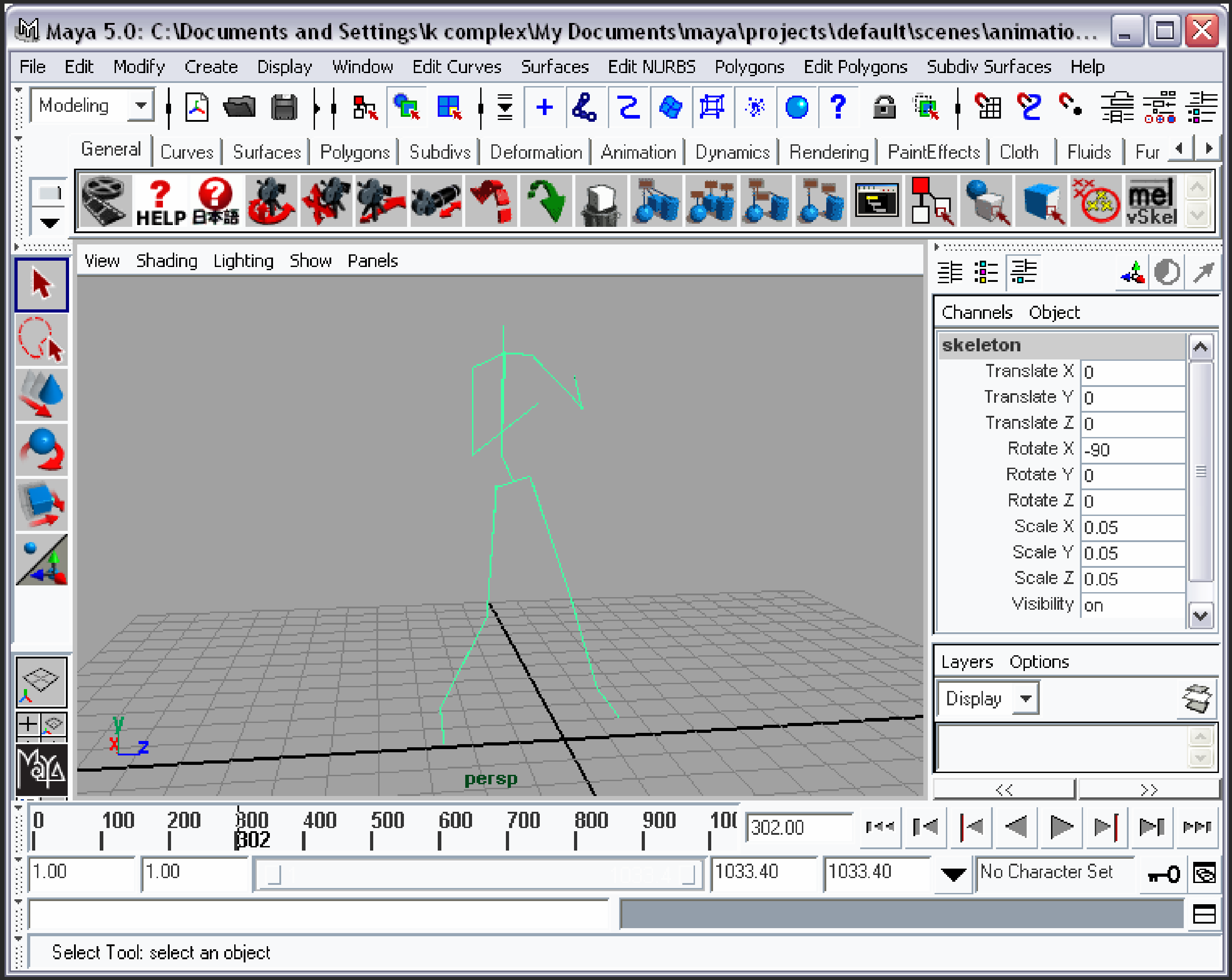
Machine Learning for Computer Graphics

Aaron Hertzmann
University of Toronto
CIAR summer school
July 16, 2005

CG is maturing …

… but it's still hard to create



… it's hard to create in real-time

Petersen, dir: *The Perfect Storm*, 2000

One shot of *The Perfect Storm*

1. "Ocean scout" looks through 1000-frame simulation, puts camera in position (days)
2. Animator refines camera motion, adds boats (days)
3. TD adds lighting, particle simulation, bow wake (8 weeks)
4. Compositor puts it all together (2 weeks)

Altogether…

- 250-300 shots (~5 seconds each)
  - 75 TDs
  - 12 animators
  - 6 “ocean scouts”
  - 15 “match movers”
  - 15 compositors
- All in all: *close to 100,000 man-hours!*

Thus, the problem

The process for creating CG doesn't scale:

- Good models are hard to come by
- Need a human expert to create each one

This is fine for big studios, but not for independent artists, home users, researchers, etc.

Data-driven computer graphics

What if we can get models from the real world?

Overview

Intro (10 mins)

Character Animation (80 mins)

- Motion textures
- Probabilistic kinematic models
- Biomechanical Models

Texture (60 mins)

- Texture synthesis and extensions
- Image Analogies

Key themes

1. Black-box models vs. “strong” models

- “Strong” models leverage relevant domain knowledge
- Require less data to be predictive (sometimes *much* less)
- Can be more efficient
- But require more effort to design

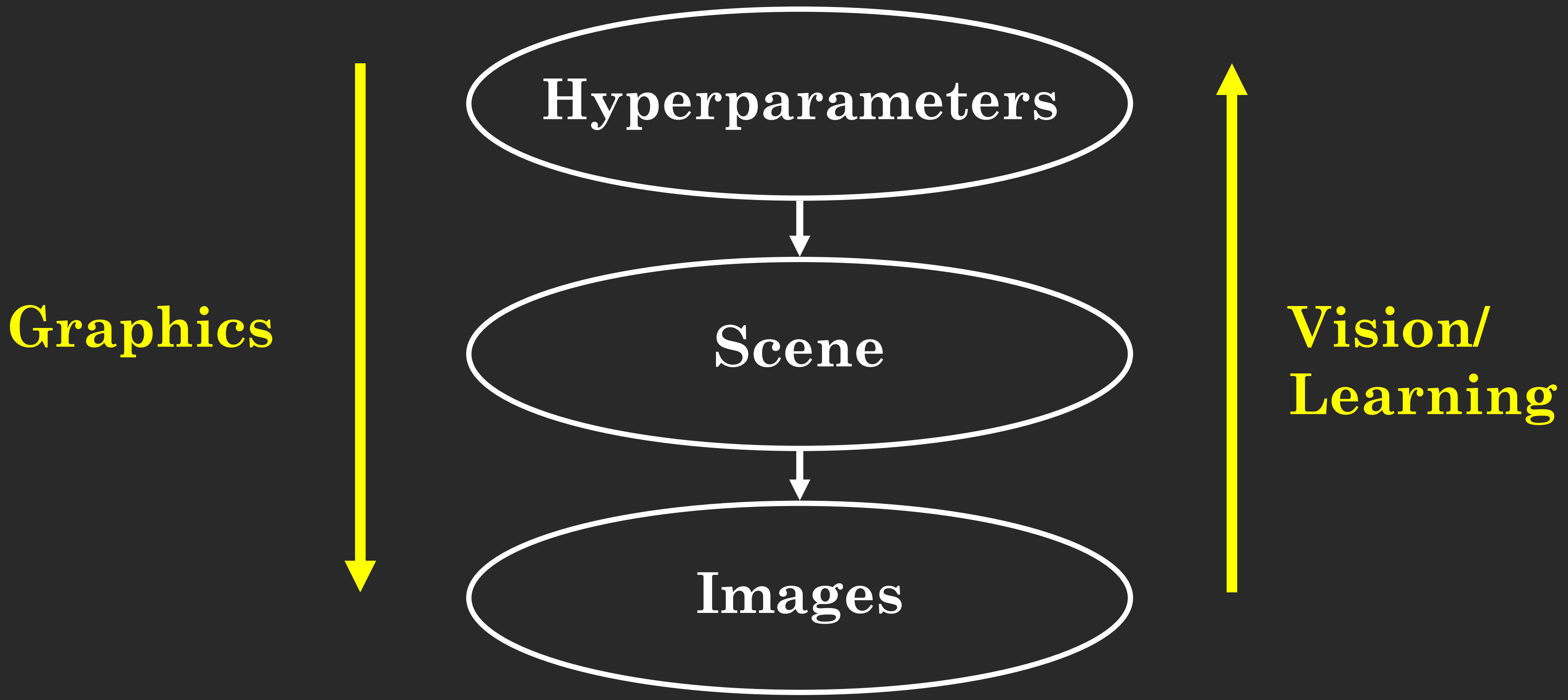
Key themes

2. Algorithms driven by the application, not the tools

    - Formulate the problem, then design a model and an algorithm
    - Does the algorithm solve the real problem?
    - Don’t just apply your favorite learning algorithm uncritically

Key themes

3. Symbiosis of graphics and vision



Key themes

4. Practical issues matter

- As researchers, we want to study the fundamentals of a problem
- But complicated and slow techniques are much less likely to be adopted
- (This contradicts theme #1)

Character animation

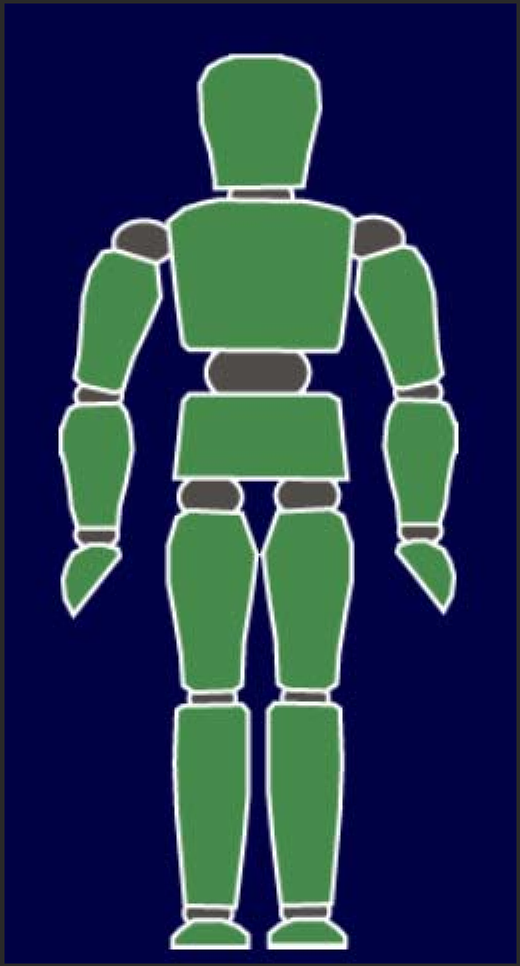
Body parameterization

Pose at time t: $\mathbf{q}_t$

- Root pos./orientation (6 DOFs)
- Joint angles (29 DOFs)

Motion

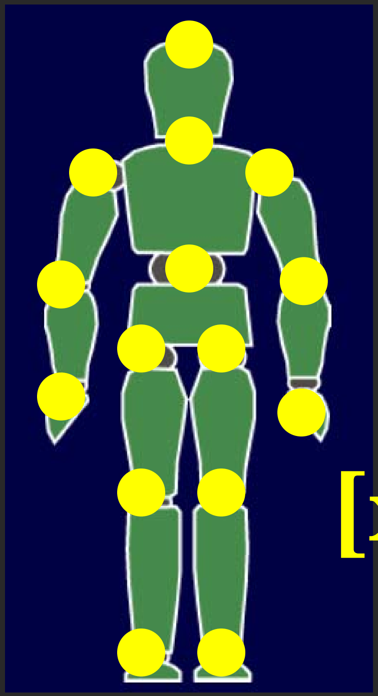
- $\mathbf{X} = [\mathbf{q}_1, \ldots, \mathbf{q}_T]$



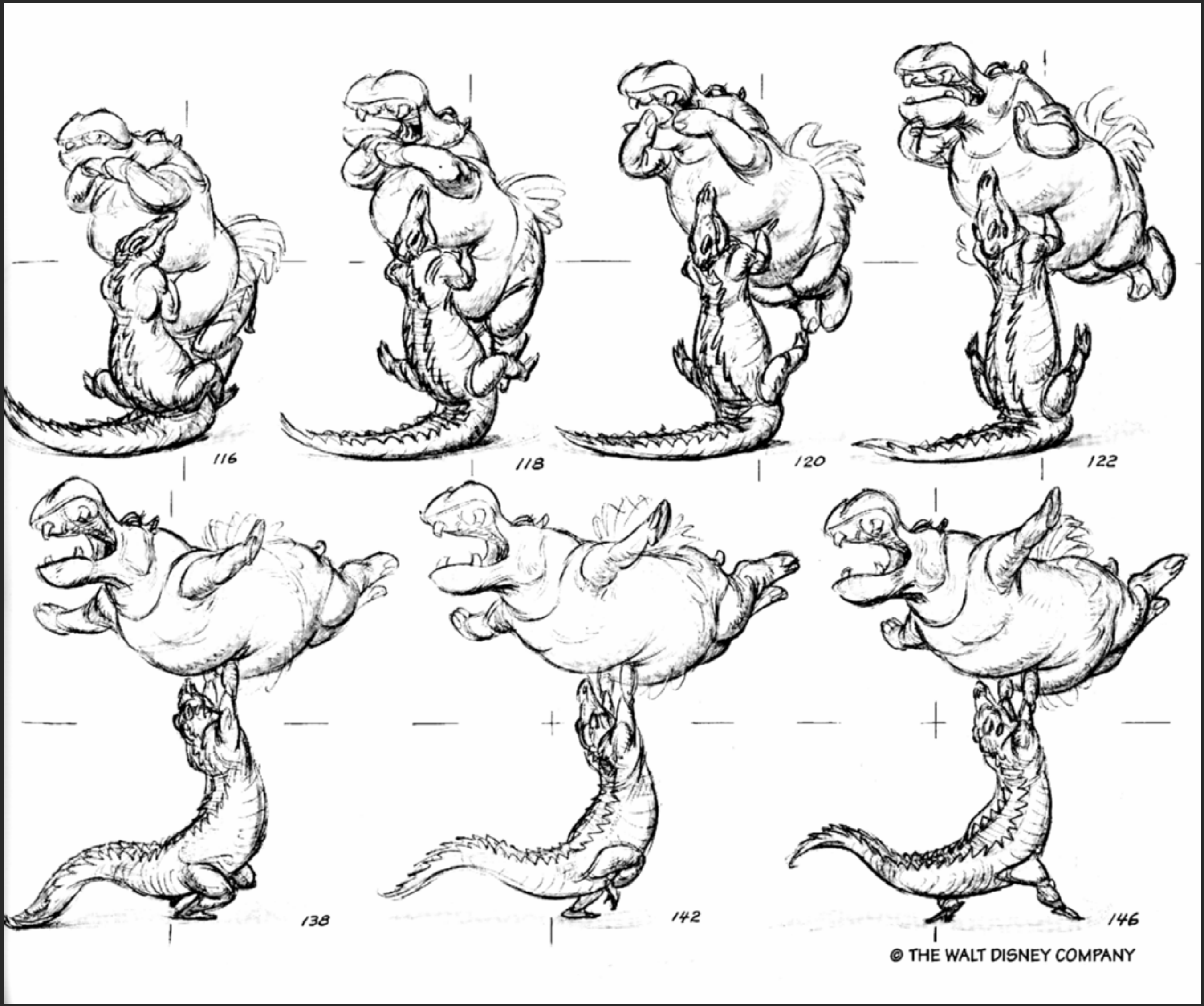
Forward kinematics

Pose to 3D positions:

$$\mathbf{q}_t \xrightarrow{\text{FK}} [\mathbf{x}_i,\mathbf{y}_i,\mathbf{z}_i]_t$$



Keyframe animation

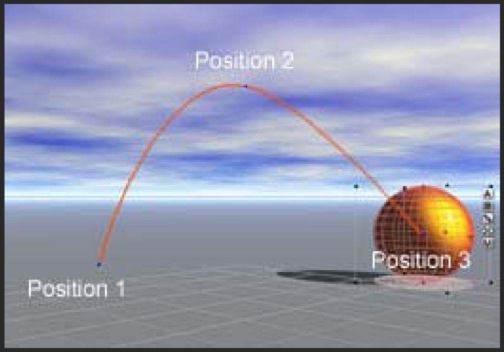


Keyframe animation

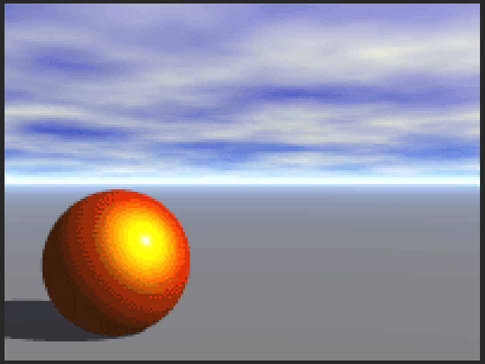
$\mathbf{q}_1$

$\mathbf{q}_2$

$\mathbf{q}_3$



$\mathbf{q}(t)$



$\mathbf{q}(t)$

http://www.cadtutor.net/dd/bryce/anim/anim.html

Keyframe animation

- Define a set of *key poses*: $[q_{1,} \ldots, q_T]$
- *Interpolate* to produce $q(t)$
  - typically, with spline curves

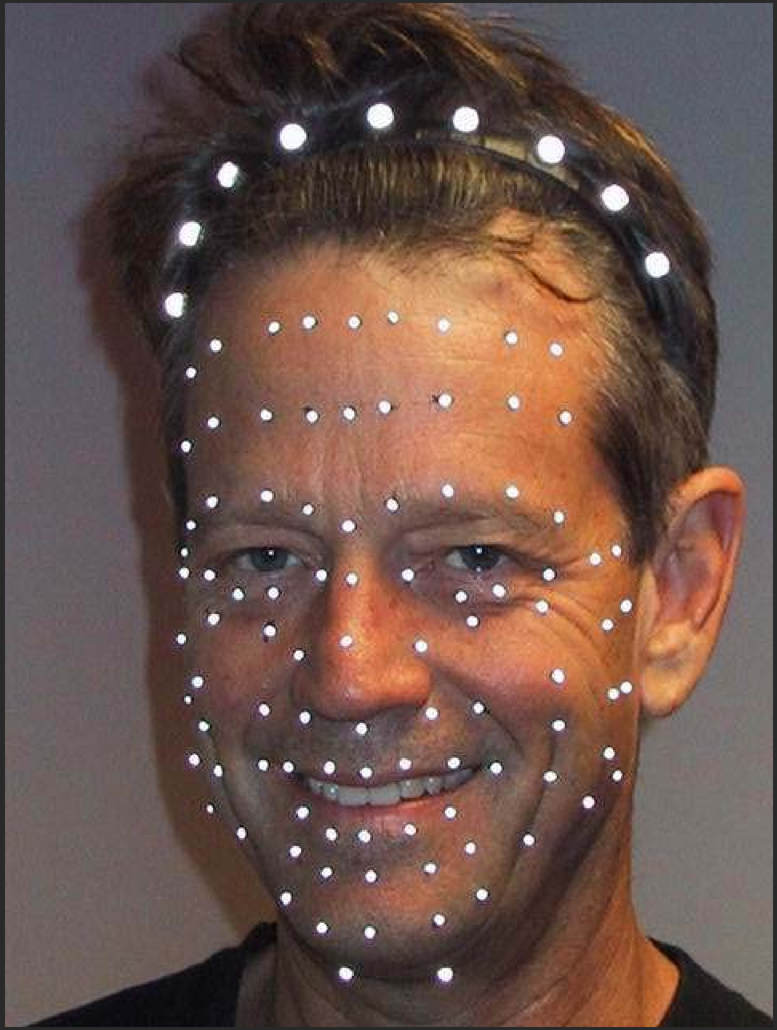
Summary of keyframing

- Pros:
  - very expressive. total control to the artist
- Cons:
  - very labor intensive
  - hard to create physical realism
  - hard to match individual style
- Uses:
  - potentially, everthing except complex physical phenomena (e.g., smoke)

Motion capture

[Images from NYU and UW]

Motion capture



Mocap is not a panacea




Motion capture

Demo

Summary of motion capture

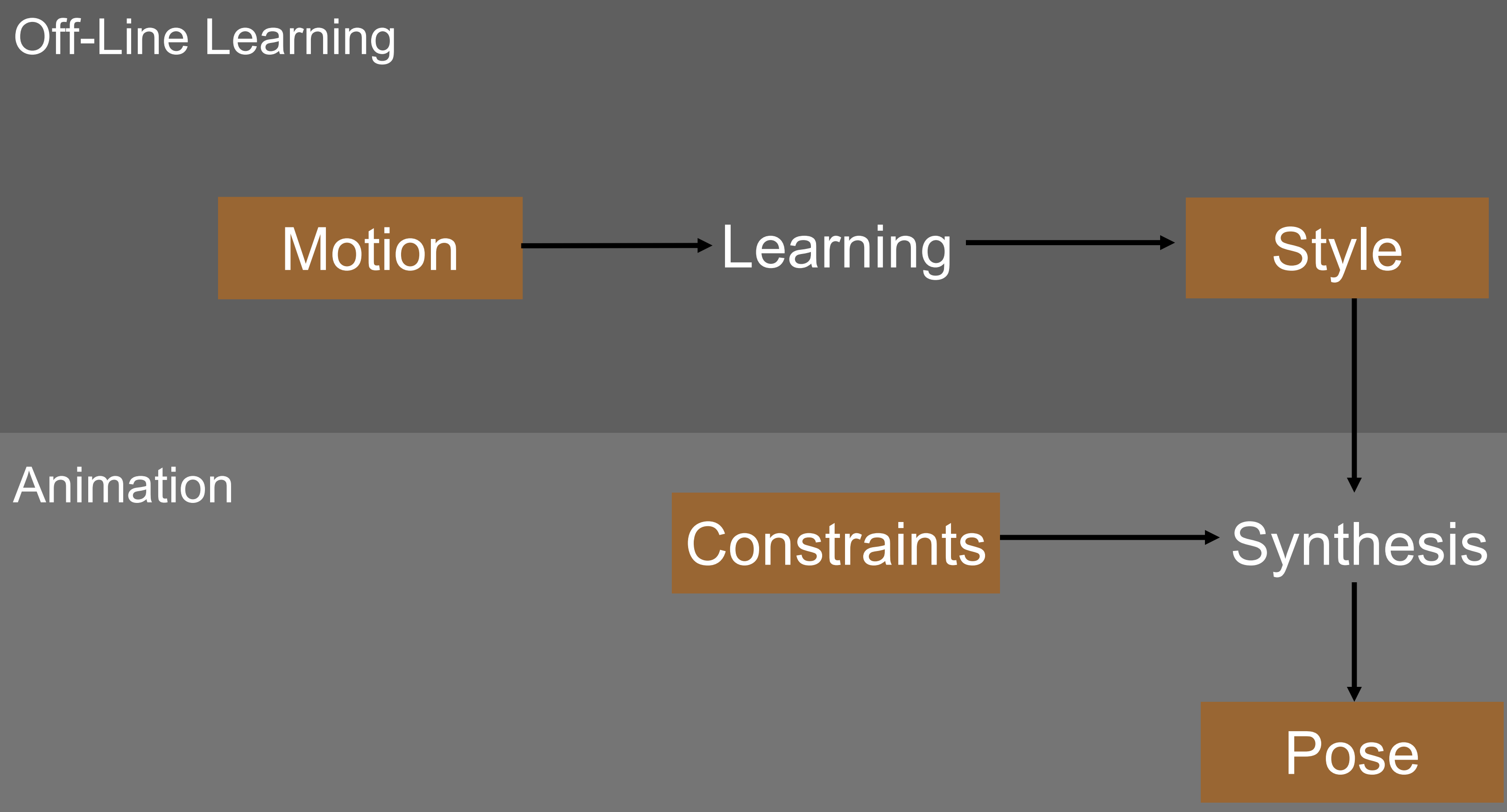
- Pros:
  - captures specific style of real actors
- Cons:
  - often not expressive enough (!)
  - time-consuming and expensive
    - lots of equipment, space, actors
    - manual clean-up
  - hard to edit
- Uses:
  - character animation
  - medicine (kinesiology, biomechanics)

Data-driven animation

Can we learn motion style from examples

- How do we model and represent style?
- Representation will directly affect quality of the results

Data-driven animation

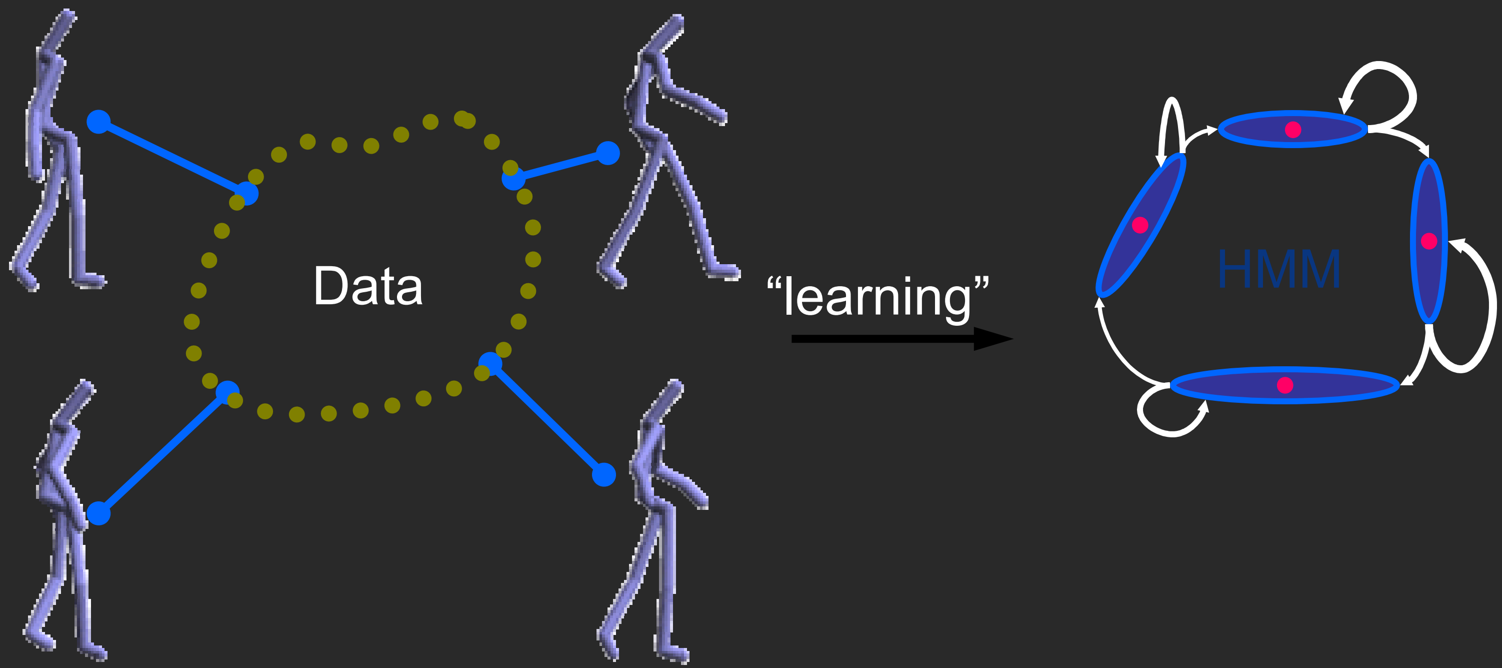


Probabilistic motion models

- Given input motion sequences, learn PDF over motions
- Generating motions: constrained sampling from PDF

Probabilistic motion models

Walk cycle in pose/velocity space



Brand and Hertzmann, SIGGRAPH 2000, “Style machines”

Style machines demo

Summary of Style machines

Pros:

- generate novel sequences of motions
- parameterized style model

Cons:

- HMM is a poor model for continuous motion
- No kinematic control yet
- No physical model
  - e.g., ground contact is smoothed out
- Our learning strategy could be improved…

Mark V. Chaney

[Shannon 48] proposed a way to generate English-looking text using N-grams:

- Assume a generalized Markov model
- Use a large text to compute probability distributions of each letter given N-1 previous letters
  - precompute or sample randomly
- Starting from a seed repeatedly sample this Markov chain to generate new letters
- One can use whole words instead of letters too:

WE NEED TO EAT CAKE

Mark V. Chaney

Results (using alt.singles corpus):

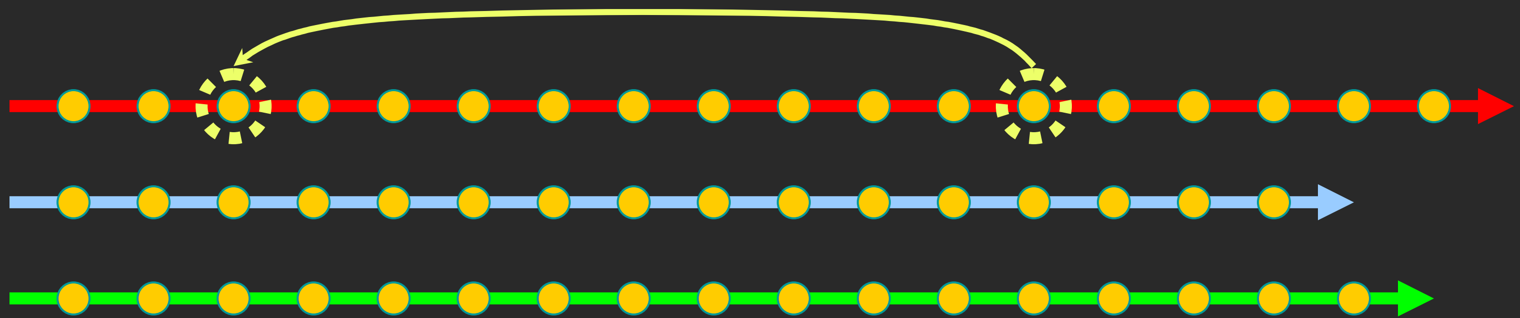
- *"As I've commented before, really relating to someone involves standing next to impossible."*
- *"One morning I shot an elephant in my arms and kissed him."*
- *"I spent an interesting evening recently with a grain of salt"*

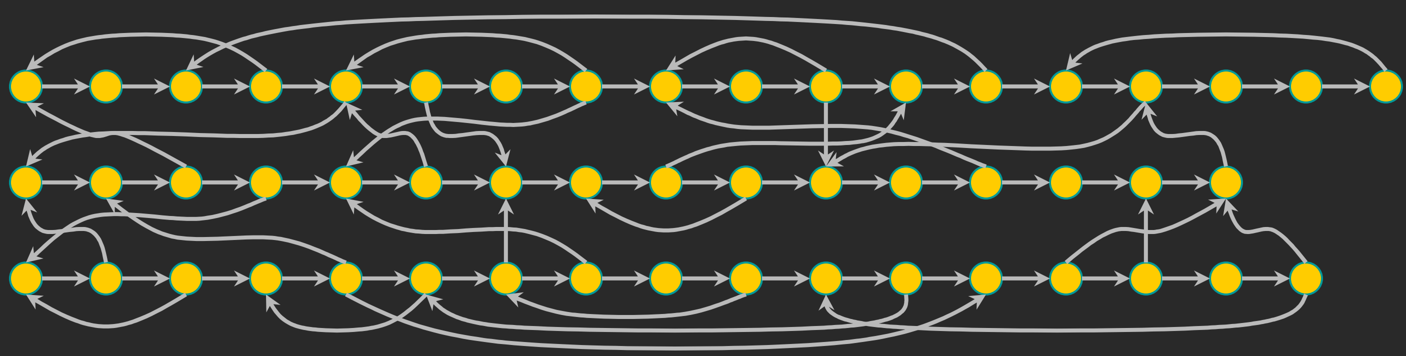
Motion graphs

- Idea: cut-and-paste motion capture to create new motion
- Four papers introduced the idea at SIGGRAPH 2002
- Inspired by texture synthesis algorithms (next hour)
- I'll outline one of them: J. Lee et al., Interactive Control of Avatars

Motion graphs

Input: raw motion capture





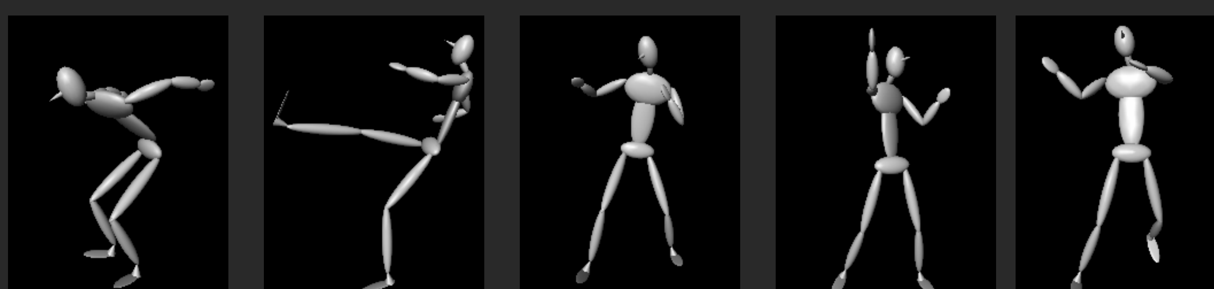
“Motion graph”

Distance between Frames

$$D(i, j) = d(p_i, p_j) + \alpha\, d(v_i, v_j)$$

Weighted differences
of joint angles

Weighted differences
of joint velocities



Pruning Transition

Contact state: Avoid transition to dissimilar contact state

Likelihood: User-specified threshold

Similarity: Local maxima

Avoid dead-ends: Strongly connected components

Run-time graph search

Best-first graph traversal

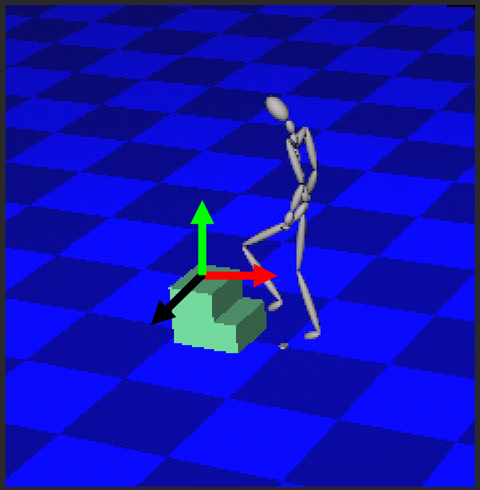
- Path length is bounded
- Fixed number of frames at each frame

Comparison to global search

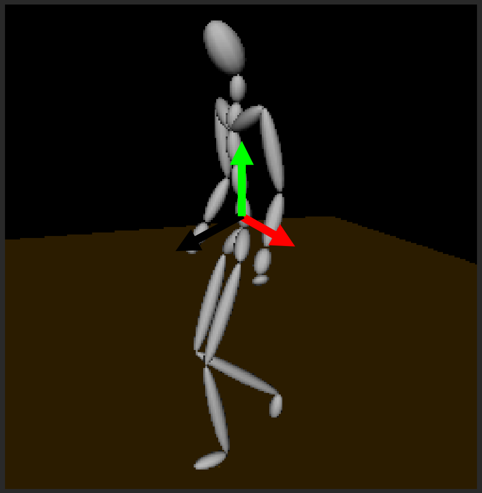
- Intended for interactive control
- Not for accurate global planning

Global vs. Local Coordinates

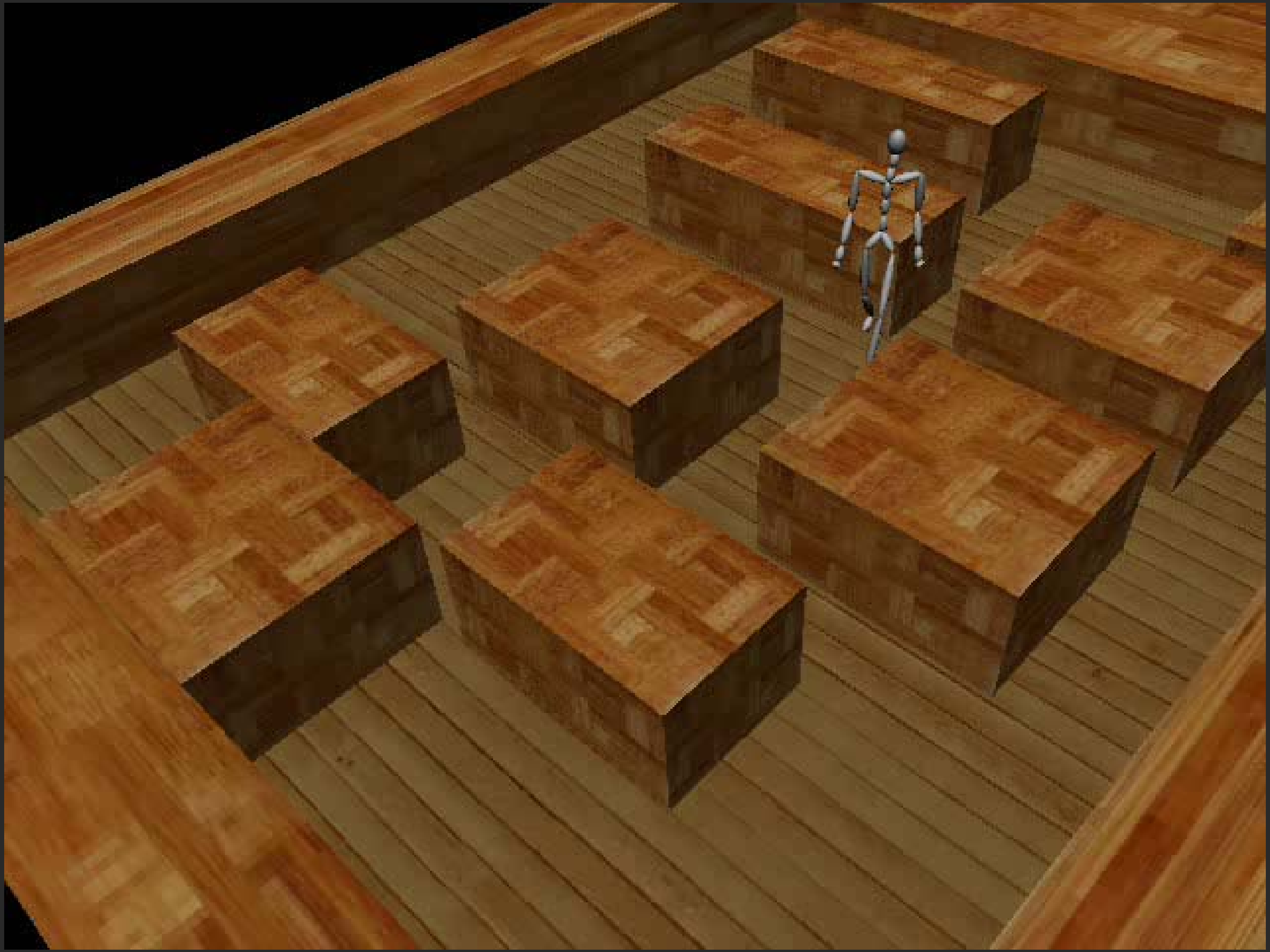
Global, fixed, object-relative coordinates



Local, moving, body-relative coordinates



Demo



Motion synthesis with annotations

- Arikan et al., SIGGRAPH 2003

Summary of motion textures

Pros:

- very realistic
- easy to understand and implement
- real-time synthesis

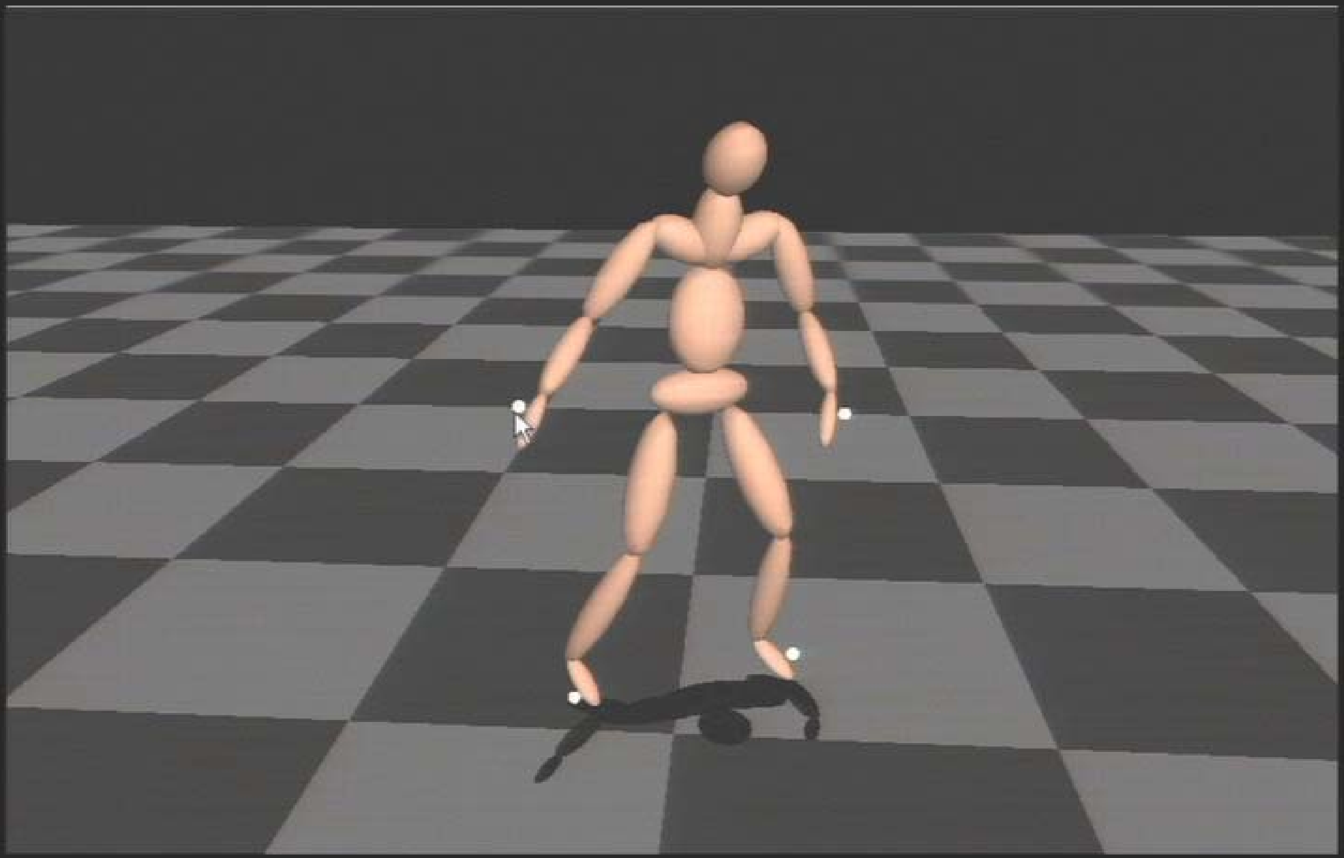
Cons:

- no generalization to new poses or new styles
- e.g., no kinematic/keyframe control

Style-Based Inverse Kinematics

with: Keith Grochow, Steve Martin, Zoran Popović

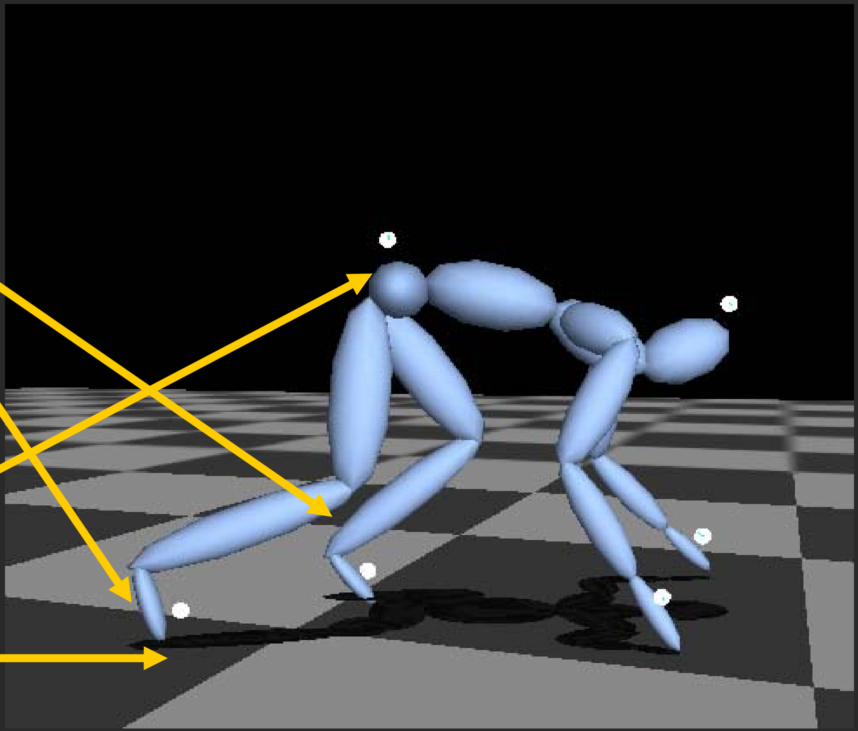
Motivation



Problem Statement

- Generate a character pose based on a chosen style subject to constraints

Degrees of freedom (DOFs) **q**



Constraints

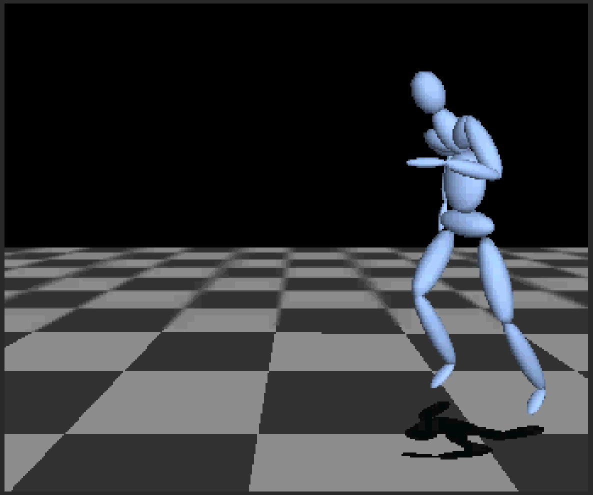
Style Representation

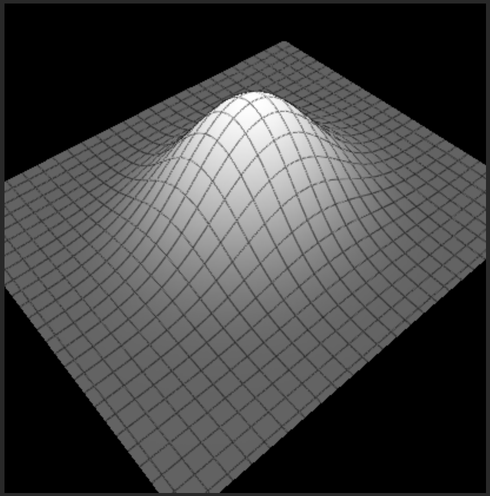
- Objective function
  - given a pose evaluate how well it matches a style
  - allow any pose

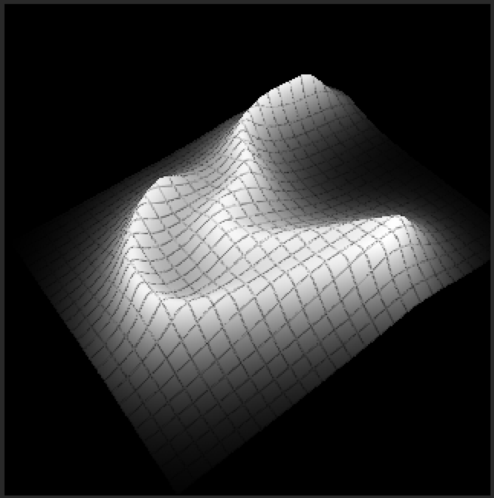
- Probability Distribution Function (PDF)
  - principled way of automatically learning the style

Goals for the PDF

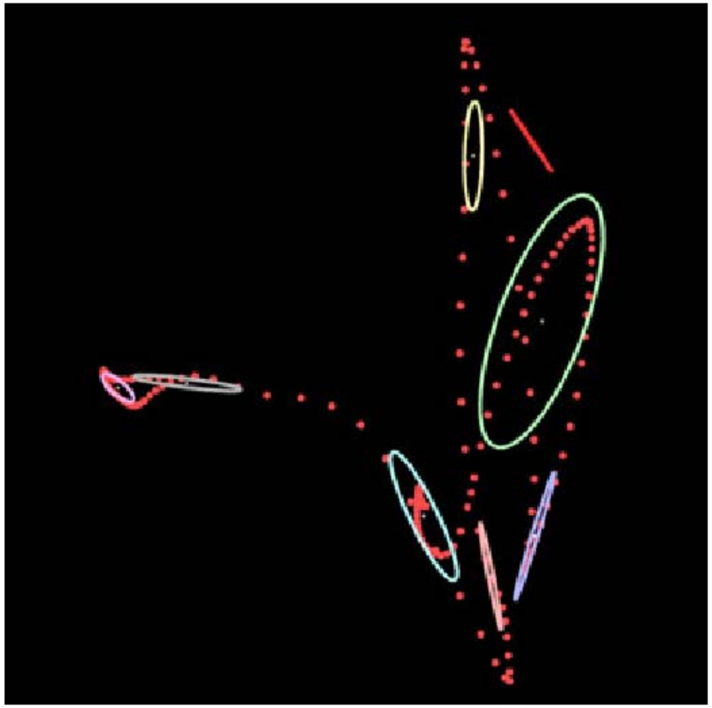
- Learn PDF from any data
- Smooth and descriptive
- Minimal parameter tuning
- Real-time synthesis



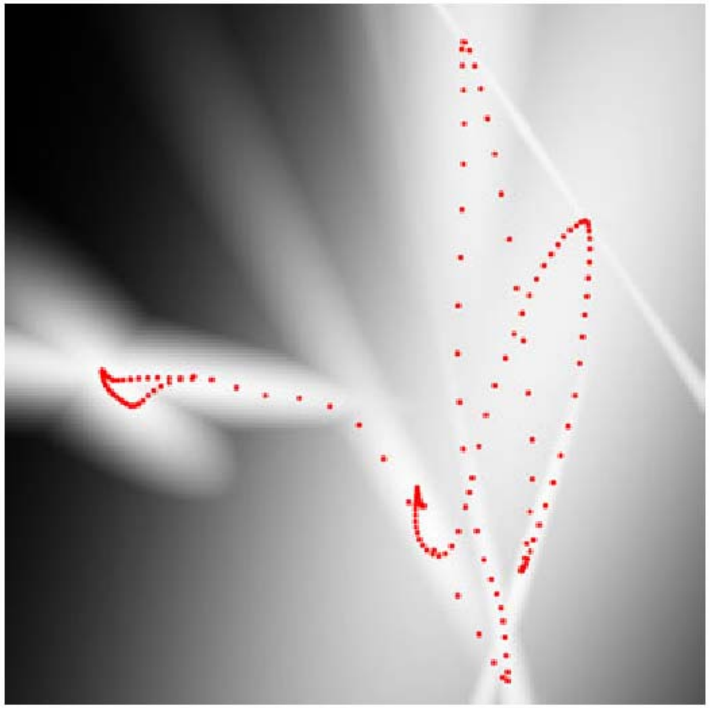




Mixtures-of-Gaussians



Gaussian components



Log-likelihood

SGPLVM

Scaled Gaussian Process Latent Variable Model

based on [Lawrence 2004]

– automatic parameter scaling
– extensions for real-time synthesis
– style interpolation

Gaussian Processes

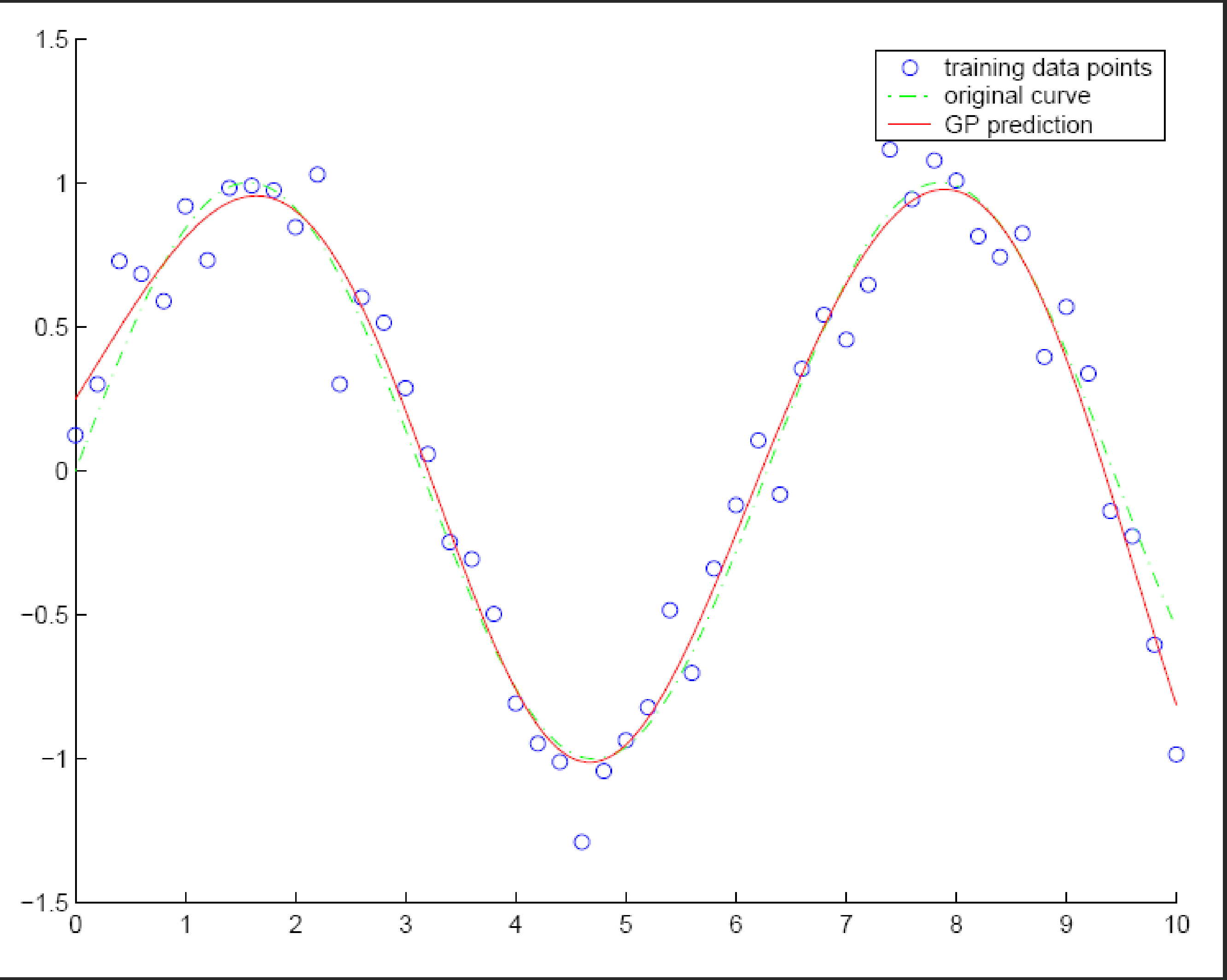
Let $g(x)$ be a nonlinear mapping, e.g., RBFs:

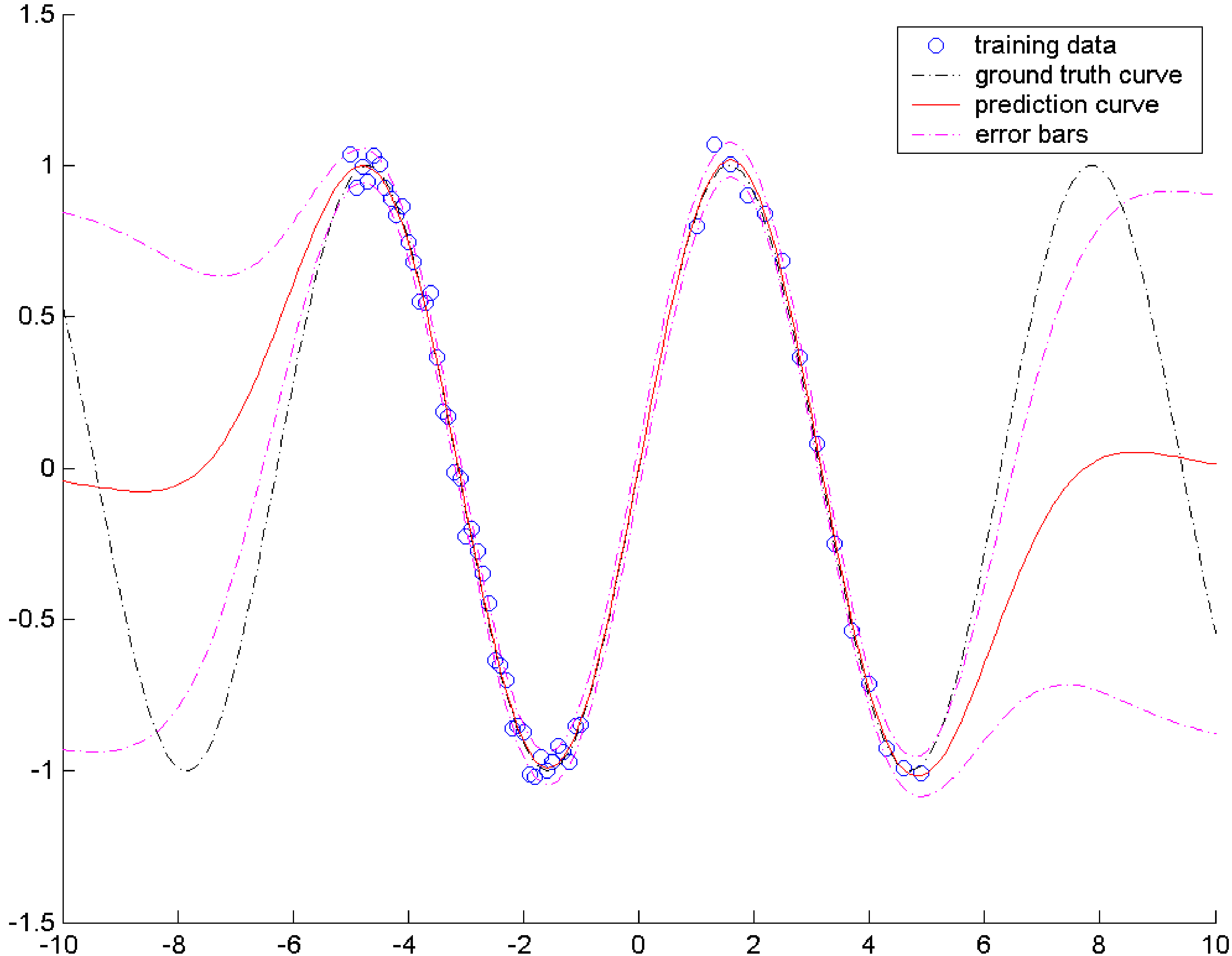
- $y = g(x) = \sum_i w_i \phi_i(x)$
- Gaussian prior on w_i

We can marginalize out w explicitly

$$p(y_{new} \mid X, Y) = \int p(y_{new}, w \mid X, Y)\, dw$$

Gaussian Processes

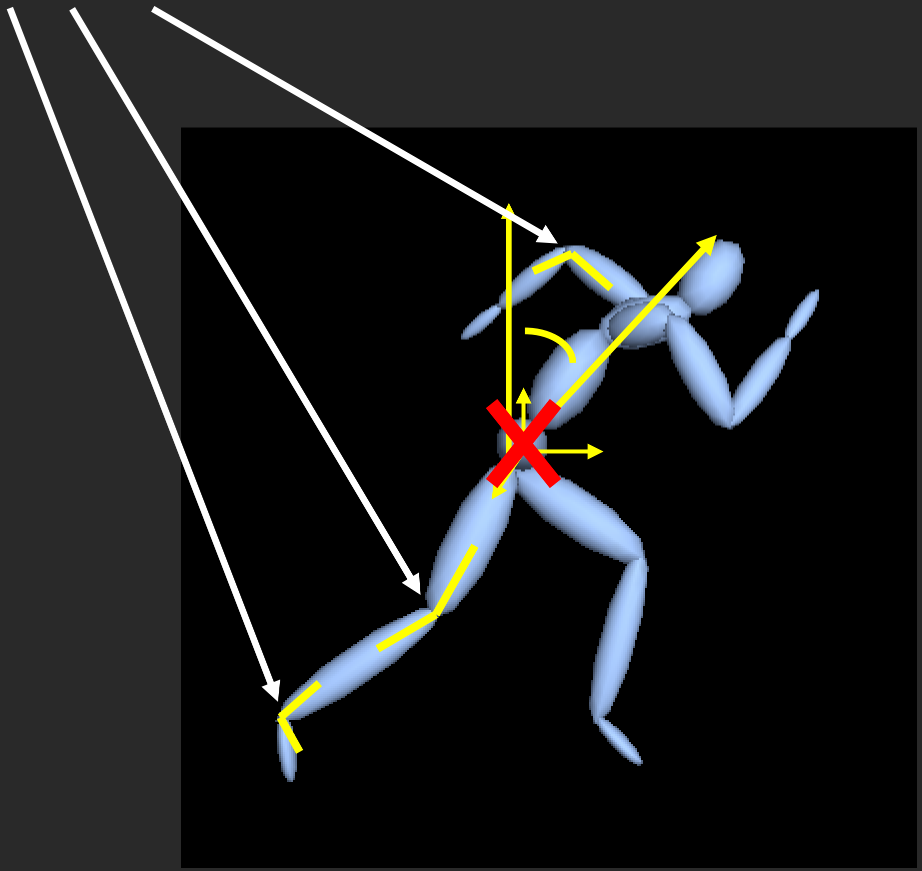




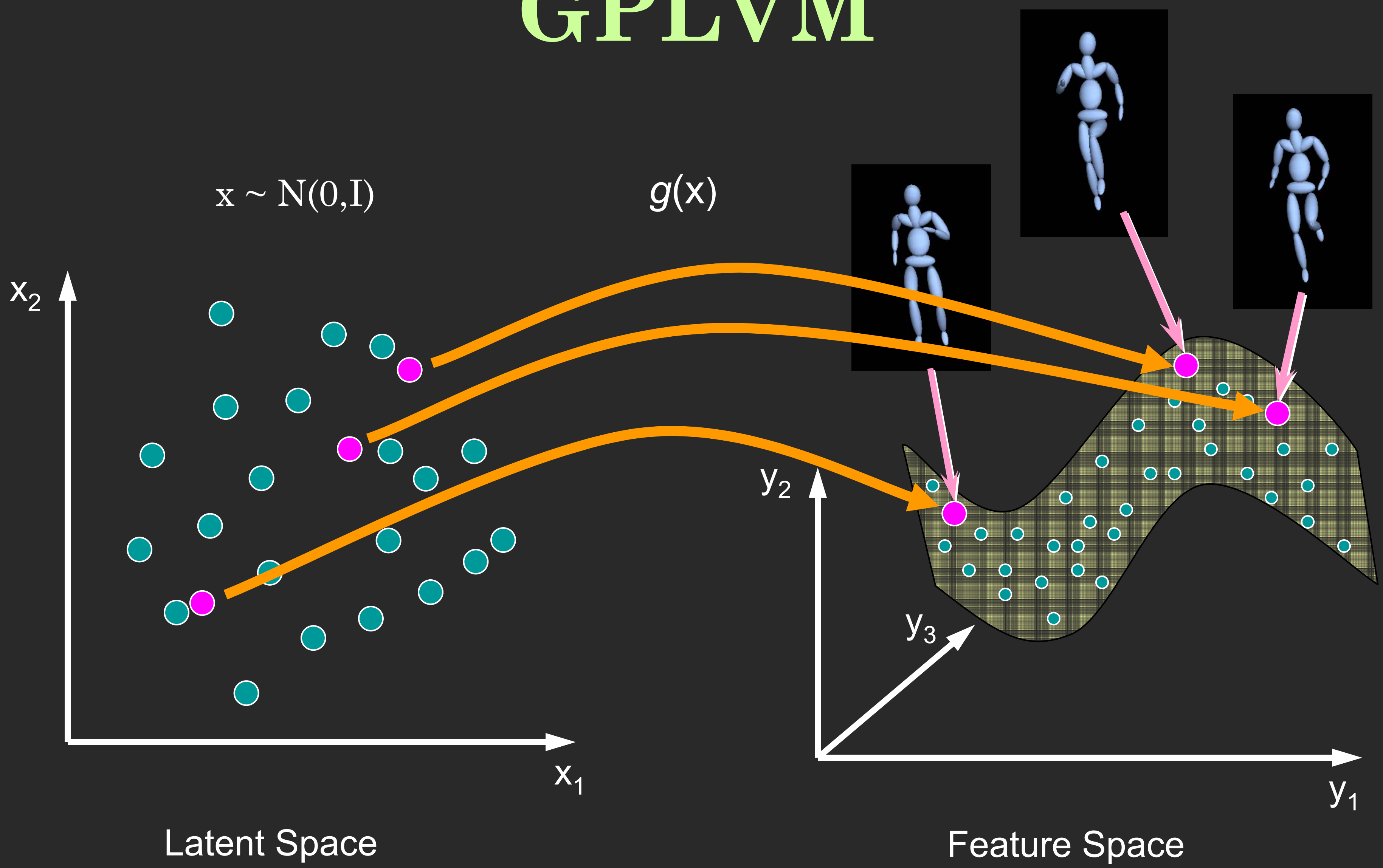
training data
ground truth curve
prediction curve
error bars
1.5
1
0.5
0
-0.5
-1
-1.5
-10
-8
-6
-4
-2
0
2
4
6
8
10

Features

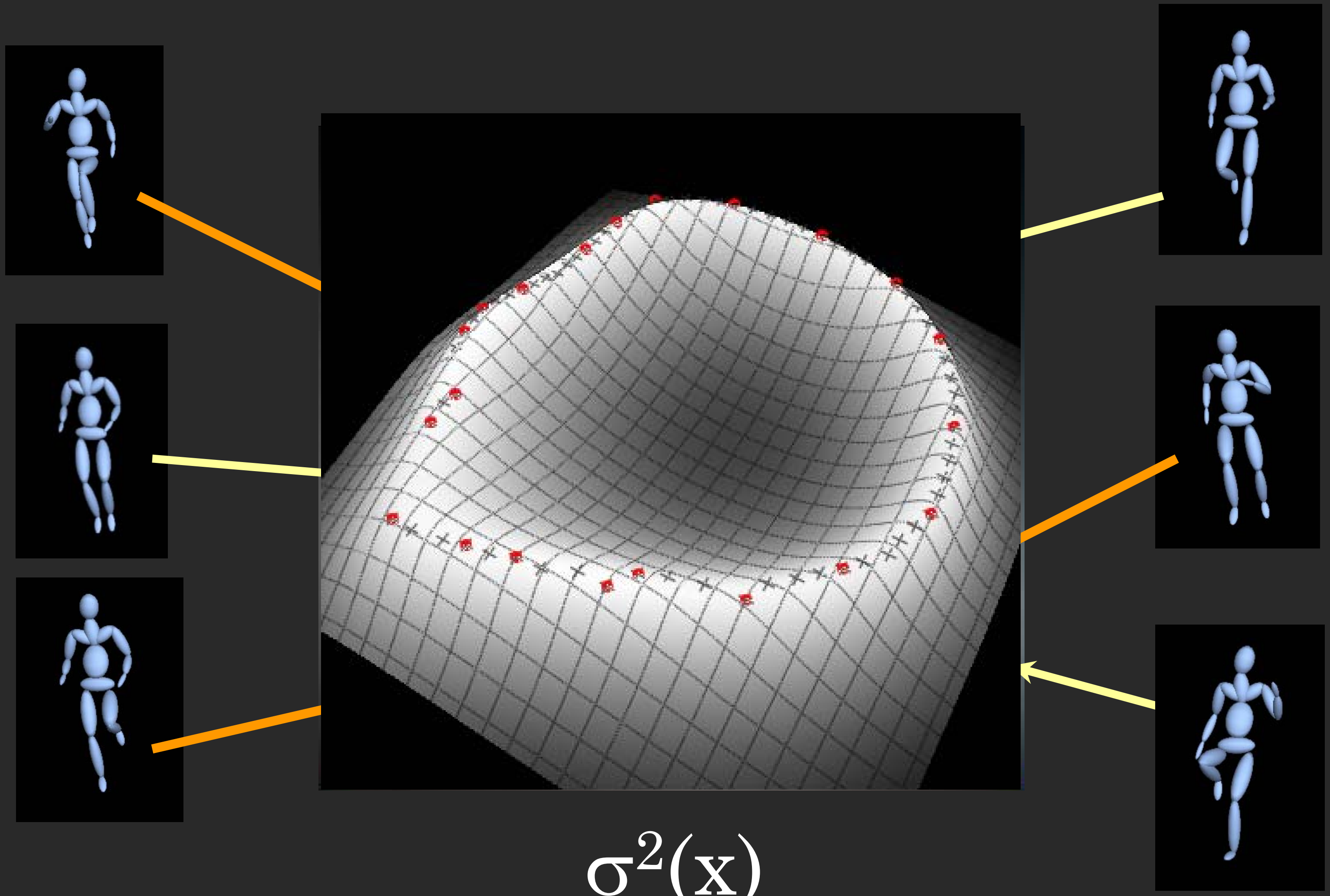
y



GPLVM

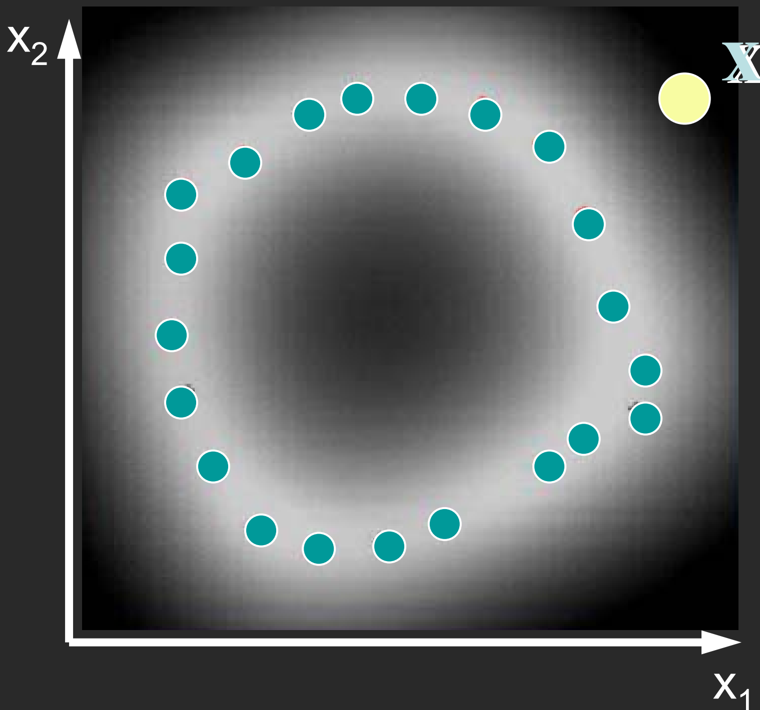


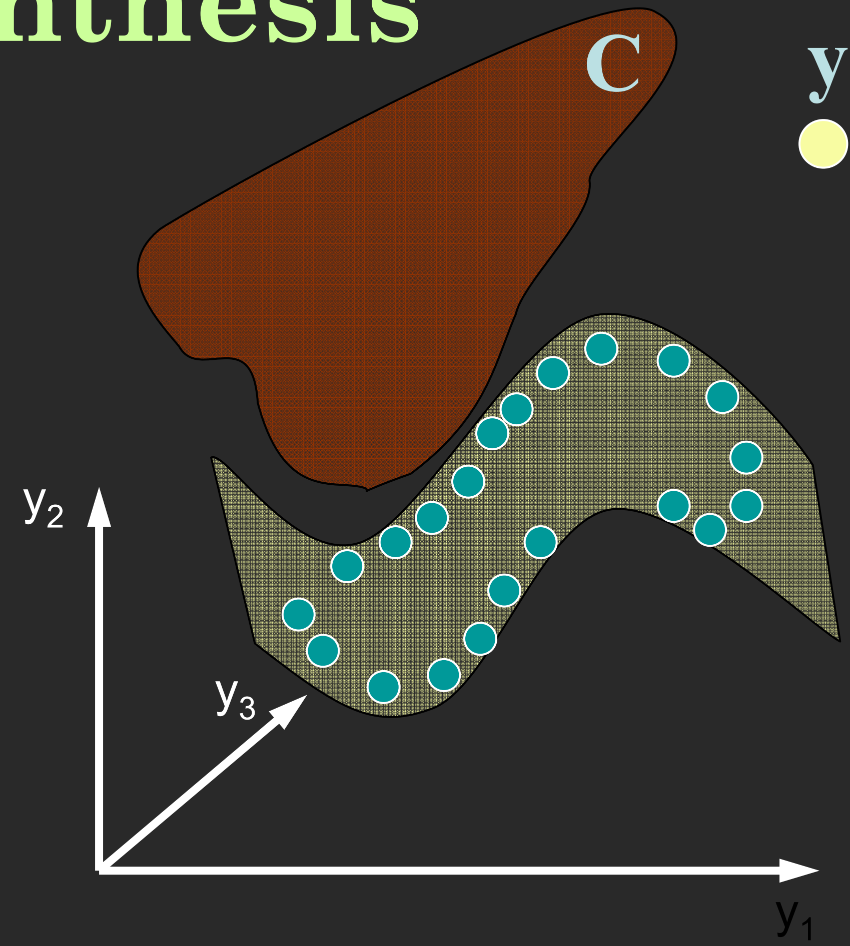
Precision in Latent Space



$\sigma^2(x)$

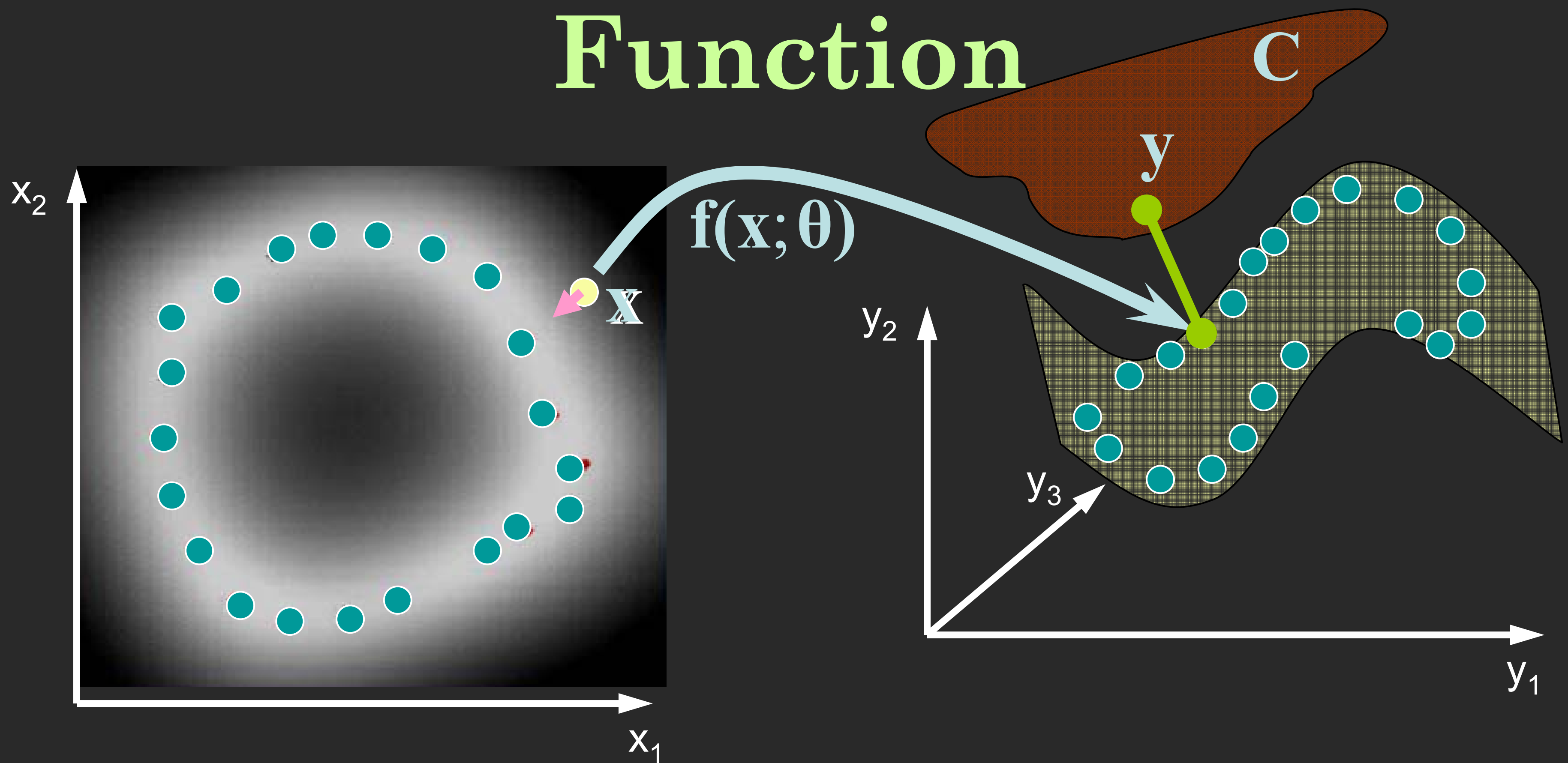
Pose Synthesis





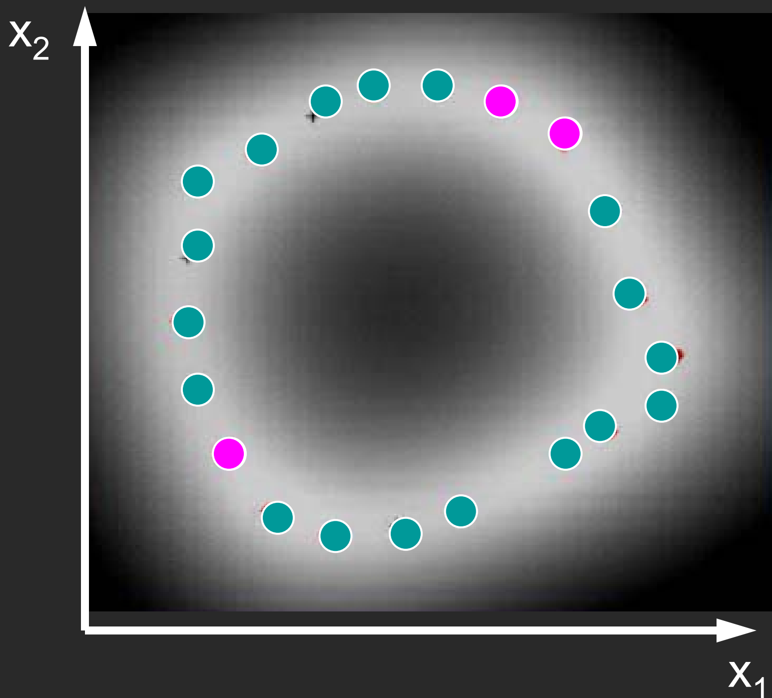
$$\arg\min_{x,q} L_{IK}(x, y(q); \theta)$$
$$\text{s.t. } C(q) = 0$$

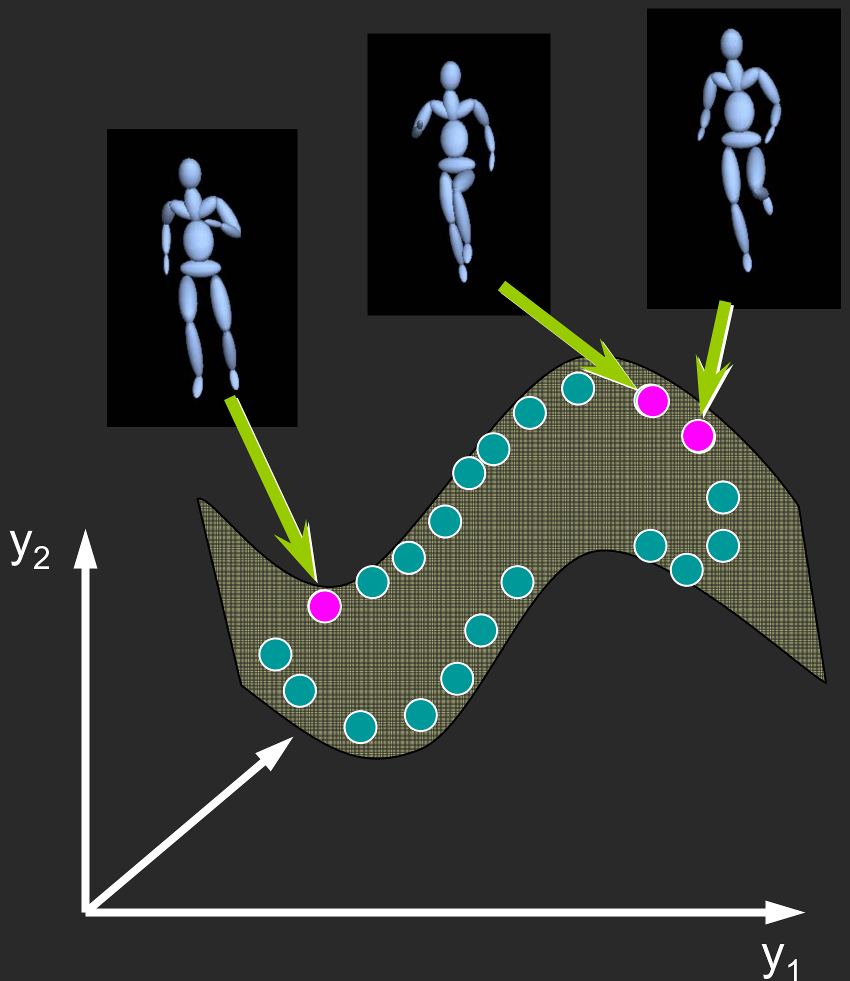
SGPLVM Objective Function



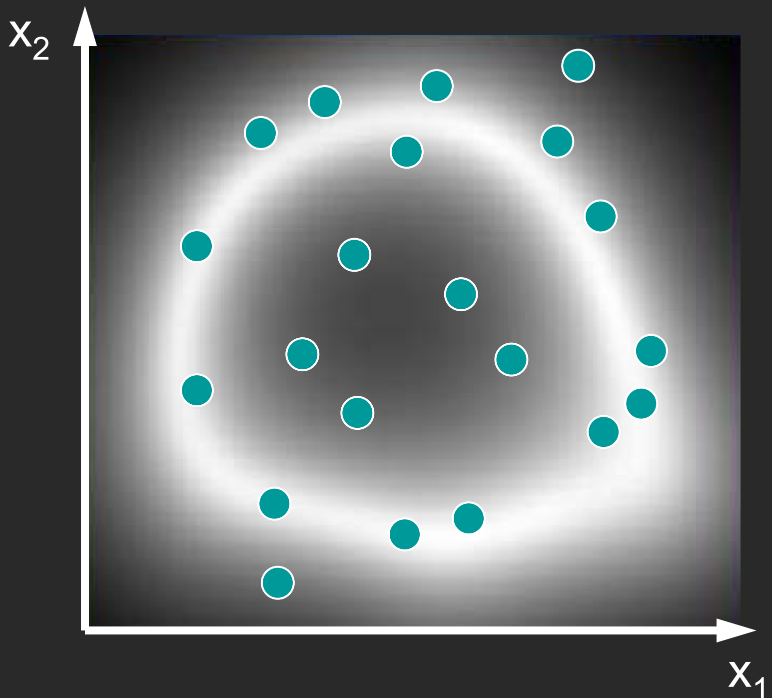
$$L_{IK}(\mathbf{x},\mathbf{y};\boldsymbol{\theta}) = \frac{\left\|\mathbf{W}_{\boldsymbol{\theta}}(\mathbf{y}-\mathbf{f}(\mathbf{x};\boldsymbol{\theta}))\right\|^2}{2\sigma^2(\mathbf{x};\boldsymbol{\theta})} + \frac{D}{2}\ln\sigma^2(\mathbf{x};\boldsymbol{\theta})$$

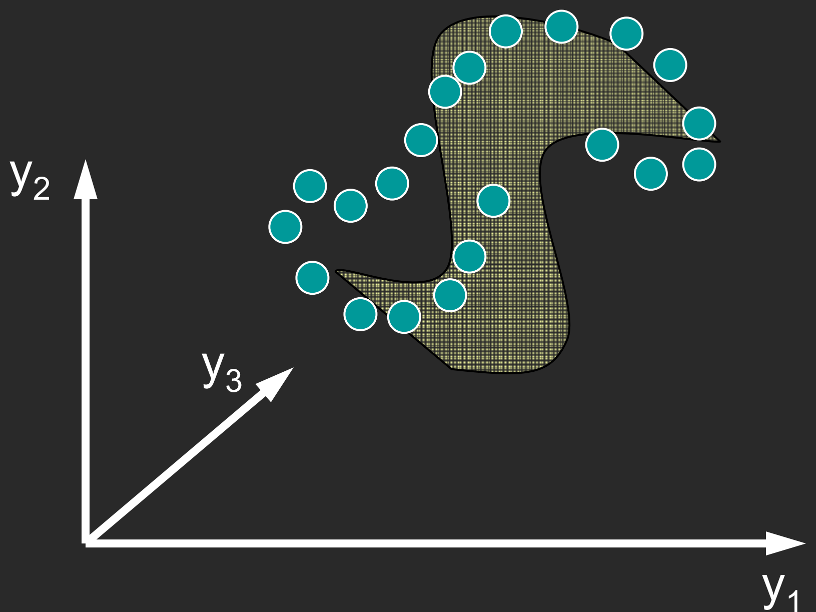
Style Learning



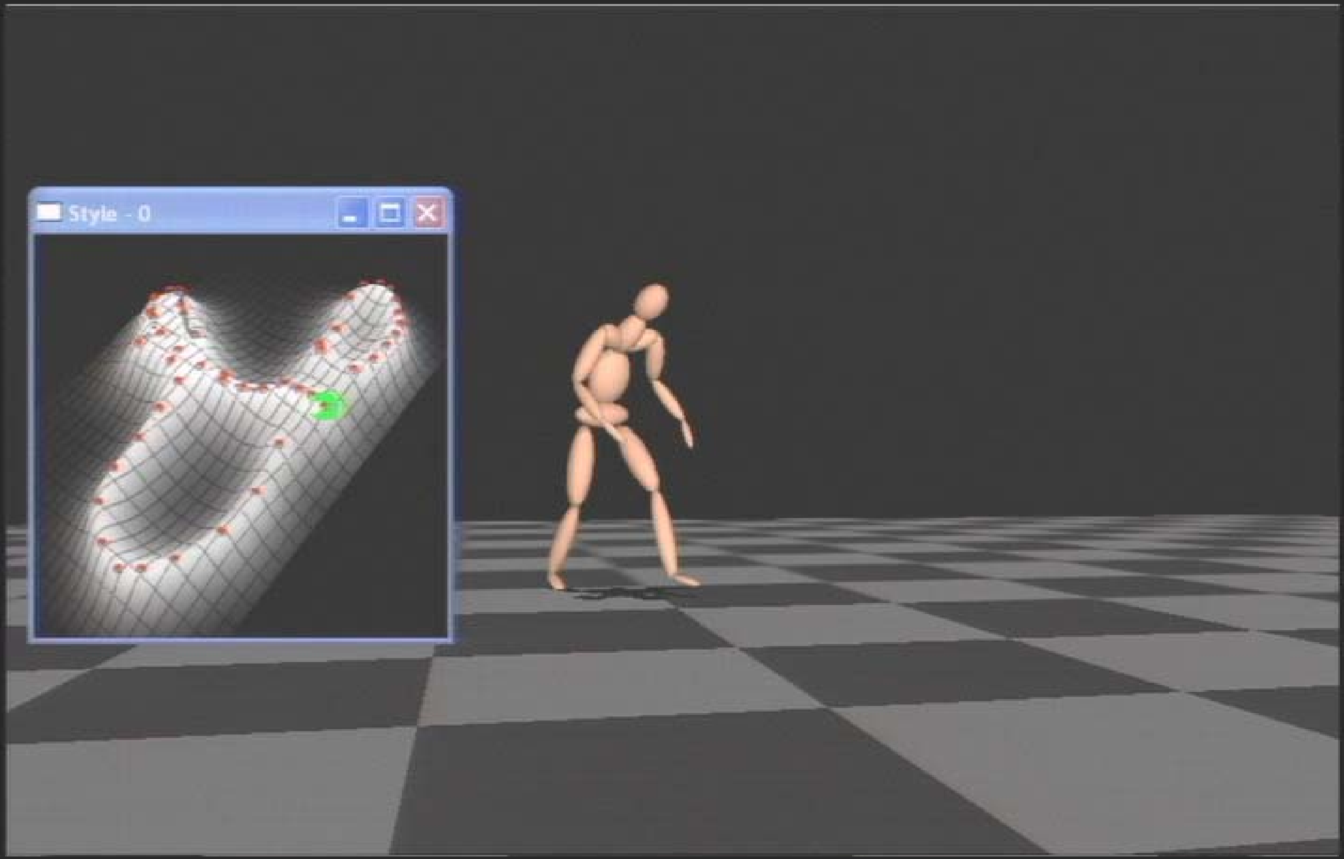


Style Learning





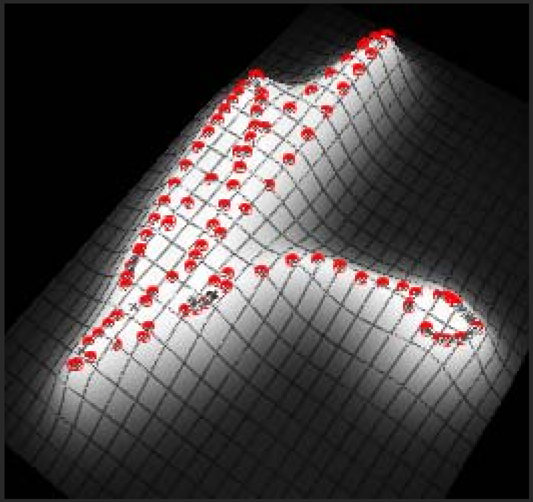
Different styles



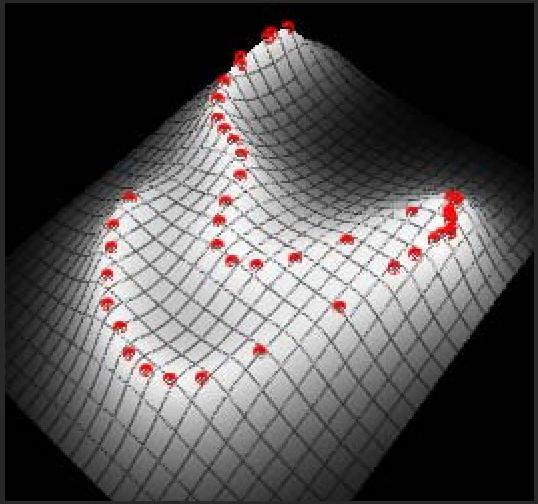
Baseball Pitch

Style interpolation

Given two styles θ_1 and θ_2, can we "interpolate" them?



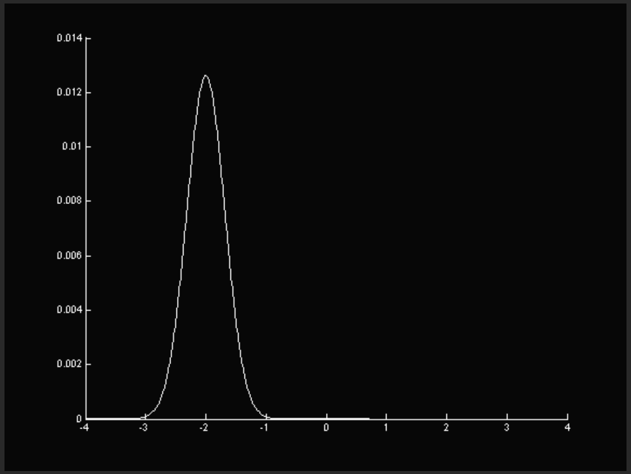
$p_1(\mathbf{y}) \propto \exp(-L_{IK}(\mathbf{y};\boldsymbol{\theta}_1))$

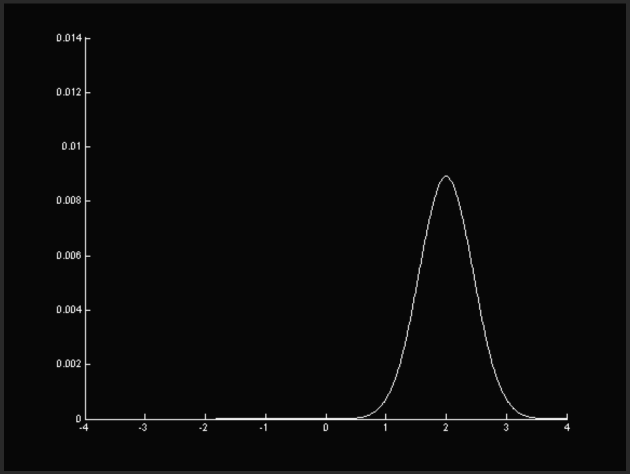


$p_2(\mathbf{y}) \propto \exp(-L_{IK}(\mathbf{y};\boldsymbol{\theta}_2))$

Approach: interpolate in log-domain

Style interpolation



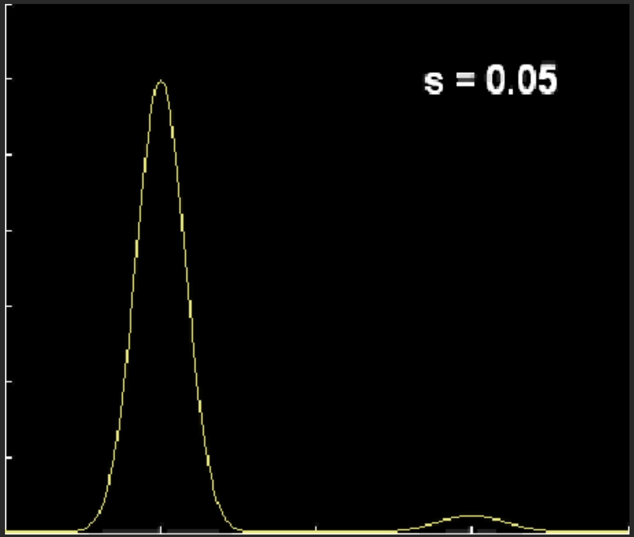


$$p_1(\mathbf{y}) \propto \exp(-L_{IK}(\mathbf{y};\boldsymbol{\theta}_1))$$

$$p_2(\mathbf{y}) \propto \exp(-L_{IK}(\mathbf{y};\boldsymbol{\theta}_2))$$

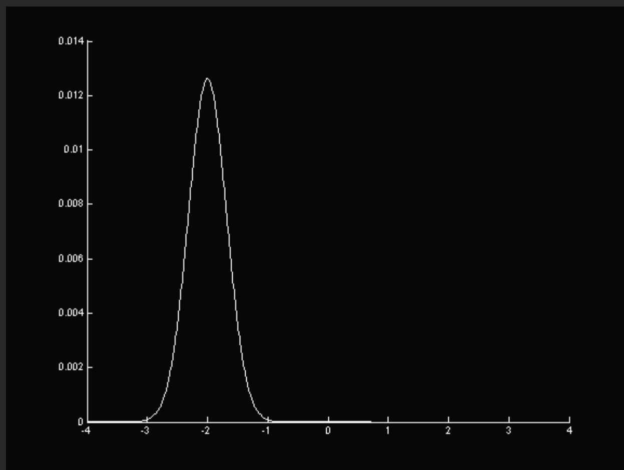
(1-s)

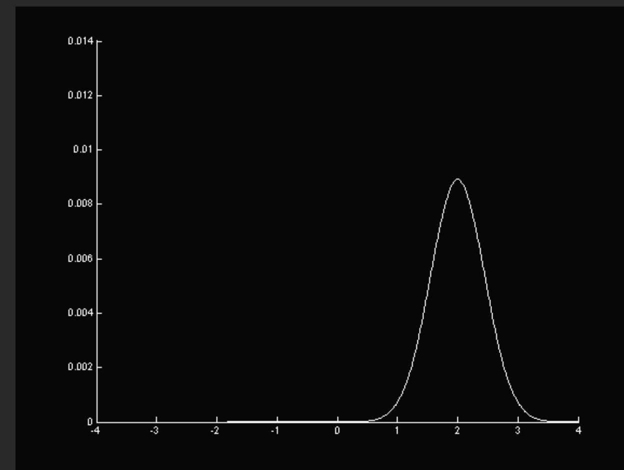
s



$$(1-\mathrm{s})\,p_1(y) + \mathrm{s}p_2(y)$$

Style interpolation in log space



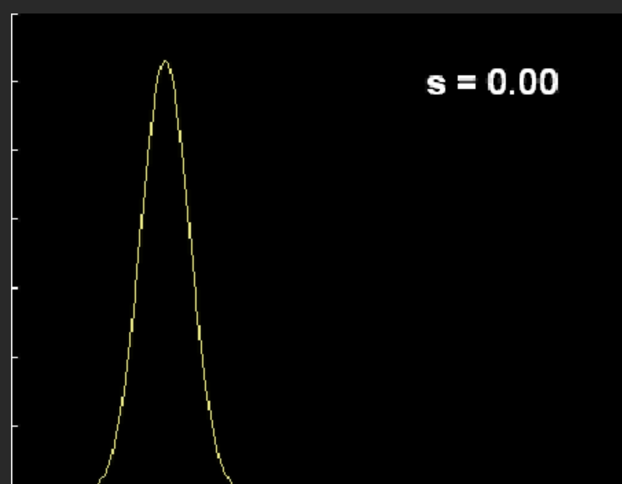


$\exp(-L_{IK}(\mathbf{y};\boldsymbol{\theta}_1))$

$\exp(-L_{IK}(\mathbf{y};\boldsymbol{\theta}_1))$

(1-s)

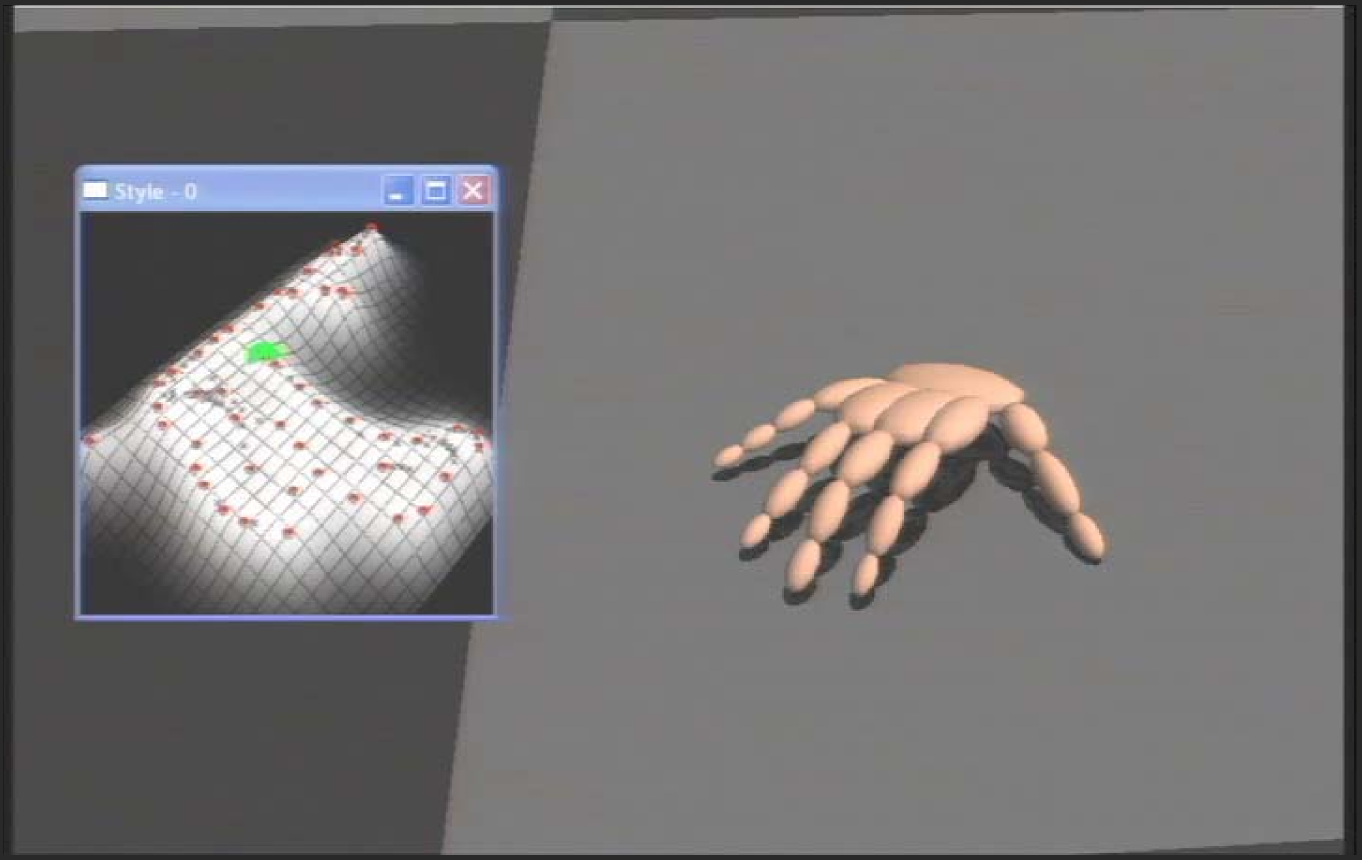
s



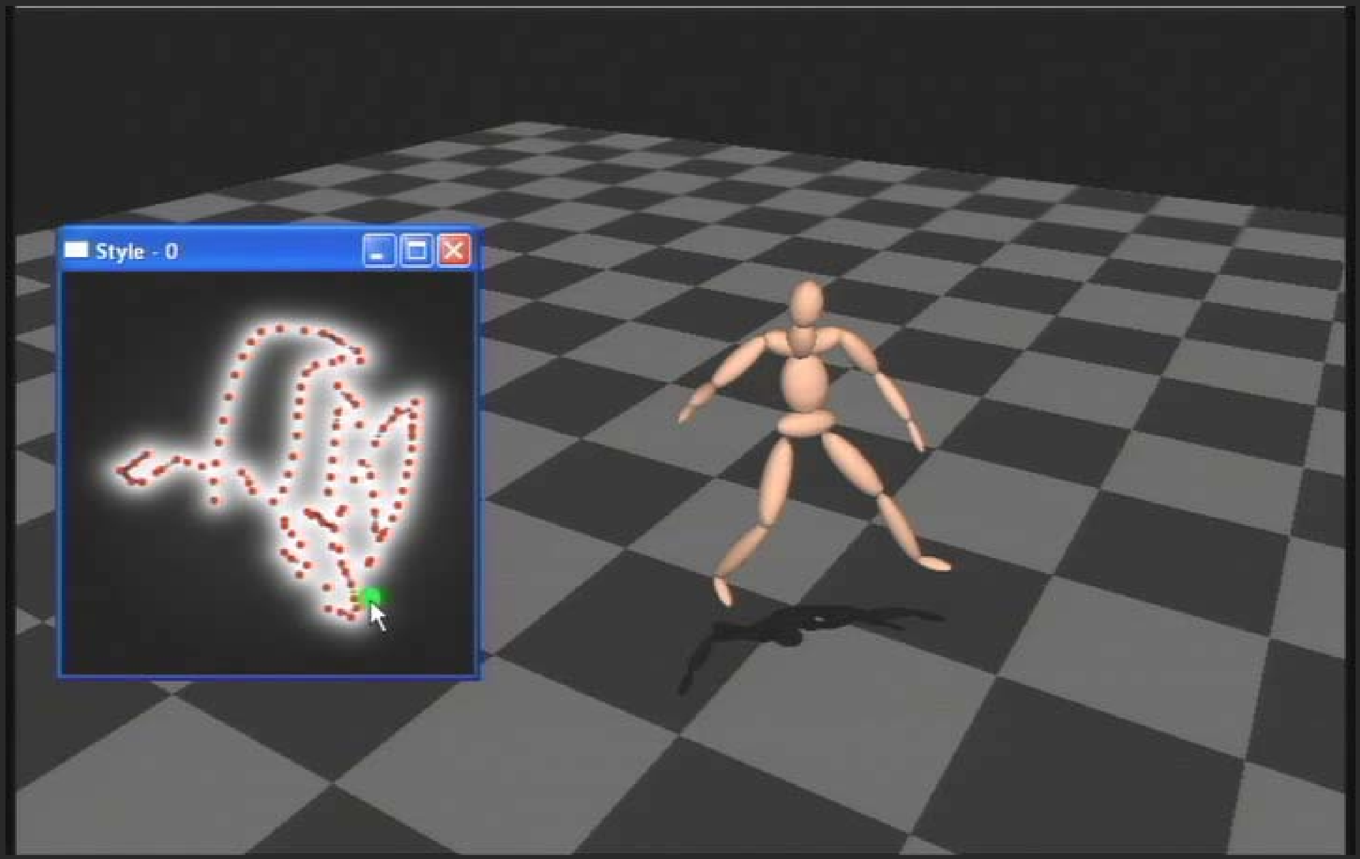
$\exp(-((1-\mathrm{s})L(\mathbf{y};\boldsymbol{\theta}_1)+\mathrm{s}L(\mathbf{y};\boldsymbol{\theta}_2)))$

Applications

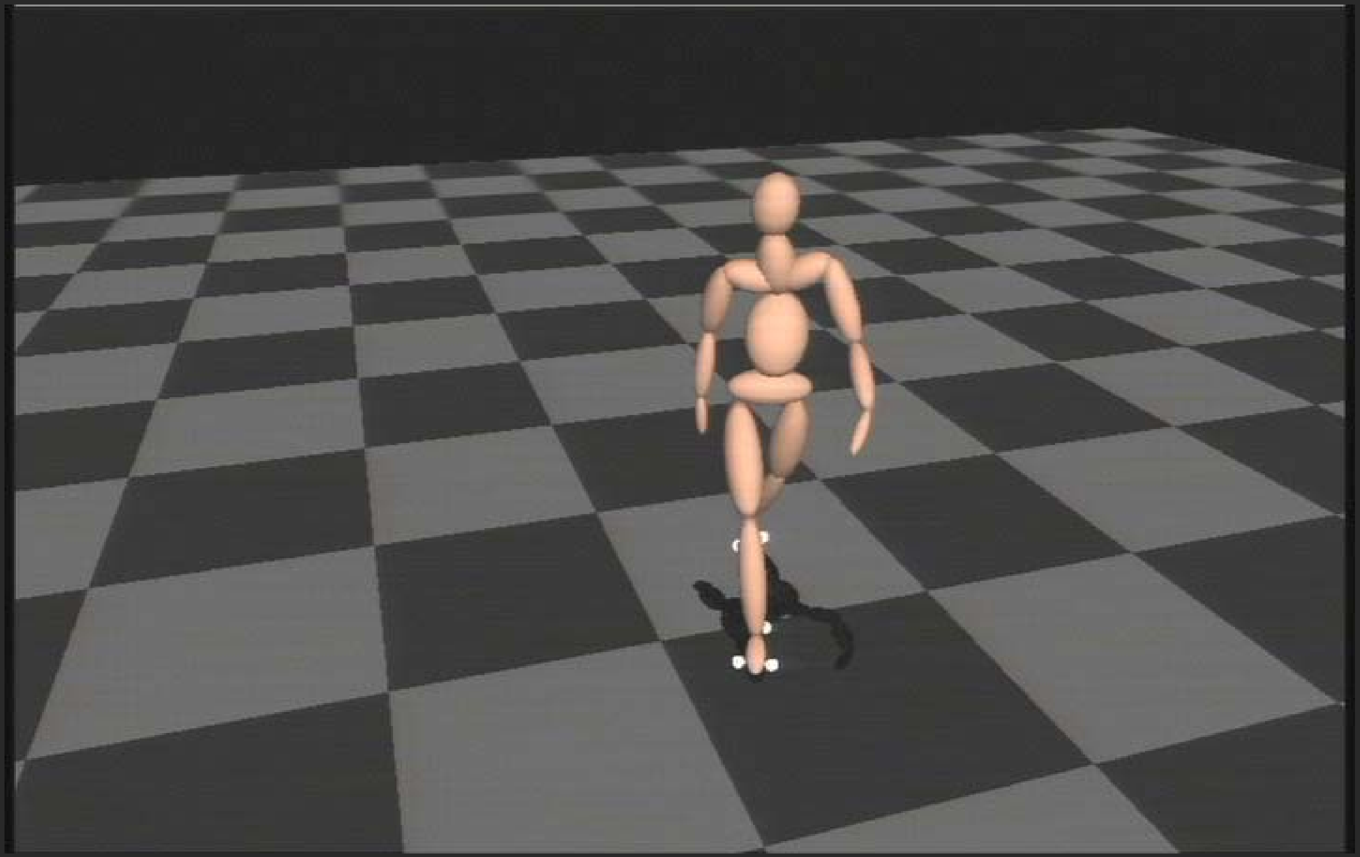
Interactive Posing



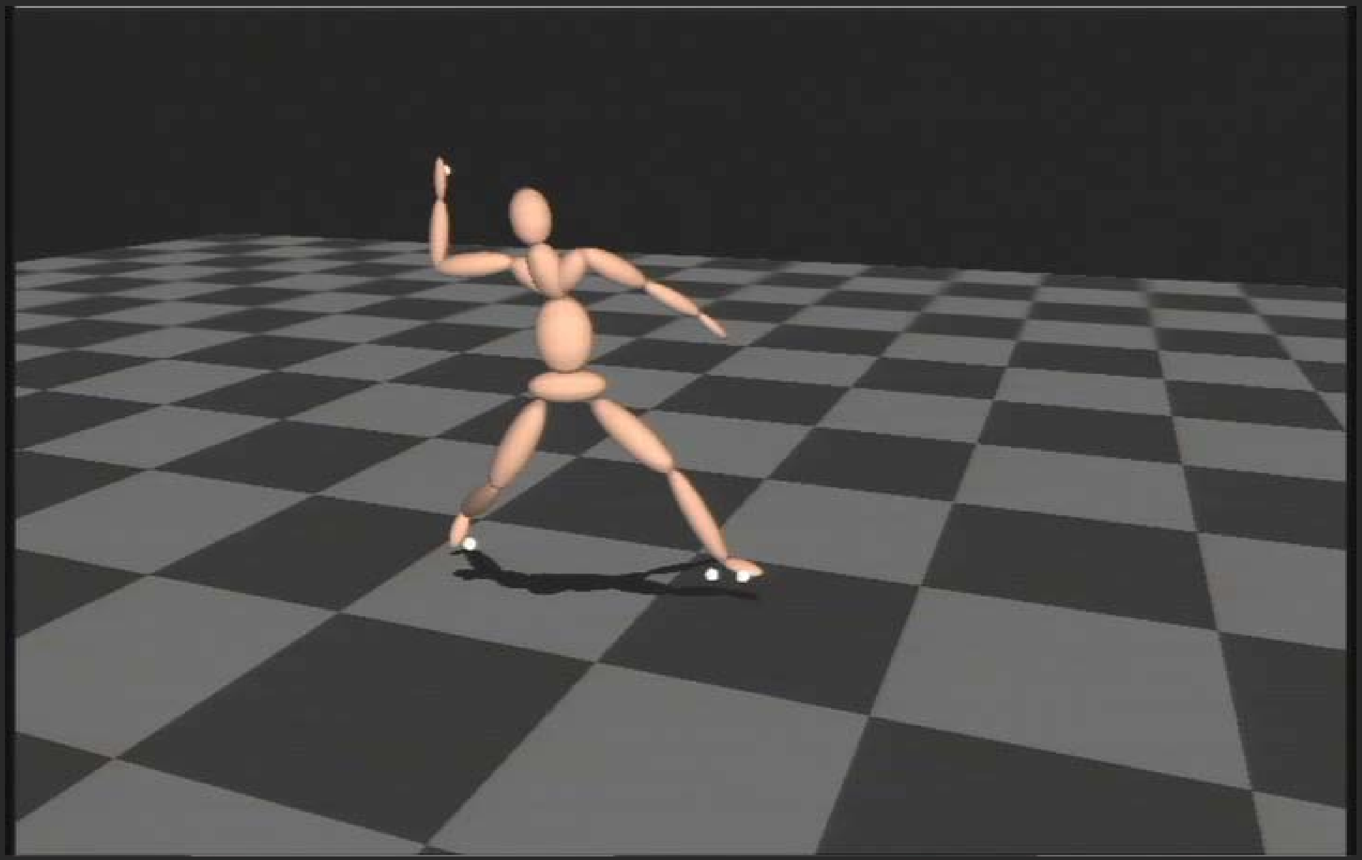
Multiple motion style



Style Interpolation



Trajectory Keyframing



Summary of Style IK

Pros:

- Arbitrary kinematic constraints
- Minimal parameter tuning

Cons:

- Weak temporal model
- Some optimization issues, particularly with large datasets
- Purely kinematic (no physical model)

Learning Biomechanical Models of Human Motion

Aaron Hertzmann
University of Toronto

with: Karen Liu, Zoran Popović
University of Washington

What determines how we move?

Individual Style:

Biology

Physics

Intention

Emotion

…

Can we build realistic and accurate models?

Goals for the model

1. Physically plausible

2. Generative models of human motion

    - Predictive, synthesis-quality

3. Generalize from a small dataset

Biomechanical principles

Optimality Theory

Hypothesis:

We optimize for efficiency, both in our body structure and movement

Criticisms of Optimality Theory

- We're not really globally optimal
  - The objective function constantly changes
  - We never really converge
  - We may be locally optimal
- Non-optimal variation, e.g., genetic drift
- Hard to build realistic models

Optimality Theory

This is controversial among biologists

Use optimization to test our assumptions about organisms

"Optimization is the only approach biology has for making predictions from first principles."

– W. Sutherland, *Nature*, June 2005

How do you walk?

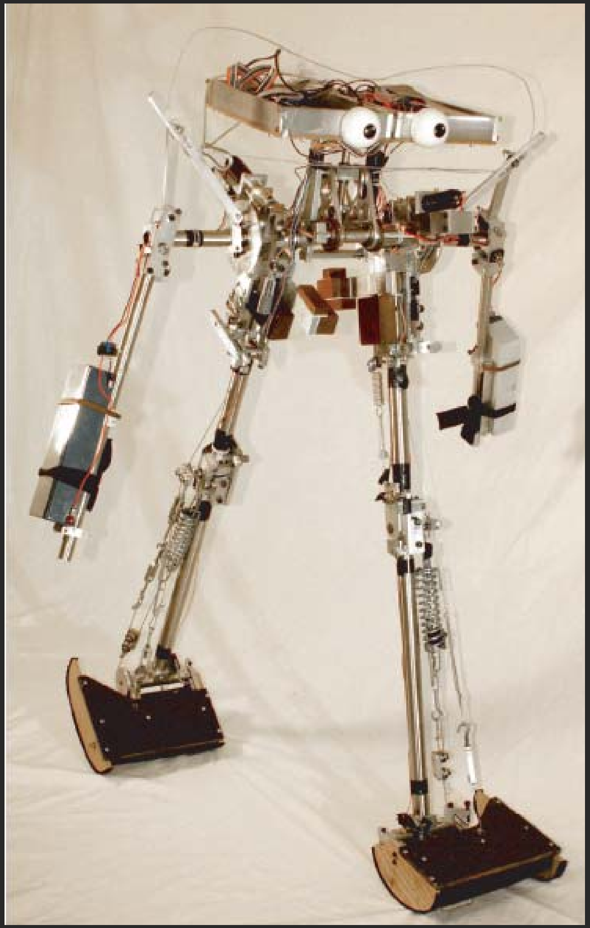
Not like this:

All joints actively actuated

Passive Dynamic Walking

Walking on a ramp without any muscles

Walking robots

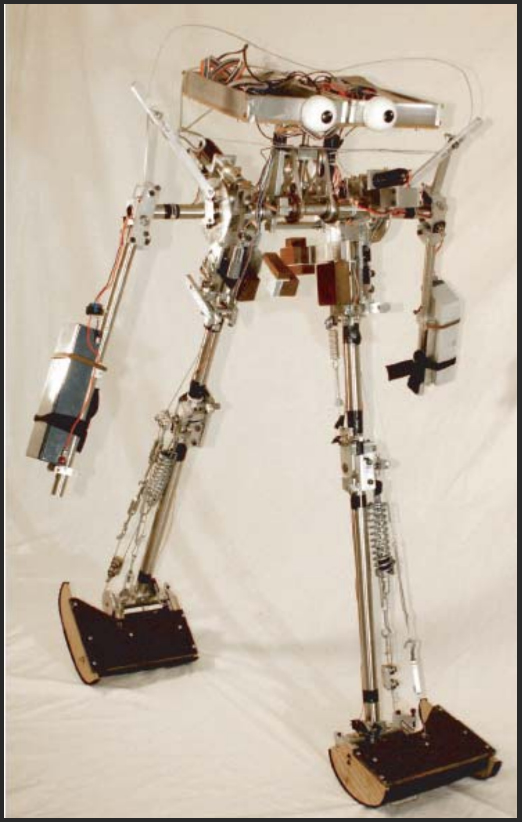


Collins et al, *Science* 2005

Efficiency of walking

DCT: 1.6

DCT: 0.05

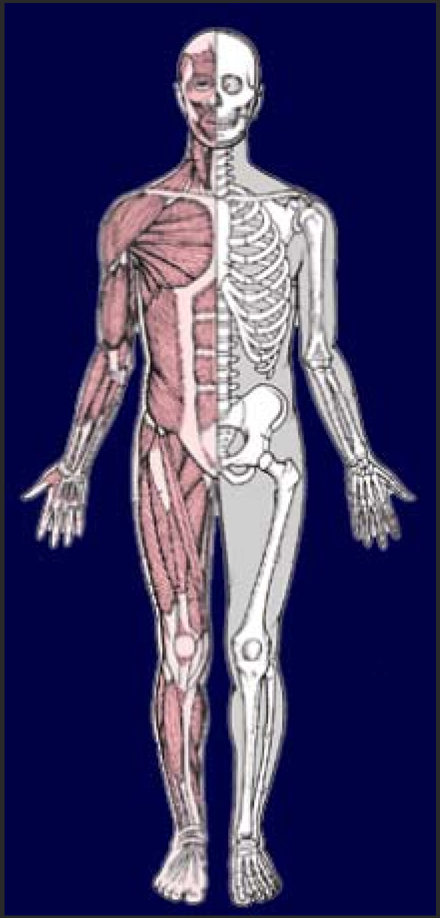


DCT: 0.055

Dimensionless Cost of Transport = Energy cost / (Body Weight * Distance)

Musculoskeletal structure

- Agonist/antagonist muscles
- Passive properties of muscles and tendons:
  - Elastic
  - Damping



Muscle stiffness and springiness

- Stiffness improves stability/robustness
  - Muscle stiffness varies for different tasks
- Passive elements help conserve energy
  - saves 20-30% of energy during running

Relative muscle preferences

- Some muscles are stronger than others
- Some muscles are more efficient
  - muscle attaches to bone in different ways
- Some body parts may be more prone to damage

Physics-based motion style

Body parameterization

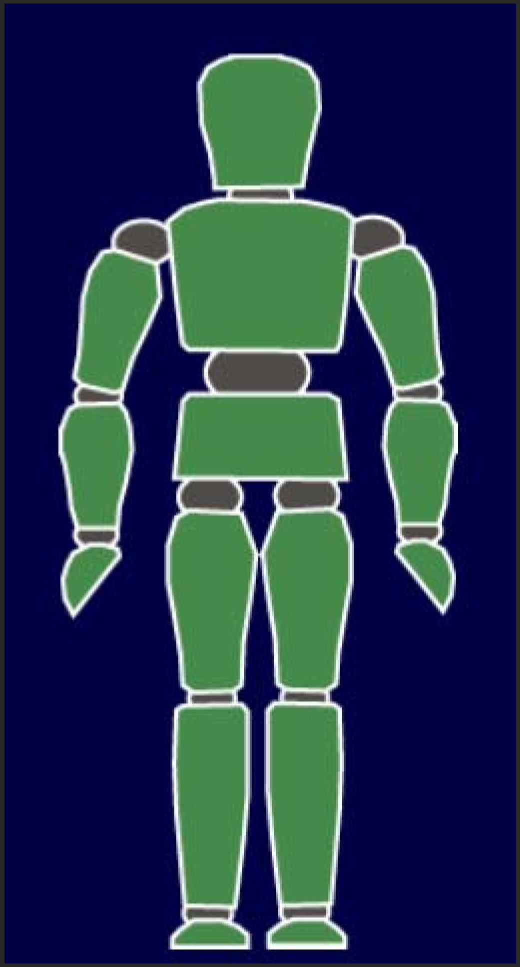
18 body nodes

Pose at time t: $\mathbf{q}_t$

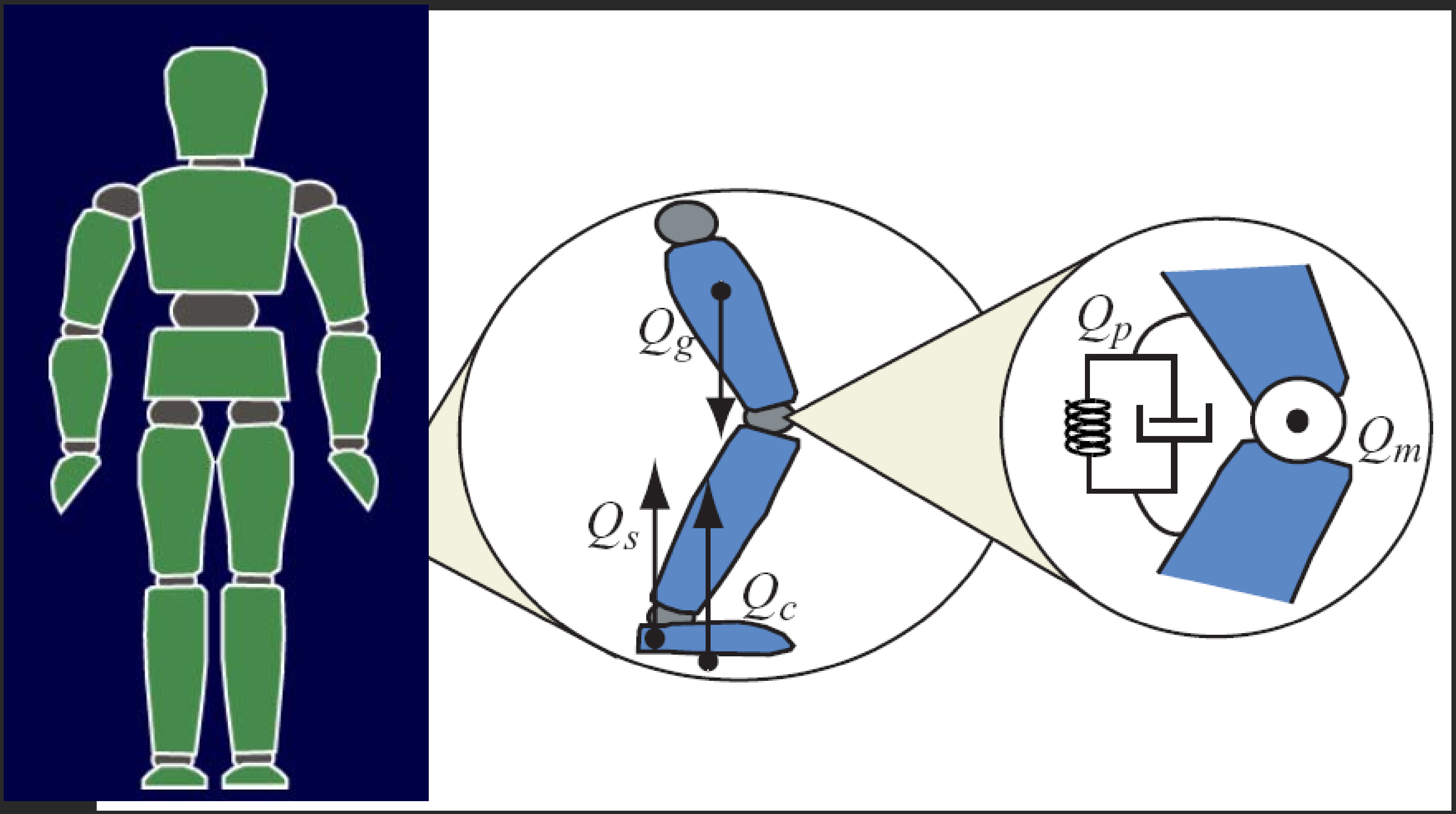
- Root pos./orientation (6 DOFs)
- Joint angles (29 DOFs)
- Using exponential maps

Motion

- $\mathbf{X} = [\mathbf{q}_1, \ldots, \mathbf{q}_T]$



Forces at a joint



Assumption: constant stiffness at each joint

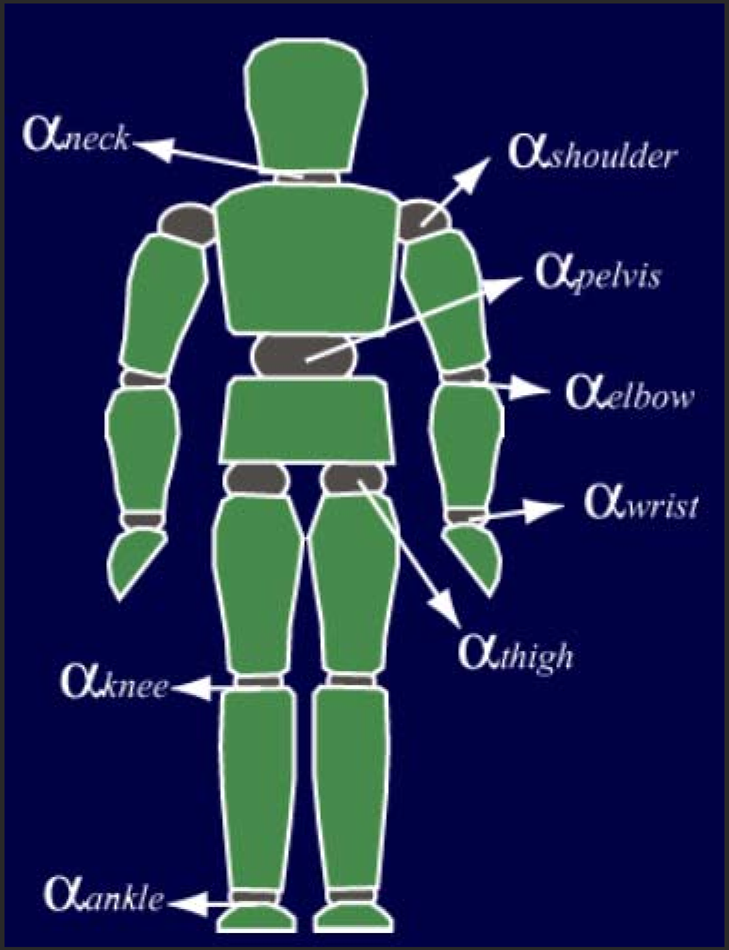
Equations of motion at a joint

(Considering only aggregate forces)

"F = ma":

$$\mathrm{tr}(d\mathbf{W}_i/d\mathbf{q}_j\ \mathbf{M}_i\ \mathbf{W}''^{T}_i) = Q_{mj} + Q_{gj} + Q_{pj} + Q_{cj} + Q_{sj}$$

Muscle preferences



Muscle preferences: α_j

Physical style

Parameter vector θ includes

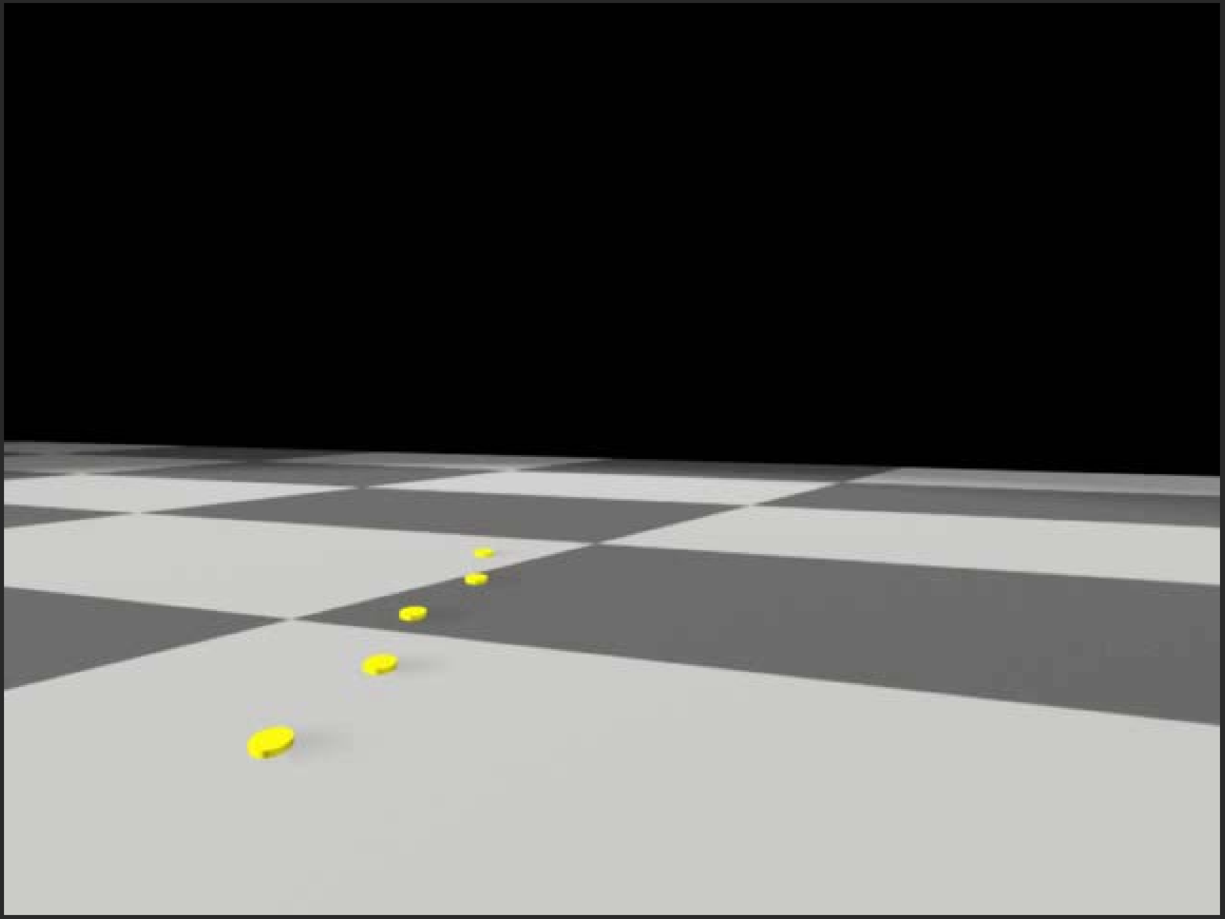
- spring constants k_{s1}, k_{s2} (two per DOF)
- rest angles $\bar{q}$ (one per DOF)
- shoe spring constants k_{shoe} (two)
- damping constants k_d (one for DOF)
- muscle preferences α_j (one per DOF)

Total: 147 dims. in θ

A vector θ defines a physical style

(Skeleton and masses from preprocessing)

Constraints on motion



Foot contact constraints **C**

Objective function for motion

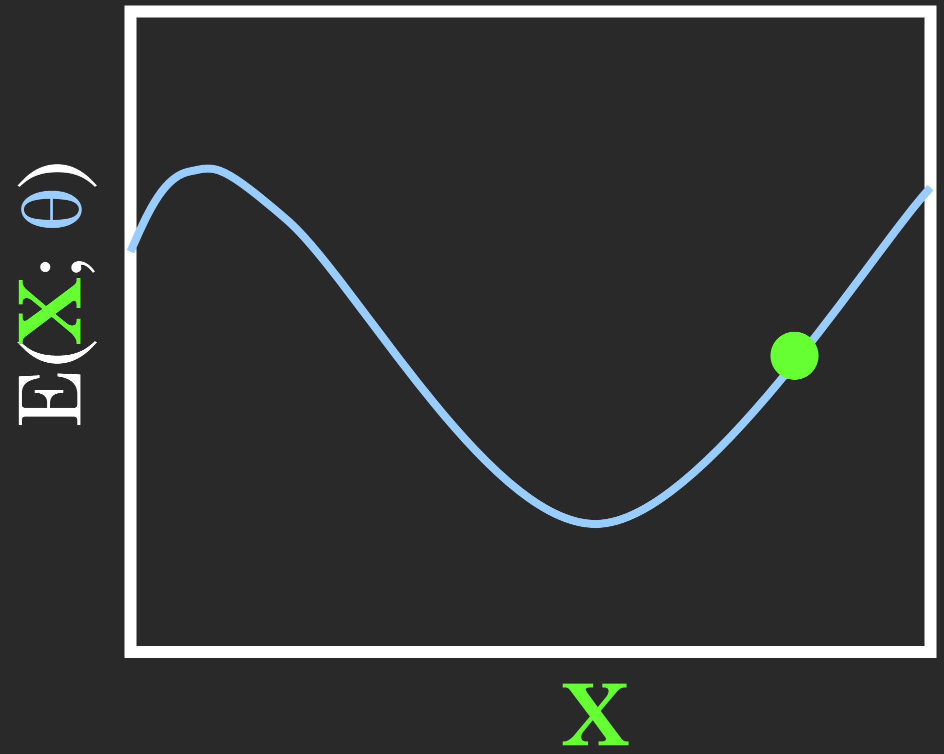
Weighted sum of magnitudes of all muscle forces:

$$E(\mathbf{X};\, \theta) = \sum_{j,t}\ \alpha_j \left(Q_{m,j}(t,\, \mathbf{X},\, \theta)\right)^2$$

$Q_{m,j}(t,X,\theta)$ determines muscle force at time t from X and θ (closed-form)

Generating motion

$$\mathbf{X}^* = \arg\min_{\mathbf{X} \in \mathbf{C}} \mathrm{E}(\mathbf{X}; \theta)$$



Problem

θ is 147-dimensional

Impossible to tune by hand

How can we acquire it from data?

Nonlinear Inverse Optimization

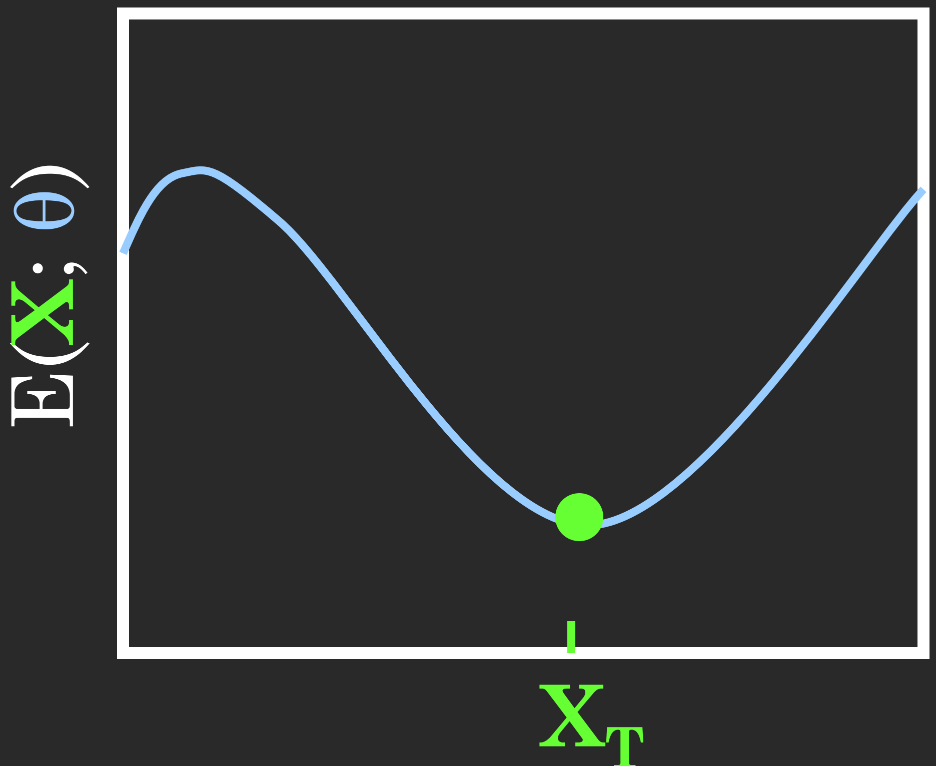
Problem statement

Given an optimal motion, what was the energy function?

Given

- mocap $\mathbf{X}_T$
- constraints $\mathbf{C}$

Determine θ



What doesn't work

Least-squares

- $|| \mathbf{X}_T - \arg\min E(\mathbf{X}; \theta) ||^2$
- not robust
- very hard to optimize

Maximum likelihood

- Gibbs distribution: $p(\mathbf{X} | \theta, \mathbf{C}) = e^{-E(\mathbf{X}; \theta, \mathbf{C})}/Z(\theta, \mathbf{C})$
- Intractable
- … even with Contrastive Divergence

Idea

Goal:

$$E(\mathbf{X}_T; \theta) = \min_{\mathbf{X} \in \mathbf{C}} E(\mathbf{X}; \theta)$$

Learning objective function:

$$G(\theta) = E(\mathbf{X}_T; \theta) - \min_{\mathbf{X} \in \mathbf{C}} E(\mathbf{X}; \theta)$$

Constraints: $\sum_j \alpha_j = 1, \alpha_j >= 0$

– to prevent E=0 everywhere

Learning

$$G(\theta) = E(\mathbf{X}_T; \theta) - \min_{\mathbf{X} \in C} E(\mathbf{X}; \theta)$$

$$\mathbf{X}_S = \arg\min_{\mathbf{X} \in C} E(\mathbf{X}; \theta)$$

$$H(\theta) = E(\mathbf{X}_T; \theta) - E(\mathbf{X}_S; \theta)$$

$$dG/d\theta \approx dH/d\theta$$

$$= dE/d\theta\,|_{\mathbf{X}_T} - dE/d\theta\,|_{\mathbf{X}_S}$$

Learning

initialize θ

while not done do

$\quad X_S = \arg\min_{X \in C} E(\mathbf{X}; \theta)$

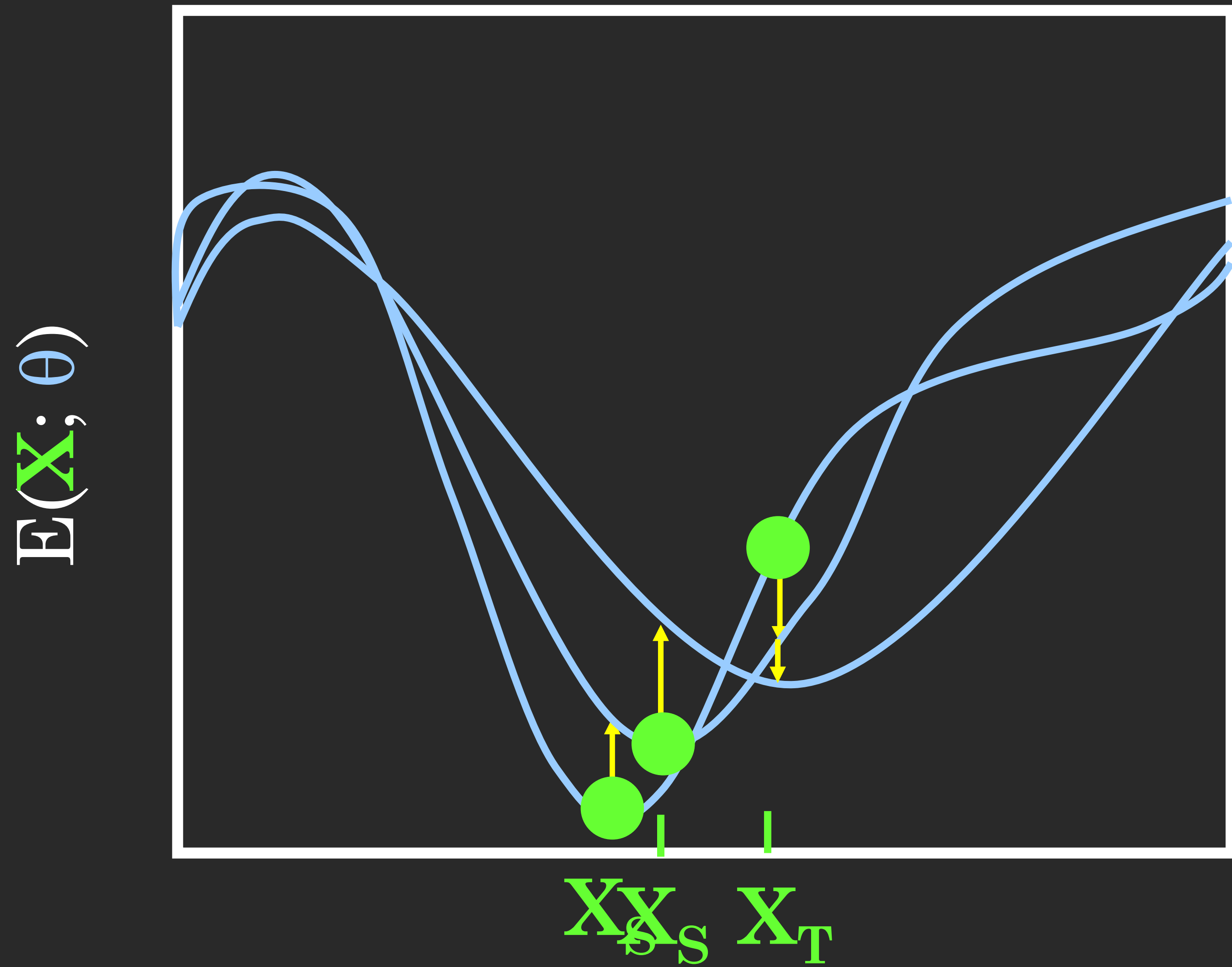
$\quad \Delta\theta = dE/d\theta\,|_{\mathbf{X}_T} - dE/d\theta\,|_{\mathbf{X}_S}$

$\quad \beta = \arg\min_{\beta} G(\theta - \beta\Delta\theta)$

$\quad \theta = \theta - \beta\,\Delta\theta$

Learning

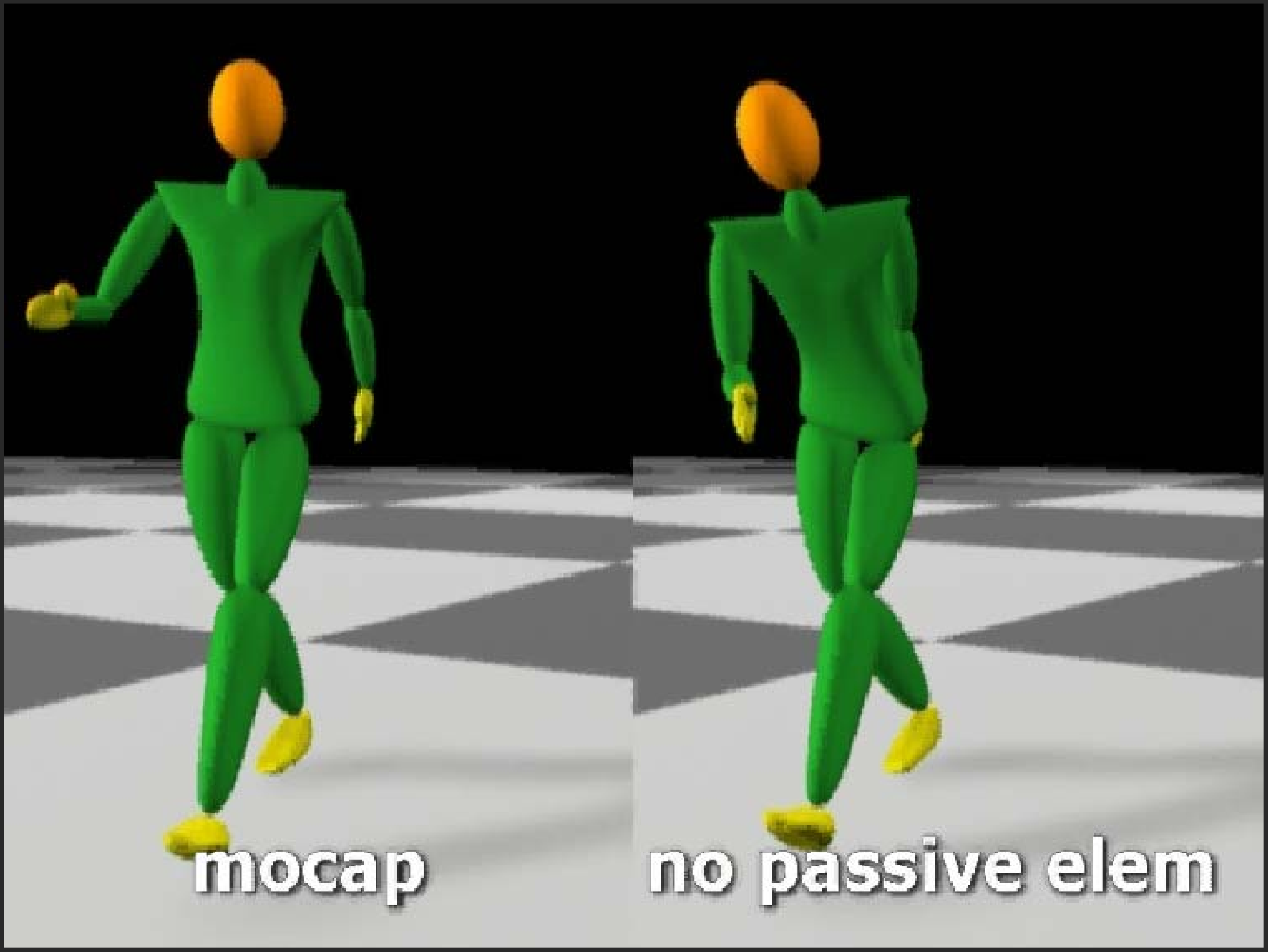
$X_S = \arg\min_{X \in C} E(\mathbf{X}; \theta)$ $\Delta\theta \approx dE/d\theta\,|_{X_T} - dE/d\theta\,|_{X_S}$



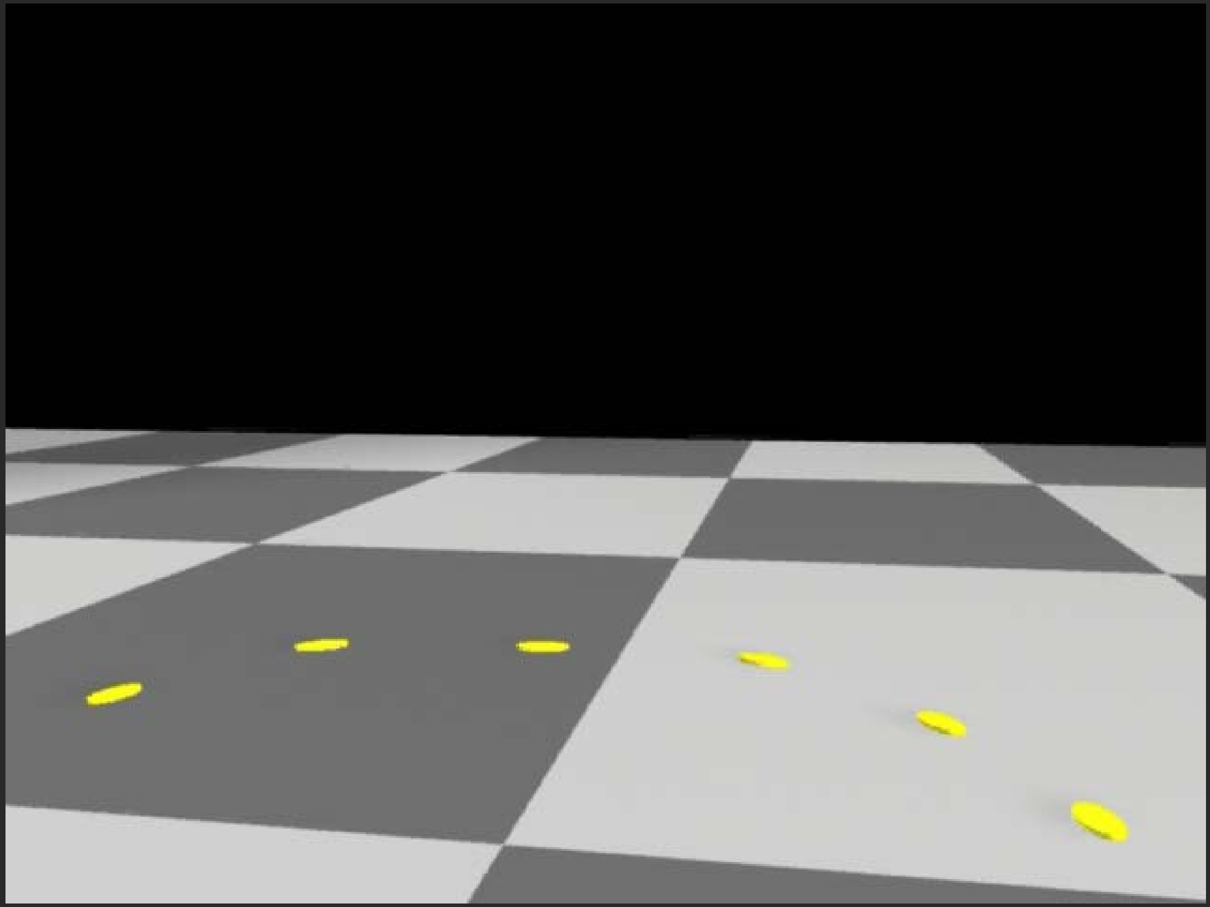
Nonlinear Inverse Optimization

- Solve for optimization parameters for any differentiable energy function
- Works with hard constraints
- All you need is a forward optimizer
- Degenerate form of CD
- Related to energy-based models of [LeCun and Huang 2005]

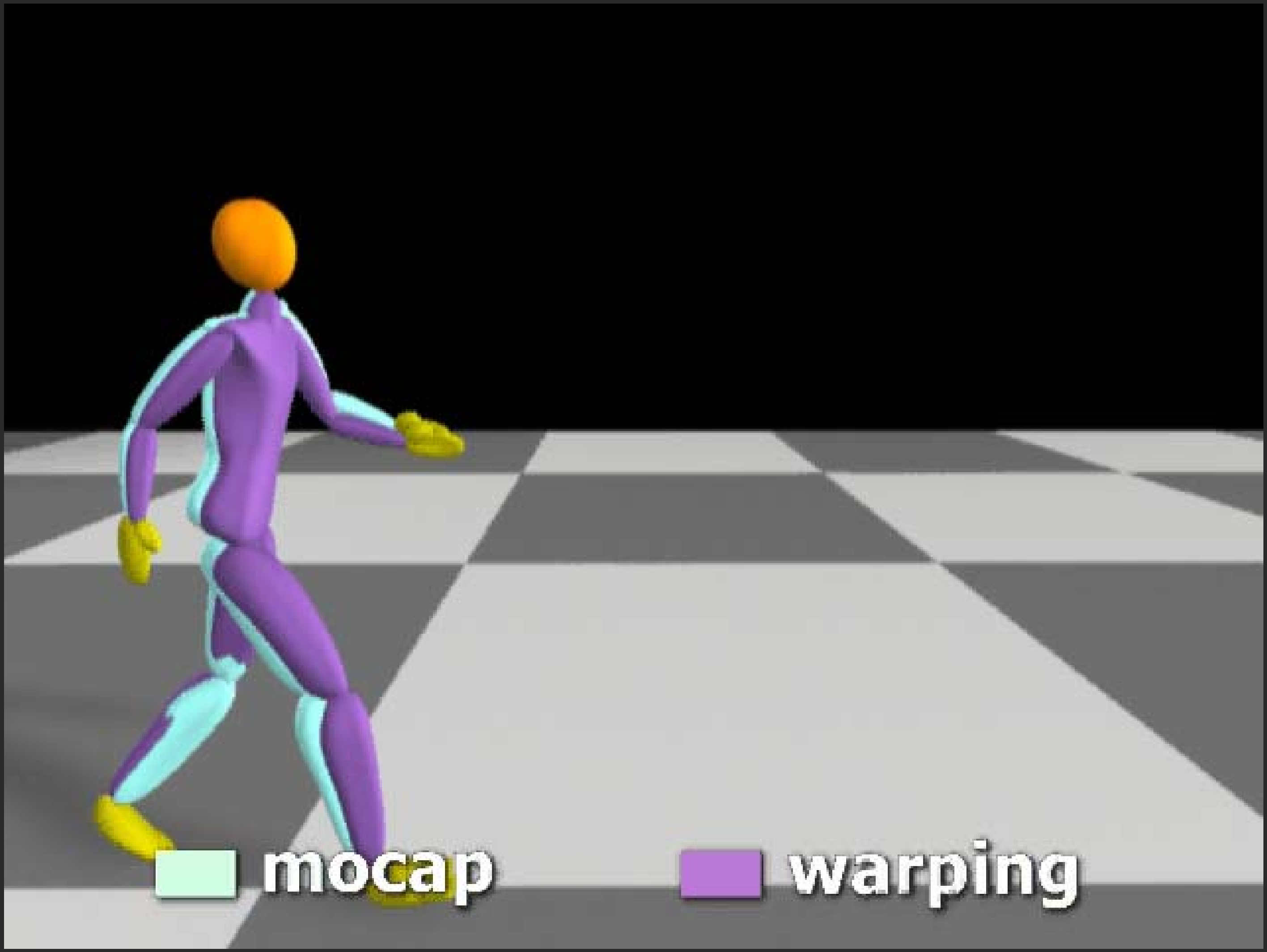
A basic walk



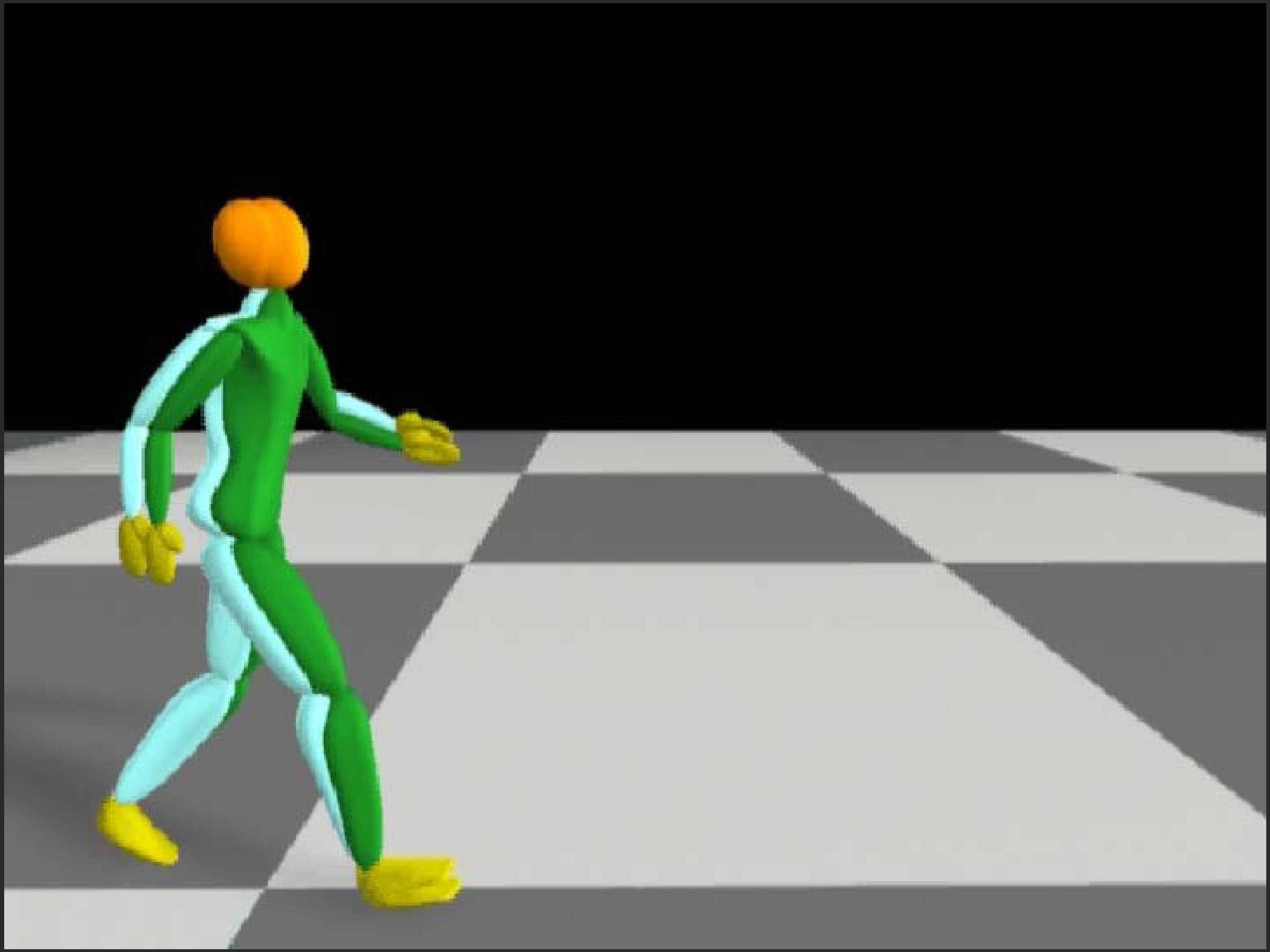
New constraints



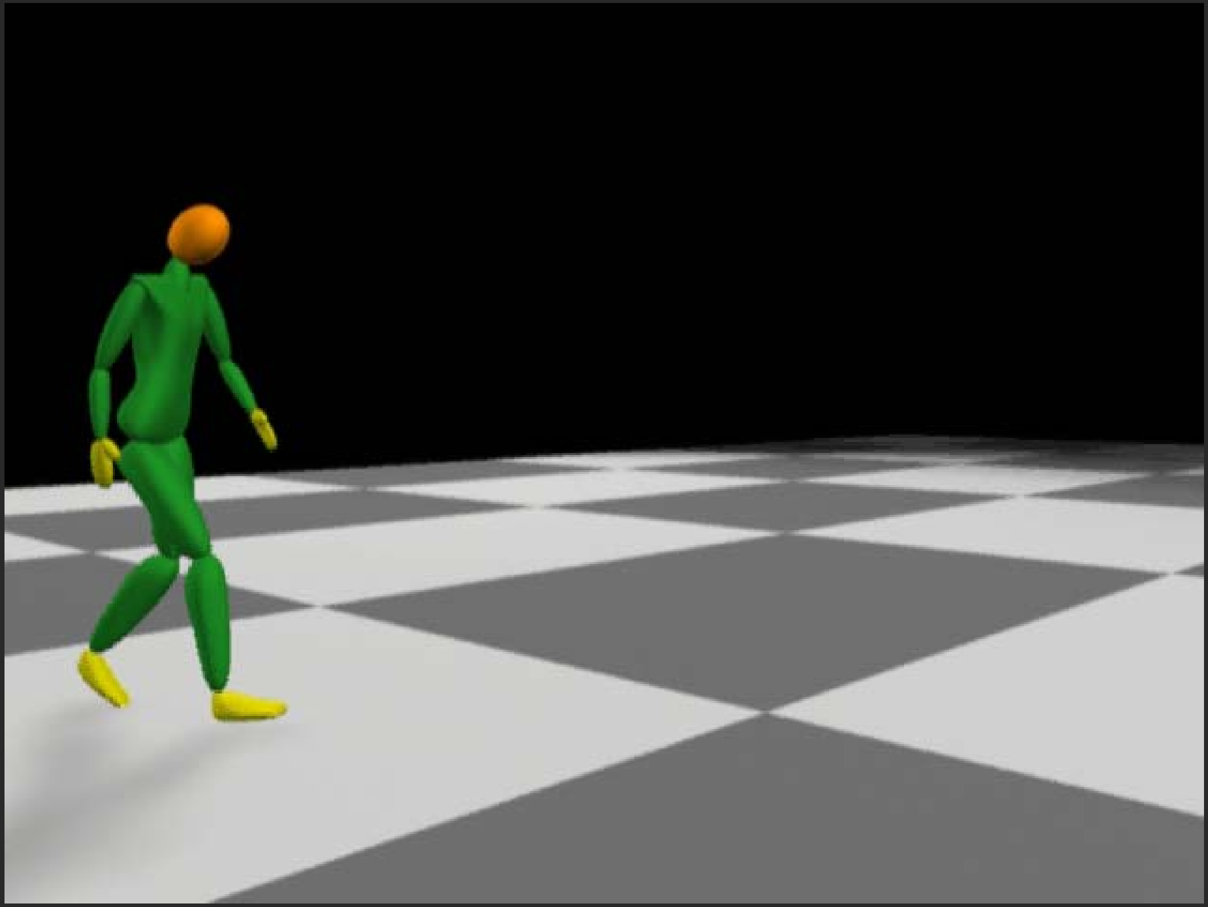
Warping vs. ground truth



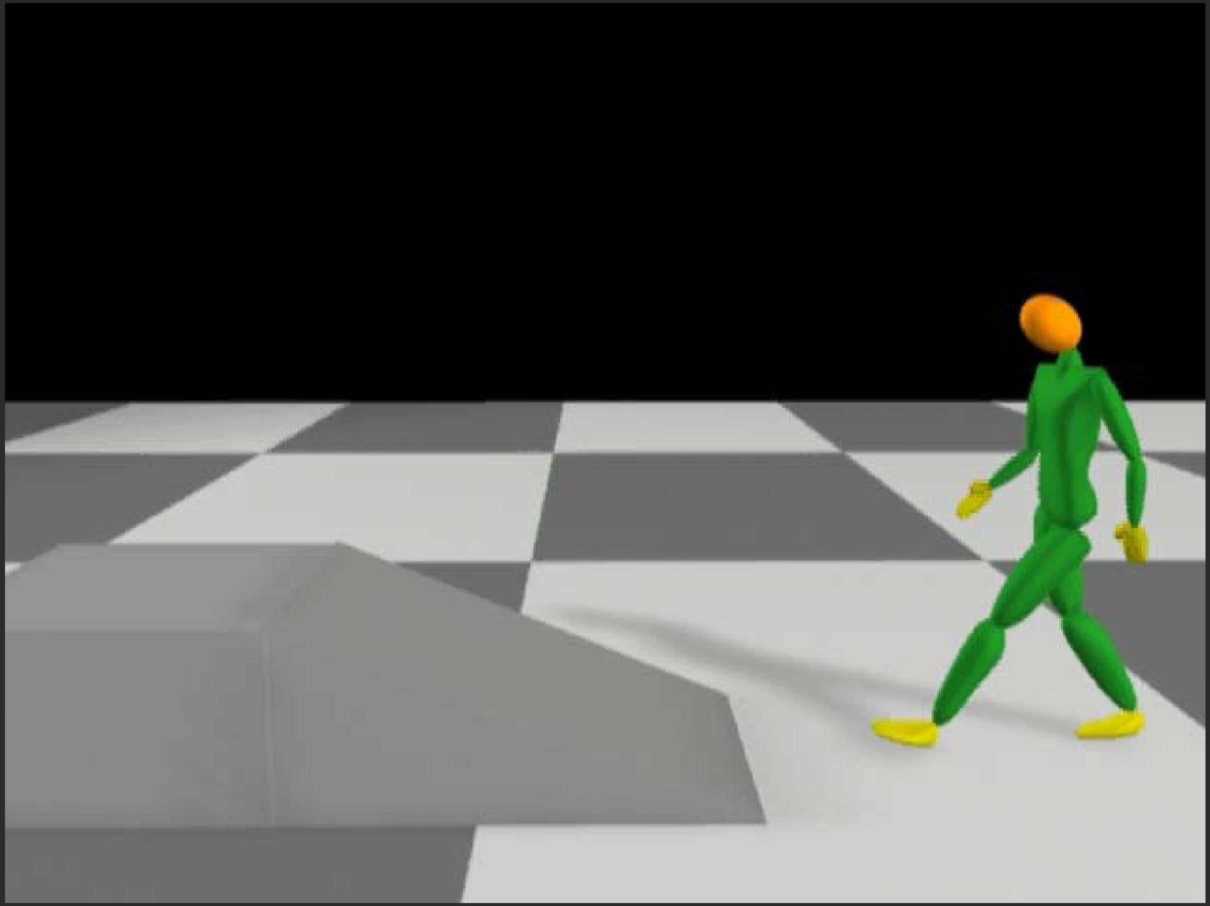
Comparison to mocap



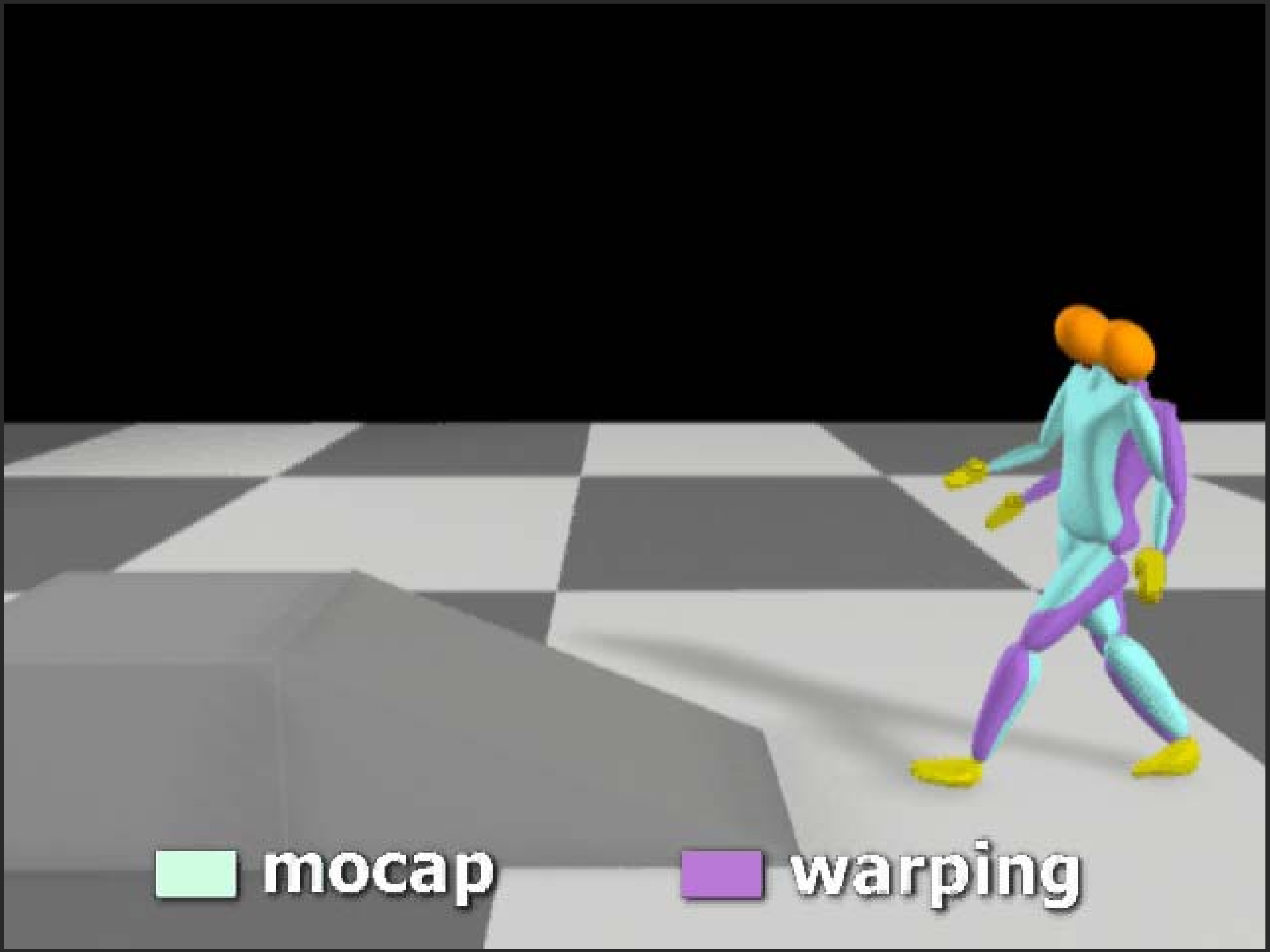
A "sad" style



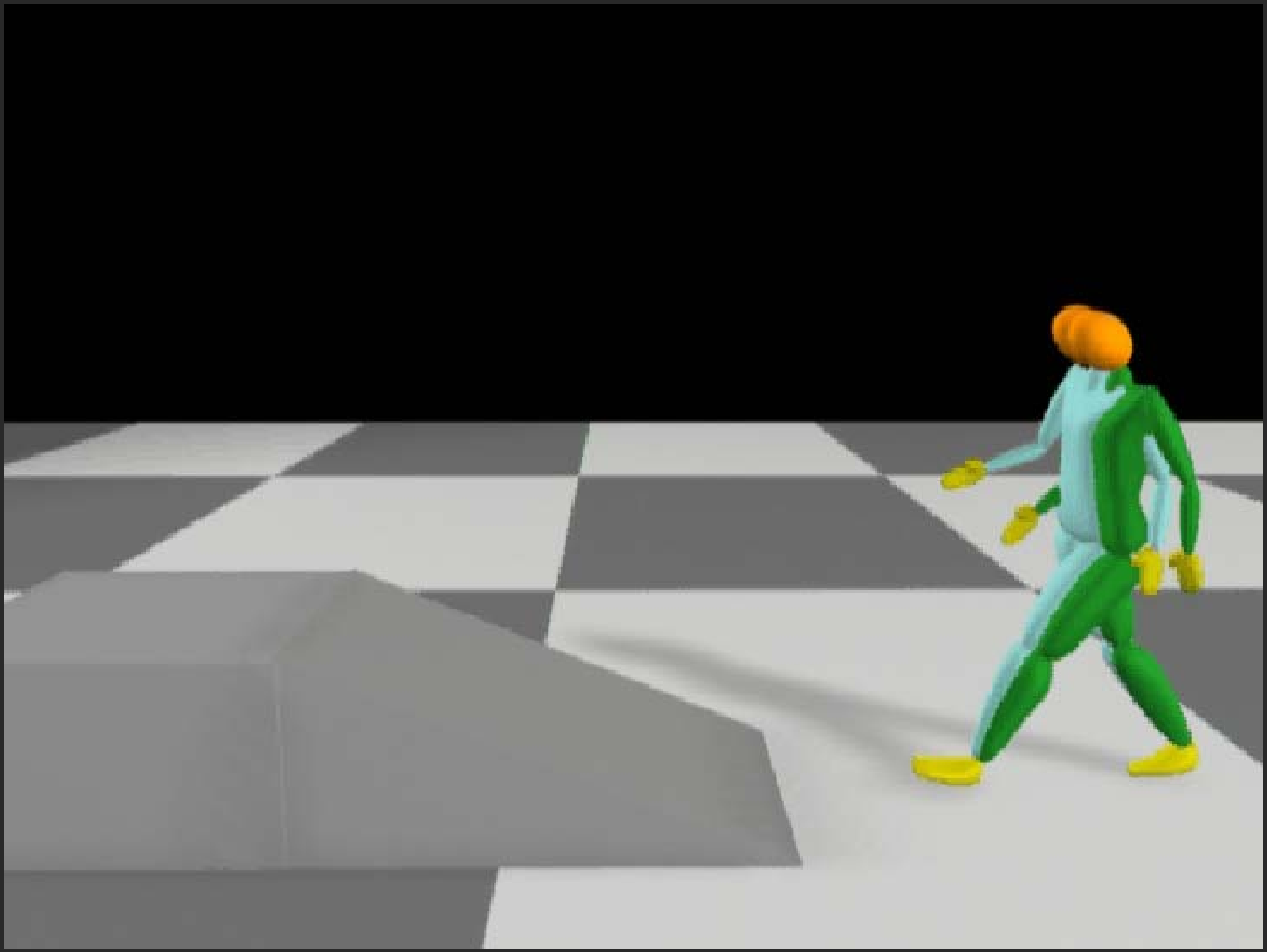
Walking uphill



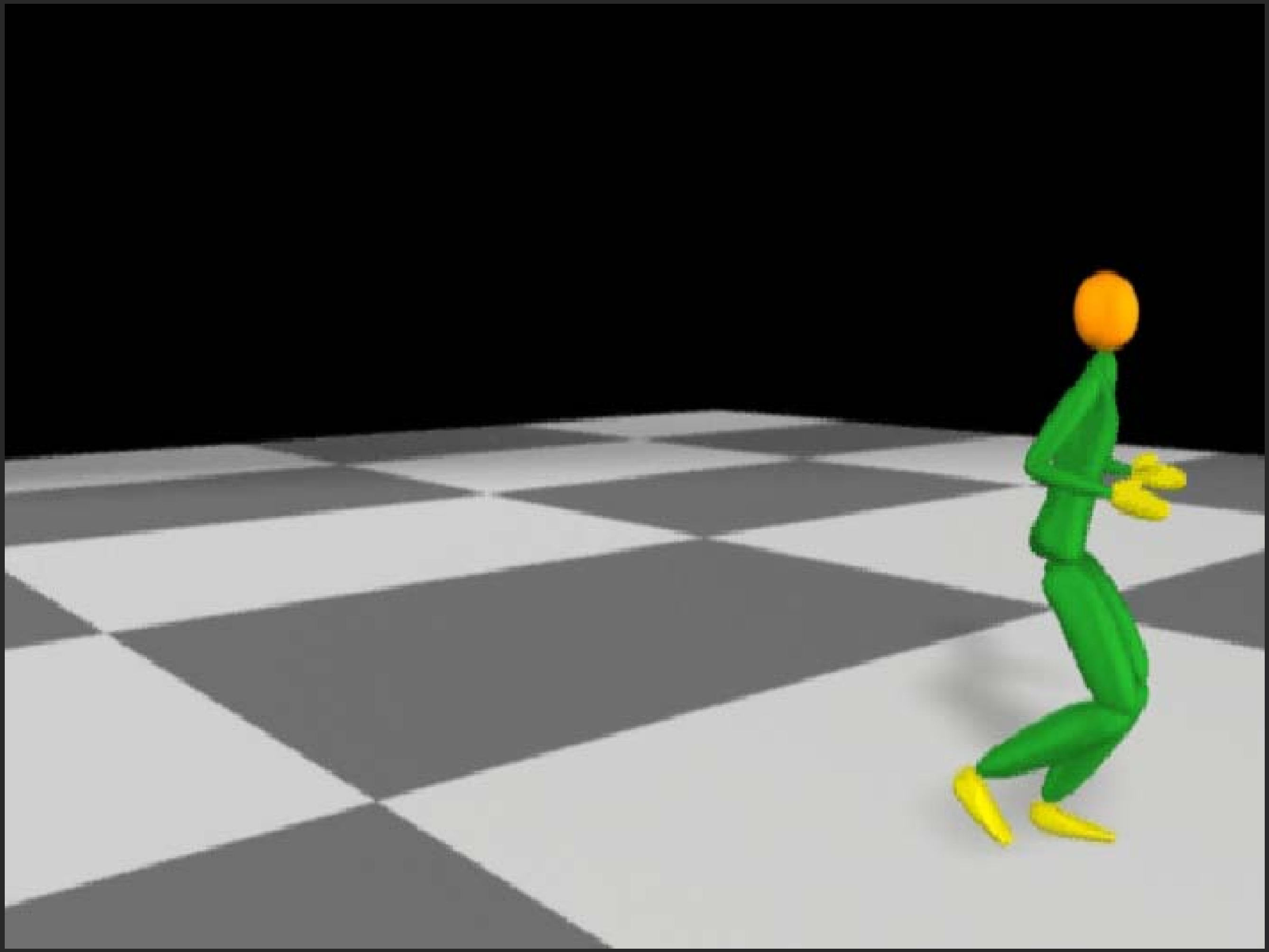
Motion warping vs. ground truth



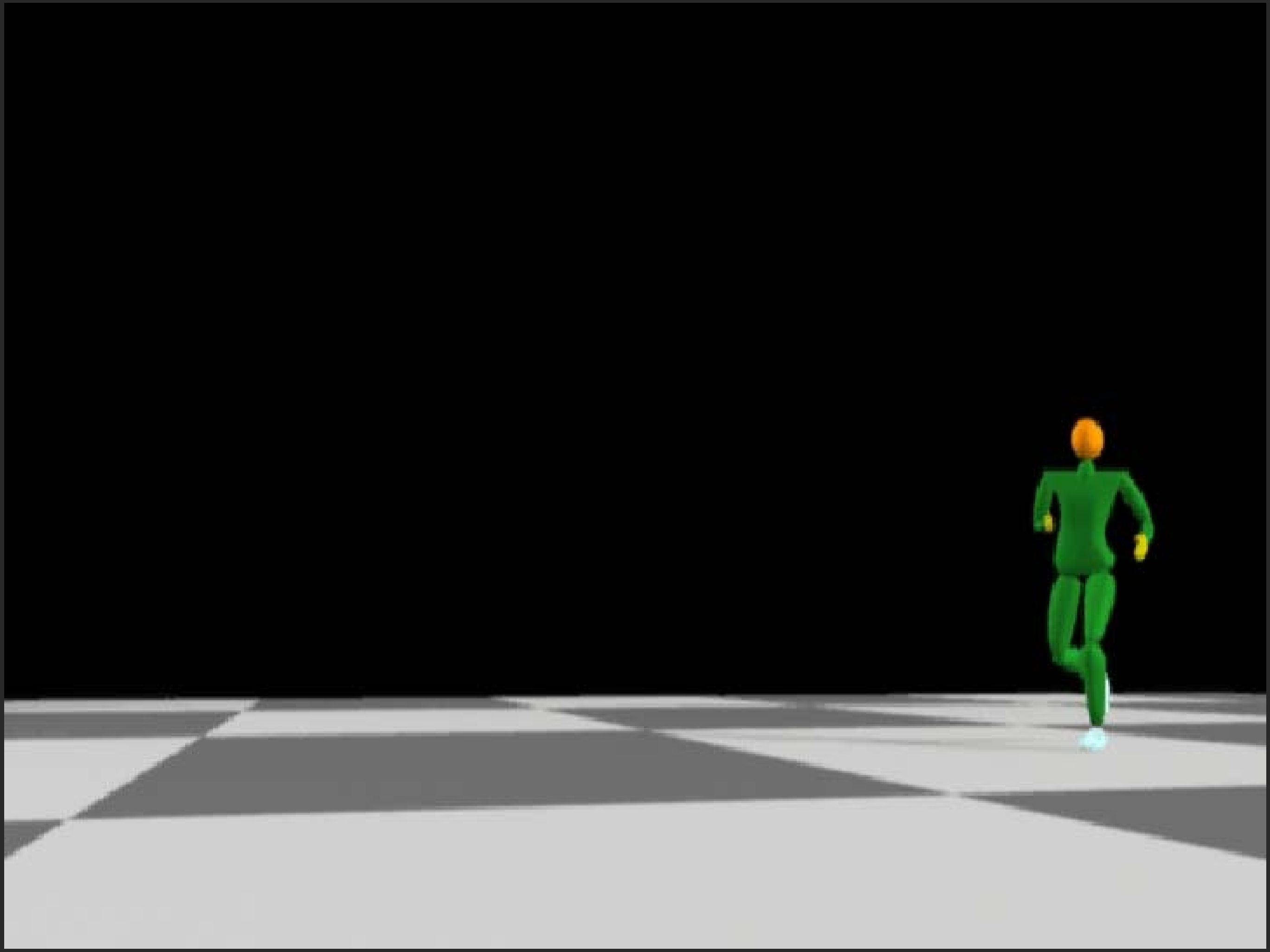
Comparison to mocap



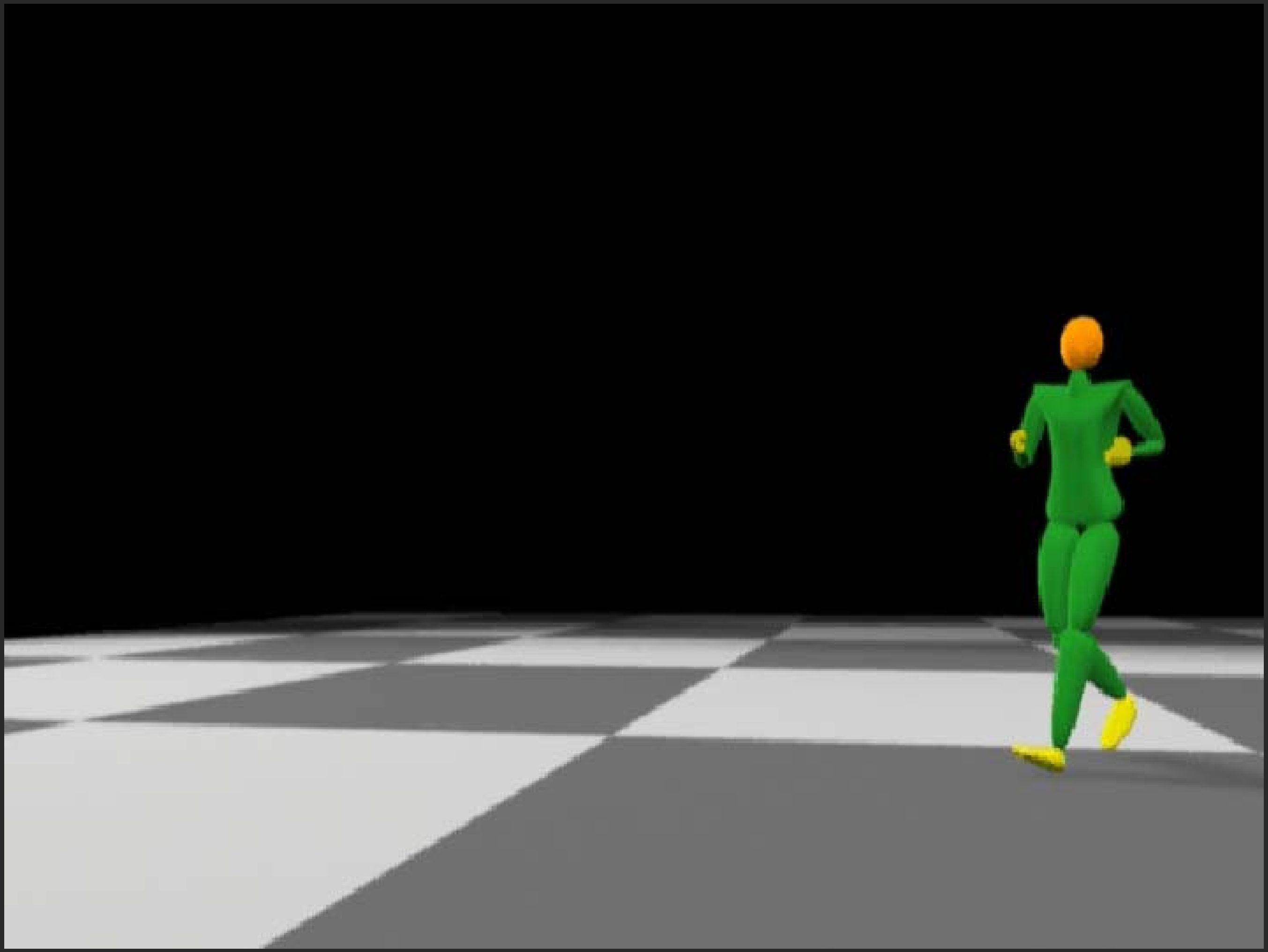
Running



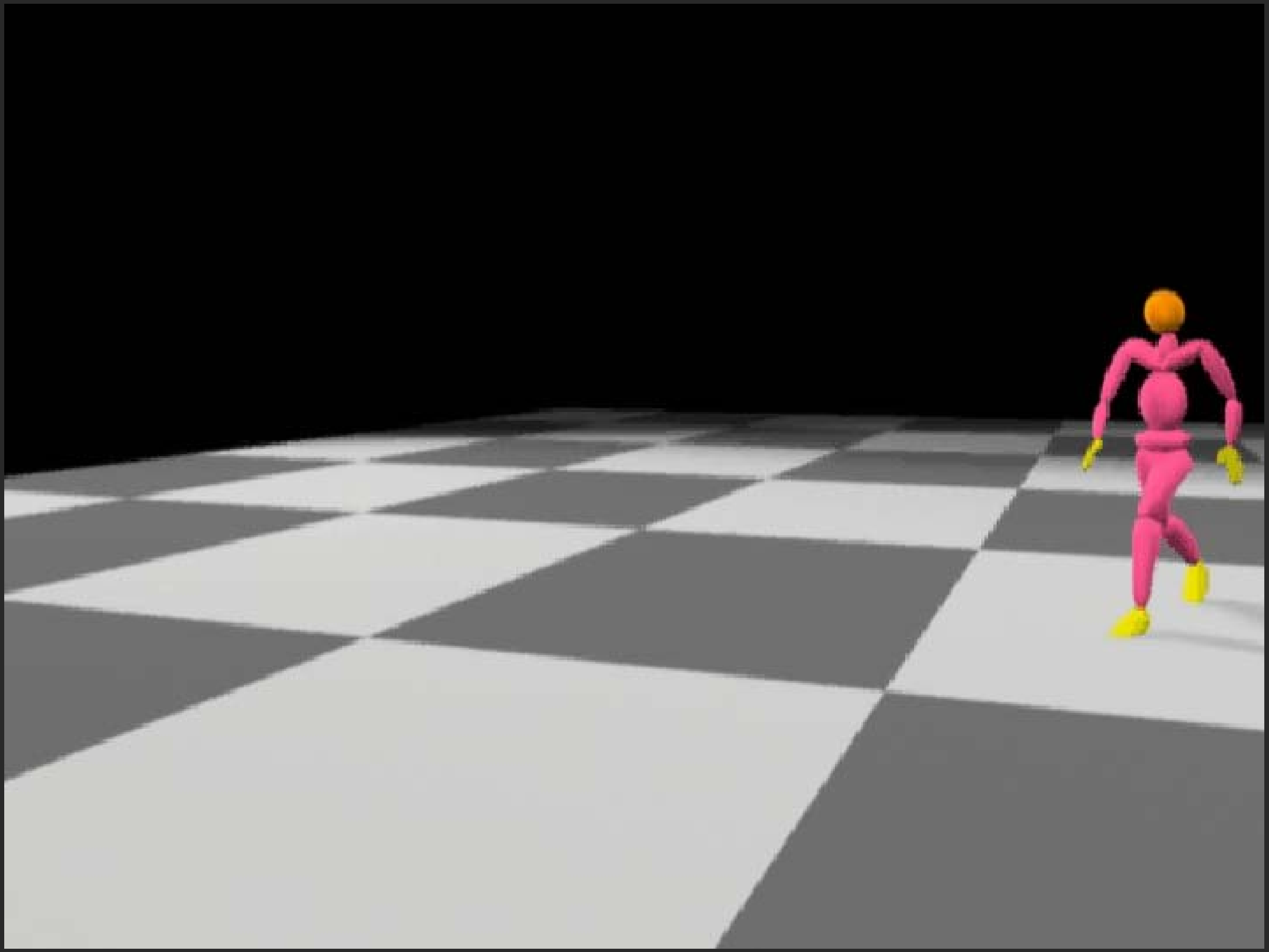
Springier shoes

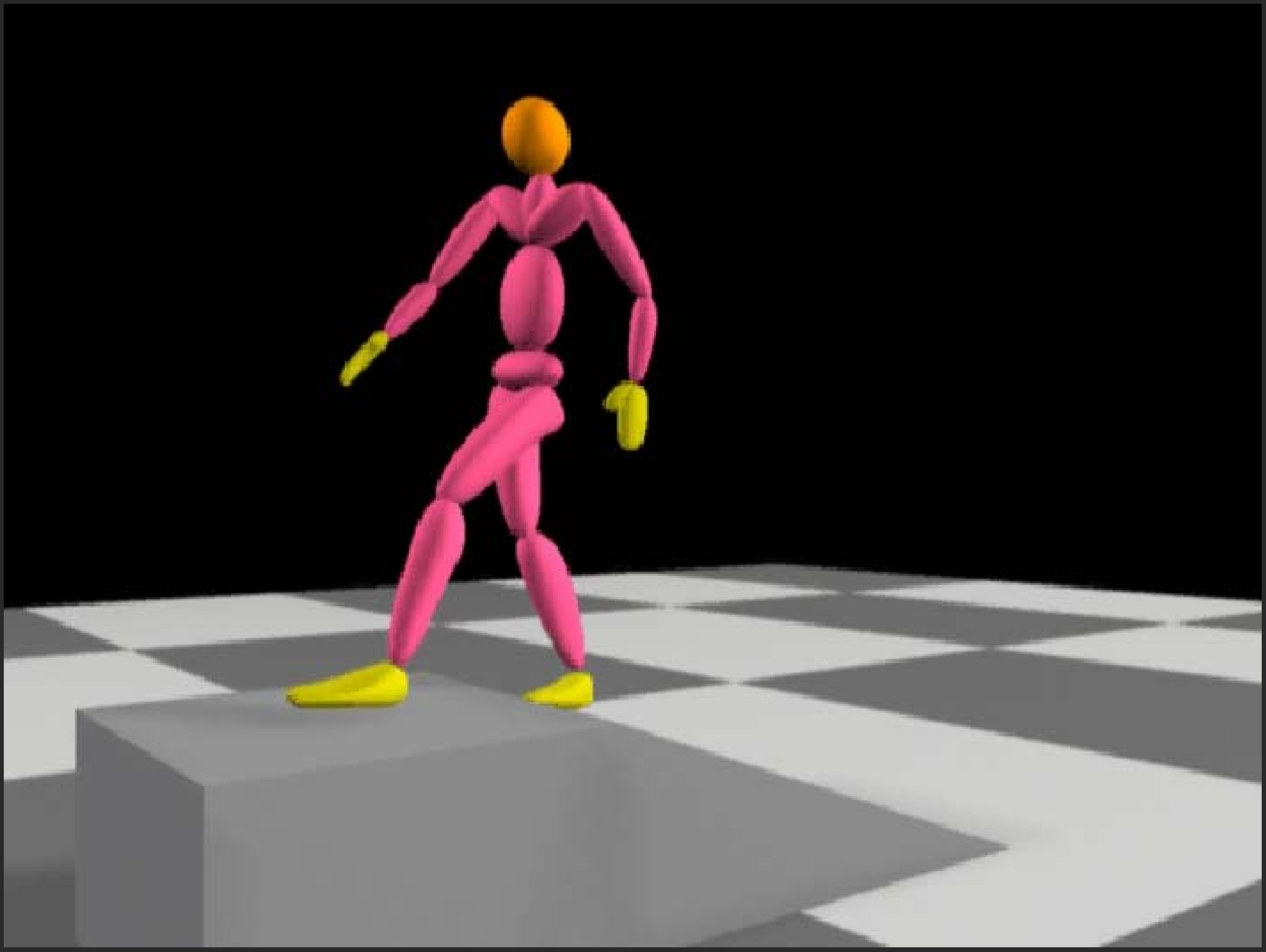


“Powerwalking”

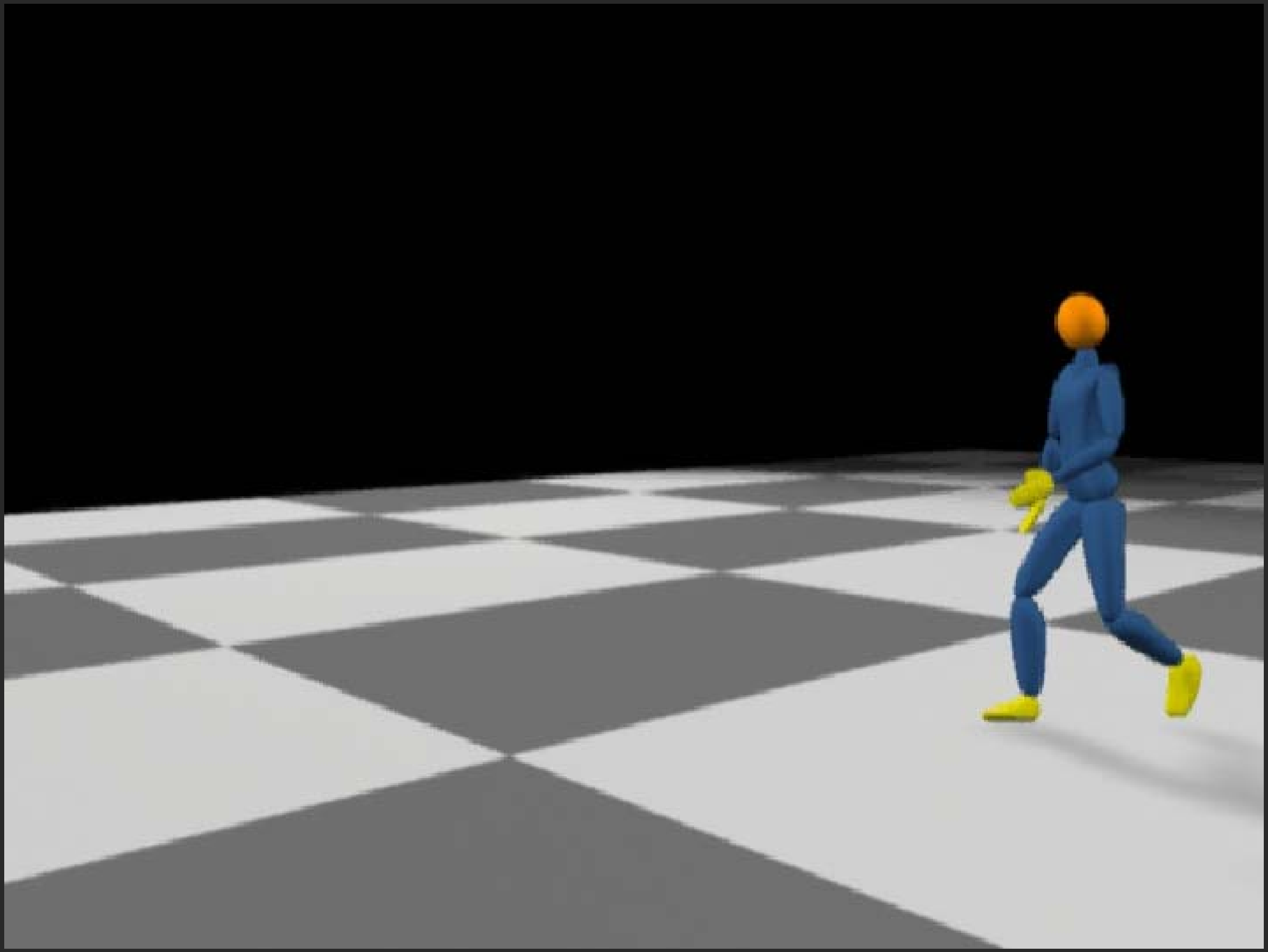


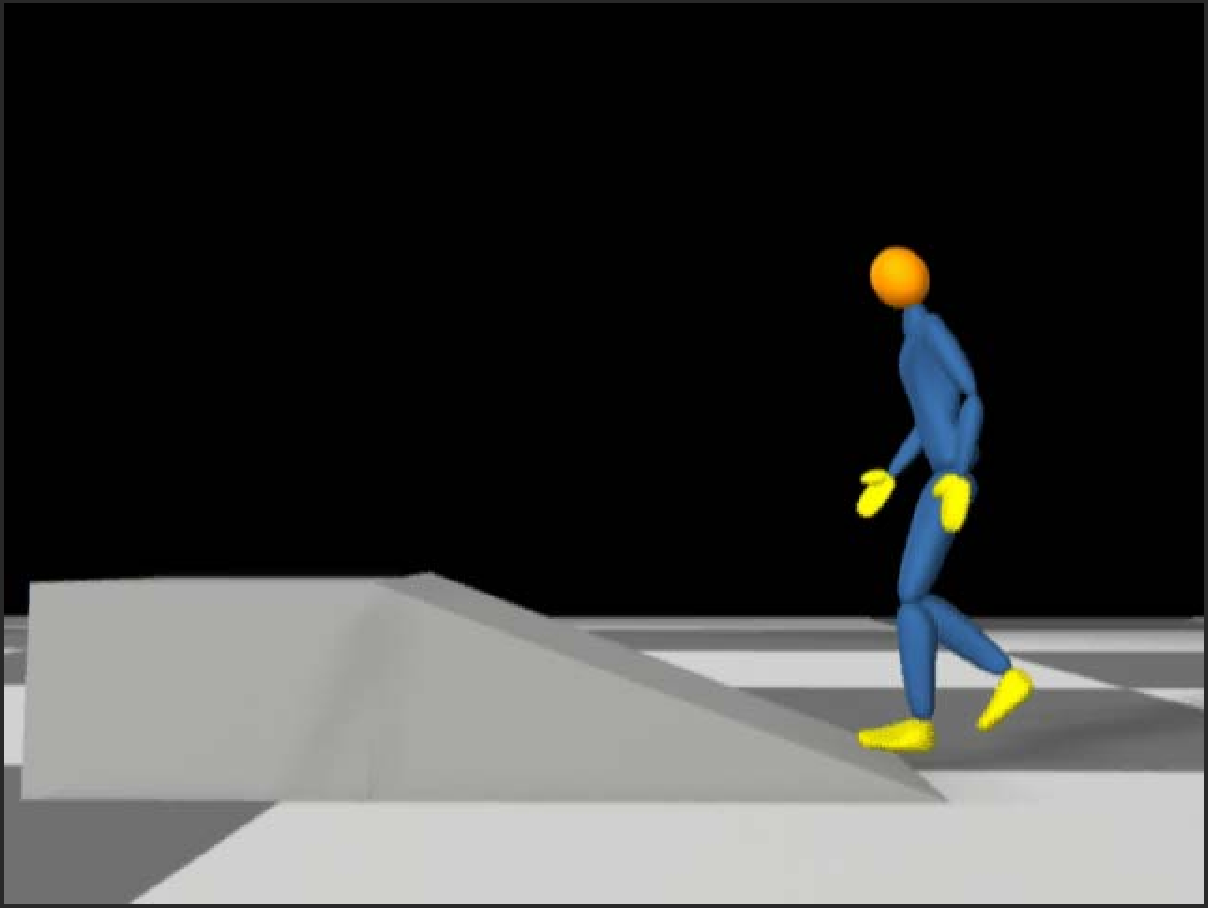
A different subject





Another subject





Summary of physics-based style

Pros:

– Generalizes to new physical situations

– Small training sets

Cons:

– expensive optimizations

– physical model is incomplete (so far)

Future work:

– does it work for broader classes of motions and styles?

– model control, space of styles, etc.

Summary of animation

Motion graphs

- simple easy fast
- can't generate new poses at all

Probabilistic kinematic models

- not quite as fast
- no physics

Physics-based style

- very slow
- potentially, very powerful