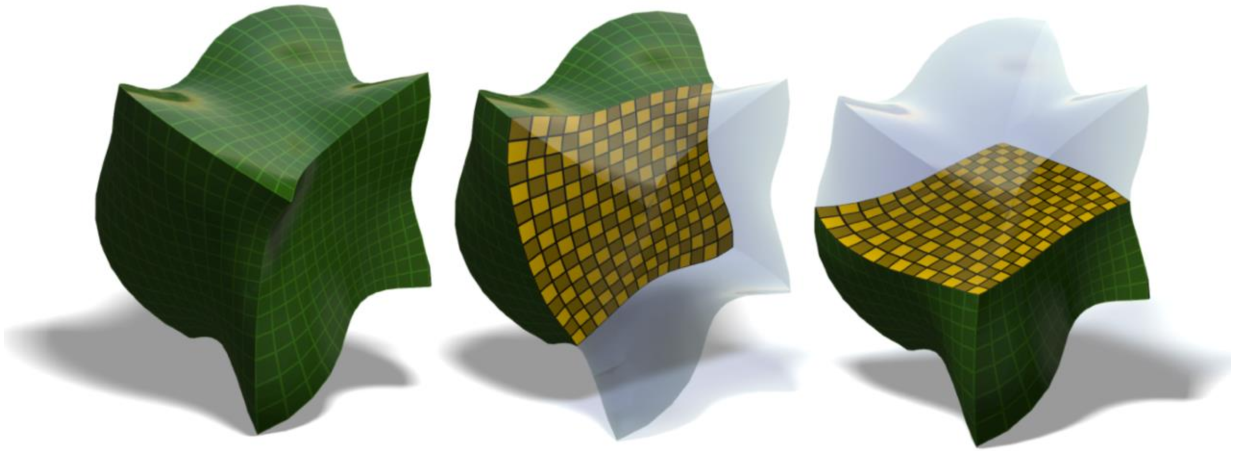
Space Transformations



Dr. Cengiz Öztireli

Computer Graphics Laboratory, ETH Zürich

Space Transformations

- Deform a space into another

$$\mathbf{d} : \mathbb{R}^d \rightarrow \mathbb{R}^d$$
$$\mathbf{x}' = \mathbf{x} + \mathbf{d}(\mathbf{x})$$

$$\mathbf{d}(\mathbf{x}) = \mathbf{R}\mathbf{x} + \mathbf{t} - \mathbf{x}$$

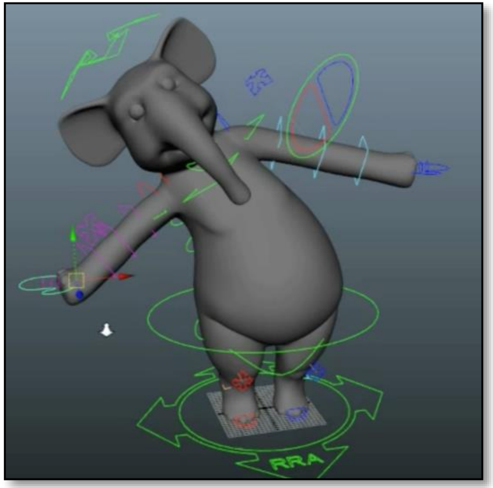
Space Transformations

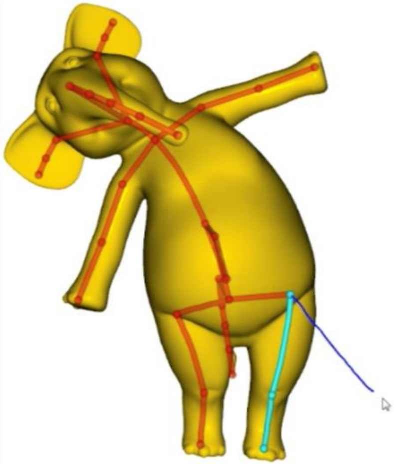
- Deform a space into another

$$\mathbf{d} : \mathbb{R}^d \rightarrow \mathbb{R}^d$$
$$\mathbf{x}' = \mathbf{x} + \mathbf{d}(\mathbf{x})$$

Space Transformations

- Application: character animation



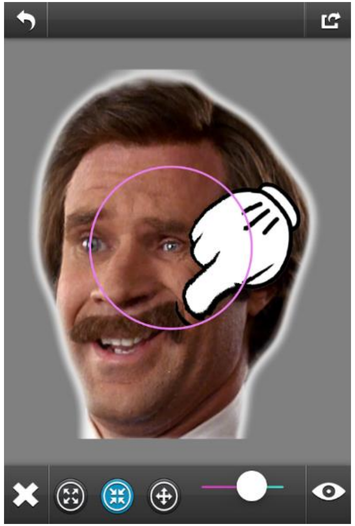


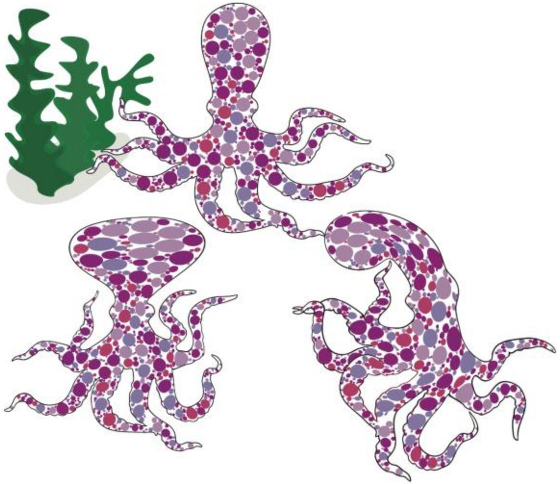
Space Transformations

- Application: motion capture & synthesis

Space Transformations

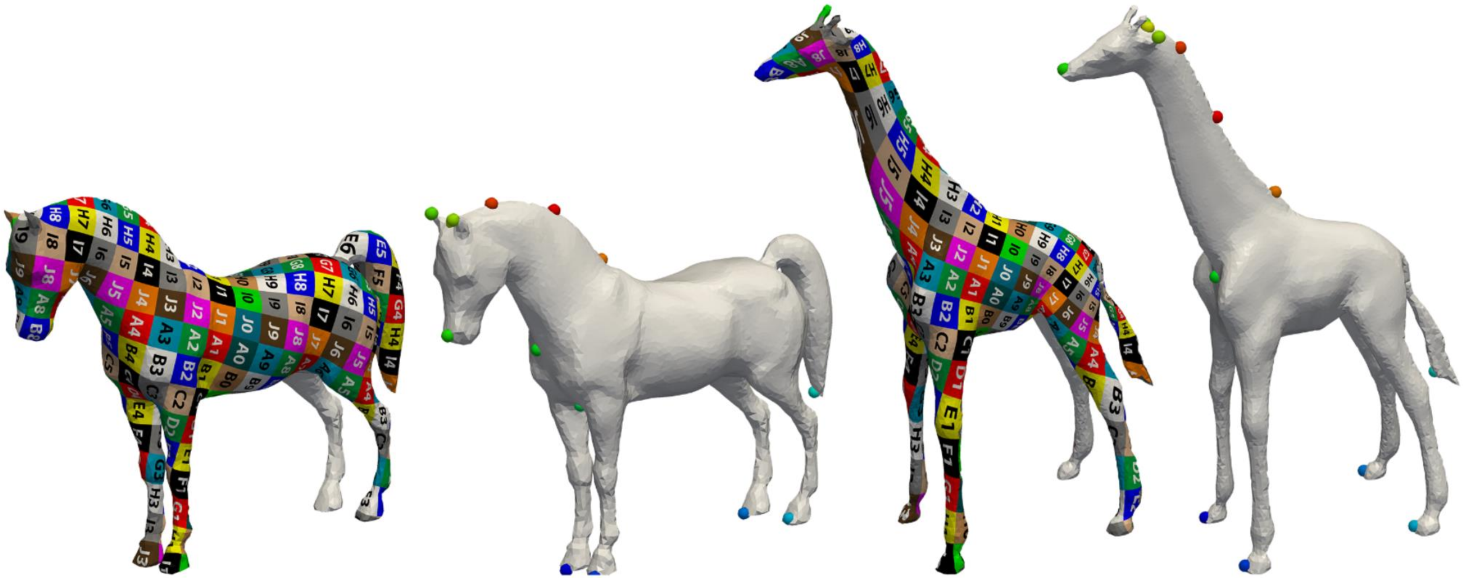
- Application: image warping





Space Transformations

- Application: maps on surfaces



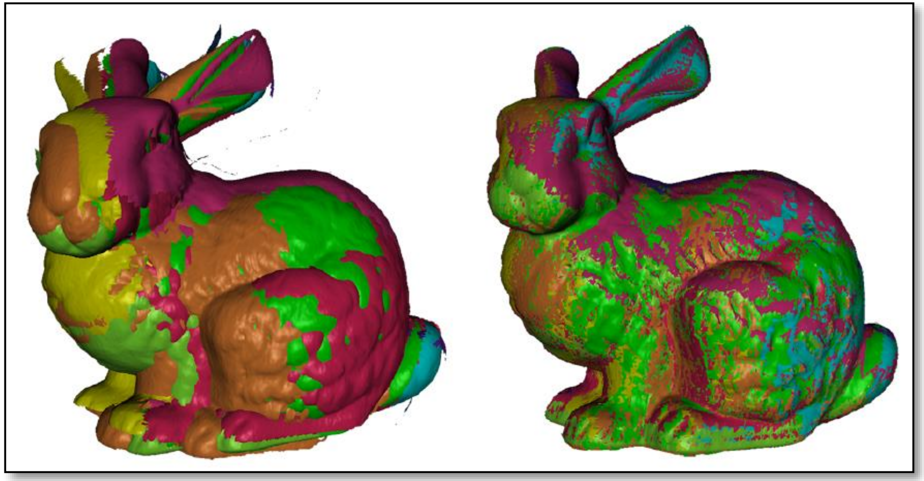
Space Transformations

- Application: camera & pose estimation

ETH zürich Disney Research cgl

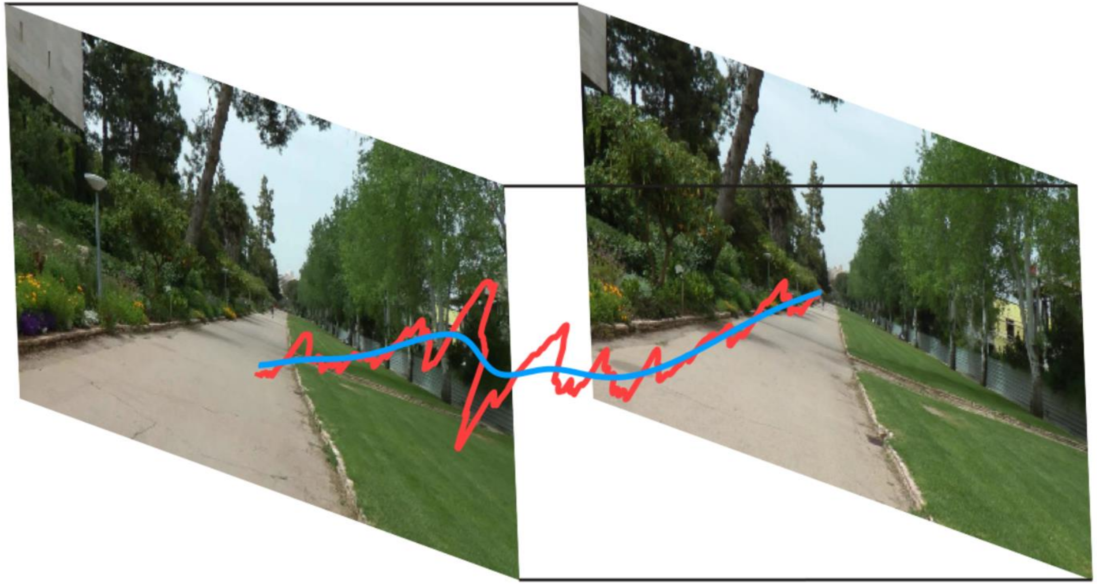
Space Transformations

- Application: surface registration
  - Align scans by rotating & translating



Space Transformations

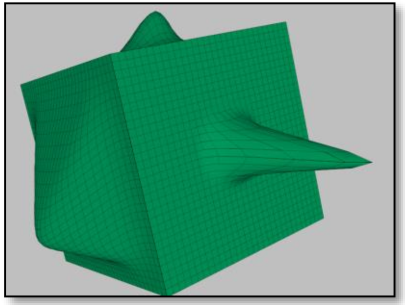
- Application: camera path interpolation



Space Transformations

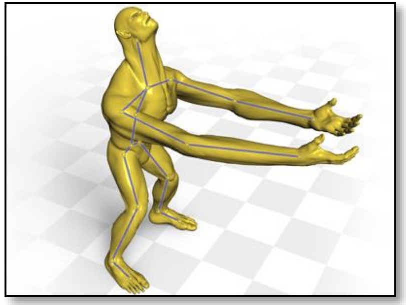
- Application: Deformations

Free-form

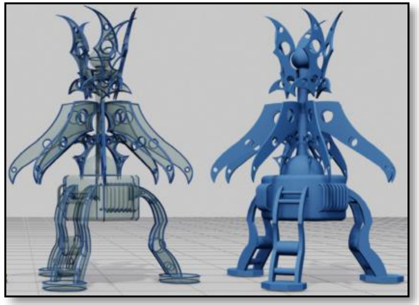


Elastic

Skeletal



Structure-aware



More structure

Space Transformations

- Today's plan
  - Blending rigid transformations
  - (Dual-) Quaternions for representation
  - Weights for blending
  - Alternative transformation methods

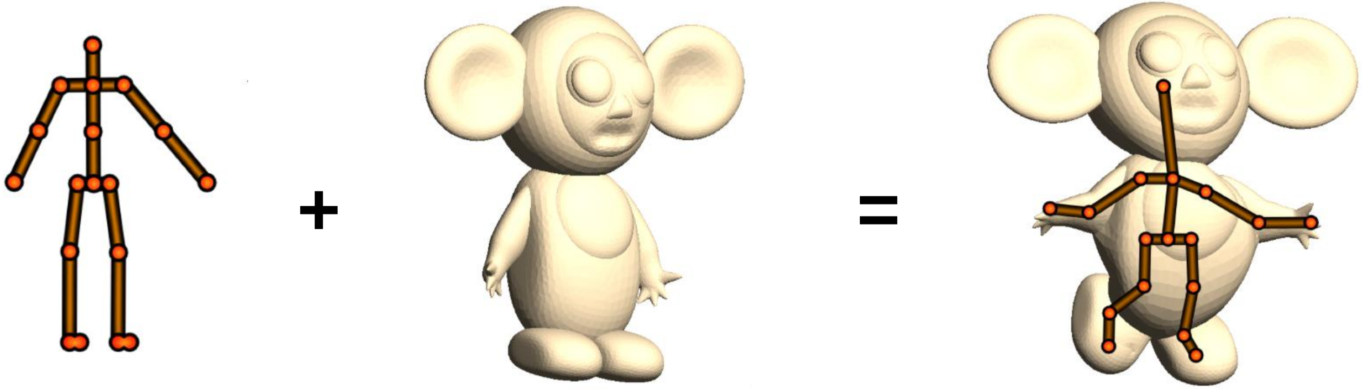
Space Transformations

- Rigid transformations
  - The most common form
  - More complex by blending

$$\mathbf{d} : \mathbb{R}^d \rightarrow \mathbb{R}^d$$
$$\mathbf{x}' = \mathbf{x} + \mathbf{d}(\mathbf{x})$$
$$\mathbf{d}(\mathbf{x}) = \mathbf{R}\mathbf{x} + \mathbf{t} - \mathbf{x}$$

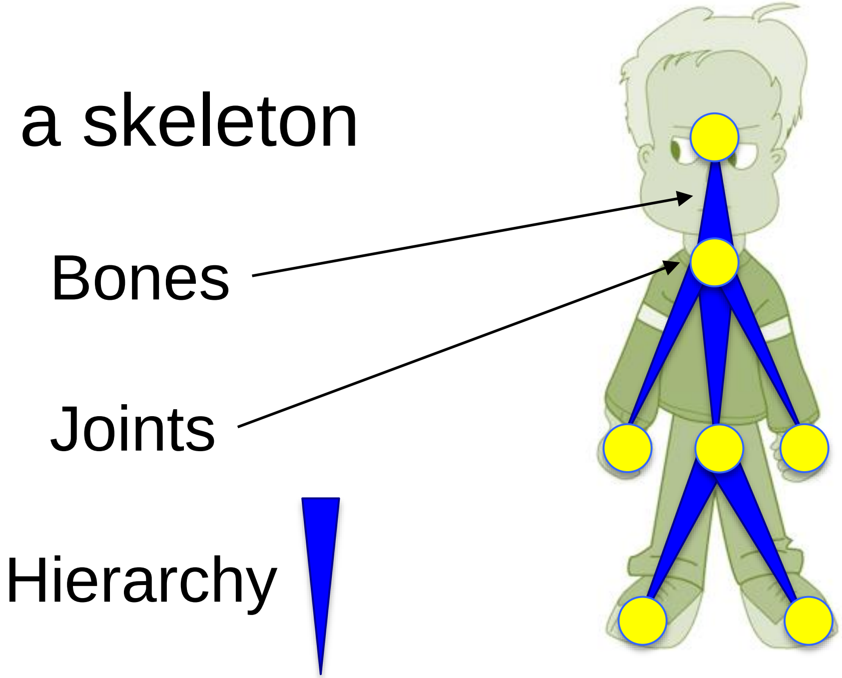
Blended Rigid Transformations

- Rigging
  - Attaching a skeleton to a model
  - Skeleton is key-framed to animate the model



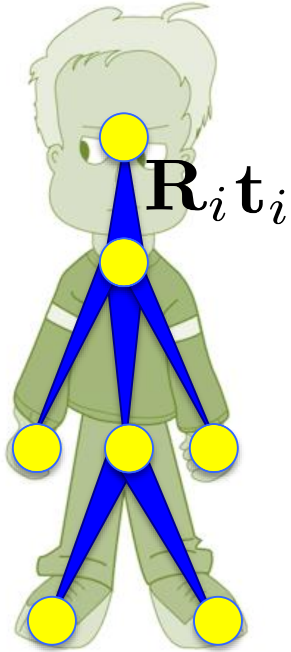
Blended Rigid Transformations

- Rigging
  - What is a skeleton



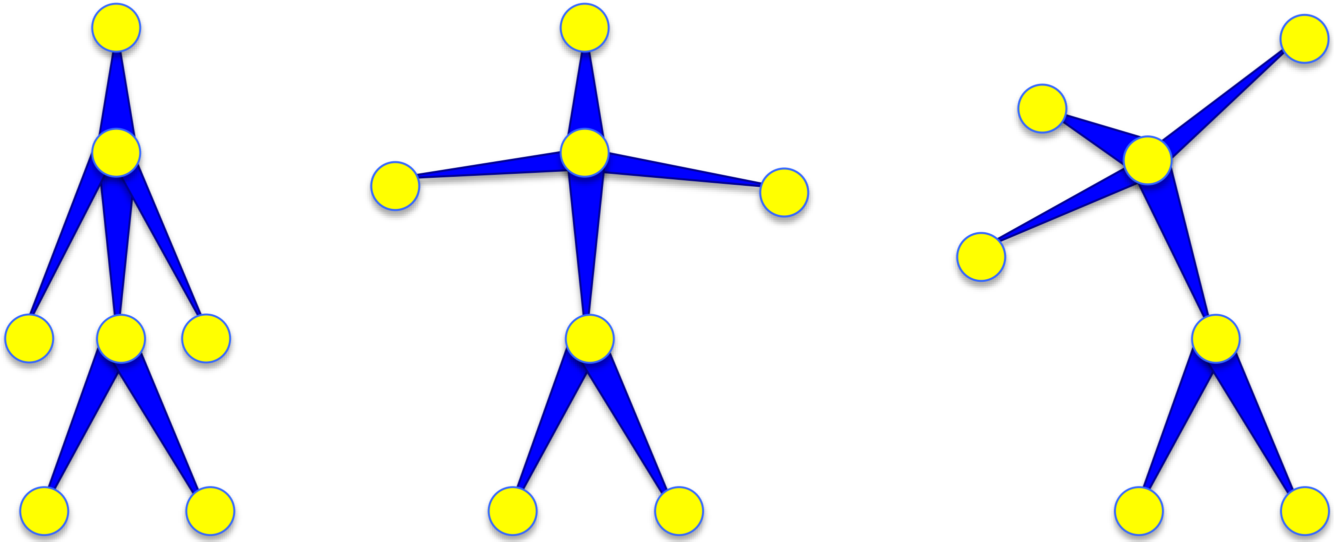
Blended Rigid Transformations

- Rigging
  - What is stored in a skeleton
    - Rigid transformations
    - On bones or joints
    - Bones can be transformed rigidly



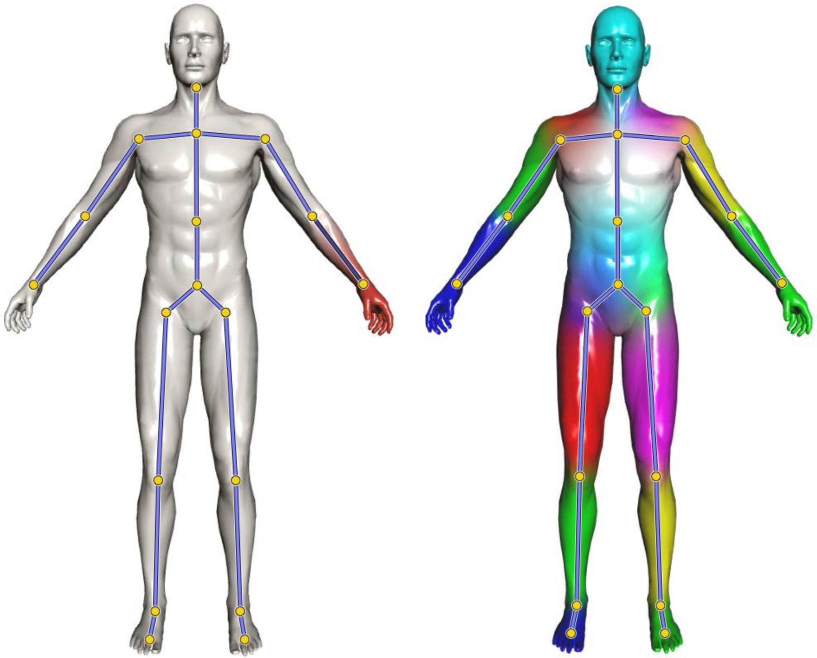
Blended Rigid Transformations

- Rigging
  - Bones can be transformed rigidly



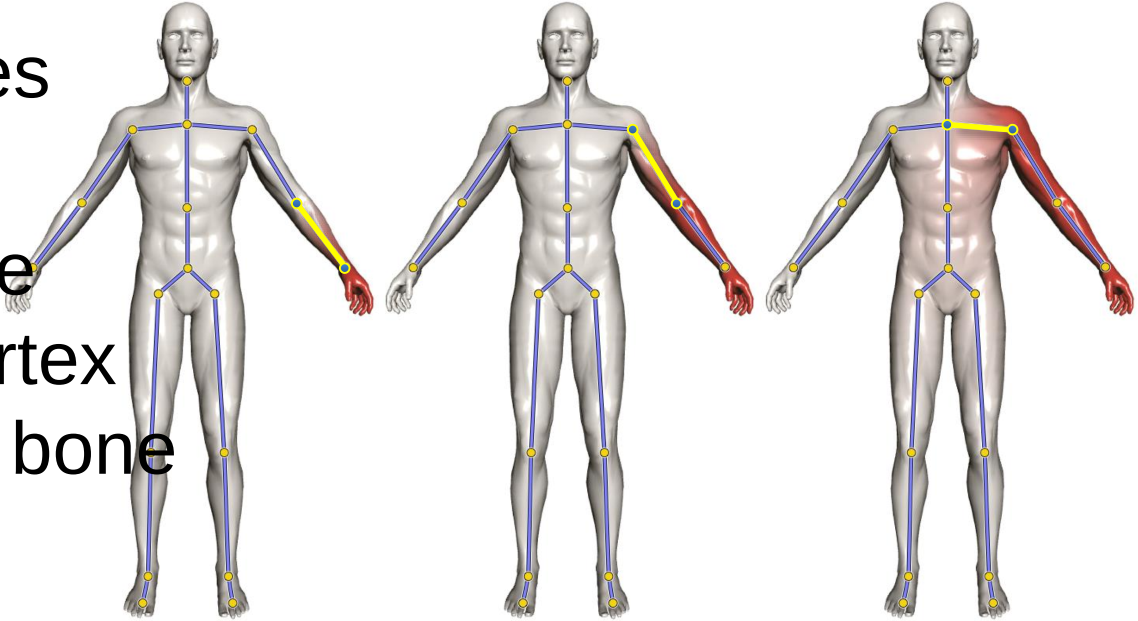
Blended Rigid Transformations

- Rigging
  - Embed the skeleton
  - Attach the bones to the model



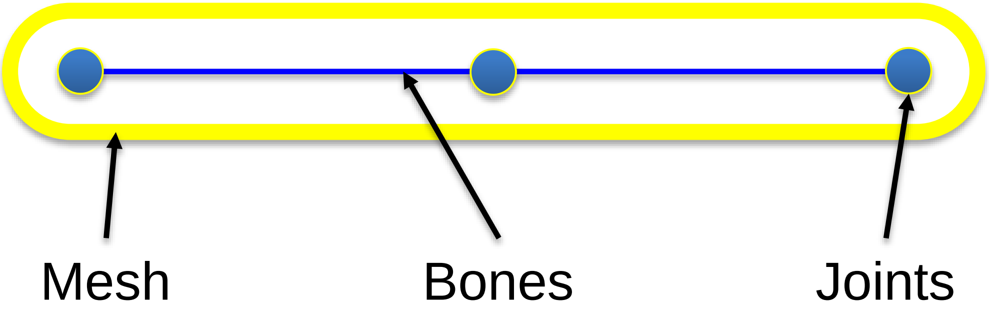
Blended Rigid Transformations

- Rigging
  - Attach the bones to the model
  - Weights indicate how much a vertex is effected by a bone



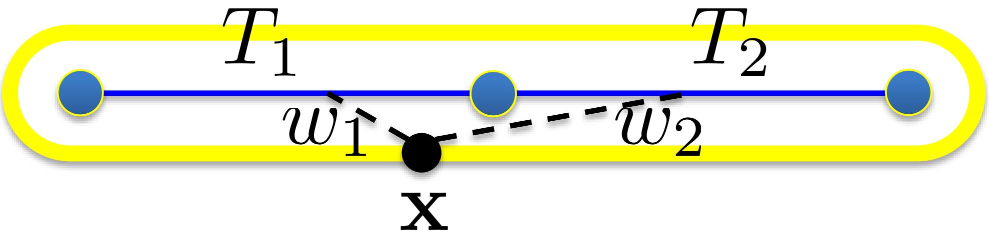
Blended Rigid Transformations

- Rigging
  - Attach the bones to the model



Blended Rigid Transformations

- Rigging
  - Attach the bones to the model



$$T(\mathbf{x}) = \mathrm{avg}(T_1, T_2, w_1, w_2)$$

Blended Rigid Transformations

- How to blend (average) transformations

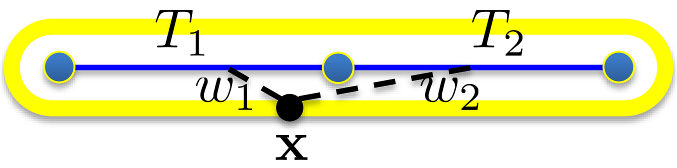
Linear Blend Skinning

Represent T_i with $\mathbf{T}_i$

in homogenous coordinates

$$\mathbf{T}(\mathbf{x}) = w_1\mathbf{T}_1 + w_2\mathbf{T}_2$$

$$\mathbf{x}' = \mathbf{T}\mathbf{x}$$

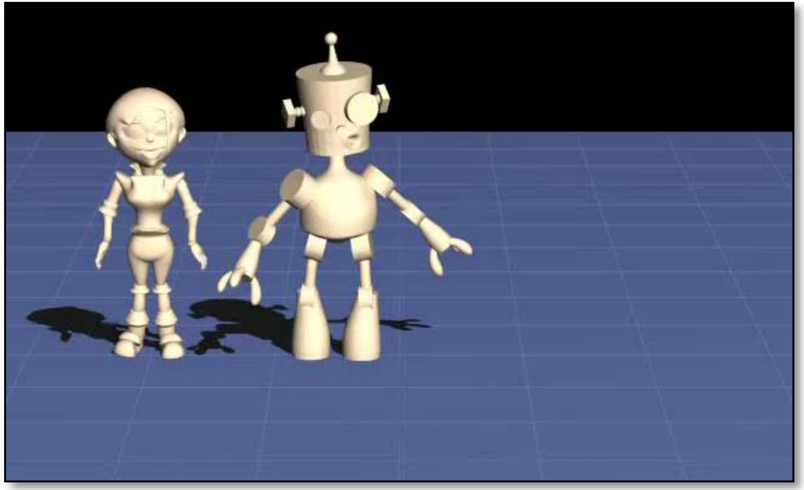


$$T(\mathbf{x}) = \operatorname{avg}(T_1, T_2, w_1, w_2)$$

Blended Rigid Transformations

- How to blend (average) transformations

Linear Blend Skinning



Blended Rigid Transformations

- How to blend (average) transformations

Linear Blend Skinning

$$w_1\mathbf{T}_1 + w_2\mathbf{T}_2$$

$$w_1\mathbf{t}_1 + w_2\mathbf{t}_2$$

Translation

$$\boxed{w_1\mathbf{R}_1 + w_2\mathbf{R}_2}$$

Rotation

Not a valid rotation matrix!

Blended Rigid Transformations

- How to blend (average) transformations

Valid rotation matrix

$$\mathbf{R}^T = \mathbf{R}^{-1}$$

$$\det(\mathbf{R}) = 1$$

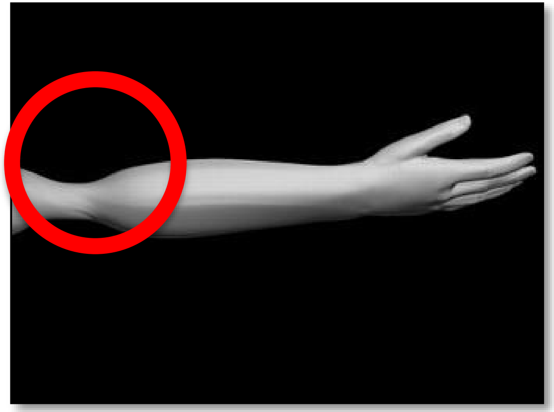
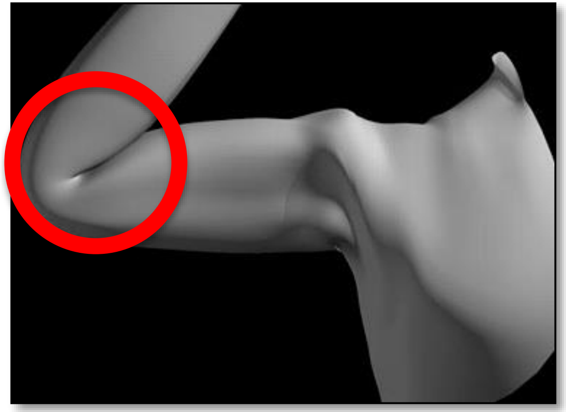
Linear blending

$$\begin{aligned} & (w_1\mathbf{R}_1 + w_2\mathbf{R}_2)^T \\ = & (w_1\mathbf{R}_1^T + w_2\mathbf{R}_2^T) \\ \neq & (w_1\mathbf{R} + w_2\mathbf{R})^{-1} \end{aligned}$$

Blended Rigid Transformations

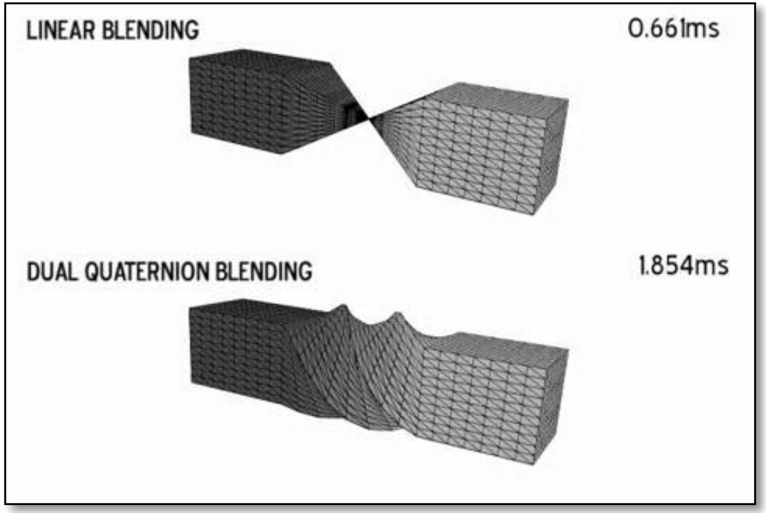
- How to blend (average) transformations

Linear Blend Skinning: problems

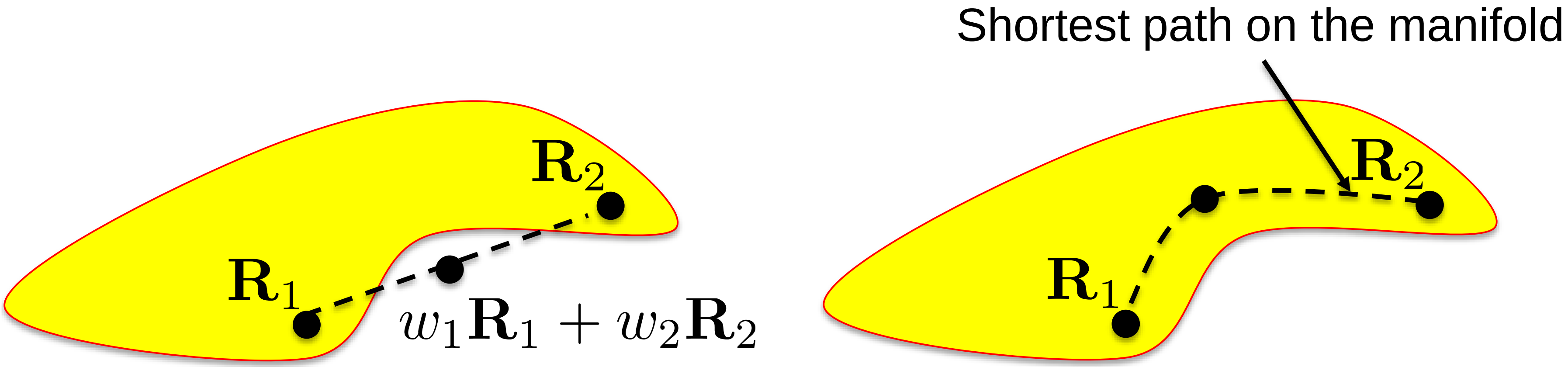
Blended Rigid Transformations

- How to blend transformations



Blended Rigid Transformations

- How to blend transformations



Manifold of rigid transformations

Manifold of rigid transformations

Rigid Transformations

- Manifold of rotations – SO (3)

Valid rotation matrix

$$\mathbf{R}^T = \mathbf{R}^{-1}$$
$$\det(\mathbf{R}) = 1$$

- Manifold of rigid transformations – SE (3)

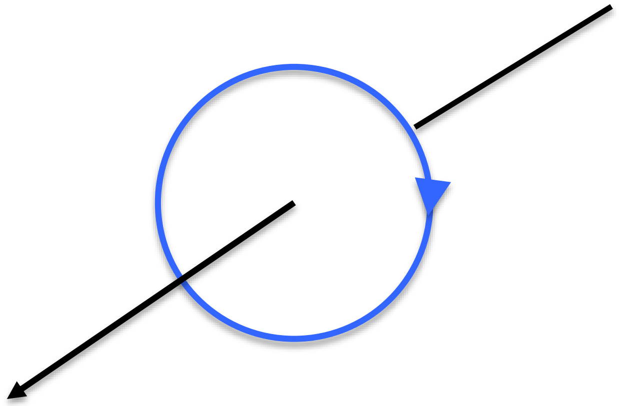
$$\mathbf{R}^T = \mathbf{R}^{-1} \qquad \mathbf{T} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ 0\;0\;0 & 1 \end{pmatrix}$$
$$\det(\mathbf{R}) = 1$$

Rigid Transformations

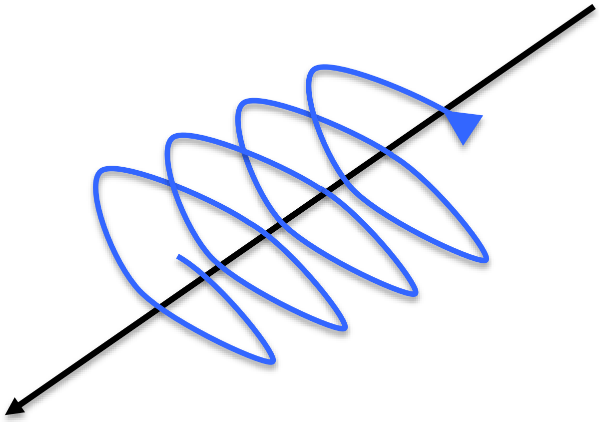
- Matrices not convenient for blending
- Alternative representation: dual quaternions

Rigid Transformations

- Representing rigid transformations



Rotations with quaternions



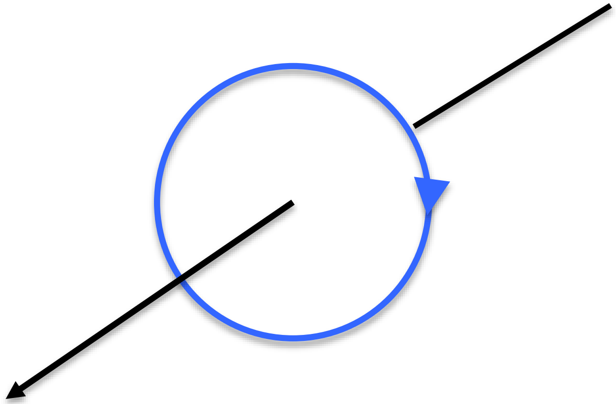
Rigid motions with dual quaternions

Rotations

- Representing rotations with quaternions

$$\mathbf{q} = \cos\left(\frac{\theta}{2}\right) + \mathbf{s}\sin\left(\frac{\theta}{2}\right)$$

Quaternion Rotation angle Rotation axis



Rotations with quaternions

Rotations

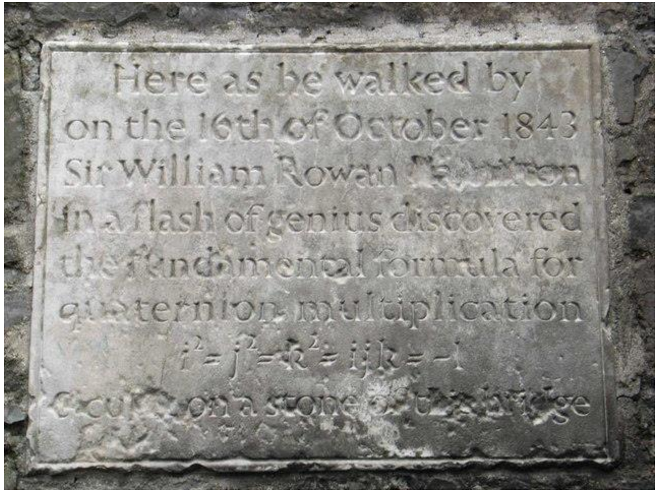
- Quaternions

$$\mathbf{q} = \cos\left(\frac{\theta}{2}\right) + \mathbf{s}\sin\left(\frac{\theta}{2}\right)$$

$$\mathbf{s} = s_i i + s_j j + s_k k$$

$$s_i^2 + s_j^2 + s_k^2 = 1$$

$$i^2 = j^2 = k^2 = ijk = -1$$



Rotations

- Operations on quaternions

$$\mathbf{q} = \cos\left(\frac{\theta}{2}\right) + \mathbf{s}\sin\left(\frac{\theta}{2}\right)$$

Conjugate

$$\mathbf{q}^* = \cos\left(\frac{\theta}{2}\right) - \mathbf{s}\sin\left(\frac{\theta}{2}\right) = \cos\left(-\frac{\theta}{2}\right) + \mathbf{s}\sin\left(-\frac{\theta}{2}\right)$$

Inverse (for unit quaternions) $\mathbf{q}^{-1} = \mathbf{q}^*$

Rotations

- Operations on quaternions

Multiplication

$$\mathbf{q}_1\mathbf{q}_2 = (a_1 + b_1 i + c_1 j + d_1 k)(a_2 + b_2 i + c_2 j + d_2 k)$$

$$\begin{tikzcd} i \arrow[r, bend right] & j \arrow[r, bend right] & k \arrow[ll, bend right] \end{tikzcd}$$

Norm

$$||\mathbf{q}||^2 = \mathbf{q}\mathbf{q}^* = \cos^2\left(\frac{\theta}{2}\right) + ||s||^2 \sin^2\left(\frac{\theta}{2}\right) = 1$$

Rotations

- Operations on quaternions

$$\mathbf{q} = \cos\left(\frac{\theta}{2}\right) + \mathbf{s}\sin\left(\frac{\theta}{2}\right)$$

Power

$$\mathbf{q}^t = e^{t\log\mathbf{q}}$$

$$\log\mathbf{q} = \frac{\theta}{2}\mathbf{s} \qquad e^{\mathbf{q}} = \cos||\mathbf{q}|| + \frac{\mathbf{q}}{||\mathbf{q}||}\sin||\mathbf{q}||$$

Rotations

- Operations on quaternions

$$\mathbf{q} = \cos\left(\frac{\theta}{2}\right) + \mathbf{s}\sin\left(\frac{\theta}{2}\right)$$

Applying to location vectors

$$\mathbf{v} = v_i i + v_j j + v_k k$$

$$\mathbf{v}' = \mathbf{q}\mathbf{v}\mathbf{q}^*$$

Rotations

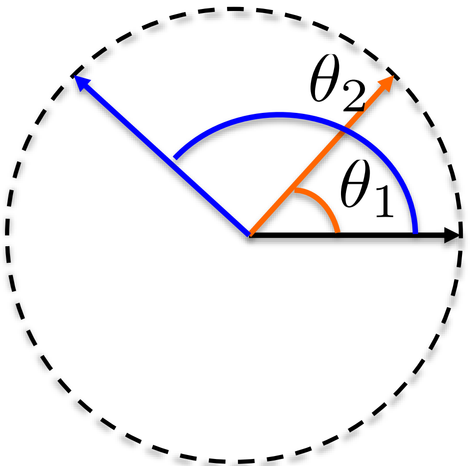
- Blending quaternions

$$\mathbf{s} = \mathbf{s}_1 = \mathbf{s}_2$$

$$\mathrm{interpolate}(\mathbf{q}_1, \mathbf{q}_2, t)$$

$$\theta(t) = (1-t)\theta_1 + t\theta_2$$

$$\mathbf{q}(t) = \cos\left(\frac{\theta(t)}{2}\right) + \mathbf{s}\sin\left(\frac{\theta(t)}{2}\right)$$



Rotations

- Blending quaternions
  - In general, $\mathbf{s}_1 \neq \mathbf{s}_2$
  - Spherical blending

$$(\mathbf{q}_2\mathbf{q}_1^*)^t\mathbf{q}_1$$

  - More than two rotations?

Rotations

- Blending quaternions

$\mathbf{q}_1 \cdots \mathbf{q}_n \quad w_1 \cdots w_n$

  - Good approximation:

$$\mathbf{b} = \sum_{i=1}^{n} w_i \mathbf{q}_i$$

Rigid Transformations

- Rotation & translation
- Dual numbers

$$\hat{x} = x_0 + \epsilon x_\epsilon \quad \epsilon^2 = 0$$

E.g. multiplication

$$(a_0 + \epsilon a_\epsilon)(b_0 + \epsilon b_\epsilon)$$
$$= a_0 b_0 + \epsilon(a_0 b_\epsilon + a_\epsilon b_0)$$

Rigid Transformations

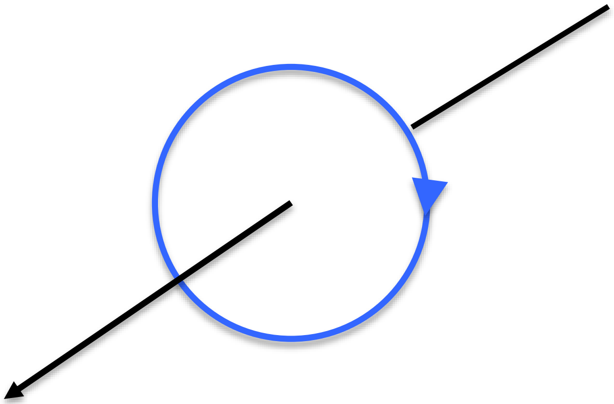
- Dual quaternions
  - Replace numbers in quaternions with dual numbers

$$\hat{\mathbf{q}} = \cos\left(\frac{\hat{\theta}}{2}\right) + \hat{\mathbf{s}}\sin\left(\frac{\hat{\theta}}{2}\right)$$

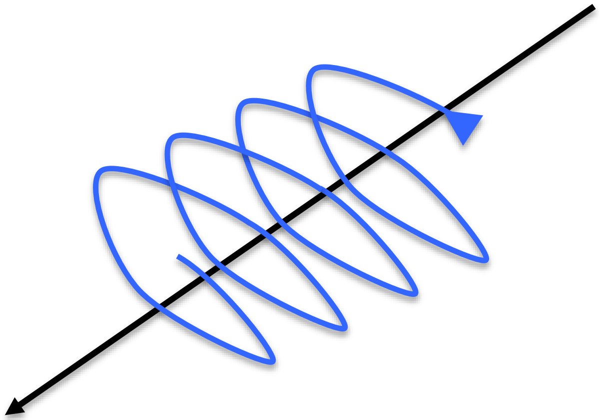
  - Almost all operations & notations are the same
  - In particular: $\hat{\mathbf{b}} = \sum_{i=1}^{n} w_i \hat{\mathbf{q}}_i$

Rigid Transformations

- Representing rigid transformations



Quaternions : 4 numbers



Dual quaternions : 8 numbers

Blended Rigid Transformations

- Properties

$$\hat{\mathbf{b}} = \sum_{i=1}^{n} w_i \hat{\mathbf{q}}_i$$

1. Generates valid transformations
   - Only if normalized!

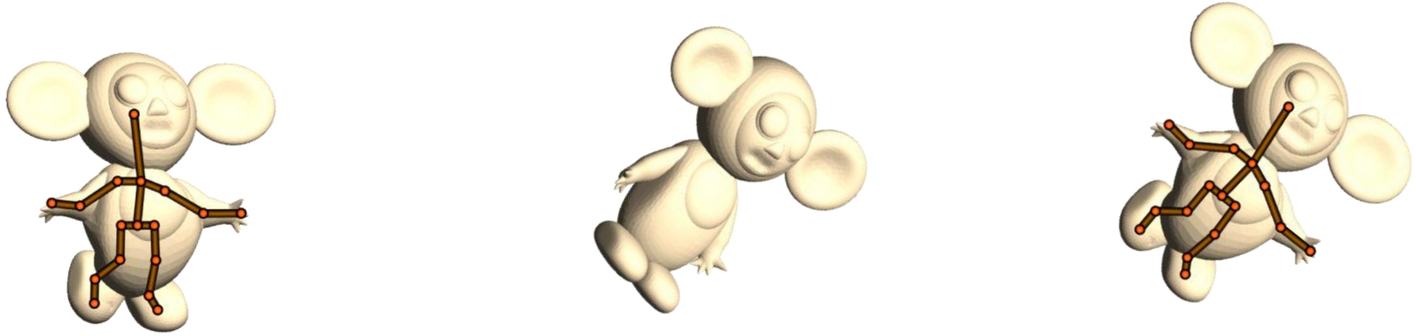
$$\hat{\mathbf{b}} = \frac{\sum_{i=1}^{n} w_i \hat{\mathbf{q}}_i}{||\sum_{i=1}^{n} w_i \hat{\mathbf{q}}_i||}$$

Blended Rigid Transformations

- Properties

$$\hat{\mathbf{b}} = \frac{\sum_{i=1}^{n} w_i \hat{\mathbf{q}}_i}{||\sum_{i=1}^{n} w_i \hat{\mathbf{q}}_i||}$$

2. Coordinate invariance

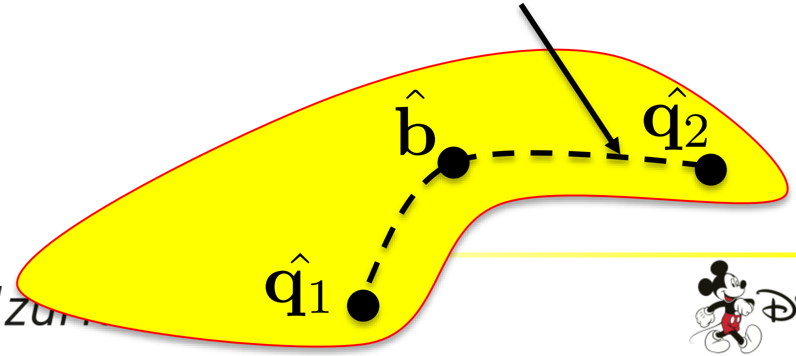


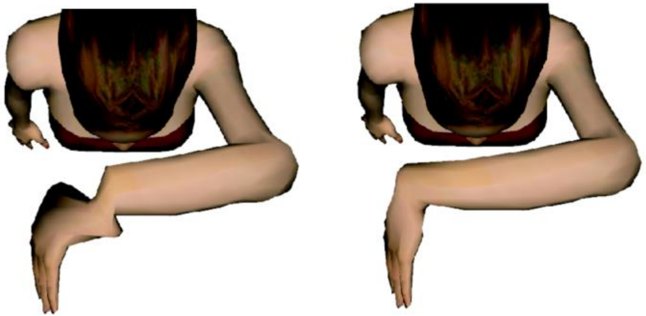
Blended Rigid Transformations

- Properties

$$\hat{\mathbf{b}} = \frac{\sum_{i=1}^{n} w_i \hat{\mathbf{q}}_i}{||\sum_{i=1}^{n} w_i \hat{\mathbf{q}}_i||}$$

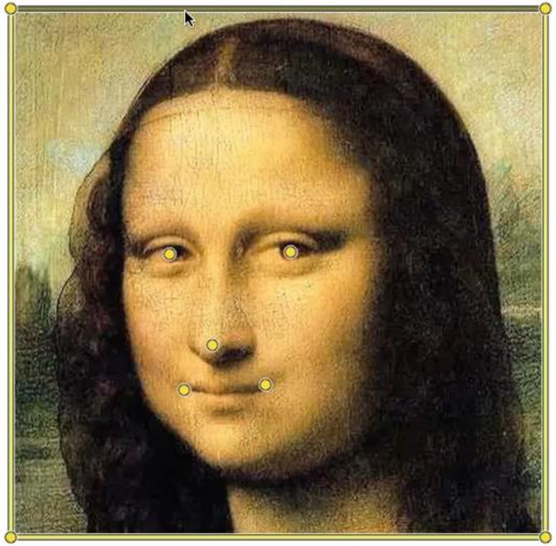
3. Shortest path on SE (3)





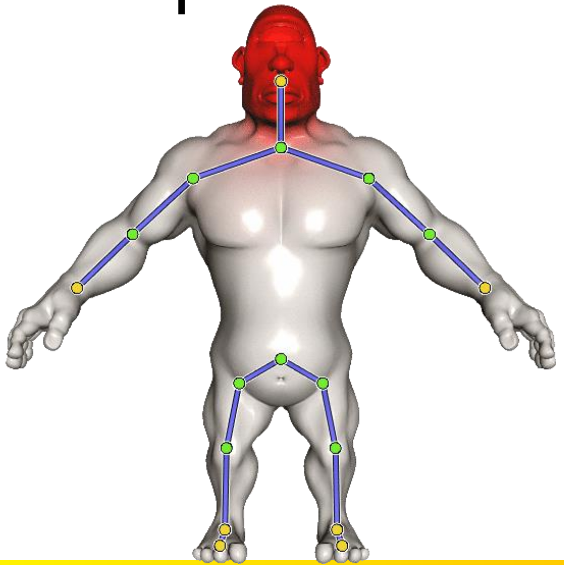
Blended Rigid Transformations

- Generalizes to other applications



Blended Rigid Transformations

- Challenges
  - Blending transformations – dual quaternions
  - Weights $w_i(\mathbf{x})$
    - Shape adaptive
    - Intuitive deformations
    - Smooth deformations

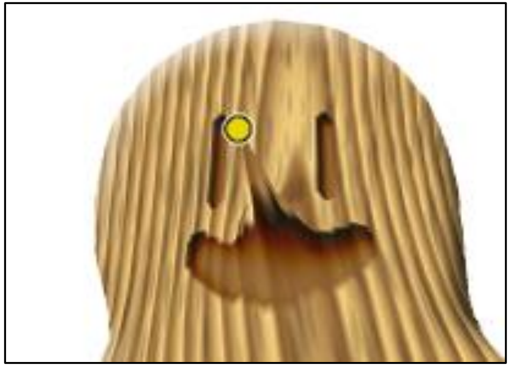


Blended Rigid Transformations

- Weights – desired properties
  - Partition of unity

$$\sum_{i=1}^{n} w_i(\mathbf{x}) = 1$$

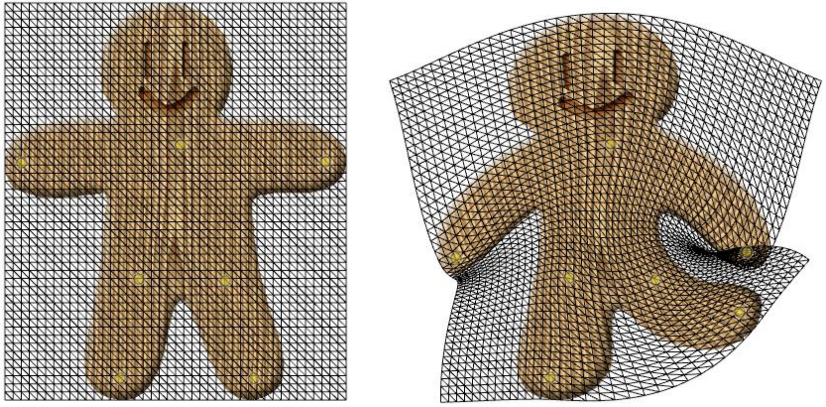
  - Smoothness



Blended Rigid Transformations

- Weights – desired properties
  - Shape-awareness

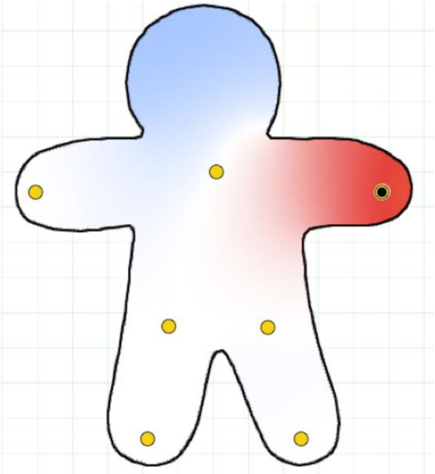
Shape-aware weights

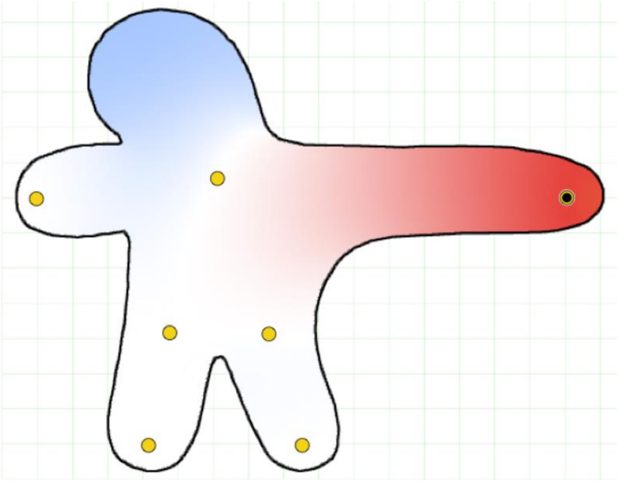


Shape-unaware weights

Blended Rigid Transformations

- Weights – desired properties
  - Non-negativity





Blended Rigid Transformations

- Blending + weights

Some references

- Applications
  - Sparse Iterative Closest Point
  - FAUST: Dataset and Evaluation for 3D Mesh Registration
  - Bundled Camera Paths for Video Stabilization
  - Auto-Directed Video Stabilization with Robust L1 Optimal Camera Paths
  - Robust Estimation of 3D Human Poses from a Single Image
- Transformation blending
  - Geometric Skinning with Approximate Dual Quaternion Blending
- Weight/coordinate computation
  - Mean Value Coordinates
  - Harmonic Coordinates for Character Articulation
  - Bounded Biharmonic Weights for Real-Time Deformation
  - Local Barycentric Coordinates
  - Automatic Rigging and Animation of 3D Characters
- Deformations
  - Lifted Bijections for Low Distortion Surface Mappings