Humans:
The Final Frontier

source: http://www.gimartex.es/myfiles/Ballet-dancer_01.jpg

Adrien Treuille

Overview

- State of the art.
- Body models.
- Animation
- Questions

source: 3dscience.com

Overview

- **State of the art.**
- Body models.
- Animation
- Questions

source: 3dscience.com

State of the Art

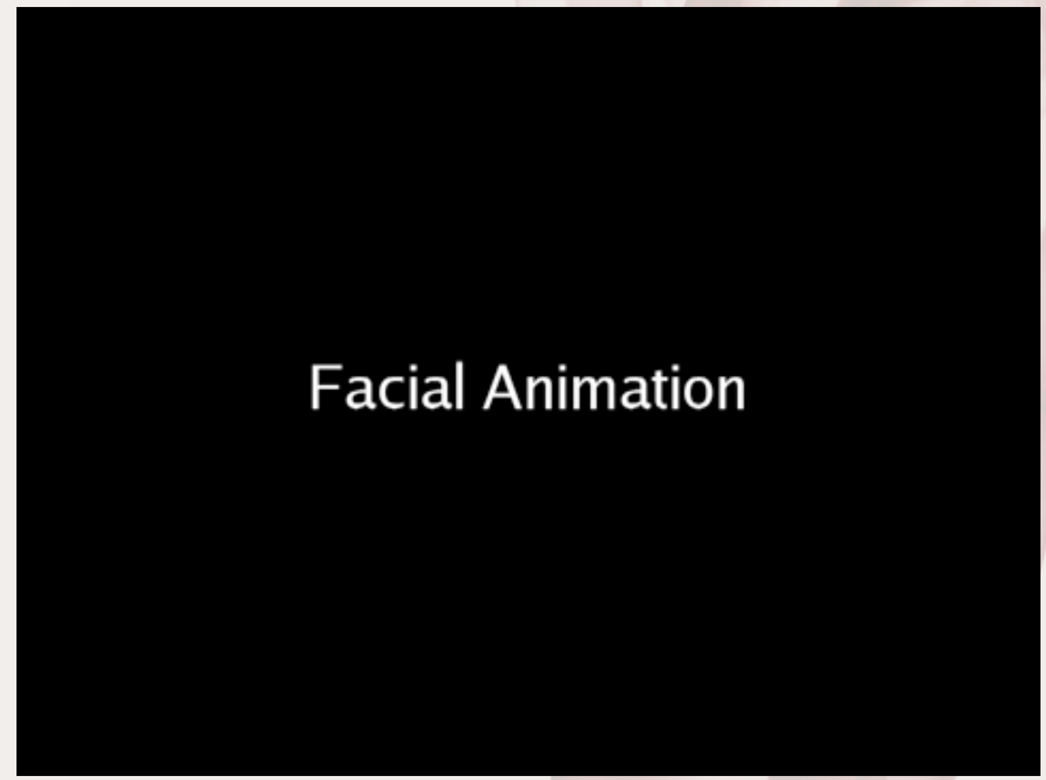
- **Steady but *slow* progress towards digital humans.**

State of the Art

- **Steady but *slow* progress towards digital humans.**
  - **(As usual, rendering ahead of animation.)**

State of the Art

- **Steady but *slow* progress towards digital humans.**
  - **(As usual, rendering ahead of animation.)**
- **State of the art for animation production.**

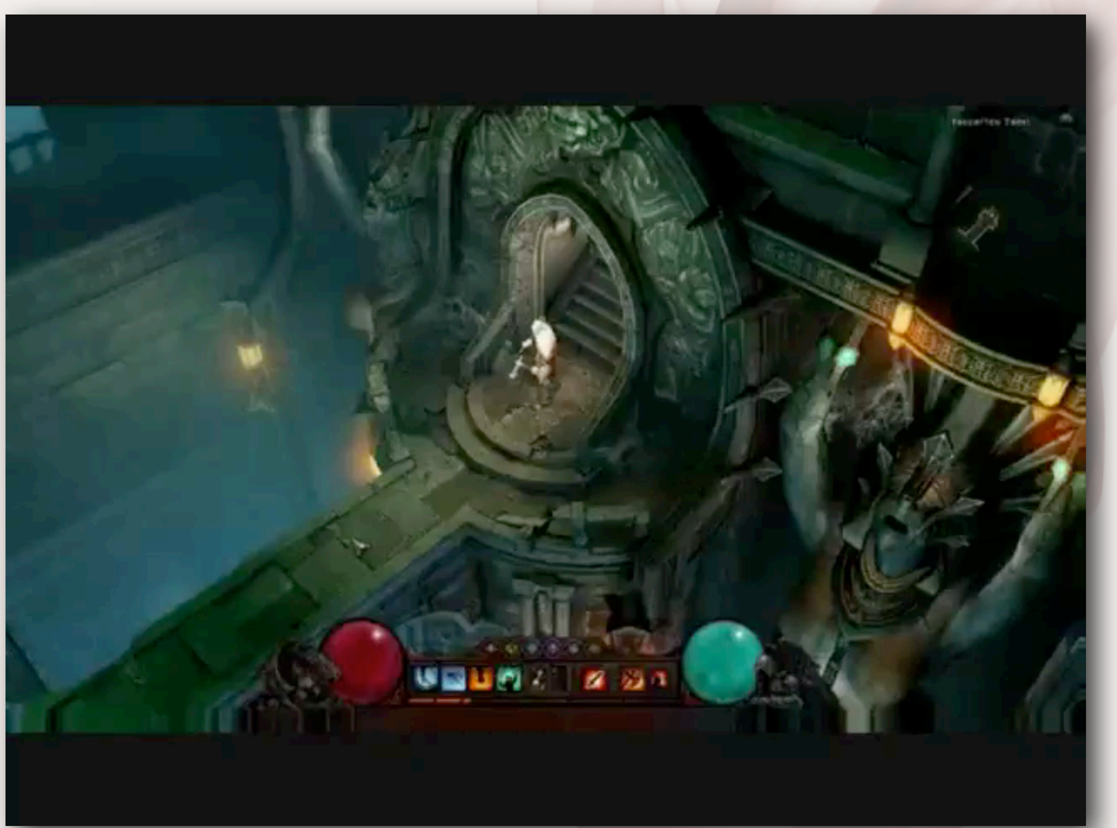


State of the Art

- **Steady but *slow* progress towards digital humans.**
  - **(As usual, rendering ahead of animation.)**
- **State of the art for animation production.**
- **State of the art for games.**

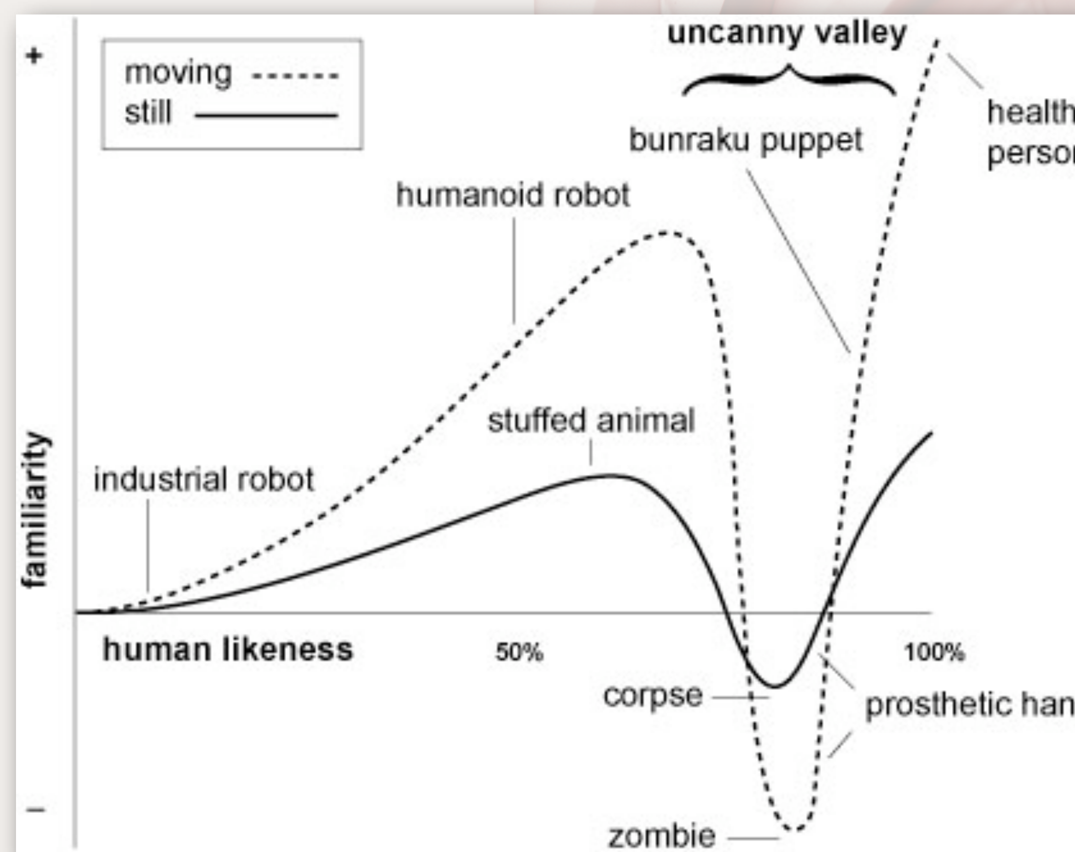
State of the Art

- Steady but *slow* progress towards digital humans.
  - (As usual, rendering ahead of animation.)
- State of the art for animation production.
- State of the art for games.



State of the Art

- Steady but *slow* progress towards digital humans.
  - (As usual, rendering ahead of animation.)
- State of the art for animation production.
- State of the art for games.
- Uncanny Valley.



State of the Art

- **Steady but *slow* progress towards digital humans.**
  - **(As usual, rendering ahead of animation.)**
- **State of the art for animation production.**
- **State of the art for games.**
- **Uncanny Valley.**
- **Facial Animation.**

State of the Art

- **Steady but *slow* progress towards digital humans.**
  - **(As usual, rendering ahead of animation.)**
- **State of the art for animation production.**
- **State of the art for games.**
- **Uncanny Valley.**
- **Facial Animation.**

State of the Art

- Steady but *slow* progress towards digital humans.
    - (As usual, rendering ahead of animation.)
- State of the art for animation production.
- State of the art for games.
- Uncanny Valley.
- Facial Animation.
- Most human animation is *data driven*.

(Like saying that graphics is solved by the camera.)

Overview

- **State of the art.**
- **Body models.**
- **Animation**
- **Questions**

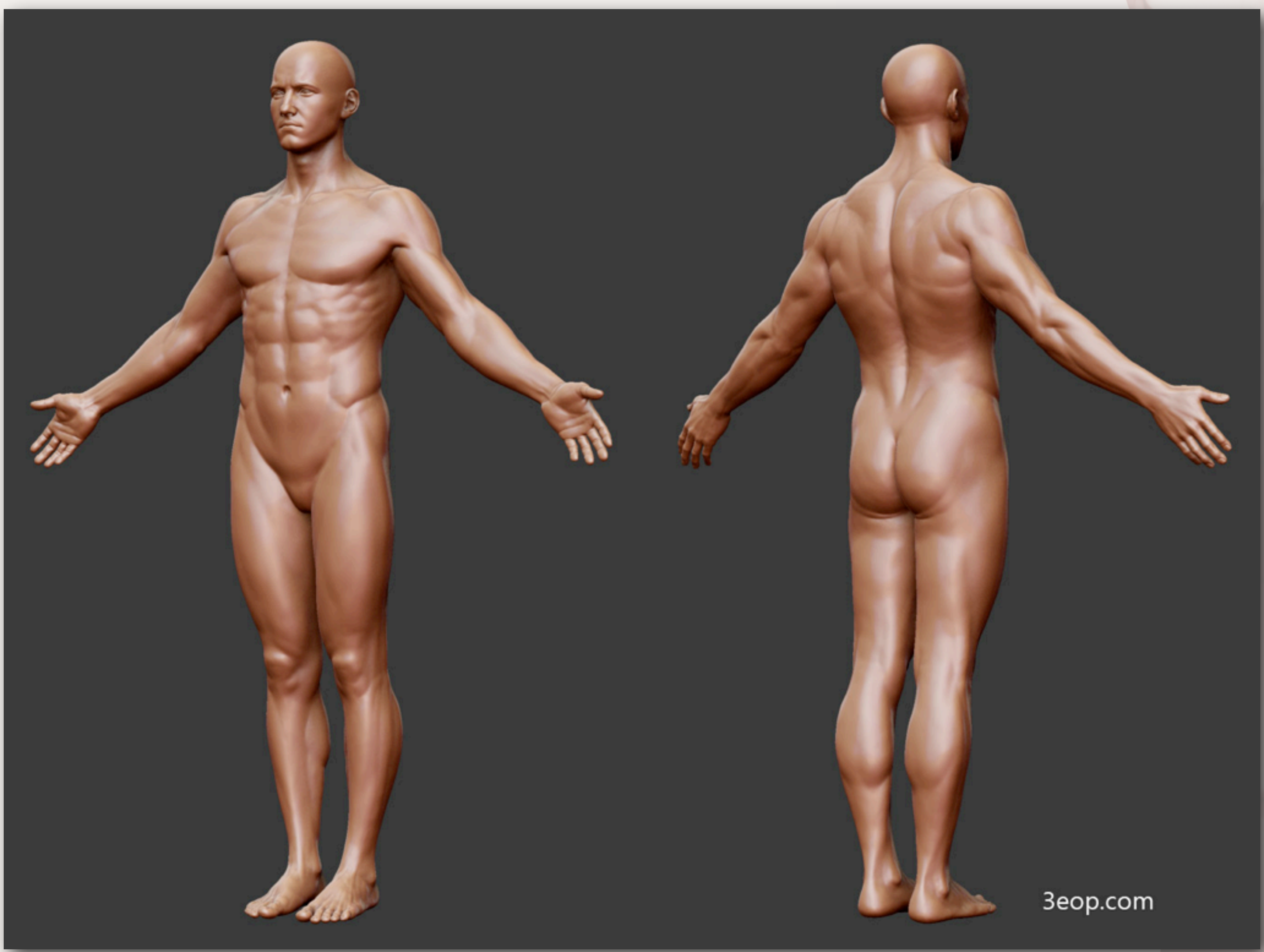
source: 3dscience.com

Overview

- State of the art.
- **Body models.**
- Animation
- Questions

source: 3dscience.com

Body Representation

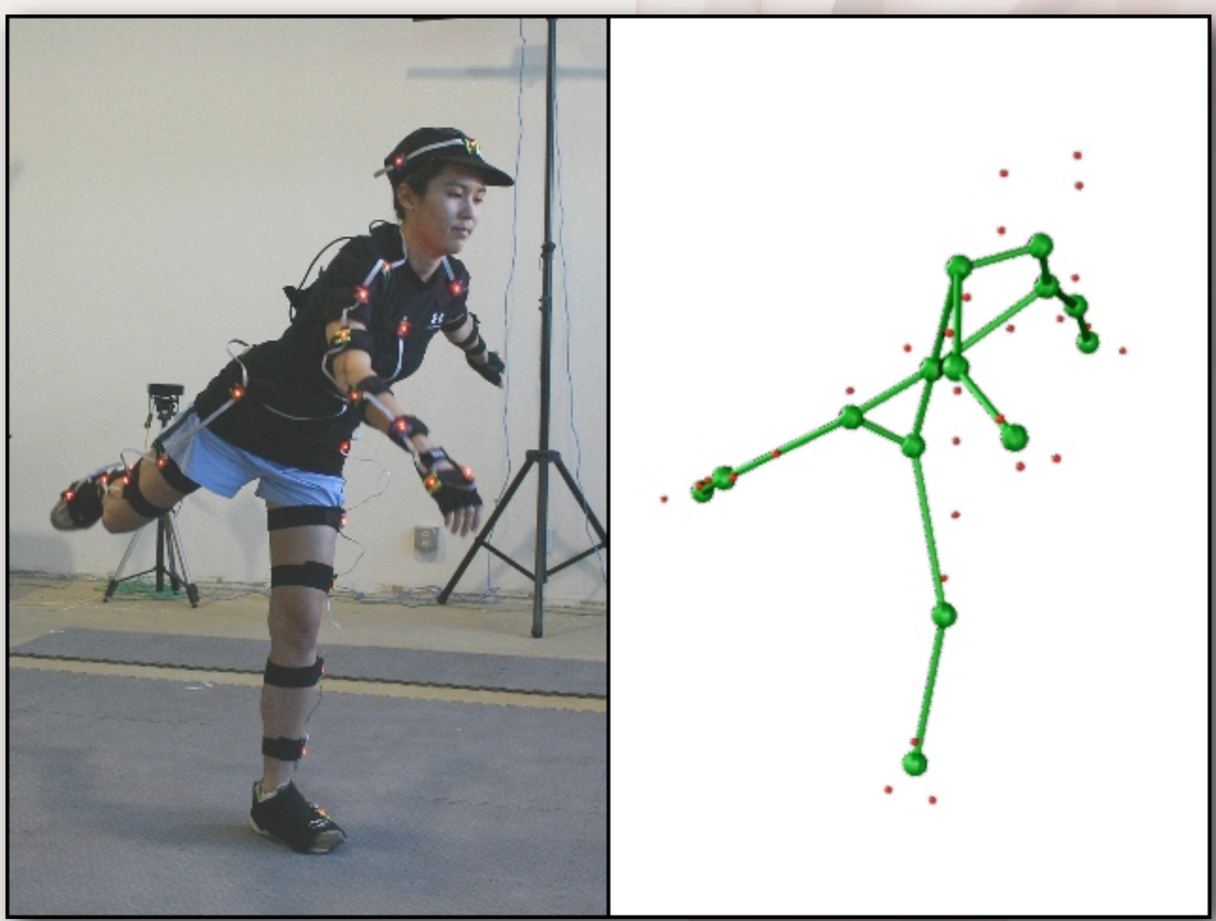


source: http://www.3eop.com/data/3d/images/08_05_26_anatomy_study_male.jpg

How to represent a human body on a computer?

Body Representation

- **Kinematic Skeleton**



source: https://buffy.eecs.berkeley.edu/PHP/resabs/resabs.php?f_year=2005&f_submit=advgrp&f_advid=10917651

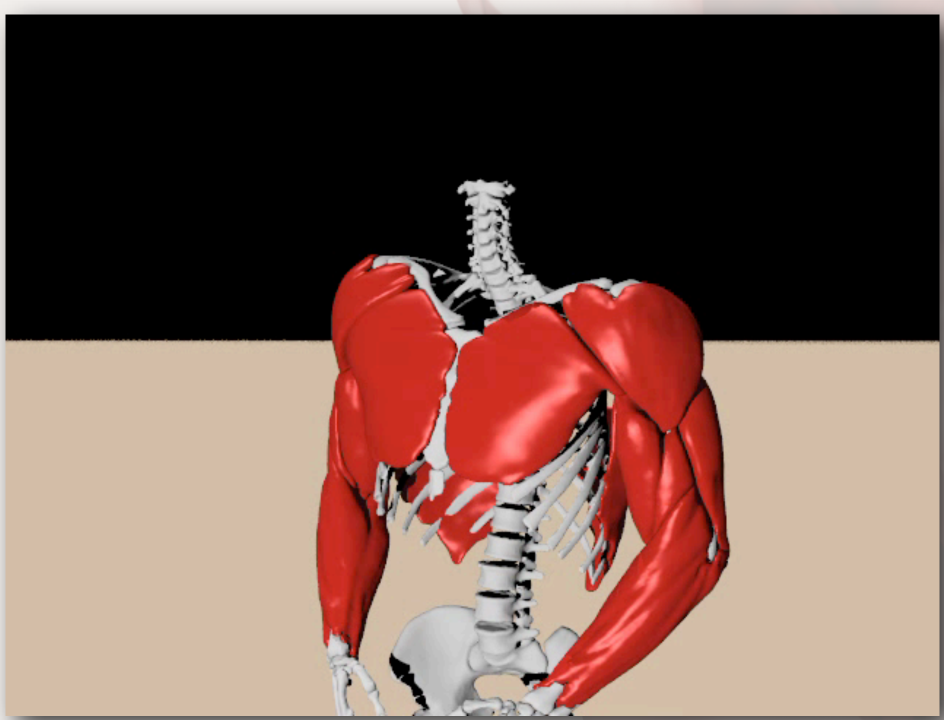
Body Representation

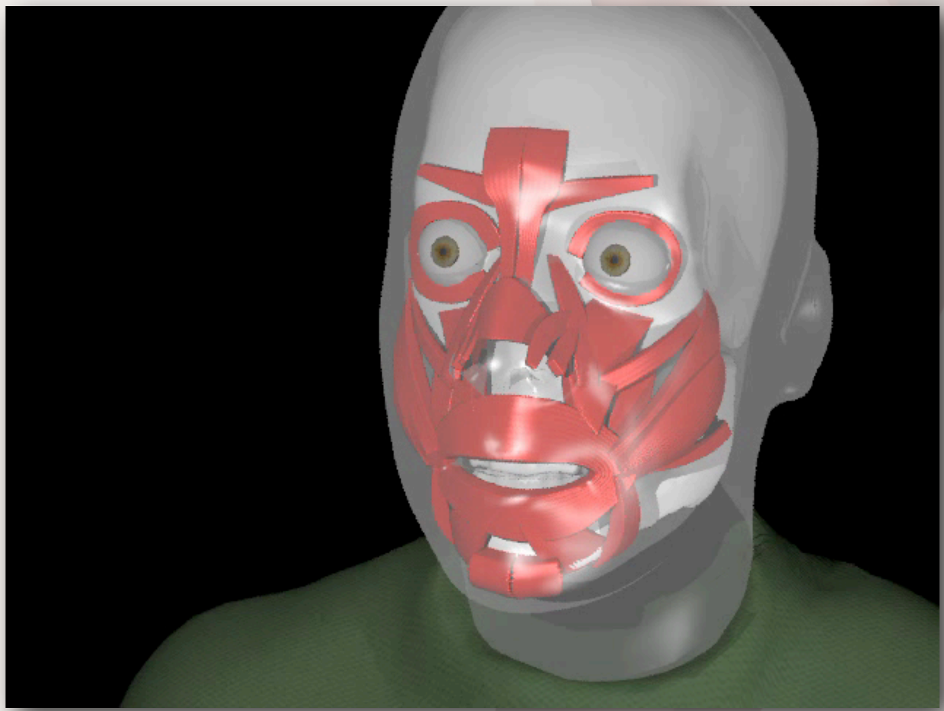
- Kinematic Skeleton
- Anatomical

source: http://physbam.stanford.edu/~fedkiw/

Body Representation

- Kinematic Skeleton
- Anatomical

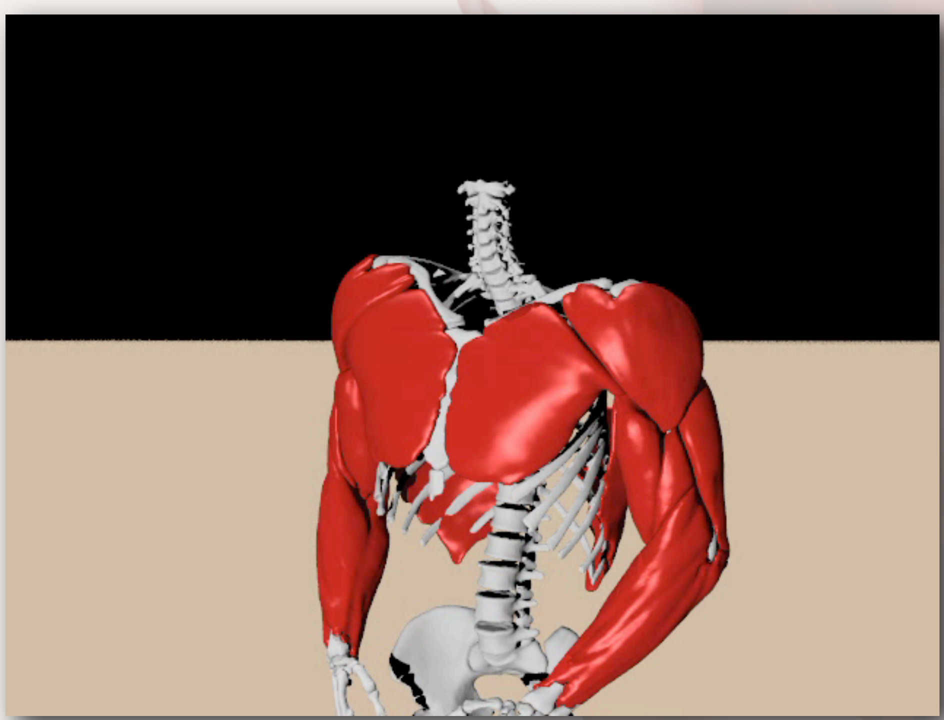


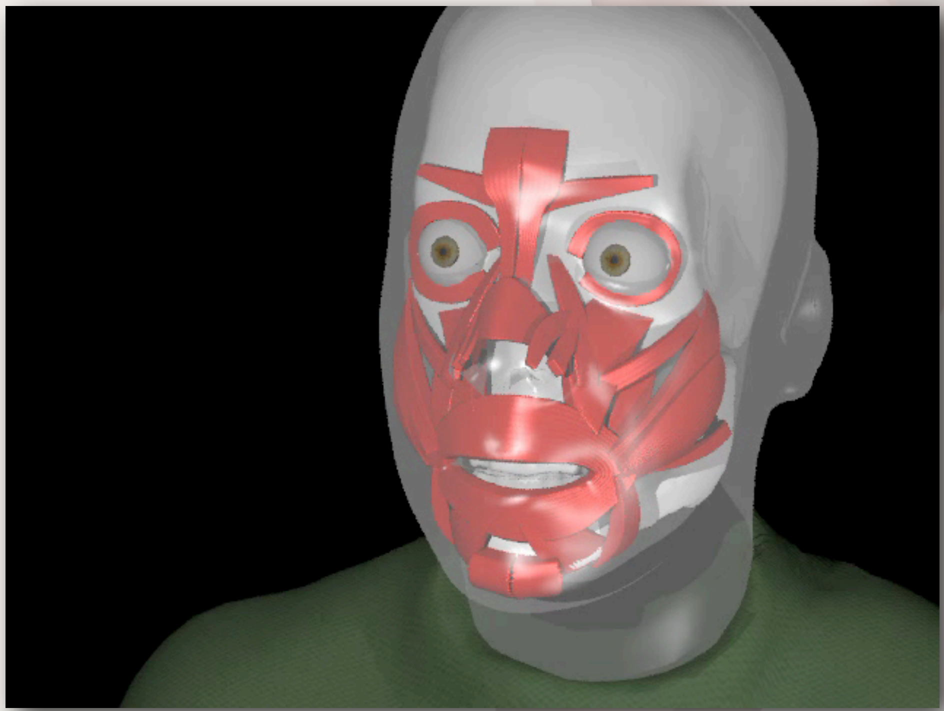


source: http://physbam.stanford.edu/~fedkiw/

Body Representation

- Kinematic Skeleton
- Anatomical

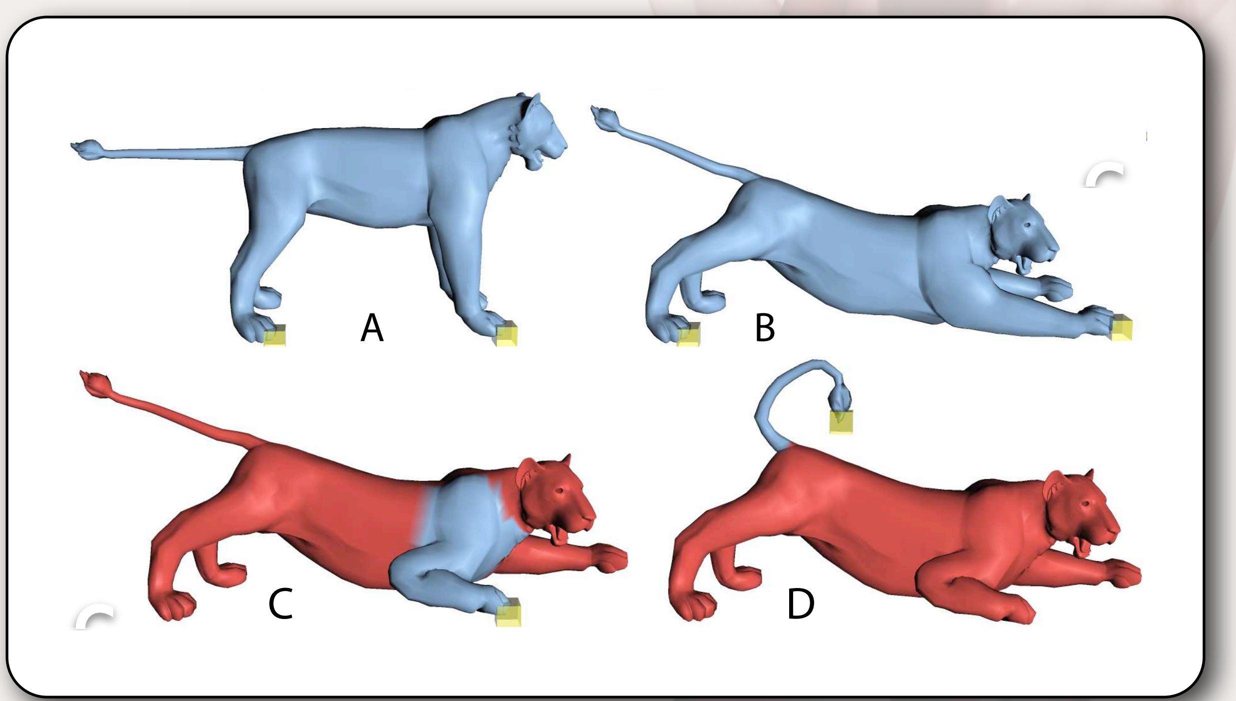




source: http://physbam.stanford.edu/~fedkiw/

Body Representation

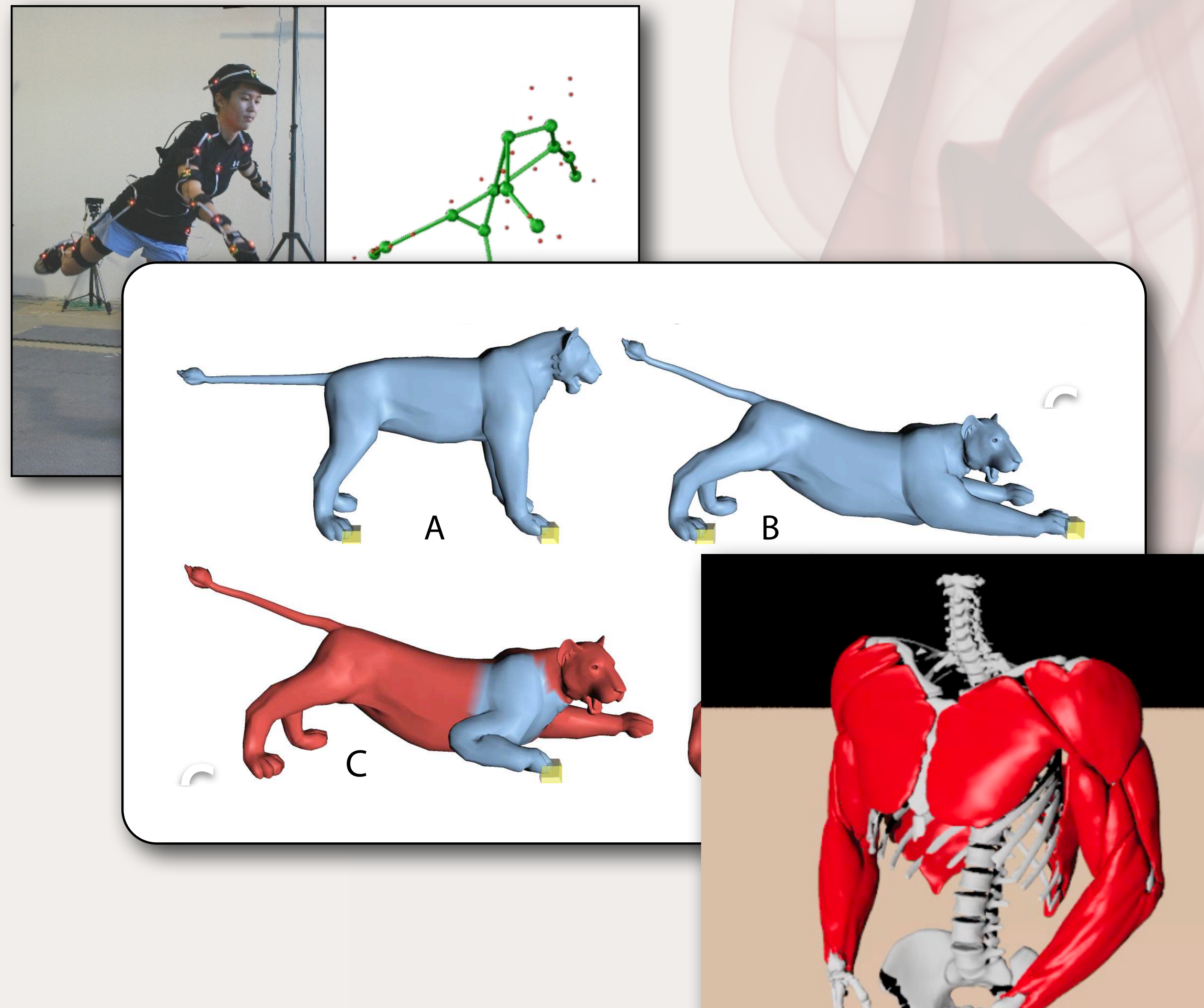
- Kinematic Skeleton
- Anatomical
- Pure Mesh



source: http://people.csail.mit.edu/sumner/research/meshik/

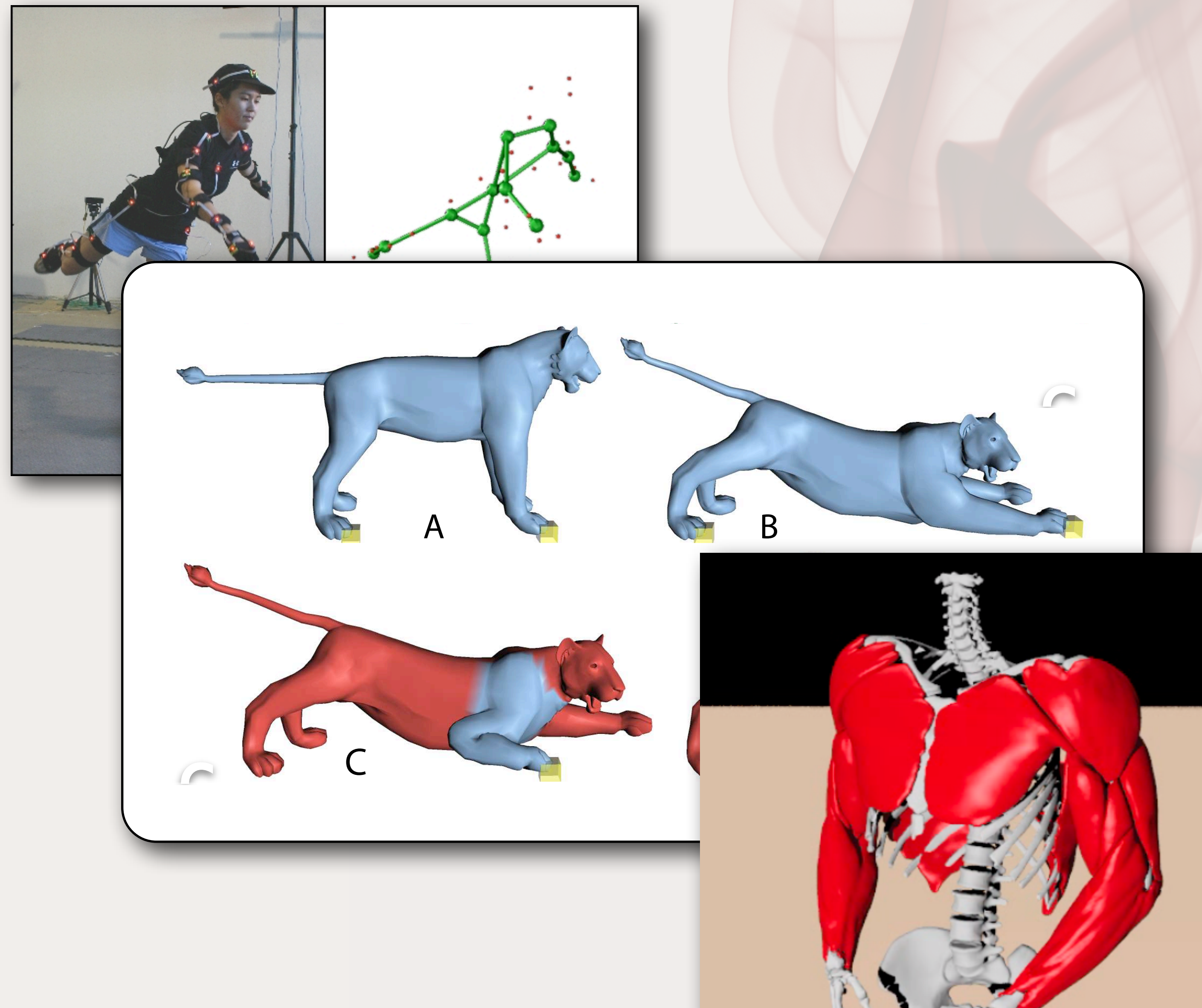
Body Representation

- **Kinematic Skeleton**
- **Anatomical**
- **Pure Mesh**
- **What are the advantages and disadvantages?**

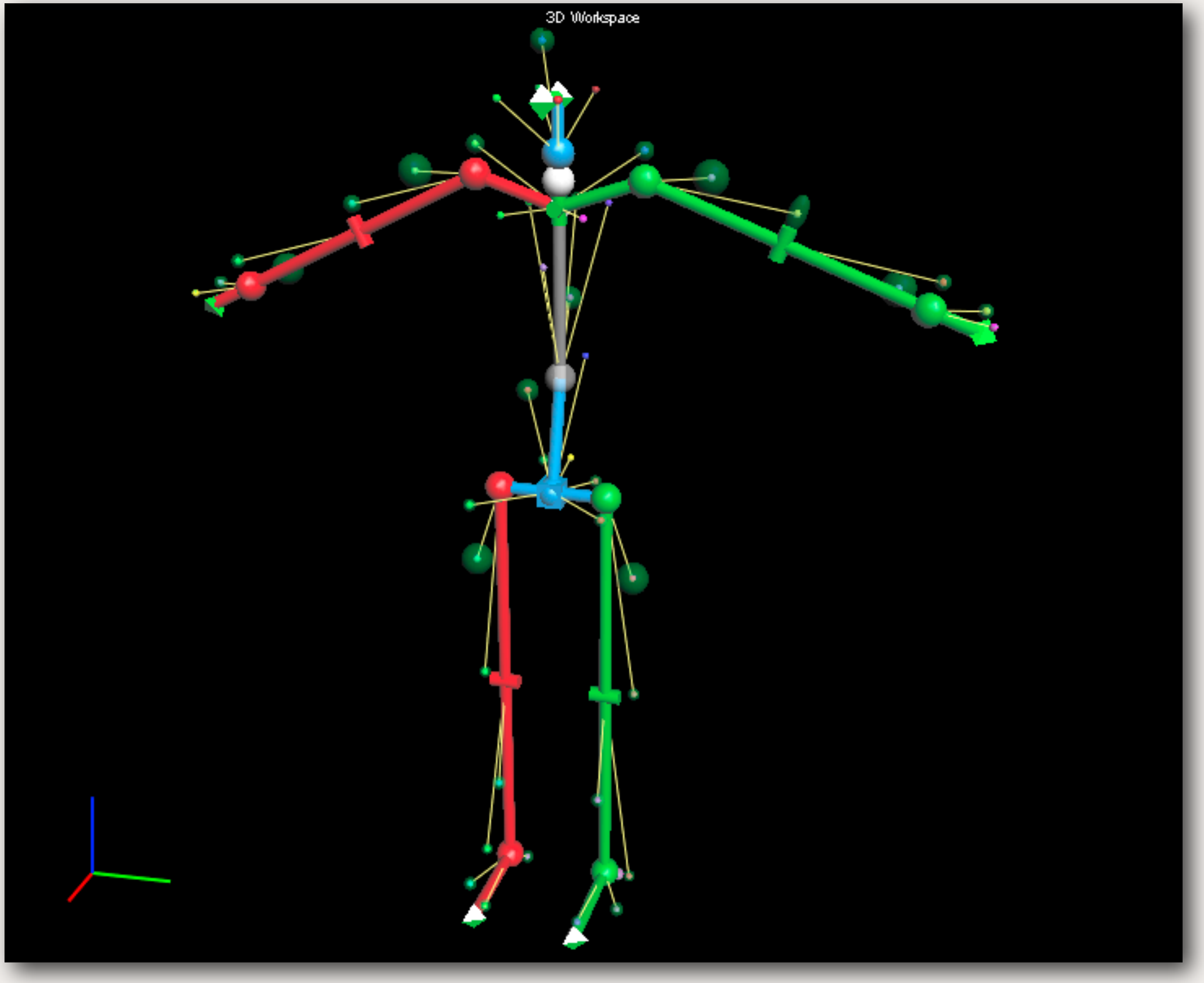


Body Representation

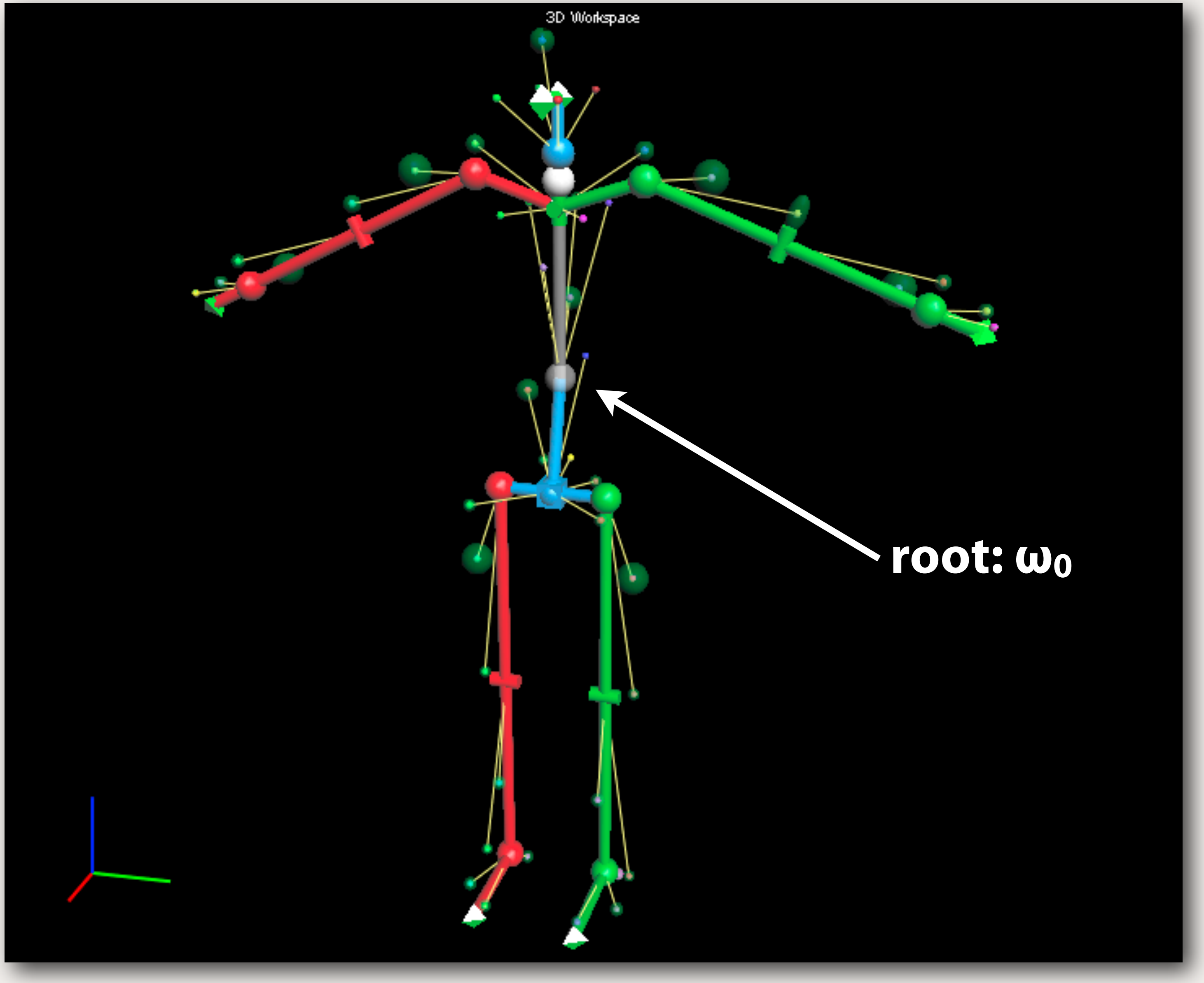
- **Kinematic Skeleton**
- **Anatomical**
- **Pure Mesh**
- **What are the advantages and disadvantages?**



Skeleton Representation

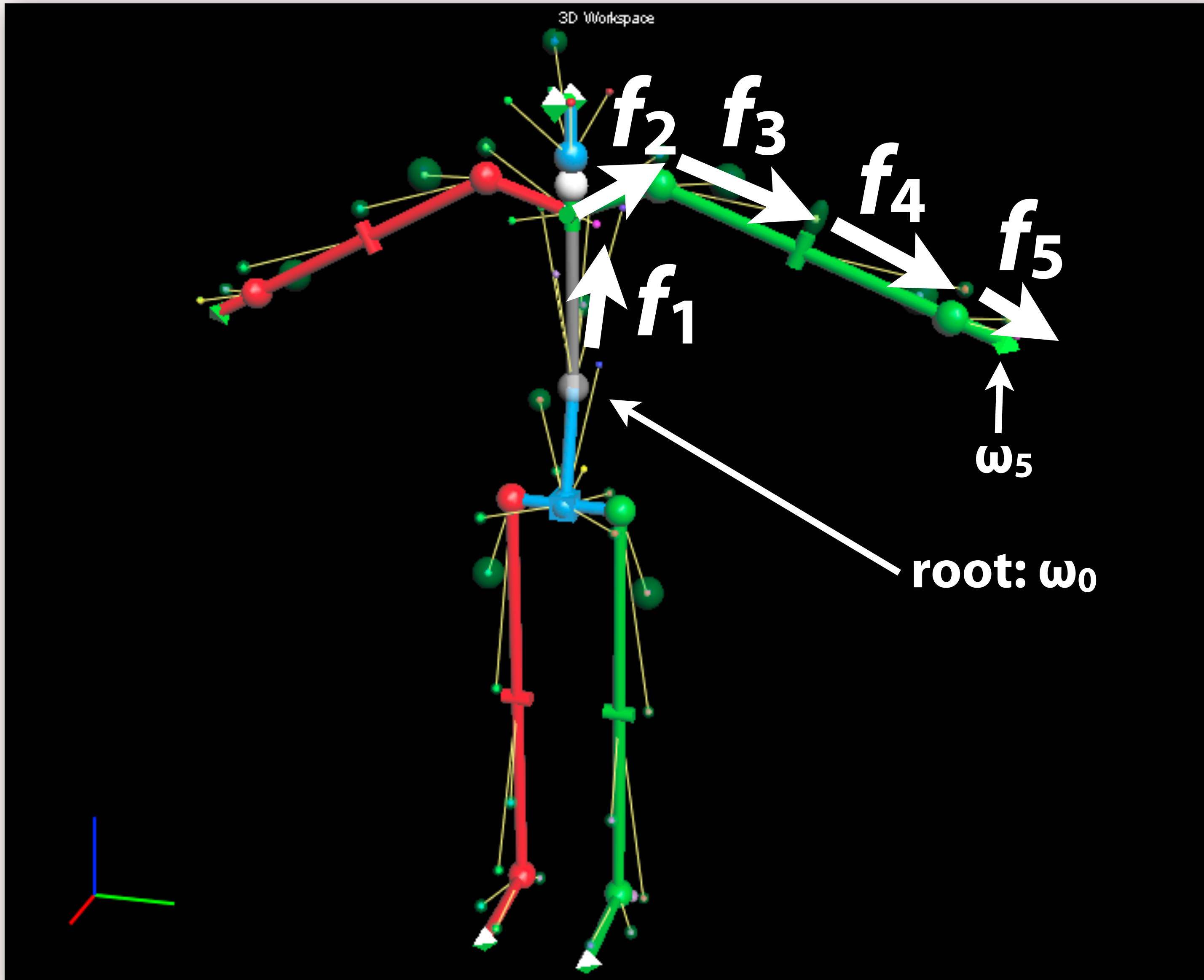


Skeleton Representation



$$\omega_0 = [\mathbf{x}_0, \theta_0] \in \mathbf{R}^6$$

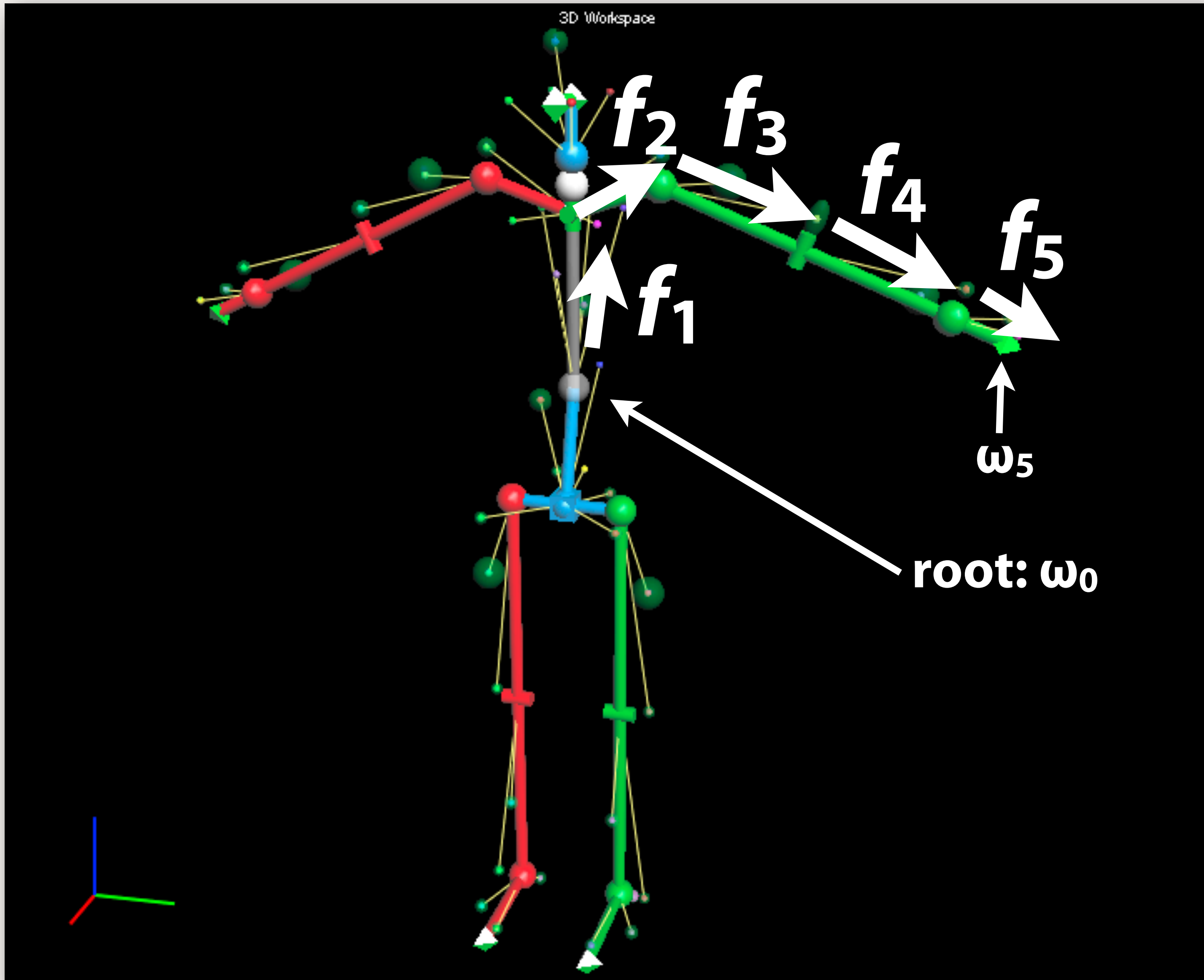
Skeleton Representation



$$\omega_0 = [\mathbf{x}_0, \theta_0] \in \mathbf{R}^6$$

$$\omega_i = f_{i,\Omega}(\omega_{i-1})$$

Skeleton Representation



Ω is the vector of *internal* joint angles, i.e. shoulders, hips, etc.

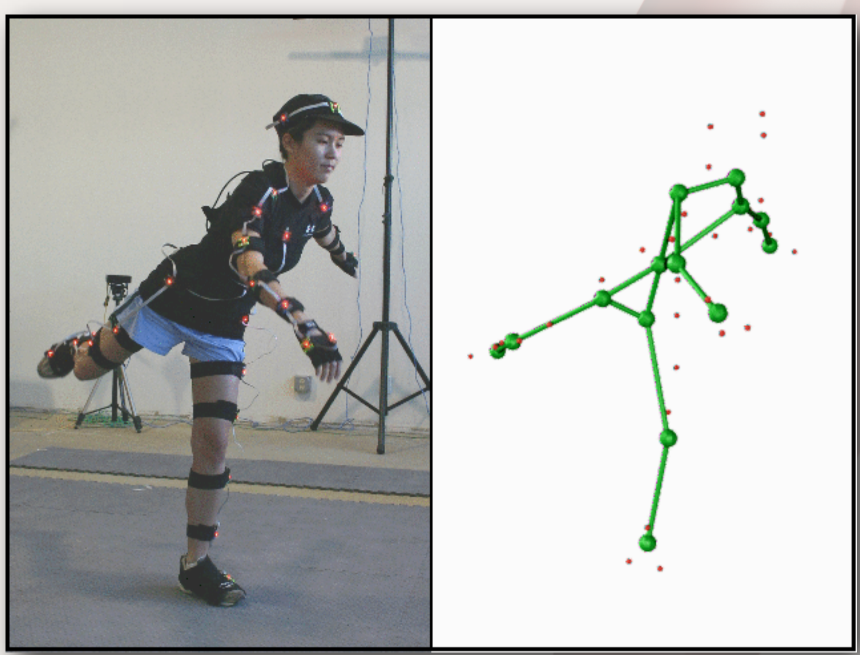
$$\omega_0 = [\mathbf{x}_0, \theta_0] \in \mathbf{R}^6$$

$$\omega_i = f_{i,\Omega}(\omega_{i-1})$$

Motion Capture

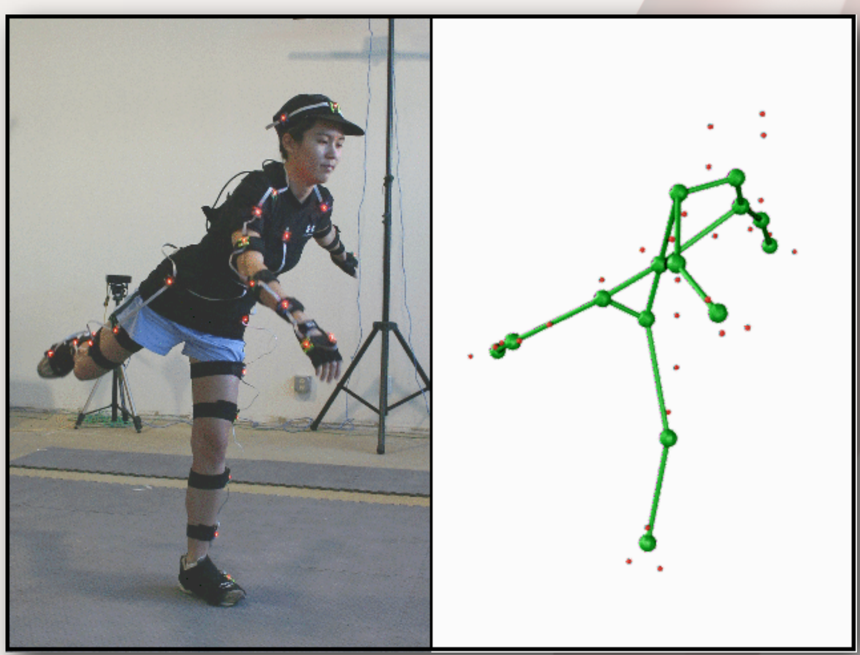
Motion Capture

- **Attach markers to a humans body.**



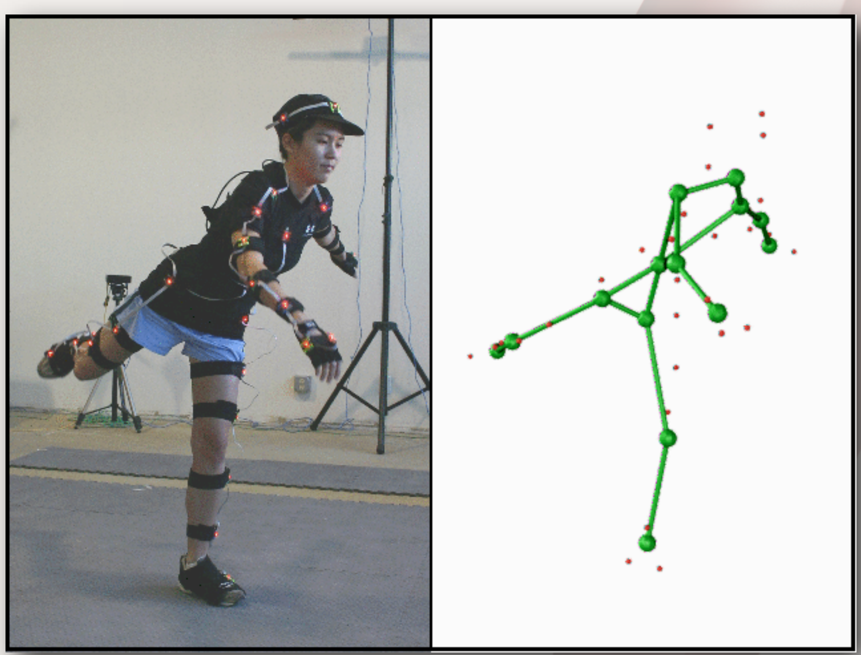
Motion Capture

- **Attach markers to a humans body.**
- **Calibrate a skeleton which makes those markers "make sense."**



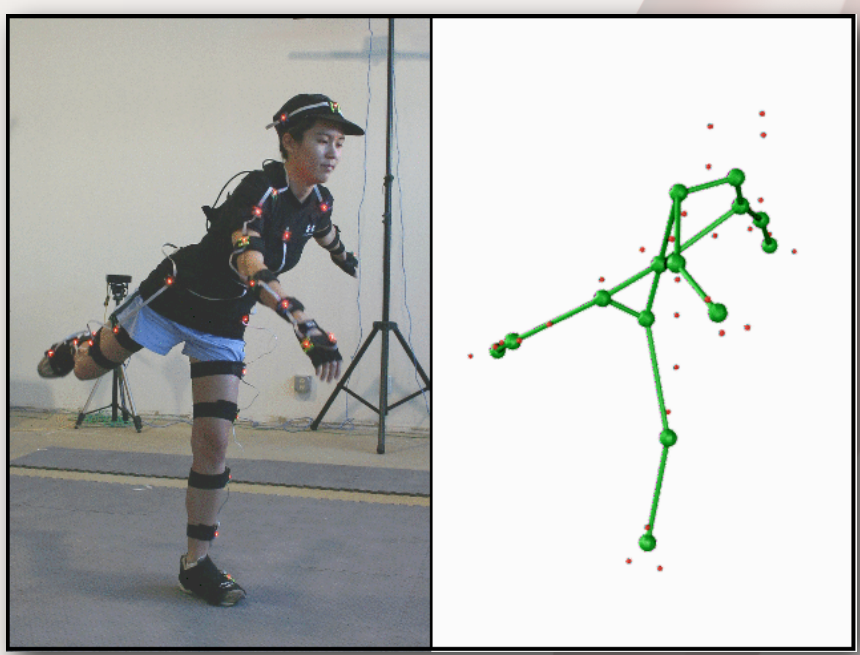
Motion Capture

- Attach markers to a humans body.
- Calibrate a skeleton which makes those markers "make sense."
- Cameras capture 2D markers positions.



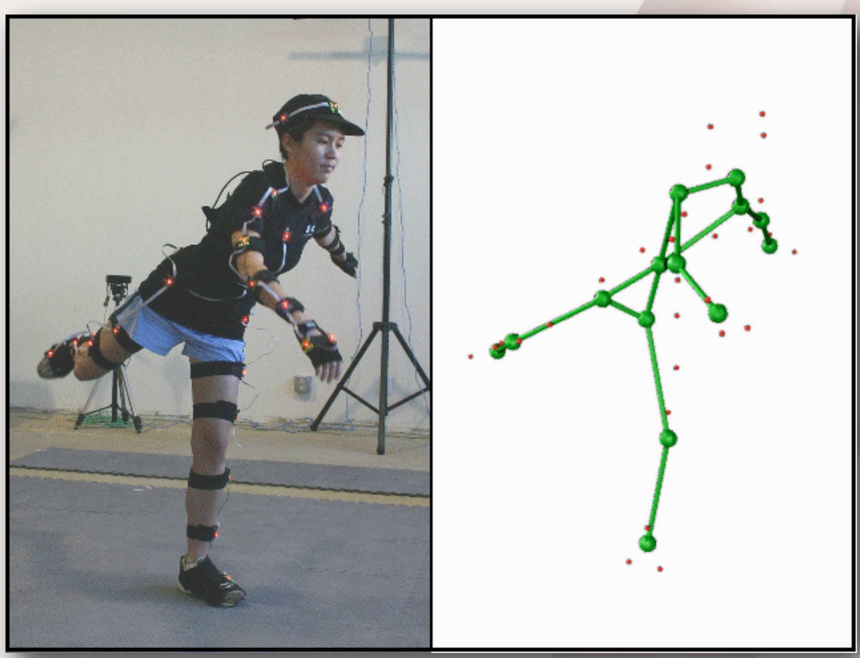
Motion Capture

- Attach markers to a humans body.
- Calibrate a skeleton which makes those markers "make sense."
- Cameras capture 2D markers positions.
- Estimate 3D marker positions.



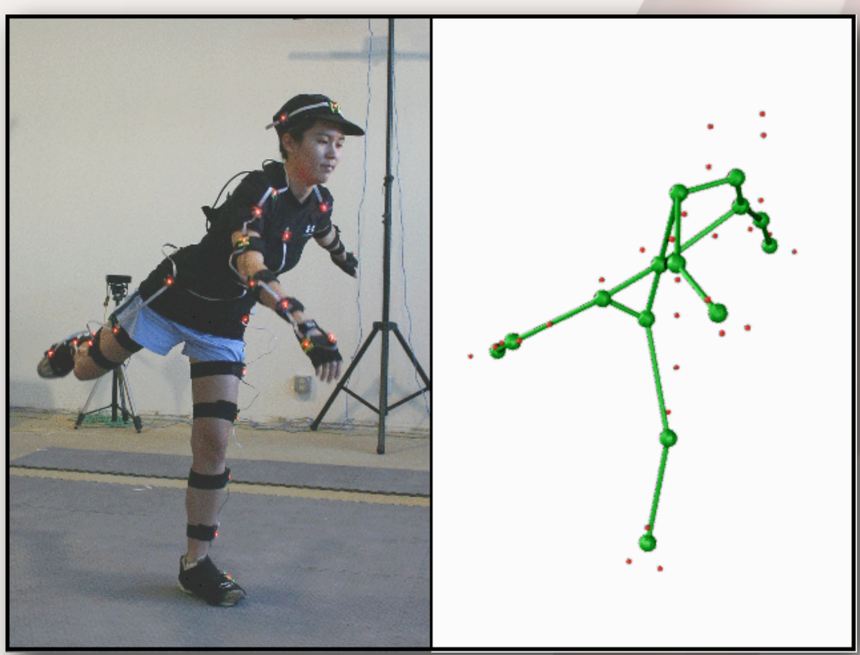
Motion Capture

- **Attach markers to a humans body.**
- **Calibrate a skeleton which makes those markers “make sense.”**
- **Cameras capture 2D markers positions.**
- **Estimate 3D marker positions.**
- ***Inverse kinematics*: convert marker positions to skeleton...**

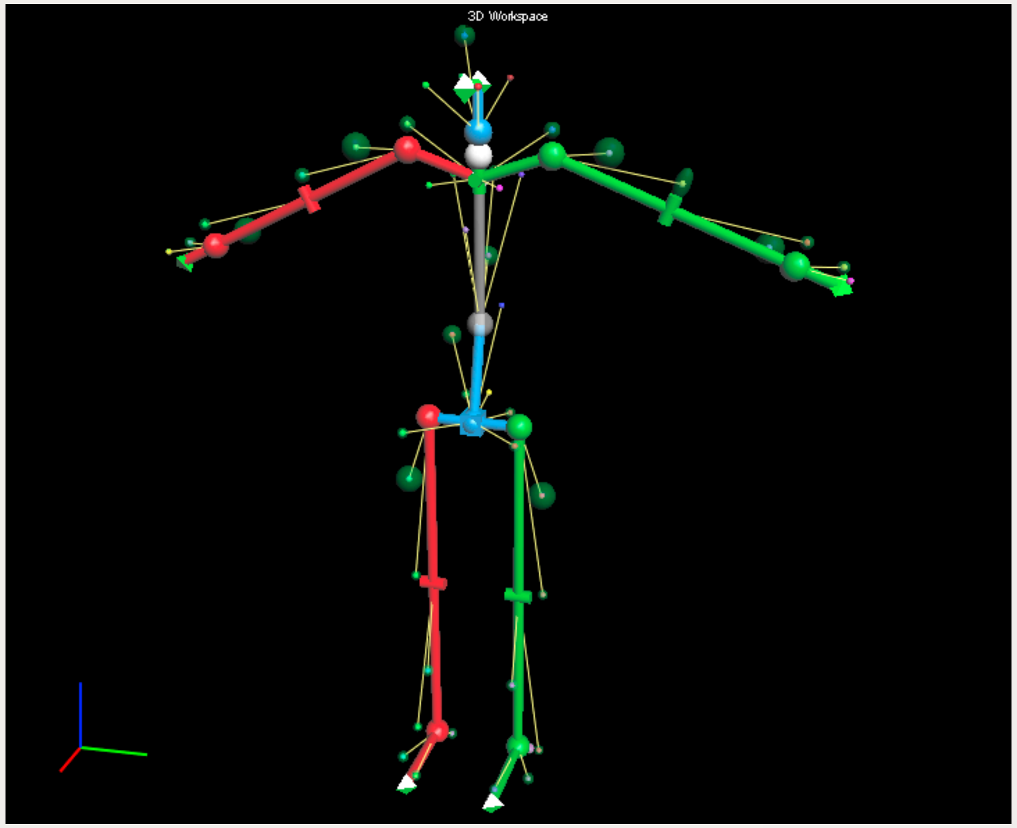


Motion Capture

- Attach markers to a humans body.
- Calibrate a skeleton which makes those markers "make sense."
- Cameras capture 2D markers positions.
- Estimate 3D marker positions.
- *Inverse kinematics*: convert marker positions to skeleton...
  - How?

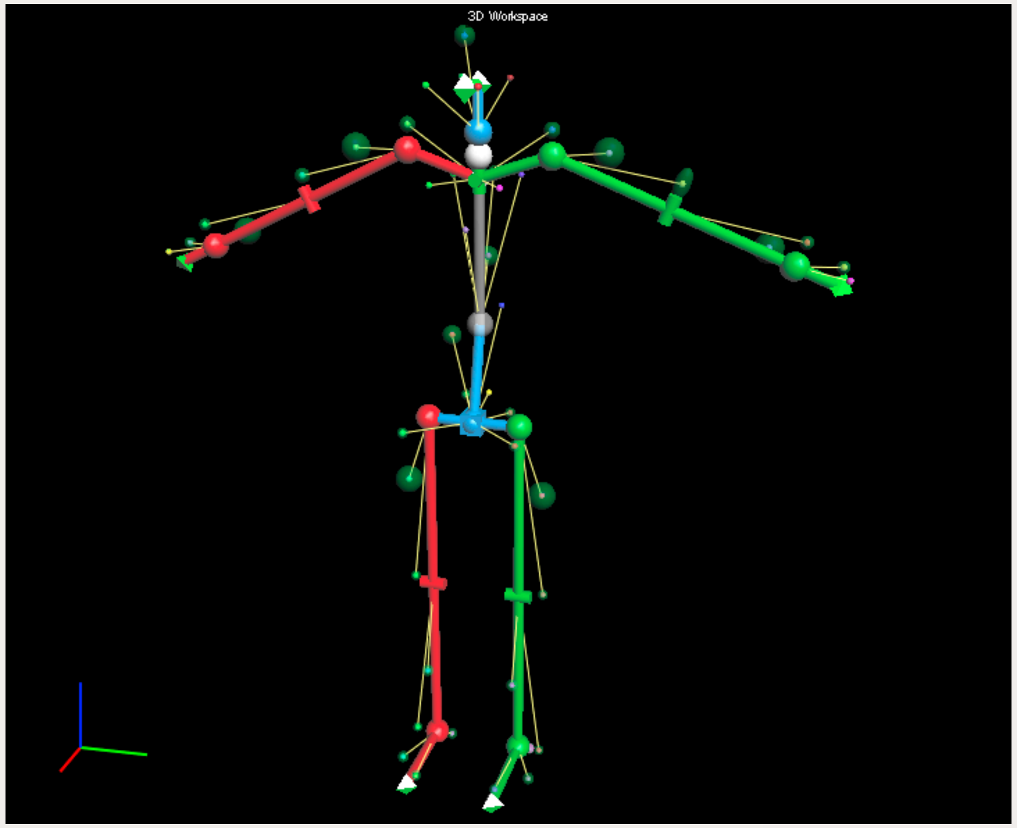


Marker Energy Function



$$\omega_i = f_{i,\Omega}(\omega_{i-1})$$

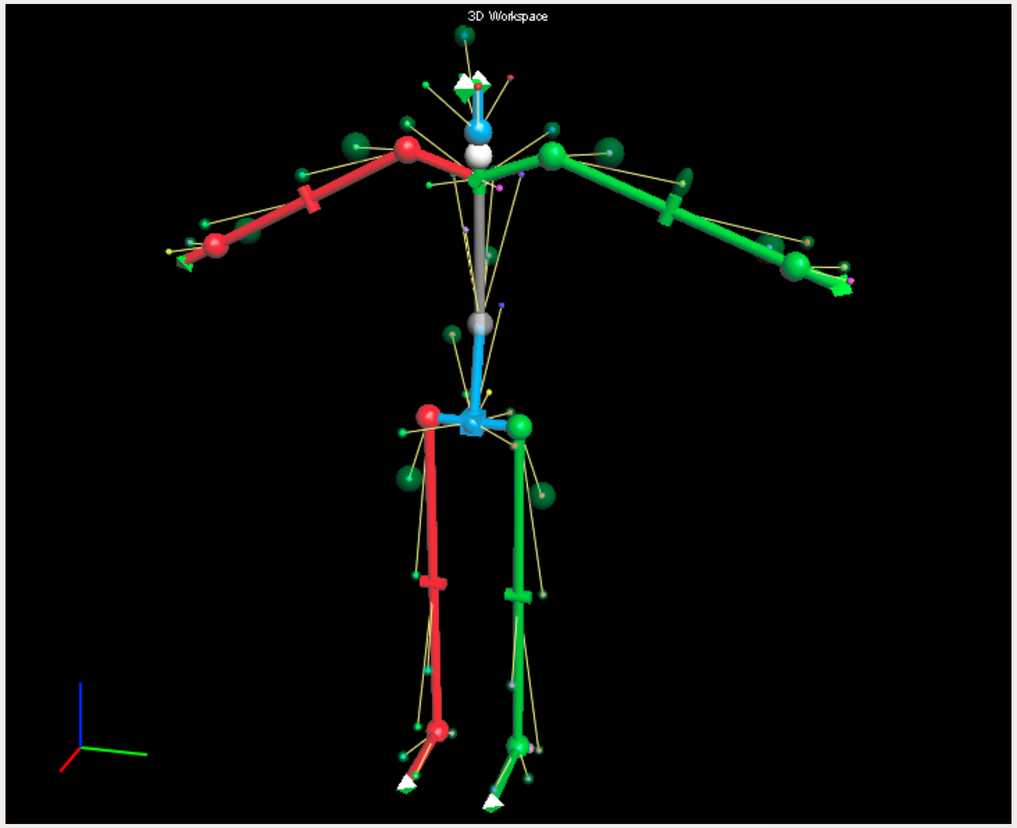
Marker Energy Function



$$\omega_i = f_{i,\Omega}(\omega_{i-1})$$

$$\hat{\mathbf{m}}_j = \tau_i(\omega_i)\mathbf{m}_j$$

Marker Energy Function

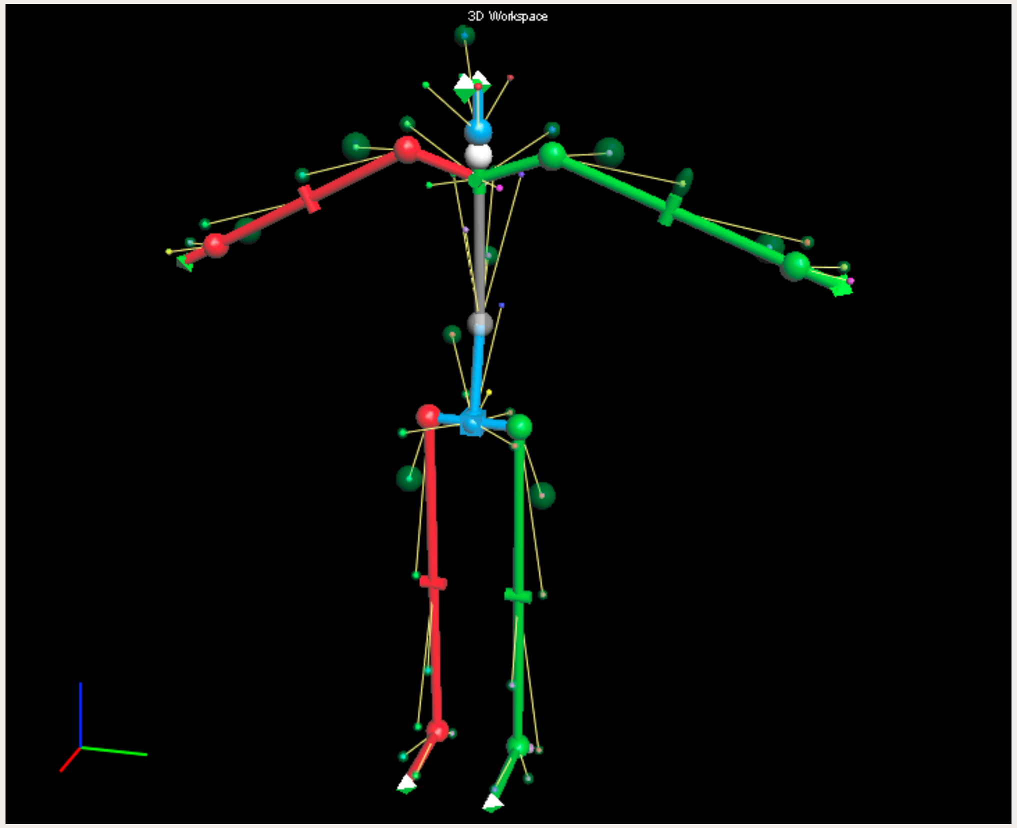


$$\omega_i = f_{i,\Omega}(\omega_{i-1})$$

$$\hat{\mathbf{m}}_j = \tau_i(\omega_i)\mathbf{m}_j$$

$$E = \sum_j ||\hat{\mathbf{m}}_j^\star - \hat{\mathbf{m}}_j||^2$$

Marker Energy Function



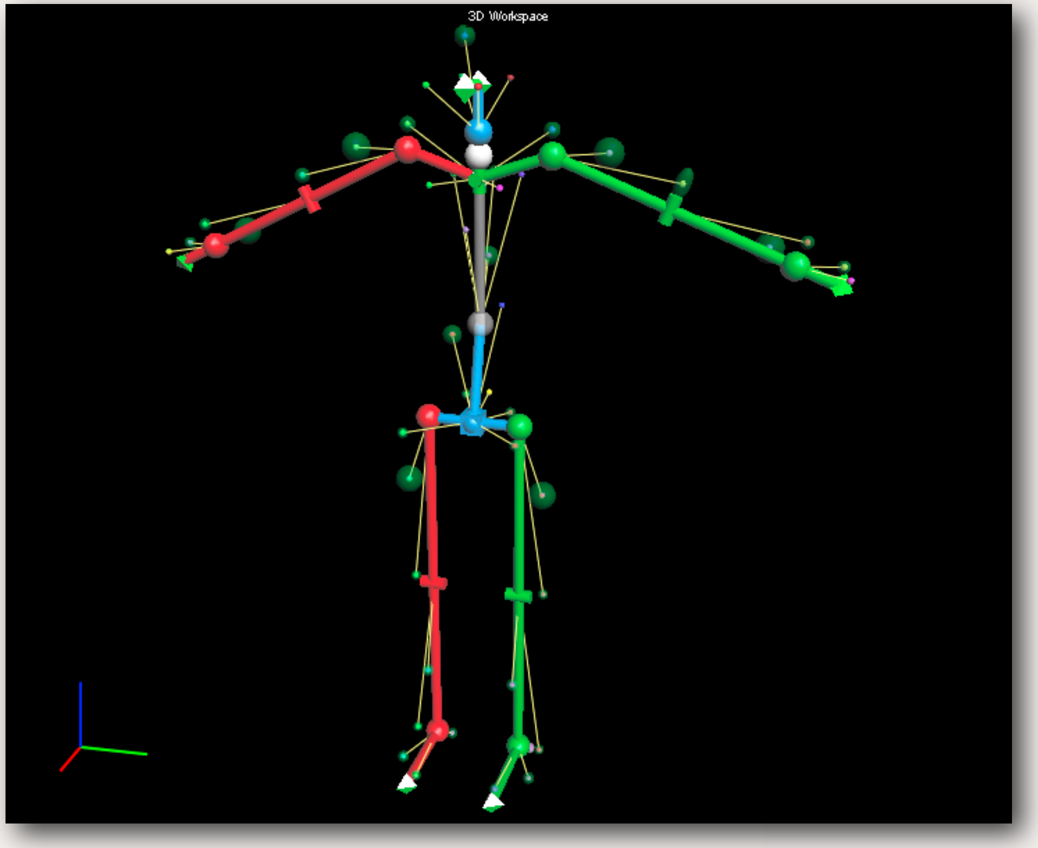
$$\omega_i = f_{i,\Omega}(\omega_{i-1})$$

$$\hat{\mathbf{m}}_j = \tau_i(\omega_i)\mathbf{m}_j$$

$$E = \sum_j ||\hat{\mathbf{m}}_j^\star - \hat{\mathbf{m}}_j||^2$$

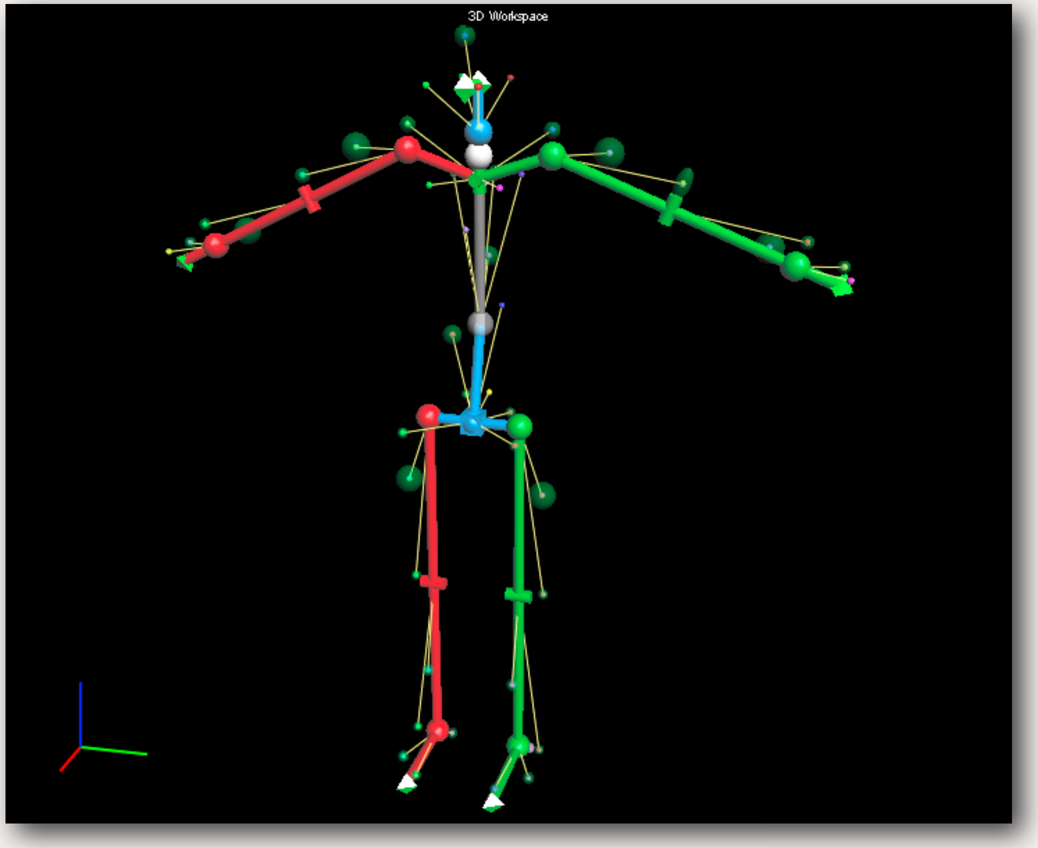
$$\boxed{\frac{dE}{d\Omega}}$$

Derivatives



$$\omega_i = f_{i,\Omega}(\omega_{i-1})$$

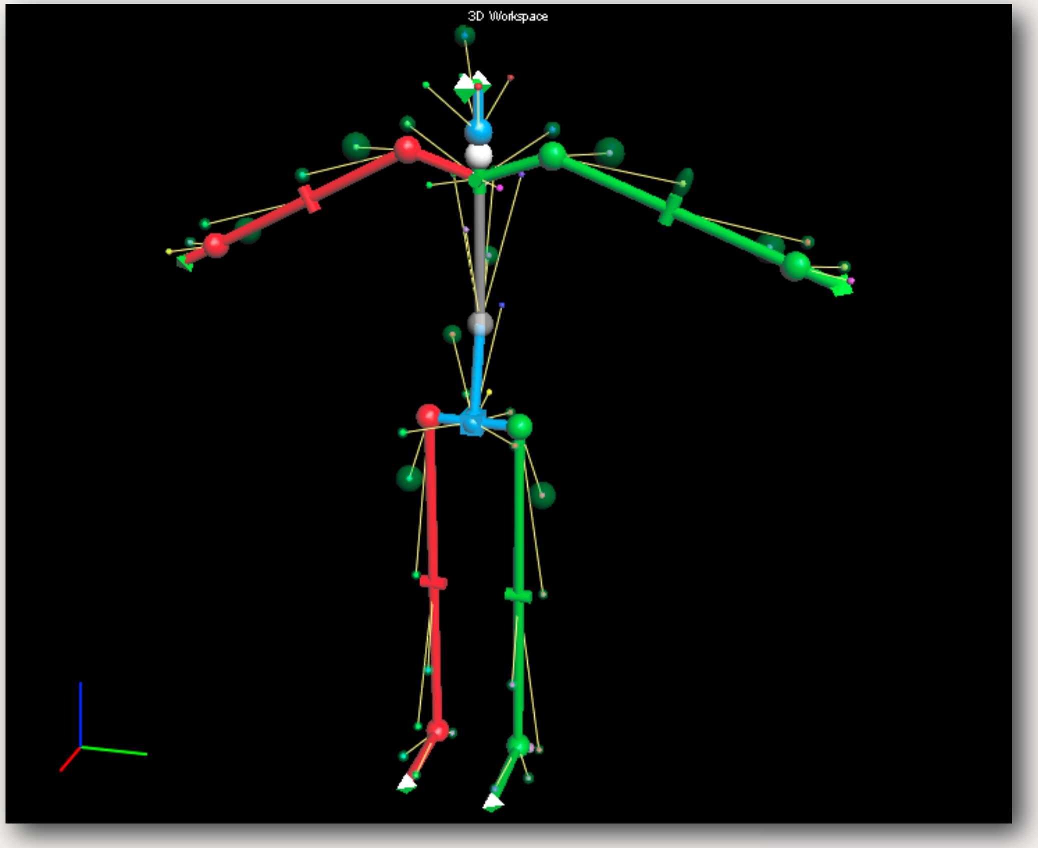
Derivatives



$$\omega_i = f_{i,\Omega}(\omega_{i-1})$$

$$\frac{dE}{d\Omega} = 2\sum_j (\hat{\mathbf{m}}_j^\star - \hat{\mathbf{m}}_j)^T \frac{d\hat{\mathbf{m}}_j}{d\Omega}$$

Derivatives

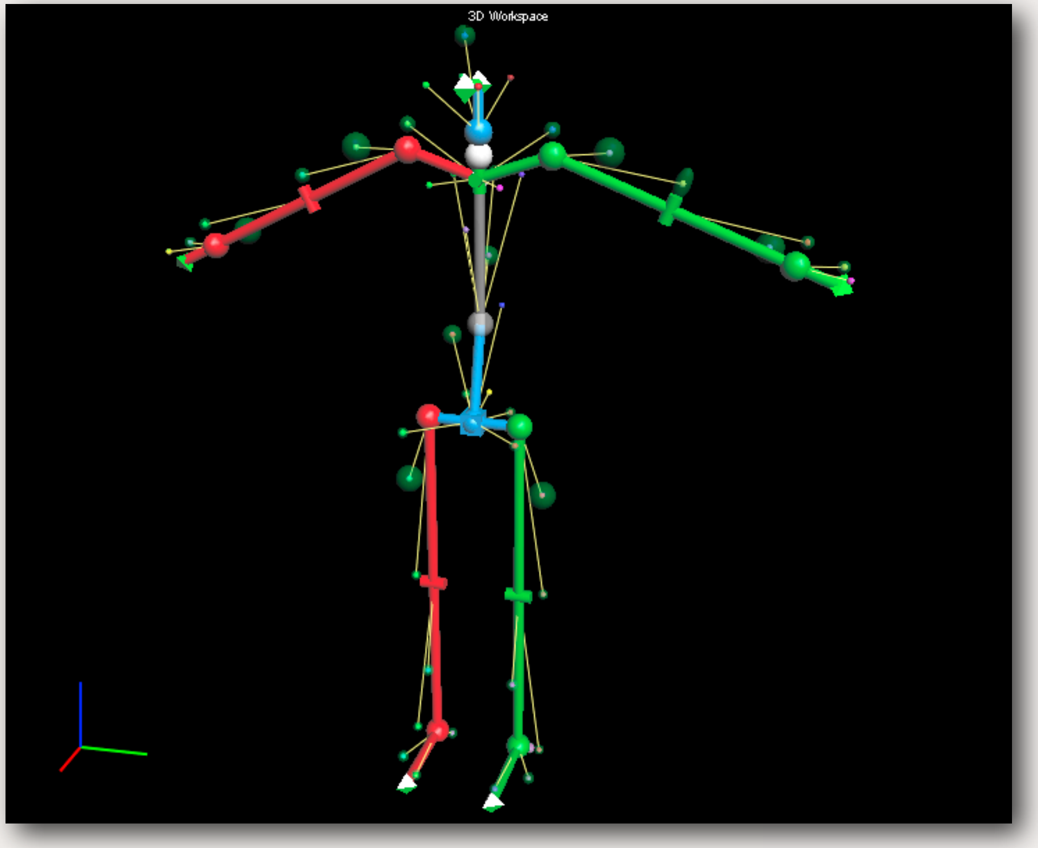


$$\omega_i = f_{i,\Omega}(\omega_{i-1})$$

$$\frac{dE}{d\Omega} = 2\sum_j (\hat{\mathbf{m}}_j^\star - \hat{\mathbf{m}}_j)^T \frac{d\hat{\mathbf{m}}_j}{d\Omega}$$

$$\frac{d\hat{\mathbf{m}}_j}{d\Omega} = \frac{\partial \hat{\mathbf{m}}_j}{\partial \omega_i}\left(\frac{\partial \omega_i}{\partial \Omega} + \frac{\partial \omega_i}{\partial \omega_{i-1}}\frac{\partial \omega_{i-1}}{\partial \Omega} + \frac{\partial \omega_i}{\partial \omega_{i-1}}\frac{\partial \omega_{i-1}}{\partial \omega_{i-2}}\frac{\partial \omega_{i-2}}{\partial \Omega} + \cdots\right)$$

Derivatives

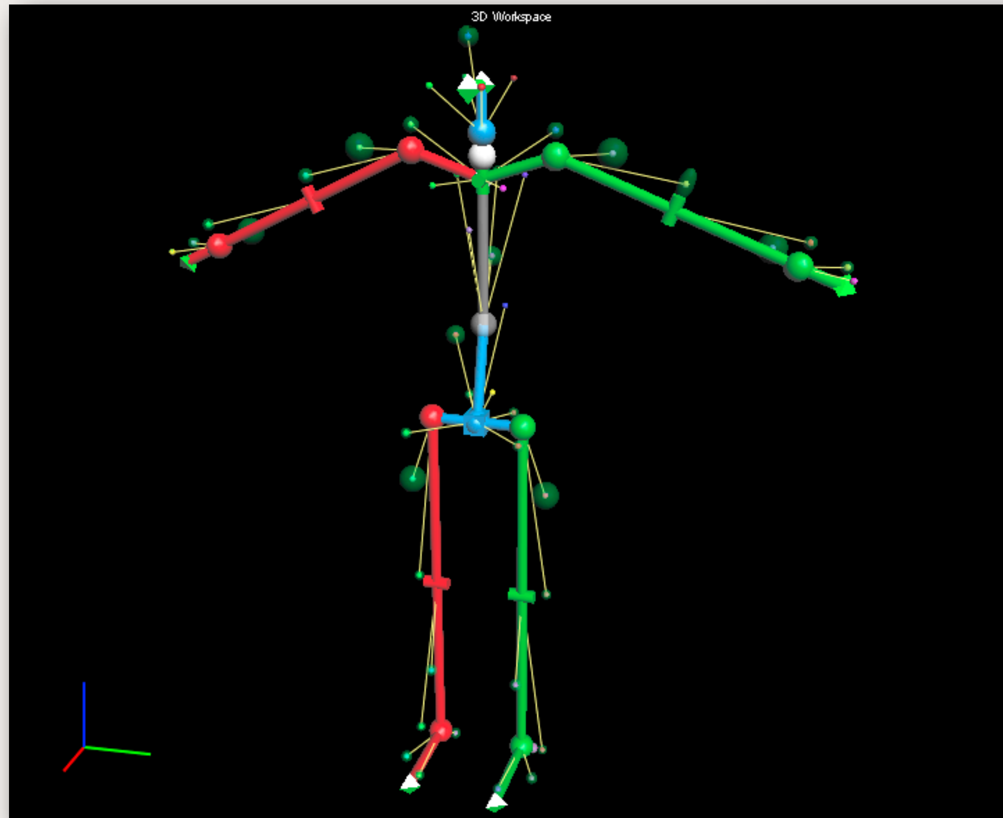


$$\omega_i = f_{i,\Omega}(\omega_{i-1})$$

$$\frac{dE}{d\Omega} = 2\sum_j \underbrace{(\hat{\mathbf{m}}_j^\star - \hat{\mathbf{m}}_j)^T}_{\text{vector}} \frac{d\hat{\mathbf{m}}_j}{d\Omega}$$

$$\frac{d\hat{\mathbf{m}}_j}{d\Omega} = \frac{\partial \hat{\mathbf{m}}_j}{\partial \omega_i}\left(\frac{\partial \omega_i}{\partial \Omega} + \frac{\partial \omega_i}{\partial \omega_{i-1}}\frac{\partial \omega_{i-1}}{\partial \Omega} + \frac{\partial \omega_i}{\partial \omega_{i-1}}\frac{\partial \omega_{i-1}}{\partial \omega_{i-2}}\frac{\partial \omega_{i-2}}{\partial \Omega} + \cdots\right)$$

Derivatives

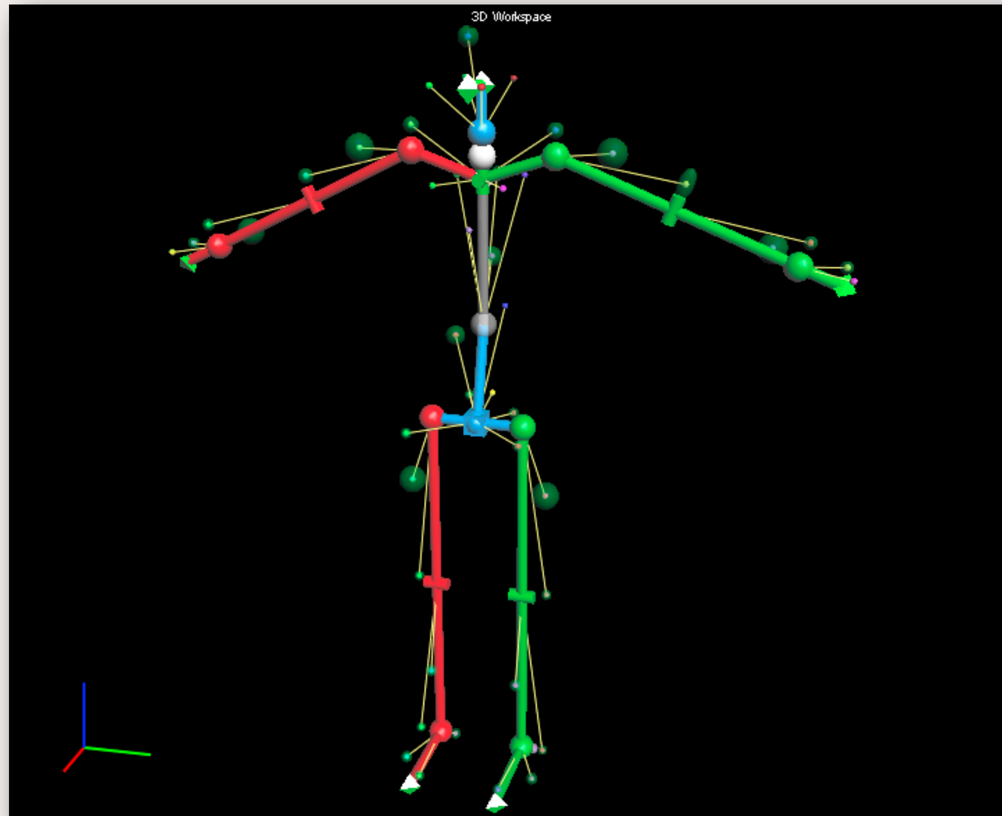


$$\omega_i = f_{i,\Omega}(\omega_{i-1})$$

$$\frac{dE}{d\Omega} = 2\sum_j \underbrace{(\hat{\mathbf{m}}_j^\star - \hat{\mathbf{m}}_j)^T}_{\text{vector}} \underbrace{\frac{d\hat{\mathbf{m}}_j}{d\Omega}}_{\text{matrix}}$$

$$\frac{d\hat{\mathbf{m}}_j}{d\Omega} = \frac{\partial\hat{\mathbf{m}}_j}{\partial\omega_i}\left(\frac{\partial\omega_i}{\partial\Omega} + \frac{\partial\omega_i}{\partial\omega_{i-1}}\frac{\partial\omega_{i-1}}{\partial\Omega} + \frac{\partial\omega_i}{\partial\omega_{i-1}}\frac{\partial\omega_{i-1}}{\partial\omega_{i-2}}\frac{\partial\omega_{i-2}}{\partial\Omega} + \cdots\right)$$

Derivatives

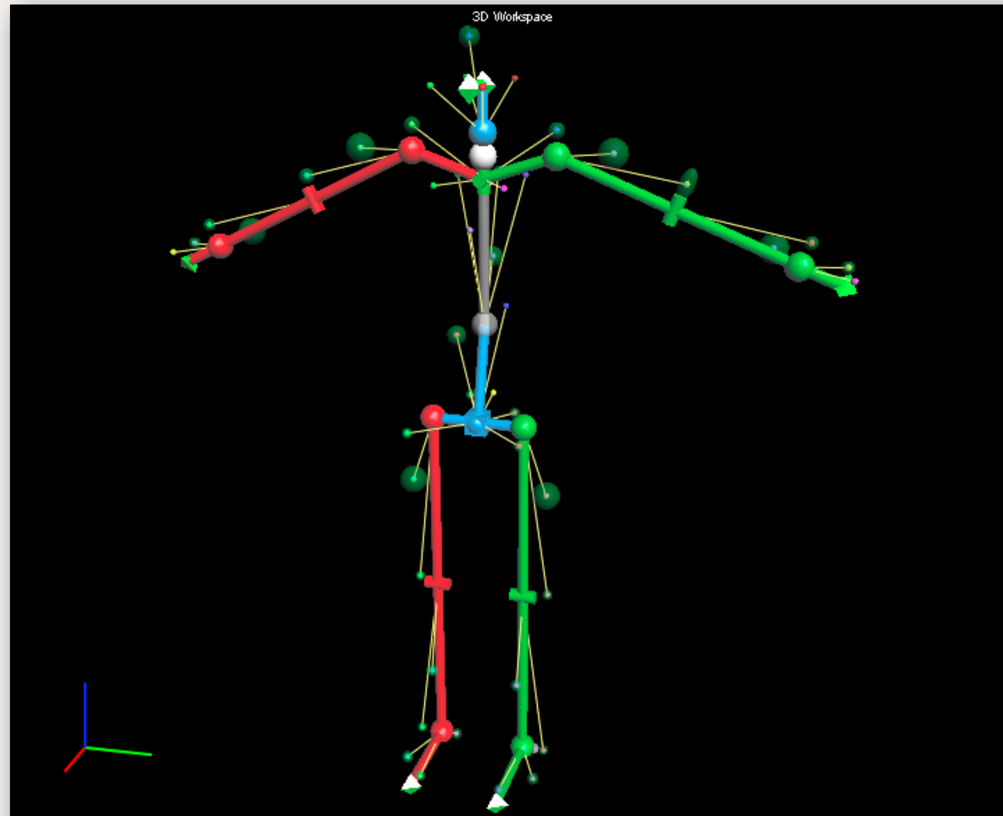


$$\omega_i = f_{i,\Omega}(\omega_{i-1})$$

$$\frac{dE}{d\Omega} = 2\sum_j \underbrace{(\hat{\mathbf{m}}_j^\star - \hat{\mathbf{m}}_j)^T}_{\text{vector}} \underbrace{\frac{d\hat{\mathbf{m}}_j}{d\Omega}}_{\text{matrix}}$$

$$\frac{d\hat{\mathbf{m}}_j}{d\Omega} = \underbrace{\frac{\partial \hat{\mathbf{m}}_j}{\partial \omega_i}}_{\text{matrix}} \left(\frac{\partial \omega_i}{\partial \Omega} + \frac{\partial \omega_i}{\partial \omega_{i-1}} \frac{\partial \omega_{i-1}}{\partial \Omega} + \frac{\partial \omega_i}{\partial \omega_{i-1}} \frac{\partial \omega_{i-1}}{\partial \omega_{i-2}} \frac{\partial \omega_{i-2}}{\partial \Omega} + \cdots \right)$$

Derivatives

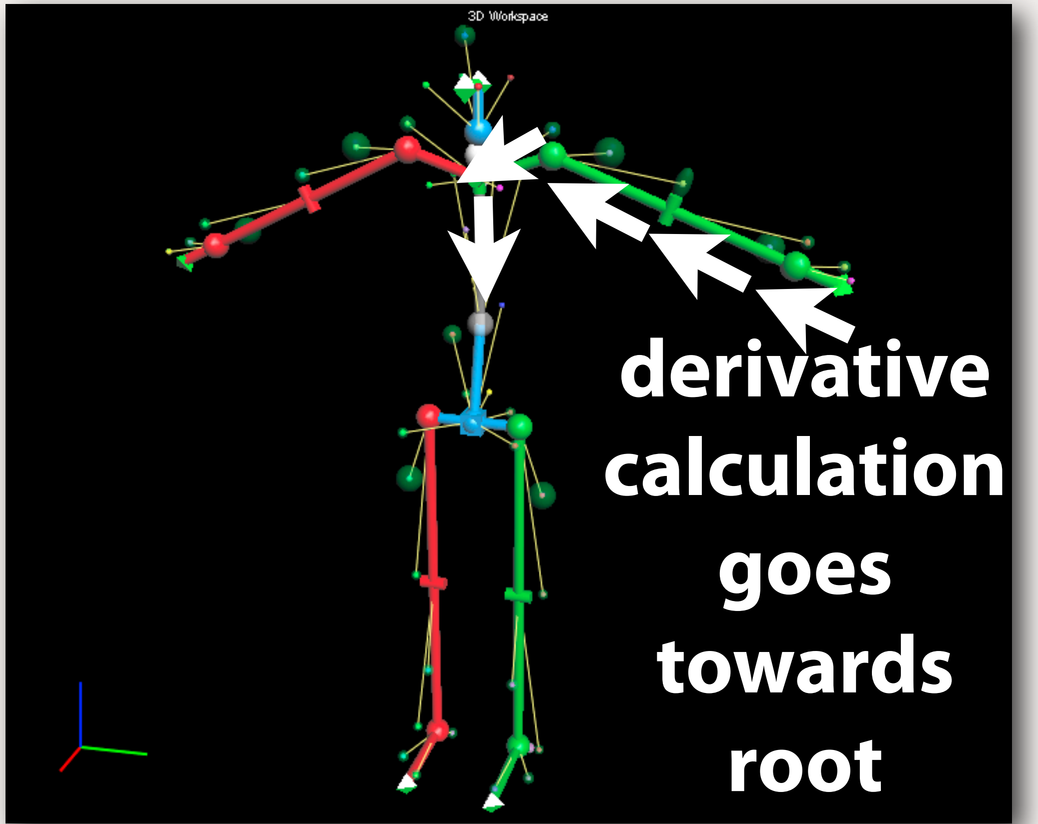


$$\omega_i = f_{i,\Omega}(\omega_{i-1})$$

$$\frac{dE}{d\Omega} = 2\sum_j \underbrace{(\hat{\mathbf{m}}_j^\star - \hat{\mathbf{m}}_j)^T}_{\text{vector}} \underbrace{\frac{d\hat{\mathbf{m}}_j}{d\Omega}}_{\text{matrix}}$$

$$\frac{d\hat{\mathbf{m}}_j}{d\Omega} = \underbrace{\frac{\partial\hat{\mathbf{m}}_j}{\partial\omega_i}}_{\text{matrix}} \underbrace{\left(\frac{\partial\omega_i}{\partial\Omega} + \frac{\partial\omega_i}{\partial\omega_{i-1}}\frac{\partial\omega_{i-1}}{\partial\Omega} + \frac{\partial\omega_i}{\partial\omega_{i-1}}\frac{\partial\omega_{i-1}}{\partial\omega_{i-2}}\frac{\partial\omega_{i-2}}{\partial\Omega} + \cdots\right)}_{\text{matrix multiplies}}$$

Derivatives



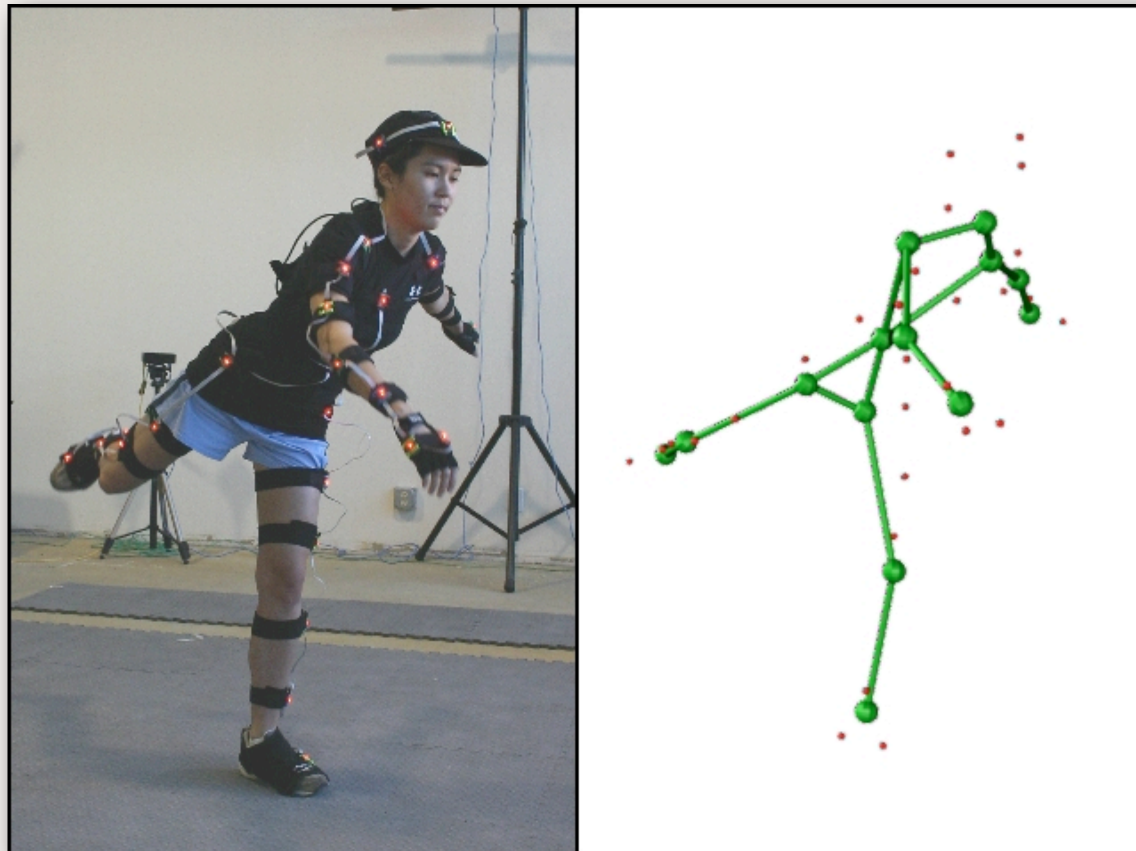
$$\omega_i = f_{i,\Omega}(\omega_{i-1})$$

$$\frac{dE}{d\Omega} = 2\sum_j \underbrace{(\hat{\mathbf{m}}_j^\star - \hat{\mathbf{m}}_j)^T}_{\text{vector}} \underbrace{\frac{d\hat{\mathbf{m}}_j}{d\Omega}}_{\text{matrix}}$$

$$\frac{d\hat{\mathbf{m}}_j}{d\Omega} = \underbrace{\frac{\partial\hat{\mathbf{m}}_j}{\partial\omega_i}}_{\text{matrix}} \left(\frac{\partial\omega_i}{\partial\Omega} + \frac{\partial\omega_i}{\partial\omega_{i-1}}\frac{\partial\omega_{i-1}}{\partial\Omega} + \frac{\partial\omega_i}{\partial\omega_{i-1}}\frac{\partial\omega_{i-1}}{\partial\omega_{i-2}}\frac{\partial\omega_{i-2}}{\partial\Omega} + \cdots \right)$$

matrix multiplies

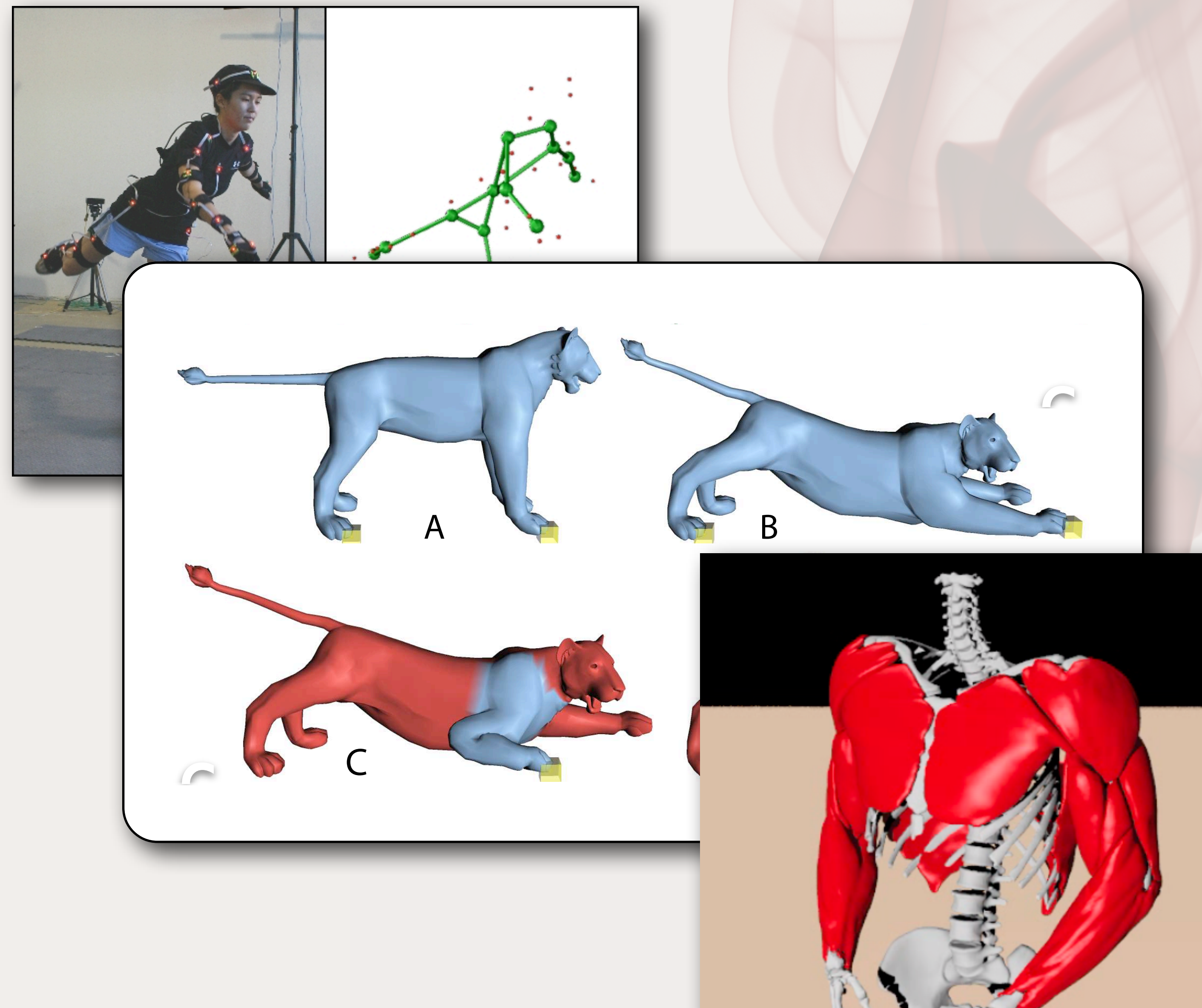
Inverse Kinematics Summary



- **Telescoping composition of functions from root.**
- **Compute derivatives in the *opposite* direction!**

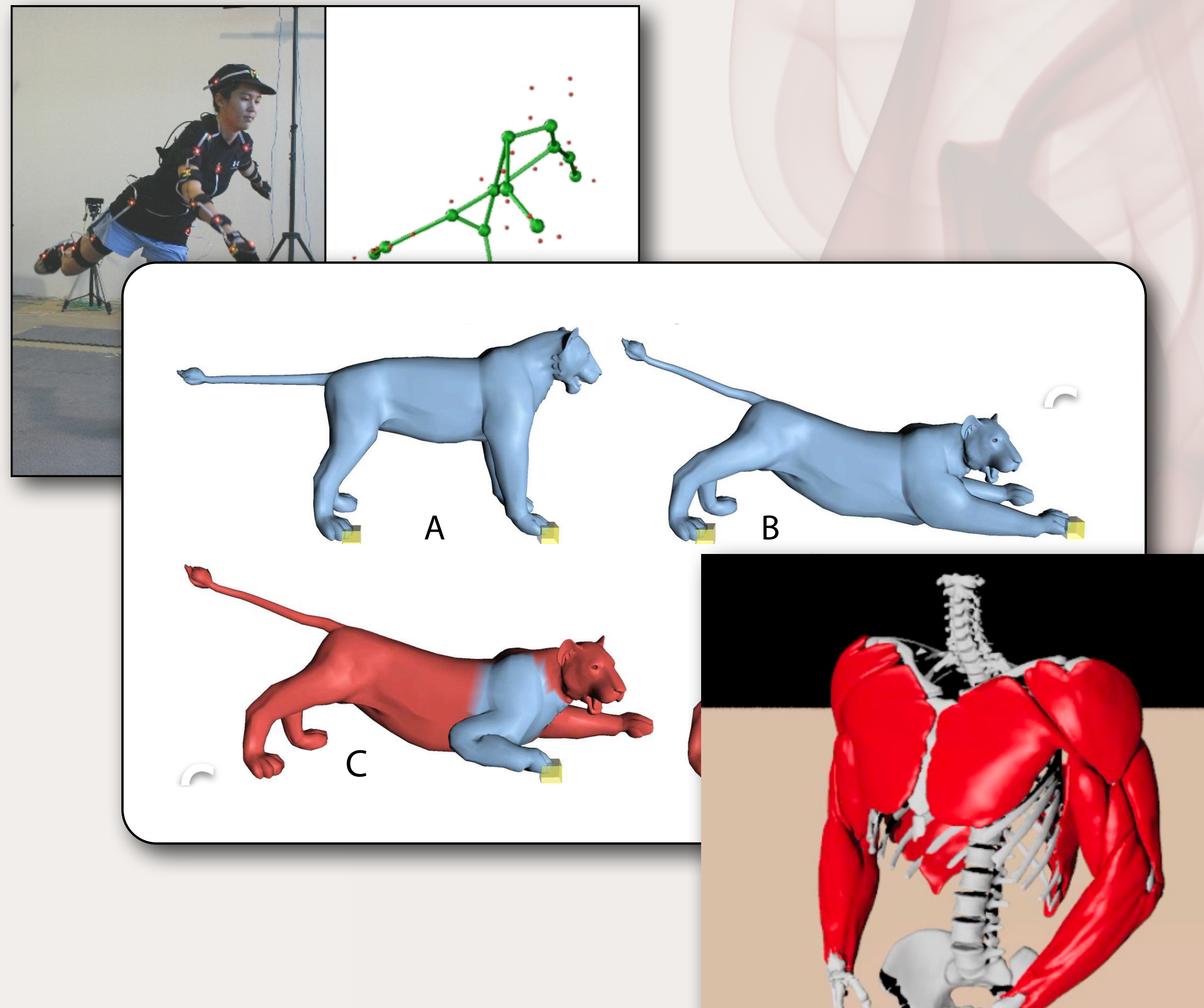
Body Representation

- **Kinematic Skeleton**
- **Anatomical**
- **Pure Mesh**
- **What are the advantages and disadvantages?**

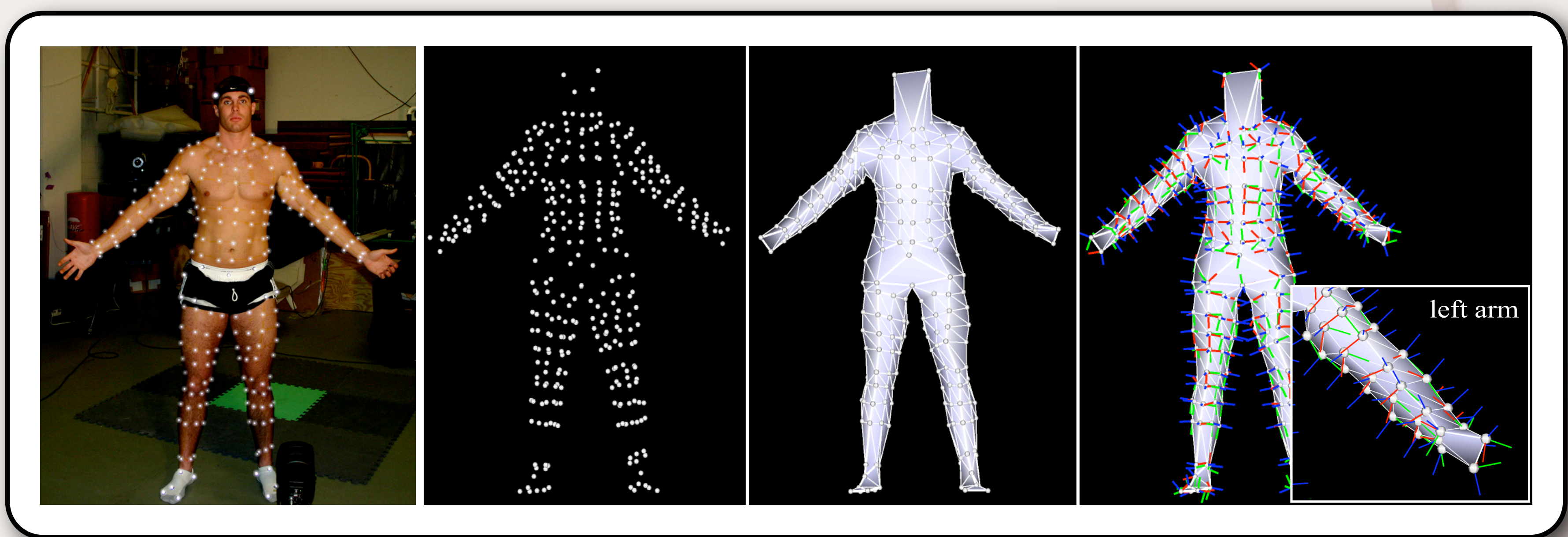


Body Representation

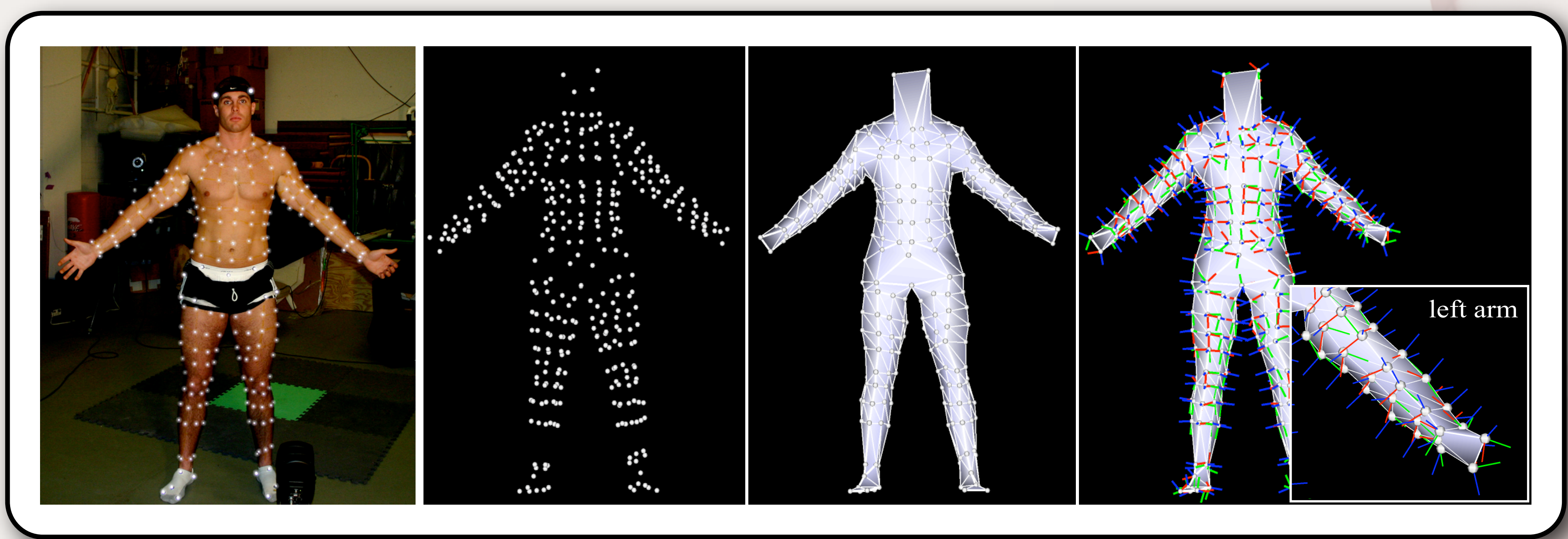
- Kinematic Skeleton
- Anatomical
- **Pure Mesh**
- What are the advantages and disadvantages?

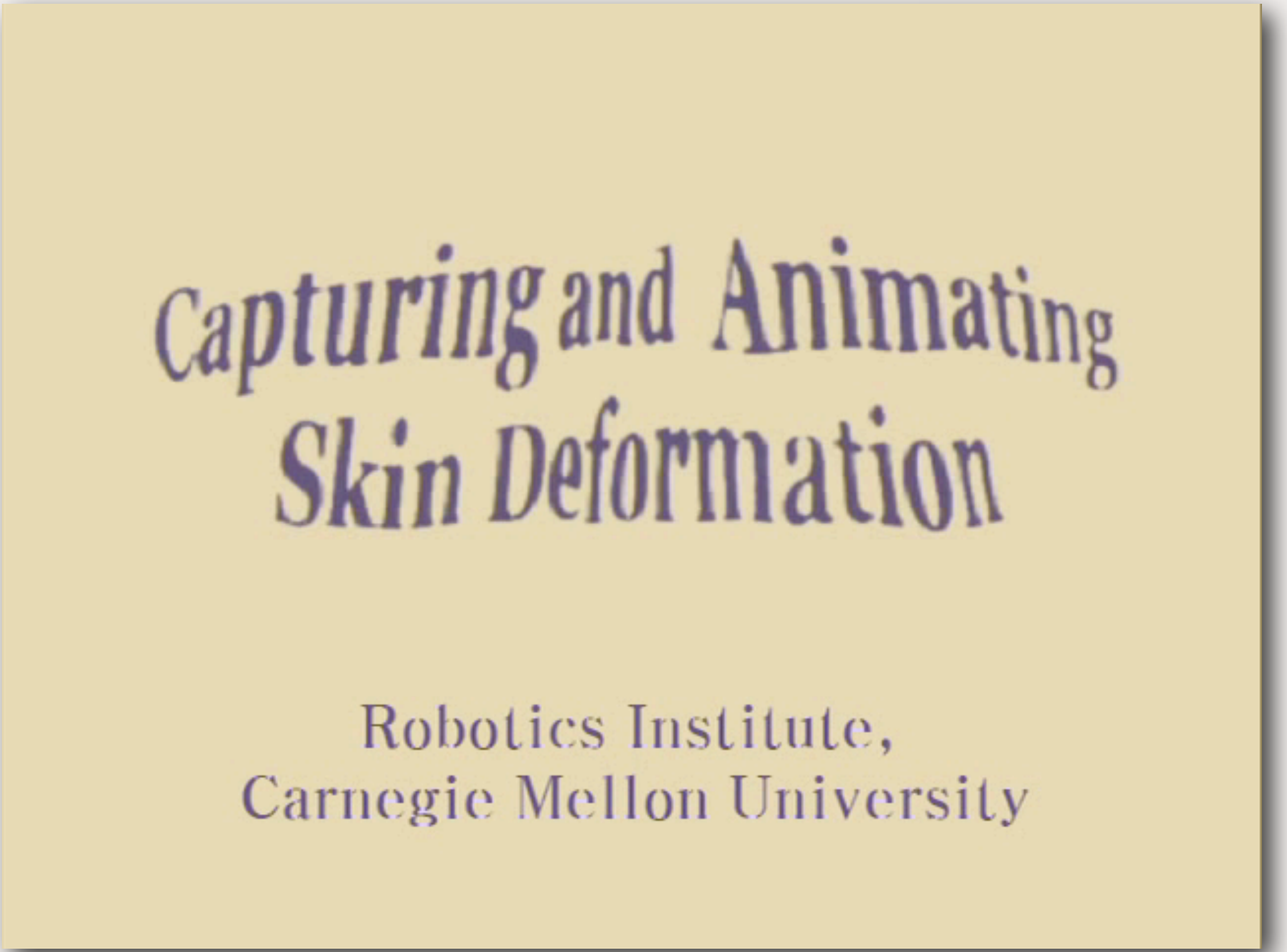


Dense Marker Capture



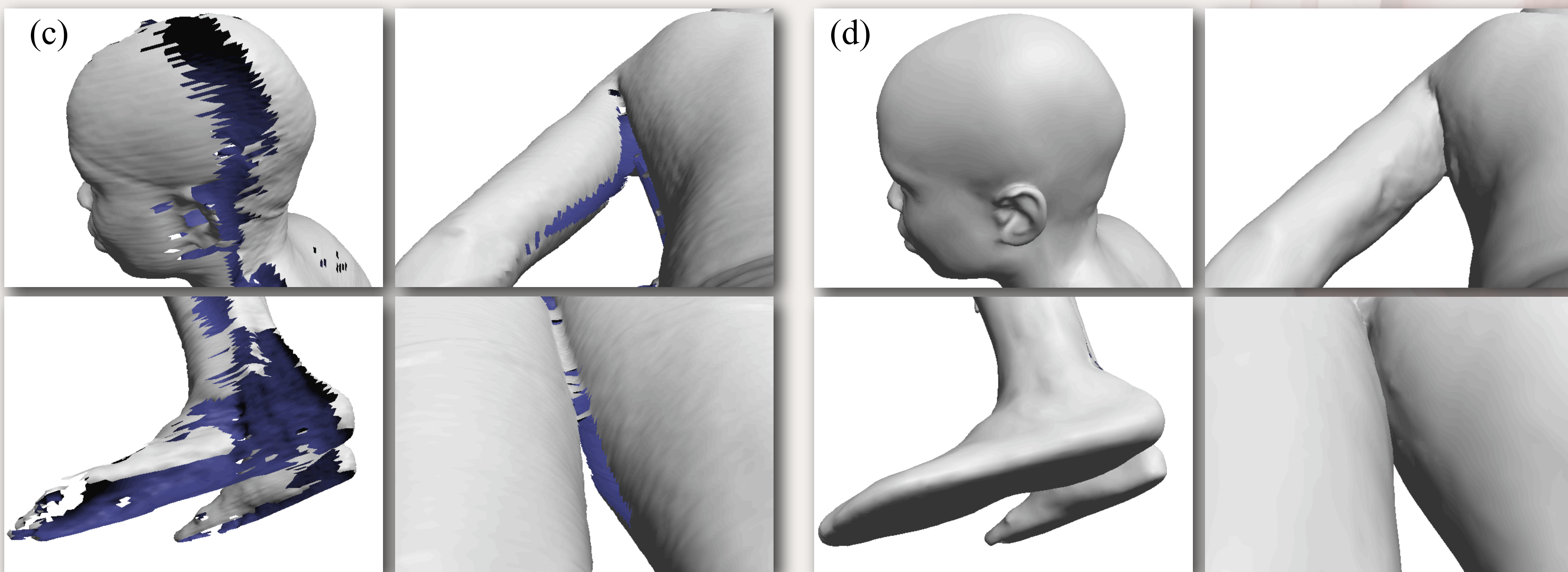
Dense Marker Capture





Laser Range Scanning

Filling in Missing Data

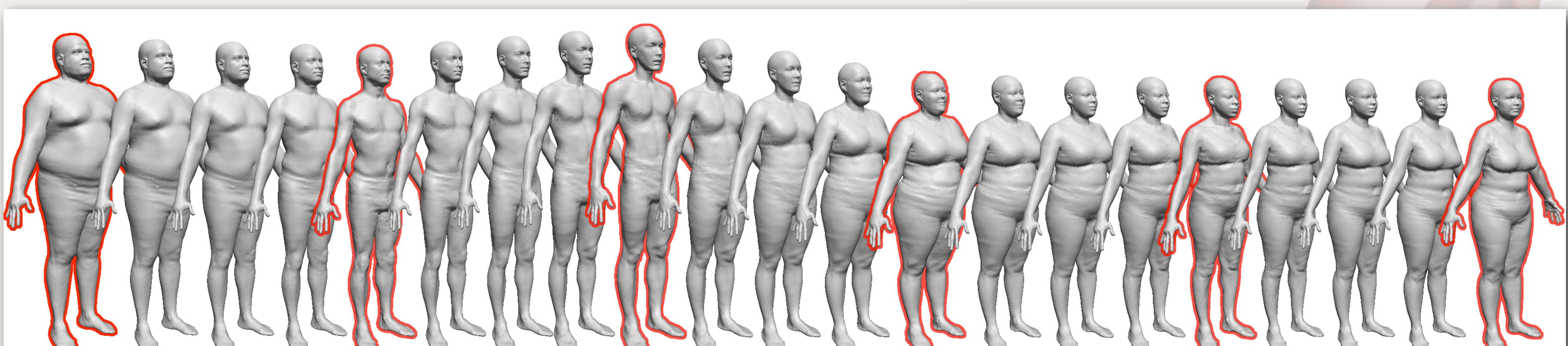


source: Allen, Curless, Popović. The space of human body shapes: reconstruction and parameterization from range scans.

How could this be accomplished?

What can you do with a huge set of human meshes in vertex correspondence?

What can you do with a huge set of human meshes in vertex correspondence?



source: Allen, Curless, Popović. The space of human body shapes: reconstruction and parameterization from range scans.

PCA Shape Analysis

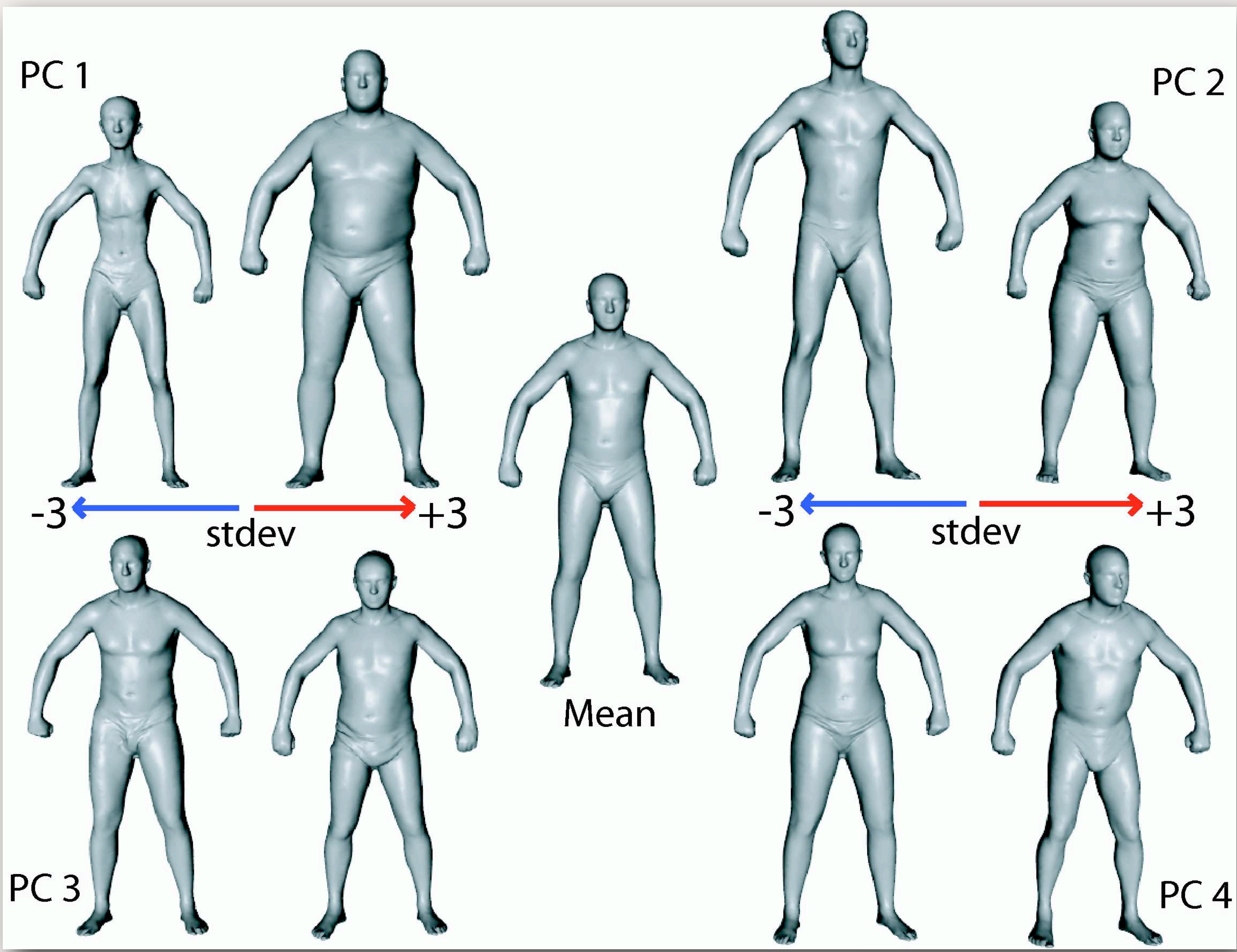
Displacements of Example Poses

PCA Shape Analysis

Displacements of Example Poses = PCA Basis Coefficient Matrix

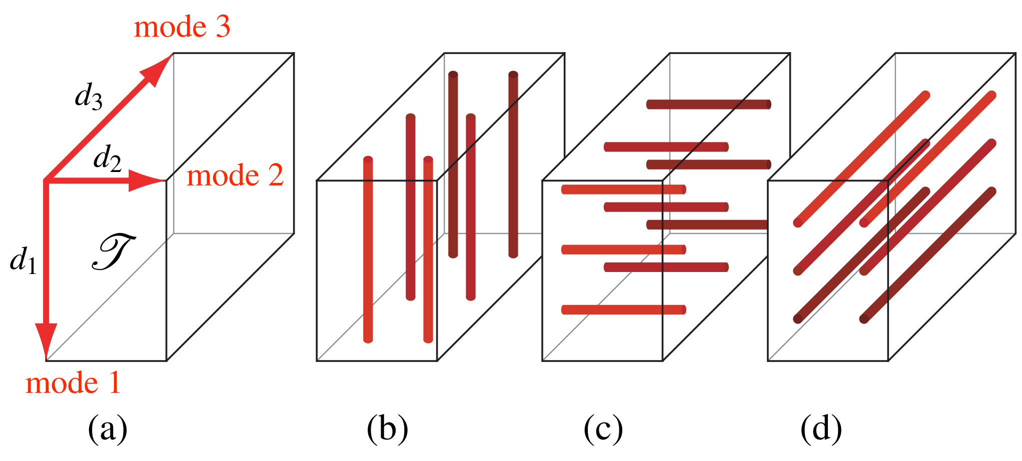
PCA Shape Analysis

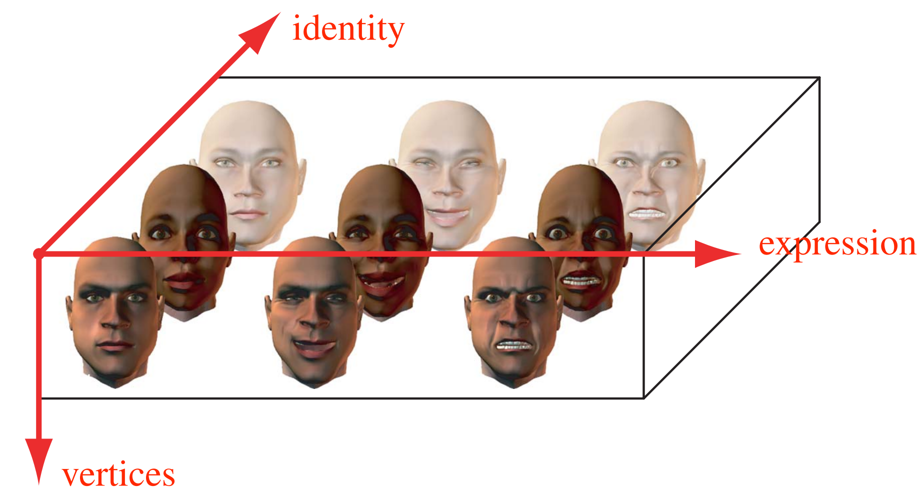
Displacements of Example Poses = PCA Basis Coefficient Matrix



source: Anguelov, Srinivasan, Koller, Thrun, Rodgers, Davis. SCAPE: Shape Completion and Animation of People.

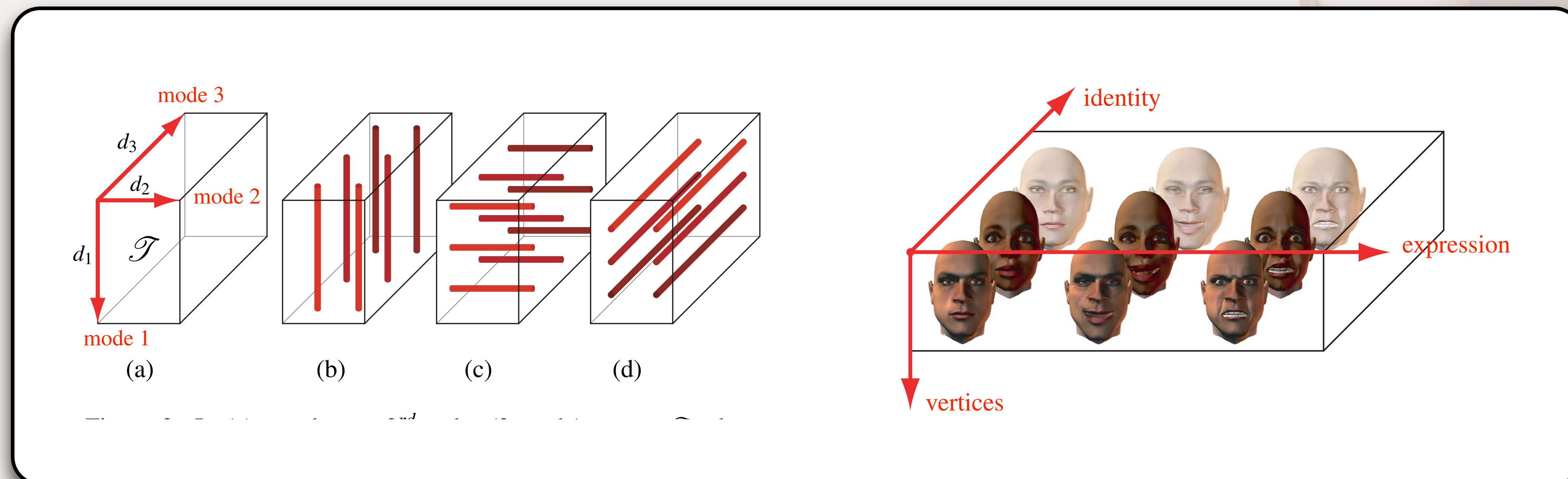
Multilinear Analysis



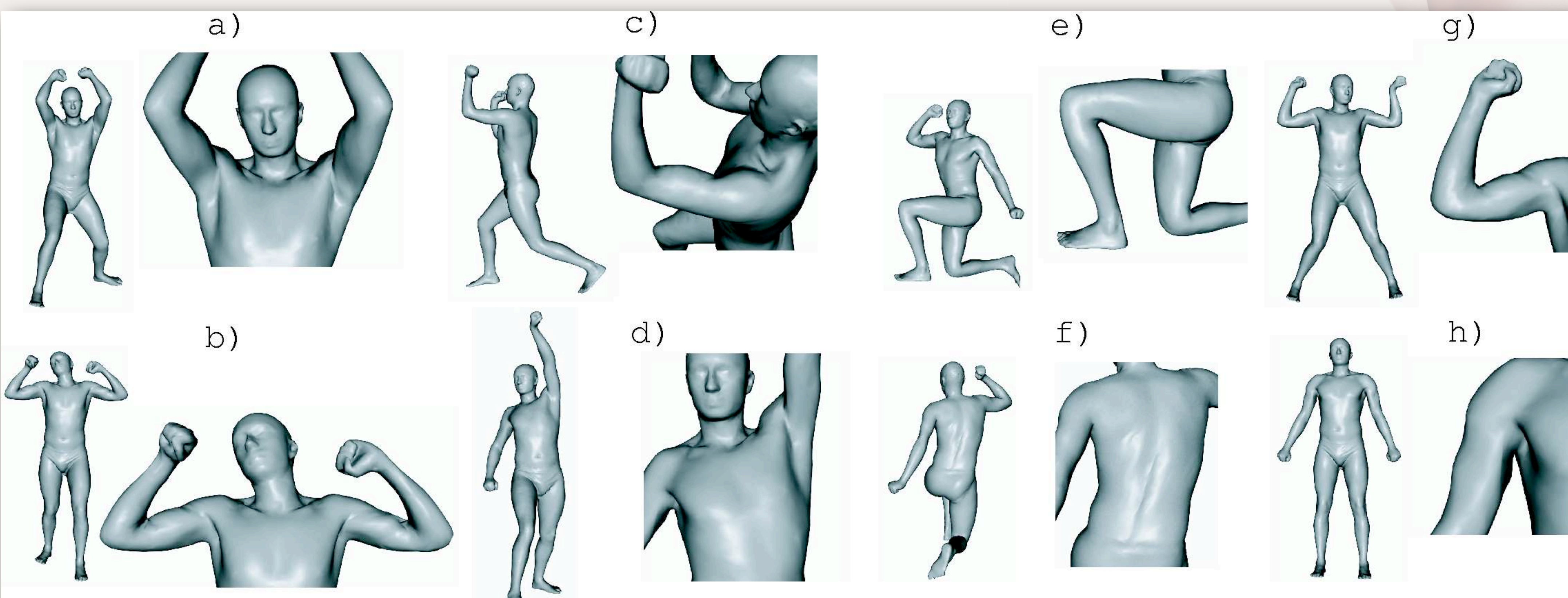


source: Vlasic, Brand, Pfister, Popović. Face Transfer with Multilinear Models.

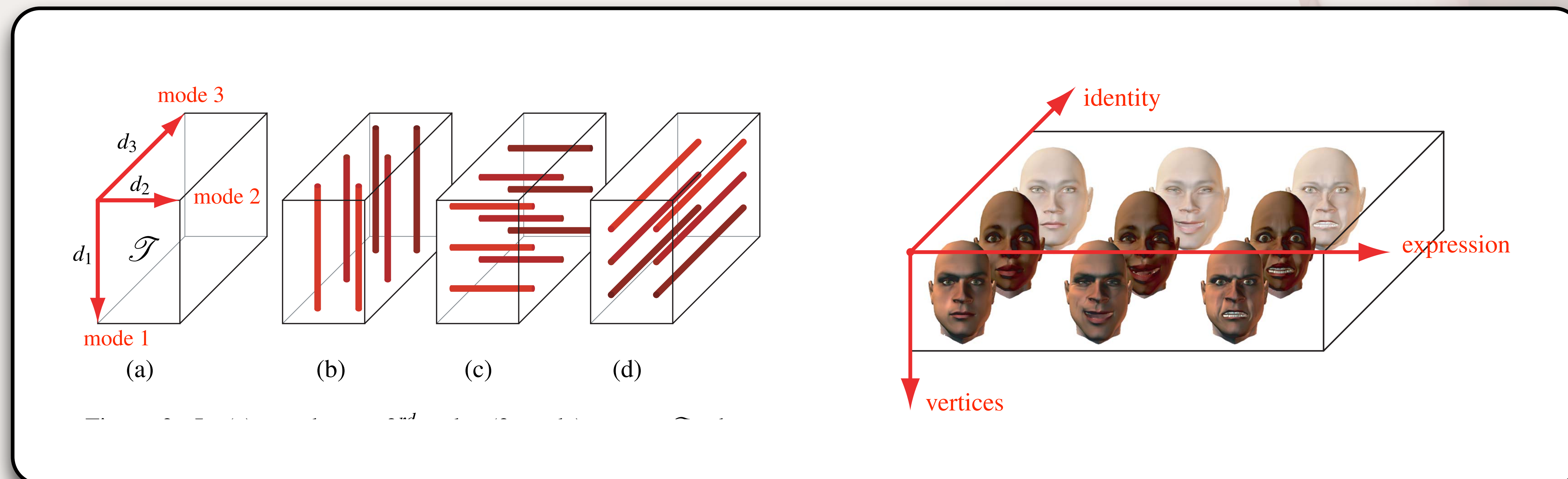
Multilinear Analysis



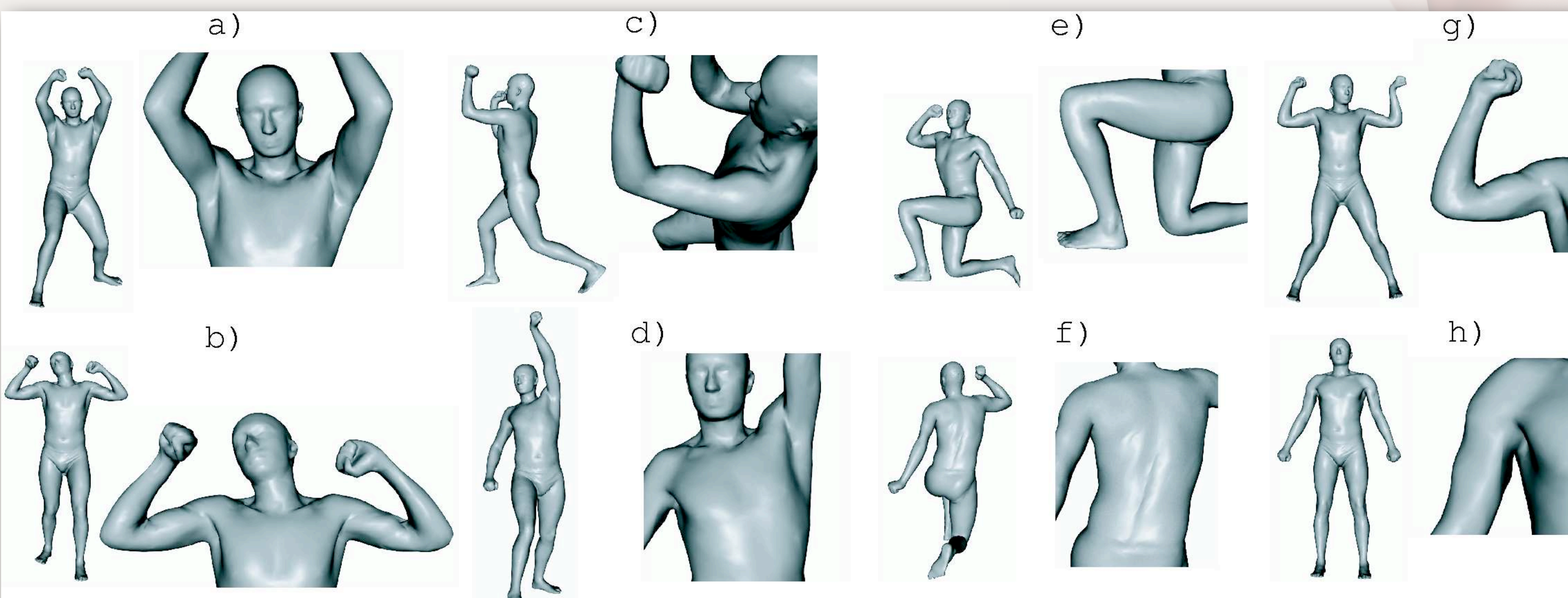
source: Vlasic, Brand, Pfister, Popović. Face Transfer with Multilinear Models.



Multilinear Analysis

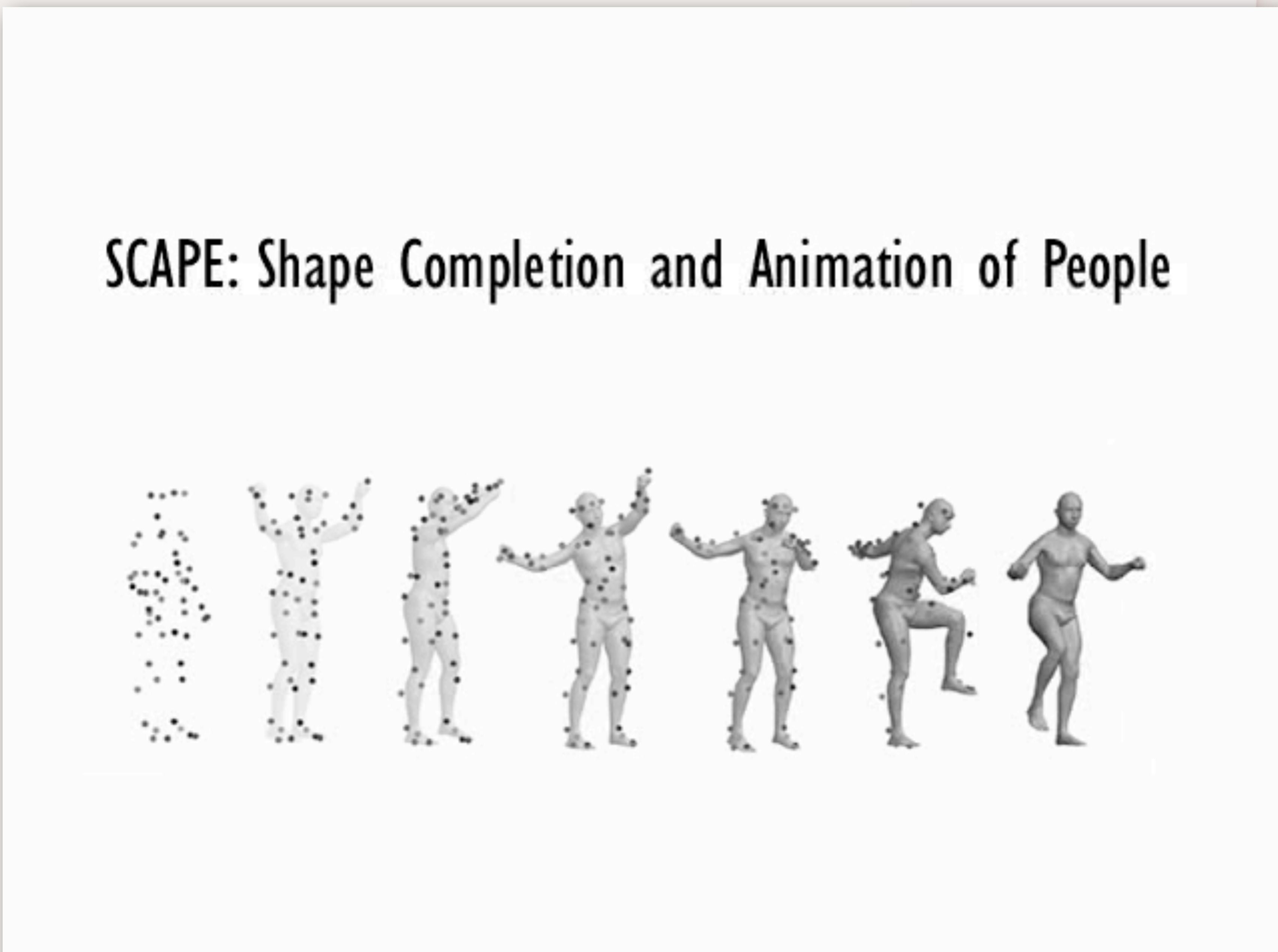


source: Vlasic, Brand, Pfister, Popović. Face Transfer with Multilinear Models.



source: Anguelov, Srinivasan, Koller, Thrun, Rodgers, Davis. SCAPE: Shape Completion and Animation of People.

Example



source: Anguelov, Srinivasan, Koller, Thrun, Rodgers, Davis. SCAPE: Shape Completion and Animation of People.

Overview

- State of the art.
- **Body models.**
- Animation
- Questions

source: 3dscience.com

Overview

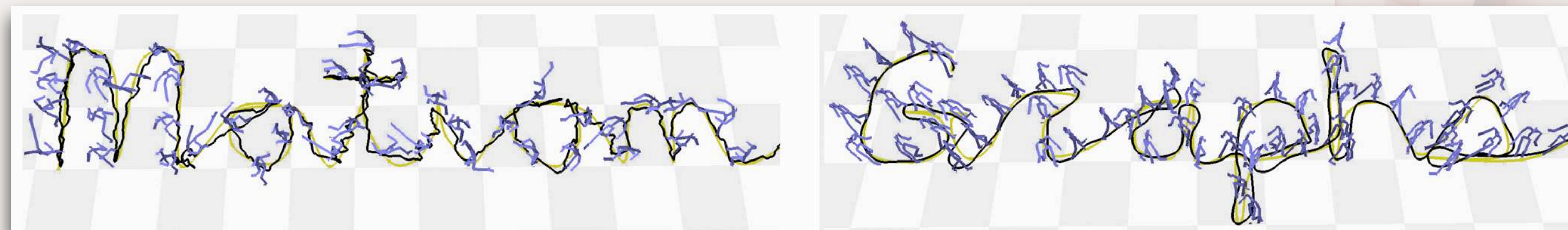
- State of the art.
- Body models.
- **Animation**
- Questions

source: 3dscience.com

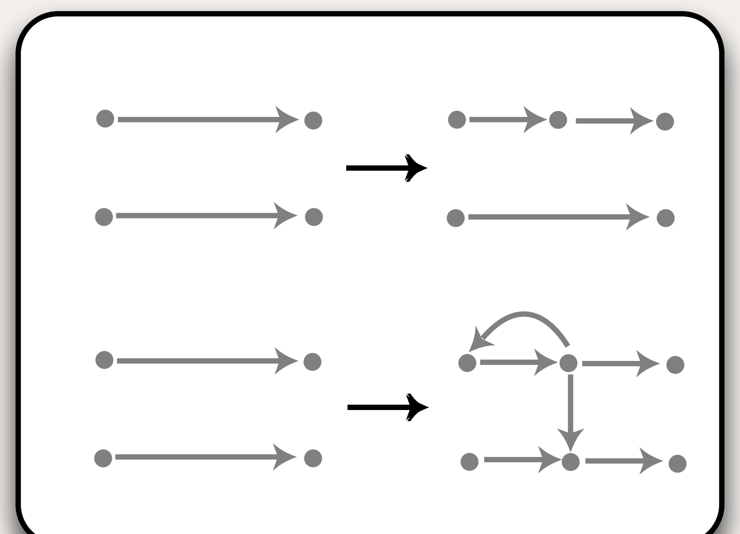
Overview

- **Data-Driven Motion**
- **Physics Based Motion**
- **Motion of other Animals**

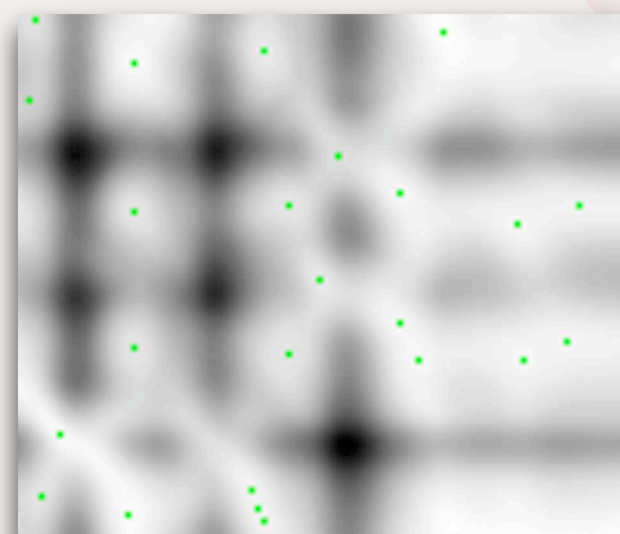
Data Driven Human Animation



source: Kovar, Gleicher, Pighin. Motion Graphs.

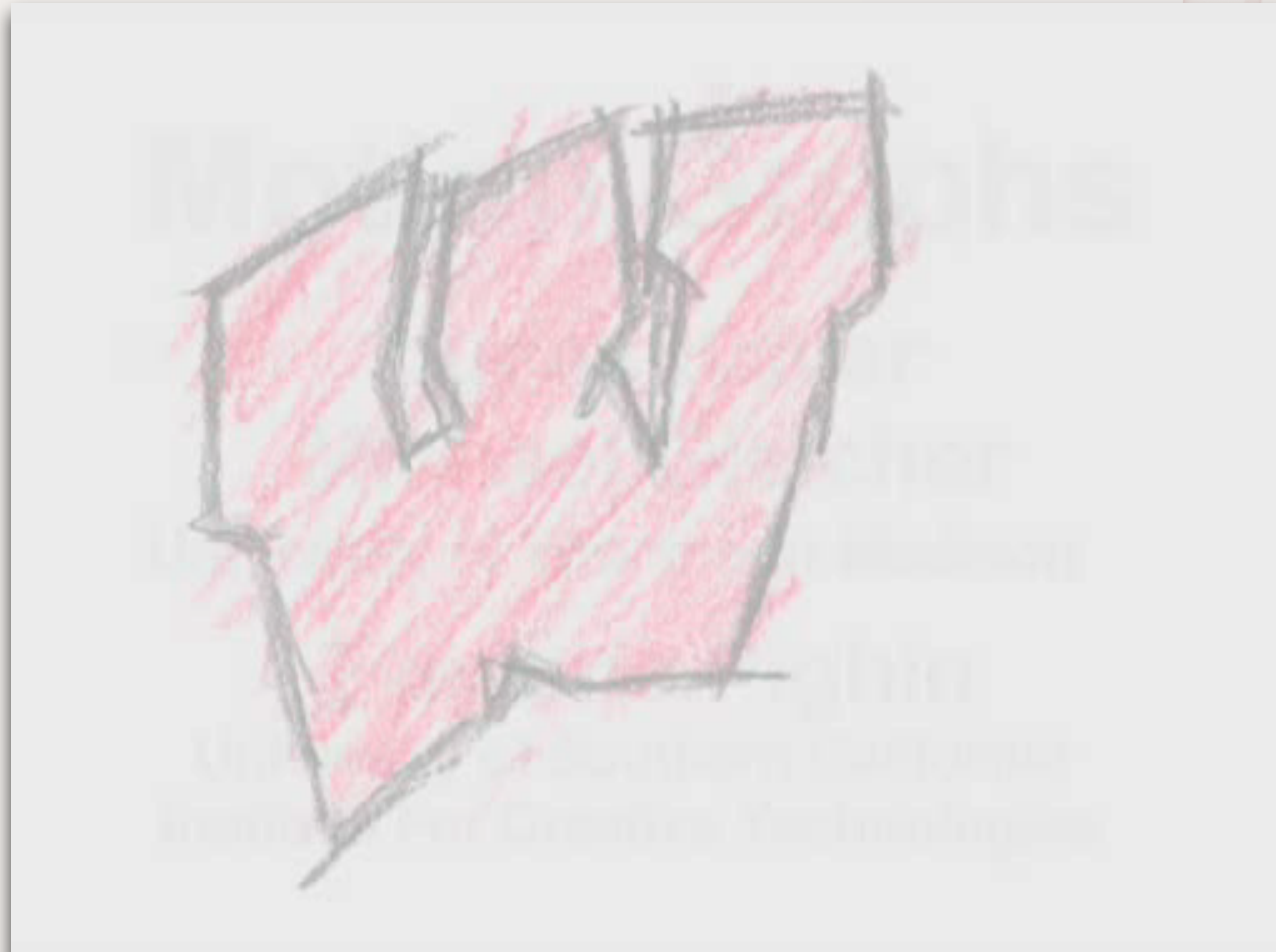


Motion Graph Schematic



Finding Candidate Transitions

Examples

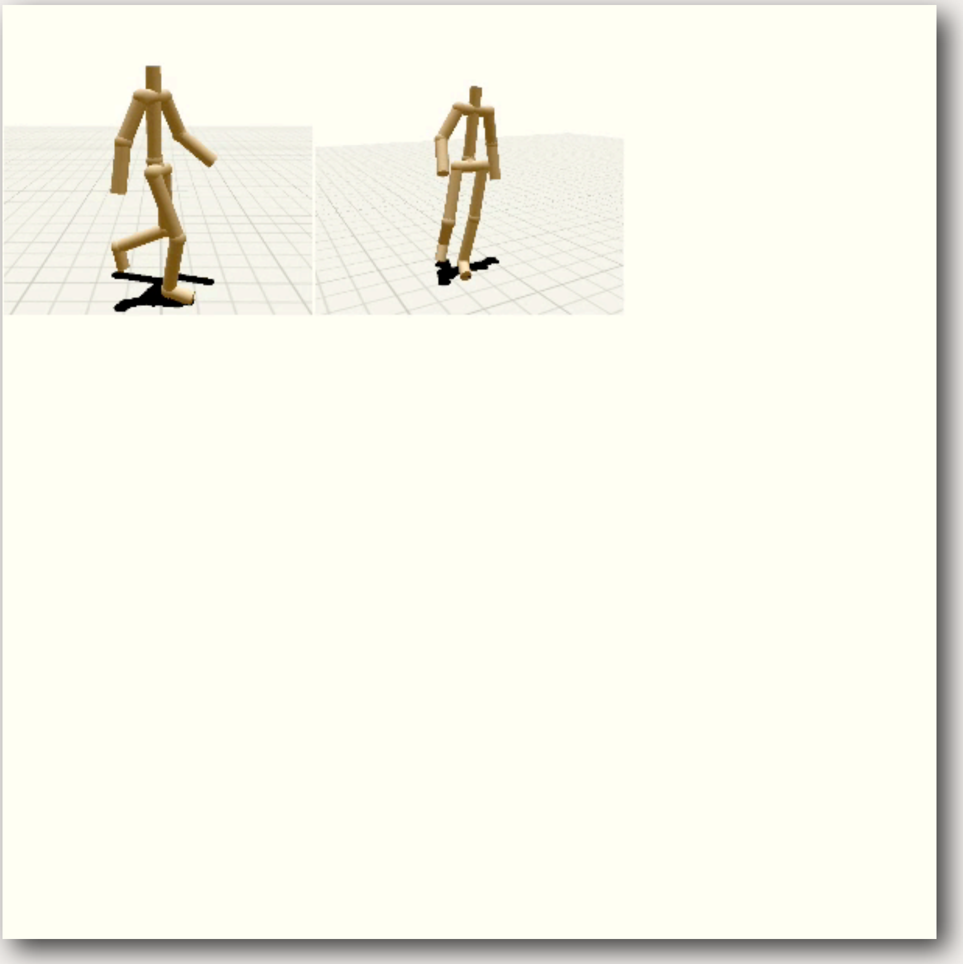


Clips

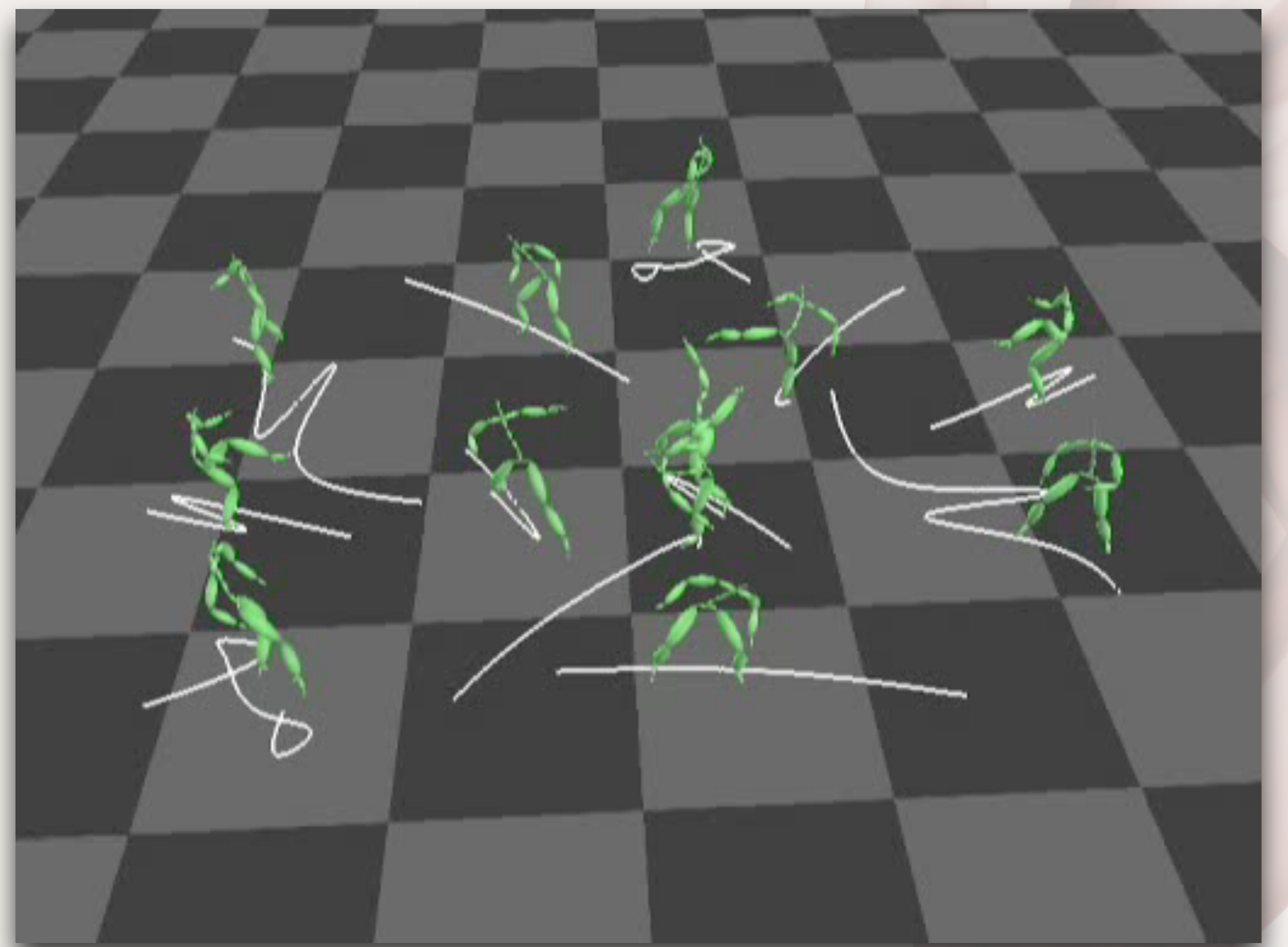
source: Treuille et al. [2002]

source: Kovar et al. [2002]

Clips

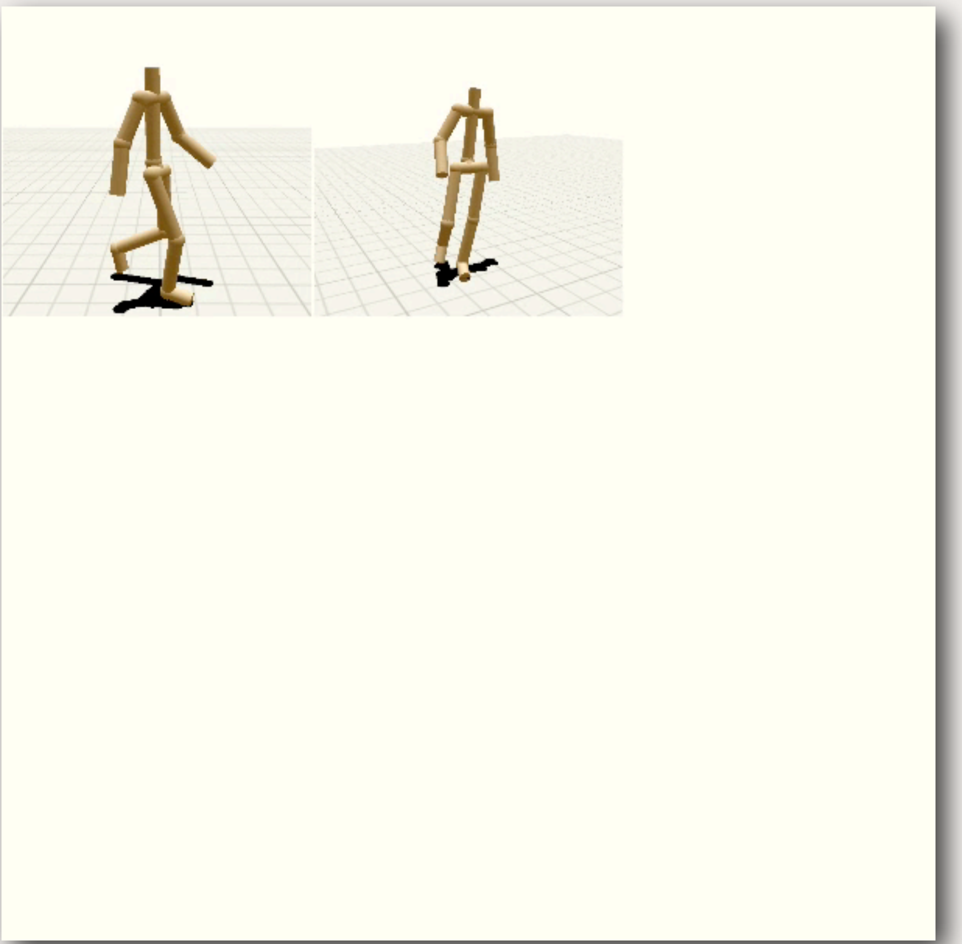


source: Treuille et al. [2002]

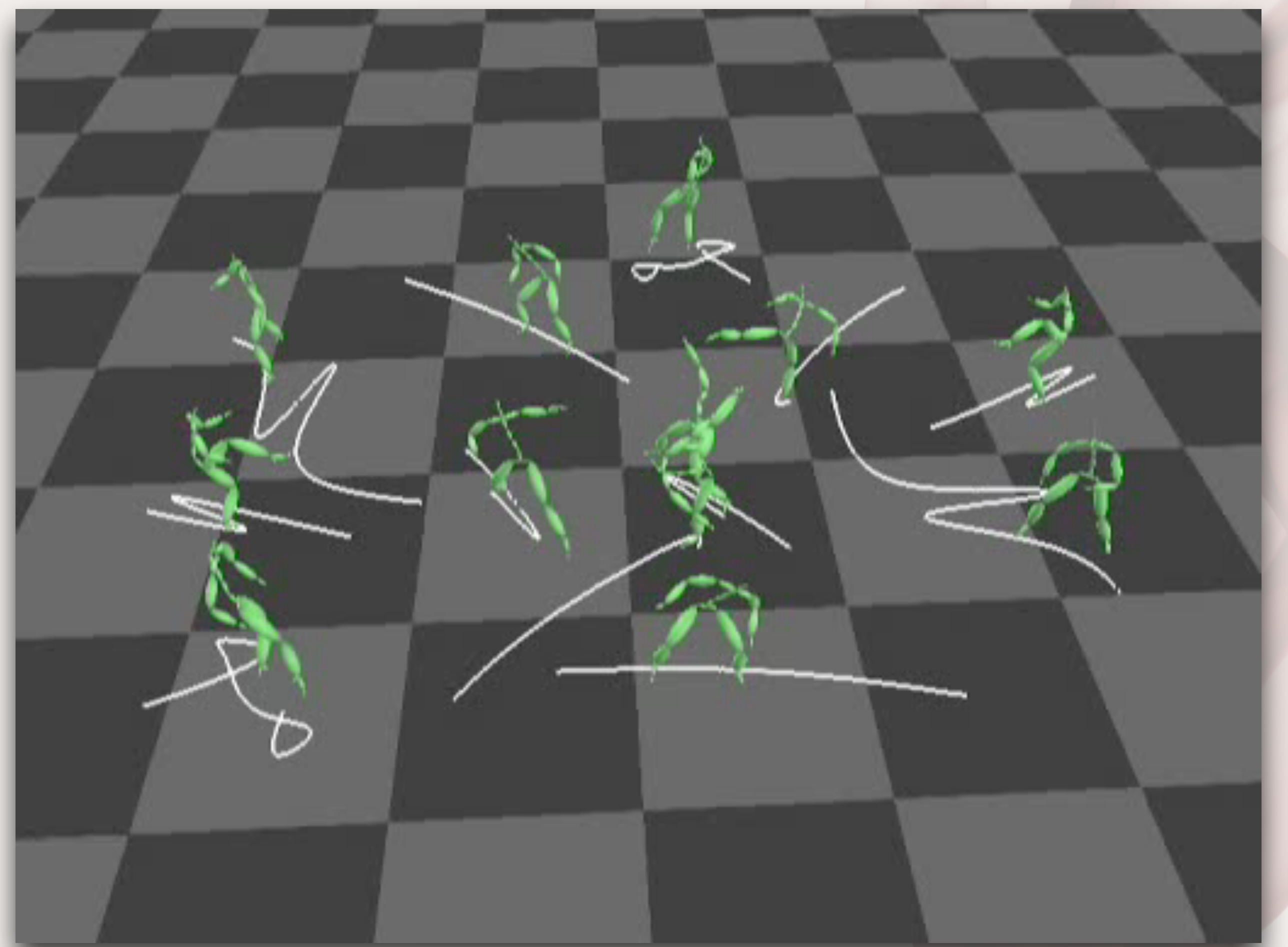


source: Kovar et al. [2002]

Clips



source: Treuille et al. [2002]



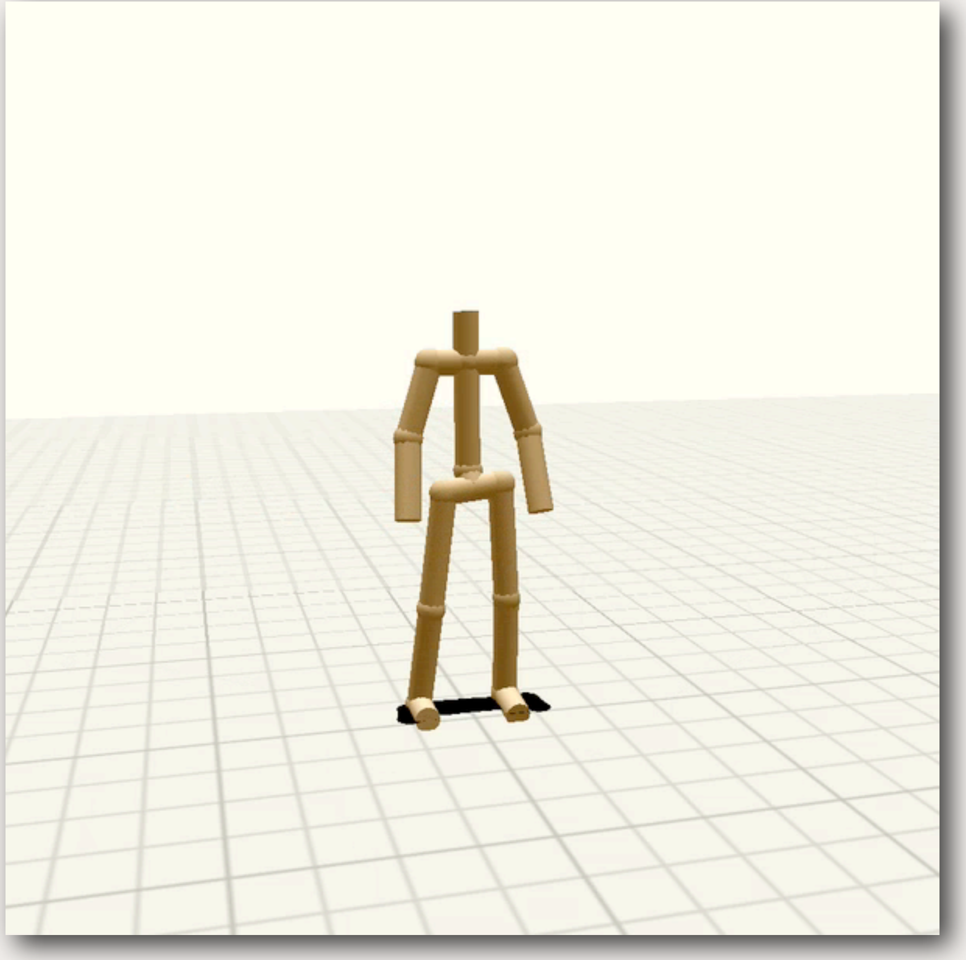
source: Kovar et al. [2002]

Sequences

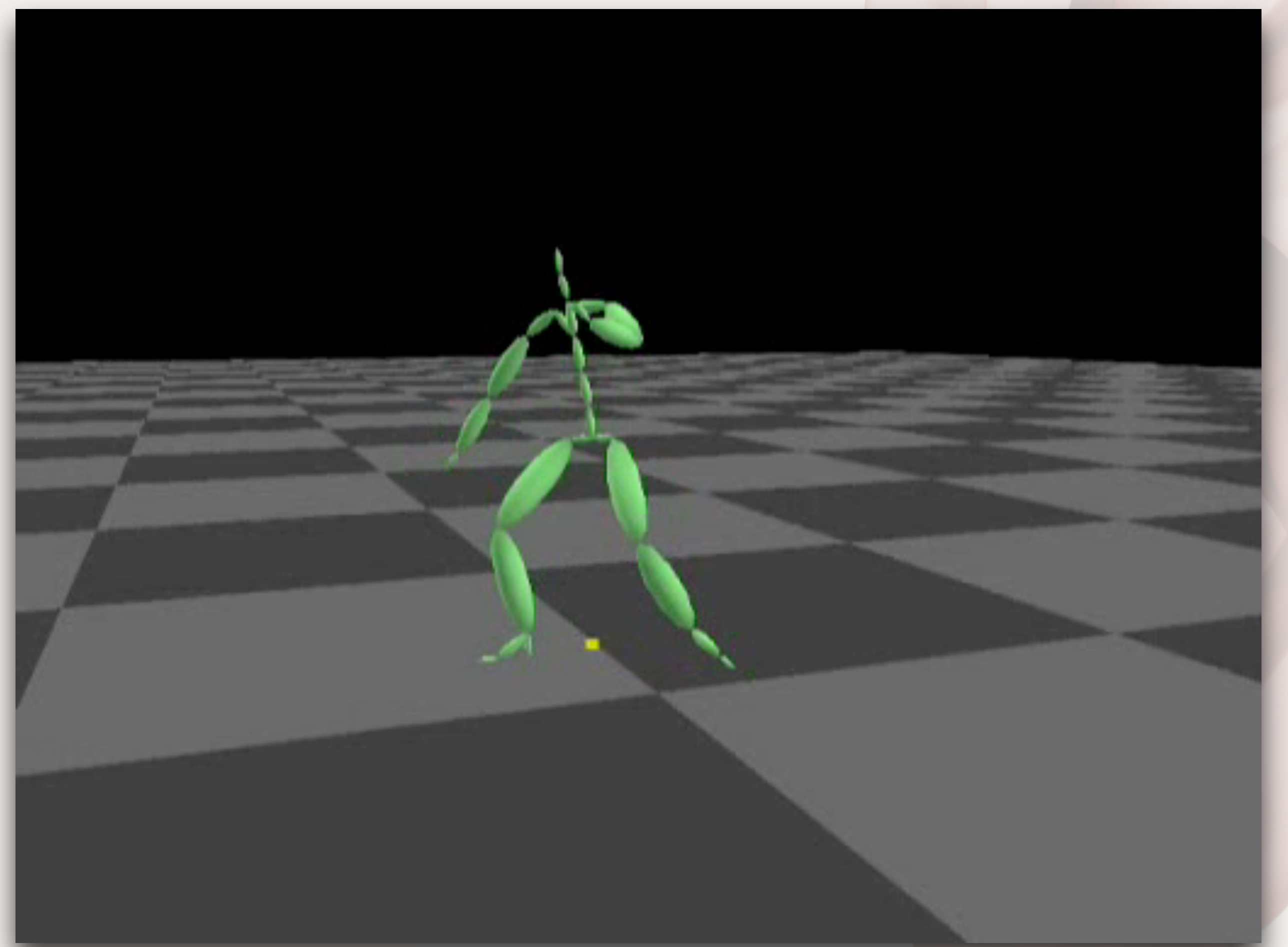
source: Treuille et al. [2002]

source: Kovar et al. [2002]

Sequences

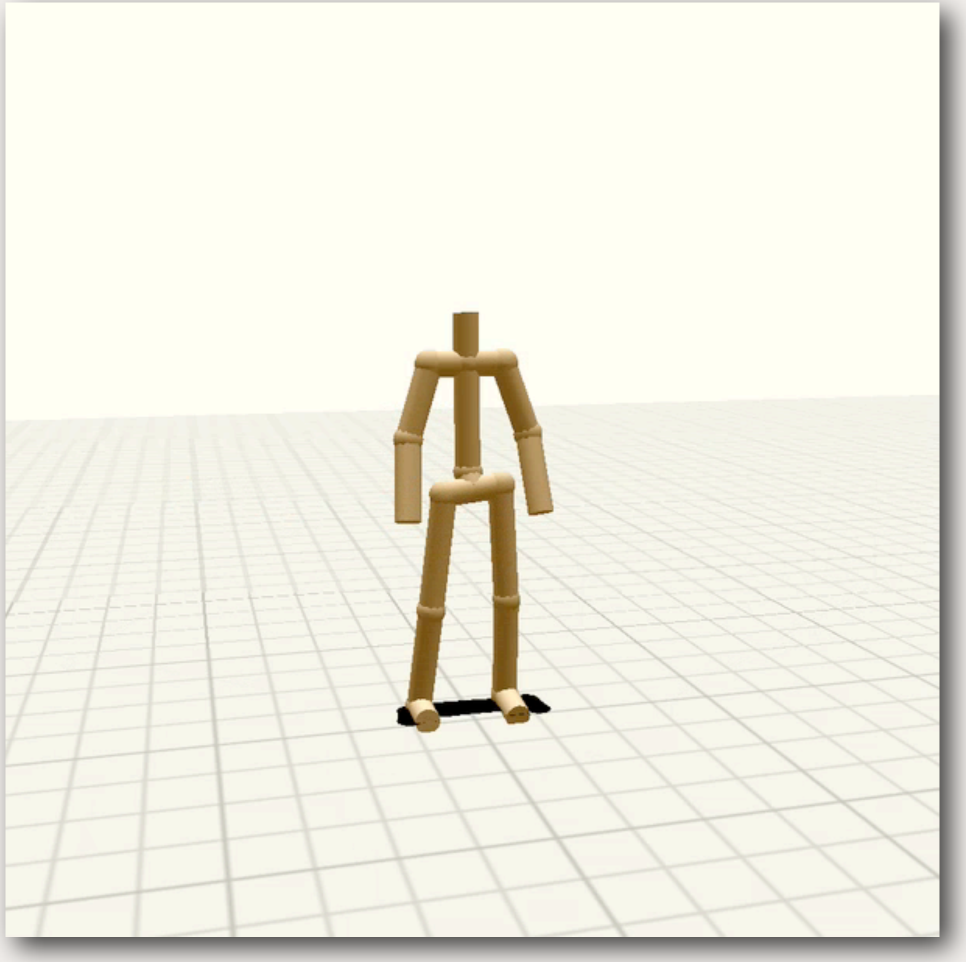


source: Treuille et al. [2002]

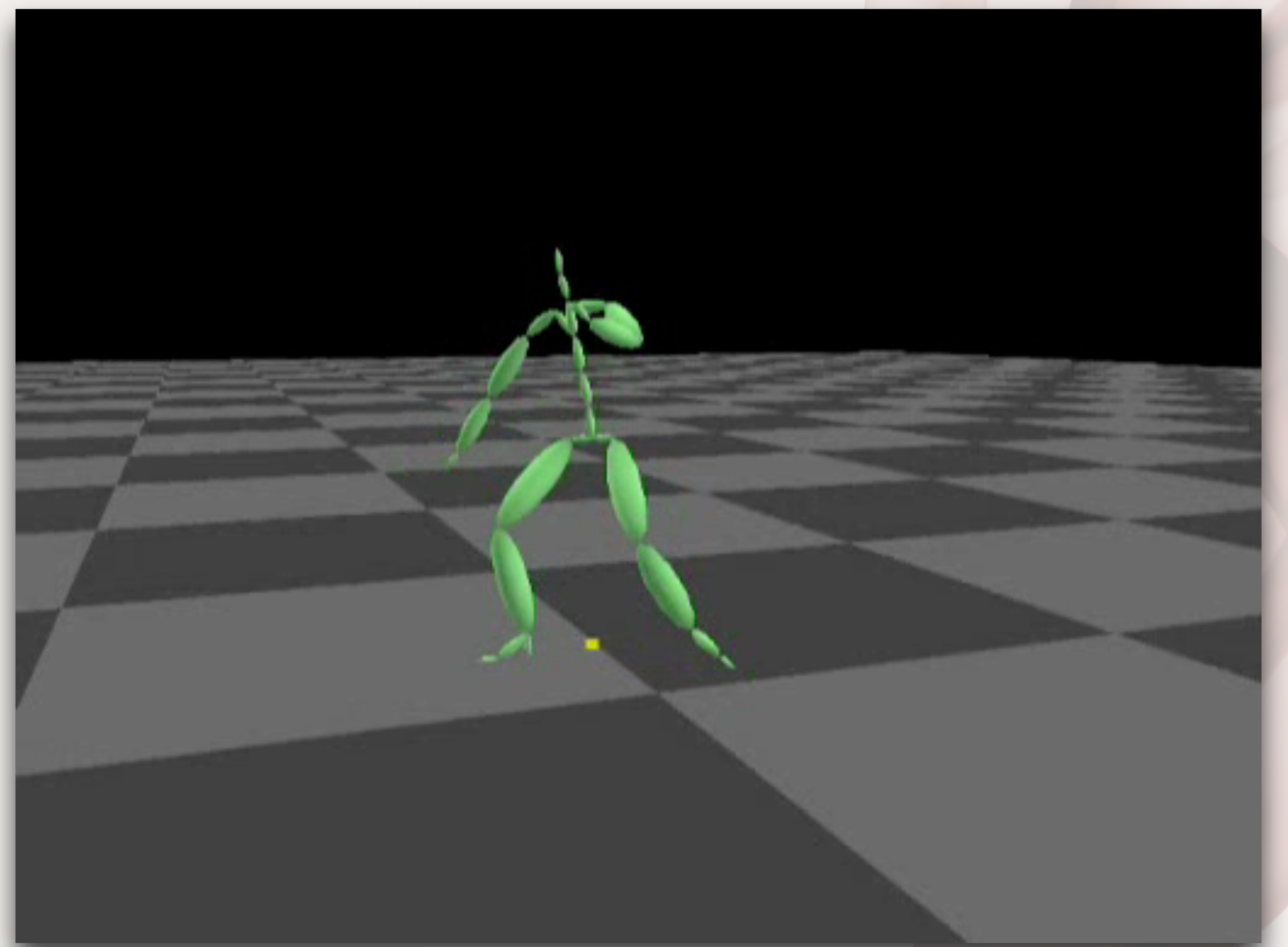


source: Kovar et al. [2002]

Sequences

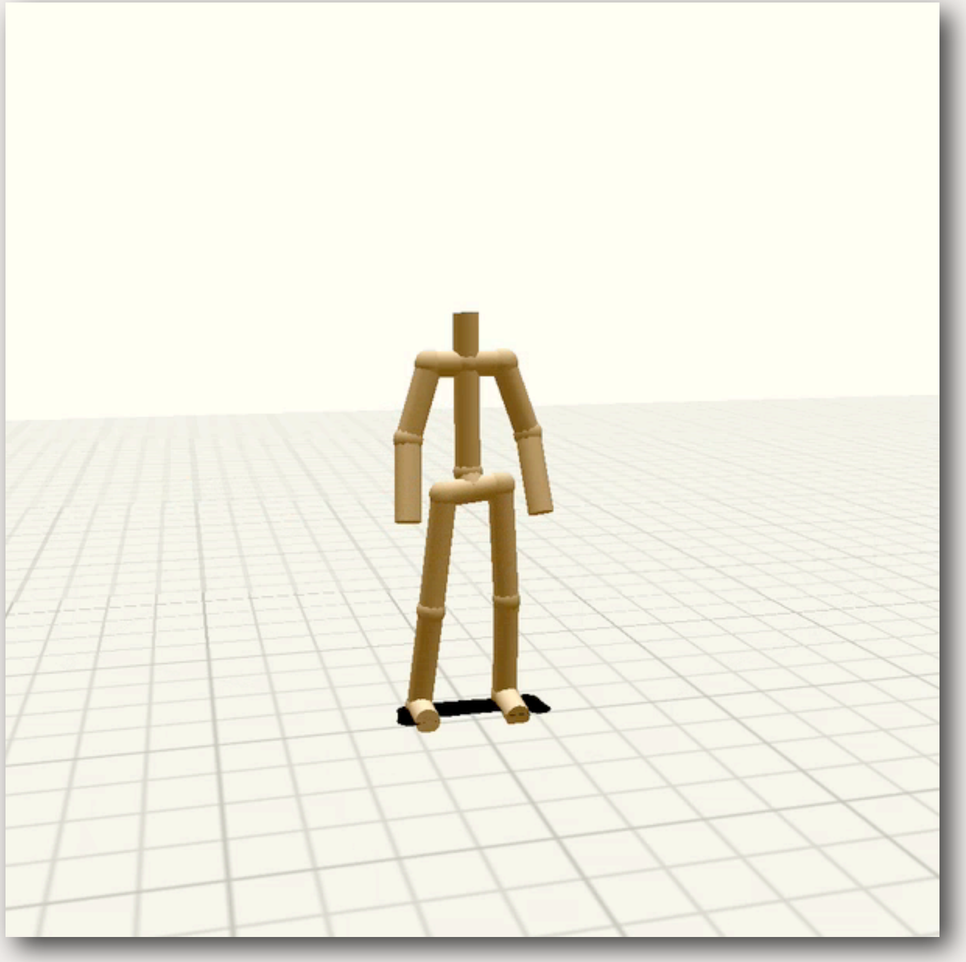


source: Treuille et al. [2002]

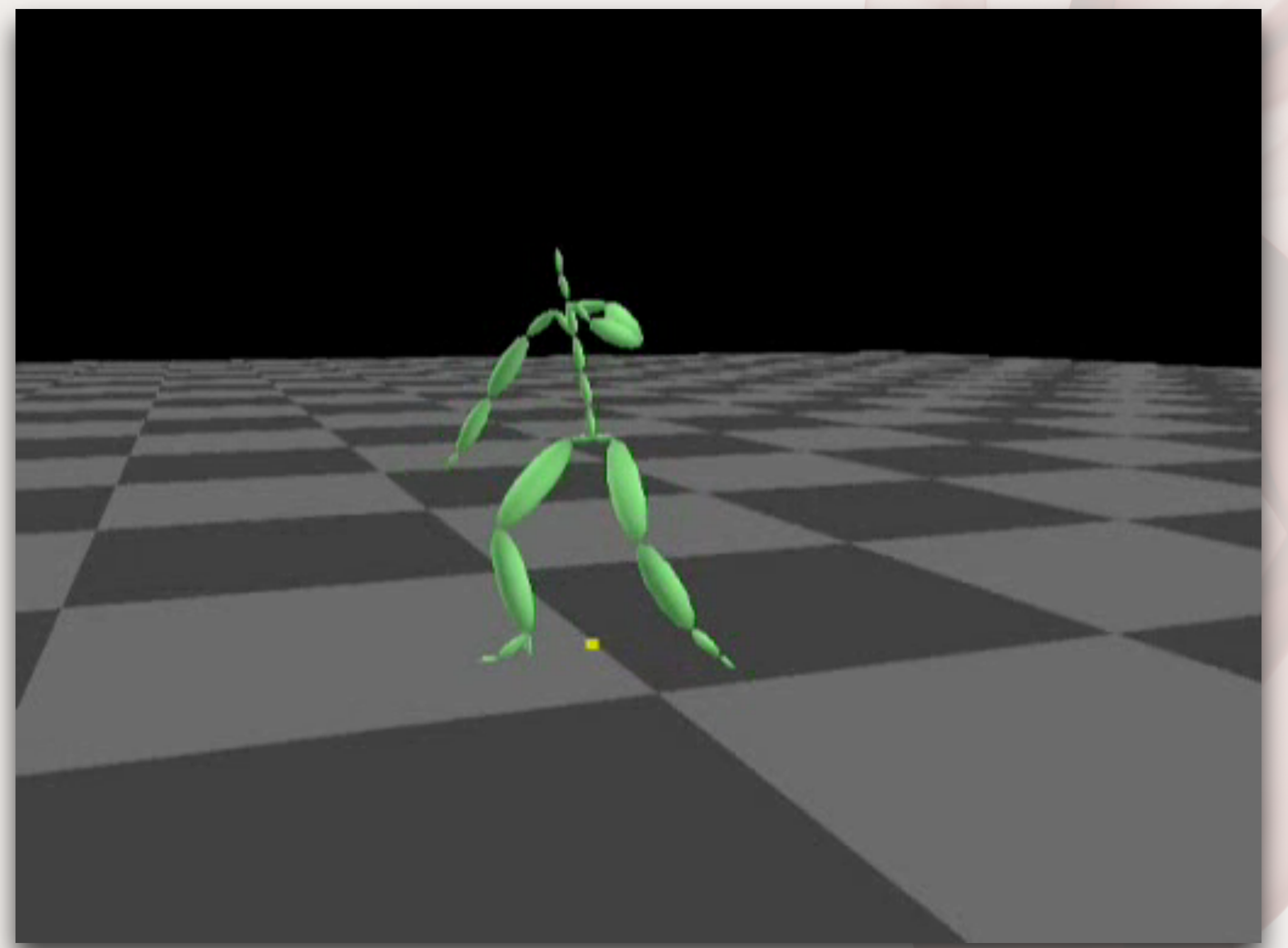


source: Kovar et al. [2002]

Sequences



source: Treuille et al. [2002]

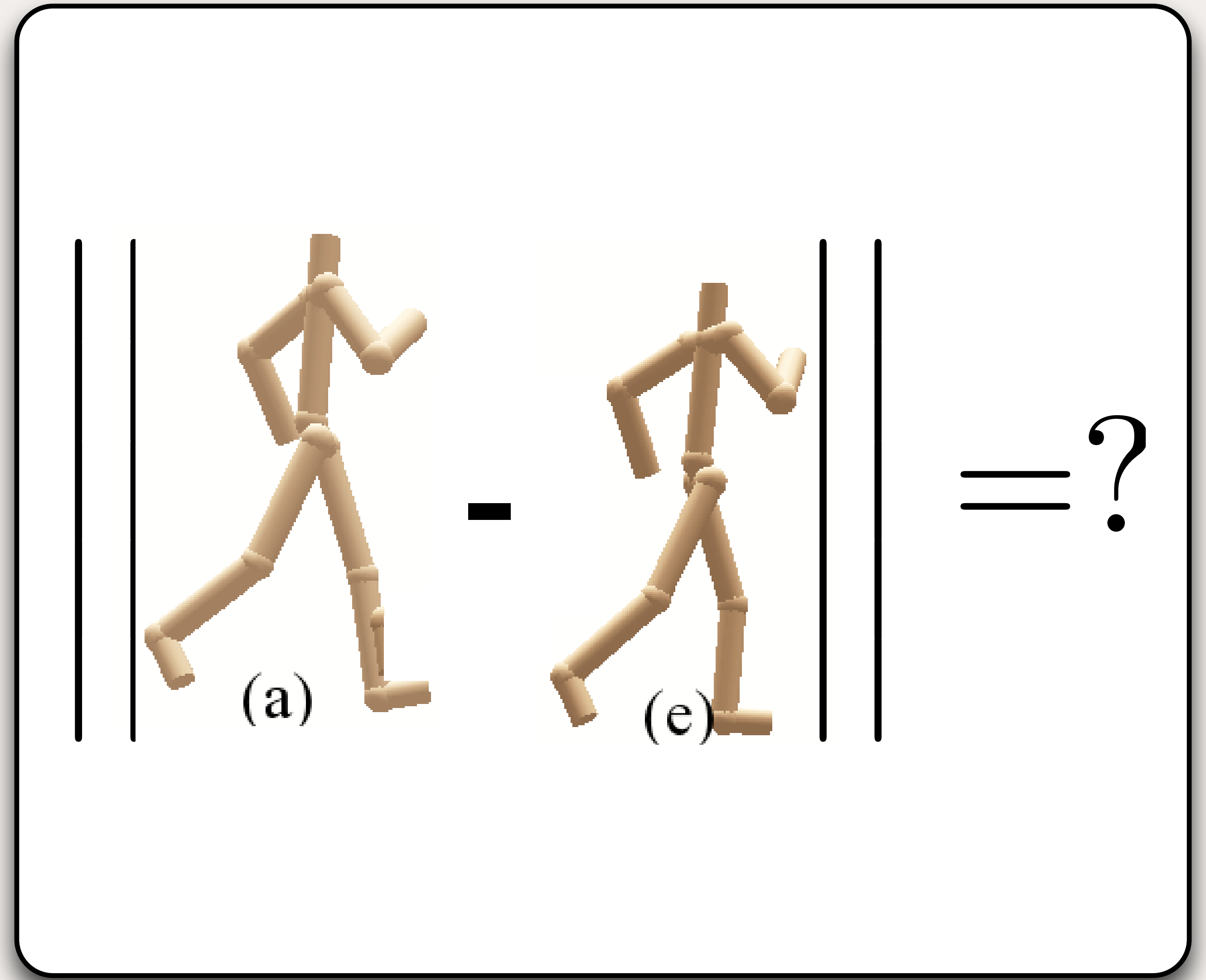


source: Kovar et al. [2002]

How?

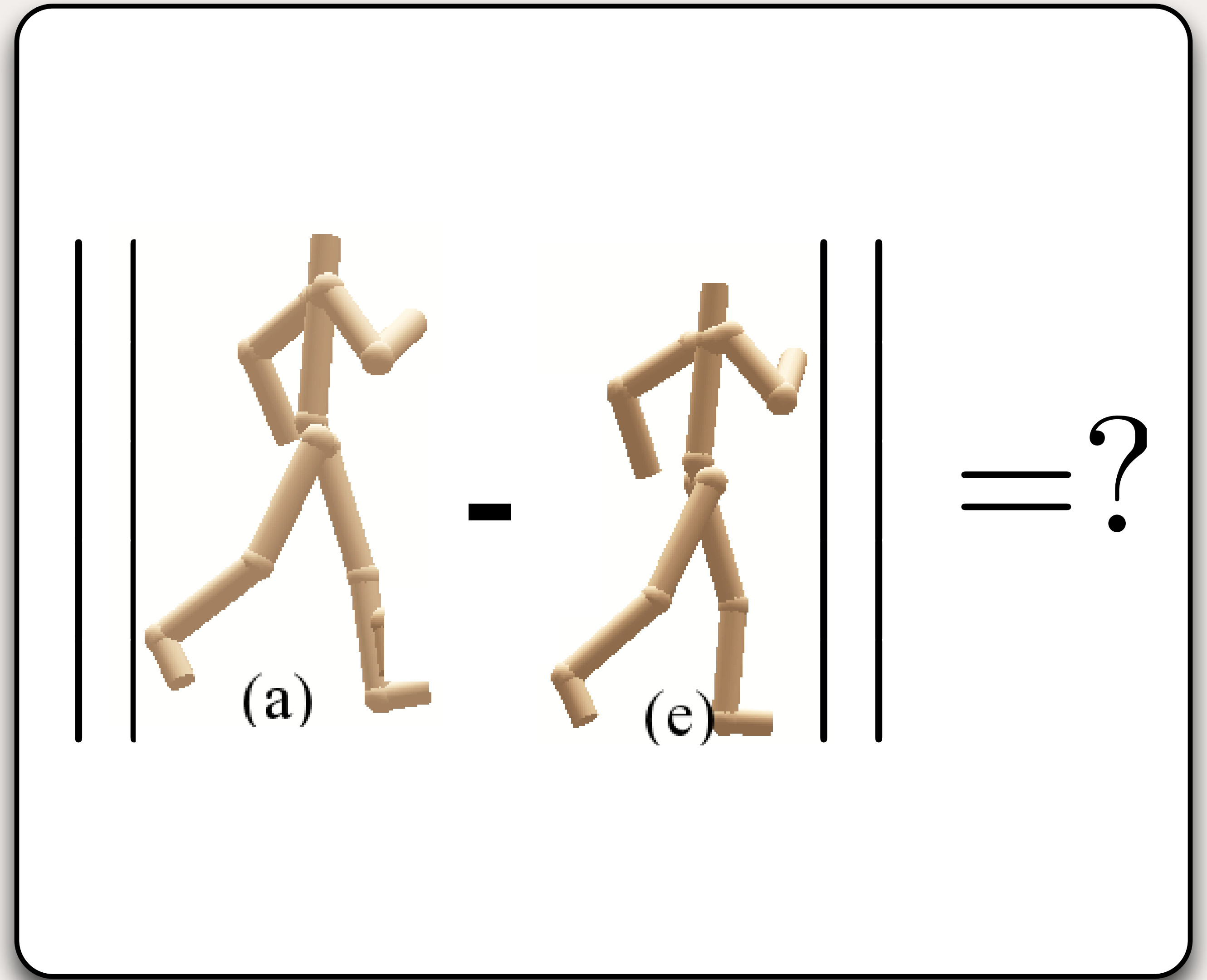
Pose Metrics

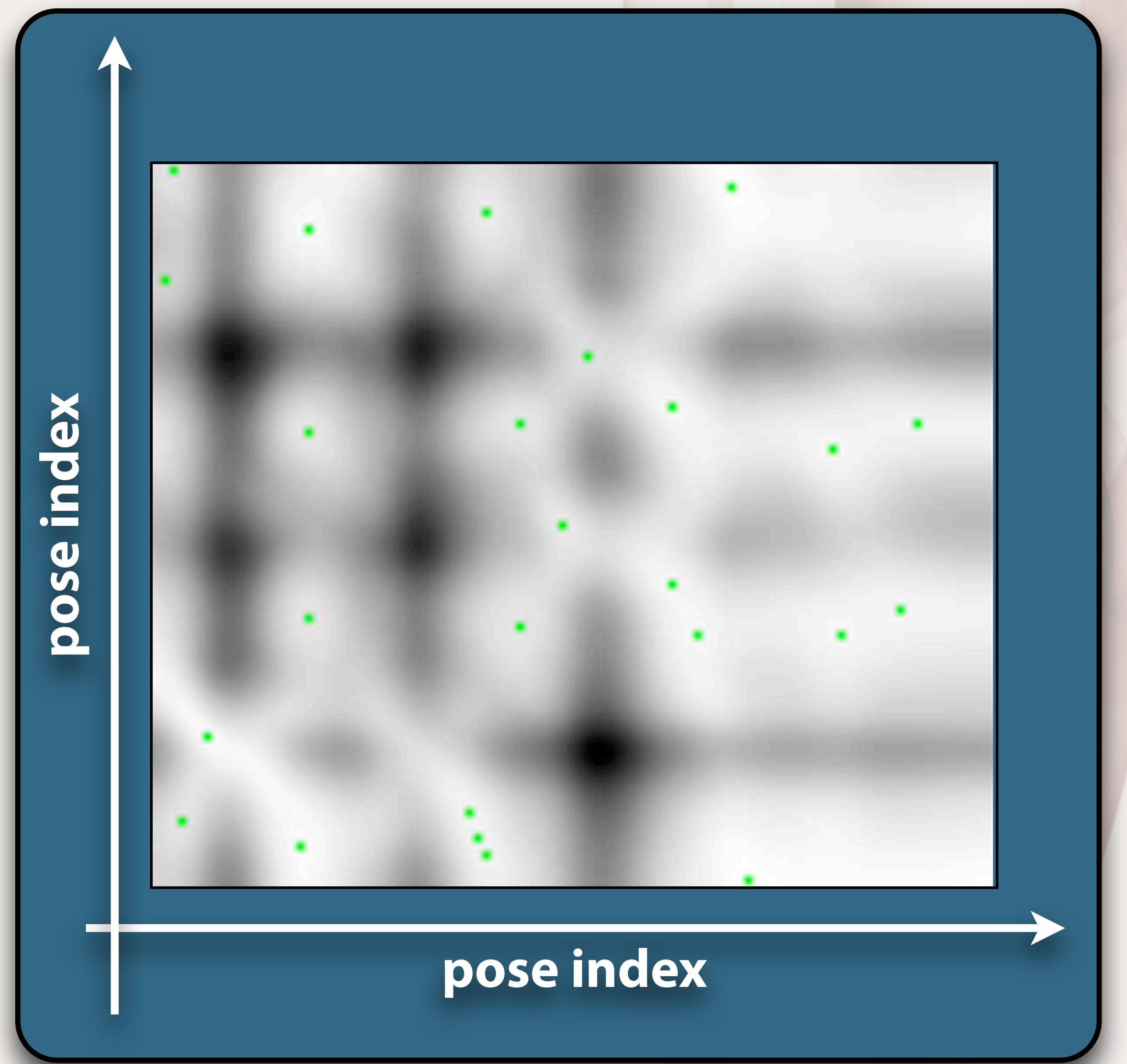
Pose Metrics



How can we define a metric on poses?

Pose Metrics

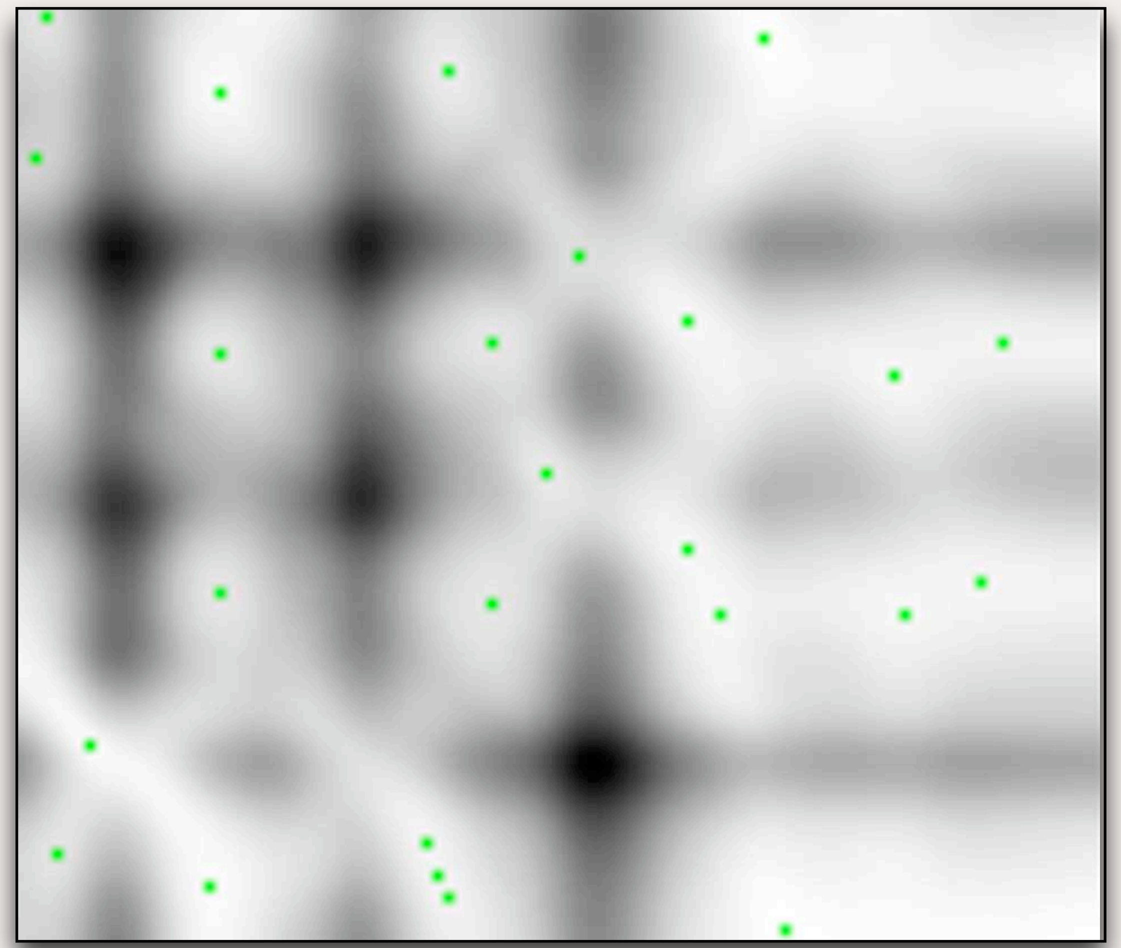




How can we define a metric on poses?

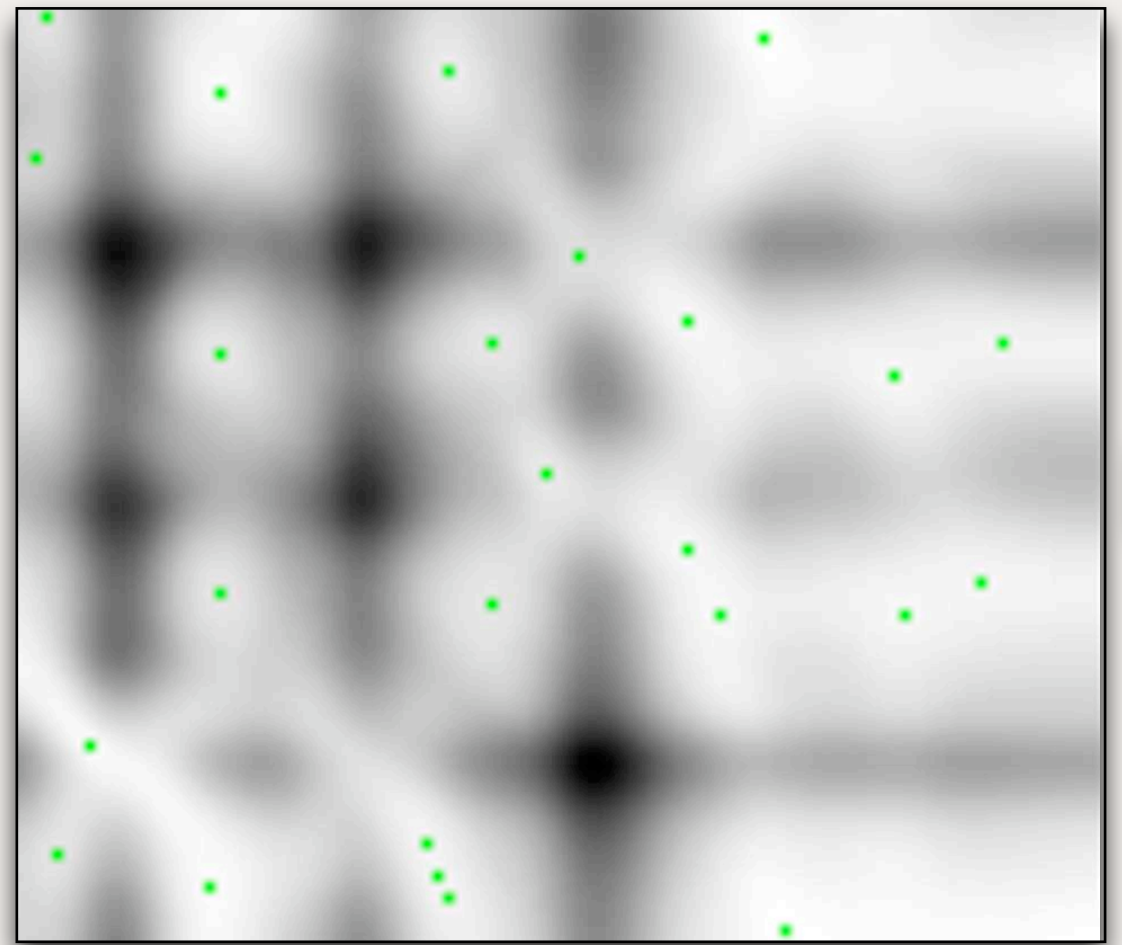
Pairwise pose differences.

Pose Metrics

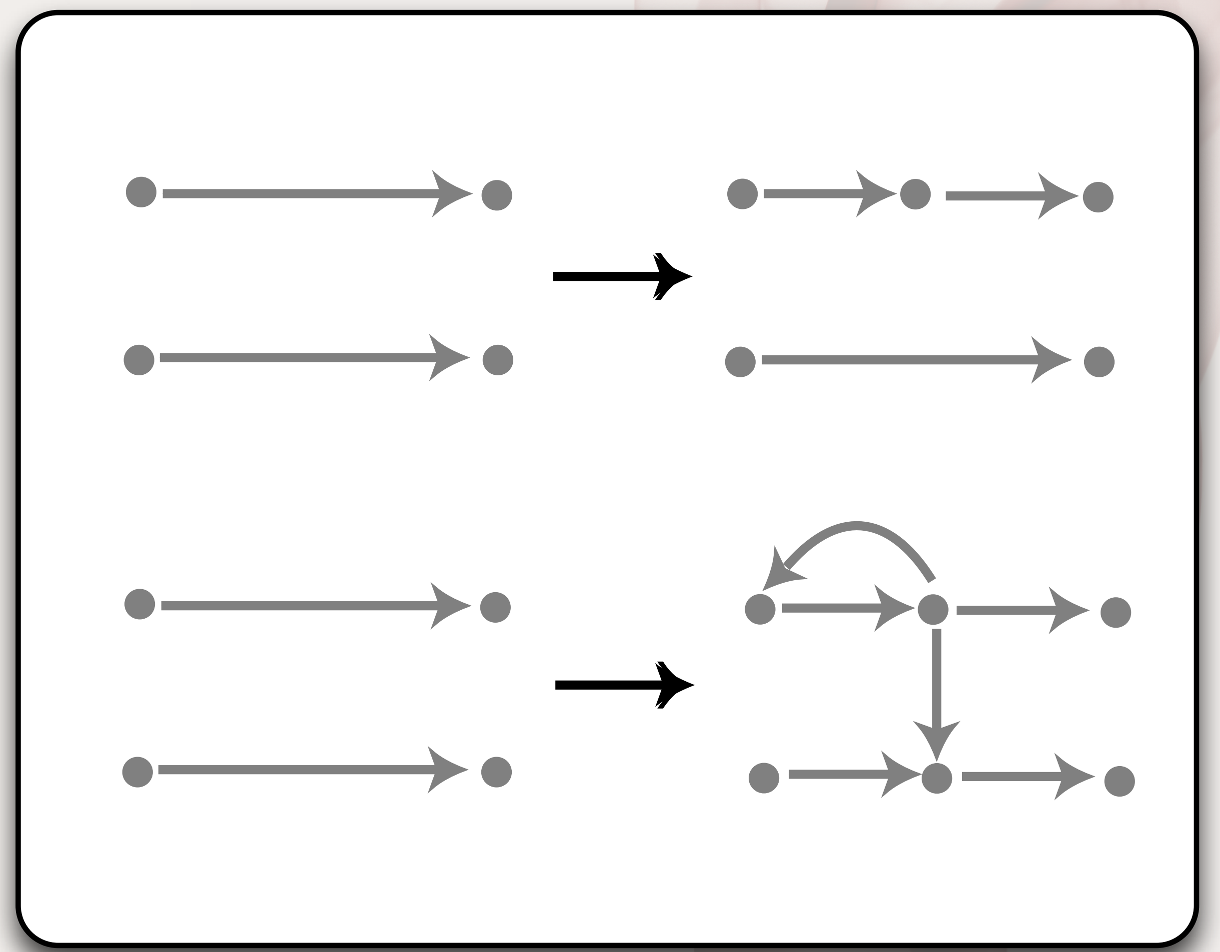


Pairwise Pose Differences

Pose Metrics



Pairwise Pose Differences

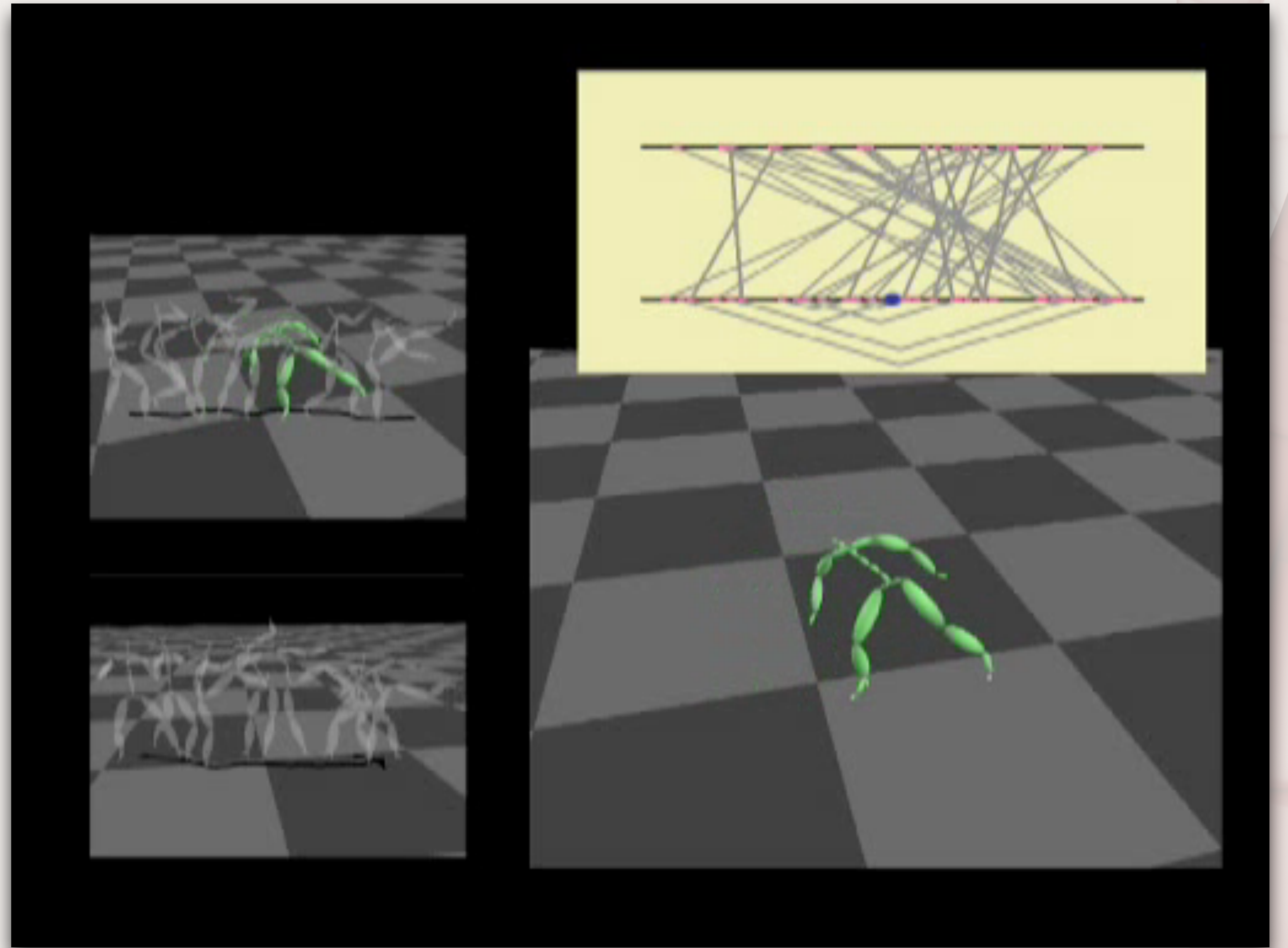


Motion Graph Schematic

Results

source: Kovar et al. [2002]

Results



source: Kovar et al. [2002]

Constraints

- Pose blending may violate physical constriants
  - Linear Momentum Conservation
  - Angular Momentum Conservation
  - Frictional Constraints (“Foot Skate”)

“Foot Skate” Poblem

http://www.youtube.com/watch?v=E_FzgtLVzbI

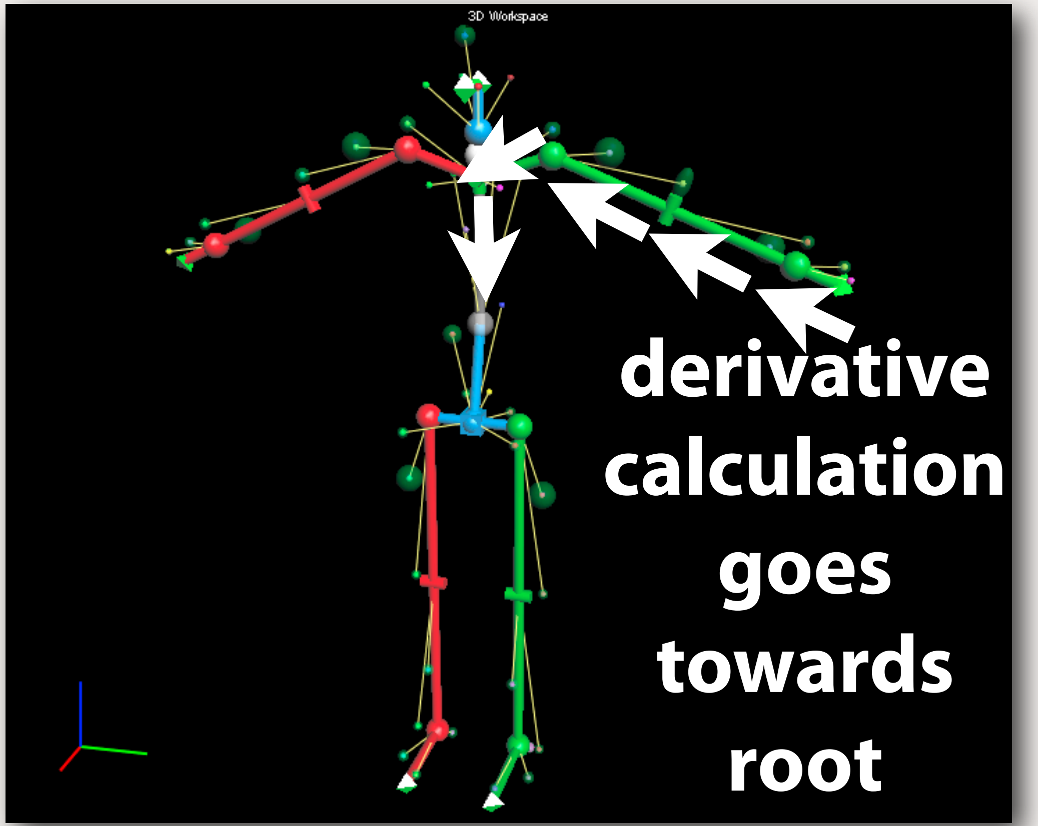
source: http://www.cs.wisc.edu/graphics/Gallery/kovar.vol/Cleanup/

“Foot Skate” Poblem

http://www.youtube.com/watch?v=E_FzgtLVzbI

source: http://www.cs.wisc.edu/graphics/Gallery/kovar.vol/Cleanup/

Inverse Kinematic Solution



$$\omega_i = f_{i,\Omega}(\omega_{i-1})$$

$$\frac{dE}{d\Omega} = 2\sum_j (\hat{\mathbf{m}}_j^\star - \hat{\mathbf{m}}_j)^T \frac{d\hat{\mathbf{m}}_j}{d\Omega}$$

$$\frac{d\hat{\mathbf{m}}_j}{d\Omega} = \frac{\partial \hat{\mathbf{m}}_j}{\partial \omega_i}\left(\frac{\partial \omega_i}{\partial \Omega} + \frac{\partial \omega_i}{\partial \omega_{i-1}}\frac{\partial \omega_{i-1}}{\partial \Omega} + \frac{\partial \omega_i}{\partial \omega_{i-1}}\frac{\partial \omega_{i-1}}{\partial \omega_{i-2}}\frac{\partial \omega_{i-2}}{\partial \Omega} + \cdots\right)$$

IK Results

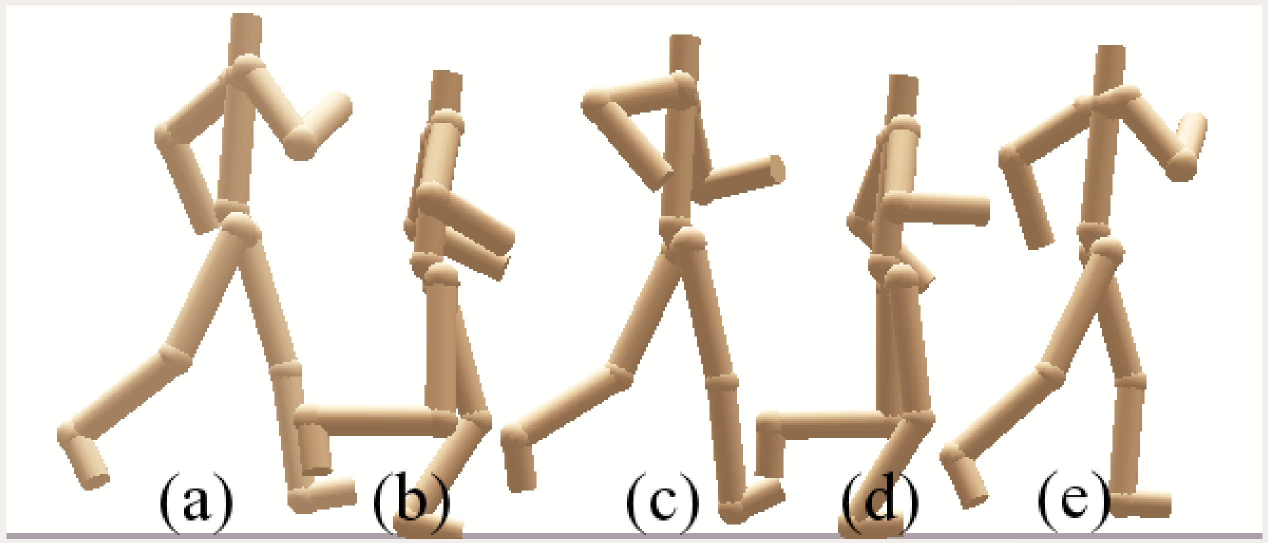
source: http://www.cs.wisc.edu/graphics/Gallery/kovar.vol/Cleanup/

IK Results

source: http://www.cs.wisc.edu/graphics/Gallery/kovar.vol/Cleanup/

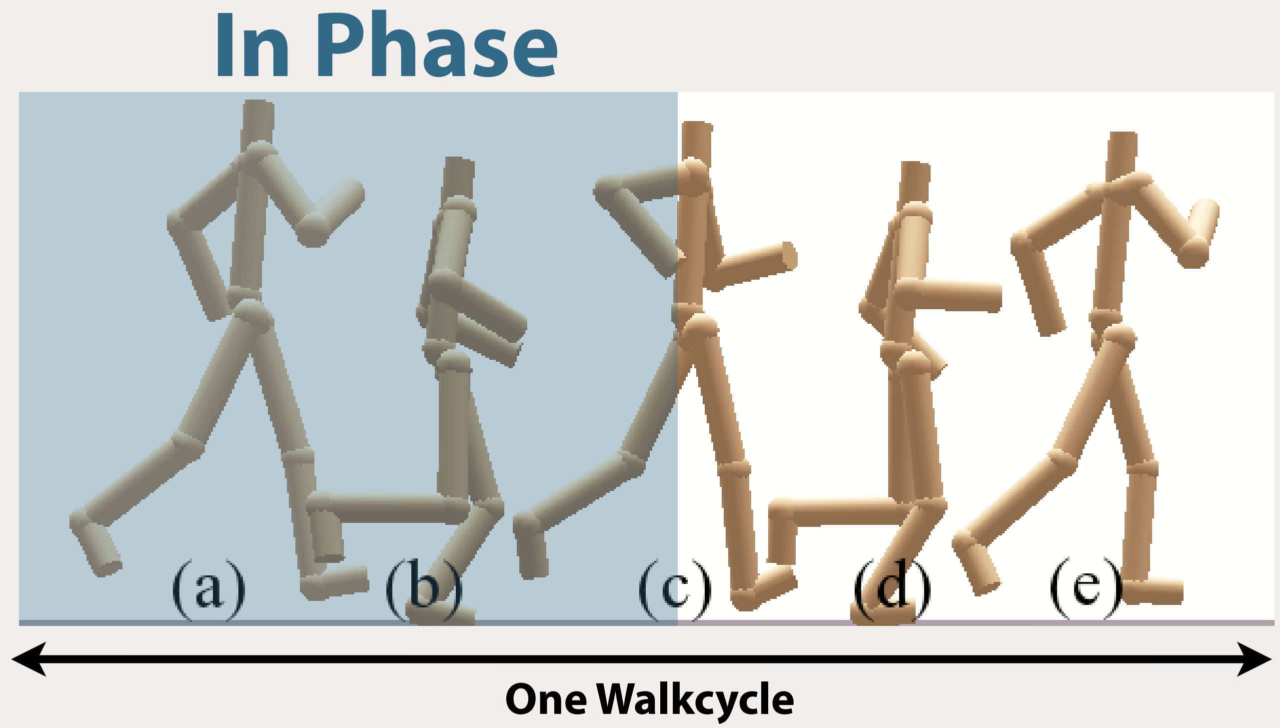
Smart Blending

Smart Blending

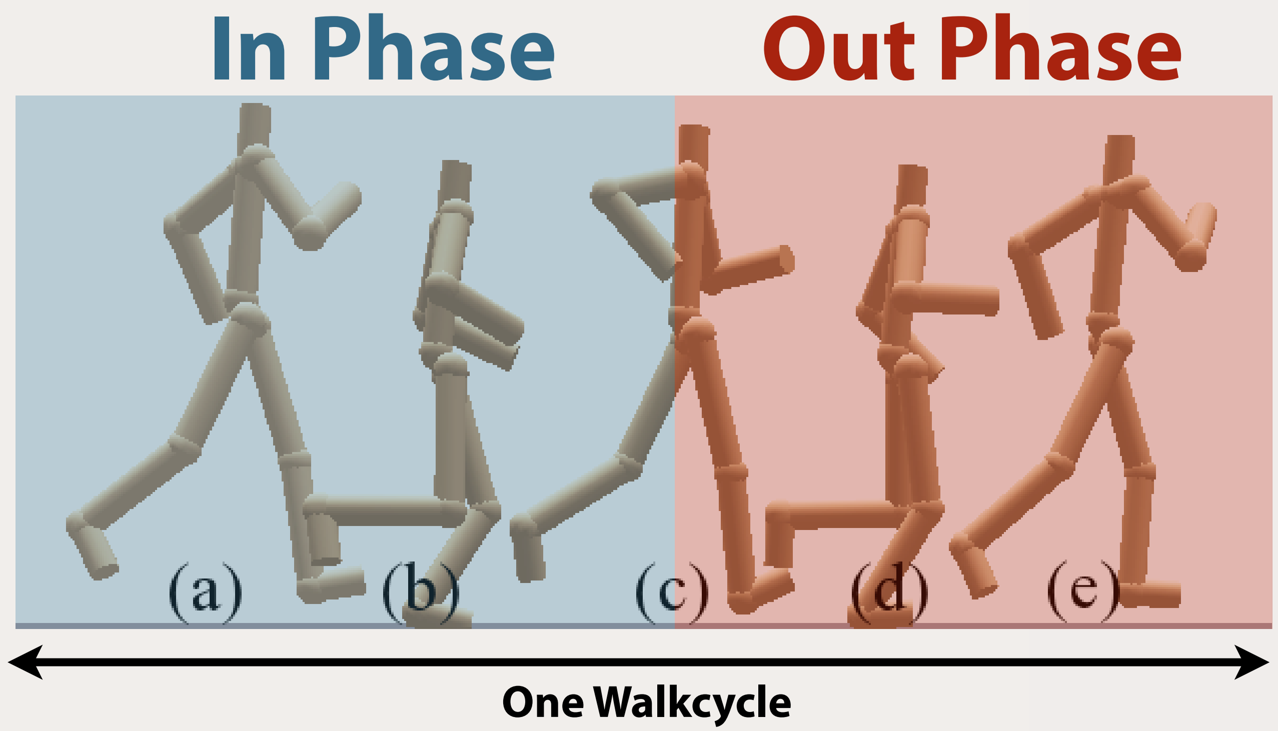


One Walkcycle

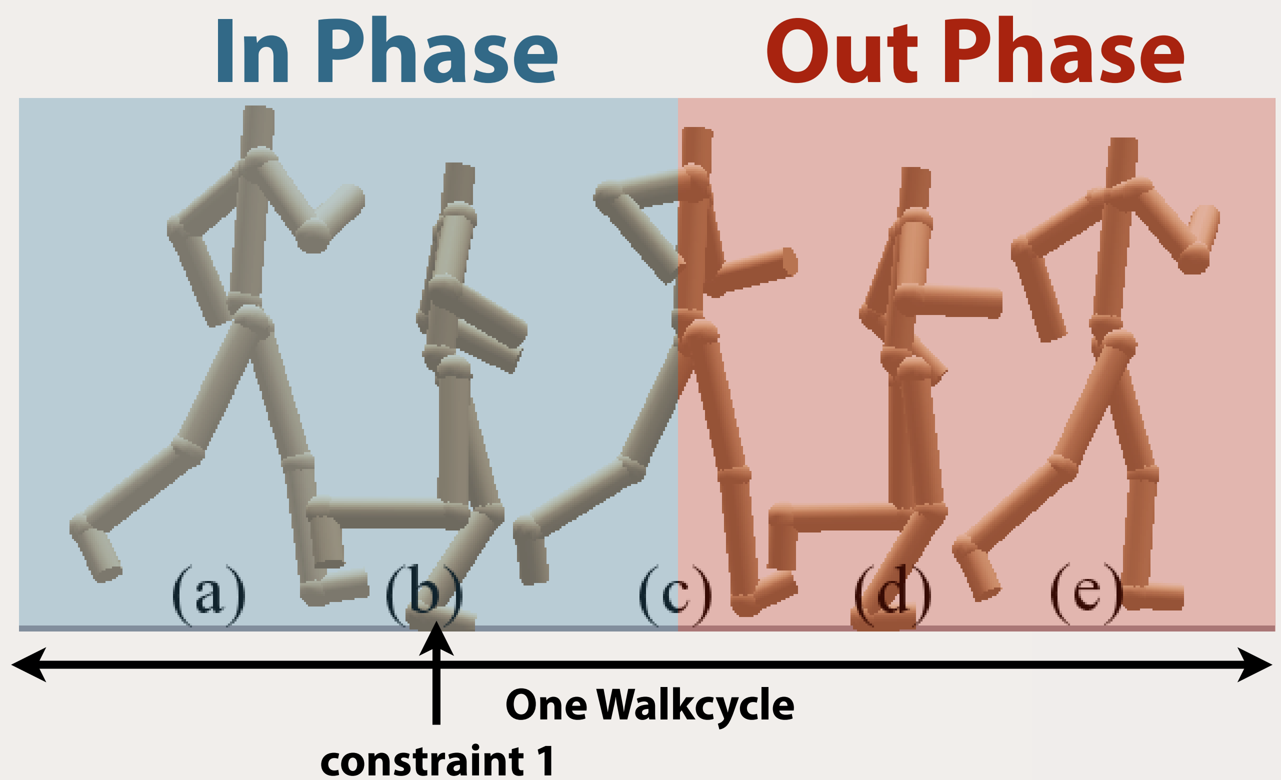
Smart Blending



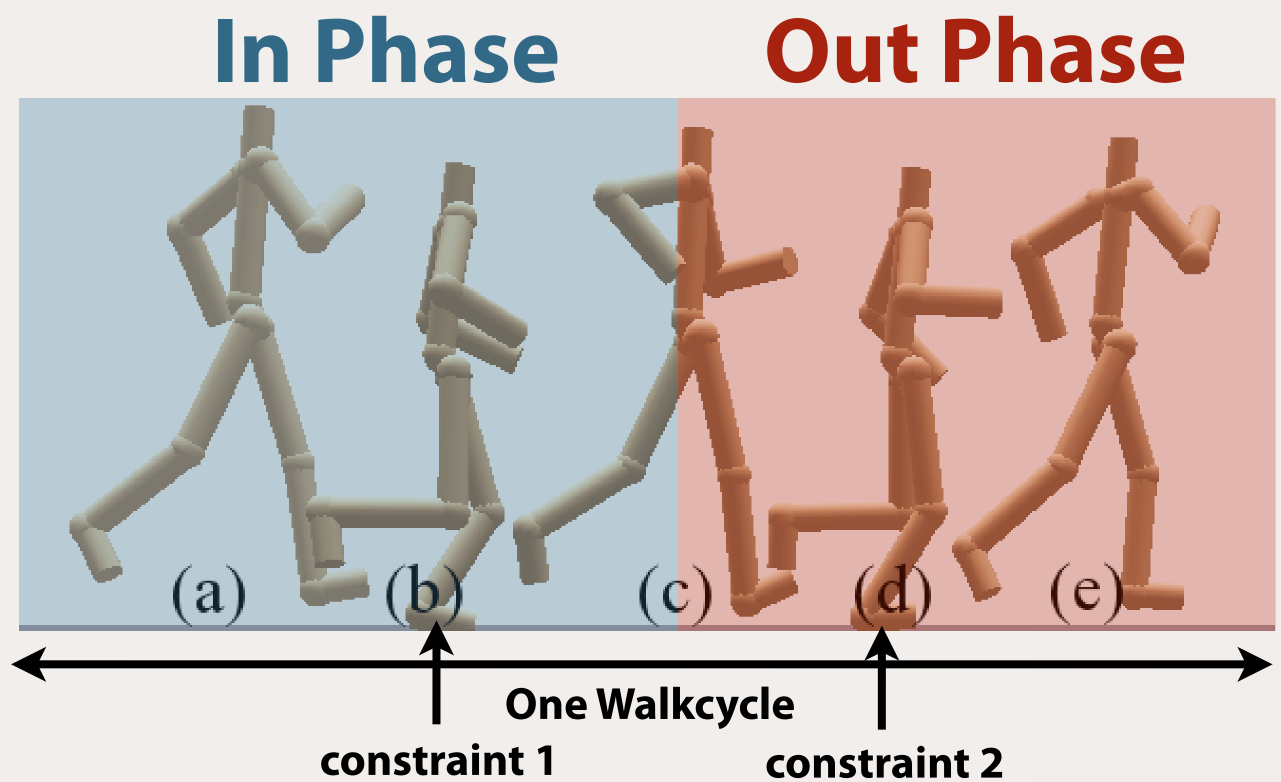
Smart Blending



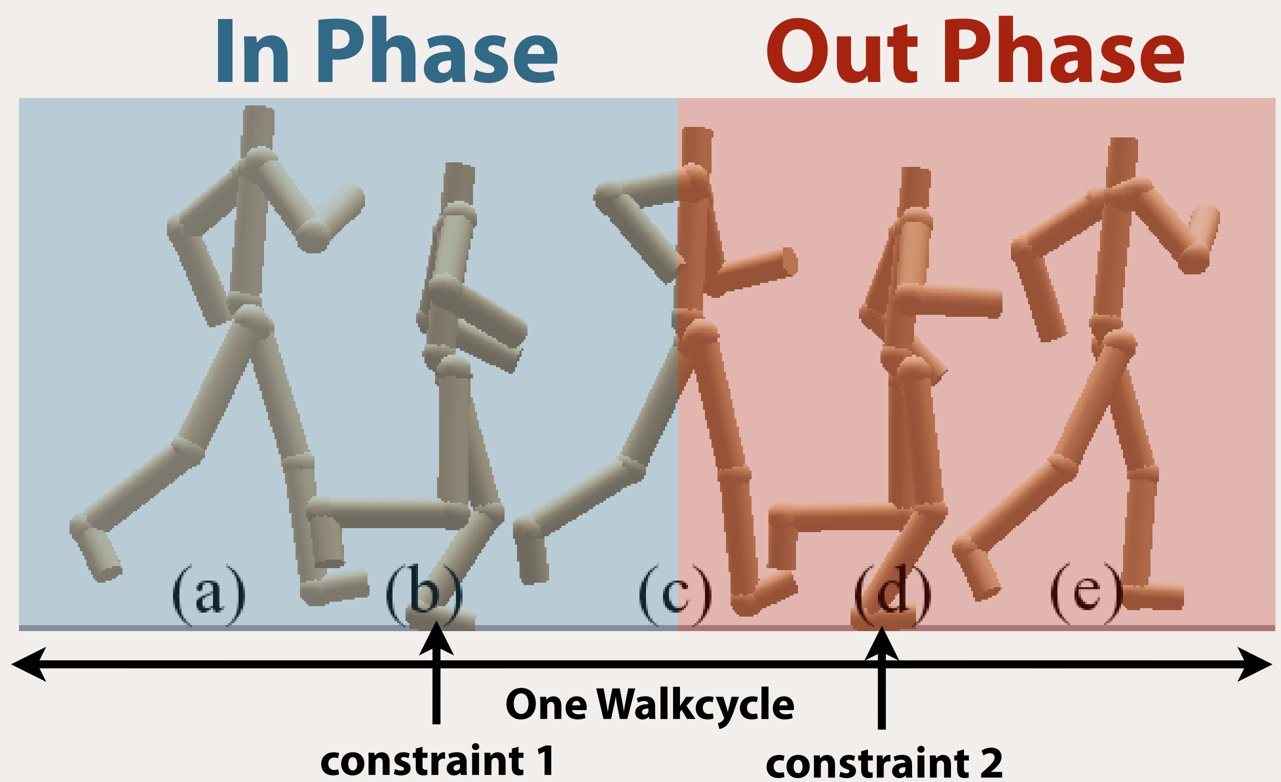
Smart Blending

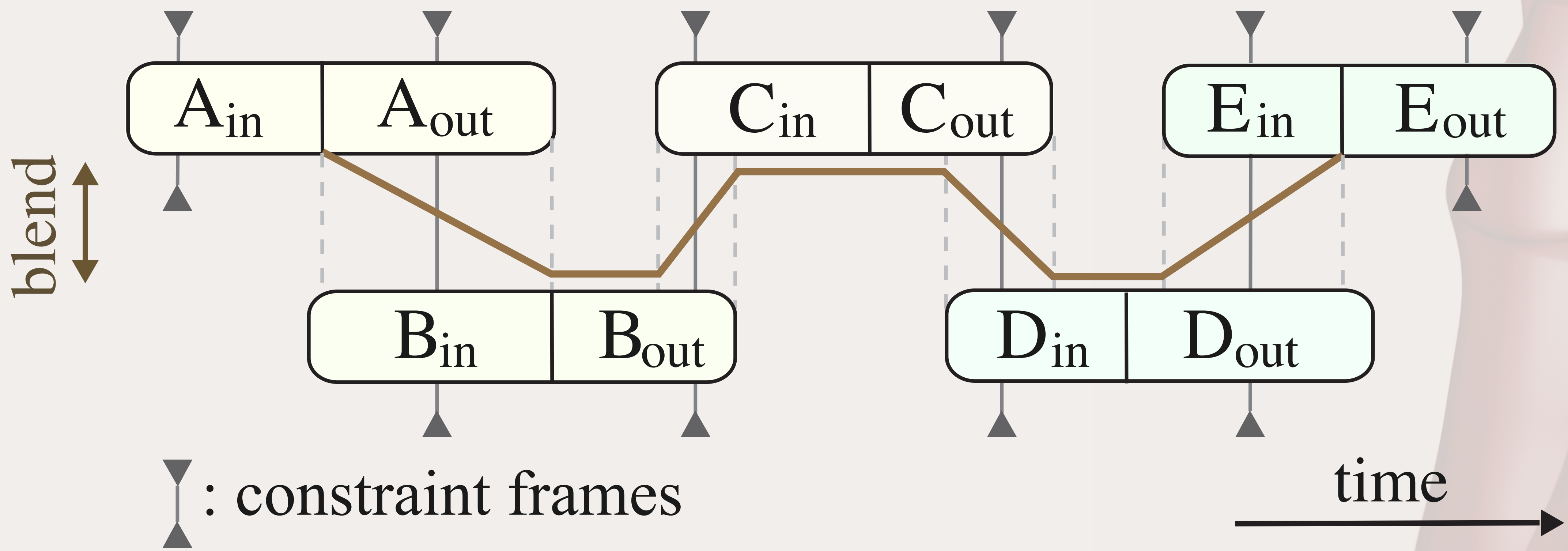


Smart Blending



Smart Blending

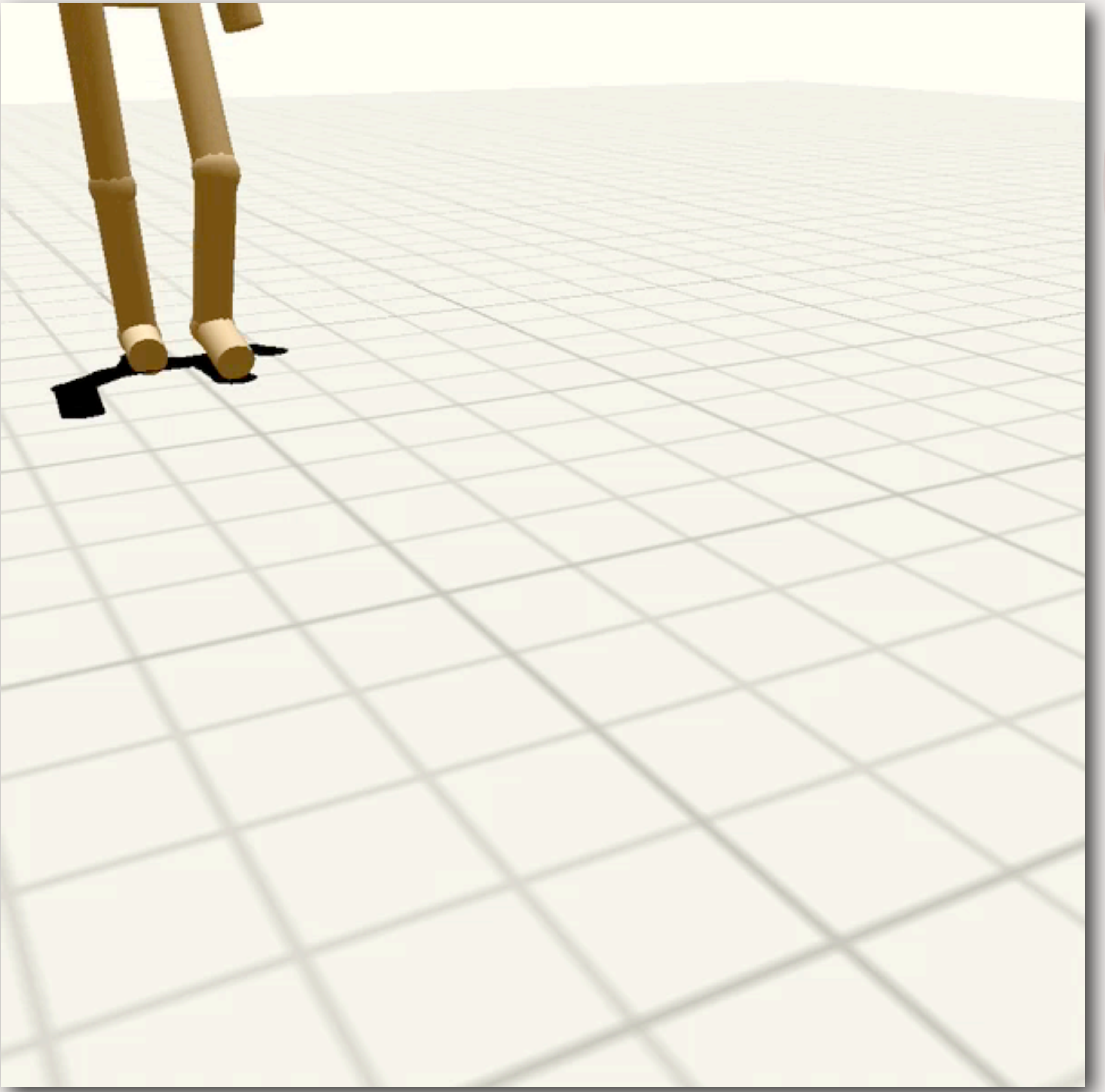




Smart Blending Example

source: Treuille et al. [2002]

Smart Blending Example



source: Treuille et al. [2002]

Open Problems

- How to pick to which clip to transition?
- How to enforce temporal contraints?
- How to generalize beyond the given clips?

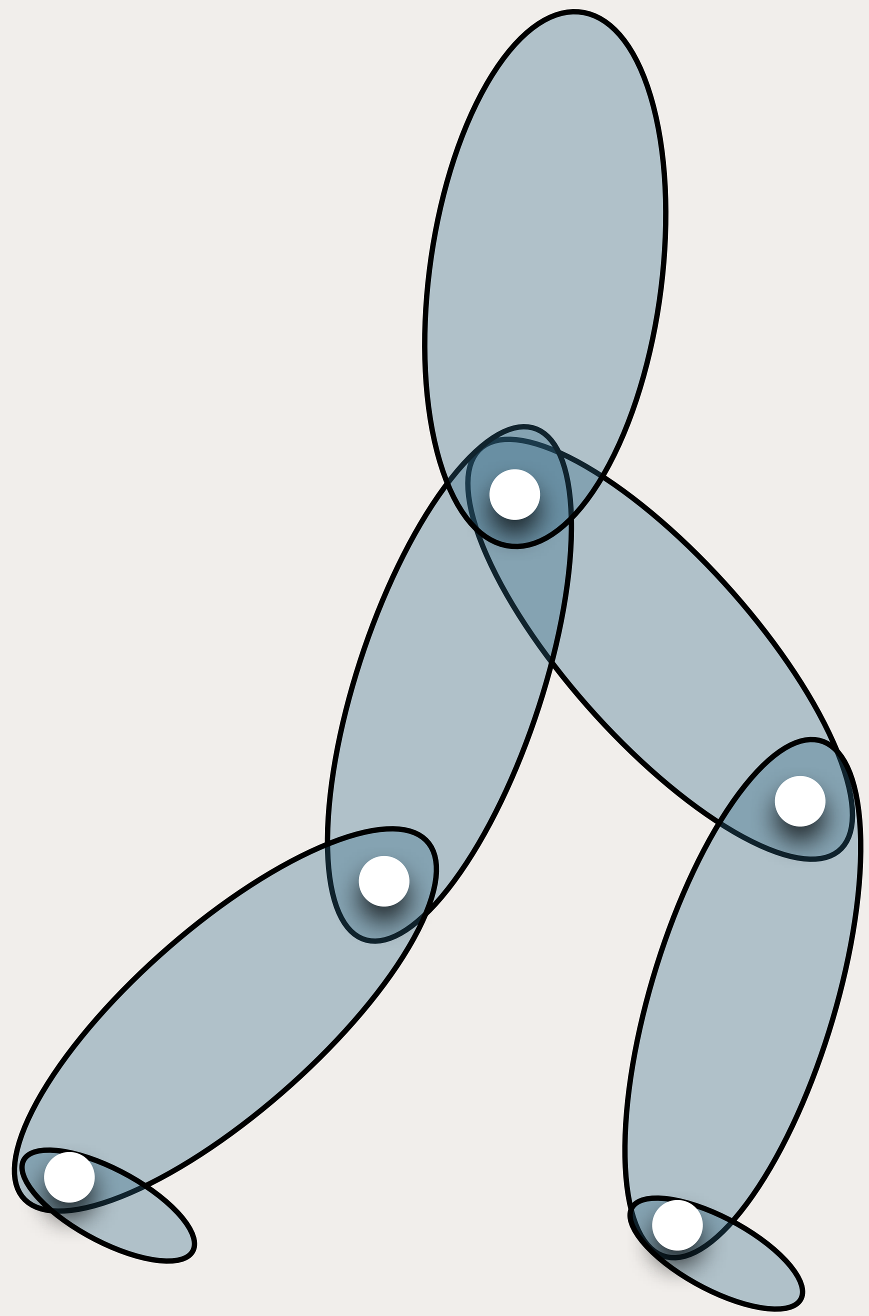
Animation Topics

- **Data-Driven Motion**
- Physics Based Motion
- Motion of other Animals

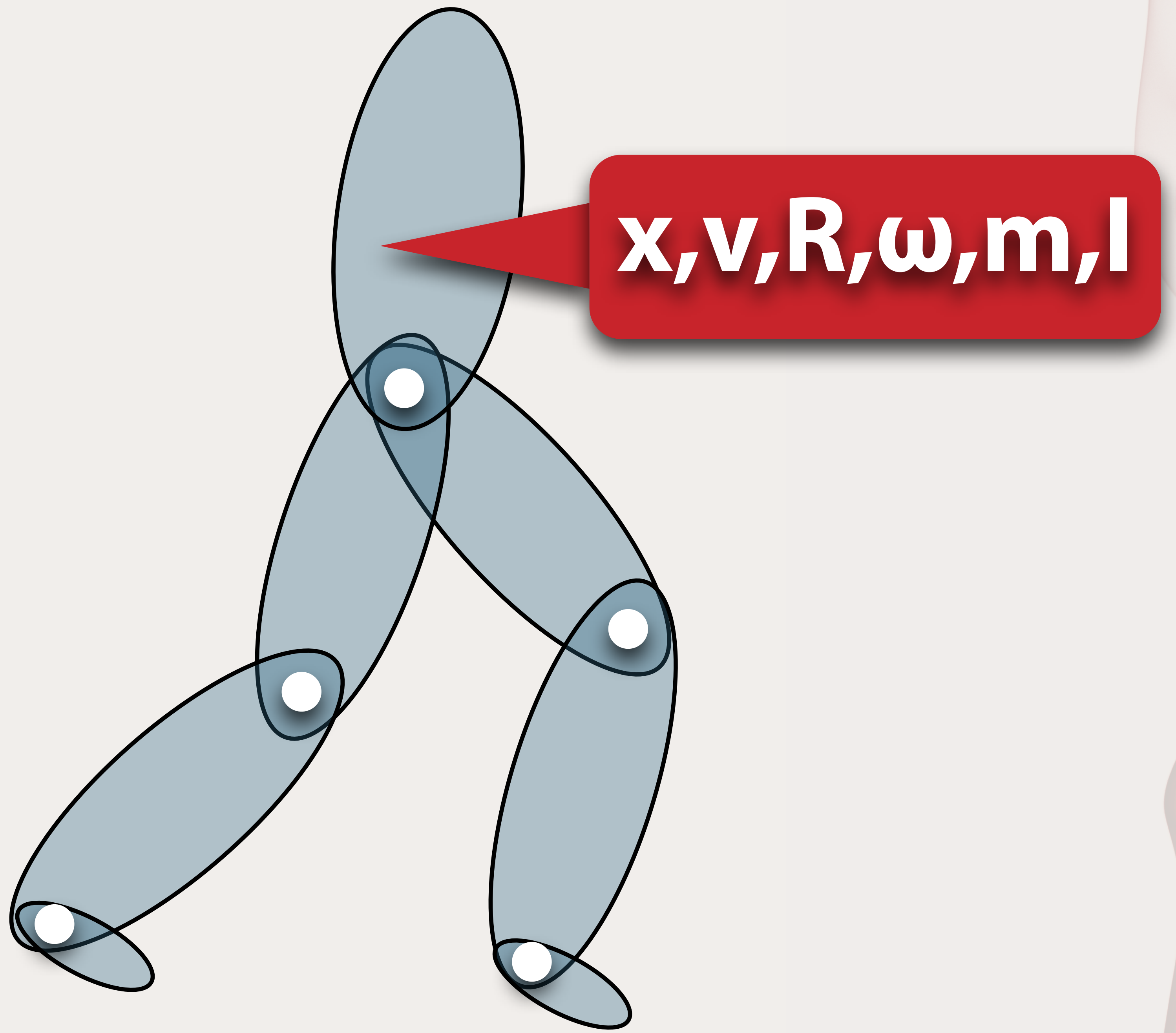
Animation Topics

- Data-Driven Motion
- **Physics Based Motion**
- Motion of other Animals

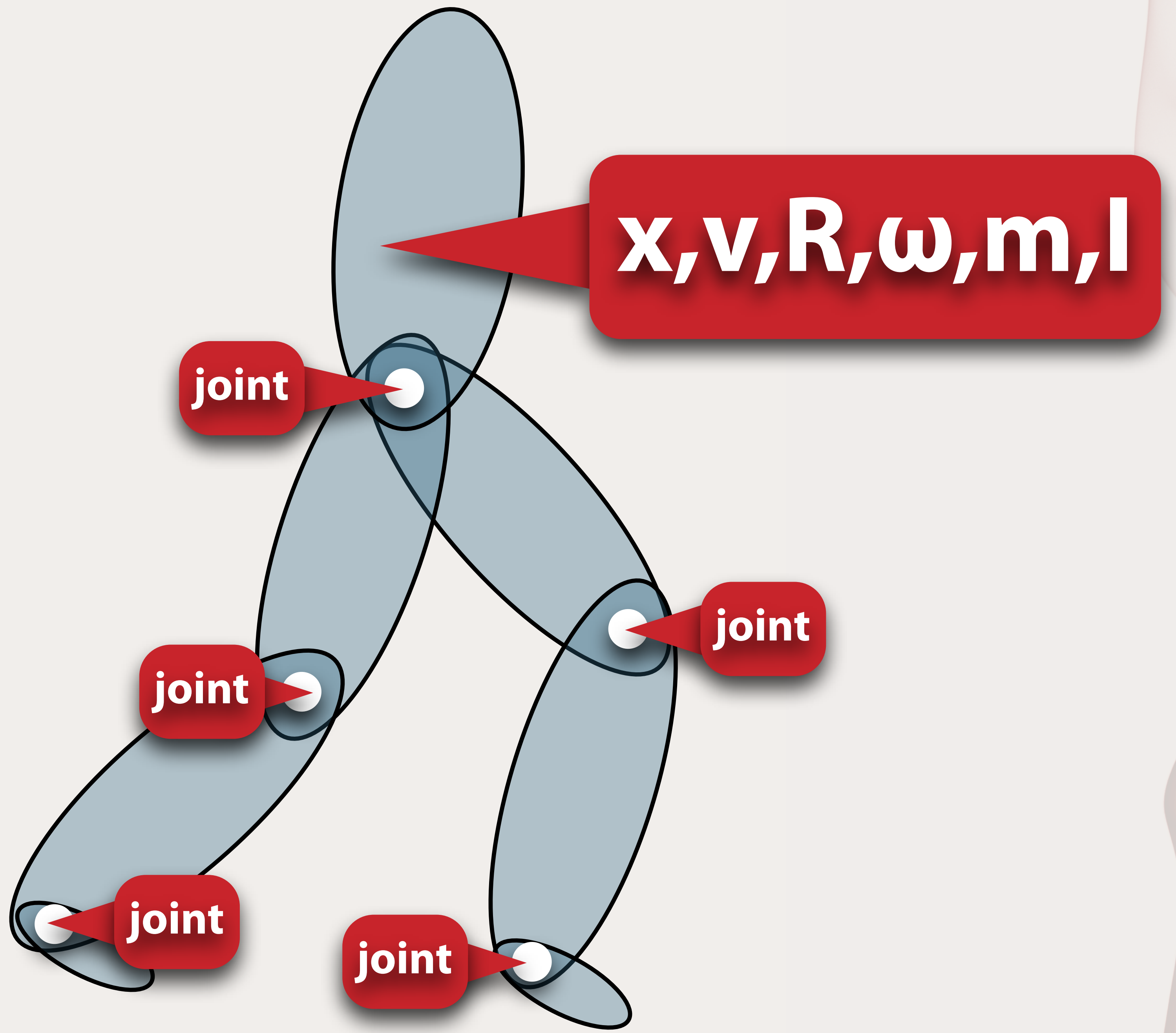
Physical Model



Physical Model



Physical Model

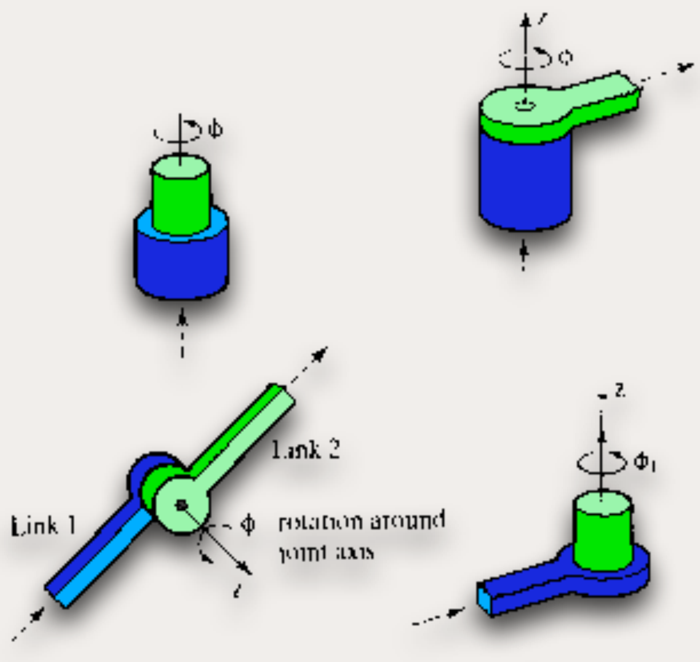


Joint Types

All joints can be written as the composition of...

Joint Types

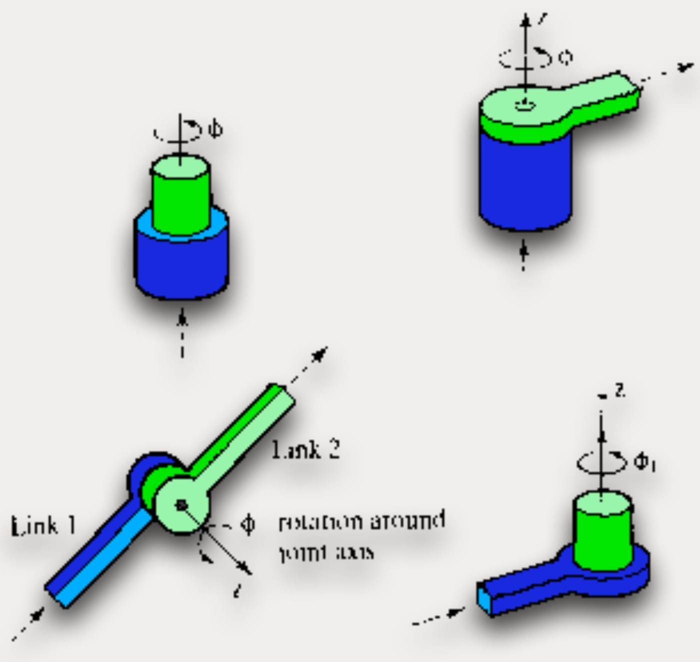
All joints can be written as the composition of...

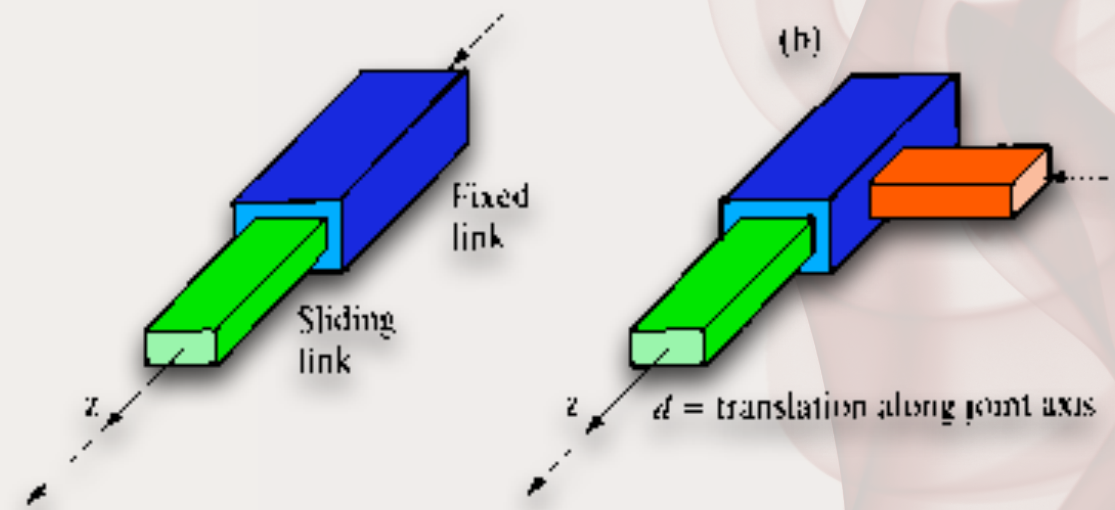


Rotary

Joint Types

All joints can be written as the composition of...

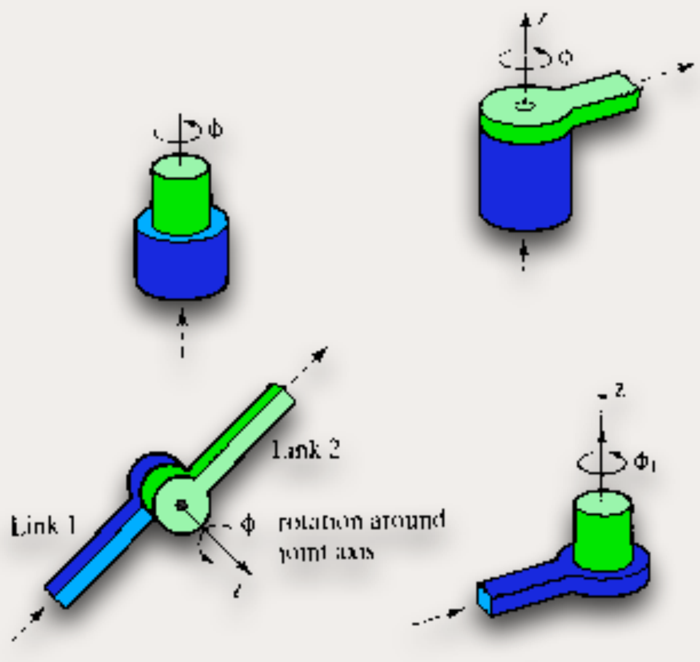




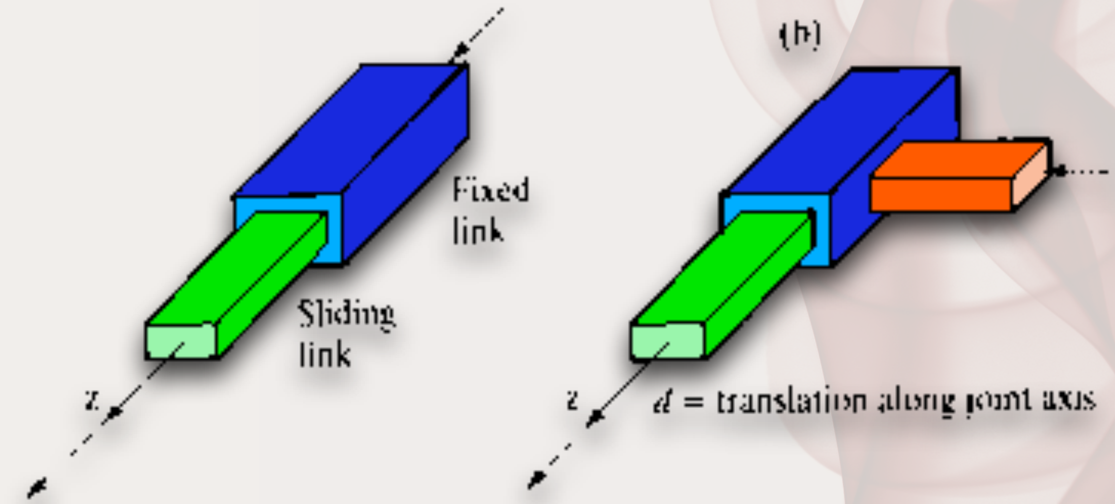
Rotary **Prismatic**

Joint Types

All joints can be written as the composition of...



Rotary

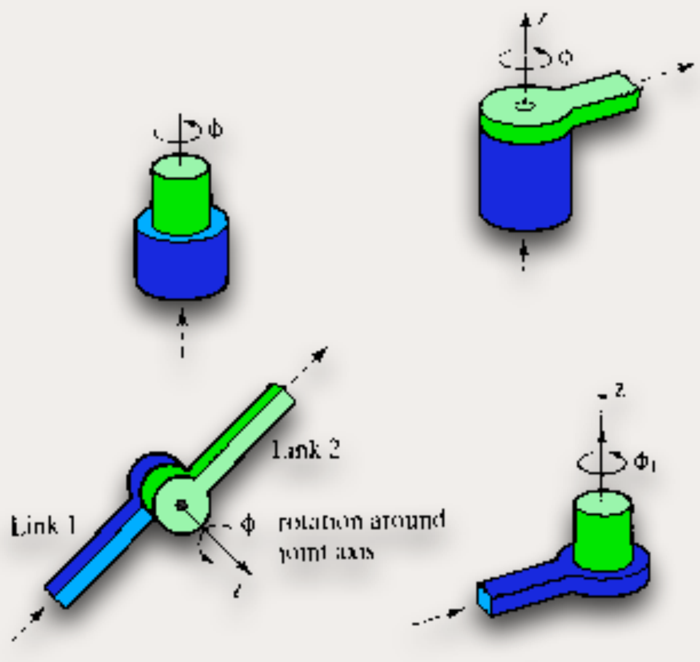


Prismatic

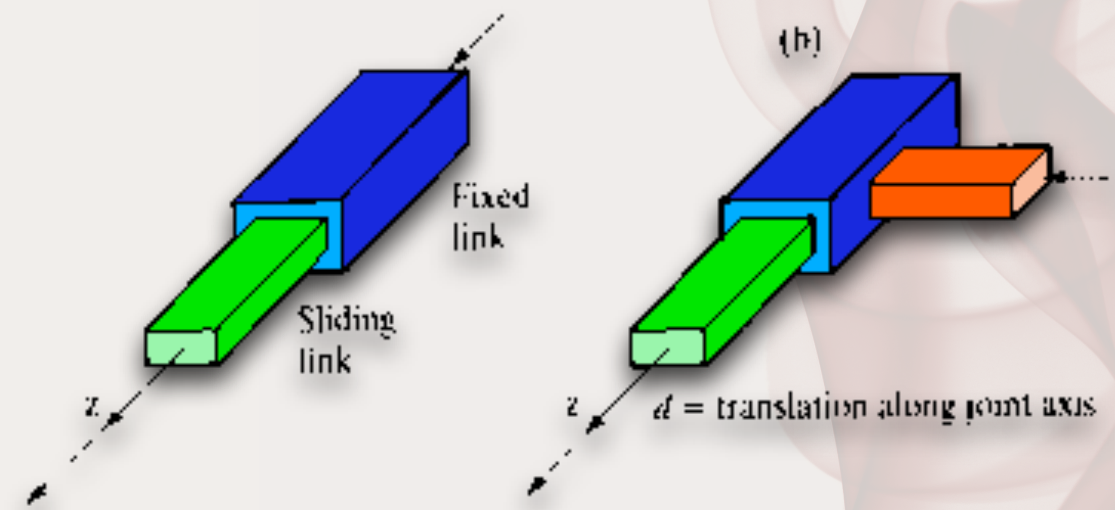
...and have two forms:

Joint Types

All joints can be written as the composition of...



Rotary



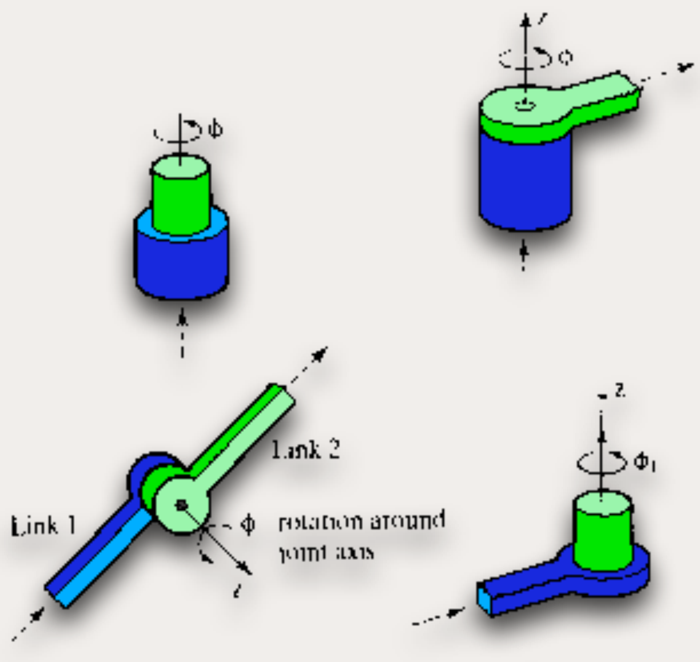
Prismatic

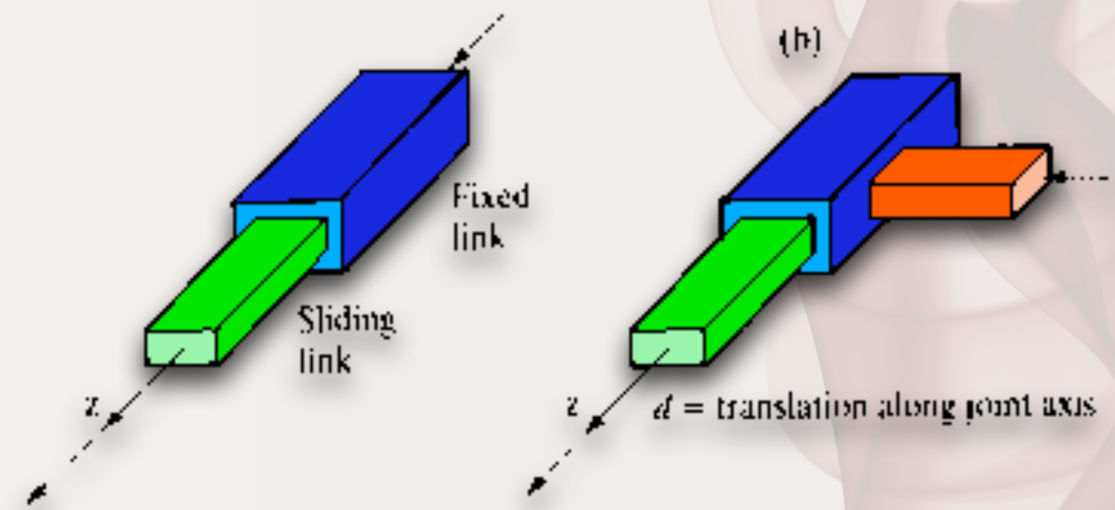
...and have two forms:

1. Constraint Form

Joint Types

All joints can be written as the composition of...



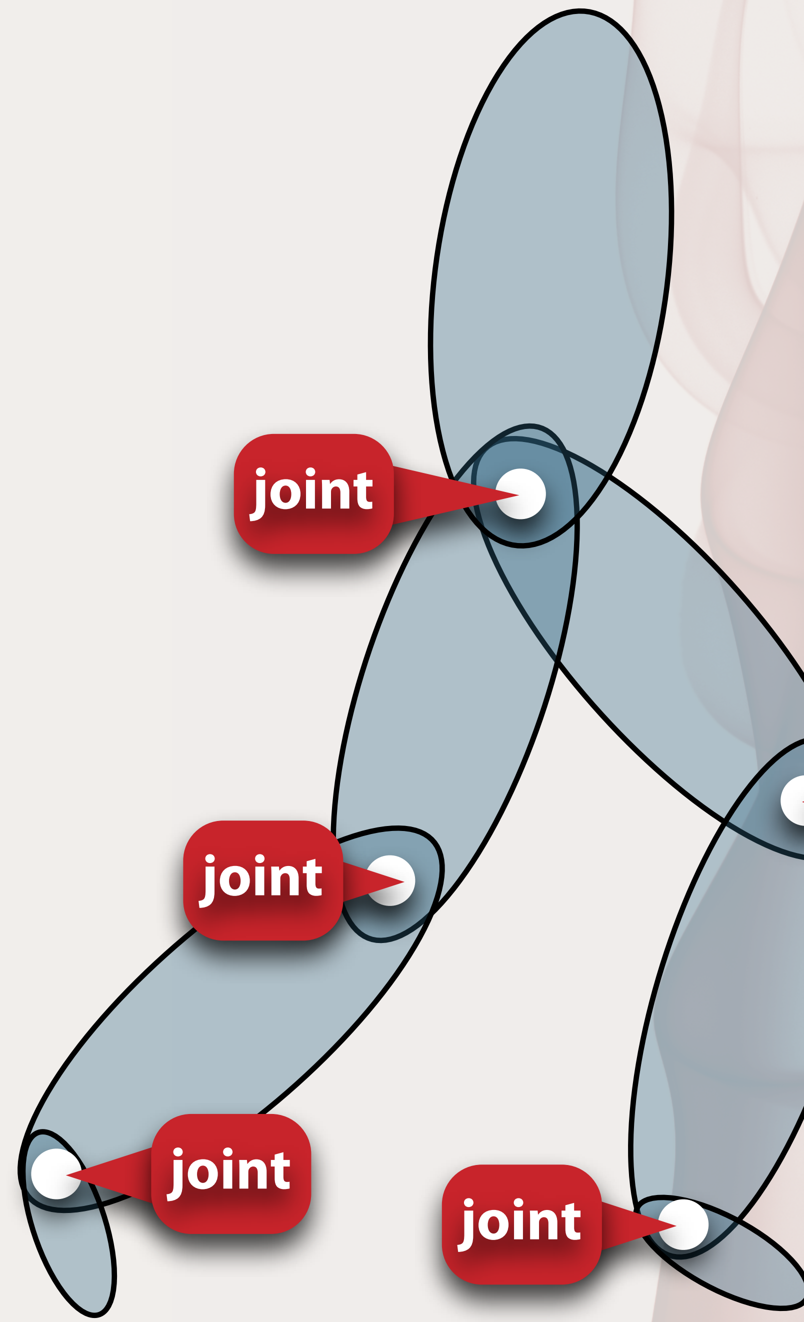


Rotary **Prismatic**

...and have two forms:

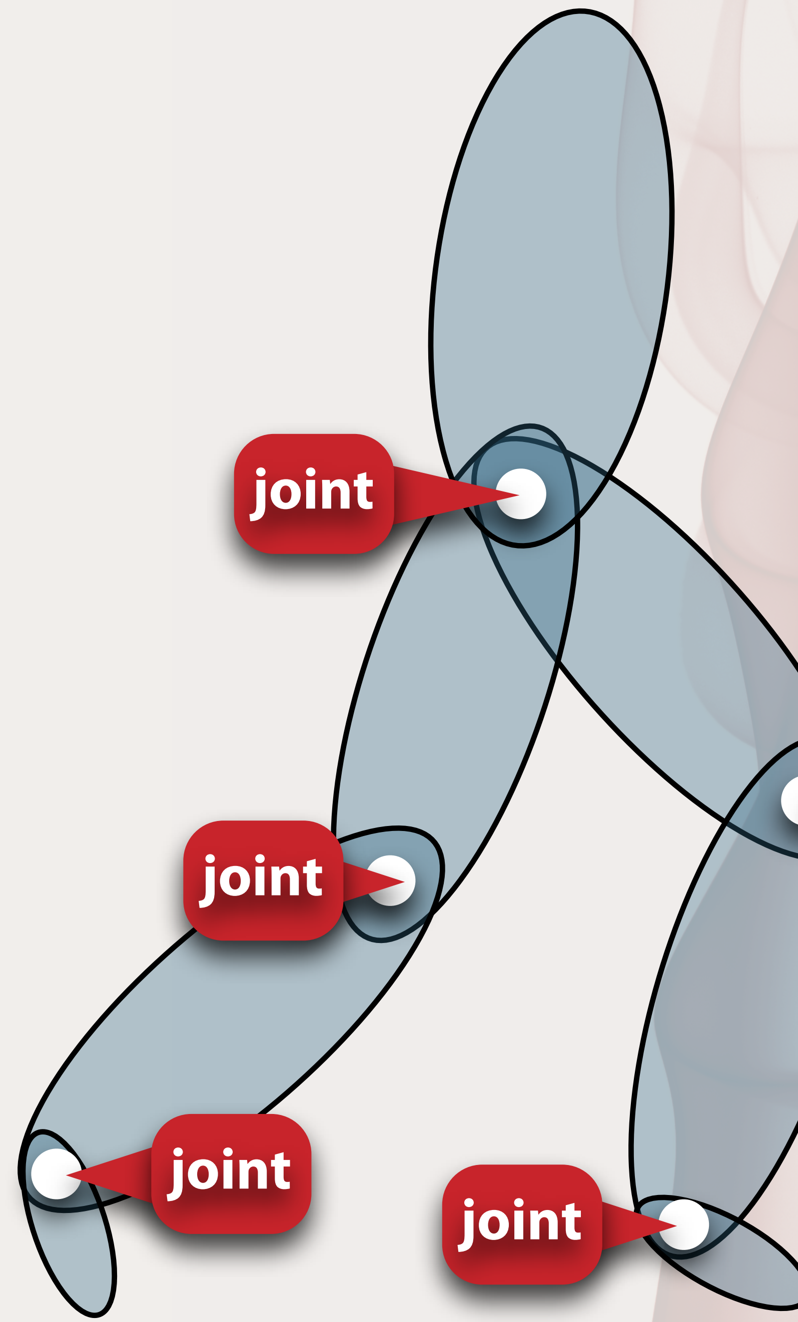
1. Constraint Form
2. Functional Form

Joint Enforcement



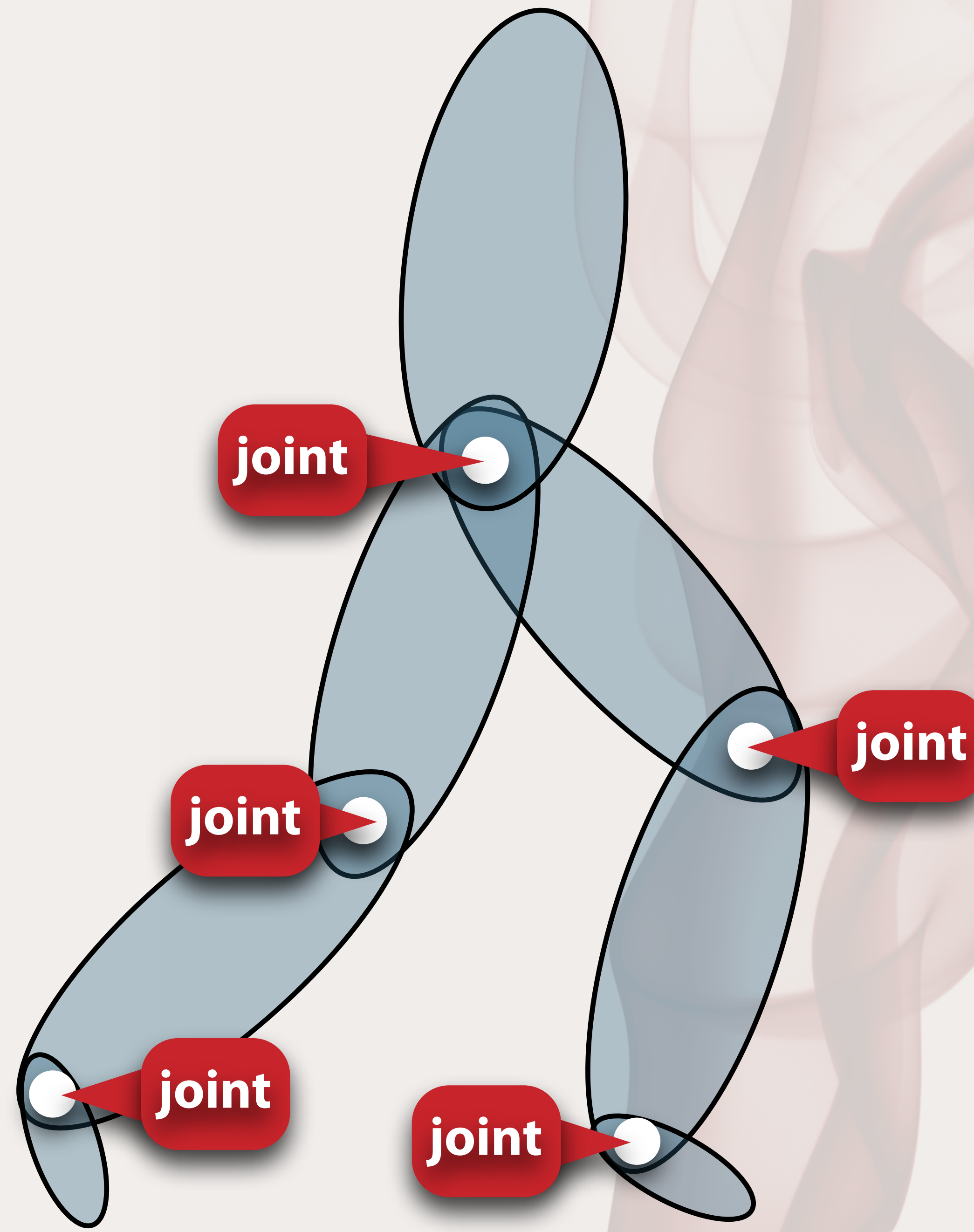
Joint Enforcement

- **Penalty Methods**



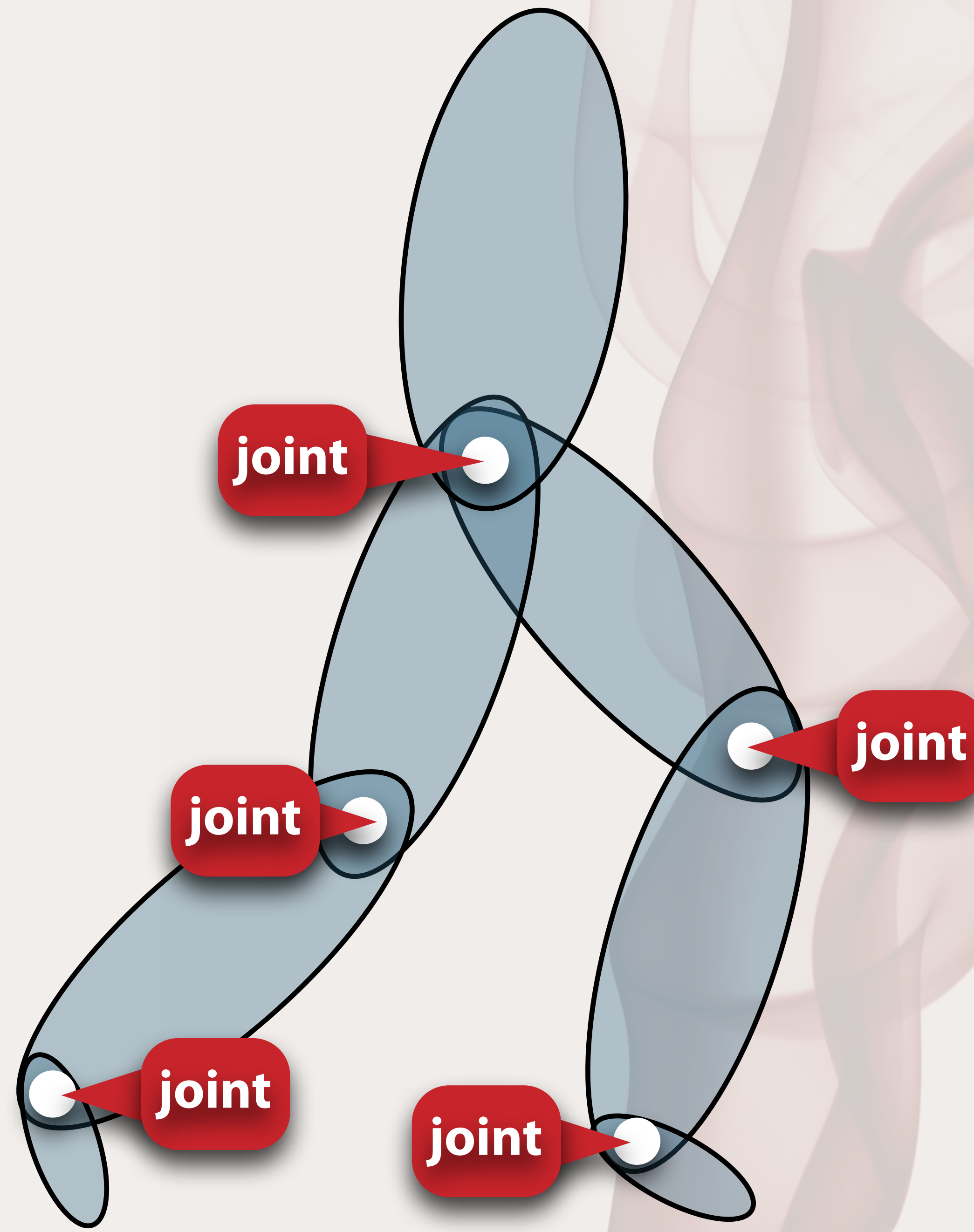
Joint Enforcement

- Penalty Methods
- Contraint Methods
  - aka Maximal Coordinate

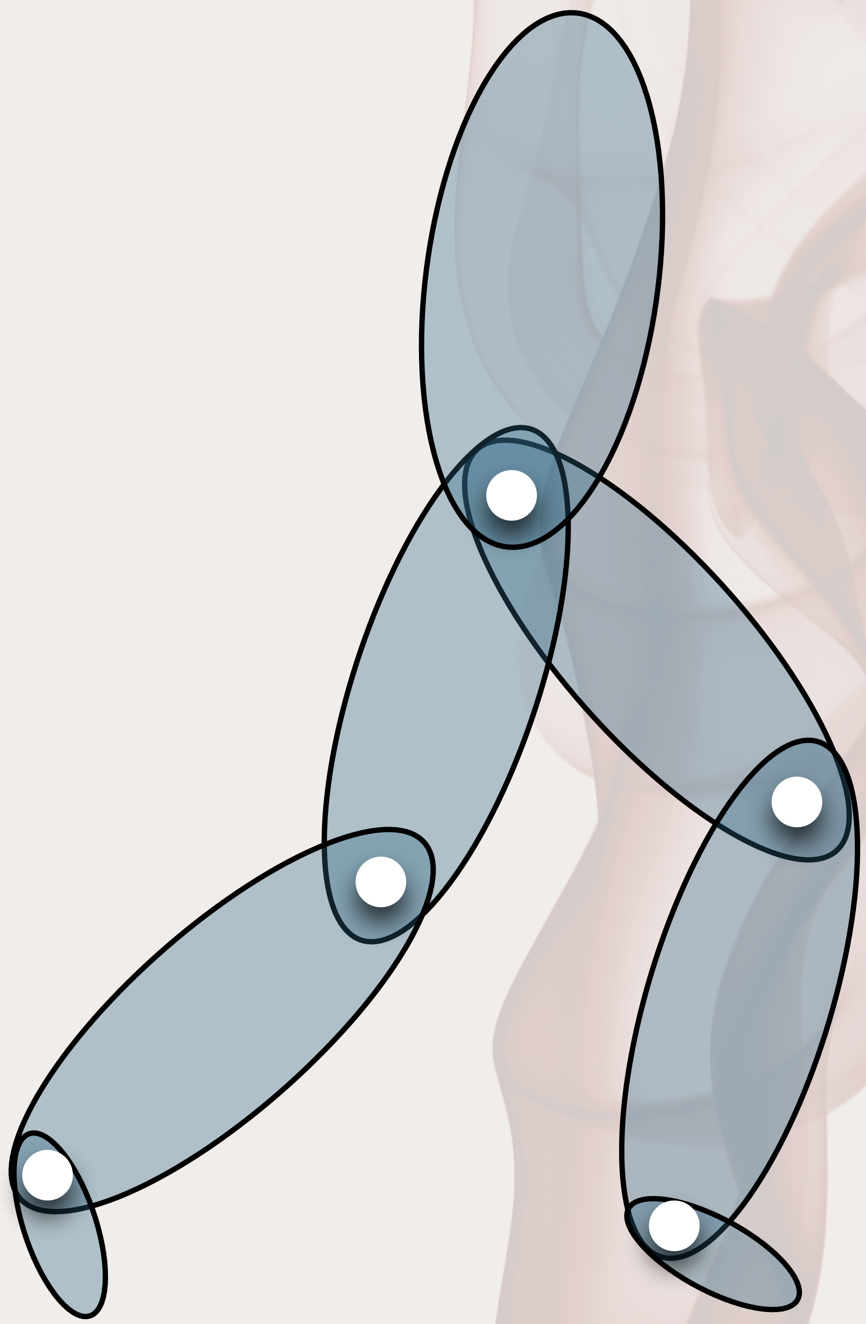


Joint Enforcement

- Penalty Methods
- Contraint Methods
  - aka Maximal Coordinate
- Minimal Coordinates

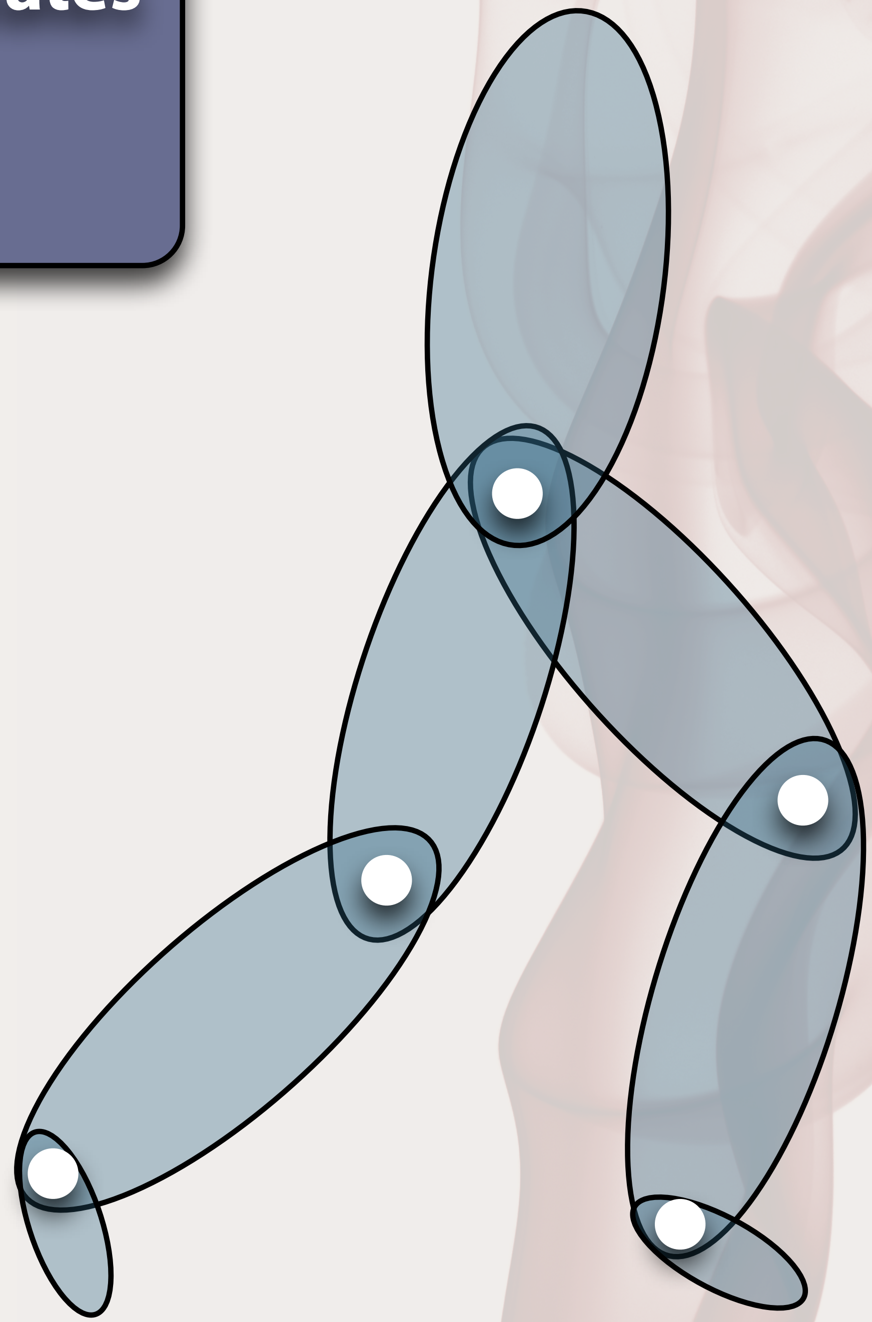


Internal Coordinates



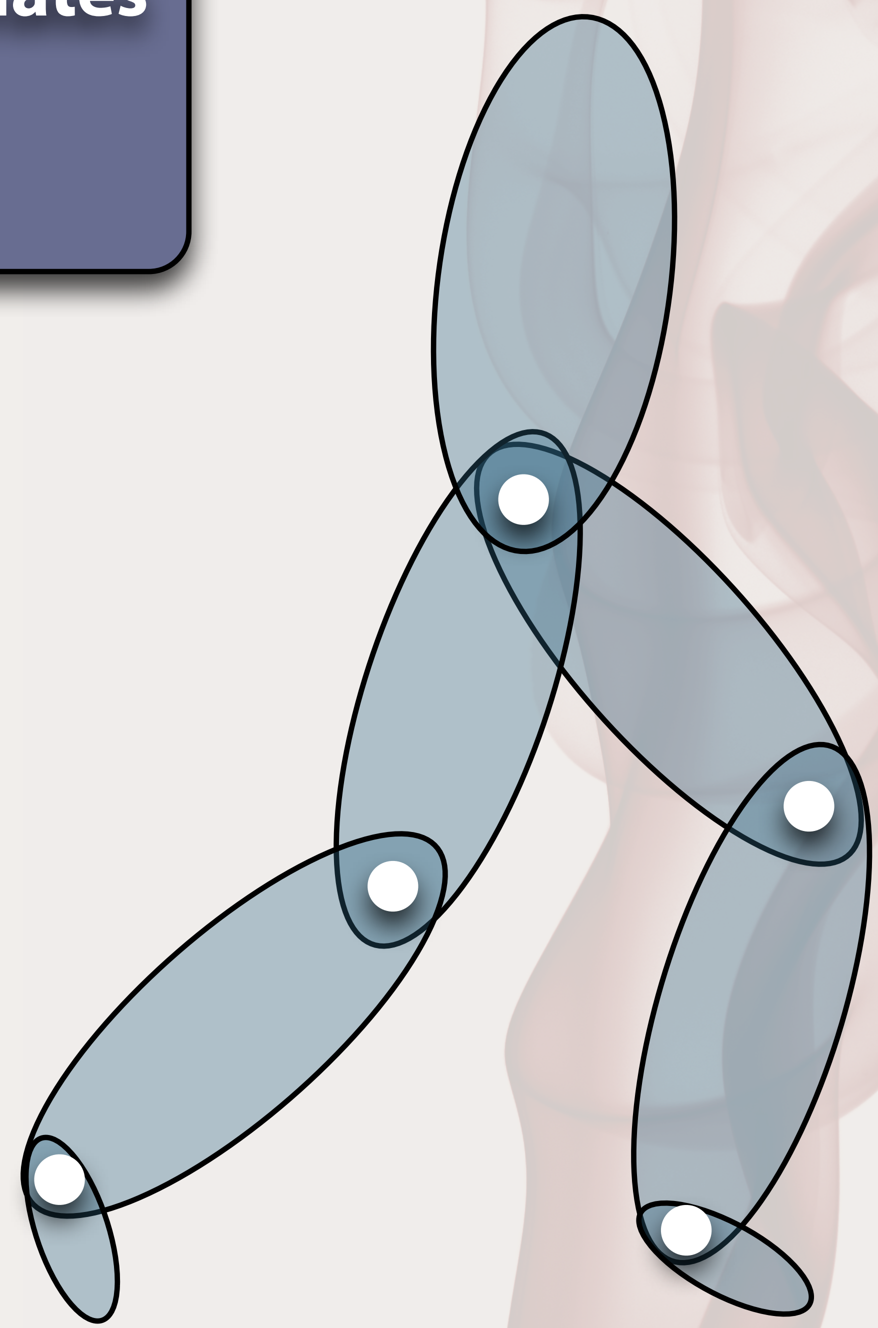
Internal Coordinates

- q: the skeletal coordinates



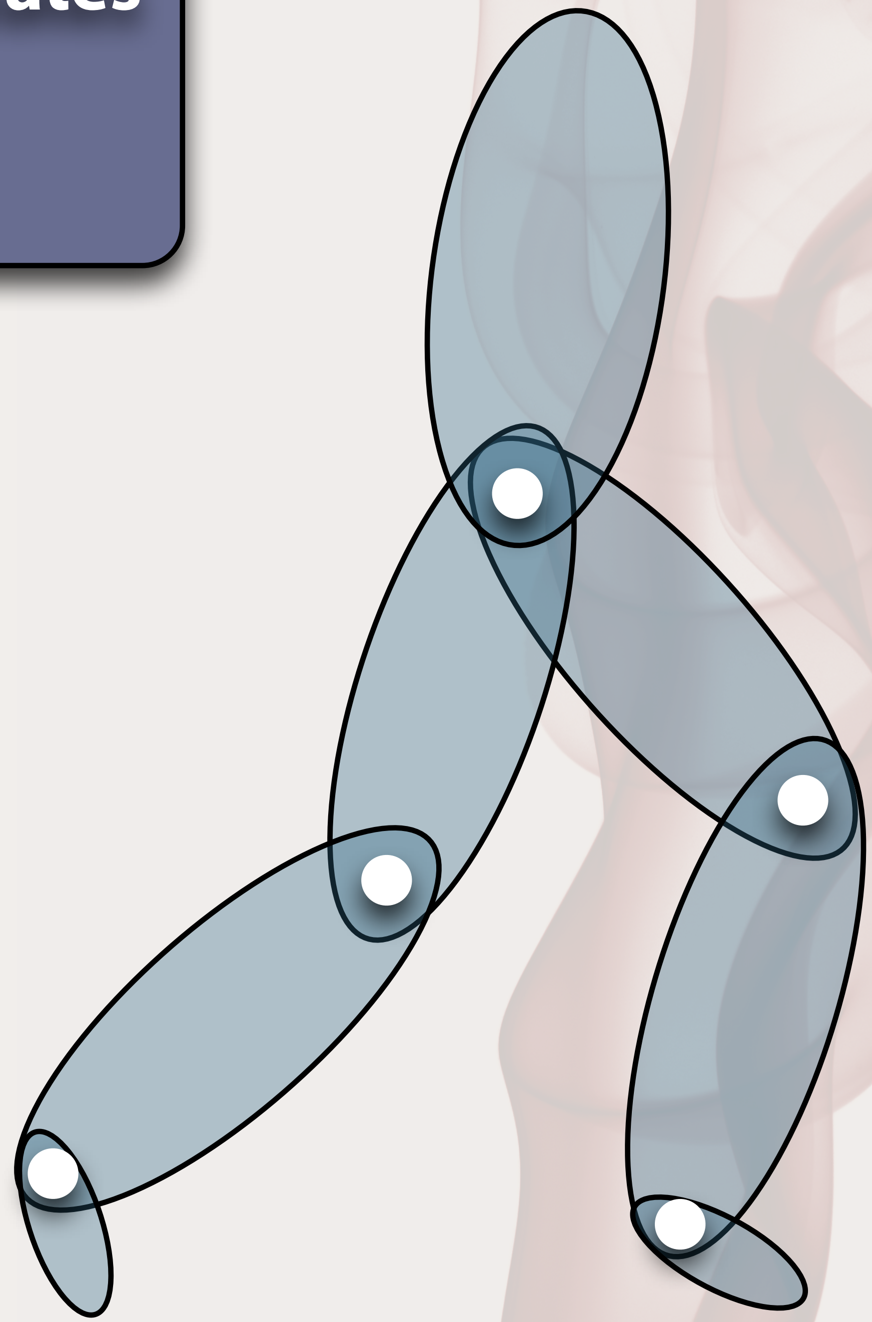
Internal Coordinates

- q: the skeletal coordinates
- $\dot{q}$: joint velocities



Internal Coordinates

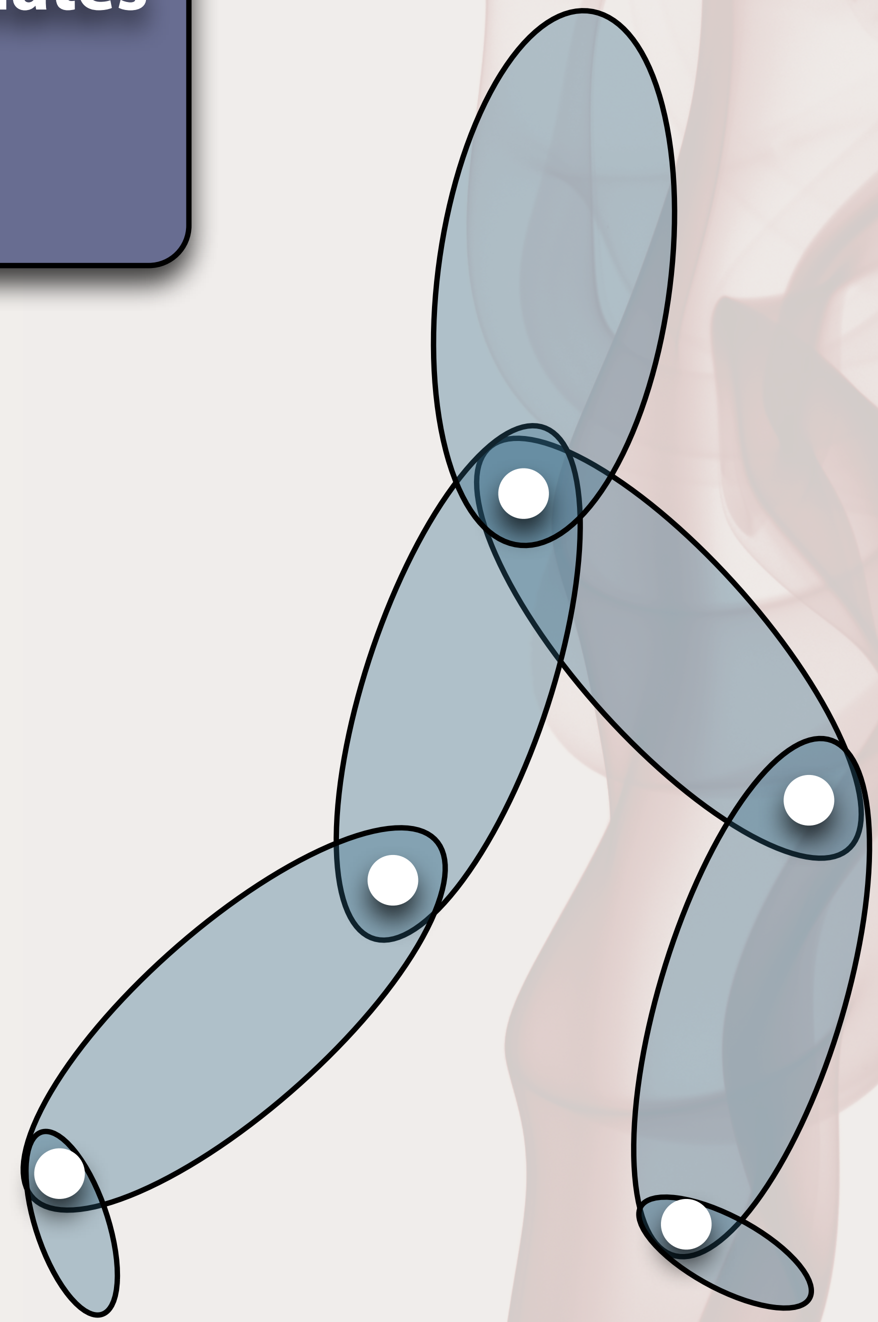
- q: the skeletal coordinates
- $\dot{q}$: joint velocities
- $\ddot{q}$: joint accelerations



Internal Coordinates

- q: the skeletal coordinates
- $\dot{q}$: joint velocities
- $\ddot{q}$: joint accelerations

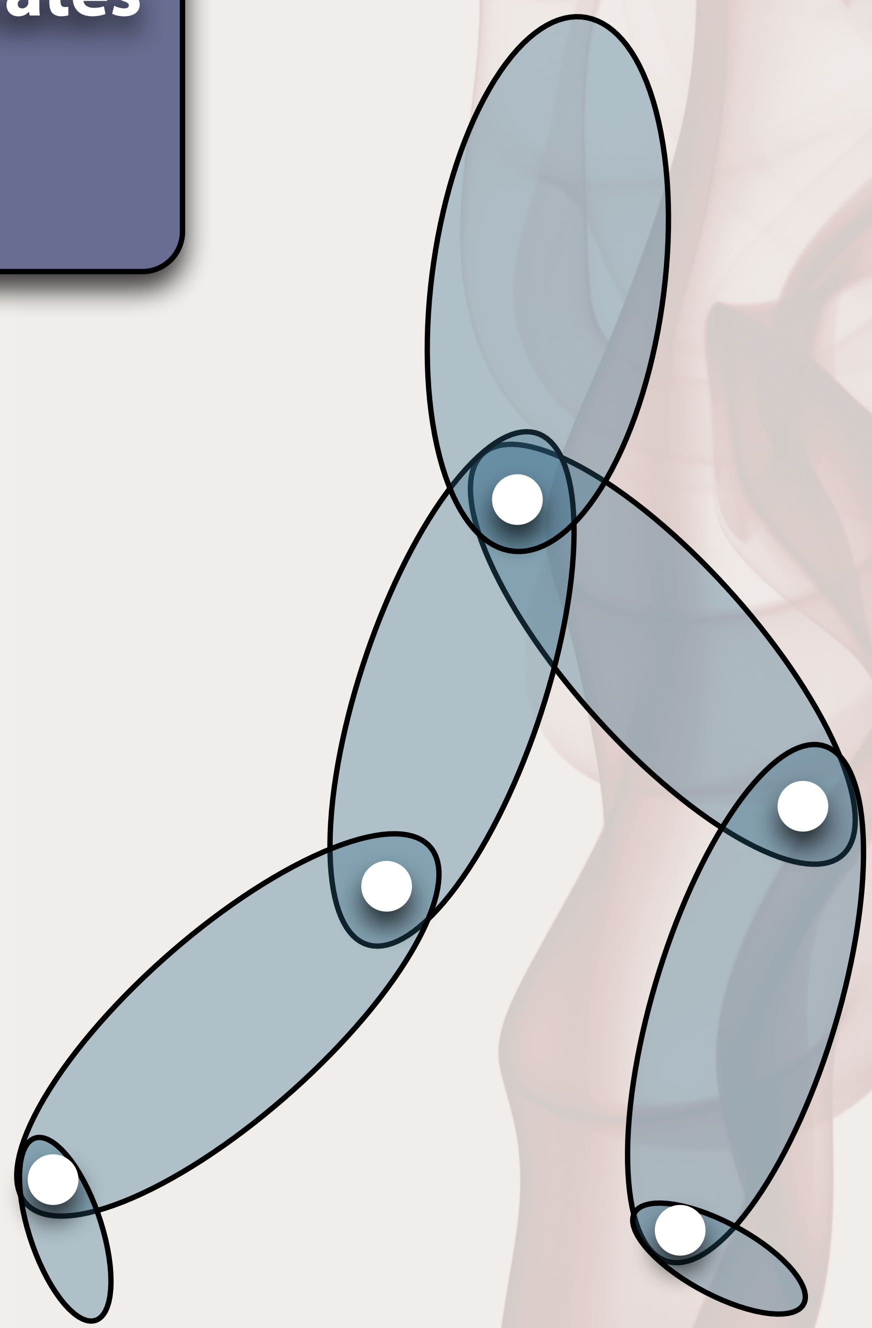
- ***Forward Dynamics Problem***
  - **Compute $\ddot{q} = F(q, \dot{q}, f, \tau)$**



Internal Coordinates

- q: the skeletal coordinates
- $\dot{q}$: joint velocities
- $\ddot{q}$: joint accelerations

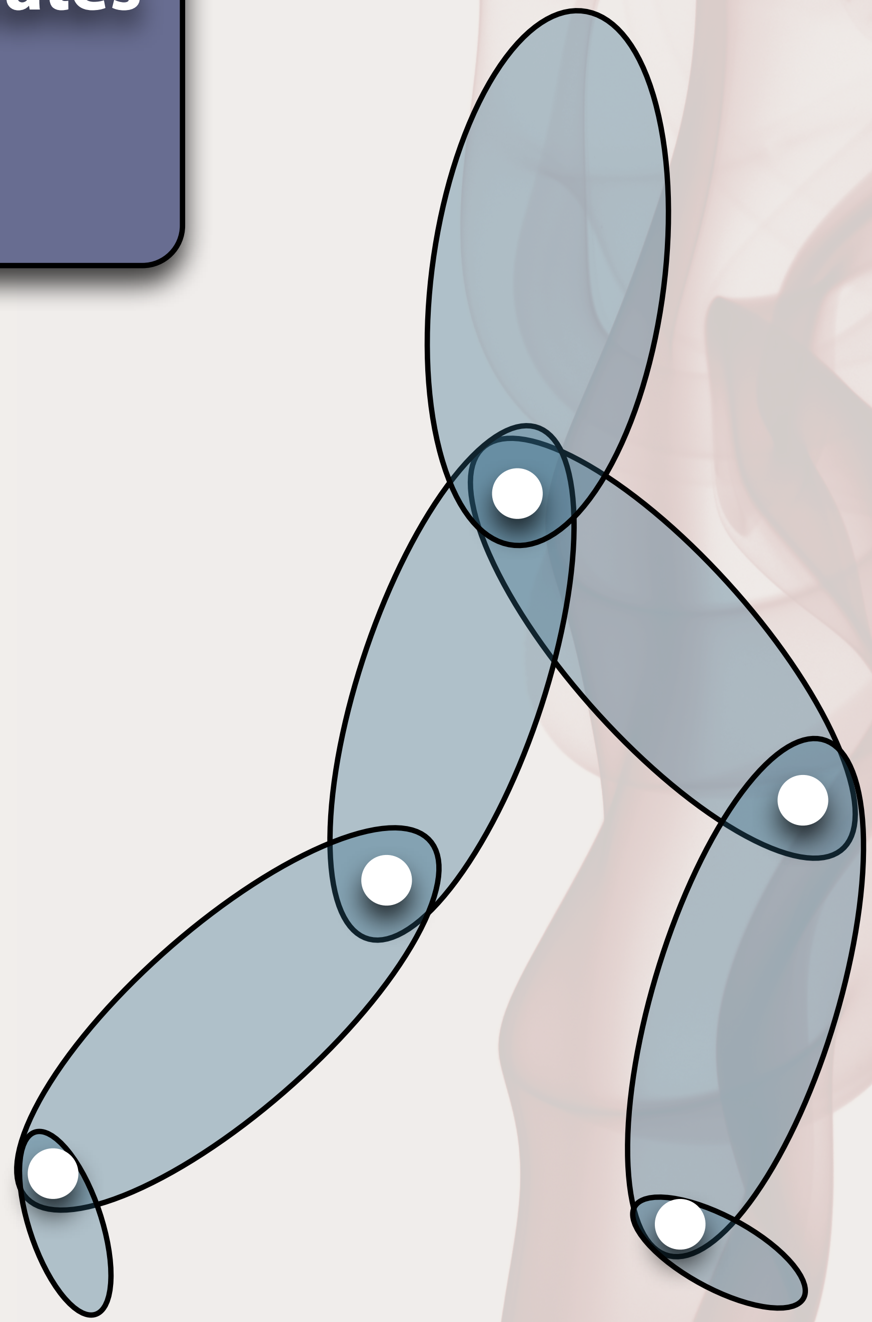
- ***Forward Dynamics Problem***
  - **Compute $\ddot{q} = F(q, \dot{q}, f, \tau)$**
    - **f: external forces**



Internal Coordinates

- q: the skeletal coordinates
- $\dot{q}$: joint velocities
- $\ddot{q}$: joint accelerations

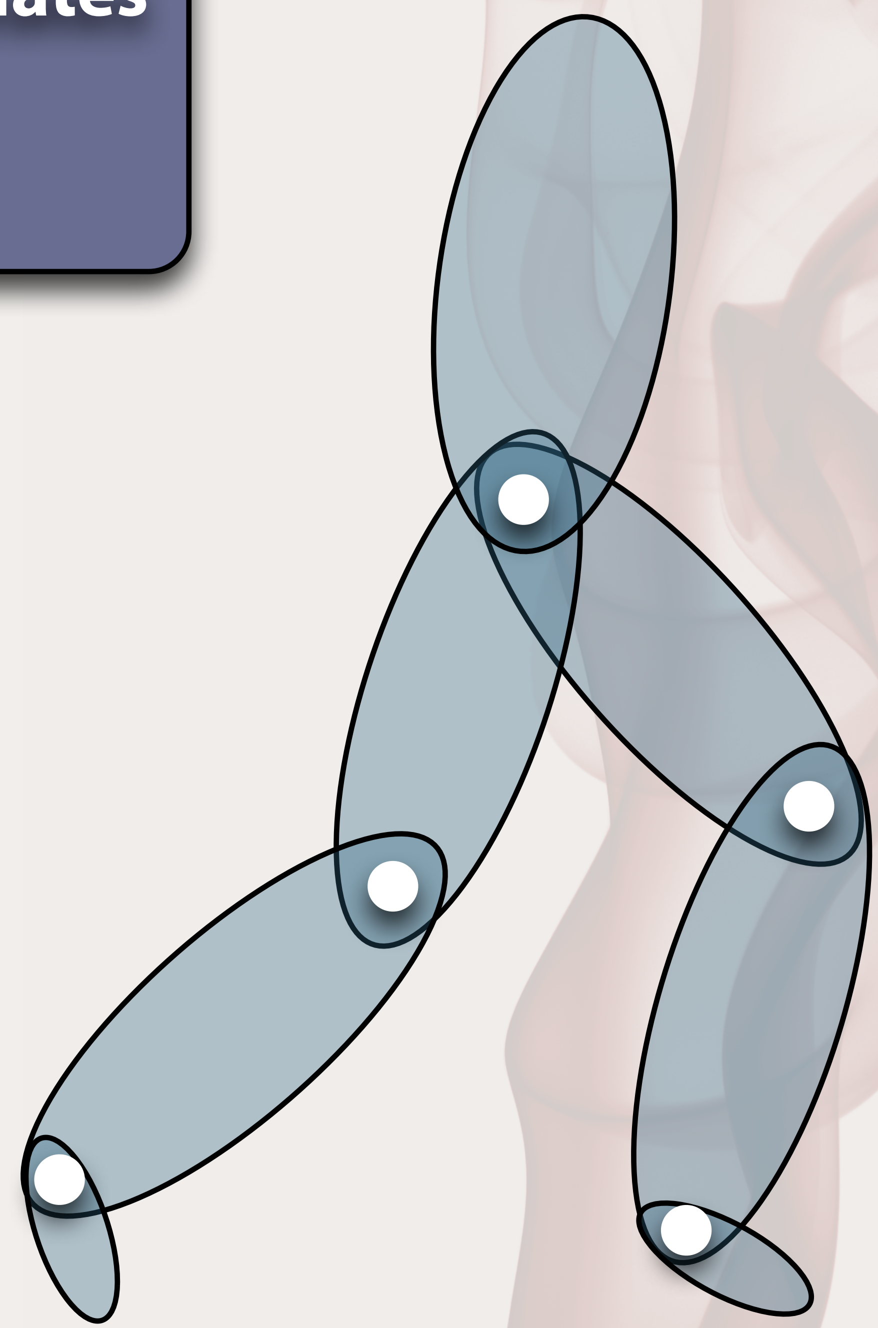
- ***Forward Dynamics Problem***
  - **Compute $\ddot{q} = F(q, \dot{q}, f, \tau)$**
    - **f: external forces**
    - **τ: internal torques**



Internal Coordinates

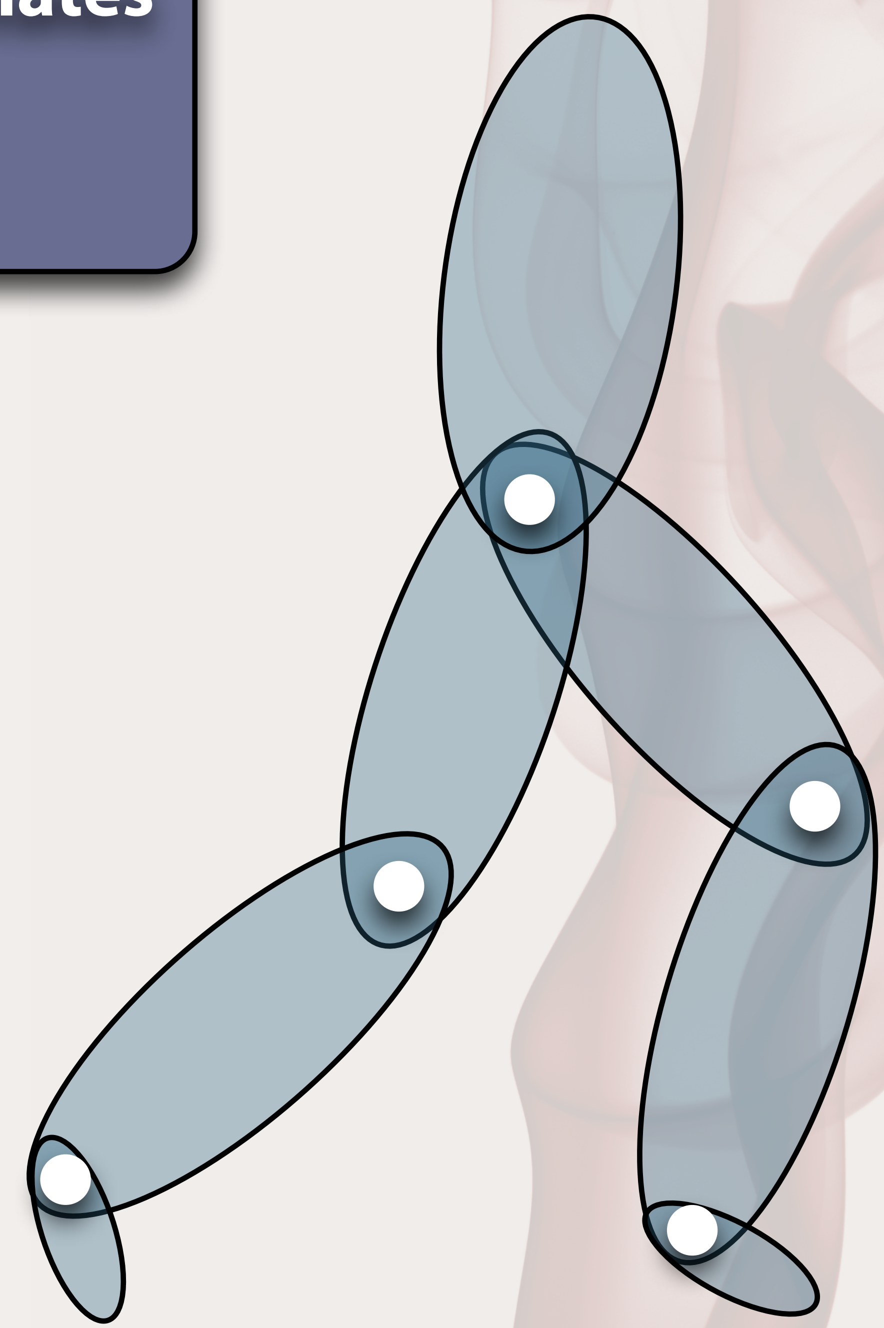
- q: the skeletal coordinates
- $\dot{q}$: joint velocities
- $\ddot{q}$: joint accelerations

- ***Forward Dynamics Problem***
  - **Compute $\ddot{q} = F(q, \dot{q}, f, \tau)$**
    - **f: external forces**
    - **τ: internal torques**
  - **Then use ODE solver.**



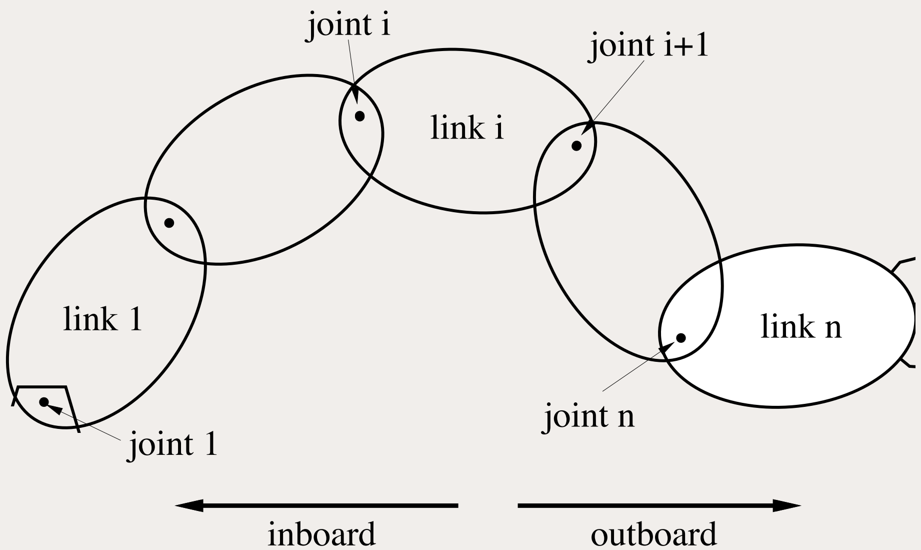
Internal Coordinates

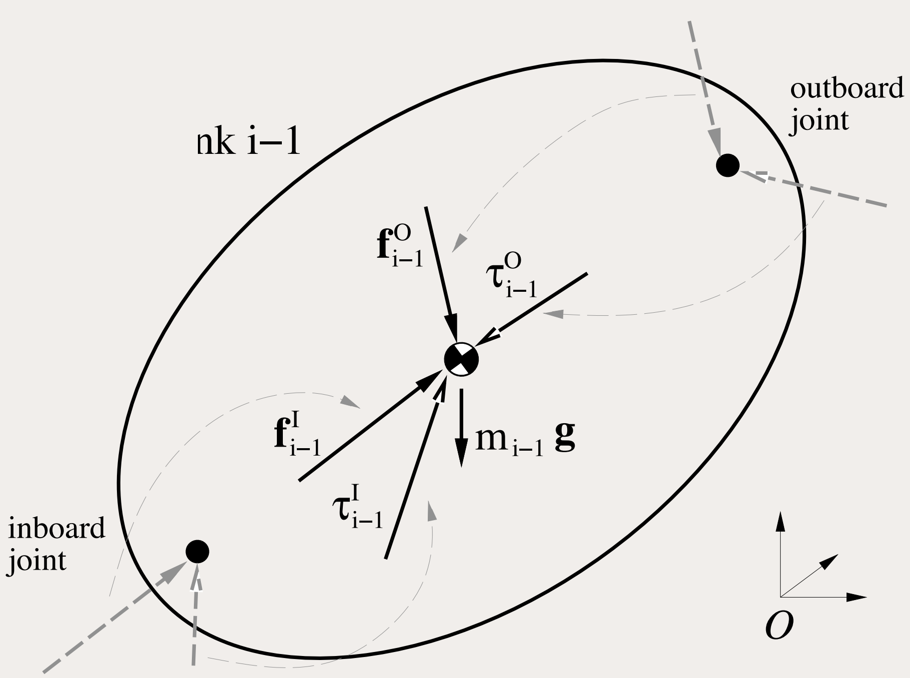
- q: the skeletal coordinates
- $\dot{q}$: joint velocities
- $\ddot{q}$: joint accelerations



- ***Forward Dynamics Problem***
  - **Compute $\ddot{q} = F(q, \dot{q}, f, \tau)$**
    - **f: external forces**
    - **τ: internal torques**
  - **Then use ODE solver.**
- ***Inverse Dynamics Problem***
  - **Compute $\tau = G(q, \dot{q}, \ddot{q}, f)$**

Featherstone Algorithm





Impulse-based Dynamic Simulation of Rigid Body Systems

by

Brian Vincent Mirtich

If joint i is prismatic,

$$\hat{\mathbf{s}}_i'\hat{\mathbf{f}}_i^I = \begin{bmatrix}\mathbf{0}\\ \mathbf{u}_i\end{bmatrix}' \begin{bmatrix}\mathbf{f}\\ \boldsymbol{\tau}-\mathbf{d}_i\times\mathbf{f}\end{bmatrix} = \mathbf{f}\cdot\mathbf{u}_i.$$

The right hand side is the component of the applied force along the joint axis. This force must be supported by the actuator, hence, it is Q_i. If joint i is revolute,

$$\hat{\mathbf{s}}_i'\hat{\mathbf{f}}_i^I = \begin{bmatrix}\mathbf{u}_i\\ \mathbf{u}\times\mathbf{d}_i\end{bmatrix}' \begin{bmatrix}\mathbf{f}\\ \boldsymbol{\tau}-\mathbf{d}_i\times\mathbf{f}\end{bmatrix} = \mathbf{f}\cdot(\mathbf{u}_i\times\mathbf{d}_i) + (\boldsymbol{\tau}-\mathbf{d}_i\times\mathbf{f})\cdot\mathbf{u}_i.$$

The right hand side reduces to $\boldsymbol{\tau}\cdot\mathbf{u}_i$, the component of the applied torque along the joint axis. This torque must be supported by the actuator, hence, it is Q_i. □

Substituting equation (4.23) for link i's spatial acceleration into (4.24) yields

$$\hat{\mathbf{f}}_i^I = \hat{\mathbf{I}}_i^A({}_i\hat{\mathbf{X}}_{i-1}\hat{\mathbf{a}}_{i-1} + \ddot{q}_i\hat{\mathbf{s}}_i + \hat{\mathbf{c}}_i) + \hat{\mathbf{Z}}_i^A.$$

Premultiplying both sides by $\hat{\mathbf{s}}_i'$ and applying Lemma 7 gives

$$Q_i = \hat{\mathbf{s}}_i'\hat{\mathbf{I}}_i^A({}_i\hat{\mathbf{X}}_{i-1}\hat{\mathbf{a}}_{i-1} + \ddot{q}_i\hat{\mathbf{s}}_i + \hat{\mathbf{c}}_i) + \hat{\mathbf{s}}_i'\hat{\mathbf{Z}}_i^A,$$

from which $\ddot{q}_i$ may be determined:

$$\ddot{q}_i = \frac{Q_i - \hat{\mathbf{s}}_i'\hat{\mathbf{I}}_i^A\,{}_i\hat{\mathbf{X}}_{i-1}\hat{\mathbf{a}}_{i-1} - \hat{\mathbf{s}}_i'\left(\hat{\mathbf{Z}}_i^A + \hat{\mathbf{I}}_i^A\hat{\mathbf{c}}_i\right)}{\hat{\mathbf{s}}_i'\hat{\mathbf{I}}_i^A\hat{\mathbf{s}}_i}. \qquad (4.27)$$

Substituting this expression for $\ddot{q}_i$ into (4.26) and rearranging gives

$$\begin{aligned}\hat{\mathbf{f}}_{i-1}^I \;=\; & \left[\hat{\mathbf{I}}_{i-1} + {}_{i-1}\hat{\mathbf{X}}_i\left(\hat{\mathbf{I}}_i^A - \frac{\hat{\mathbf{I}}_i^A\hat{\mathbf{s}}_i\hat{\mathbf{s}}_i'\hat{\mathbf{I}}_i^A}{\hat{\mathbf{s}}_i'\hat{\mathbf{I}}_i^A\hat{\mathbf{s}}_i}\right){}_i\hat{\mathbf{X}}_{i-1}\right]\hat{\mathbf{a}}_{i-1} \\ & + \hat{\mathbf{Z}}_{i-1} + {}_{i-1}\hat{\mathbf{X}}_i\left[\hat{\mathbf{Z}}_i^A + \hat{\mathbf{I}}_i^A\hat{\mathbf{c}}_i + \frac{\hat{\mathbf{I}}_i^A\hat{\mathbf{s}}_i\left[Q_i - \hat{\mathbf{s}}_i'\left(\hat{\mathbf{Z}}_i^A + \hat{\mathbf{I}}_i^A\hat{\mathbf{c}}_i\right)\right]}{\hat{\mathbf{s}}_i'\hat{\mathbf{I}}_i^A\hat{\mathbf{s}}_i}\right].\end{aligned}$$

Comparing this to the desired form (4.24),

$$\hat{\mathbf{I}}_{i-1}^A = \hat{\mathbf{I}}_{i-1} + {}_{i-1}\hat{\mathbf{X}}_i\left(\hat{\mathbf{I}}_i^A - \frac{\hat{\mathbf{I}}_i^A\hat{\mathbf{s}}_i\hat{\mathbf{s}}_i'\hat{\mathbf{I}}_i^A}{\hat{\mathbf{s}}_i'\hat{\mathbf{I}}_i^A\hat{\mathbf{s}}_i}\right){}_i\hat{\mathbf{X}}_{i-1} \qquad (4.28)$$

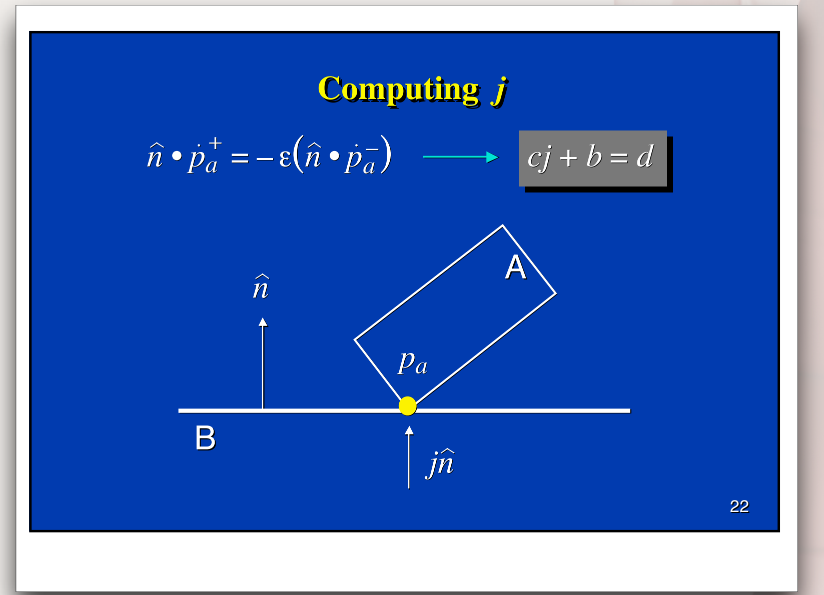
$$\hat{\mathbf{Z}}_{i-1}^A = \hat{\mathbf{Z}}_{i-1} + {}_{i-1}\hat{\mathbf{X}}_i\left[\hat{\mathbf{Z}}_i^A + \hat{\mathbf{I}}_i^A\hat{\mathbf{c}}_i + \frac{\hat{\mathbf{I}}_i^A\hat{\mathbf{s}}_i\left[Q_i - \hat{\mathbf{s}}_i'\left(\hat{\mathbf{Z}}_i^A + \hat{\mathbf{I}}_i^A\hat{\mathbf{c}}_i\right)\right]}{\hat{\mathbf{s}}_i'\hat{\mathbf{I}}_i^A\hat{\mathbf{s}}_i}\right]. \qquad (4.29)$$

Constraints

Constraints

- **Accelerations are *linear* in applied torques and forces.**

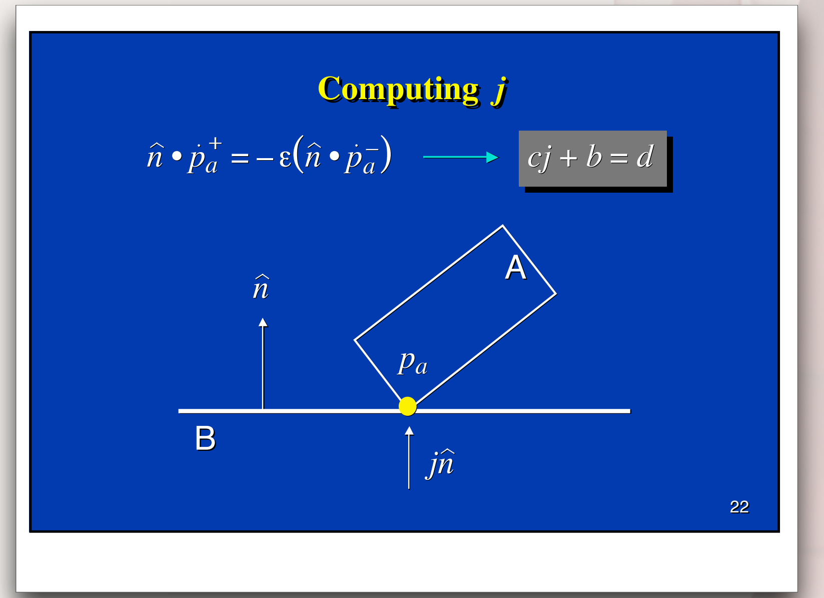
$$a_f = kf + a_0$$



Constraints

- **Accelerations are *linear* in applied torques and forces.**

$$a_f = kf + a_0$$

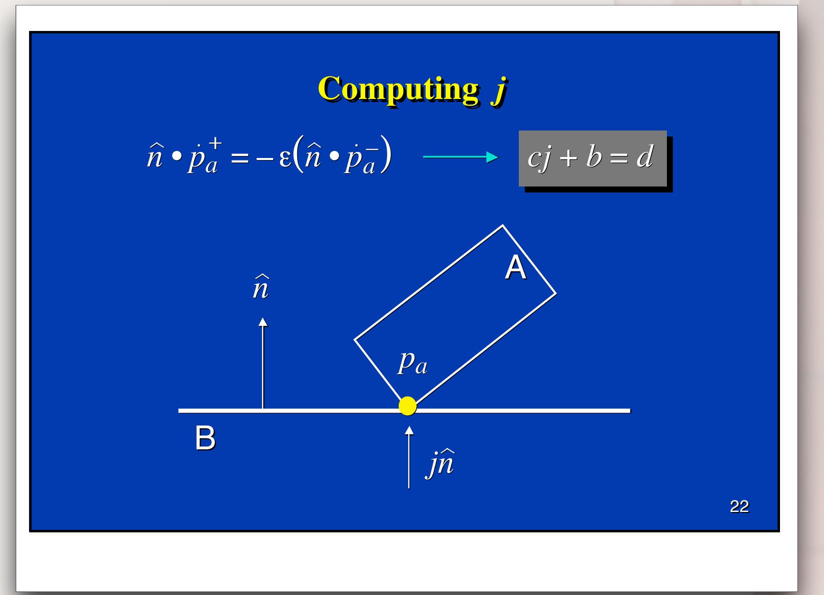


- **Use of "test forces"**

Constraints

- **Accelerations are *linear* in applied torques and forces.**

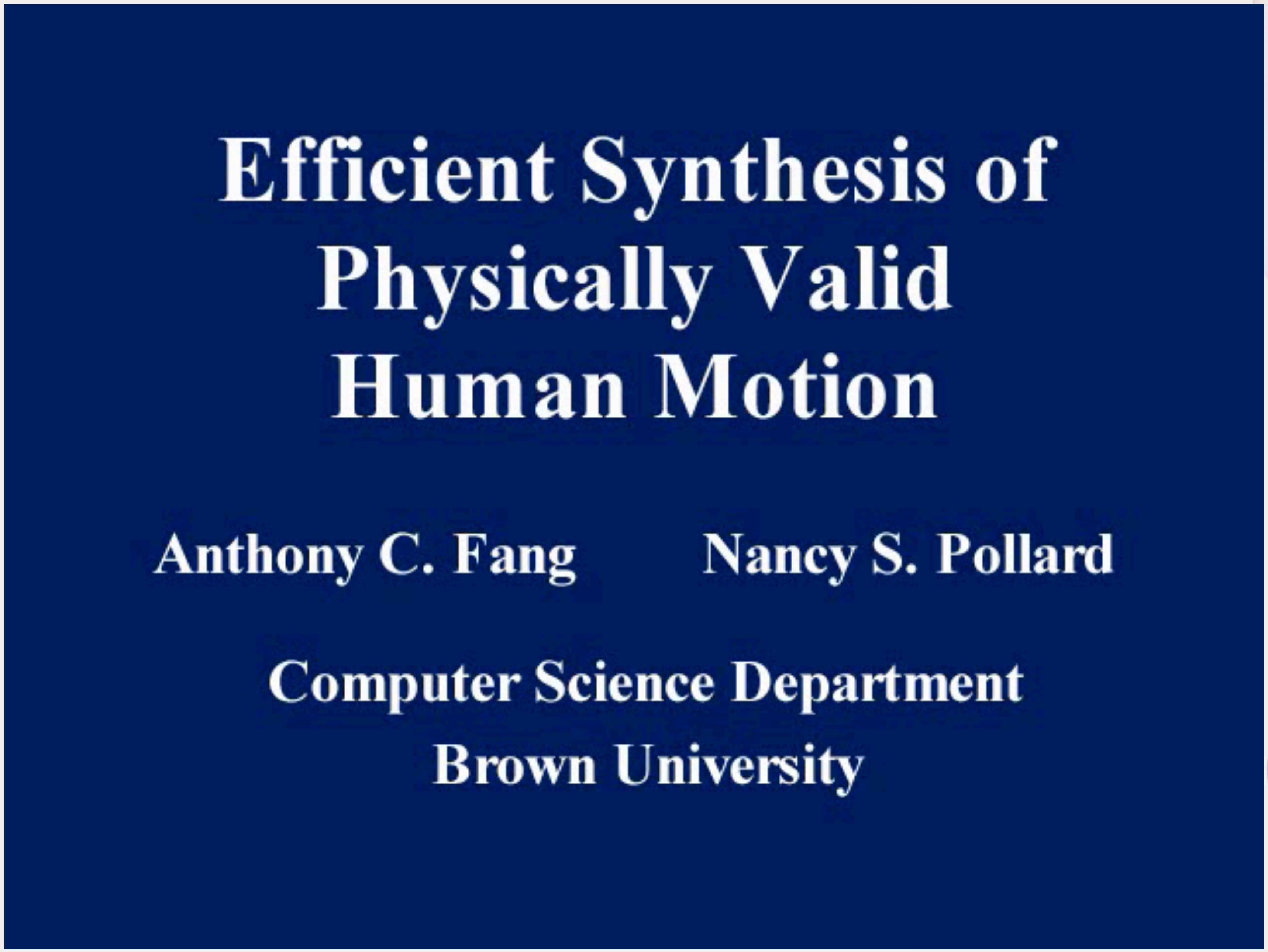
$$a_f = kf + a_0$$



- **Use of "test forces"**
- **Multiple test forces**

$$\mathbf{a}_f = K\mathbf{f} + \mathbf{a}_0$$

Examples


Efficient Synthesis of
Physically Valid
Human Motion
Anthony C. Fang
Nancy S. Pollard
Computer Science Department
Brown University

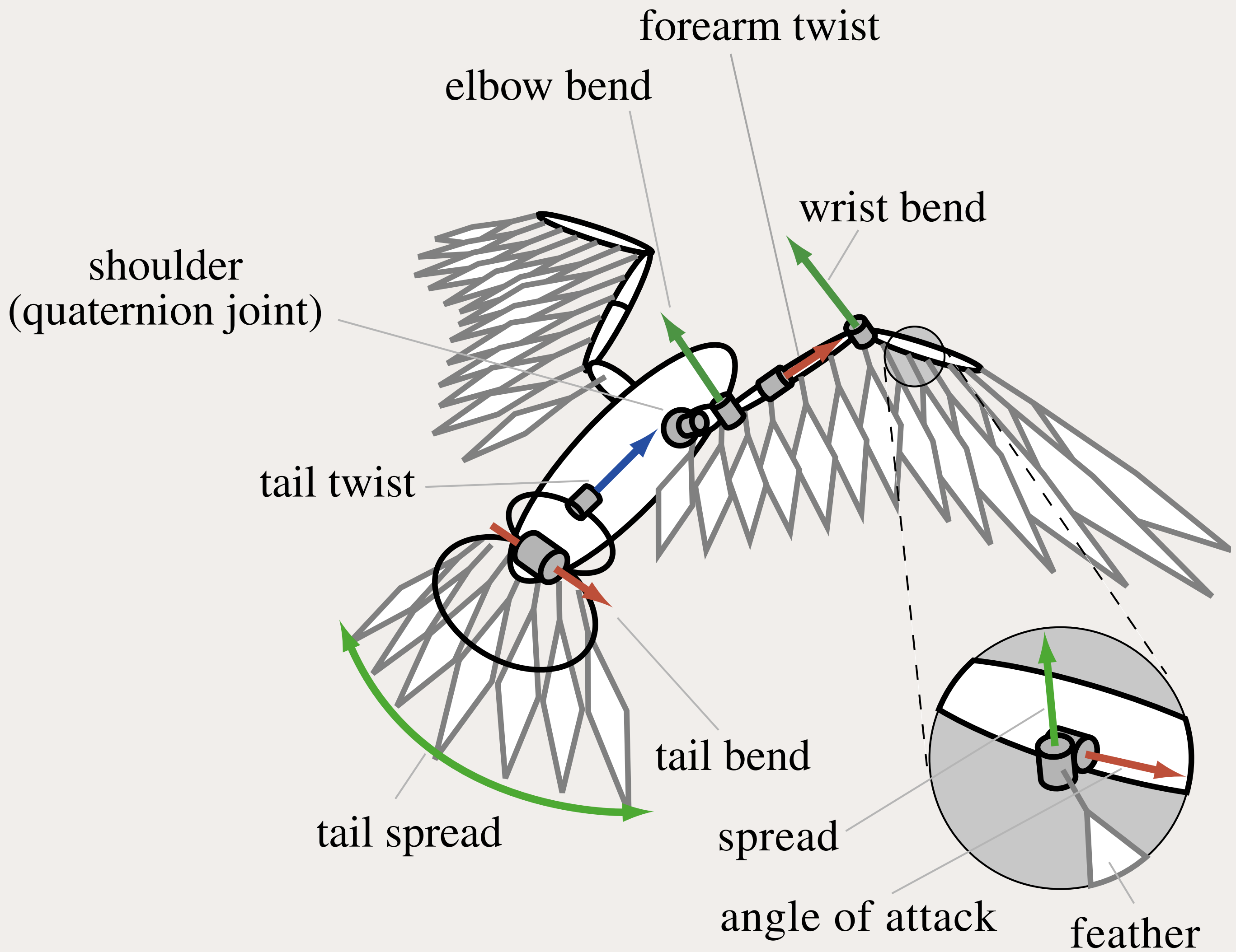
Animation Topics

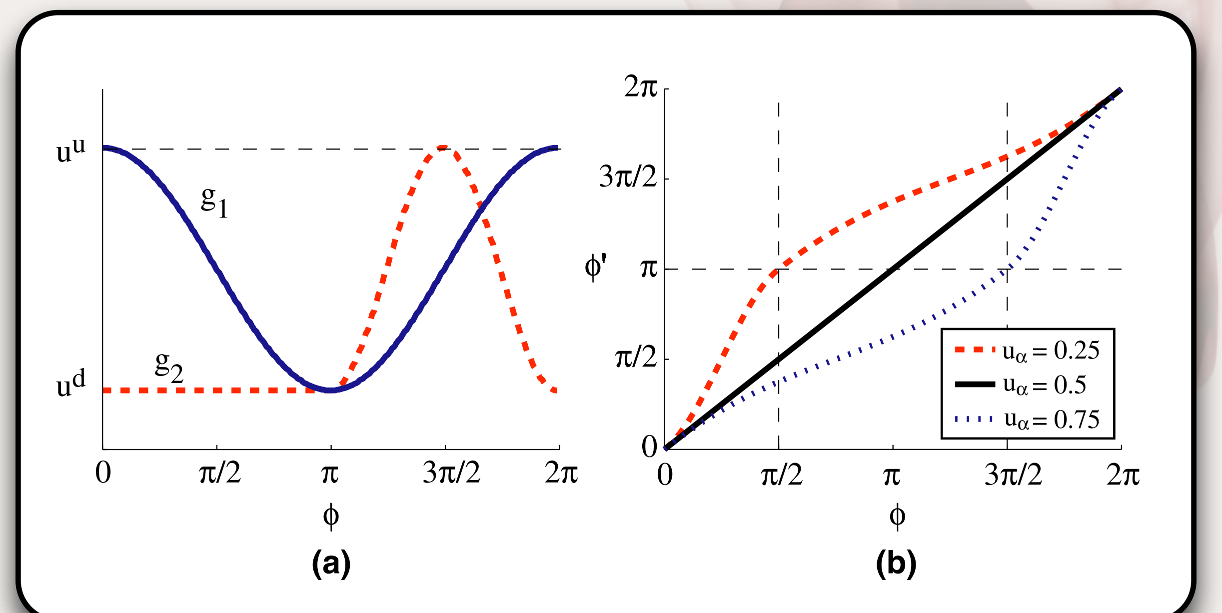
- Data-Driven Motion
- **Physics Based Motion**
- Motion of other Animals

Animation Topics

- Data-Driven Motion
- Physics Based Motion
- **Motion of other Animals**

Bird Flight

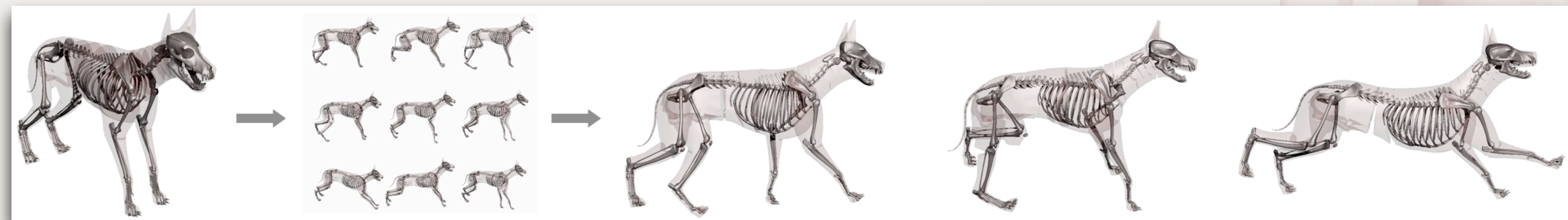




source: Wu and Popović [2003]

Bird Flight Examples

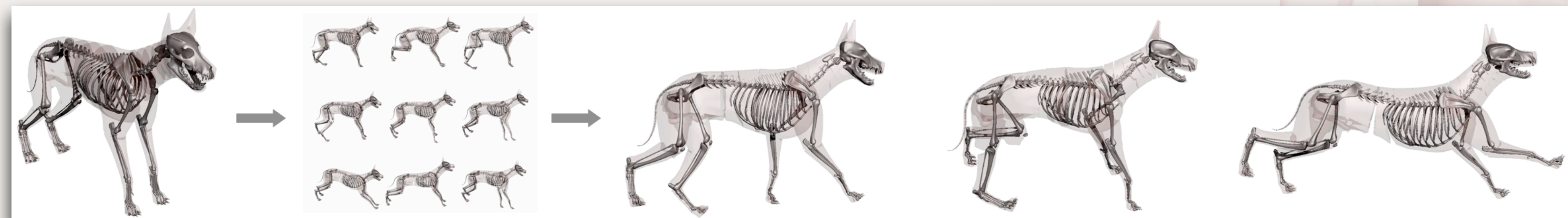
Dogs



Model Vibration Modes Trot = 3 + 14 + 20 Amble = 4 + 14 + 20 Gallop = 0 + 1 + 3

source: Kry et al [2007]

Dogs



Model Vibration Modes Trot = 3 + 14 + 20 Amble = 4 + 14 + 20 Gallop = 0 + 1 + 3

source: Kry et al [2007]

Animation Topics

- Data-Driven Motion
- Physics Based Motion
- **Motion of other Animals**

Overview

- State of the art.
- Body models.
- **Animation**
- Questions

source: 3dscience.com

Overview

- State of the art.
- Body models.
- Animation
- **Questions**

source: 3dscience.com

Questions

- What might be the best physics control paradigm from the animator's perspective?
- How can we control physics (in general)?
- What about linear dynamics (like springs), vs. nonlinear dynamics (like fluids)?
- What about constraints? (e.g. joints)

Student Answewers

- What might be the best physics control paradigm from the animator's perspective?
    - Direct control
        - bezier curves
            - including editing "higher level controls" like accelaration
        - directly editing the density field (in a fluid)
    - "Keyframe" paradigm
    - Many worlds browsing
        - simulate a bunch and then see what looks good
- How can we control physics (in general)?
    - Let the user specify initial conditions
        - And make physical constants customizable
    - Interpolate the state between the keyframes
        - kinematically
        - dynaically
            - this becomes an <u>optimization</u>
    - Set up an energy functions
        - some specify "goals"
        - some specify "not goals"
        - and <u>optimize</u> it
    - we can <u>combine</u> these
        - micro and macro controls
- What about linear dynamics (like springs), vs. nonlinear dynamics (like fluids)?
    - perhaps nonlineararity is a benefit!!! (people don't know what's "real")
- What about constraints? (e.g. joints)
    - can we use these as a form of control?

Overview

- State of the art.
- Body models.
- Animation
- **Questions**

source: 3dscience.com

Overview

- State of the art.
- Body models.
- Animation
- Questions

source: 3dscience.com