

# コンピュータグラフィックス論

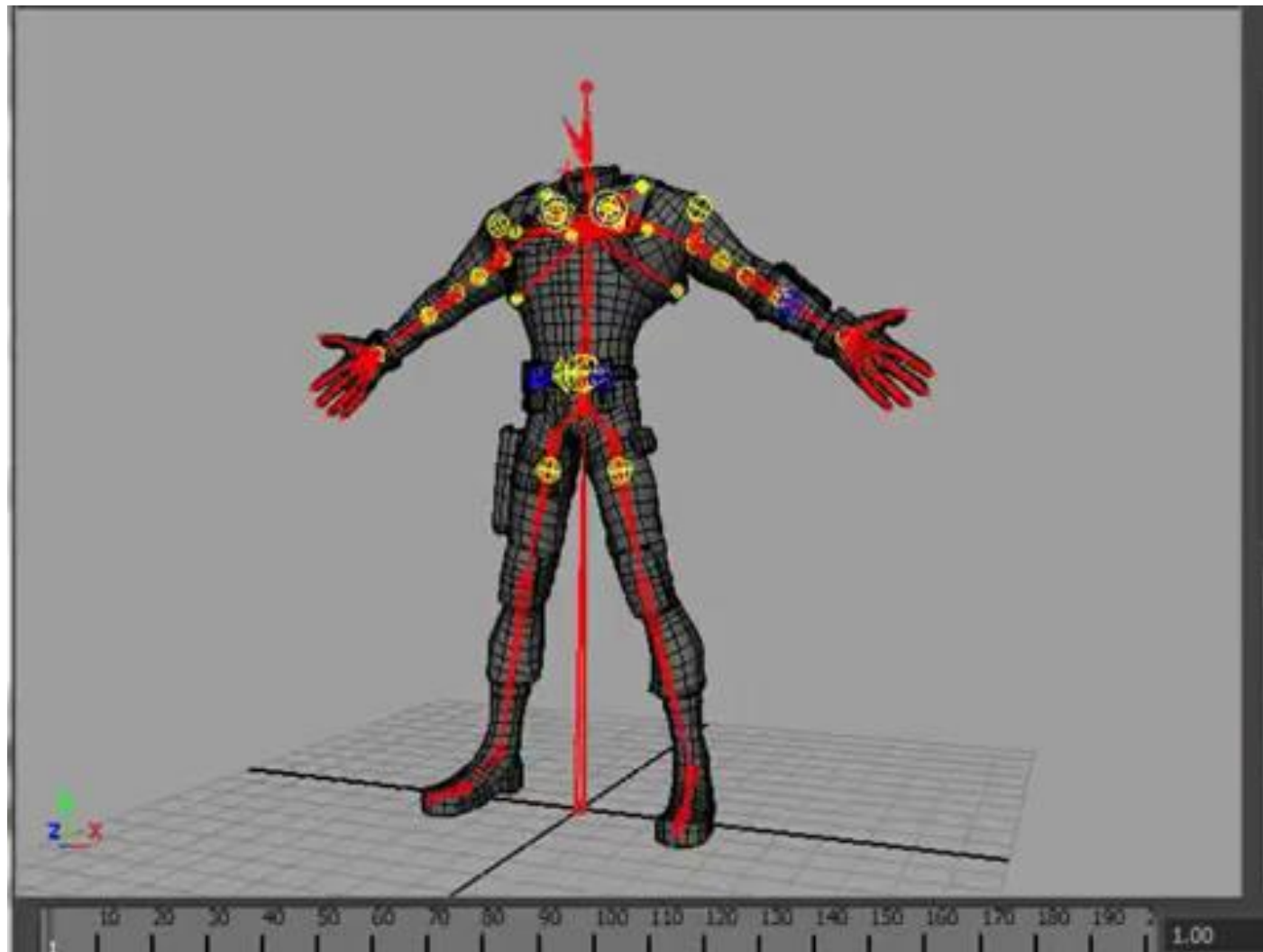
## －アニメーション(1)－

2019年5月23日

高山 健志

# スケルトンによるアニメーション

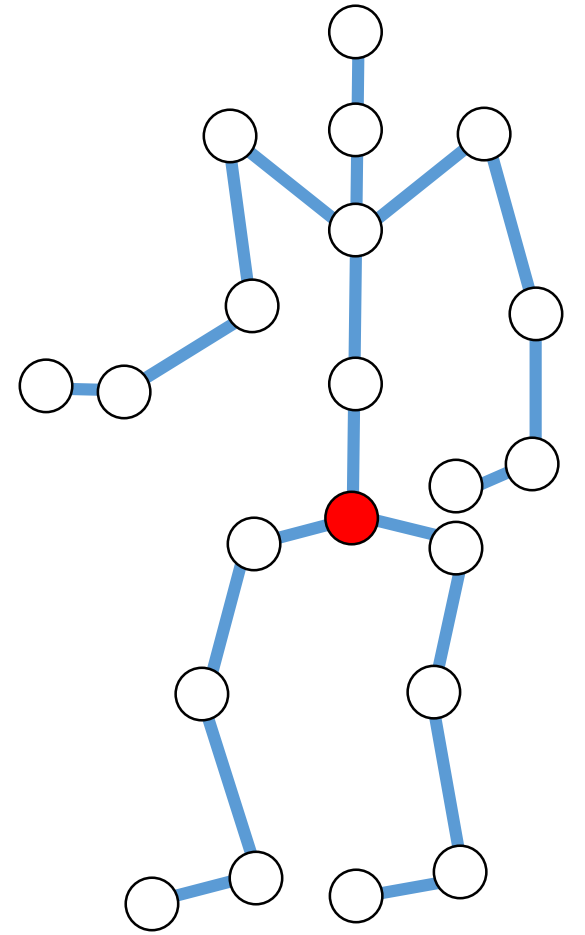
- 単純な仕組み
- 直感的な挙動
- 低い計算コスト



<https://www.youtube.com/watch?v=DsoNab58QVA>

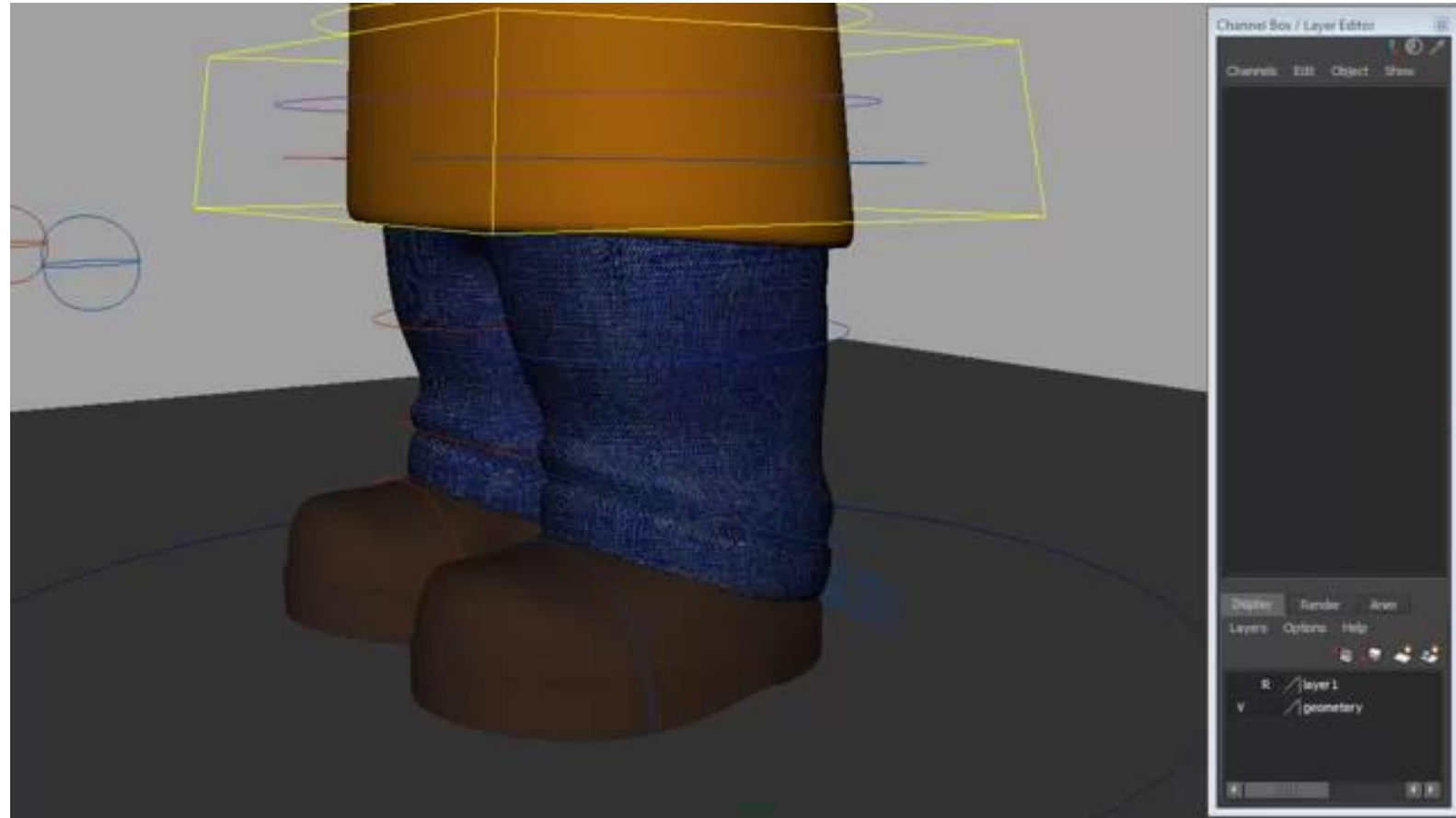
# スケルトンによる姿勢の表現

- ボーンと関節から成る木構造
- ボーンは親関節を基準とした相対的な回転角を保持
- 各関節の回転角によって全体の姿勢を決定  
(**F**orward **K**inematics)
- ロボティクス分野と深く関連



# Inverse Kinematics

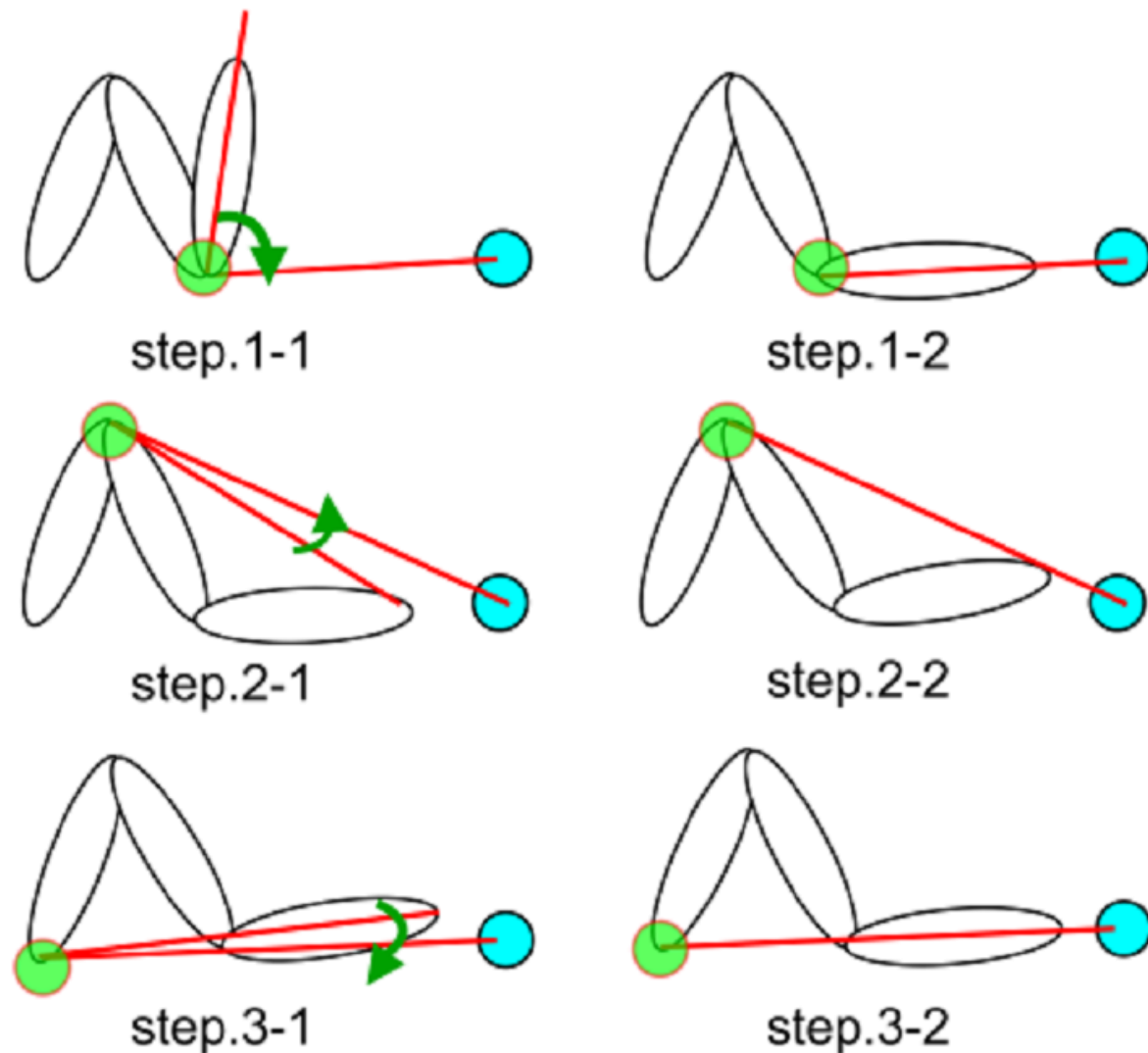
- 末端関節の位置を与えると、それを満たす関節角を逆算
- IK で手早く姿勢を作り、FK で微調整



[https://www.youtube.com/watch?v=e1qnZ9rV\\_kw](https://www.youtube.com/watch?v=e1qnZ9rV_kw)

# IK の一解法：Cyclic Coordinate Descent

- 関節角を一つずつ順番に変更
  - 末端関節を目標に近づける
  - 順番が重要！末端が最初
- 実装が簡単 → 基本課題 (デモ)
- より高度な手法
  - ヤコビ法 (方向等の様々な制約)
  - 変形エネルギーの最小化 [Jacobson 12]



# 変形エネルギーに基づく IK

## Fast Automatic Skinning Transformations

Alec Jacobson<sup>1</sup>

Ilya Baran<sup>2</sup>

Ladislav Kavan<sup>1</sup>

Jovan Popović<sup>3</sup>

Olga Sorkine<sup>1</sup>

<sup>1</sup>ETH Zurich

<sup>2</sup>Disney Research, Zurich

<sup>3</sup>Adobe Systems, Inc.

This video contains narration.

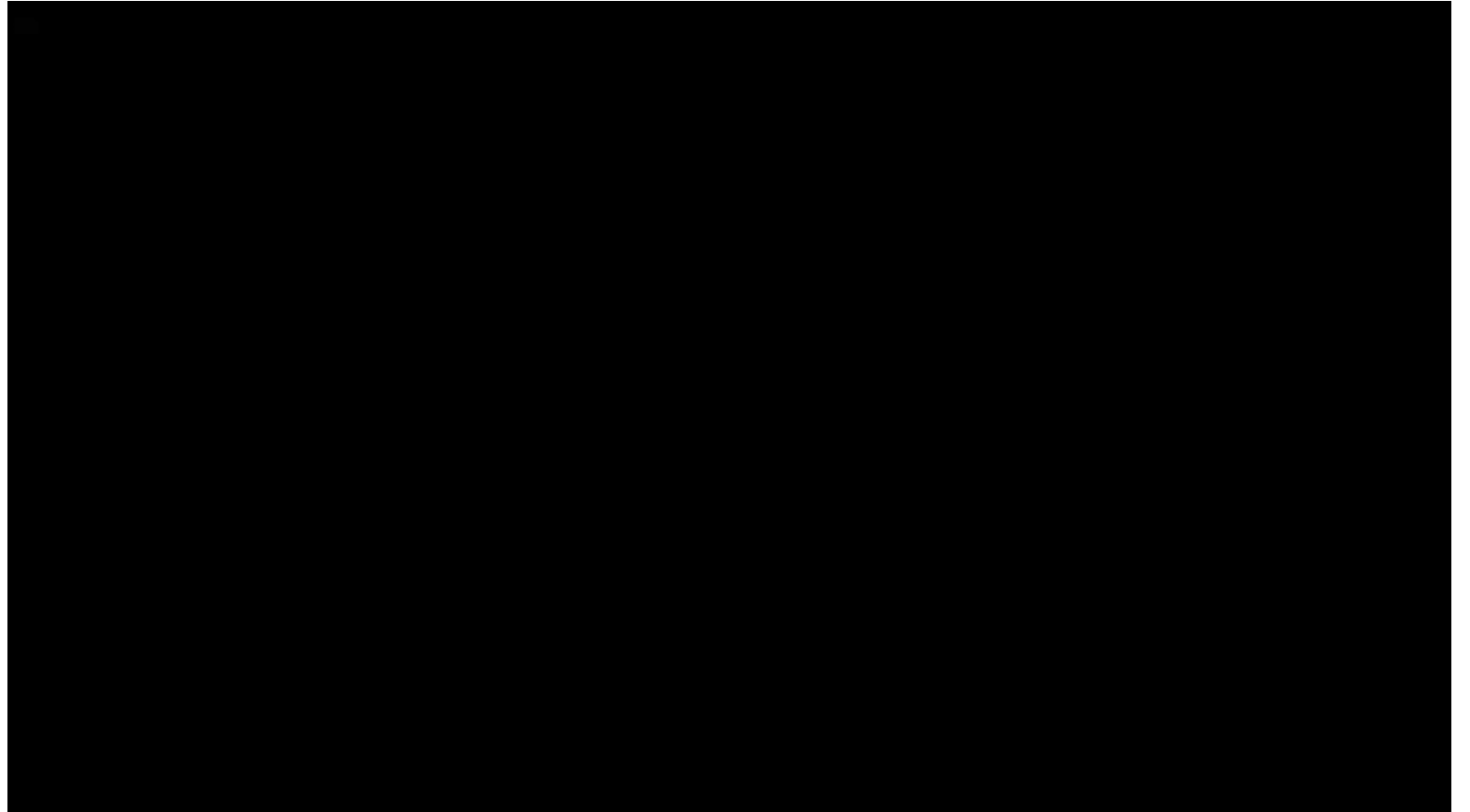
# モーションデータの取得・生成方法

# 光学式モーションキャプチャ

- 役者にマーカールを取り付け、多数 (~48) のカメラで撮影



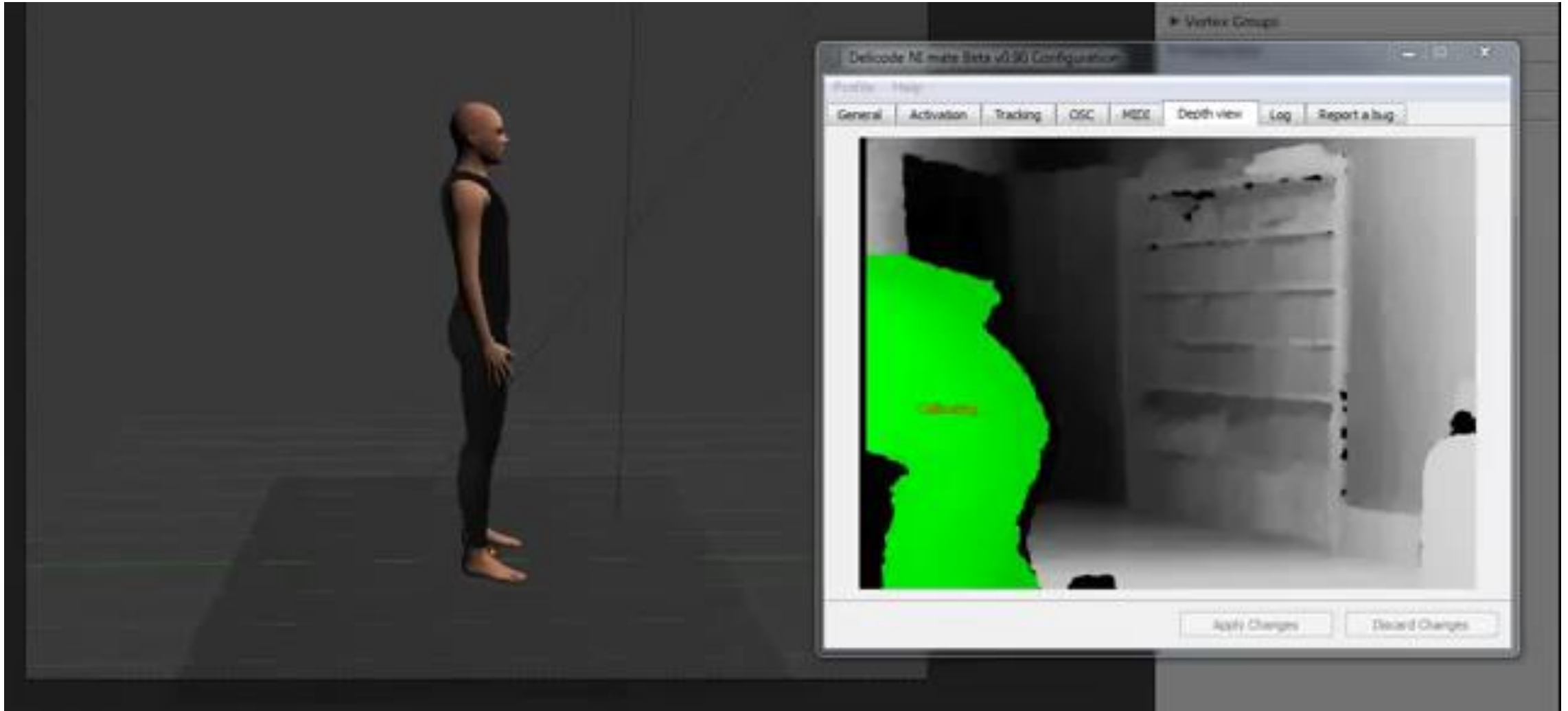
from Wikipedia



<https://www.youtube.com/watch?v=c6X64LhcUyQ>



# 安価なデプスカメラによるモーキャプ



<https://www.youtube.com/watch?v=qC-fdgPJhQ8>

# 屋外で使えるモーキャプ

## Motion Capture from Body-Mounted Cameras



(with audio)

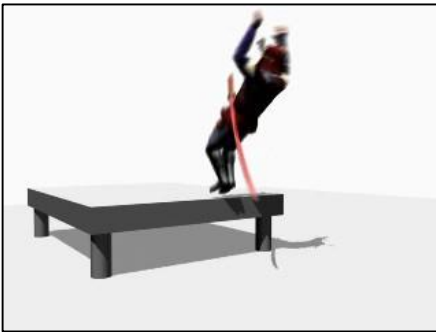
Takaaki Shiratori , Hyun Soo Park , Leonid Sigal ,  
Yaser Sheikh , Jessica K. Hodgins \*

\* Disney Research, Pittsburgh + Carnegie Mellon University

<https://www.youtube.com/watch?v=xbl-NWMfGPs>

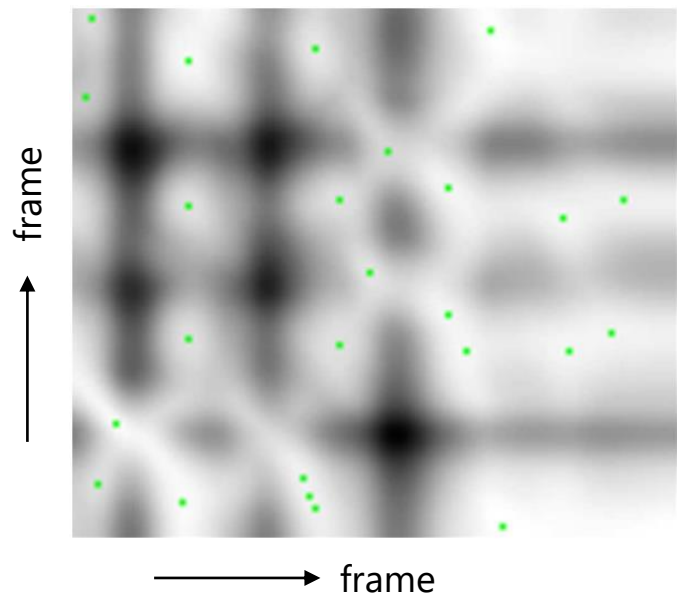
# モーションデータベース

- <http://mocap.cs.cmu.edu/>
- 6 カテゴリ、合計 2605個
- 研究促進のために無償公開 (補間、連結、解析、検索、etc)

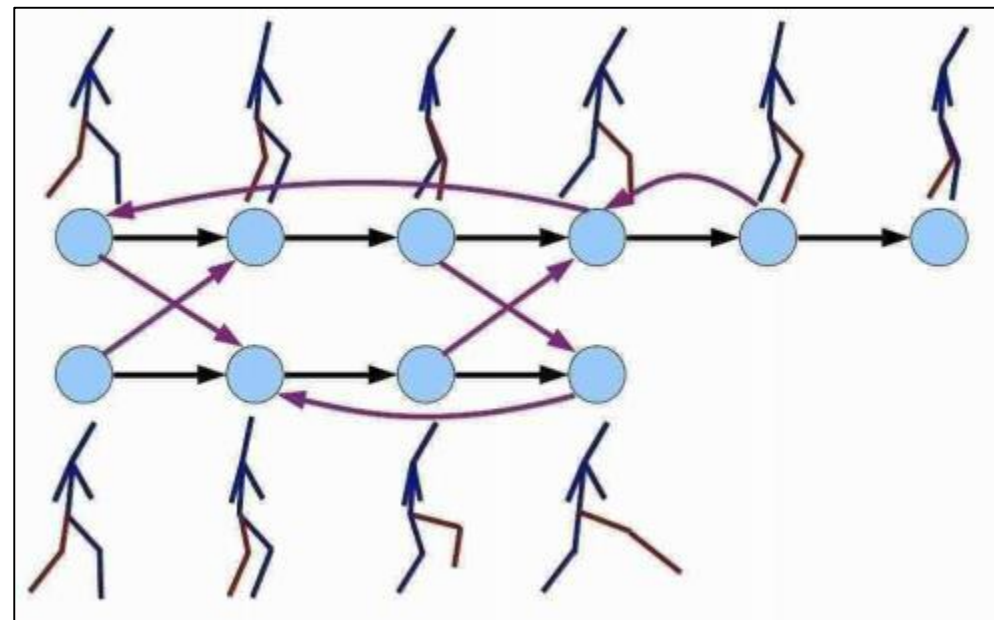
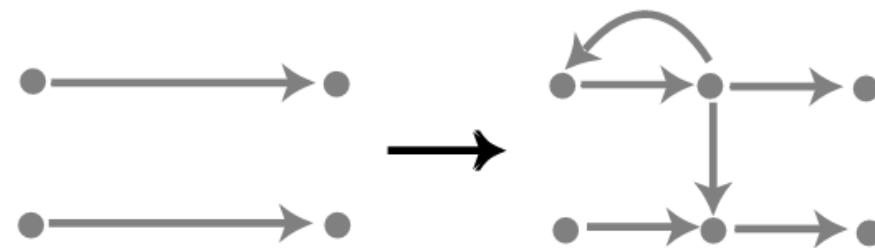


# モーションの連結

- 二つのフレームで姿勢が似ていれば、遷移を許す



フレーム間の姿勢の類似度



Motion Graphs [Kovar SIGGRAPH02]

Motion Patches: Building Blocks for Virtual Environments Annotated with Motion Data [Lee SIGGRAPH06]

<https://graphics.cs.wisc.edu/Papers/2002/KGP02/mograph.pdf>

# シミュレーションによるモーション生成

- モーキャプできない対象に使える
- 体型に合った自然な動作を生成できる
- 動的に変化する環境に適應できる

## **Flexible Muscle-Based Locomotion for Bipedal Creatures**

SIGGRAPH ASIA 2013

**Thomas Geijtenbeek  
Michiel van de Panne  
Frank van der Stappen**

<https://www.youtube.com/watch?v=pgaEE27nsQw>

# 専用デバイスによるポーズ作成

## Tangible and Modular Input Device for Character Articulation

Alec Jacobson<sup>1</sup>

Daniele Panozzo<sup>1</sup>

Oliver Glauser<sup>1</sup>

Cédric Pradalier<sup>2</sup>

Otmar Hilliges<sup>1</sup>

Olga Sorkine-Hornung<sup>1</sup>

<sup>1</sup>ETH Zurich

<sup>2</sup>GeorgiaTech Lorraine



*This video contains narration*

# キャラクターの動きに関する様々なトピック

Keyframe animation  
by  
topology coordinates

複数キャラクターの  
インタラクション

[https://www.youtube.com/  
watch?v=1S\\_6wSKI\\_nU](https://www.youtube.com/watch?v=1S_6wSKI_nU)

Synthesis of Detailed Hand Manipulations  
Using Contact Sampling

Yuting Ye C. Karen Liu  
Georgia Institute of Technology

物体をつかむ動作

[https://www.youtube.com/  
watch?v=x8c27XYTLTo](https://www.youtube.com/watch?v=x8c27XYTLTo)

Aggregate Dynamics  
for Dense Crowd Simulation

Submission 0042

群衆シミュレーション

[https://www.youtube.com/  
watch?v=pqBSNAOsMDc](https://www.youtube.com/watch?v=pqBSNAOsMDc)

Cost Functions

Path planning

<https://vimeo.com/33409868>

Character motion synthesis by topology coordinates [Ho EG09]

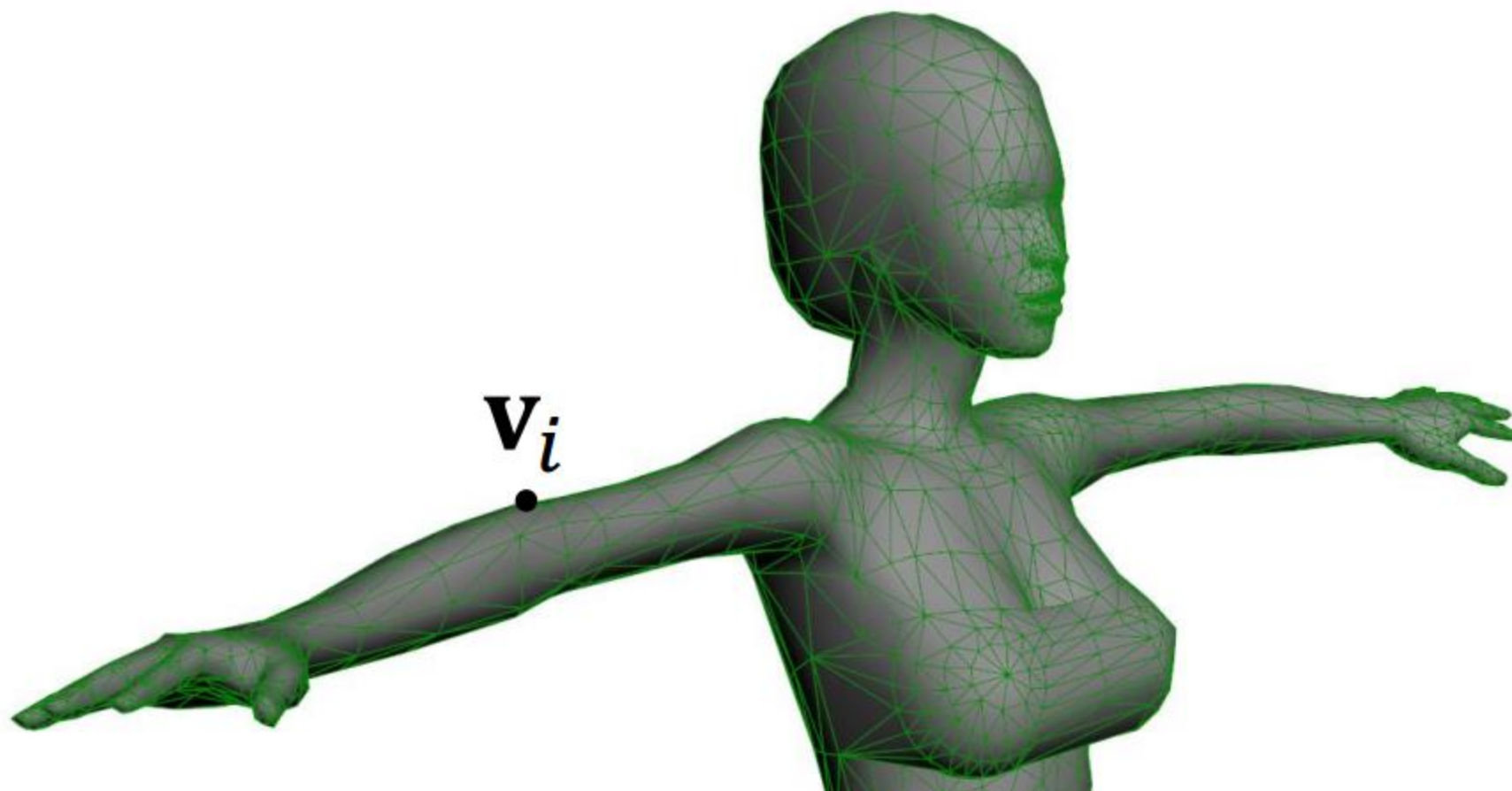
Aggregate Dynamics for Dense Crowd Simulation [Narain SIGGRAPHAsia09]

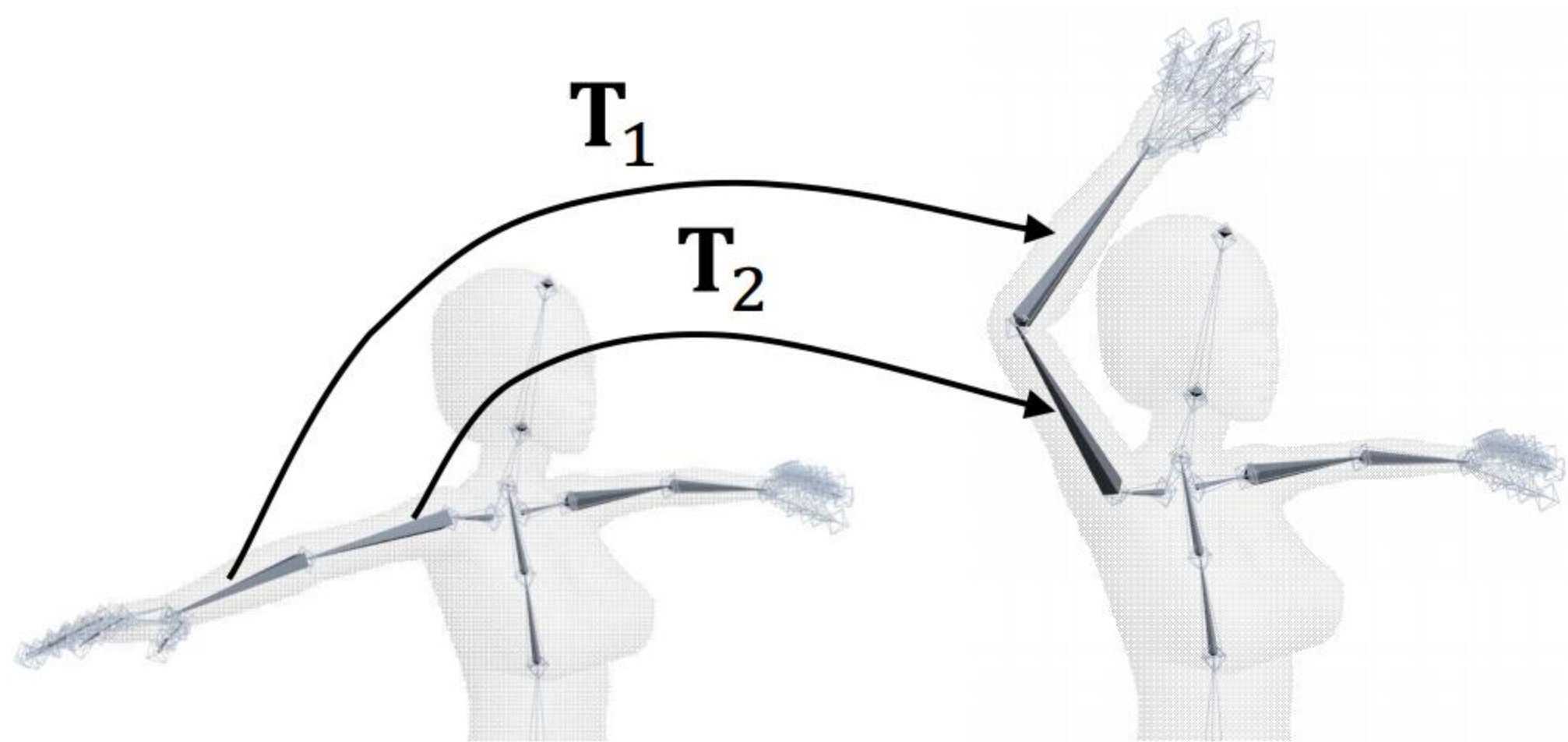
Synthesis of Detailed Hand Manipulations Using Contact Sampling [Ye SIGGRAPH12]

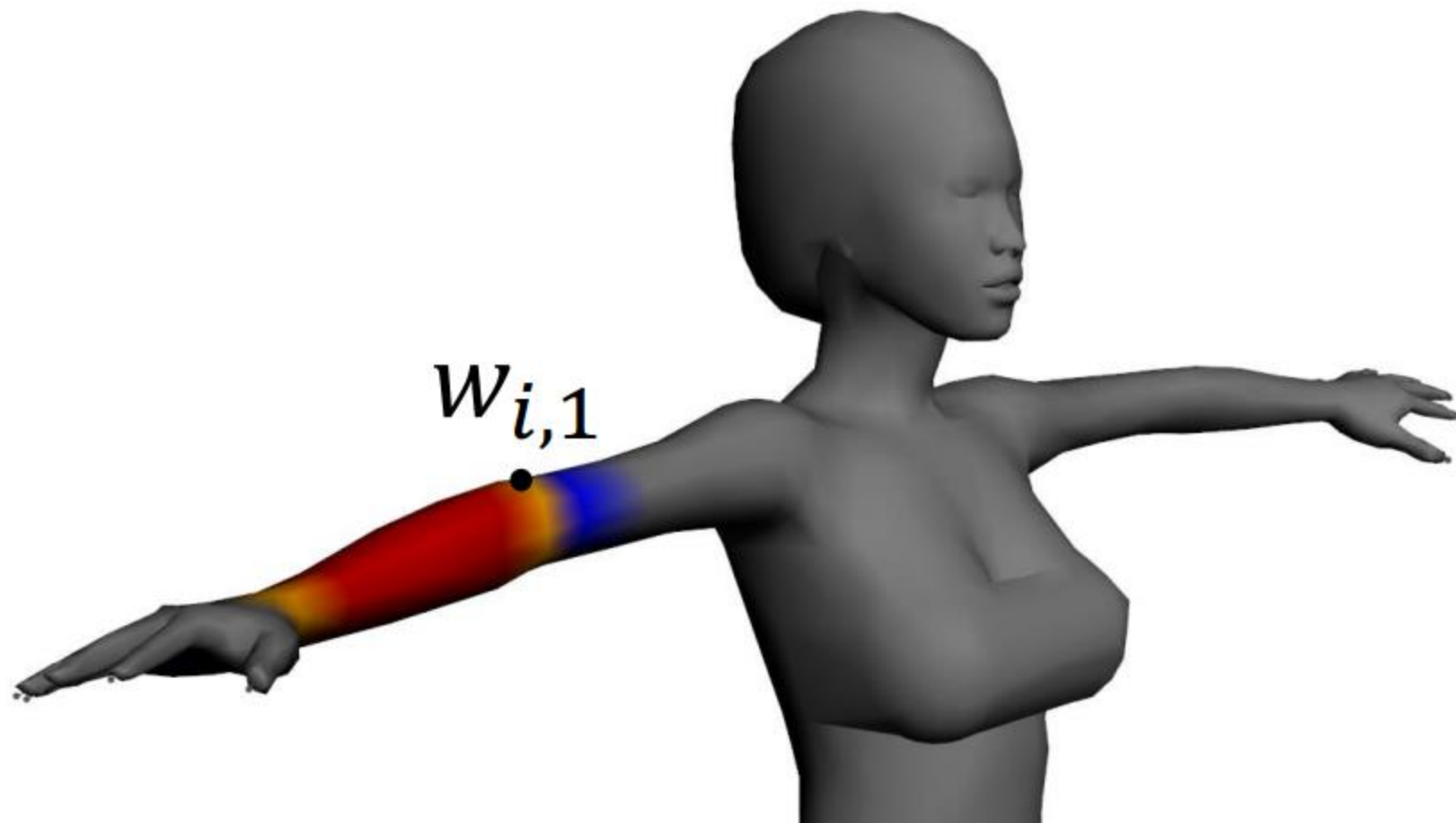
Space-Time Planning with Parameterized Locomotion Controllers.[Levine TOG11]

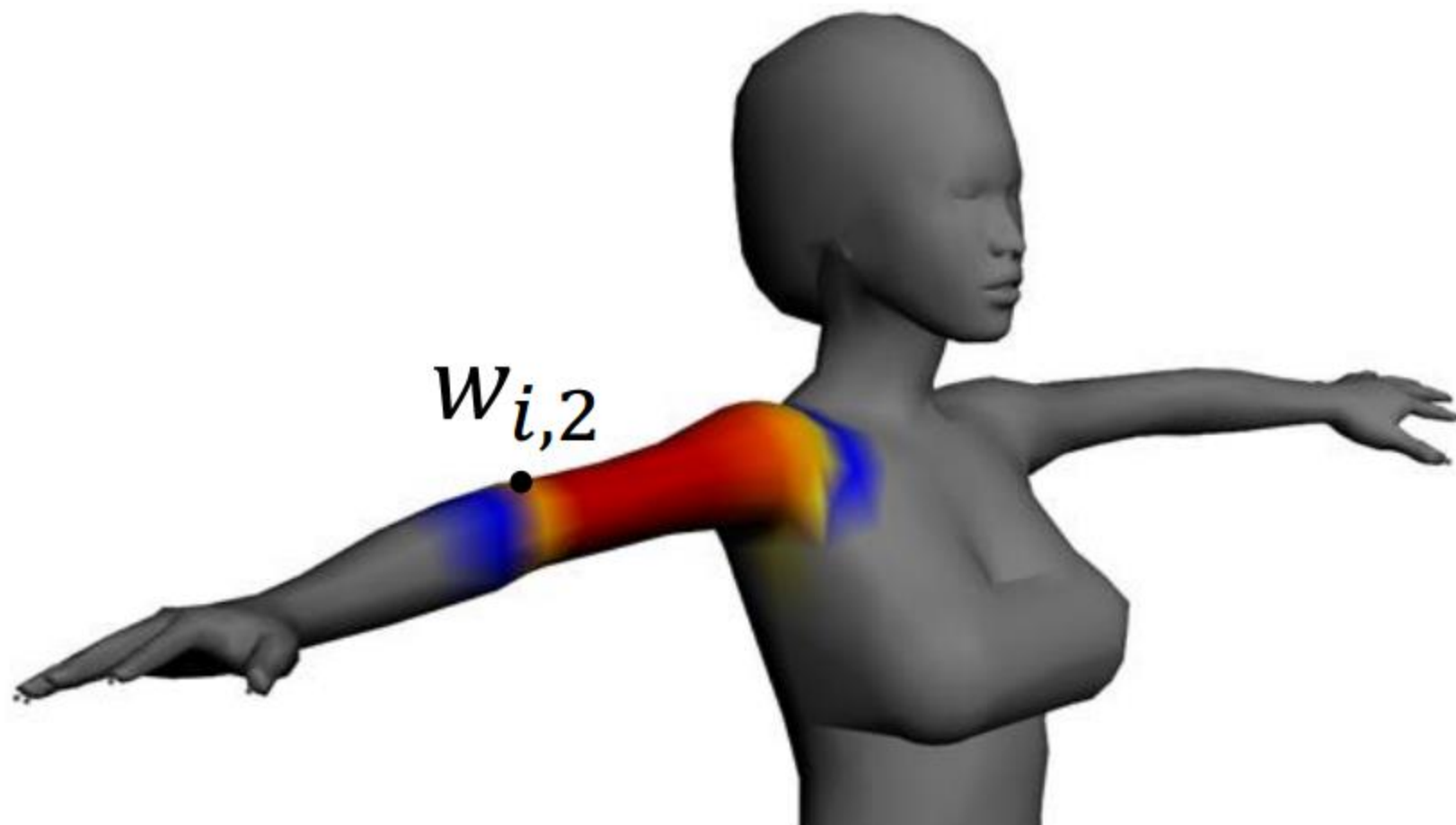
スキニング

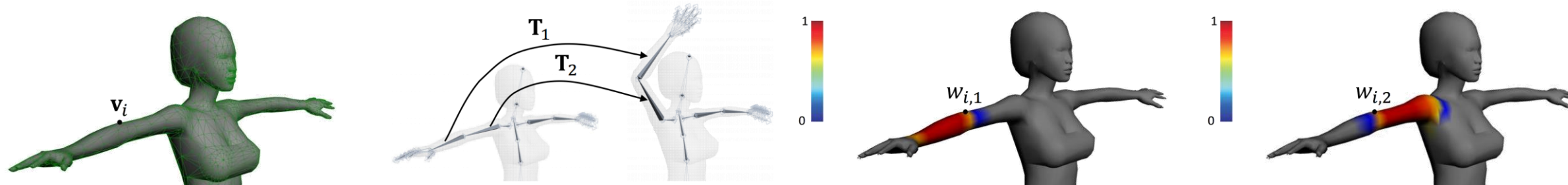








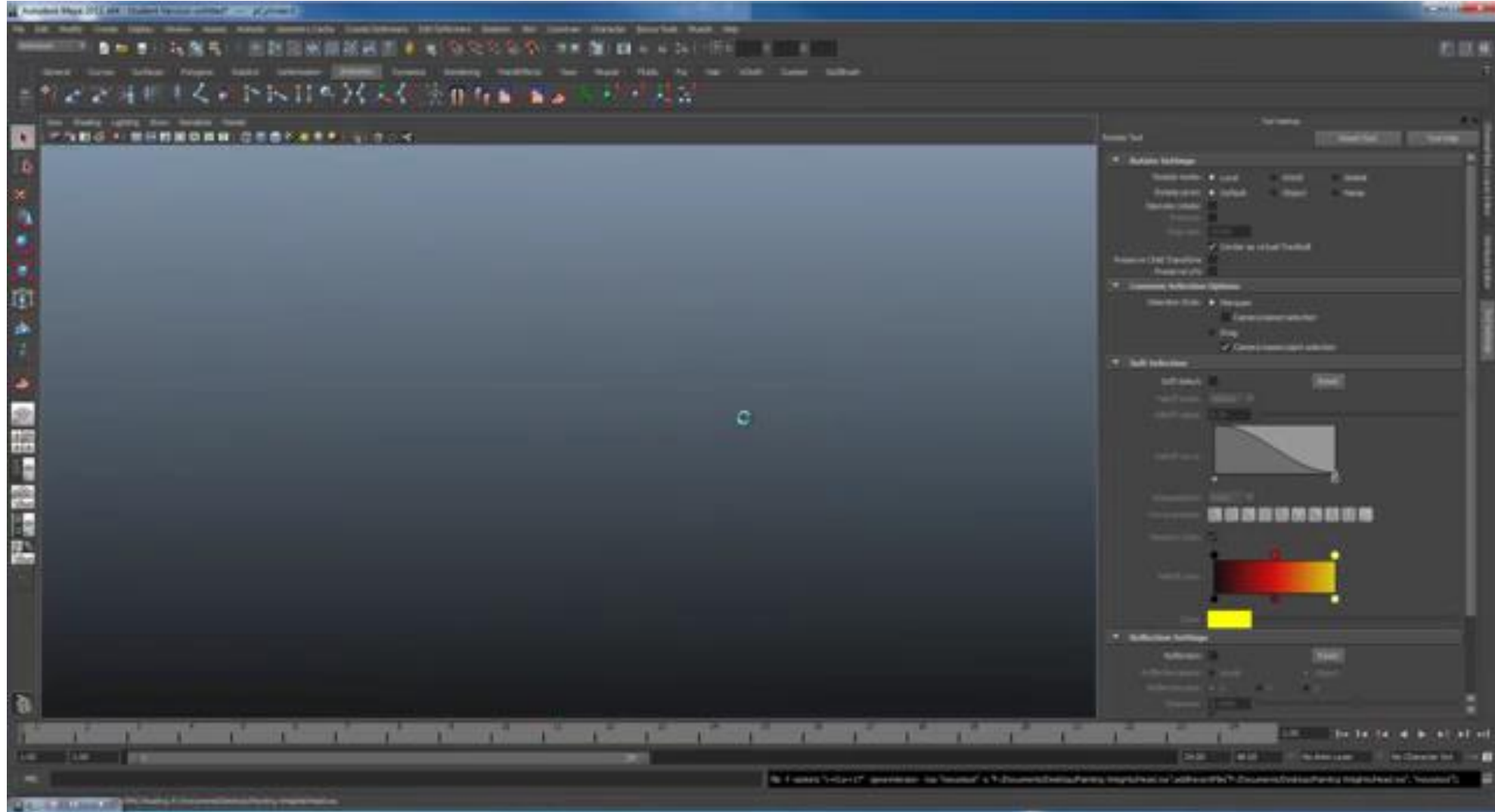




$$\mathbf{v}'_i = \text{blend}(\langle w_{i,1}, \mathbf{T}_1 \rangle, \langle w_{i,2}, \mathbf{T}_2 \rangle, \dots)(\mathbf{v}_i)$$

- 入力
  - メッシュ頂点座標  $\{\mathbf{v}_i\} \ i = 1, \dots, n$
  - ボーンの剛体変換  $\{\mathbf{T}_j\} \ j = 1, \dots, m$
  - 各ボーンから各メッシュ頂点への重み  $\{w_{i,j}\} \ i = 1, \dots, n \ j = 1, \dots, m$
- 出力
  - 変形後のメッシュ頂点座標  $\{\mathbf{v}'_i\} \ i = 1, \dots, n$
- 技術的なポイント
  - 重み  $\{w_{i,j}\}$  をどう与えるか
  - 変換をどうブレンドするか

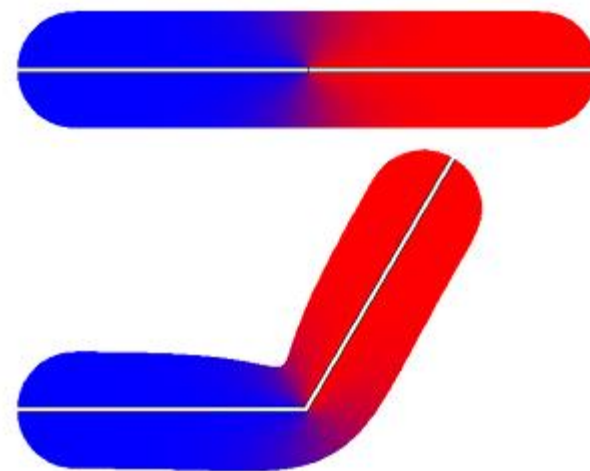
# 重みの与え方：手作業でペイント



# 重みの与え方：自動計算

- j 番目のボーンの重み  $w_j$  を、
  - j 番目のボーン上で 1 を取り、それ以外のボーン上で 0 を取り、
  - それ以外では滑らかなスカラー場として定式化

- 一階微分  $\int_{\Omega} \|\nabla w_j\|^2 dA$  を最小化 [Baran 07]
  - サーフェス上で近似的に解く → 簡単、高速
- 二階微分  $\int_{\Omega} (\Delta w_j)^2 dA$  を最小化 [Jacobson 11]
  - 不等式制約  $0 \leq w_j \leq 1$  も導入
  - ボリューム上で二次計画問題を解く → 高品質



Pinocchio デモ

# 変換の混合手法：Linear Blend Skinning

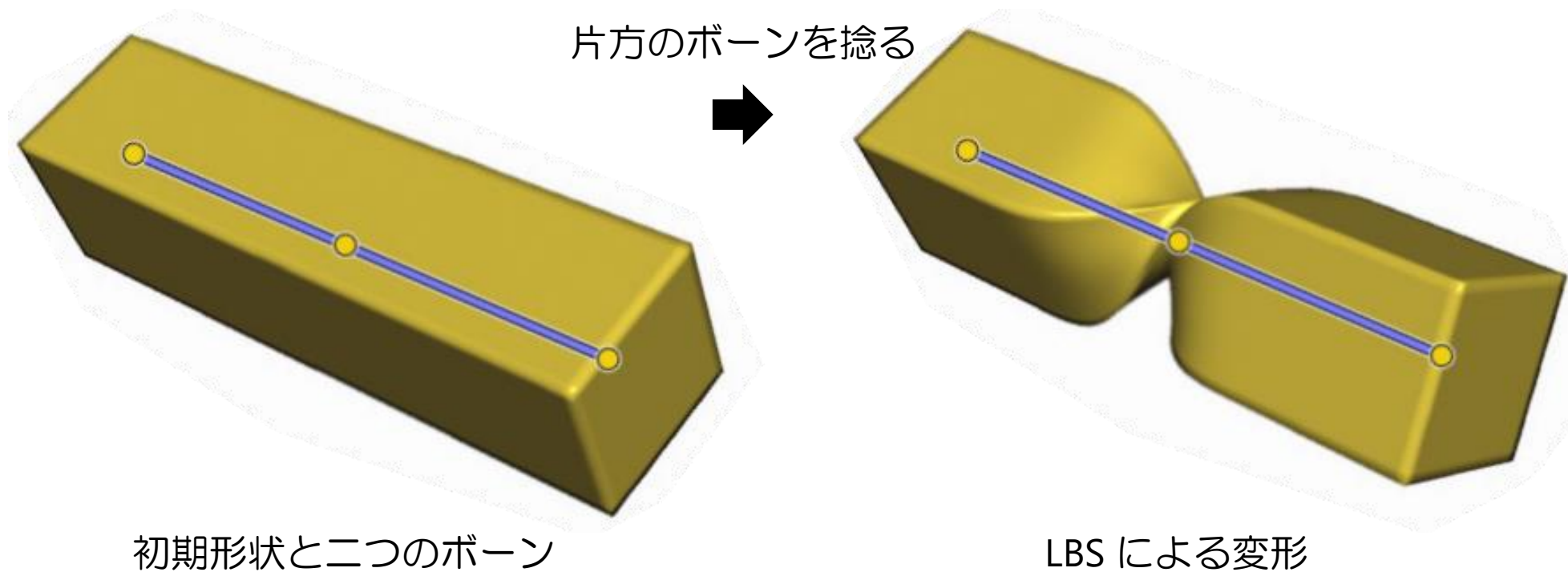
- 剛体変換  $\mathbf{T}_j$  は、回転行列  $\mathbf{R}_j \in \mathbb{R}^{3 \times 3}$  と移動ベクトル  $\mathbf{t}_j \in \mathbb{R}^3$  を並べた  $3 \times 4$  行列として表される

$$\mathbf{v}'_i = \left( \sum_j w_{i,j} (\mathbf{R}_j \quad \mathbf{t}_j) \right) \begin{pmatrix} \mathbf{v}_i \\ 1 \end{pmatrix}$$

- 単純で高速
  - 頂点シェーダで実装：フレーム毎に  $\{\mathbf{v}'_i\}$  を GPU に送るのではなく、初期化時に  $\{\mathbf{v}_i\}$  と  $\{w_{i,j}\}$  を送り、フレーム毎に  $\{\mathbf{T}_j\}$  を送る
- 業界で最も一般的

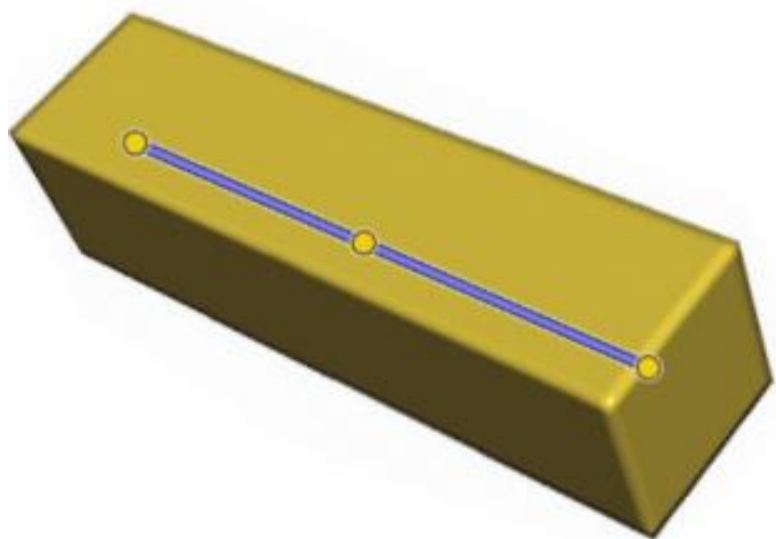


# LBS の欠陥：“candy wrapper” effect

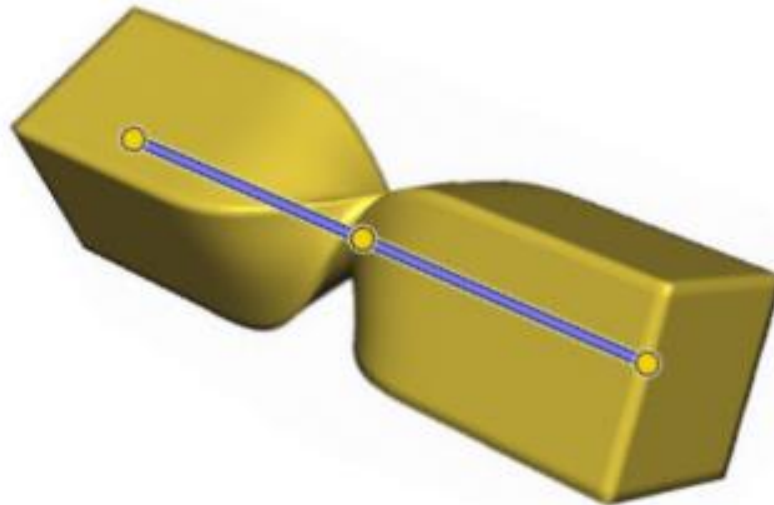


- 剛体変換の線形和は剛体変換にならない！
  - 180度捻ると関節の周りが一点に凝縮

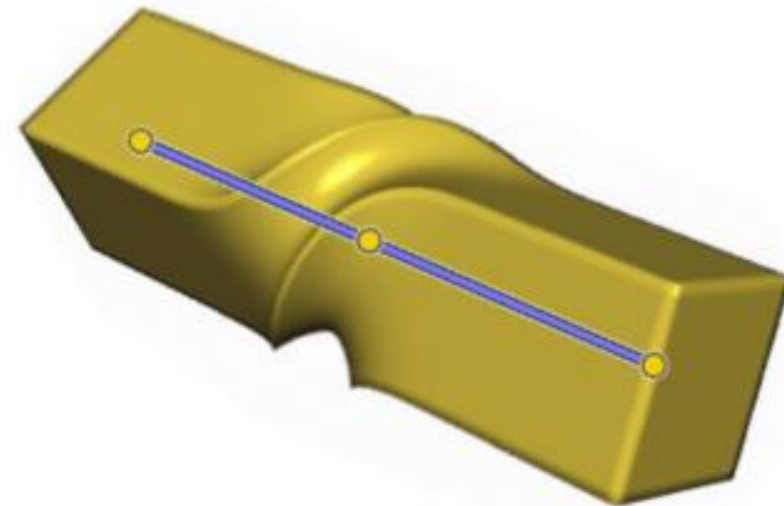
# LBS に代わる手法：Dual Quaternion Skinning



初期形状



LBS による変形



DQS による変形

- アイディア

- Quaternion (四つの実数)  $\rightarrow$  3D 回転変換
- Dual quaternion (二つの quaternion)  $\rightarrow$  3D 剛体変換 (回転 + 移動)

# Dual number と dual quaternion

- Dual number

- $\varepsilon^2 = 0$  という演算規則を持つ dual 単位  $\varepsilon$  を導入 (cf. 虚数単位  $i$ )
- Primal 成分と dual 成分 の和として dual number を定義：  $\hat{a} := a_0 + \varepsilon a_\varepsilon$   
 $a_0, a_\varepsilon \in \mathbb{R}$
- Dual 共役：  $\bar{\hat{a}} = \overline{a_0 + \varepsilon a_\varepsilon} = a_0 - \varepsilon a_\varepsilon$

- Dual quaternion

- Quaternion の 各成分が dual number であるようなもの
- 二つの quaternion を使って書ける  $\hat{\mathbf{q}} := \mathbf{q}_0 + \varepsilon \mathbf{q}_\varepsilon$
- Dual 共役：  $\bar{\hat{\mathbf{q}}} = \overline{\mathbf{q}_0 + \varepsilon \mathbf{q}_\varepsilon} = \mathbf{q}_0 - \varepsilon \mathbf{q}_\varepsilon$
- Quaternion 共役：  $\hat{\mathbf{q}}^* = (\mathbf{q}_0 + \varepsilon \mathbf{q}_\varepsilon)^* = \mathbf{q}_0^* + \varepsilon \mathbf{q}_\varepsilon^*$

# Dual quaternion による剛体変換

- 平行移動成分が  $\vec{t} = (t_x, t_y, t_z)$  で、回転成分が  $\mathbf{q}_0$  (単位quaternion) であるような剛体変換を表す単位 dual quaternion :

$$\hat{\mathbf{q}} = \mathbf{q}_0 + \frac{\varepsilon}{2} \vec{t} \mathbf{q}_0$$

注意：3Dベクトルは、実数成分を持たないquaternionと見なす

- 単位 dual quaternion  $\hat{\mathbf{q}}$  による、3D座標  $\vec{v} = (v_x, v_y, v_z)$  の剛体変換 :

$$\hat{\mathbf{q}}(1 + \varepsilon \vec{v})\overline{\hat{\mathbf{q}}}^* = 1 + \varepsilon \vec{v}'$$

- $\vec{v}'$  が変換後の3D座標

# Dual quaternion による剛体変換

- $\hat{\mathbf{q}} = \mathbf{q}_0 + \frac{\varepsilon}{2} \vec{\mathbf{t}} \mathbf{q}_0$

- $\hat{\mathbf{q}}(1 + \varepsilon \vec{\mathbf{v}}) \overline{\hat{\mathbf{q}}^*} = \left( \mathbf{q}_0 + \frac{\varepsilon}{2} \vec{\mathbf{t}} \mathbf{q}_0 \right) (1 + \varepsilon \vec{\mathbf{v}}) \overline{\left( \mathbf{q}_0 + \frac{\varepsilon}{2} \vec{\mathbf{t}} \mathbf{q}_0 \right)^*}$

$$= \left( \mathbf{q}_0 + \frac{\varepsilon}{2} \vec{\mathbf{t}} \mathbf{q}_0 \right) (1 + \varepsilon \vec{\mathbf{v}}) \overline{\left( \mathbf{q}_0^* + \frac{\varepsilon}{2} (\vec{\mathbf{t}} \mathbf{q}_0)^* \right)}$$

$$\begin{aligned} (\vec{\mathbf{t}} \mathbf{q}_0)^* &= \mathbf{q}_0^* (\vec{\mathbf{t}})^* \\ &= -\mathbf{q}_0^* \vec{\mathbf{t}} \end{aligned}$$

$$= \left( \mathbf{q}_0 + \frac{\varepsilon}{2} \vec{\mathbf{t}} \mathbf{q}_0 \right) (1 + \varepsilon \vec{\mathbf{v}}) \overline{\left( \mathbf{q}_0^* - \frac{\varepsilon}{2} \mathbf{q}_0^* \vec{\mathbf{t}} \right)}$$

$$= \left( \mathbf{q}_0 + \frac{\varepsilon}{2} \vec{\mathbf{t}} \mathbf{q}_0 \right) (1 + \varepsilon \vec{\mathbf{v}}) \left( \mathbf{q}_0^* + \frac{\varepsilon}{2} \mathbf{q}_0^* \vec{\mathbf{t}} \right)$$

$$= \left( \mathbf{q}_0 + \frac{\varepsilon}{2} \vec{\mathbf{t}} \mathbf{q}_0 \right) \left( \mathbf{q}_0^* + \varepsilon \vec{\mathbf{v}} \mathbf{q}_0^* + \frac{\varepsilon}{2} \mathbf{q}_0^* \vec{\mathbf{t}} + \cancel{\frac{\varepsilon^2}{2} \vec{\mathbf{v}} \mathbf{q}_0^* \vec{\mathbf{t}}} \right)$$

$$= \mathbf{q}_0 \mathbf{q}_0^* + \frac{\varepsilon}{2} \vec{\mathbf{t}} \mathbf{q}_0 \mathbf{q}_0^* + \varepsilon \mathbf{q}_0 \vec{\mathbf{v}} \mathbf{q}_0^* + \cancel{\frac{\varepsilon^2}{2} \vec{\mathbf{t}} \mathbf{q}_0 \vec{\mathbf{v}} \mathbf{q}_0^*} + \frac{\varepsilon}{2} \mathbf{q}_0 \mathbf{q}_0^* \vec{\mathbf{t}} + \cancel{\frac{\varepsilon^2}{4} \vec{\mathbf{t}} \mathbf{q}_0 \mathbf{q}_0^* \vec{\mathbf{t}}}$$

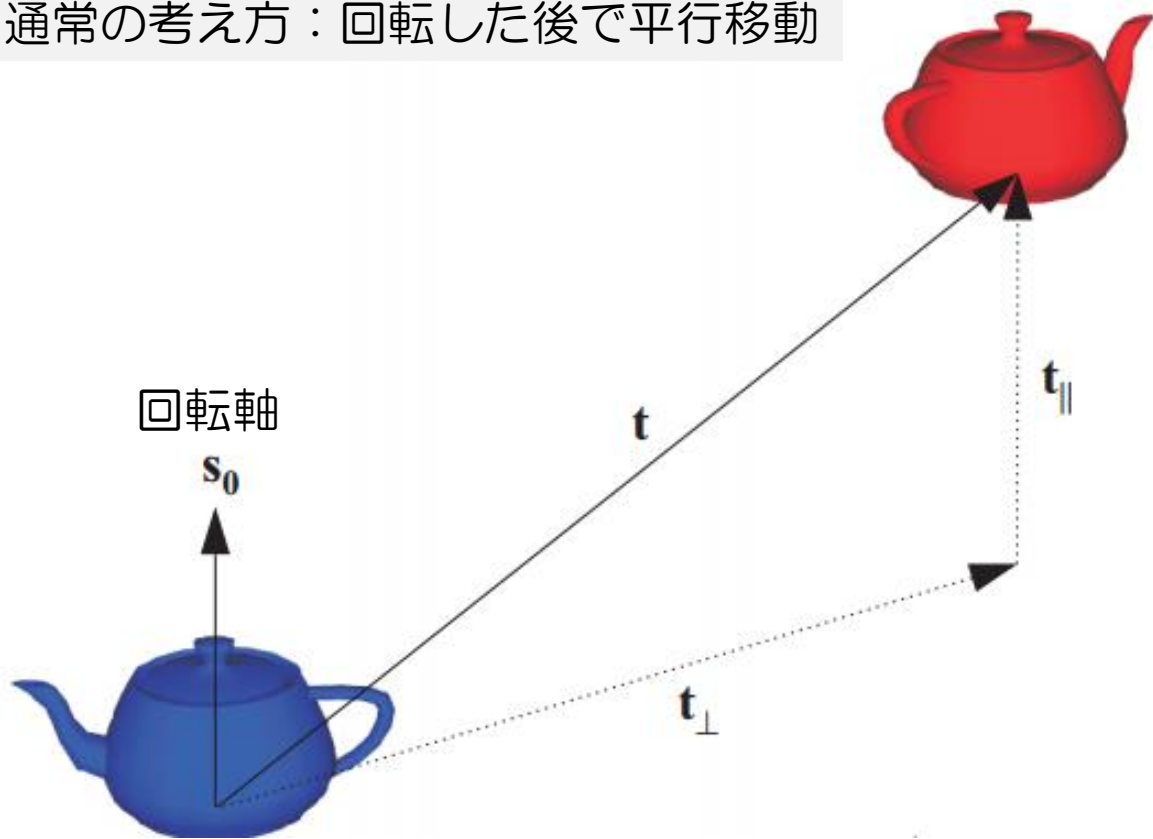
$$\|\mathbf{q}_0\|^2 = 1$$

$$= 1 + \varepsilon (\vec{\mathbf{t}} + \mathbf{q}_0 \vec{\mathbf{v}} \mathbf{q}_0^*)$$

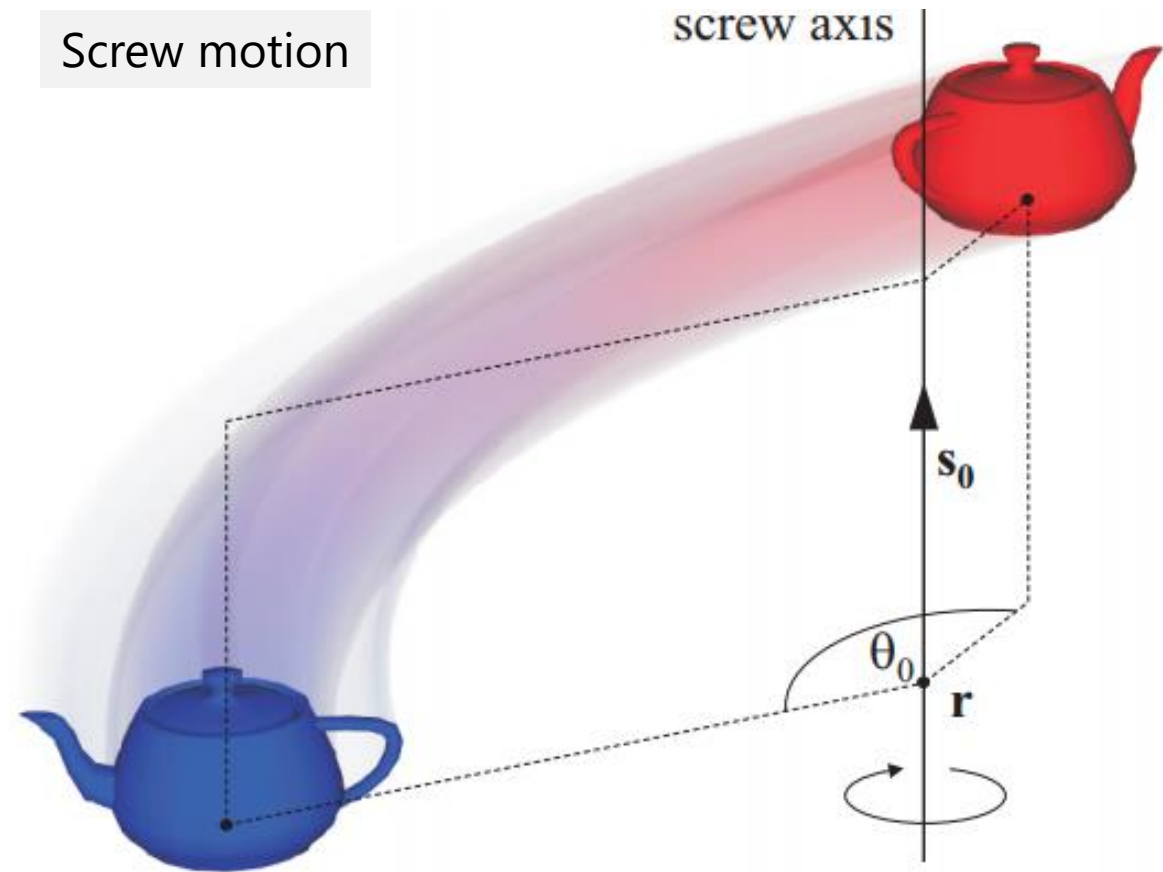
3D座標  $\vec{\mathbf{v}}$  をquaternion  $\mathbf{q}_0$  で回転した結果

# "Screw motion" としての剛体運動

通常の方：回転した後で平行移動



Screw motion



- 任意の剛体運動は、screw motion として一意に記述できる

# Screw motion と dual quaternion

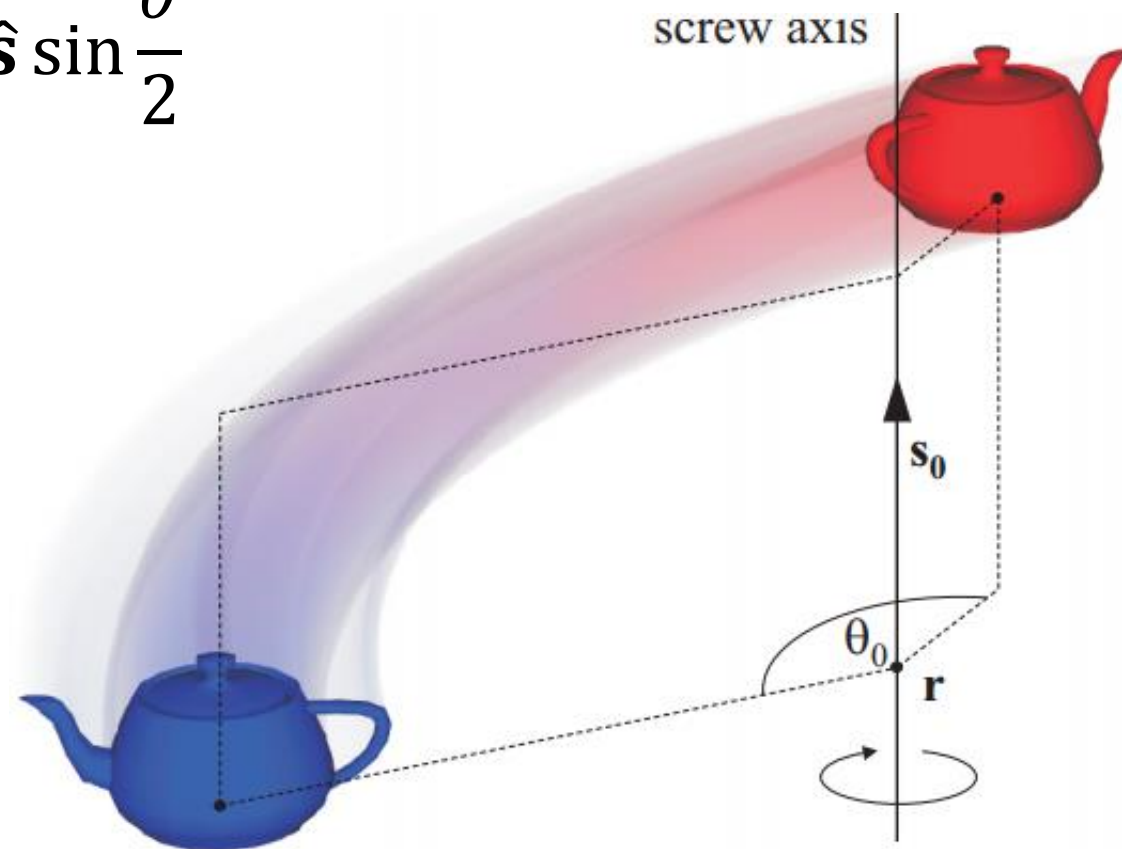
- 単位 dual quaternion  $\hat{\mathbf{q}}$  は、以下の形で表せる：

$$\hat{\mathbf{q}} = \cos \frac{\hat{\theta}}{2} + \hat{\mathbf{s}} \sin \frac{\hat{\theta}}{2}$$

- $\hat{\theta} = \theta_0 + \varepsilon \theta_\varepsilon$   $\theta_0, \theta_\varepsilon$  : 実数
- $\hat{\mathbf{s}} = \vec{\mathbf{s}}_0 + \varepsilon \vec{\mathbf{s}}_\varepsilon$   $\vec{\mathbf{s}}_0, \vec{\mathbf{s}}_\varepsilon$  : 単位3Dベクトル

- 幾何的な意味

- $\vec{\mathbf{s}}_0$  : 回転軸方向
- $\theta_0$  : 回転量
- $\theta_\varepsilon$  : 回転軸方向の平行移動量
- $\vec{\mathbf{s}}_\varepsilon$  : 回転軸が  $\vec{\mathbf{r}}$  を通るとき、  
 $\vec{\mathbf{s}}_\varepsilon = \vec{\mathbf{r}} \times \vec{\mathbf{s}}_0$  を満たす



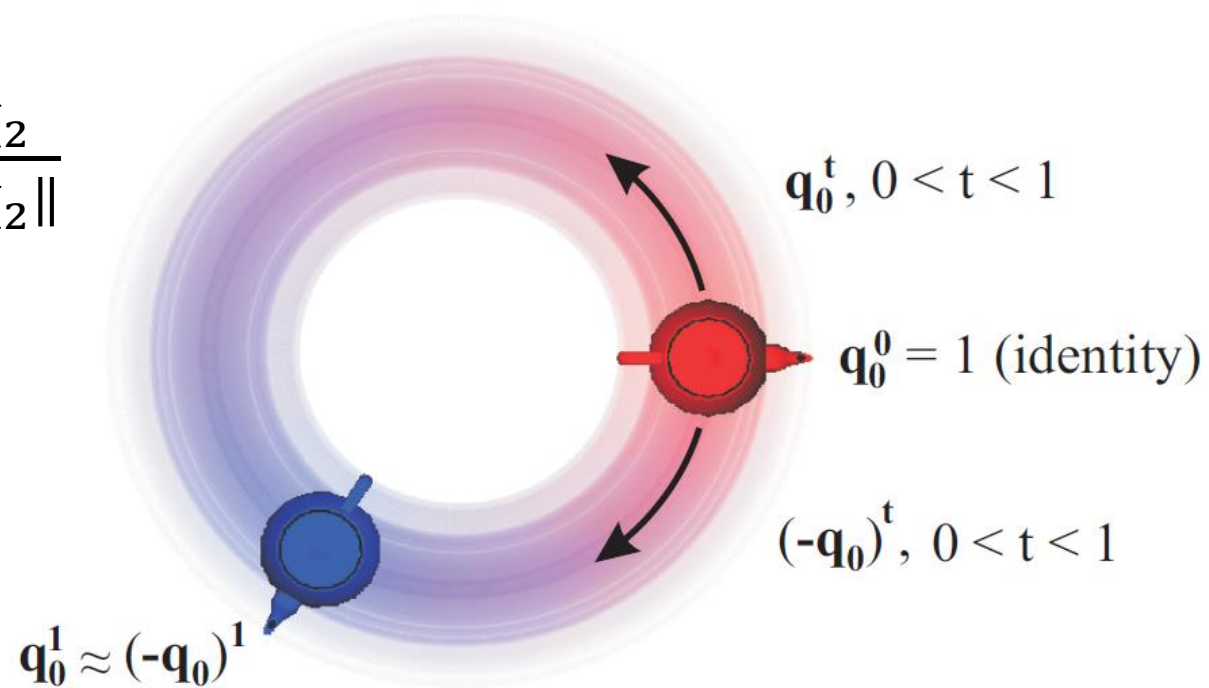
# 二つの剛体変換の補間

- 線形補間＋正規化 (nlerp)

$$\text{nlerp}(\hat{\mathbf{q}}_1, \hat{\mathbf{q}}_2, t) := \frac{(1-t)\hat{\mathbf{q}}_1 + t\hat{\mathbf{q}}_2}{\|(1-t)\hat{\mathbf{q}}_1 + t\hat{\mathbf{q}}_2\|}$$

- 注意： $\hat{\mathbf{q}}$ と $-\hat{\mathbf{q}}$ は同じ剛体変換を表すが、過程が正反対

- $\hat{\mathbf{q}}_1$  と  $\hat{\mathbf{q}}_2$  それぞれの non-dual な quaternion の 4D 内積が負であれば、 $\hat{\mathbf{q}}_1$  の補間相手を  $-\hat{\mathbf{q}}_2$  とする

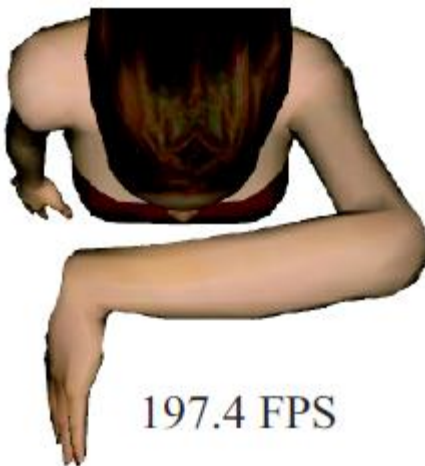




# Dual quaternion による剛体変換のブレンド

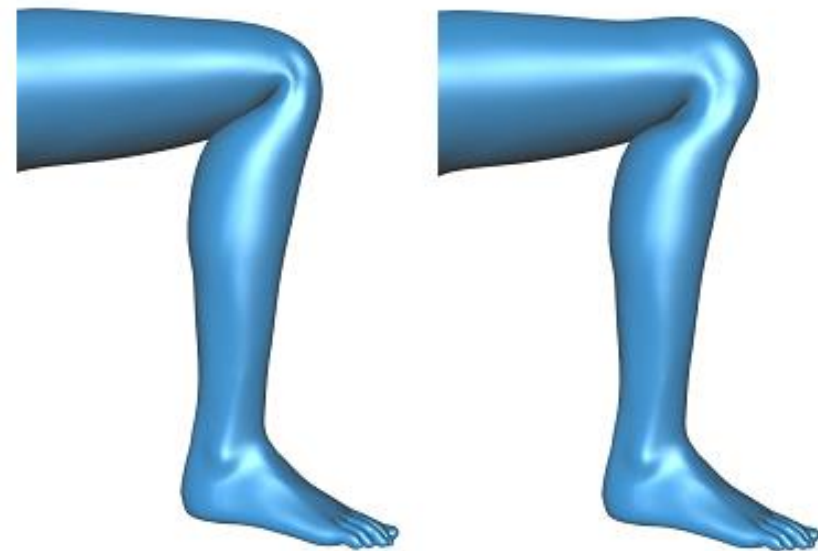
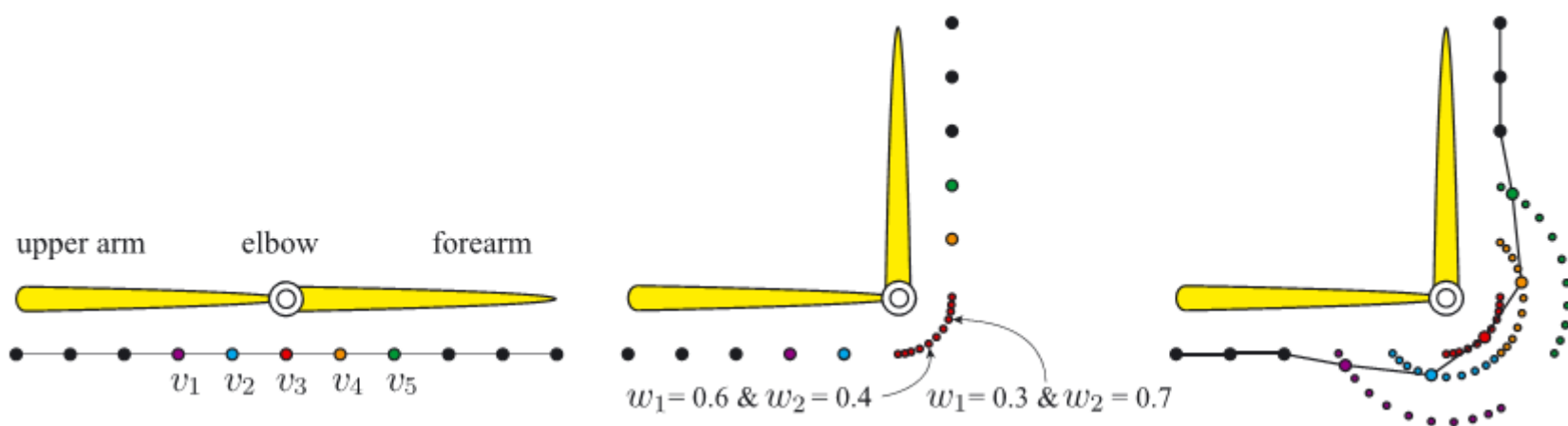
$$\text{blend}(\langle w_1, \hat{\mathbf{q}}_1 \rangle, \langle w_2, \hat{\mathbf{q}}_2 \rangle, \dots) := \frac{w_1 \hat{\mathbf{q}}_1 + w_2 \hat{\mathbf{q}}_2 + \dots}{\|w_1 \hat{\mathbf{q}}_1 + w_2 \hat{\mathbf{q}}_2 + \dots\|}$$

- Quaternion による回転と同様
- 入力データ形式が LBS と同一、計算コスト低い
- 市販CGソフトの多くに標準装備



# DQS の欠点：“bulging” effect

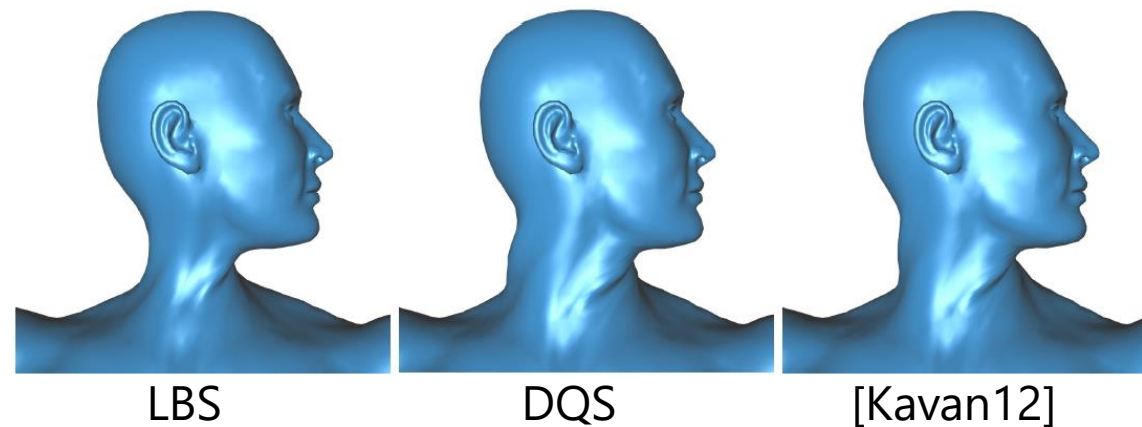
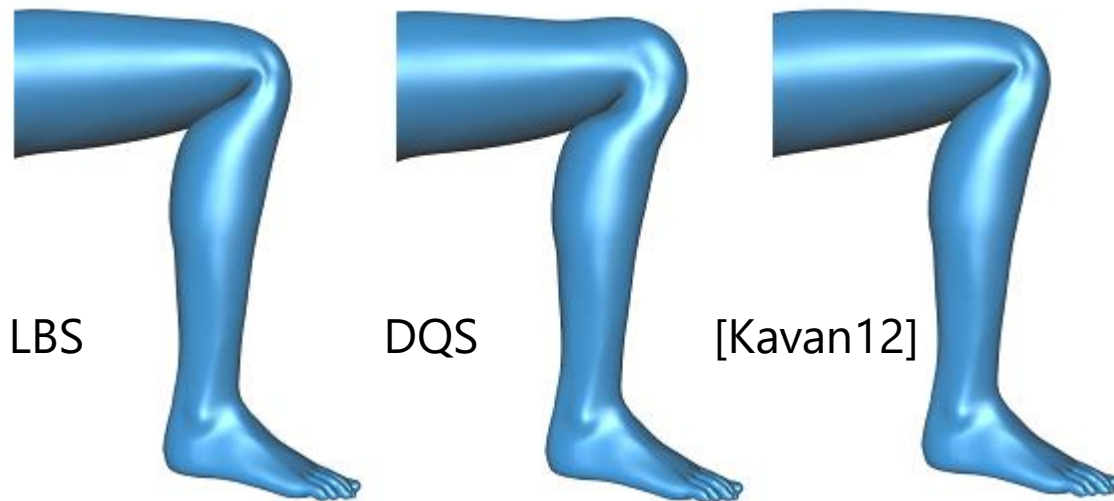
- 曲げの際に、関節を中心とした球面上に沿ったような軌跡を描く



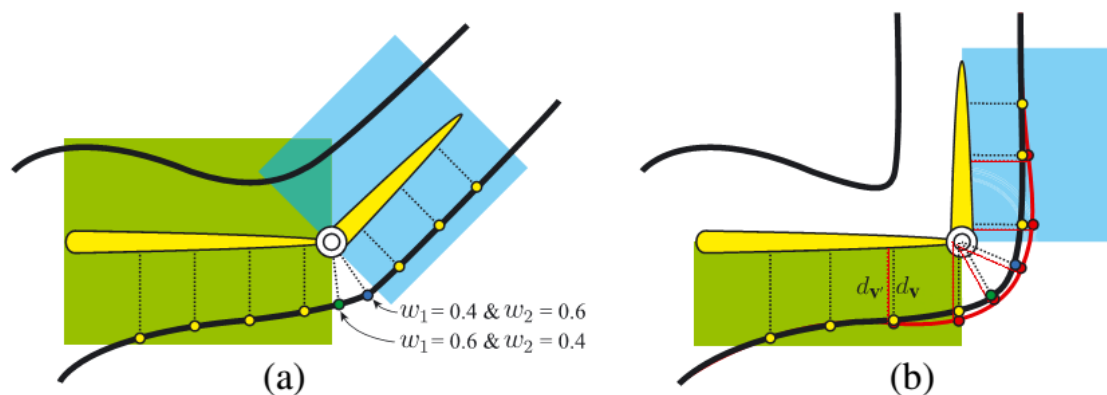
LBS

DQS

# DQS の欠点の克服

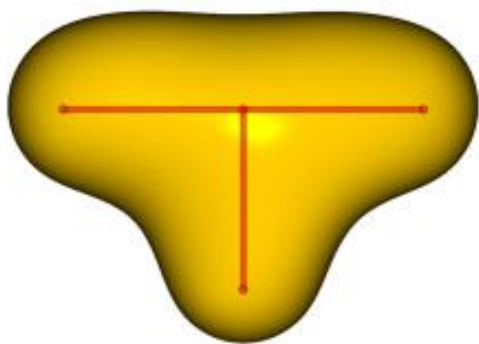


変換を bend と twist に分解し、別々に補間 [Kavan12]

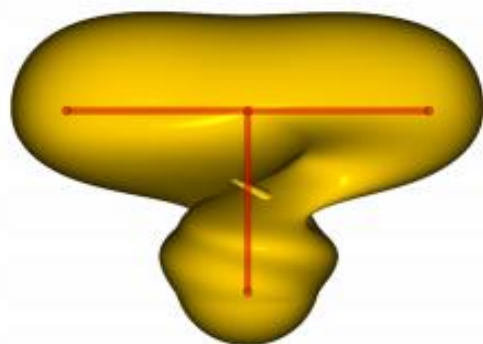


DQS で動かした後、法線方向にオフセット [Kim14]

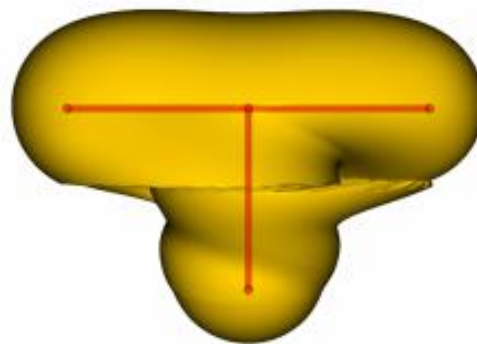
# DQS の欠点：捻りの回転量の制限



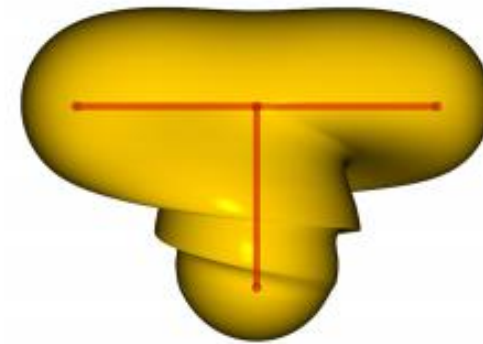
Rest pose



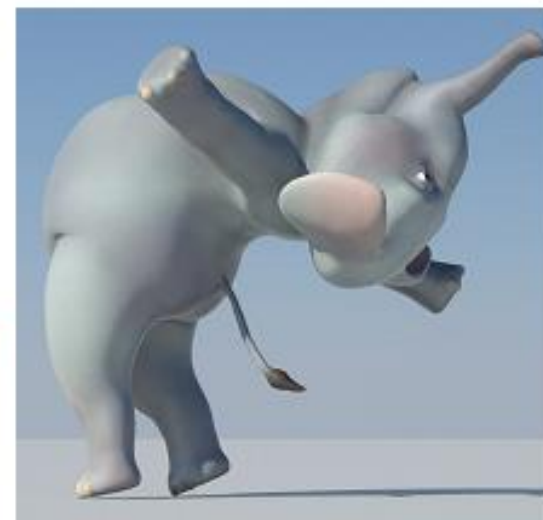
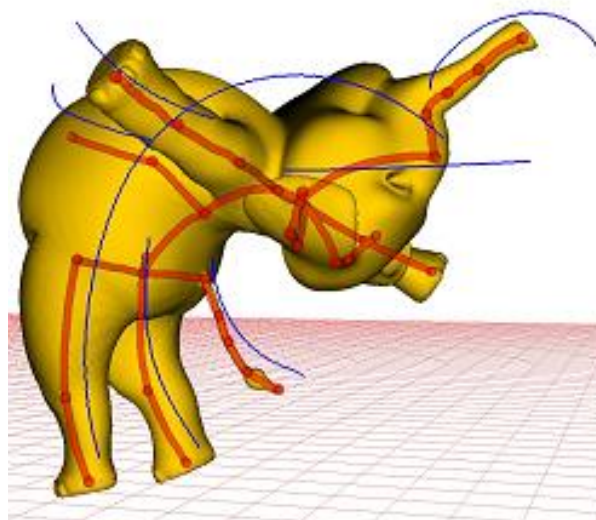
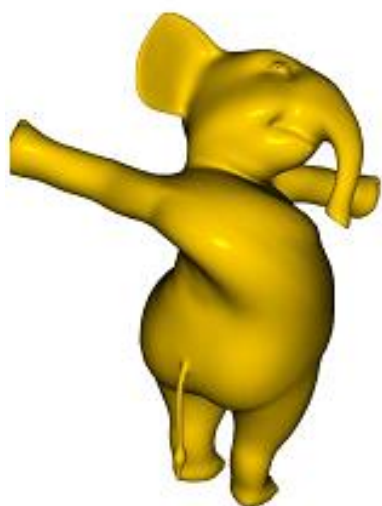
Linear blending



Dual quaternion blending



Differential blending



# 自己交差を回避するスキニング

- 陰関数の性質を活用



<https://www.youtube.com/watch?v=RHYSGLqEgyk>



# スケルトン以外の変形インタフェース

点、ケージ、スケルトンの統合 [Jacobson 11]

## Bounded Biharmonic Weights for Real-Time Deformation

Alec Jacobson<sup>1</sup>

Ilya Baran<sup>2</sup>

Jovan Popović<sup>3</sup>

Olga Sorkine<sup>1,4</sup>

<sup>1</sup>New York University

<sup>2</sup>Disney Research, Zurich

<sup>3</sup>Adobe Systems, Inc.

<sup>4</sup>ETH Zurich

This video contains narration

<https://www.youtube.com/watch?v=P9fqm8vgdB8>

BlendShape



<https://www.youtube.com/watch?v=BFPAlU8hwQ4>

# 参考情報

- [http://en.wikipedia.org/wiki/Motion\\_capture](http://en.wikipedia.org/wiki/Motion_capture)
- <http://skinning.org/>
- <http://mukai-lab.org/category/library/legacy>
- CG Gems JP 2012 Chapter 8 インバーススキネマティクス