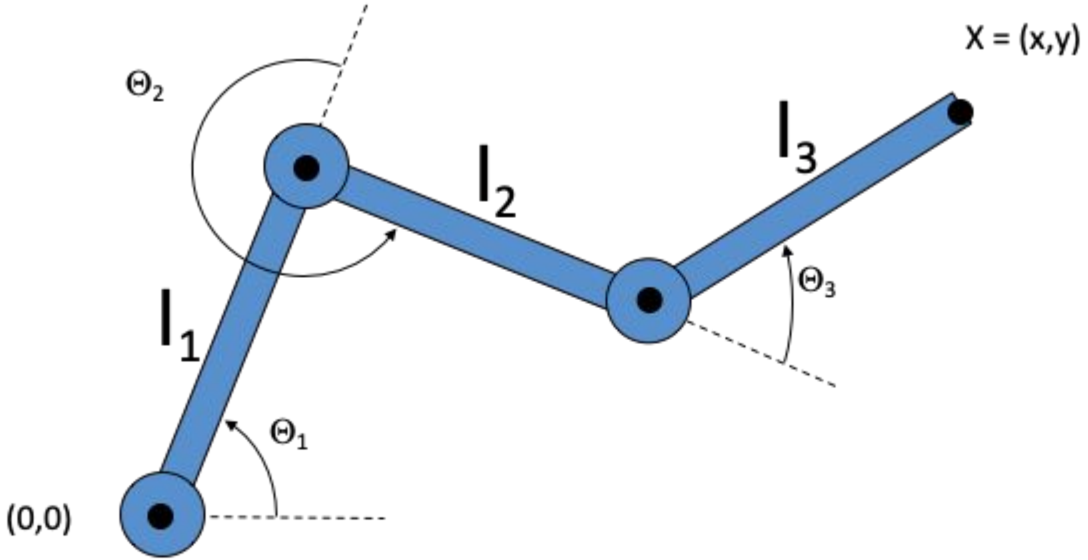
- System of equations is usually under-constrained
- Multiple solutions

Inverse Kinematics

Toy example - robotic arm

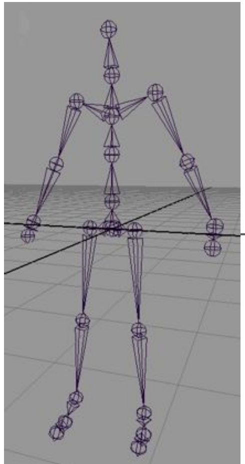
Solution:

- Find best solution (e.g., minimize energy in motion)



Zooming out - Skeleton Rigging

Input: 3D mesh



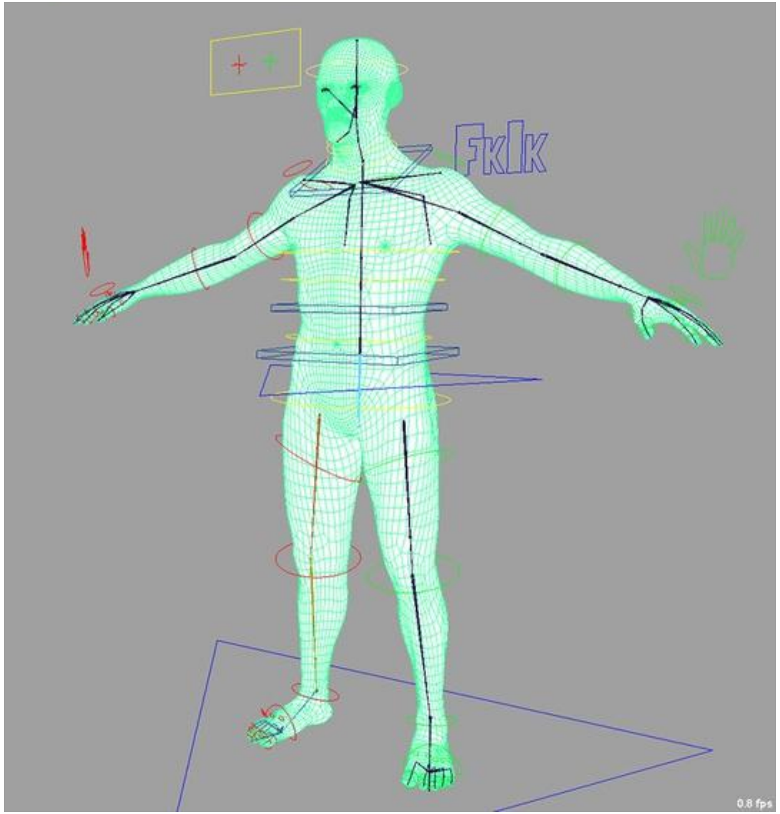
1. Define a skeleton

2. Skinning

Output: 3D *rigged* mesh

Skinning

- Deforming the mesh according to skeleton movement



Linear Blend Skinning

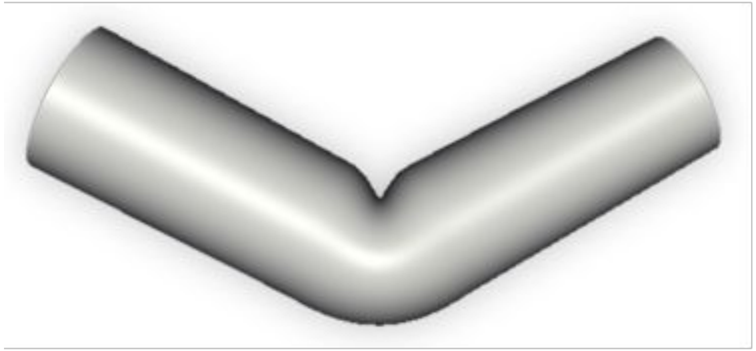
- Each vertex of skin potentially influenced by all bones
  - Normalized weight vector $w^{(v)}$ gives influence of each bone transform
  - When bones move, influenced vertices also move
- Computing a transformation T_v for a skinned vertex
  - For each bone
    - Compute global bone transformation T_b from transformation hierarchy
  - For each vertex
    - Take a linear combination of bone transforms
    - Apply transformation to vertex in original pose

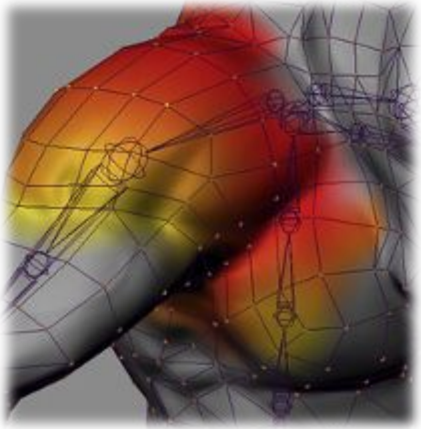
$$T_v = \sum_{b \in B} w_b^{(v)} T_b \longrightarrow v_{trans} = \sum_{b \in B} w_b^{(v)} (T_b v)$$

Linear Blend Skinning

$$v_{trans} = \sum_{b \in B} w_b^{(v)} (T_b v)$$

Smoothness of skinned surface depends on smoothness of weights!





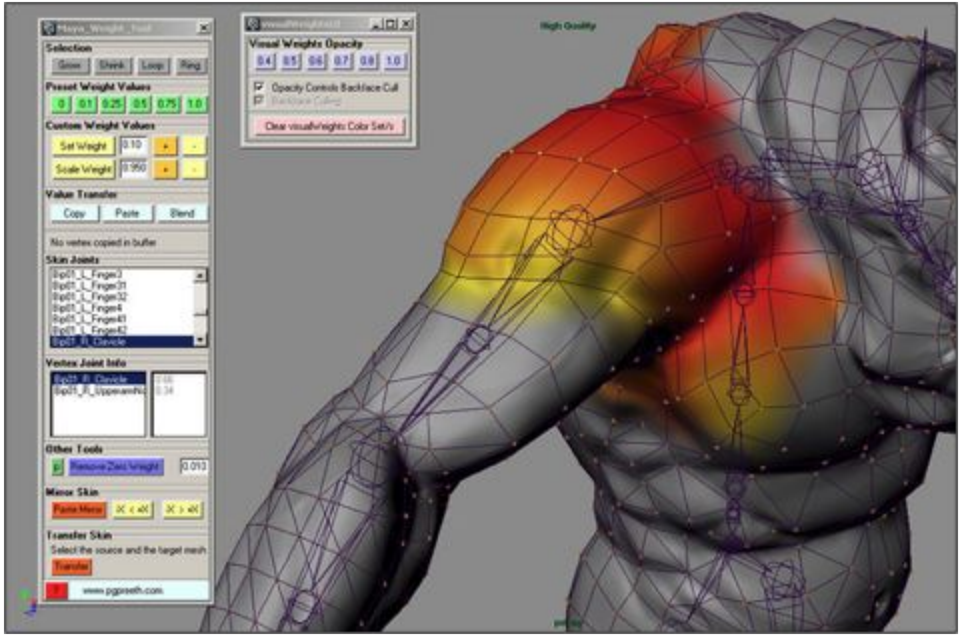
Define the blend

Weights matrix:

$$W \in \mathbb{R}^{|B| \cdot |V|}$$

Define the weights:

- Painted by hand



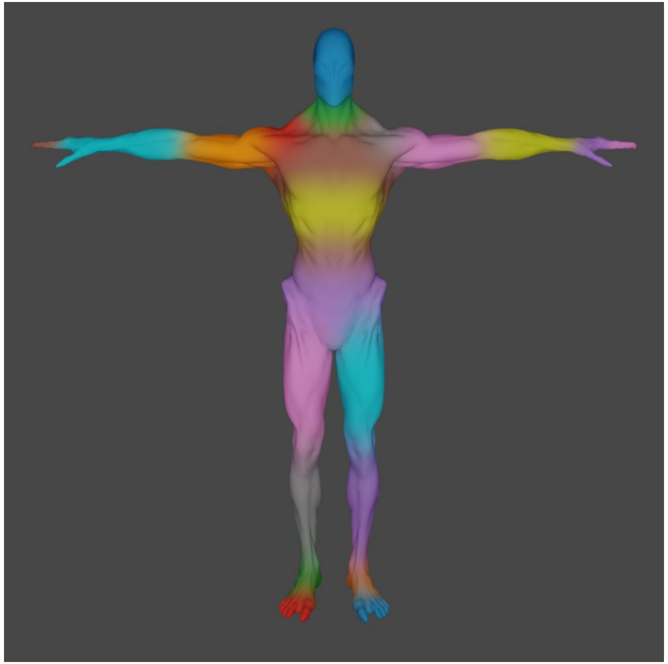
Define the blend

Weights matrix:

$$W \in \mathbb{R}^{|B| \cdot |V|}$$

Define the weights:

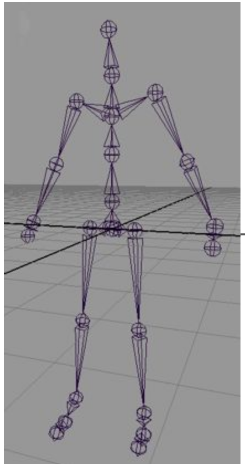
- Painted by hand
- Automatic
  - Relative distances to nearest bones
  - Machine learning based



Li et al. 2021

Zooming out - Skeleton Rigging

Input: 3D mesh



1. Define a skeleton

2. Skinning

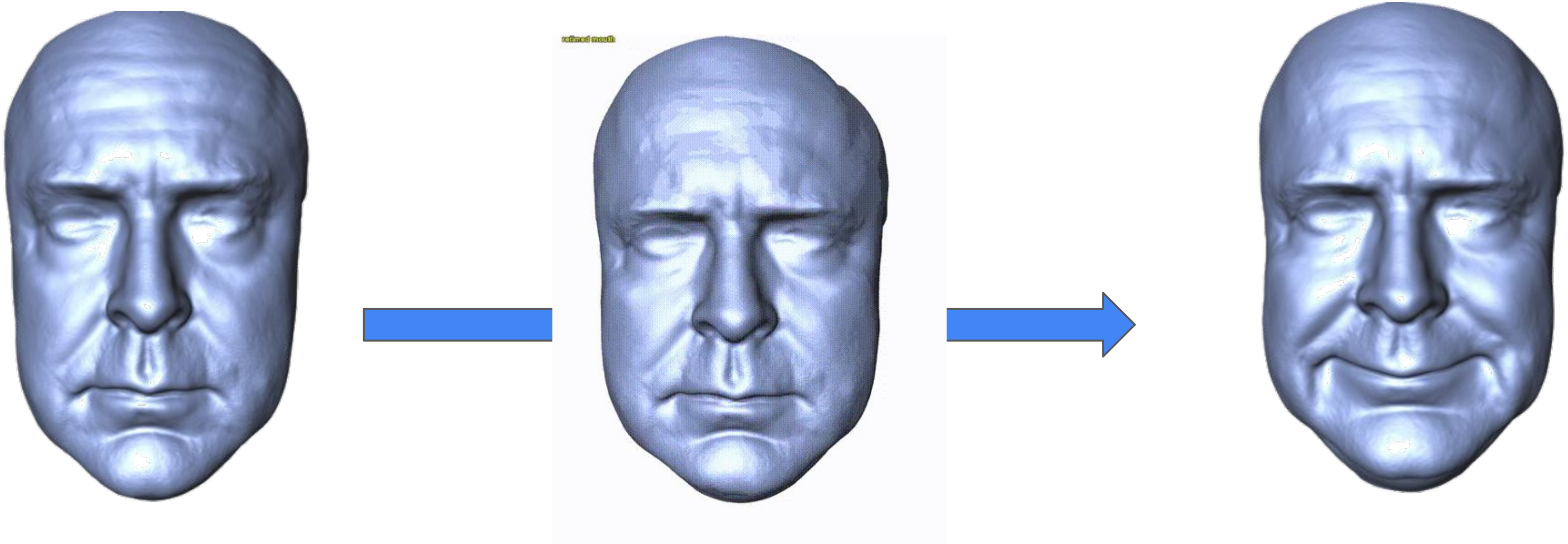
Output: 3D *rigged* mesh

Zooming out - Today

- Model
  - ✓ Skeleton Rigging
  - ➡ Blendshapes
- Animation
  - Key-Frames
  - Motion capture

Facial Blendshapes

How to apply motion on human face?



V_0

Linear interpolation:

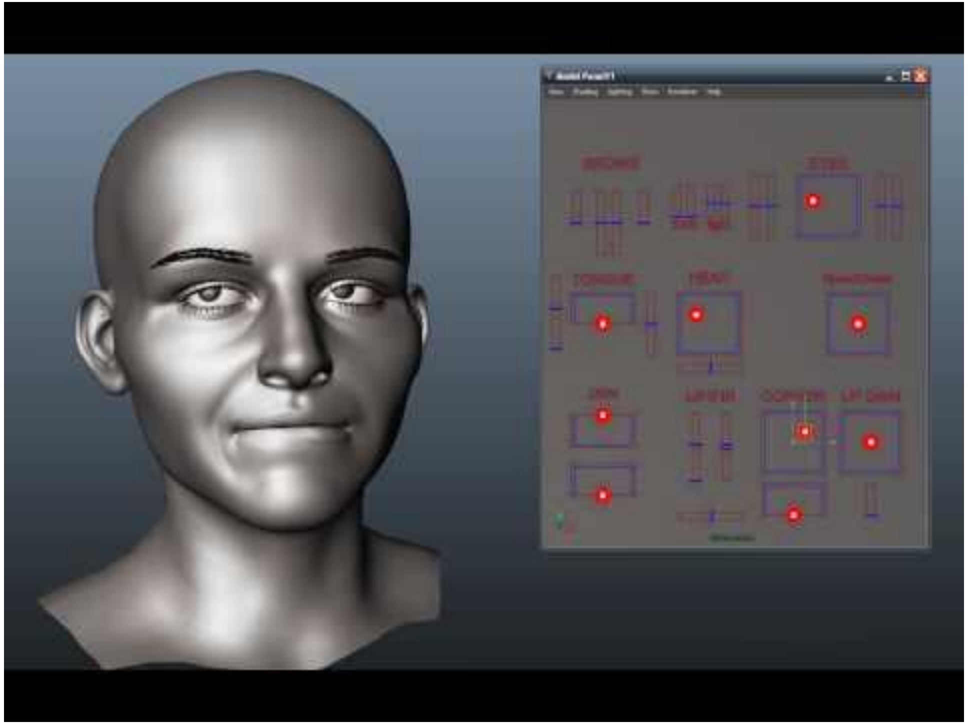
$$v(t) = \left(1 - \frac{t}{T}\right) v_0 + \frac{t}{T} v_T$$

V_T

Facial Blendshapes

- Generalize:
  - Span all facial expressions with linear combinations

Facial Blendshapes



Zooming out - Today

- Model
  - ✓ Skeleton Rigging
  - ✓ Blendshapes
- Animation
  - Key-Frames
  - Motion capture

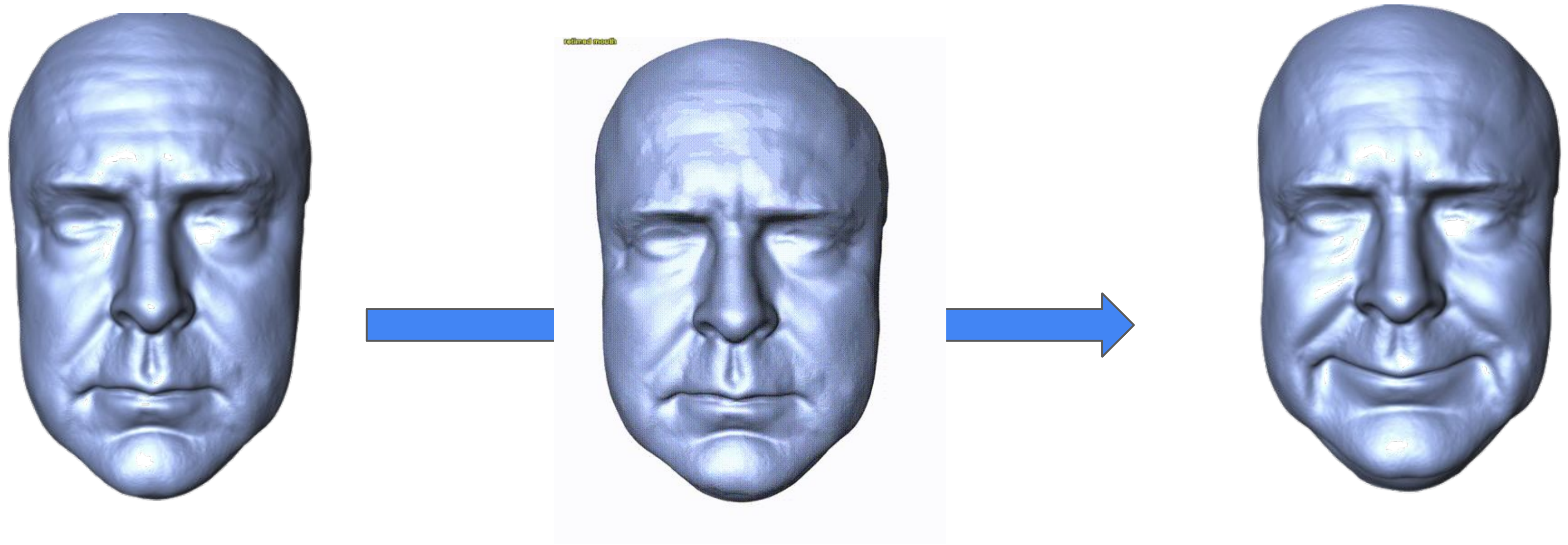
Result - full body rig

Zooming out - Today

- Model
  - Skeleton Rigging
  - Blendshapes
- Animation
  - Key-Frames
  - Motion capture

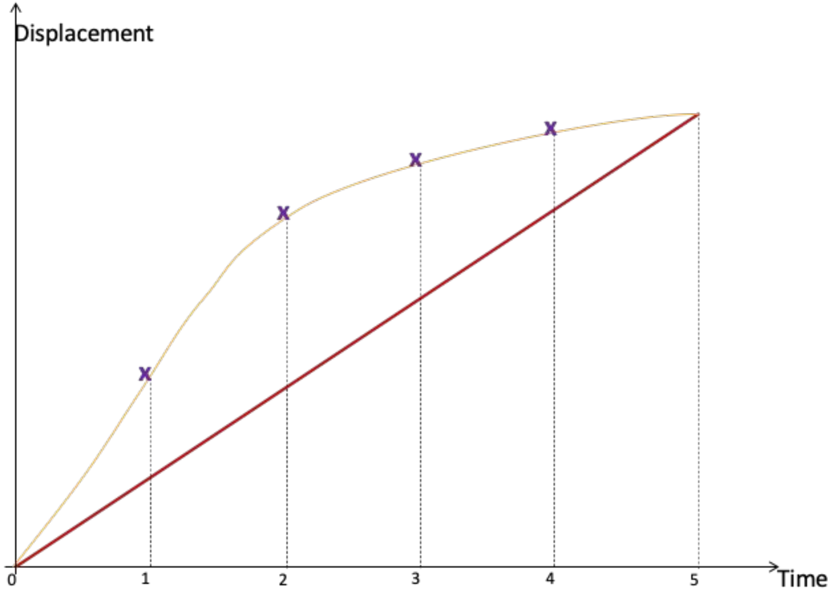
Key-framing animation

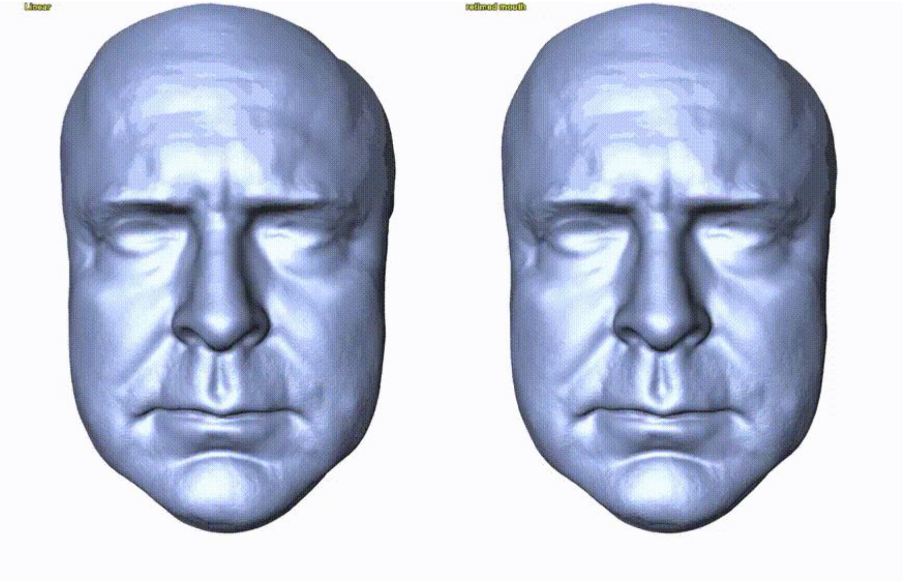
1. Define key-frames
2. Interpolate rig between frames



In-betweening methods

- Interpolation
  - Linear vs. non-linear



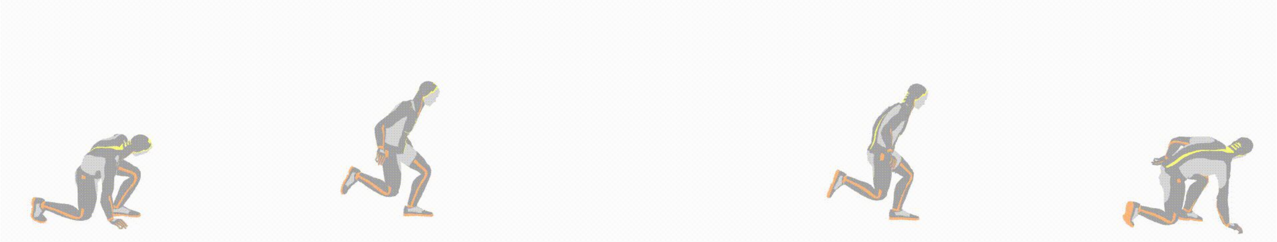


Linear

Non-inear

In-betweening methods

- Interpolation
  - Linear vs. non-linear
- Machine learning based



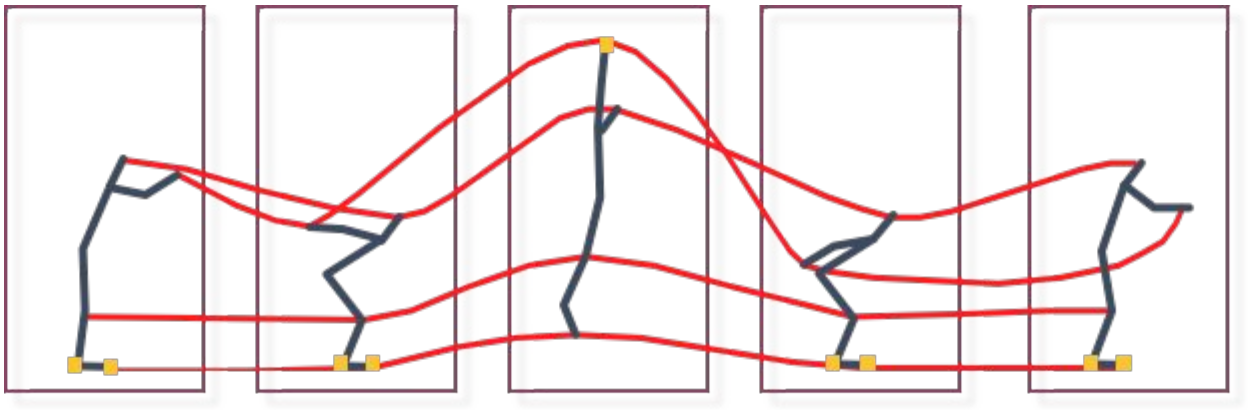
Harvey et al. 2020

Zooming out - Today

- Model
  - ✓ Skeleton Rigging
  - ✓ Blendshapes
- Animation
  - ✓ Key-Frames
  - ➡ Motion capture

Motion capture

- Record motion of real character
- Then “play it back” with kinematics



Captured Motion

Motion capture

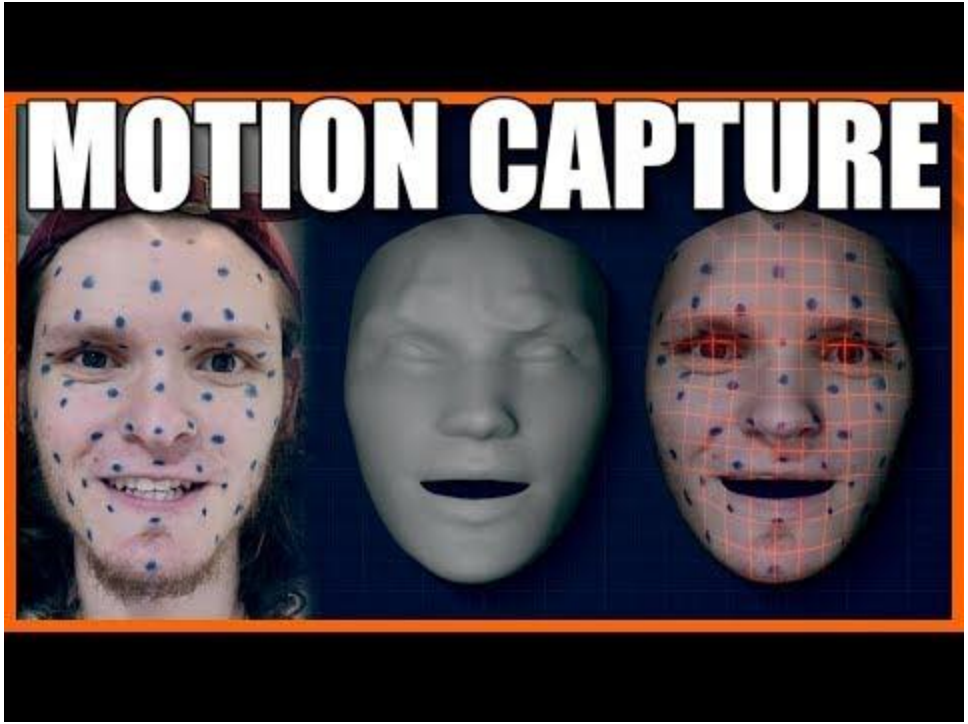
Motion capture systems:

- Wearable motion sensors

Motion capture

Motion capture systems:

- Wearable motion sensors
- Visual (cameras + body markers)

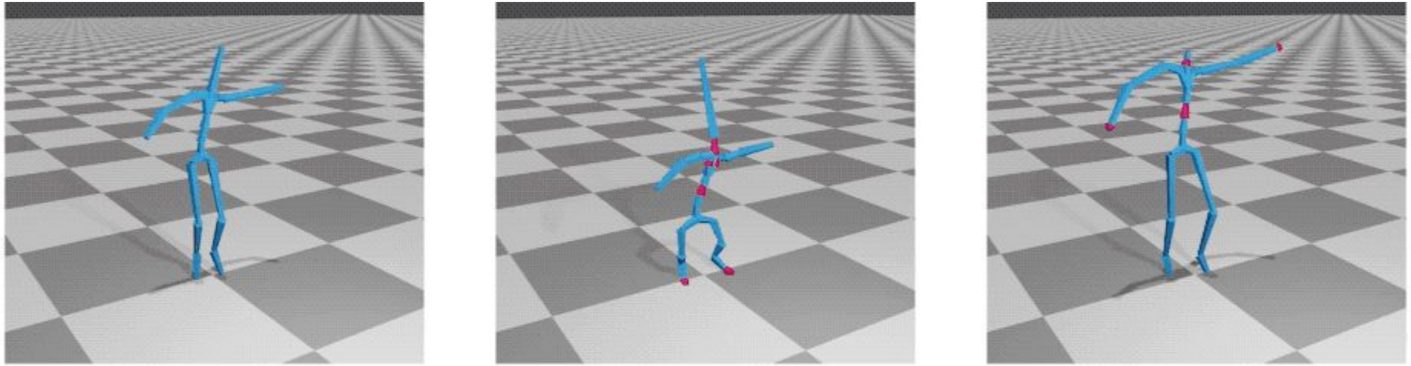


Motion retargeting

- **Problem:** Animate Shrek with a professional dancer moves
- **Solutions:**
  - Artistic corrections (e.g. inverse kinematics)

Motion retargeting

- **Problem:** Animate Shrek with a professional dancer moves
- **Solutions:**
    - Artistic corrections (e.g. inverse kinematics)
    - Machine learning based



Aberman et al. 2020

Zooming out - Today

- Model
  - Skeleton Rigging
  - Blendshapes
- Animation
  - Key-Frames
  - Motion capture

Not Today - Physical simulation

Thanks!