Deformation & Skinning
形变与蒙皮

北京大学 前沿计算研究中心

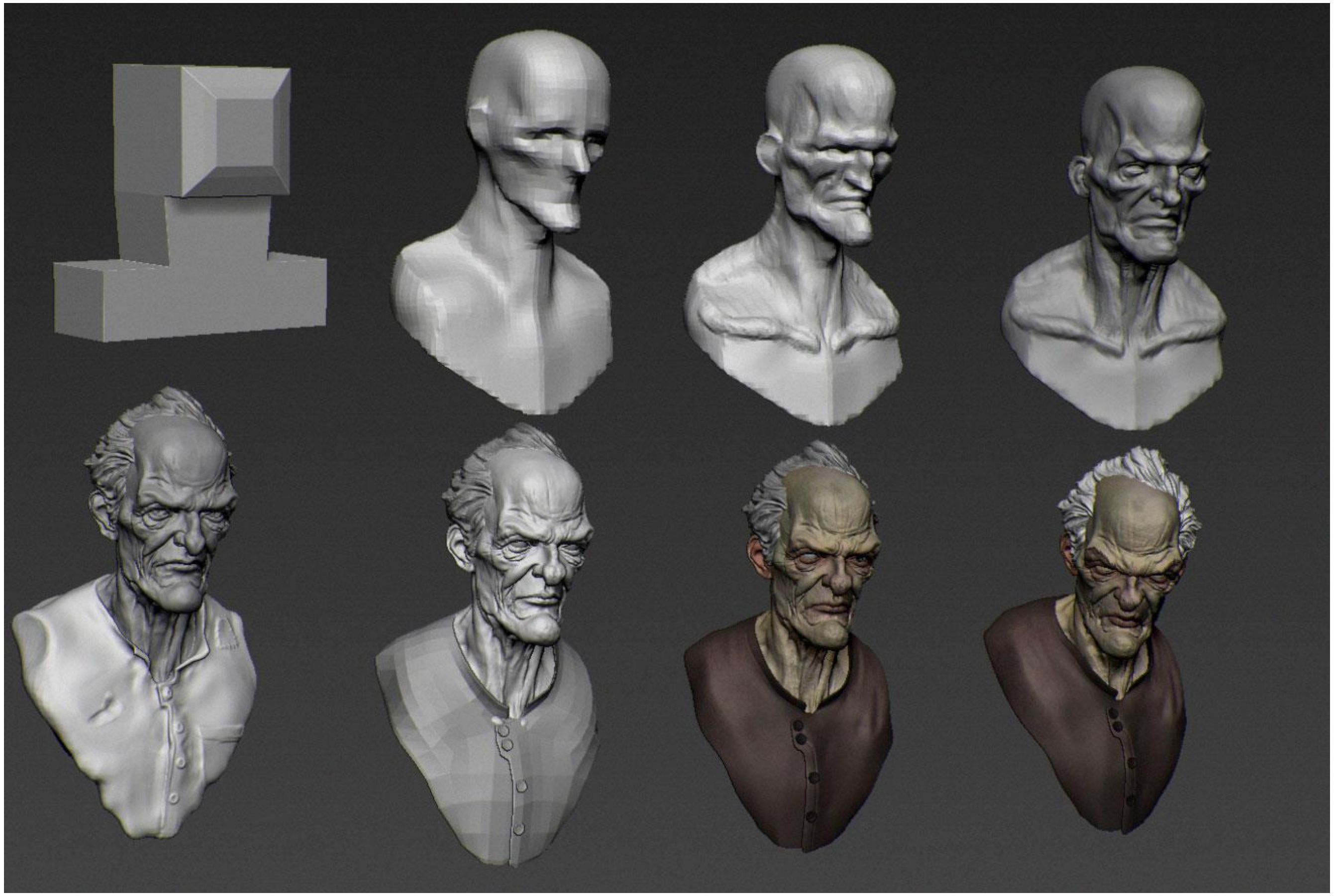
刘利斌

本节内容

- 几何形状的编辑
  - 自由形变 (Free-form Deformation)
  - 基于优化的形变
- 蒙皮形变 (Skinning Deformation)
  - 线性蒙皮
    - Linear-Blend Skinning (LBS)
    - Multi-linear Skinning
  - 非线性蒙皮
    - Dual-quaternion Skinning (DQS)
  - 基于样例的蒙皮
    - Blendshapes
  - 人脸动画

几何形变与编辑

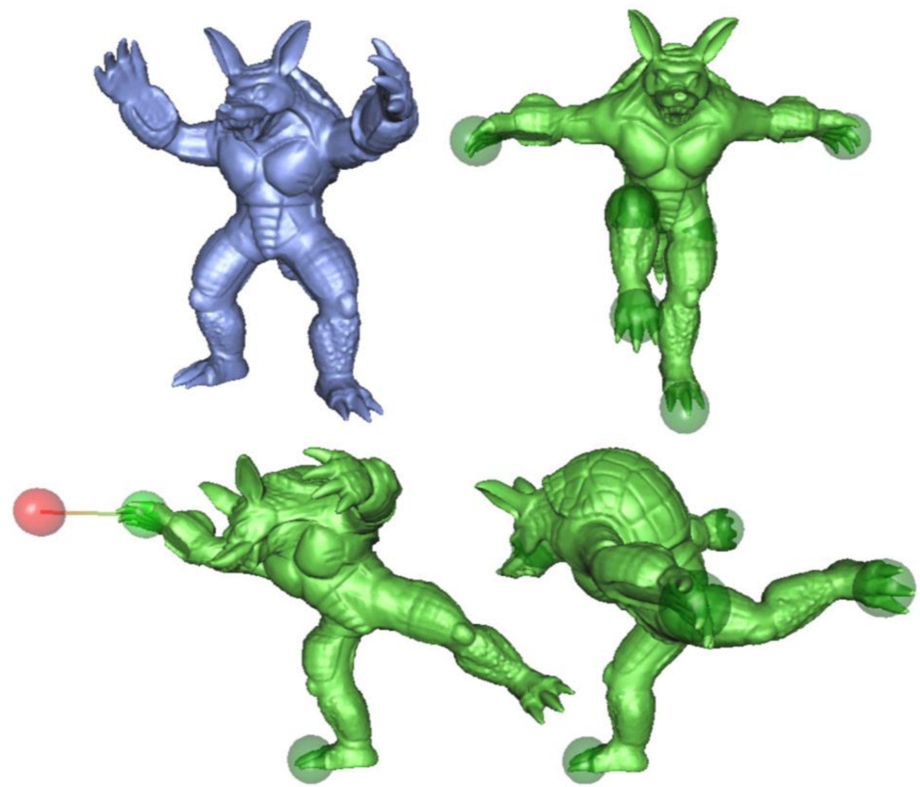
- 直接编辑几何模型顶点/控制点



https://khanhha.github.io/portfolio/skeleton-to-mesh-conversion/

几何形变与编辑

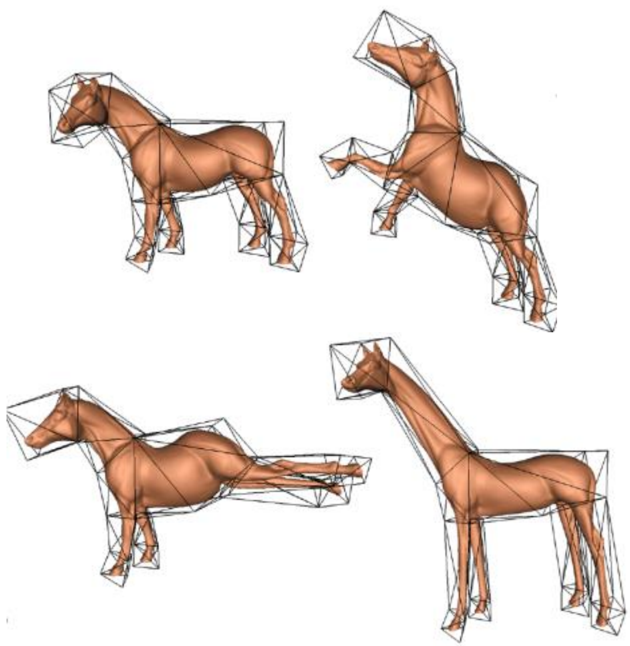
- 编辑修改几何形状，同时保留几何模型细节
- 一般通过整体性的改变几何模型的参数来实现
  - 网格 -> 顶点
  - NURBS -> 控制点
  - ……



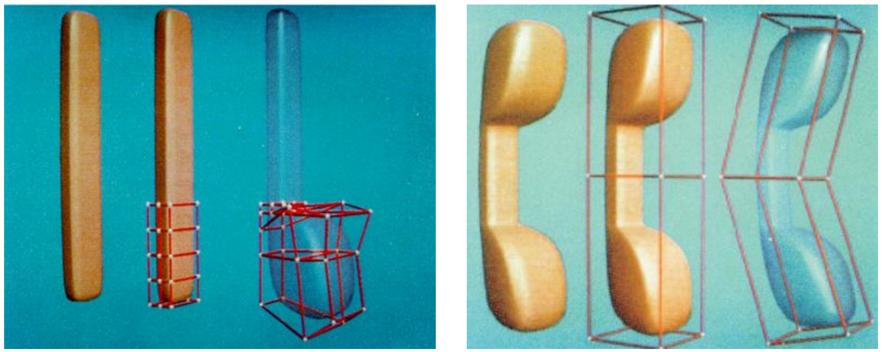
Xiaohan Shi, Kun Zhou, Yiying Tong, Mathieu Desbrun, Hujun Bao, and Baining Guo. 2007. ***Mesh puppetry: cascading optimization of mesh deformation with inverse kinematics***. *ACM Trans. Graph.* 26, 3 (July 2007),

自由变形与Cage-based Deformation

- “扭曲”模型附近的空间来实现变形
  - 将几何模型嵌入到简单网格中
  - 改变简单网格的形状间接的改变模型的形状



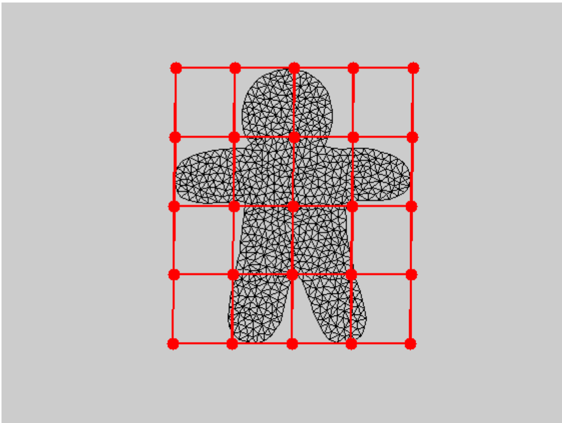
Ju, T., Schaefer, S., Warren, J.: ***Mean value coordinates for closed triangular meshes***. In: ACM SIGGRAPH 2005 Papers, SIGGRAPH ’05, pp. 561–566.

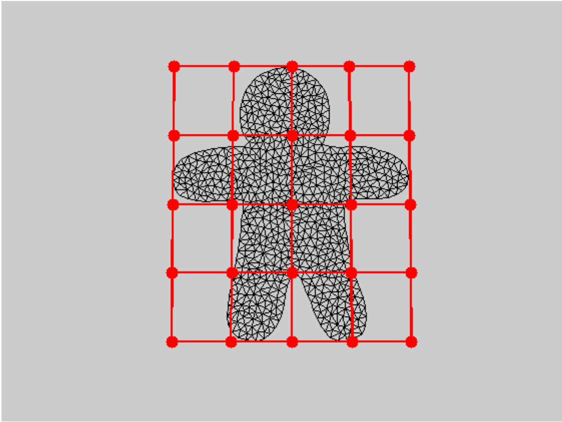


Thomas W. Sederberg and Scott R. Parry. 1986. ***Free-form deformation of solid geometric models***. In Proceedings of the 13th annual conference on Computer graphics and interactive techniques (SIGGRAPH '86).

自由变形

- 如何定义嵌入关系?
  - 嵌入的精细网格 $\{x_i\}$
  - 简单网格 $\{p_j\}$



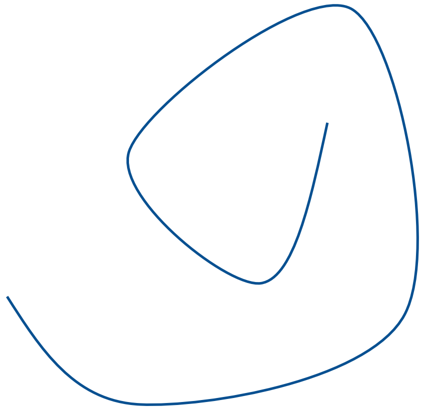


嵌入关系:

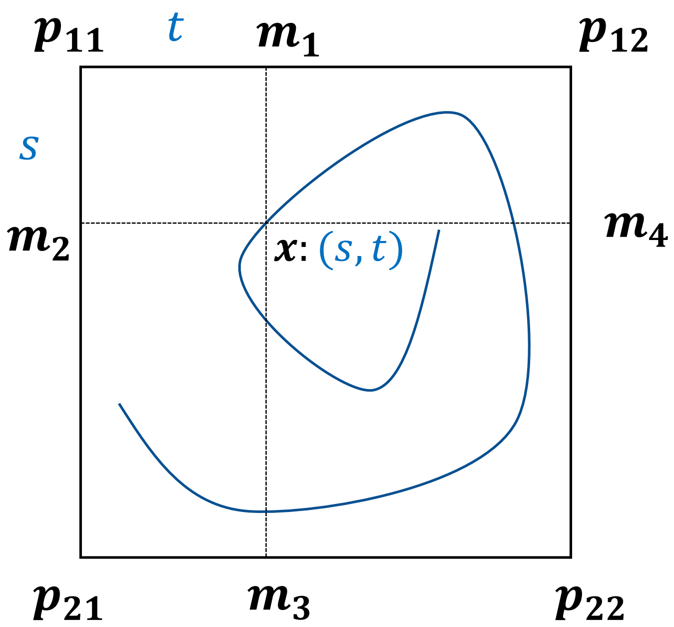
$$x_i = \sum_j \omega_{ij} p_j$$

- 如何确定 ω_{ij} ??

线性插值

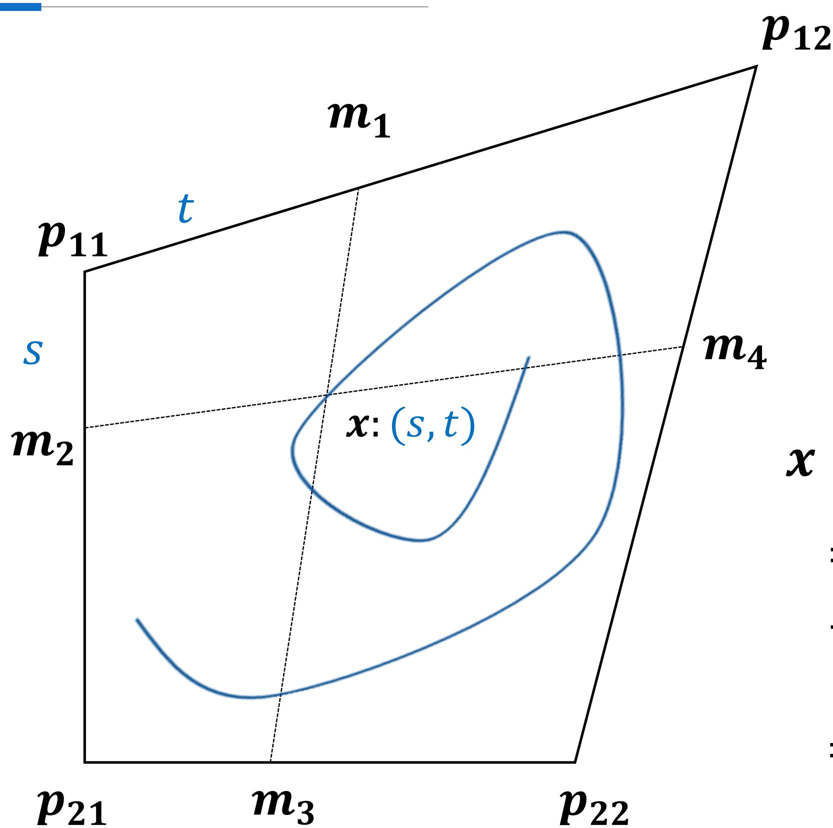


线性插值



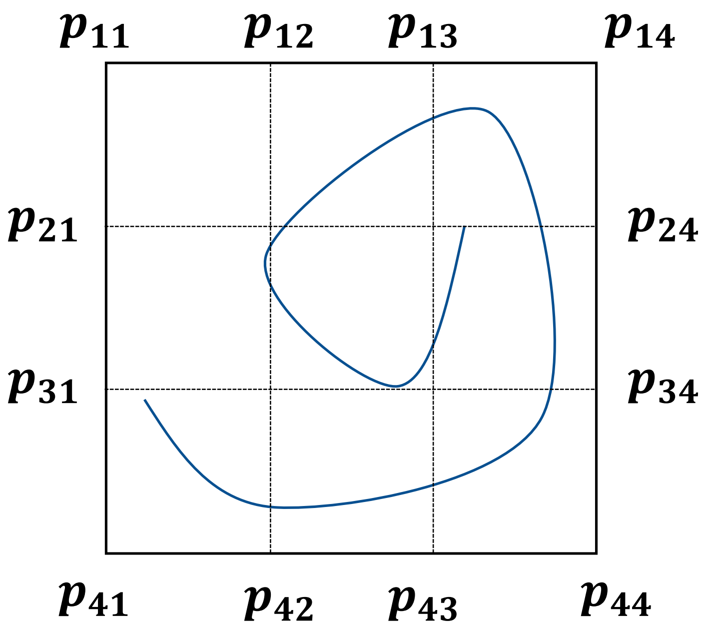
$$
\begin{aligned}
\boldsymbol{x} &= (1-t)\boldsymbol{m}_2 + t\boldsymbol{m}_4 \\
&= (1-t)\big((1-s)\boldsymbol{p}_{11} + s\boldsymbol{p}_{21}\big) \\
&+ \quad\quad t\ \big((1-s)\boldsymbol{p}_{12} + s\boldsymbol{p}_{22}\big) \\
&= \sum_i \sum_j B_i(s)B_j(t)\boldsymbol{p}_{ij}
\end{aligned}
$$

线性插值



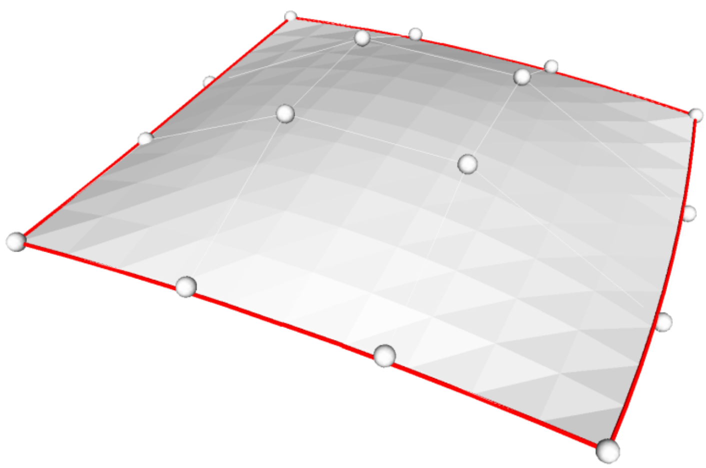
$$
\begin{aligned}
\boldsymbol{x} &= (1-t)\boldsymbol{m}_2 + t\boldsymbol{m}_4 \\
&= (1-t)\big((1-s)\boldsymbol{p}_{11} + s\boldsymbol{p}_{21}\big) \\
&+ \quad t\,\big((1-s)\boldsymbol{p}_{12} + s\boldsymbol{p}_{22}\big) \\
&= \sum_i \sum_j B_i(s) B_j(t) \boldsymbol{p}_{ij}
\end{aligned}
$$

曲线插值



$$x = \sum_{i=0}^{n} \sum_{j=0}^{m} w_{i,j}(u, v) \boldsymbol{p}_{i,j}$$

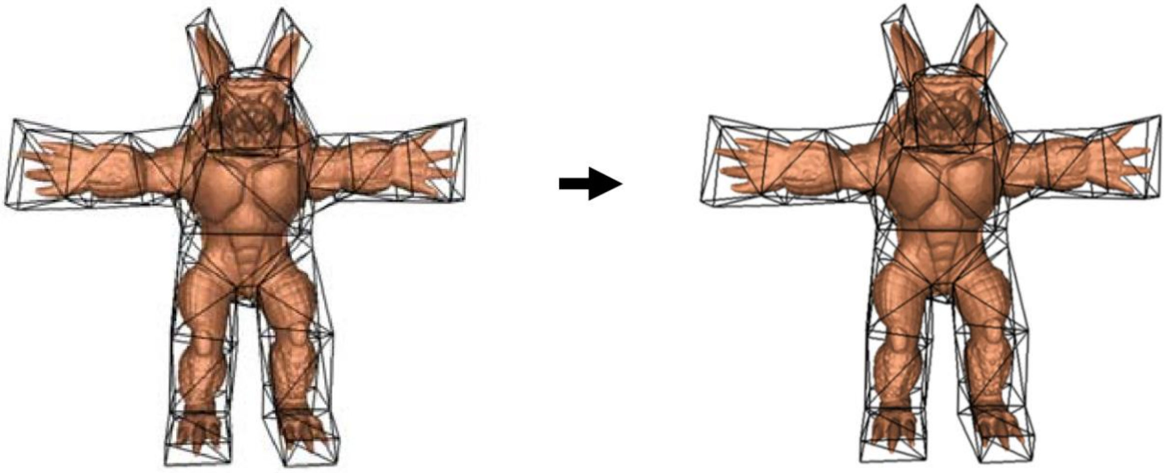
Bézier Surface, NURBS Surface…



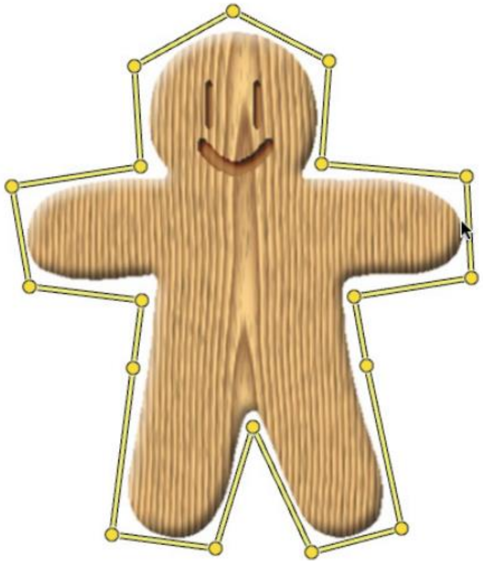
Cage-based Deformation

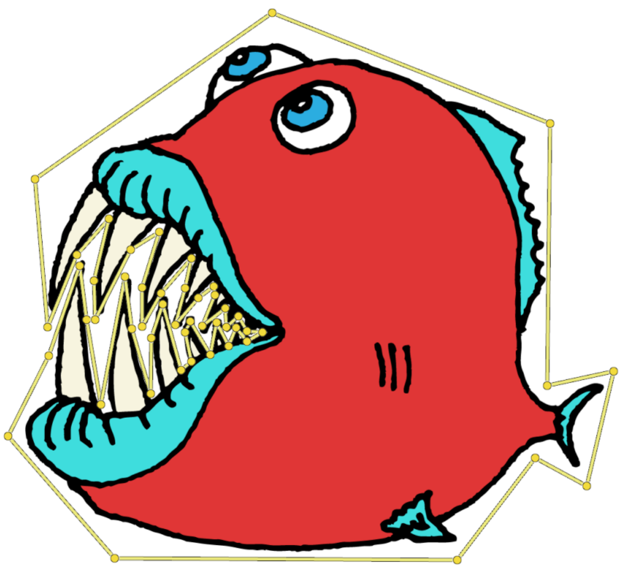
$$x_i = \sum_j \omega_{ij} p_j$$

基于Cage计算权重

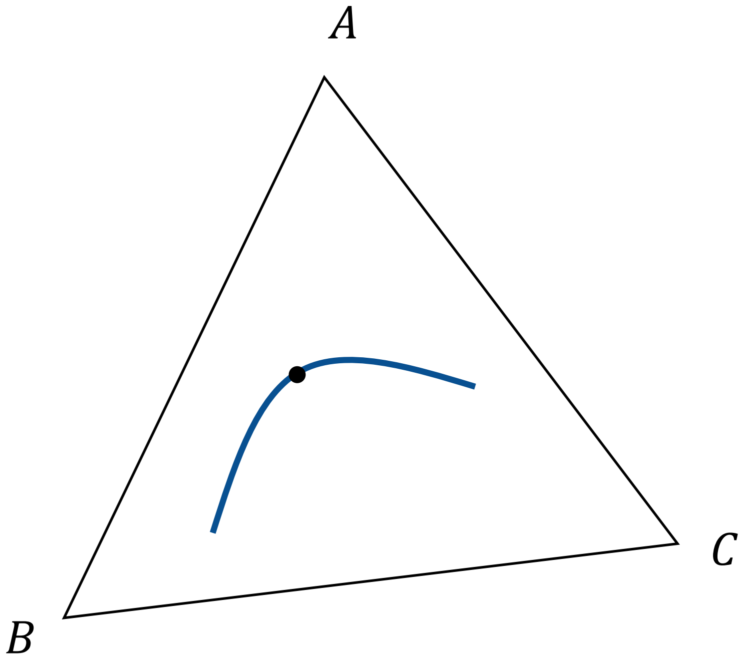


任意形状

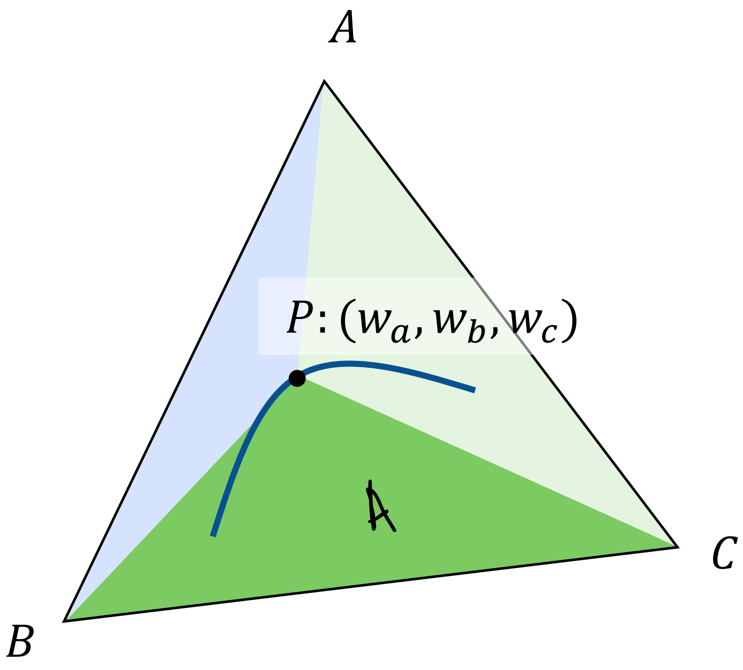




重心坐标



重心坐标



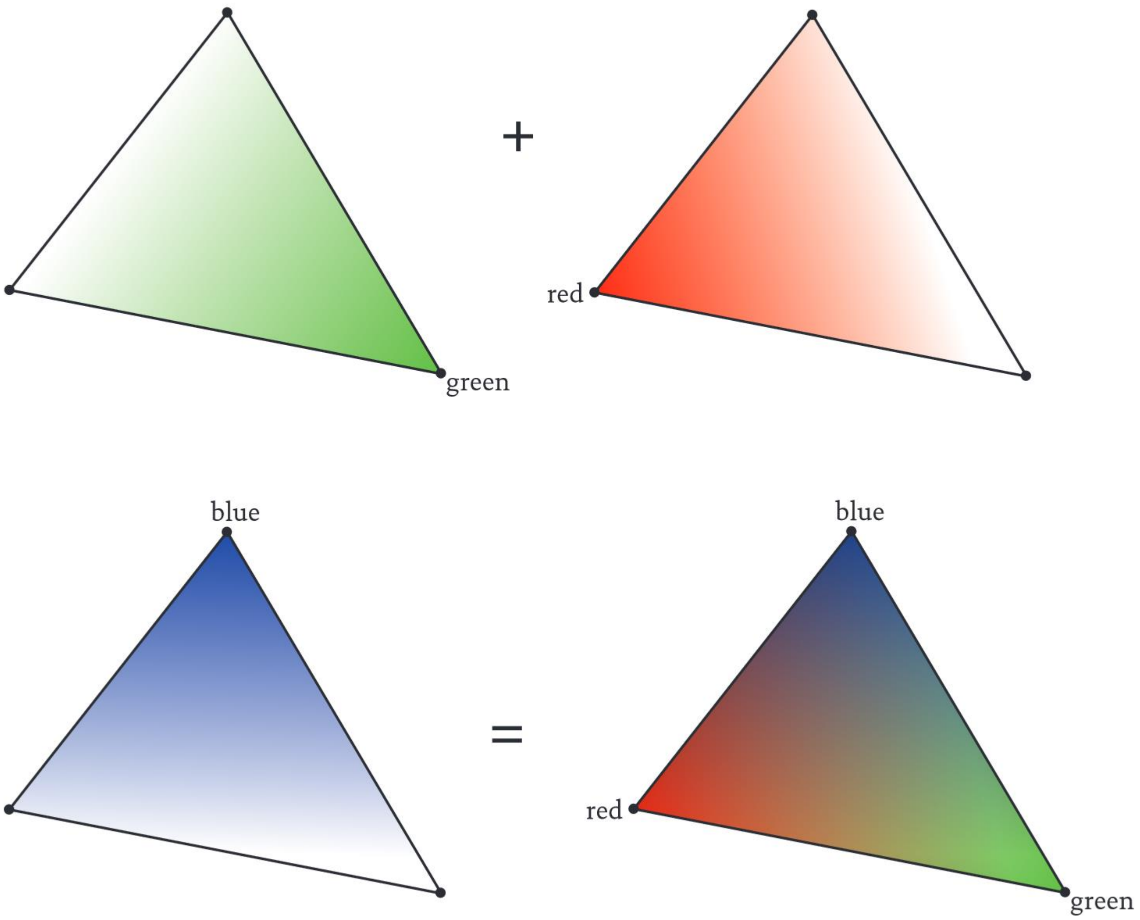
$$P = w_a A + w_b B + w_c C$$

$$w_a = \frac{\Delta PBC}{\Delta ABC}$$

$$w_b = \frac{\Delta PCA}{\Delta ABC}$$

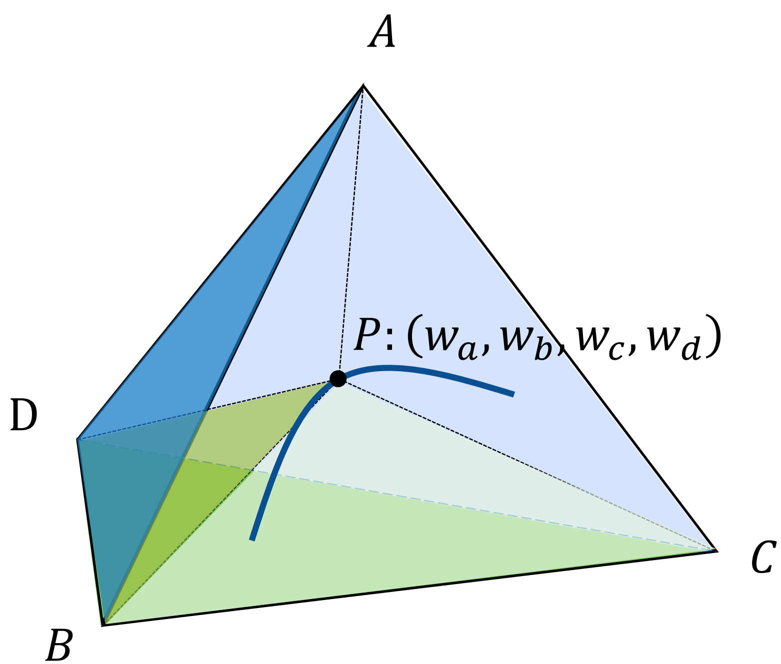
$$w_c = \frac{\Delta PAC}{\Delta ABC}$$

重心坐标



Lidberg, Petter. "Barycentric and Wachspress coordinates in two dimensions: theory and implementation for shape transformations." (2011).

重心坐标



$$P = w_a A + w_b B + w_c C + w_d D$$

$$w_a = \frac{\Delta PBCD}{\Delta ABCD}$$

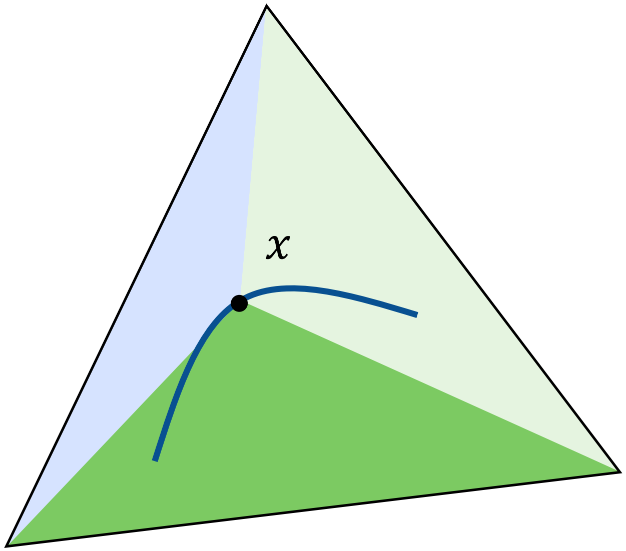
$$w_b = \frac{\Delta PCDA}{\Delta ABCD}$$

$$w_c = \frac{\Delta PDAB}{\Delta ABCD}$$

$$w_d = \frac{\Delta PABC}{\Delta ABCD}$$

三角形的重心坐标

$$x = \sum_i w_i p_i \,, \qquad \sum_i w_i = 1, \text{且}\, w_i \geq 0$$



$$x = \frac{\sum_i w_i' p_i}{\sum_i w_i'}$$

- 可写作 $x = (w_0, w_1, \dots, w_n)$
- 三角形中的点有唯一重心坐标（why?）
  - 不同坐标组合仅差常系数
  - Homogeneous coordinates
  - 可扩展到N-维单纯形

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \left[(w_1 \;\; w_2 \;\; w_3) \begin{bmatrix} A_x & A_y & 1 \\ B_x & B_y & 1 \\ C_x & C_y & 1 \end{bmatrix} \right]^T$$

任意多边形的情况

$$x = \sum_i w_i p_i \,/ \sum_i w_i$$

- 任意多边形重心坐标不唯一
  - *Wachspress* (WP) coordinates

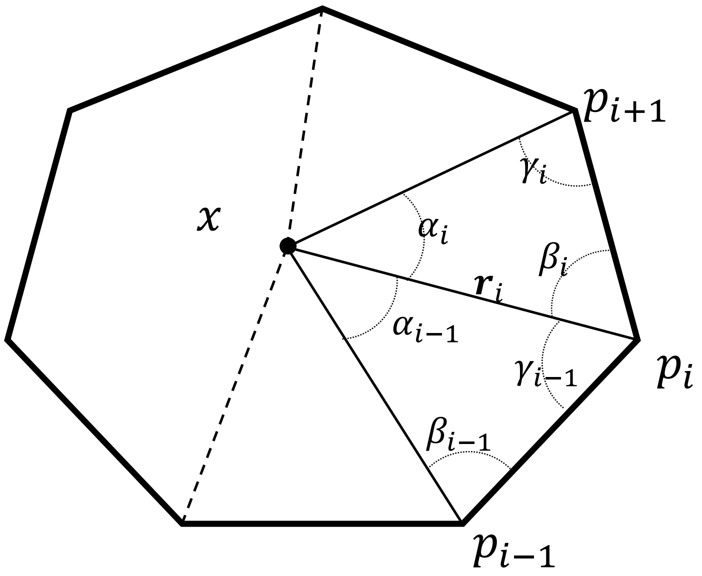
$$w_i = \frac{\cot \gamma_{i-1} + \cot \beta_i}{{r_i}^2}$$

  - *mean value* (MV) coordinates

$$w_i = \frac{\tan(\alpha_{i-1}/2) + \tan(\alpha_i/2)}{r_i}$$

  - *discrete harmonic* (DH) coordinates

$$w_i = \cot \beta_{i-1} + \cot \gamma_i$$



https://www.inf.usi.ch/hormann/barycentric/

任意多边形的情况

$$x = \sum_i w_i p_i \,/ \sum_i w_i$$

- 任意多边形重心坐标不唯一
  - *Wachspress* (WP) coordinates

$$w_i = \frac{\cot\gamma_{i-1} + \cot\beta_i}{{r_i}^2}$$

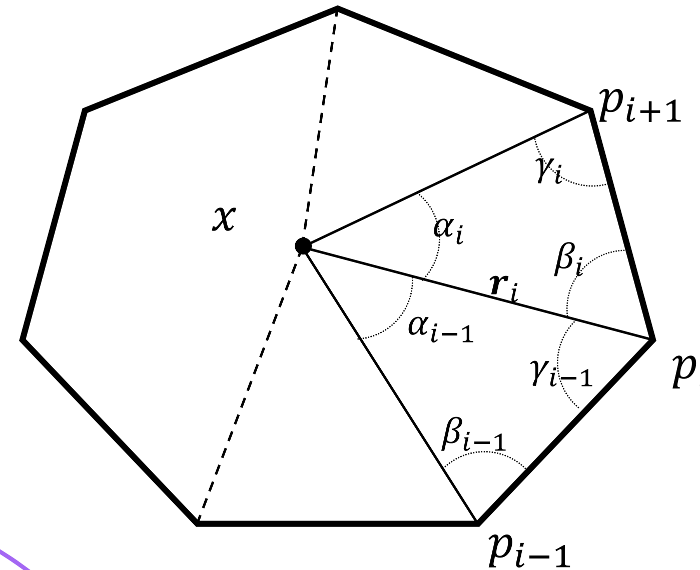
  - *mean value* (MV) coordinates

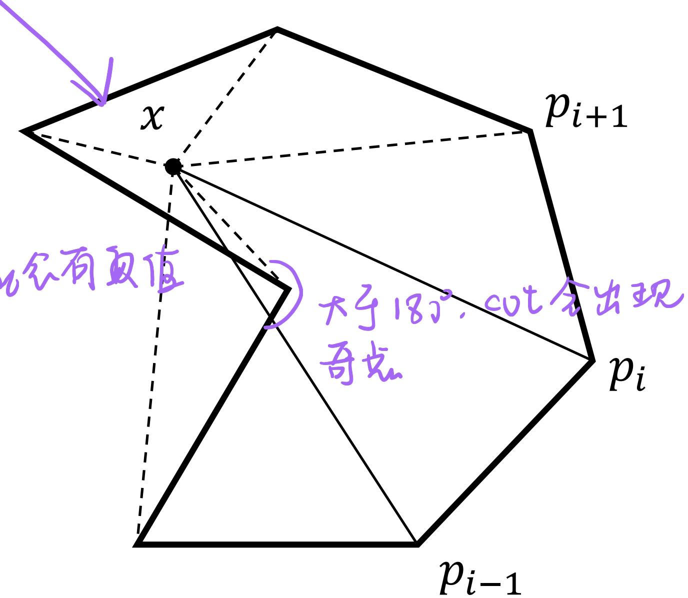
$$w_i = \frac{\tan(\alpha_{i-1}/2) + \tan(\alpha_i/2)}{r_i}$$

非凸的时候可能会有负值

  - *discrete harmonic* (DH) coordinates

$$w_i = \cot\beta_{i-1} + \cot\gamma_i$$

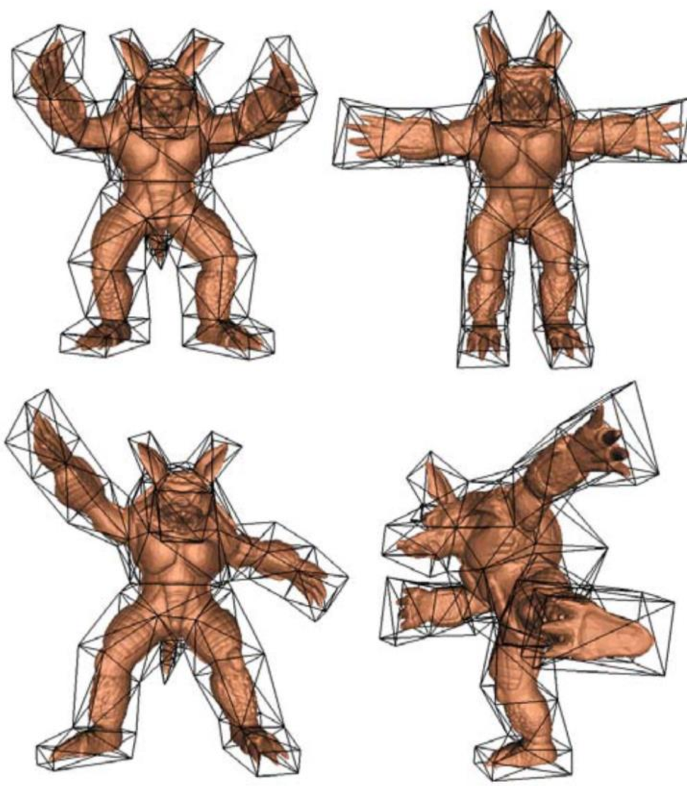




https://www.inf.usi.ch/hormann/barycentric/

任意多边形的情况

- Generalize to 3D Polyhedron is non-trivial
  - See https://www.inf.usi.ch/hormann/barycentric/



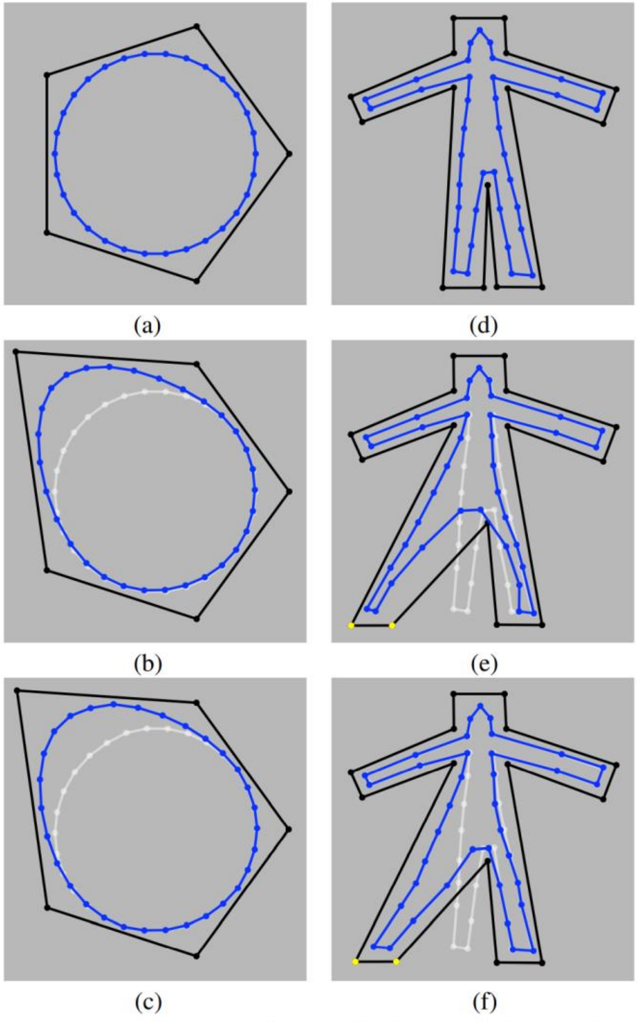
Ju, T., Schaefer, S., Warren, J.: ***Mean value coordinates for closed triangular meshes***. In: ACM SIGGRAPH 2005 Papers, SIGGRAPH '05, pp. 561–566.

更一般的情况

- Cage权重的一些性质:
  - 控制点本身的权重为1
  - 每个点的权重和为1
  - Cage内部权重连续/光滑
  - 权重非负
  - 支持线性插值

$$x_i = \sum_j \omega_{ij} p_j$$

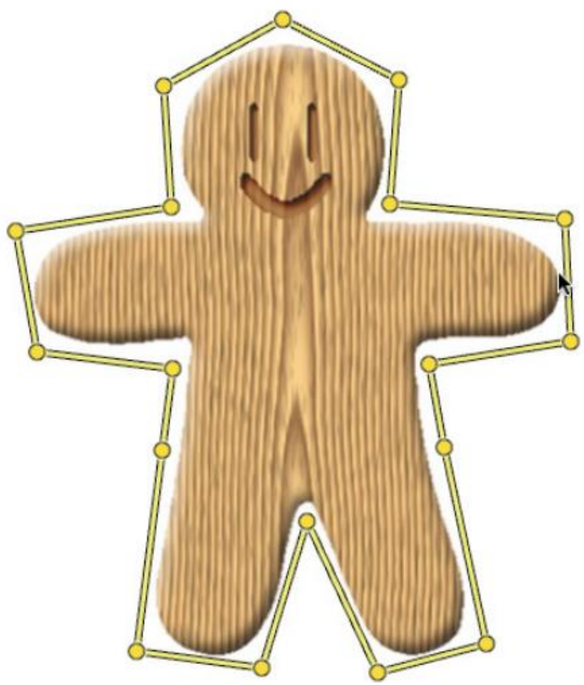
拆三角形（三角形过大）可能导致不光滑



Pushkar Joshi, Mark Meyer, Tony DeRose, Brian Green, and Tom Sanocki. 2007. ***Harmonic coordinates for character articulation***. ACM Trans. Graph. 26, 3 (July 2007), 71–es.

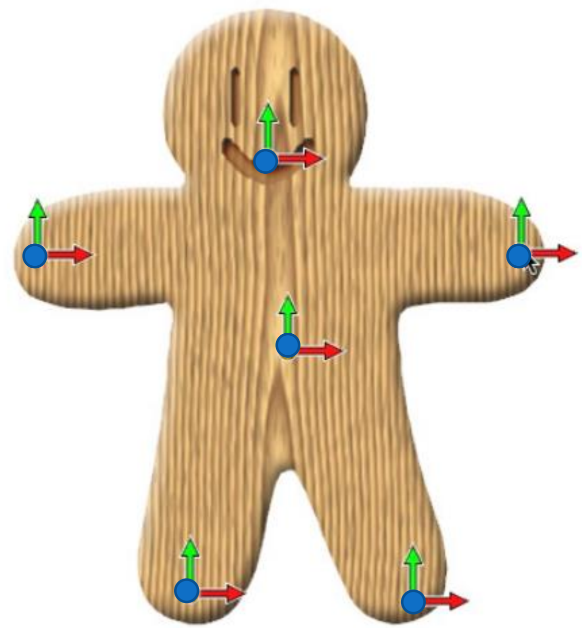
另一种形式的cage

$$x_i = \sum_j \omega_{ij} p_j$$



Cage

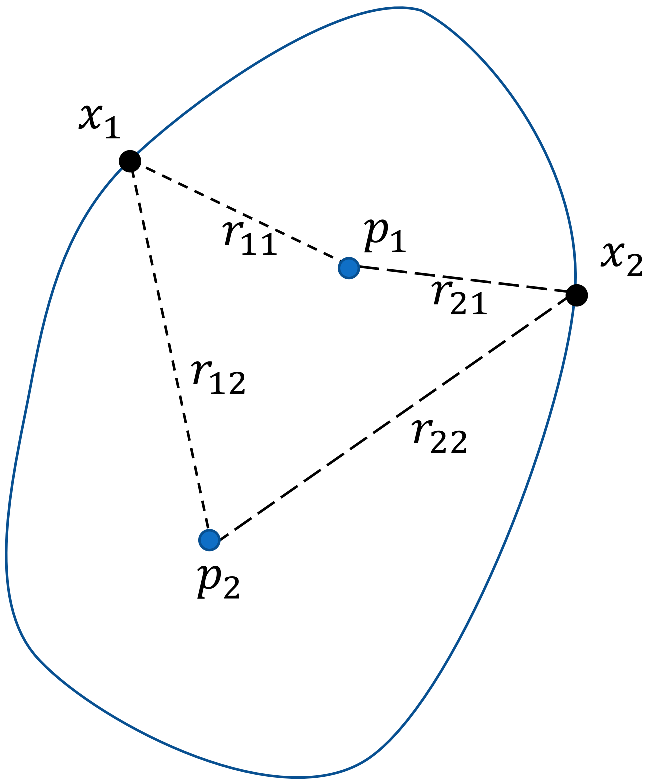
$$x_i = \sum_j \omega_{ij} p_j \quad ??$$



控制点

Alec Jacobson, Ilya Baran, Jovan Popović, Olga Sorkine-Hornung, ***Bounded Biharmonic Weights for Real-Time Deformation***
https://igl.ethz.ch/projects/bbw/

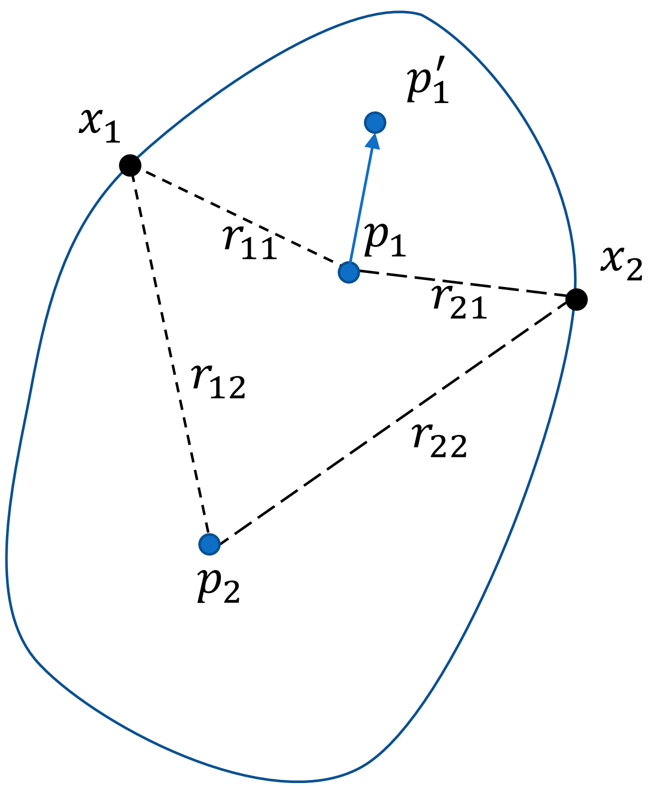
基于控制点的变形



$$x_i = \sum_j \omega_{ij}\left(x_i - p_j + p_j\right)$$

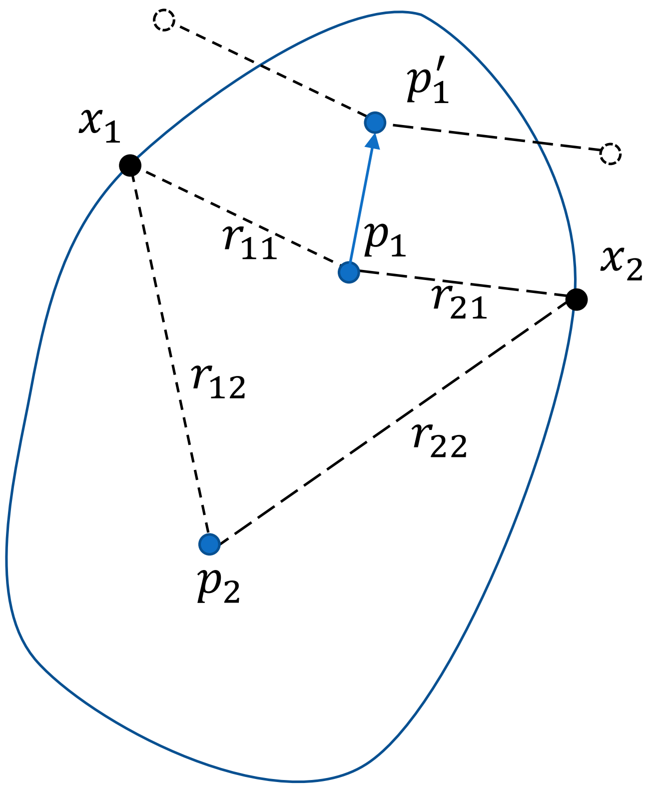
$$= \sum_j \omega_{ij}\left(r_{ij} + p_j\right)$$

基于控制点的变形



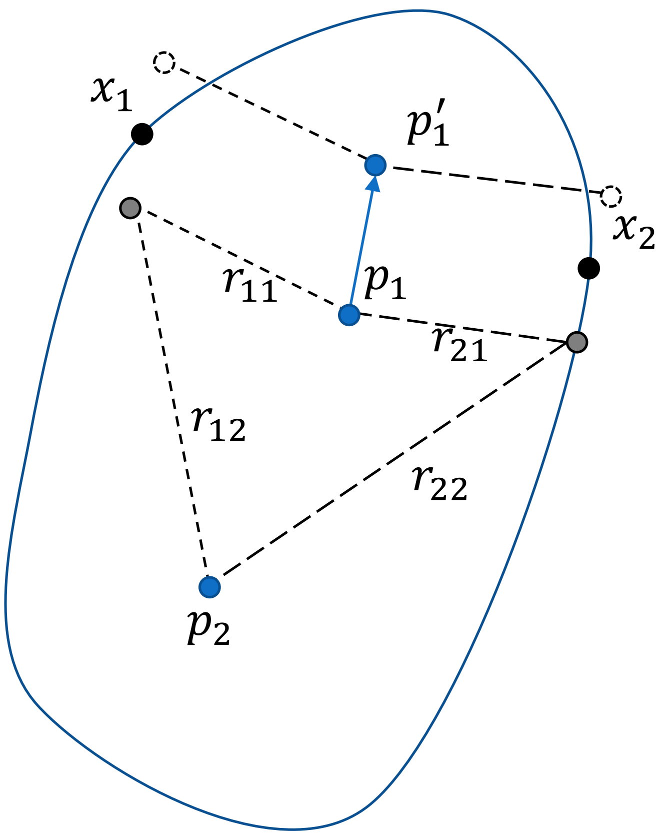
$$x_i' = \sum_j \omega_{ij}\left(r_{ij} + p_j'\right)$$

基于控制点的变形



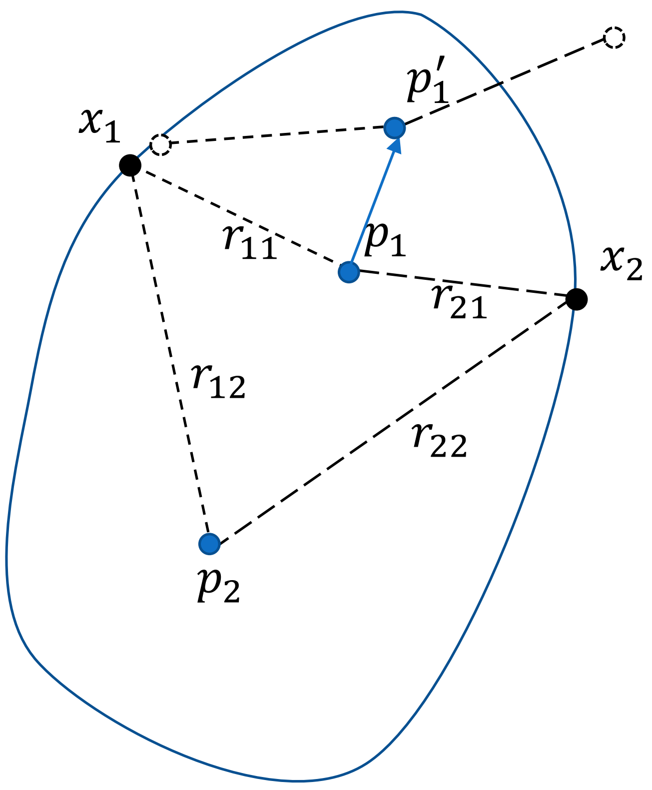
$$x_i' = \sum_j \omega_{ij}\left(r_{ij} + p_j'\right)$$

基于控制点的变形



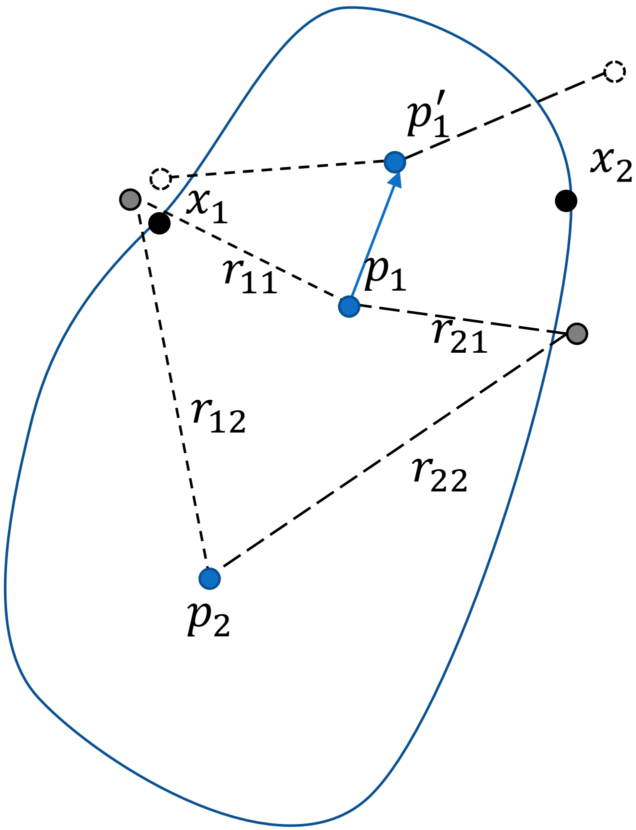
$$x_i' = \sum_j \omega_{ij}(r_{ij} + p_j')$$

基于控制点的变形



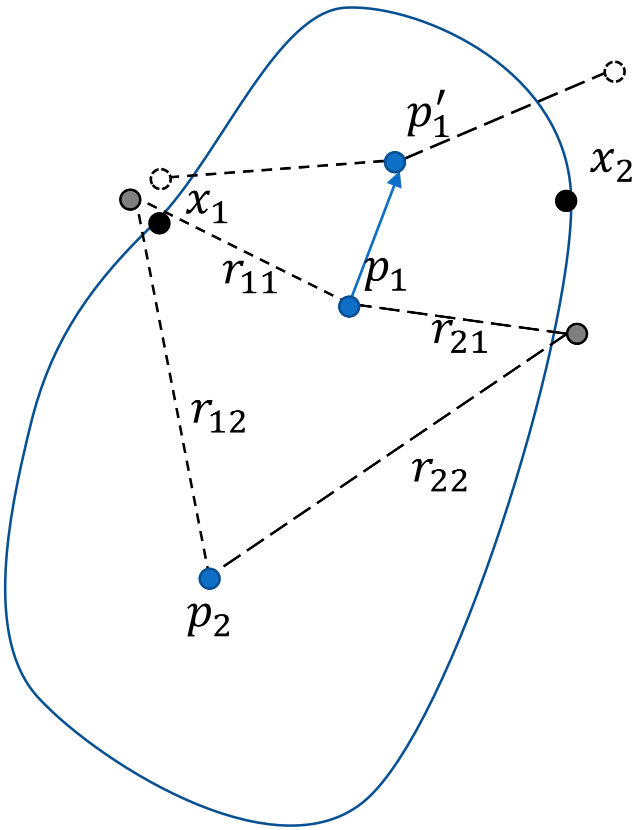
$$x_i' = \sum_j \omega_{ij}(R_j r_{ij} + p_j')$$

基于控制点的变形



$$x'_i = \sum_j \omega_{ij}\left(R_j r_{ij} + p'_j\right)$$

基于控制点的变形

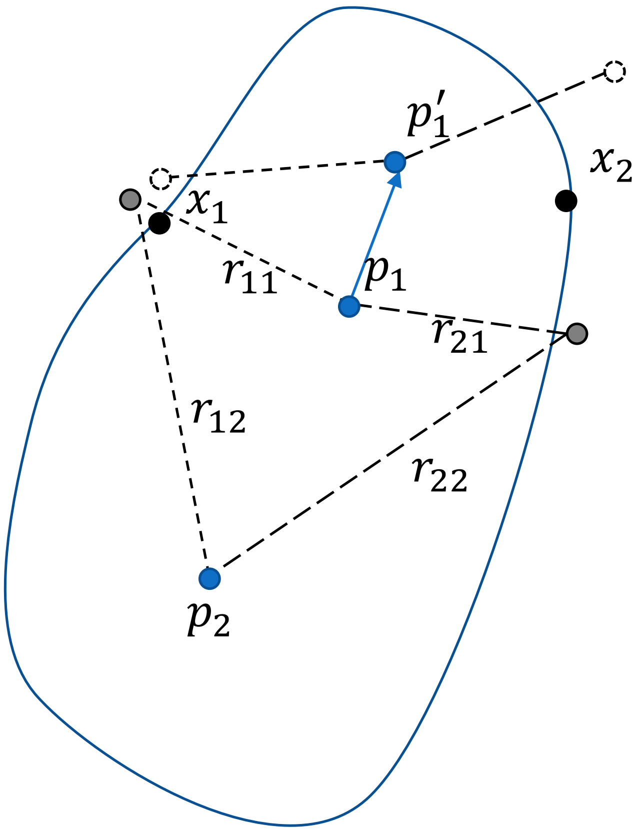


$$x_i' = \sum_j \omega_{ij}(R_j r_{ij} + p_j')$$

$$r_{ij} = x_i - p_j$$

$$x_i' = \sum_j \omega_{ij}(R_j x_i + t_j)$$

基于控制点的变形



$$x_i' = \sum_j \omega_{ij}(R_j r_{ij} + p_j')$$

$$r_{ij} = x_i - p_j$$

$$x_i' = \sum_j \omega_{ij}(R_j x_i + t_j)$$

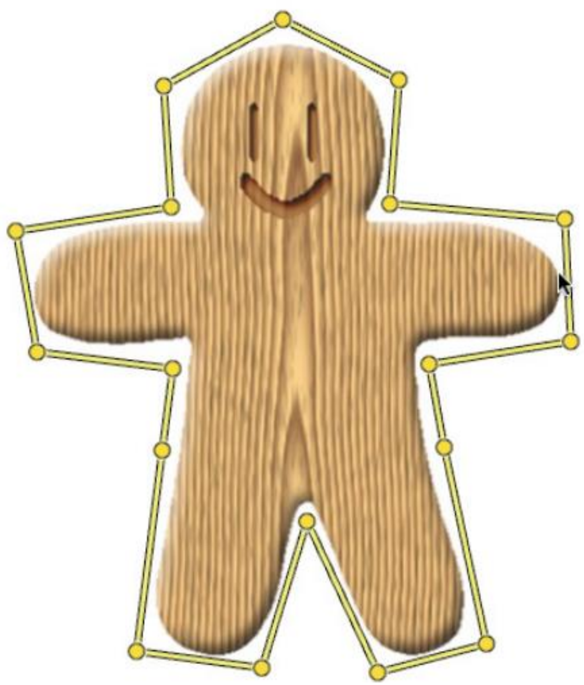
$$T = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} x_i' \\ 1 \end{pmatrix} = \sum_j \omega_{ij} T_j \begin{pmatrix} x_i \\ 1 \end{pmatrix}$$

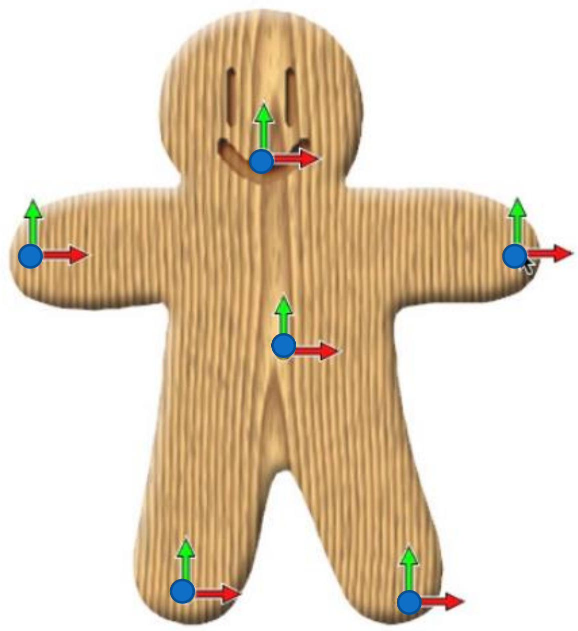
另一种形式的cage

$$x_i = \sum_j \omega_{ij} p_j$$

$$x_i' = \sum_j \omega_{ij} T_j x_i$$



Cage

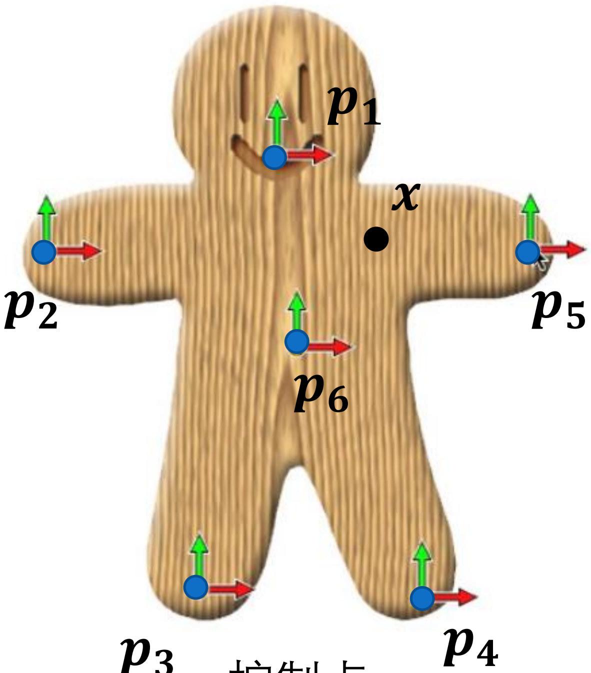


控制点

Alec Jacobson, Ilya Baran, Jovan Popović, Olga Sorkine-Hornung, ***Bounded Biharmonic Weights for Real-Time Deformation***
https://igl.ethz.ch/projects/bbw/

控制点权重的计算

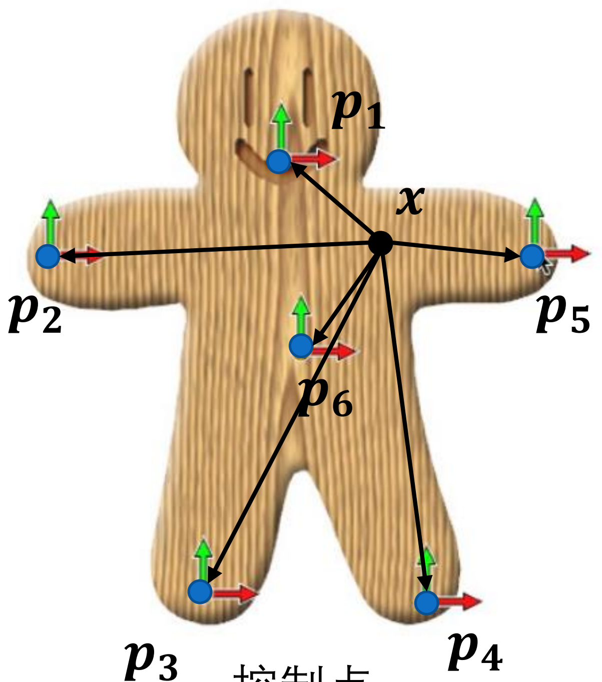
$$x_i' = \sum_j \omega_{ij} T_j x_i$$



控制点

控制点权重的计算

$$x_i' = \sum_j \omega_{ij} T_j x_i$$



控制点

基于距离的权重

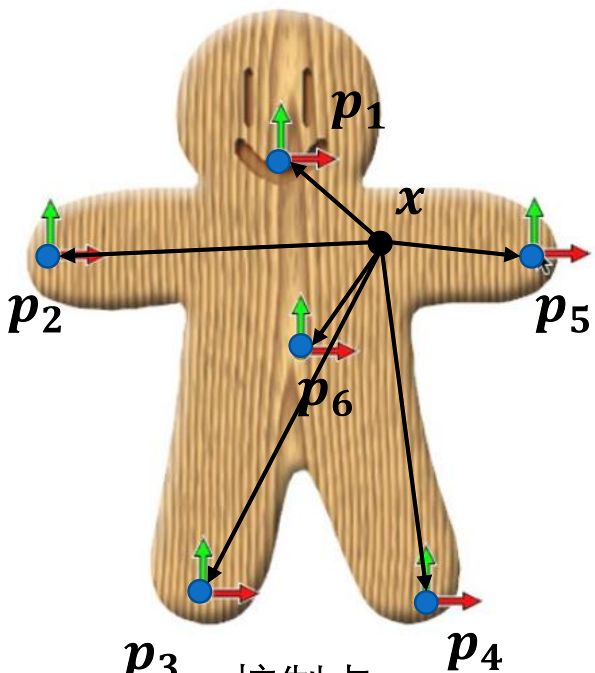
欧氏距离：$d_j = \|\boldsymbol{x} - \boldsymbol{p}_j\|$

权重：$w_{ij} = \begin{cases} 1, & d_j < d_k, \forall k \neq j \\ 0, & \text{otherwise} \end{cases}$

常用 但是不连续

控制点权重的计算

$$x_i' = \sum_j \omega_{ij} T_j x_i$$



控制点

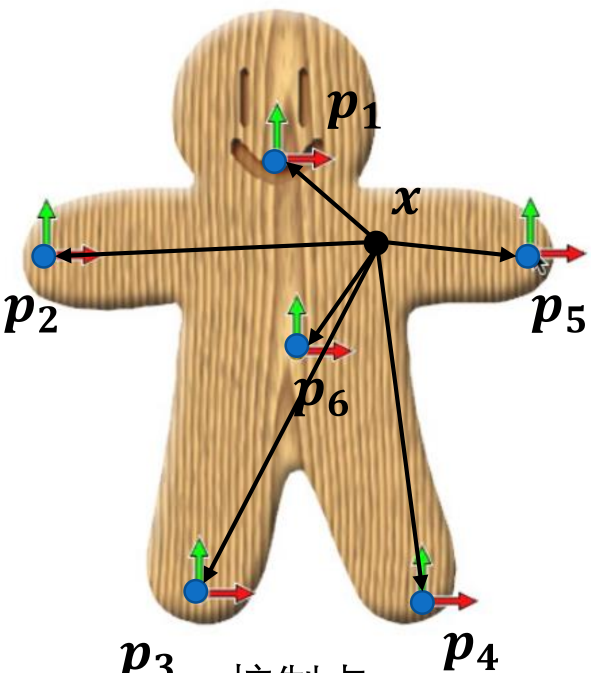
基于距离的权重

欧氏距离： $d_j = \|\boldsymbol{x} - \boldsymbol{p}_j\|$

权重： $w_{ij} = \frac{1}{d_j^p}$

控制点权重的计算

$$x_i' = \sum_j \omega_{ij} T_j x_i$$



控制点

基于距离的权重

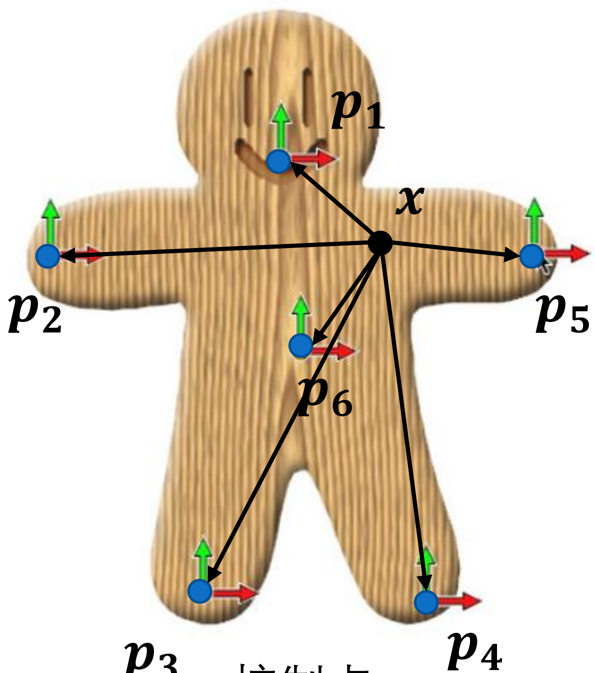
欧氏距离： $d_j = \left\|\boldsymbol{x} - \boldsymbol{p}_j\right\|$

权重： $\widehat{w}_{ij} = \dfrac{1}{d_j^p}$

$$w_{ij} = \frac{\widehat{w}_{ij}}{\sum_j \widehat{w}_{ij}}$$

控制点权重的计算

$$x_i' = \sum_j \omega_{ij} T_j x_i$$



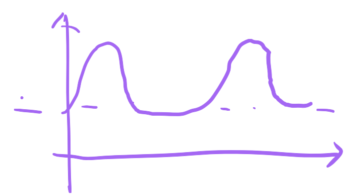
控制点

基于距离的权重

欧氏距离： $d_j = \|\boldsymbol{x} - \boldsymbol{p}_j\|$

权重： $\hat{w}_{ij} = \dfrac{1}{d_j^p + \varepsilon}$

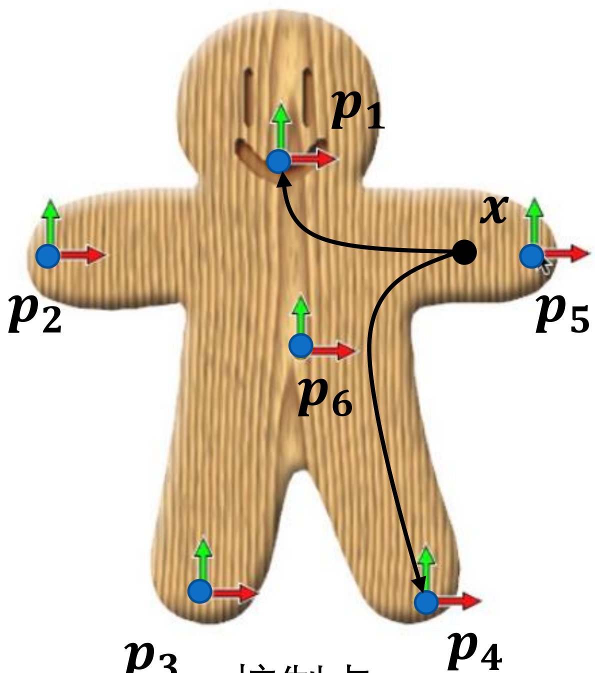
$$w_{ij} = \frac{\hat{w}_{ij}}{\sum_j \hat{w}_{ij}}$$



p越大，尖峰越陡
控制点的影响范围越小，实际中20即可

控制点权重的计算

$$x_i' = \sum_j \omega_{ij} T_j x_i$$



基于距离的权重

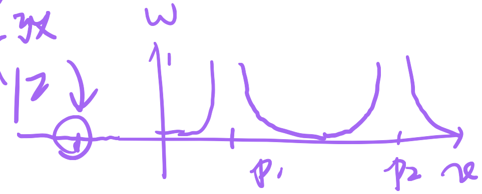
几何距离： $d_j = d(x, \boldsymbol{p}_i)$

权重： $\hat{w}_{ij} = \frac{1}{d_j^p + \varepsilon}$

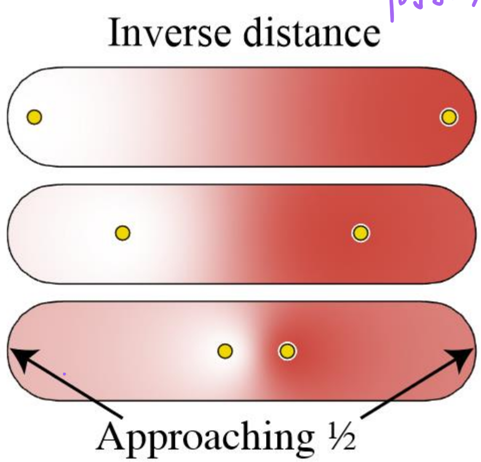
$$w_{ij} = \frac{\hat{w}_{ij}}{\sum_j \hat{w}_{ij}}$$

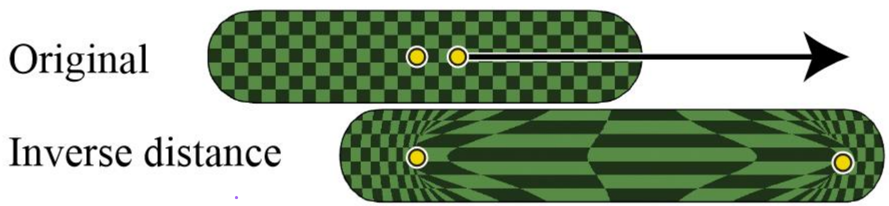
控制点权重的计算

归一后
接近一致
即为1/2



例如矩离





Alec Jacobson, Zhigang Deng, Ladislav Kavan, and J. P. Lewis. 2014. ***Skinning: real-time shape deformation****. In ACM SIGGRAPH 2014 Courses (SIGGRAPH '14)*

控制点权重的计算

- 基于能量优化的控制点权重
  - 权重在控制点处为1，平滑过渡到整个空间
  - 平滑能量函数

$w_i(p_i)=1,\quad i=1,\dots,m$

m个控制点

权重分布与热量传递相似

Common name	Least squares description	Energy	PDE
Dirichlet	f should be as constant as possible	$\int_\Omega \|\nabla f\|^2 dV$	$\Delta f = 0$
Laplacian	f should be as harmonic as possible	$\int_\Omega (\Delta f)^2 dV$	$\Delta^2 f = 0$
Laplacian gradient	Δf should be as constant as possible	$\int_\Omega \|\nabla \Delta f\|^2 dV$	$\Delta^3 f = 0$
$\dots$			

位置变化

速度变化

平滑性了

$f(x)=\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2}$

控制点权重的计算

Bounded Biharmonic Weights

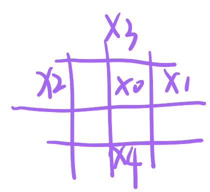
凸，有唯一解

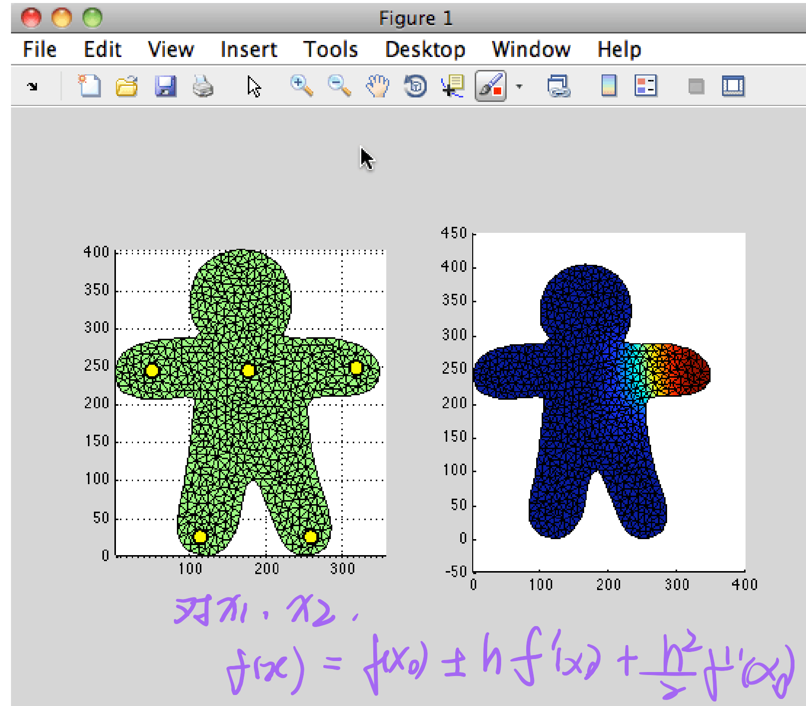
$$\underset{w_1,\ldots,w_m}{\operatorname{argmin}} \sum_{j=1}^{m} \frac{1}{2} \int_{\Omega} (\Delta w_j(\mathbf{v}))^2 \, dV,$$

$$\text{subject to } w_j(\mathbf{v}) = \varphi_j(\mathbf{v}) \; \forall \mathbf{v} \in H,$$

$$w_j(\mathbf{v}) > 0,$$

$$\sum_{j=1}^{m} w_j(\mathbf{v}) = 1.$$





对 x_1，x_2，

$f(x) = f(x_0) \pm h f'(x_0) + \frac{h^2}{2} f''(x_0)$

$f(x_1) + f(x_2) = 2f(x_0) + h^2 f''(x_0)$

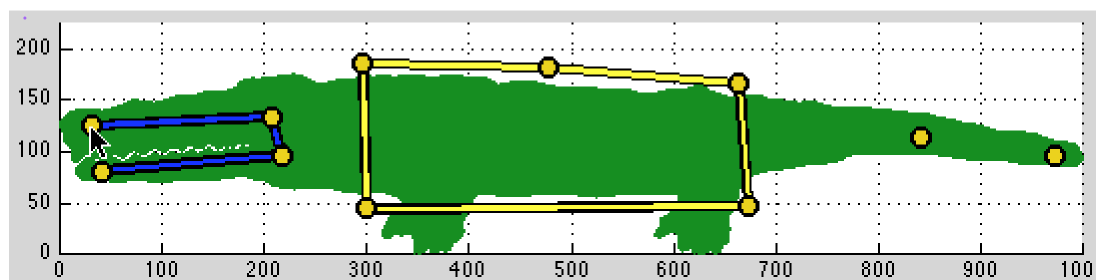
$\frac{\partial^2 f}{\partial x^2} = \frac{x_1 + x_2 - 2x_0}{2}$ $\qquad$ $\frac{\partial^2 f}{\partial y^2} = \frac{x_3 + x_4 - 2x_0}{2}$

$\Delta f(x_0) = \frac{1}{4}(x_1 + x_2 + x_3 + x_4) - x_0$

$= (-1, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}) \begin{bmatrix} x_0 \\ \vdots \\ x_4 \end{bmatrix}$

$\Delta f(x) = \begin{bmatrix} \; \end{bmatrix} \begin{bmatrix} x_0 \\ \vdots \\ x_n \end{bmatrix}$

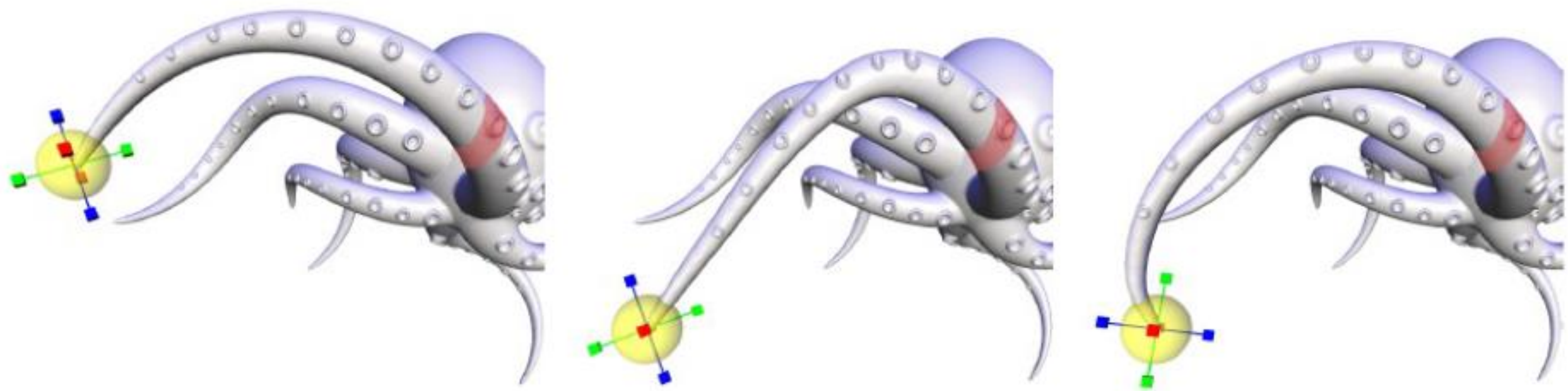
稀疏的矩阵



Alec Jacobson, Ilya Baran, Jovan Popović, and Olga Sorkine-Hornung. 2014. ***Bounded biharmonic weights for real-time deformation***. *Commun. ACM* 57, 4 (April 2014)

作为优化问题的几何形变

- 更加直接的形变编辑
  - 直接操作几何模型上的点
  - 将修改平滑地施加在整个几何模型上
  - 构造优化问题求解

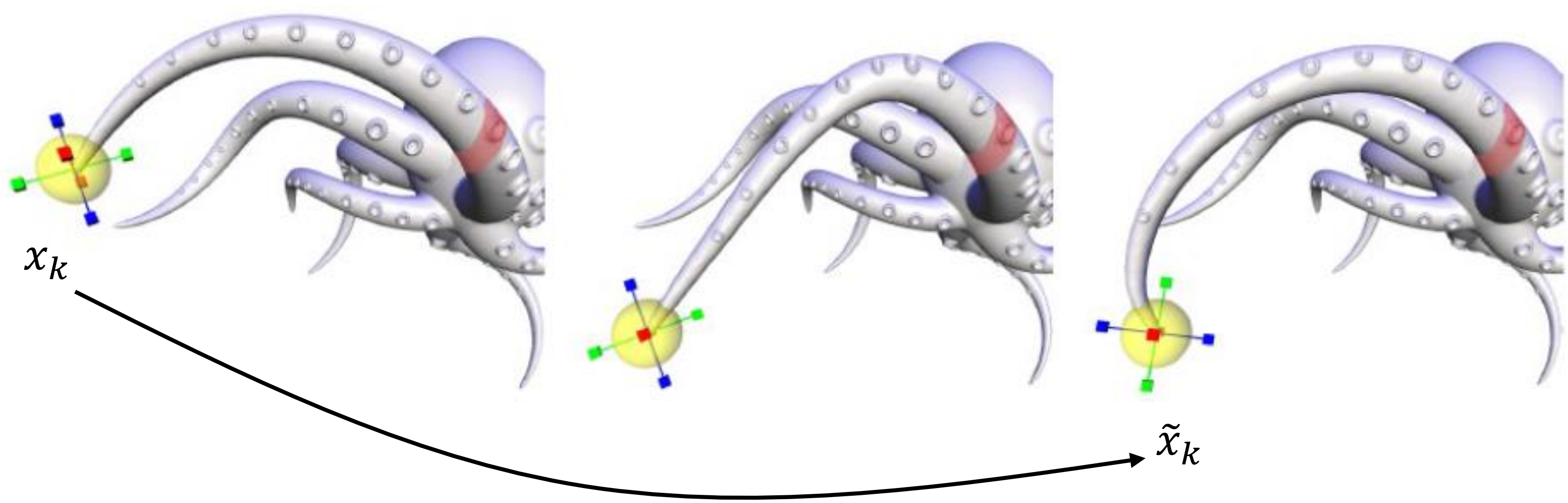


作为优化问题的几何形变

$$\min_{X'} \sum_k \|x_k' - \tilde{x}_k\|^2 + E(X, X')$$

编辑约束

平滑能量



作为优化问题的几何形变

$$\min_{X'} \sum_k \|x'_k - \tilde{x}_k\|^2 + E(X, X')$$

编辑约束

平滑能量

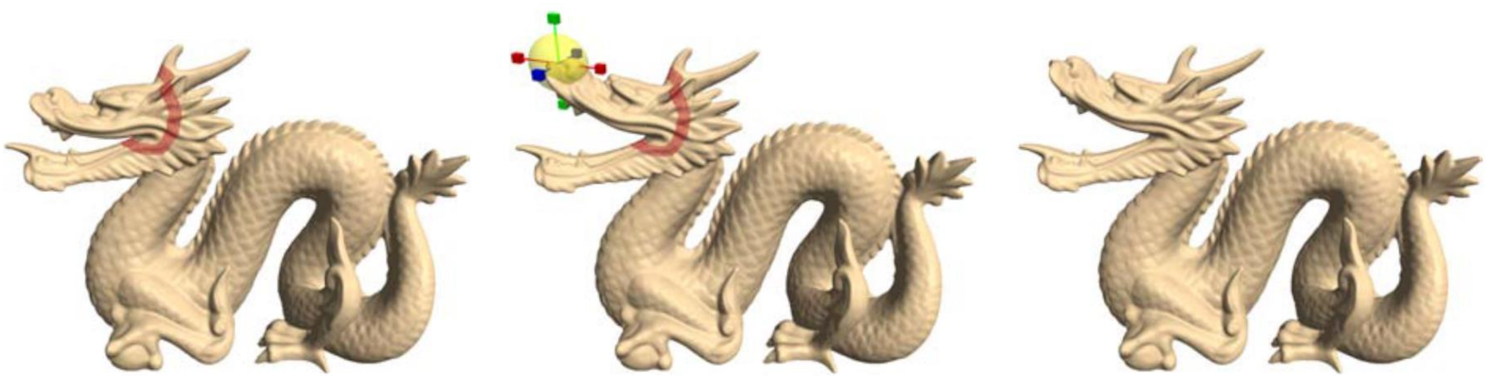
Common name	Least squares description	Energy	PDE
Dirichlet	f should be as constant as possible	$\int_\Omega \|\nabla f\|^2 dV$	$\Delta f = 0$
Laplacian	f should be as harmonic as possible	$\int_\Omega (\Delta f)^2 dV$	$\Delta^2 f = 0$
Laplacian gradient	Δf should be as constant as possible	$\int_\Omega \|\nabla \Delta f\|^2 dV$	$\Delta^3 f = 0$
...			

Laplacian Mesh Editing

编辑约束

平滑能量

$$\min_{X'} \sum_k \|x_k' - \tilde{x}_k\|^2 + \sum_i \|T_i(X')\mathcal{L}(x_i) - \mathcal{L}(x_i')\|^2$$



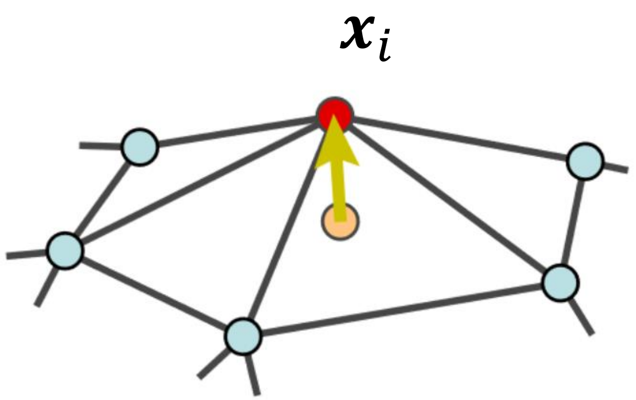
O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. 2004. ***Laplacian surface editing***. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing* (SGP '04), Association for Computing Machinery, New York, NY, USA, 175–184.

Laplacian坐标

$$\min_{X'} \sum_k \|x'_k - \tilde{x}_k\|^2 + \sum_i \|T_i(X')\mathcal{L}(x_i) - \mathcal{L}(x'_i)\|^2$$

$$\mathcal{L}(x) = \Delta \boldsymbol{f}(x)$$

Laplace-Beltrami operator



$\mathcal{N}_i$: 1-neighbors

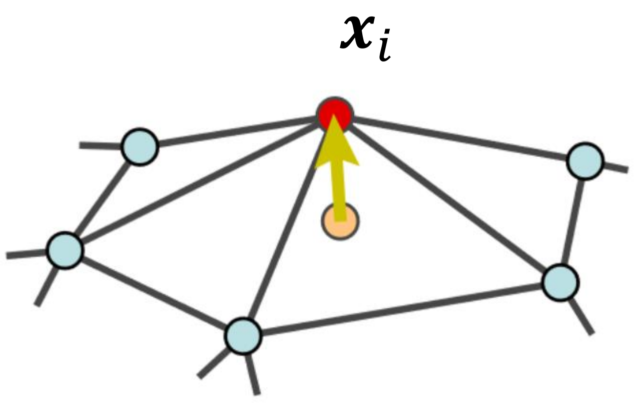
Laplacian坐标

$$\min_{X'} \sum_k \|x'_k - \tilde{x}_k\|^2 + \sum_i \|T_i(X')\mathcal{L}(x_i) - \mathcal{L}(x'_i)\|^2$$

$$\mathcal{L}(\boldsymbol{x}_\mathrm{i}) = \boldsymbol{x}_i - \frac{1}{d_i}\sum_{j\in\mathcal{N}_i} \boldsymbol{x}_j$$

“Umbrella” operator

可以告知某个地方是否有尖点，说明一个平面的光滑性



$\mathcal{N}_i$: 1-neighbors

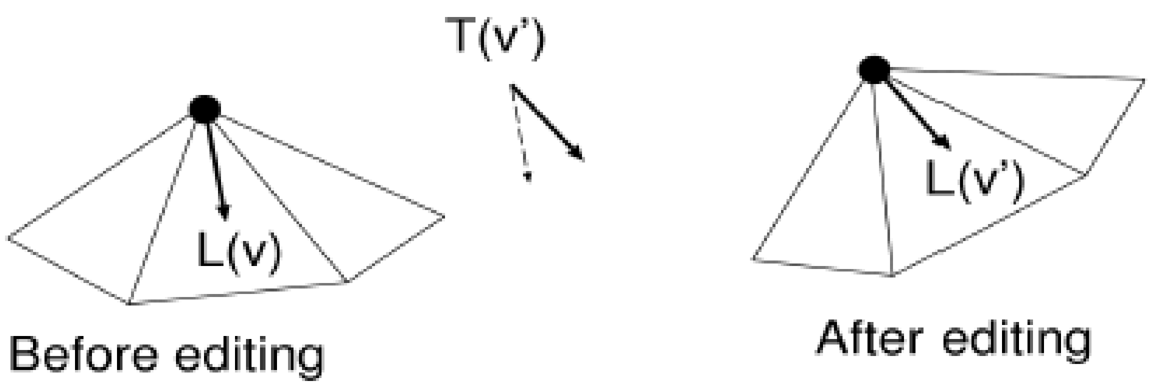
一阶邻居

Laplacian坐标

$$\min_{X'} \sum_k \|x_k' - \tilde{x}_k\|^2 + \sum_i \|T_i(X')\mathcal{L}(x_i) - \mathcal{L}(x_i')\|^2$$

- 我们希望变换过后每个点的Laplacian坐标不变

维持平滑性



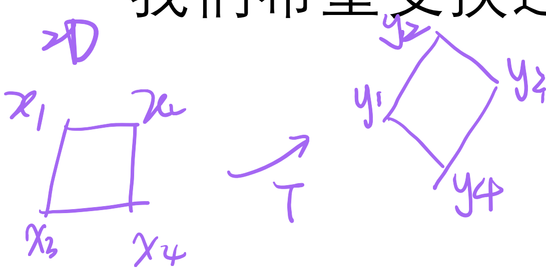
Laplacian坐标

$$\min_{X', T_i} \sum_k \|x'_k - \tilde{x}_k\|^2 + \sum_i \|T_i(X')\mathcal{L}(x_i) - \mathcal{L}(x'_i)\|^2$$

T的求解：

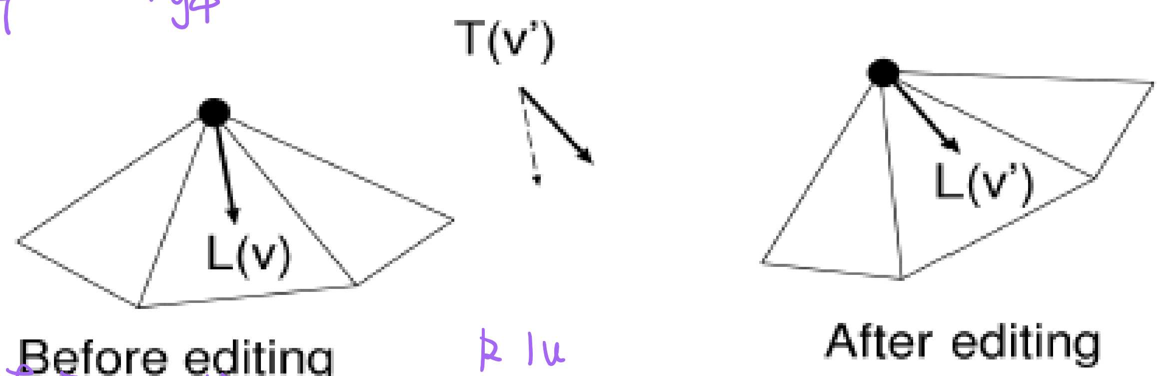
T_i用来变换方向，通过迭代求解。先固定估计一个T，再对x求优化，再估计T……

- 我们希望变换过后每个点的Laplacian坐标不变



$Tx_i \to y_i$

$T\begin{bmatrix}x_i^1\\x_i^2\\1\end{bmatrix} = \begin{bmatrix}y_i^1\\y_i^2\end{bmatrix}$



k lu

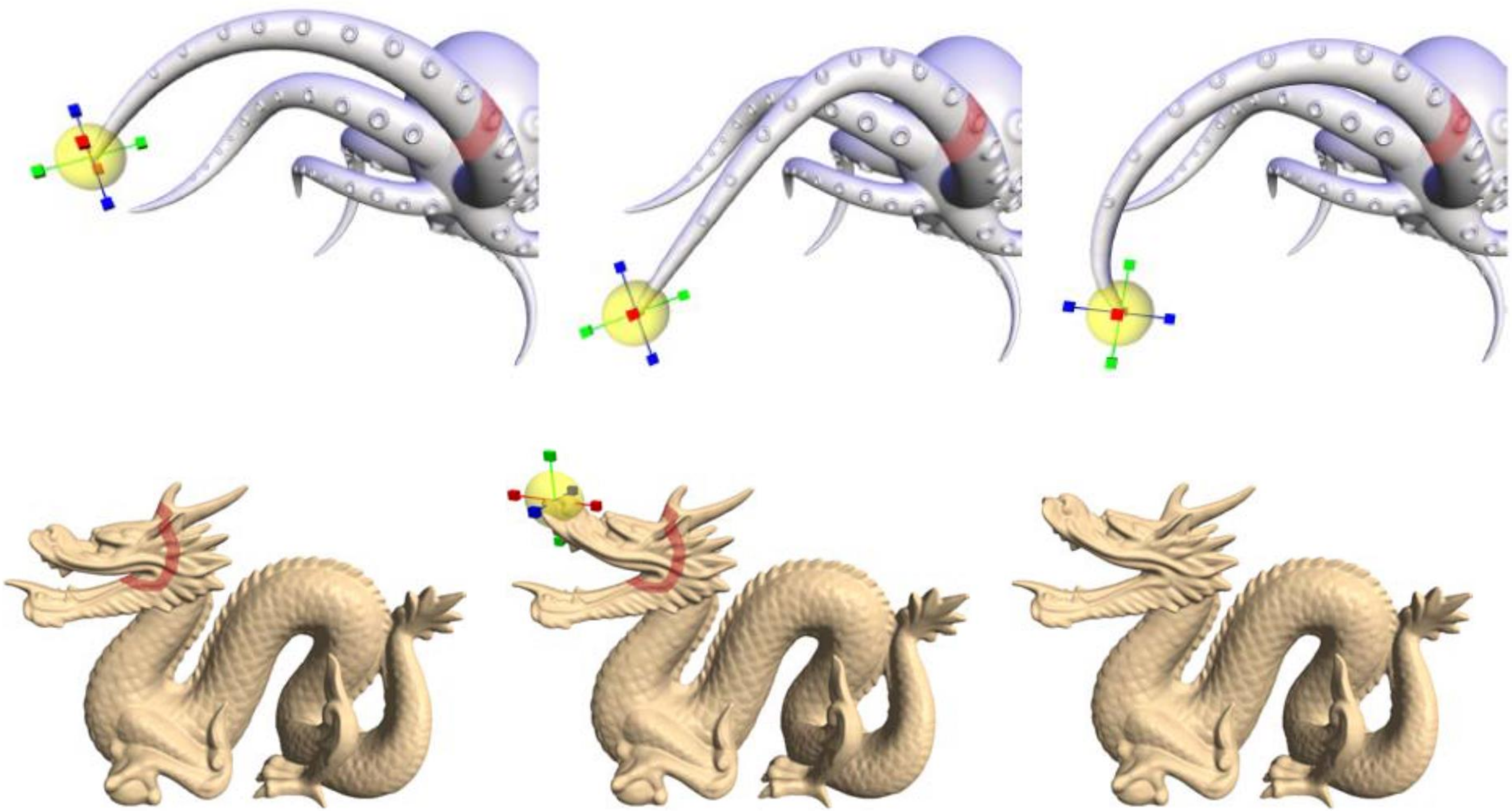
$$\begin{bmatrix} x_i^1 & x_i^2 & 0 & 1 & 0 & 0 \\ 0 & 0 & x_i^1 & x_i^2 & 0 & 1 \\ & & \cdots & & & \end{bmatrix}\begin{bmatrix}t_1\\t_2\\t_3\\t_4\\v_1\\v_2\end{bmatrix} = \begin{bmatrix}y_i^1\\y_i^2\\ \vdots\end{bmatrix}$$

$T = \begin{bmatrix} t_1, & t_2, & v_1 \\ t_3, & t_4, & v_2 \end{bmatrix}$

合在一起是一个最小二乘问题（过约束），左边为一个8×6矩阵

Laplacian Mesh Editing

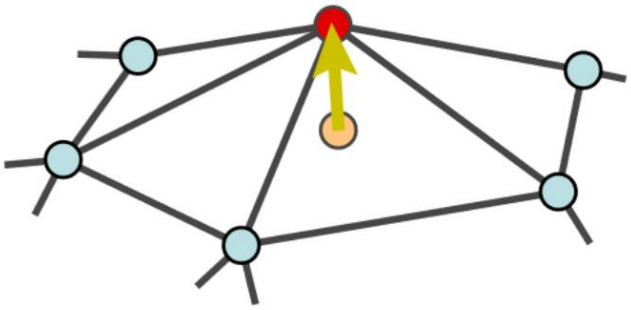
3个点控制一个旋转

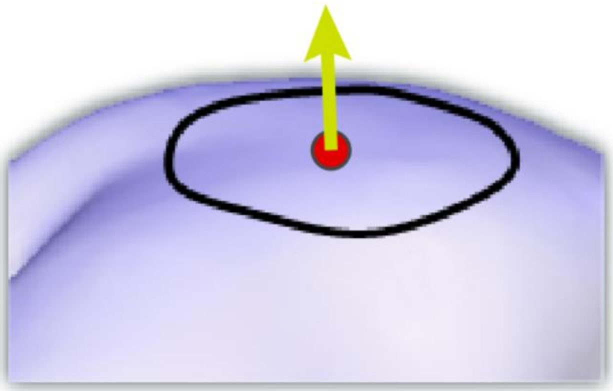


O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. 2004. ***Laplacian surface editing***. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing* (SGP '04), Association for Computing Machinery, New York, NY, USA, 175–184.

Laplacian坐标的含义

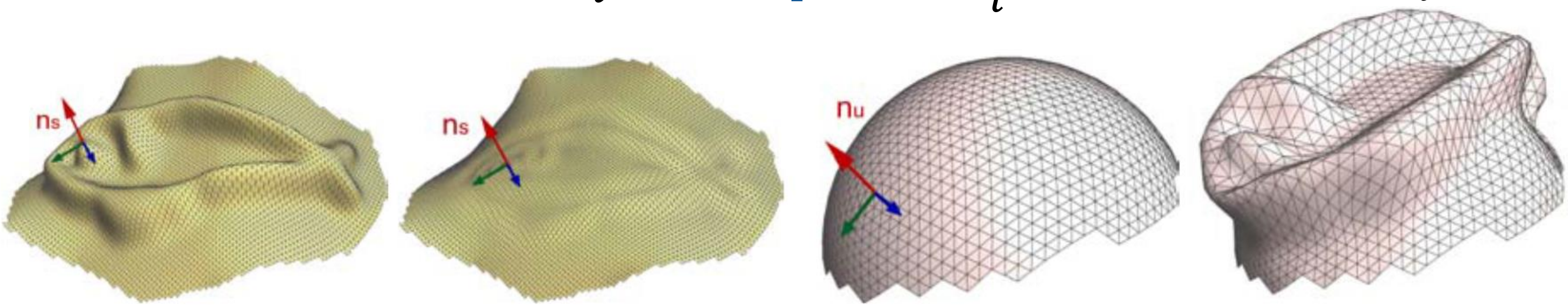
- 几何表面**局部**的细节
  - 方向 ≈ 法向
  - 大小 ≈ 曲率



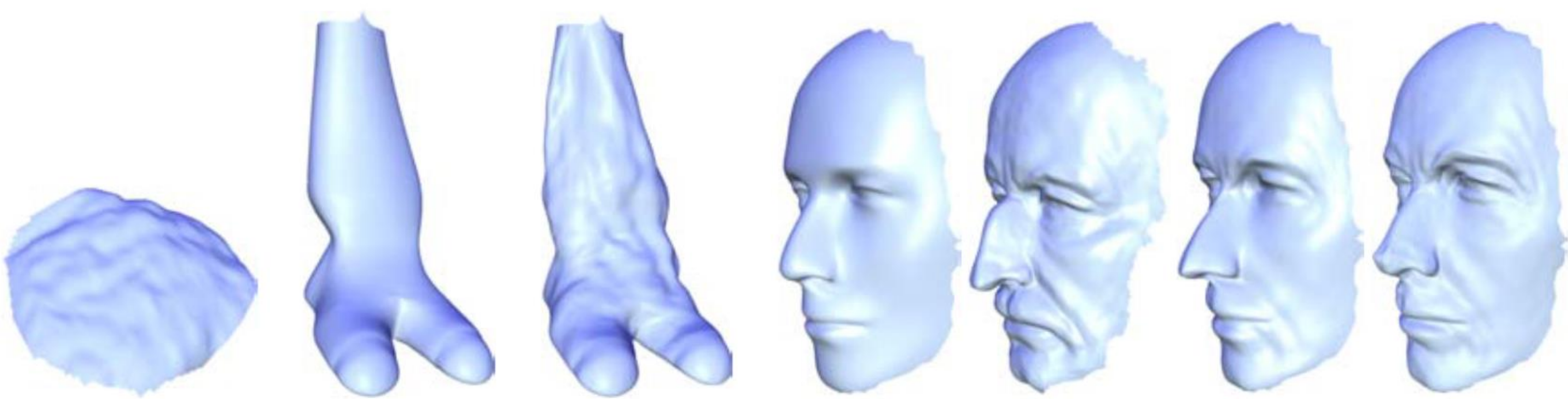


Coating Transfer

$$\delta_i = \mathcal{L}(x_i) - \hat{\delta}_i + \hat{\delta}'_i = \delta'_i$$

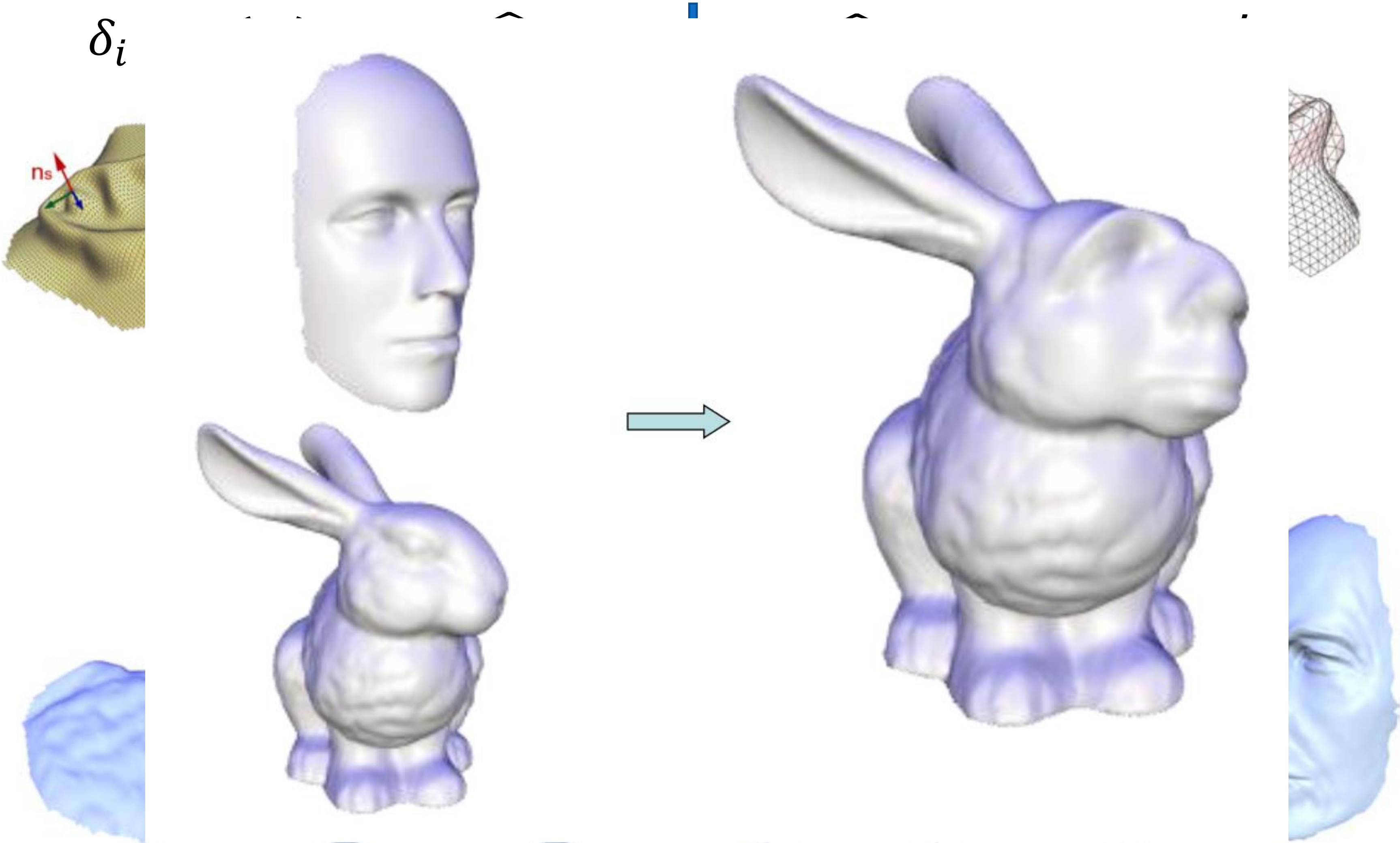


(a) (b) (c) (d)



Coating Transfer

δ_i



As Rigid As Possible Deformation

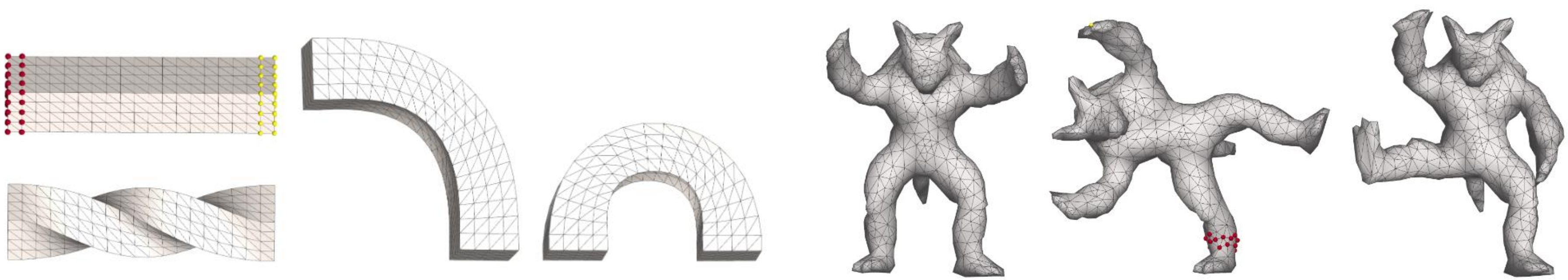
编辑约束

$$\min_{X'} \sum_k \|x_k' - \tilde{x}_k\|^2 + E(X, X')$$

平滑能量项

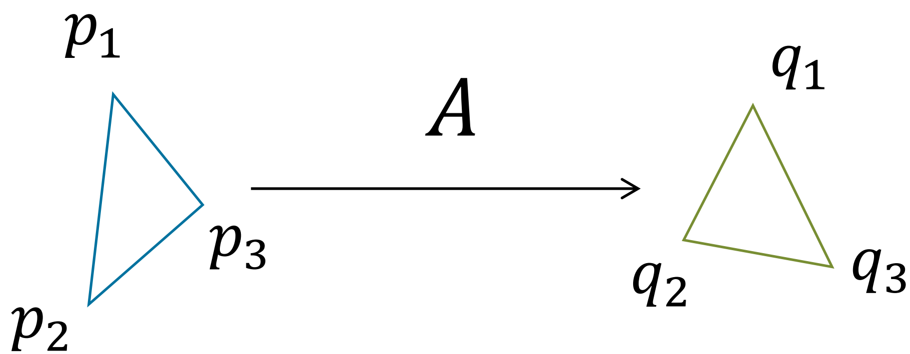
刚性旋转 $T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$

尽可能减少非“刚性”形变 ➔ 减少不必要的缩放 $\det(T) = 1$



Olga Sorkine and Marc Alexa. 2007. ***As-rigid-as-possible surface modeling***. In *Proceedings of the fifth Eurographics symposium on Geometry processing* (SGP ’07), Eurographics Association, Goslar, DEU, 109–116.

As Rigid As Possible Deformation



$$\boldsymbol{q}_i = A\boldsymbol{p}_i + \boldsymbol{t}$$

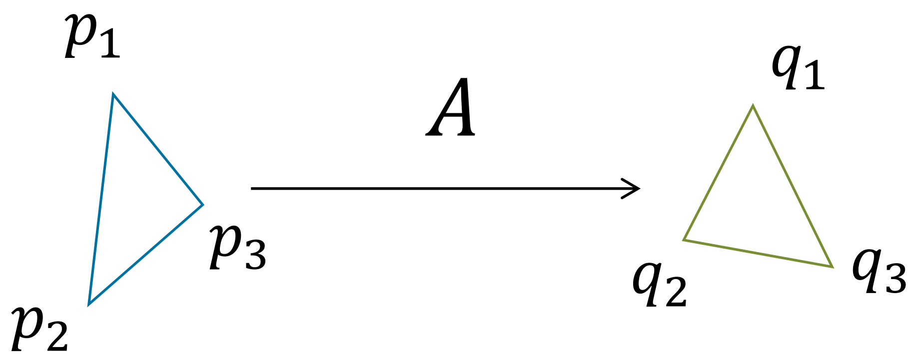
刚性旋转、平移 + 拉伸

$$\min_{R,t} \sum_{i=1}^{3} \| R p_i + t - q_i \|^2$$

$$\text{s.t. } R^T R = 1$$

$$A = U\Sigma V^T = (UV^T)(VDV^T) = RS$$

As Rigid As Possible Shape Interpolation



Interpolation between two shapes

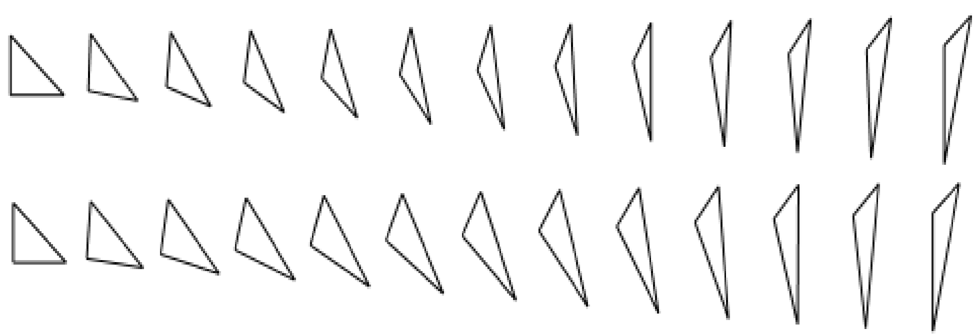
$$\boldsymbol{q}_i(t) = \mathrm{A}(t)\boldsymbol{p}_i + \boldsymbol{t}$$

R、S 分别插值，S 正定且对称

$$\mathrm{A}(t) = R(t)\big((1-t)I + tS\big)$$

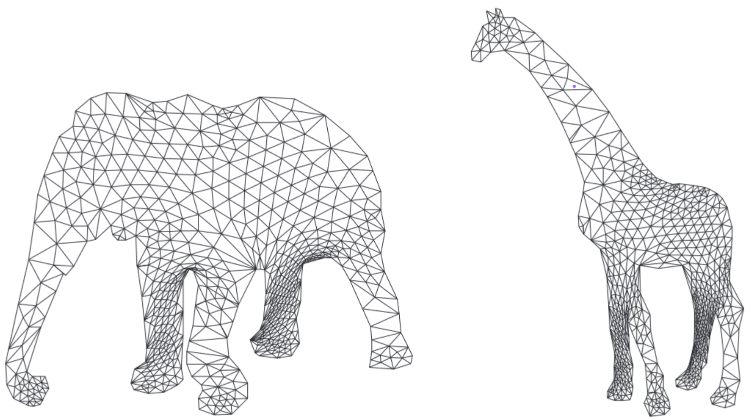
顶点线性插值

刚性变换插值



As Rigid As Possible Shape Interpolation

多个三角形通过之前的计算方式可能会分开，但是可以通过优化所有的顶点的坐标值，使得每个三角形的变换与插值得到的R、S相近，这样就可以保持一个完整的形状。这个问题实际上也是一个二次问题



$$\min_{X_i} = \sum_{i=1}^{N} \| A(x_i) - R_t S_t \|_F^2$$

Marc Alexa, Daniel Cohen-Or, and David Levin. 2000. ***As-rigid-as-possible shape interpolation***. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (SIGGRAPH '00)

As Rigid As Possible Deformation

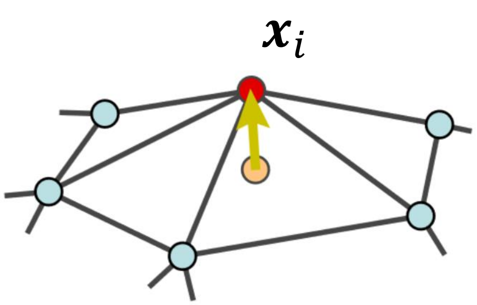
编辑约束

$$\min_{X'} \sum_k \|x'_k - \tilde{x}_k\|^2 + E(X, X')$$

平滑能量项

尽可能减少非"刚性"形变 ➔ 减少不必要的缩放

$$E(X, X') = \sum_{j \in \mathcal{N}_i} w_{ij} \left\| (x'_i - x'_j) - R_i (x_i - x_j) \right\|^2$$



$\mathcal{N}_i$: 1-neighbors

As Rigid As Possible Deformation

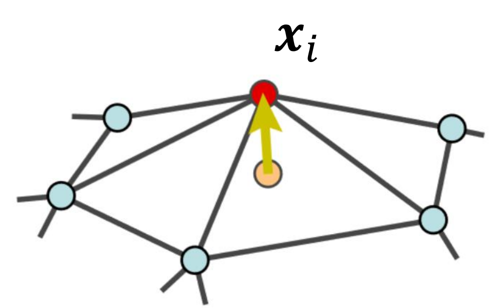
编辑约束

$$\min_{X',\boldsymbol{R}_i} \sum_k \|x'_k - \tilde{x}_k\|^2 + E(X, X')$$

平滑能量项

尽可能减少非“刚性”形变 ➔ 减少不必要的缩放

$$E(X, X') = \sum_{j \in \mathcal{N}_i} w_{ij} \left\| \left(x'_i - x'_j\right) - R_i\left(x_i - x_j\right) \right\|^2$$



$\mathcal{N}_i$: 1-neighbors

As Rigid As Possible Deformation

尽可能的保持了体积不变

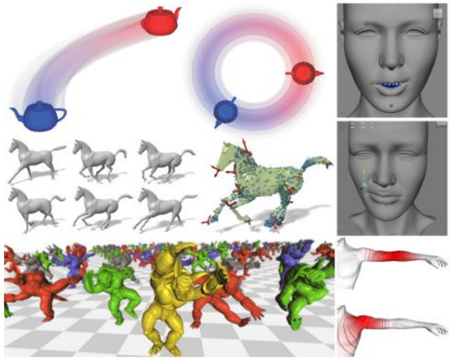
Olga Sorkine and Marc Alexa. 2007. ***As-rigid-as-possible surface modeling***. In *Proceedings of the fifth Eurographics symposium on Geometry processing* (SGP ’07), Eurographics Association, Goslar, DEU, 109–116.

蒙皮变形

Skinning

What is covered

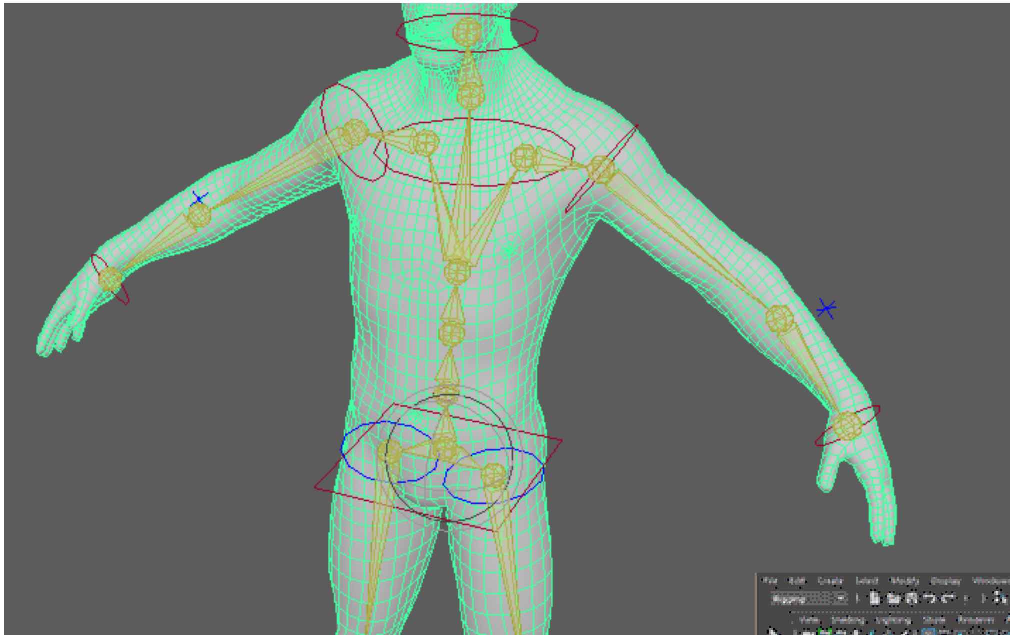
- How to deform a mesh according to skeletal animation?
  - Linear Blend Skinning (LBS)
    - Multi-linear Skinning
  - Non-linear Skinning
    - Dual-Quaternion Skinning (DQS)
- How to compute weights?
- Example-based methods?

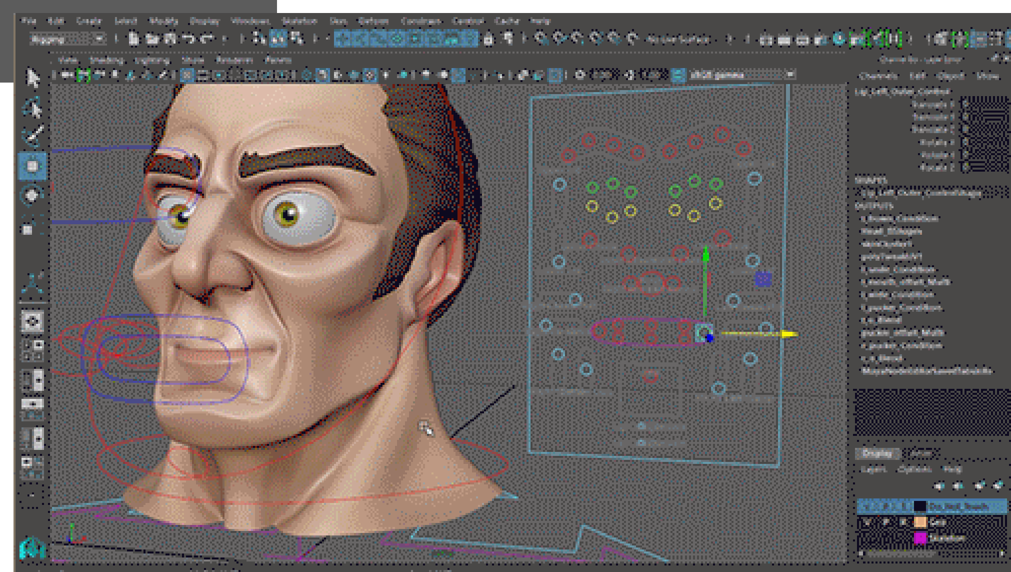


Many images are from: https://skinning.org/

Alec Jacobson, Zhigang Deng, Ladislav Kavan, and J. P. Lewis. 2014. ***Skinning: real-time shape deformation****. In ACM SIGGRAPH 2014 Courses (SIGGRAPH '14)*

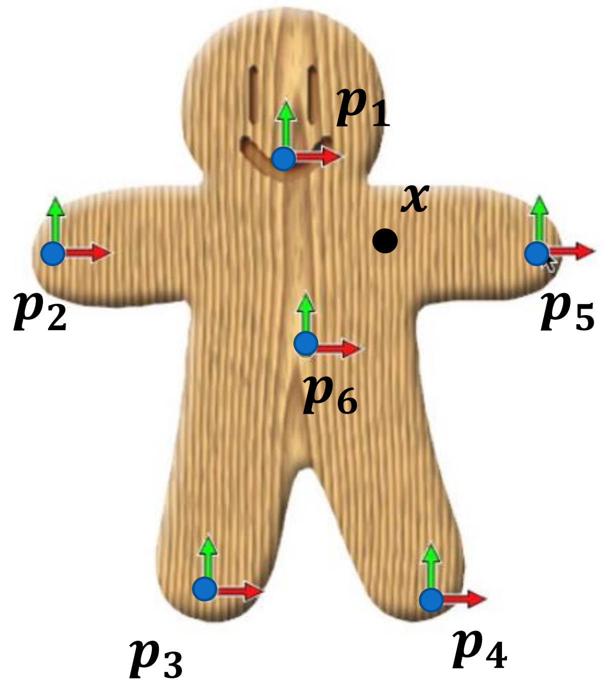
Skinning





Recall: deformation with control points

$$x_i' = \sum_j \omega_{ij} T_j x_i$$

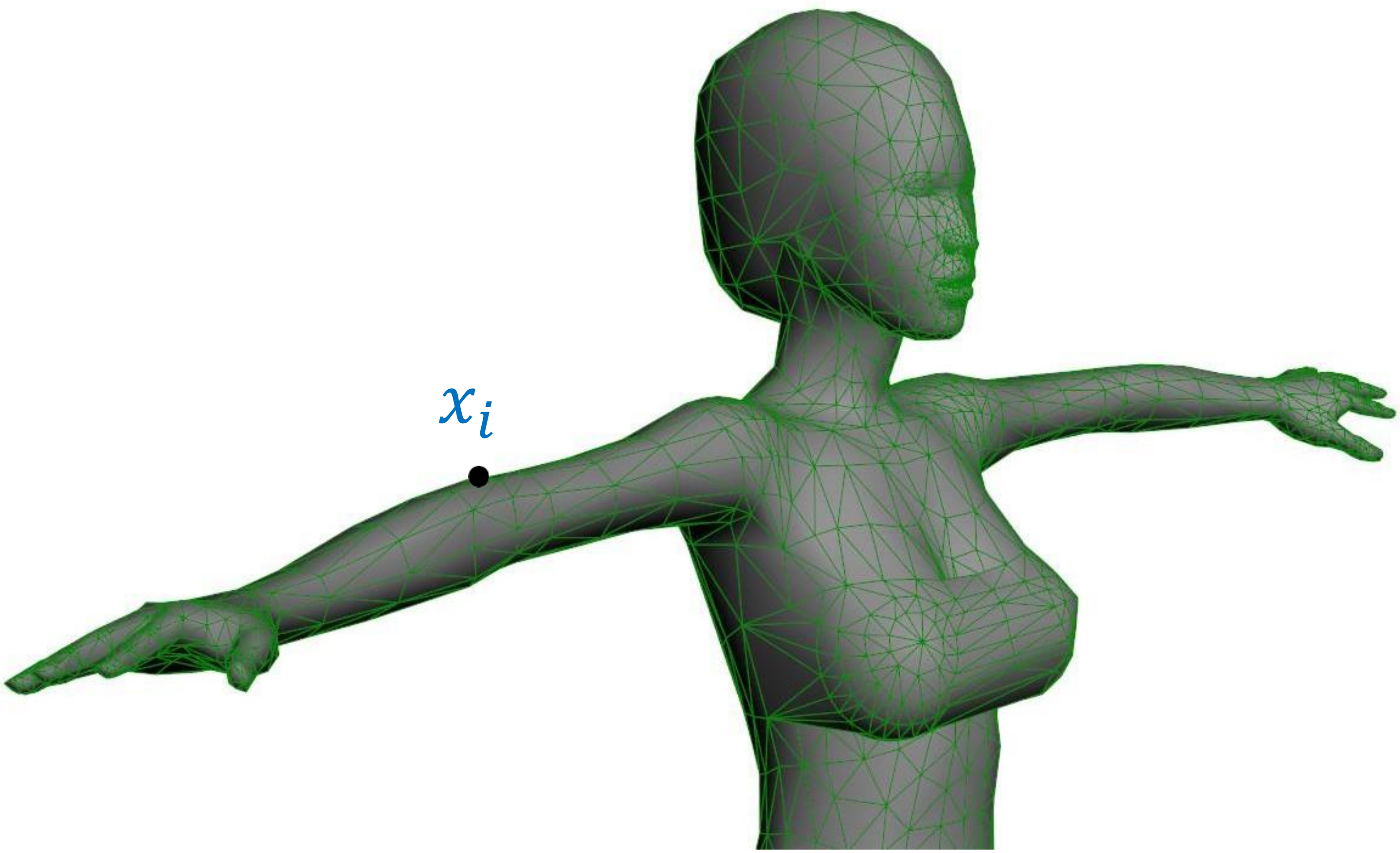


Basic Setup

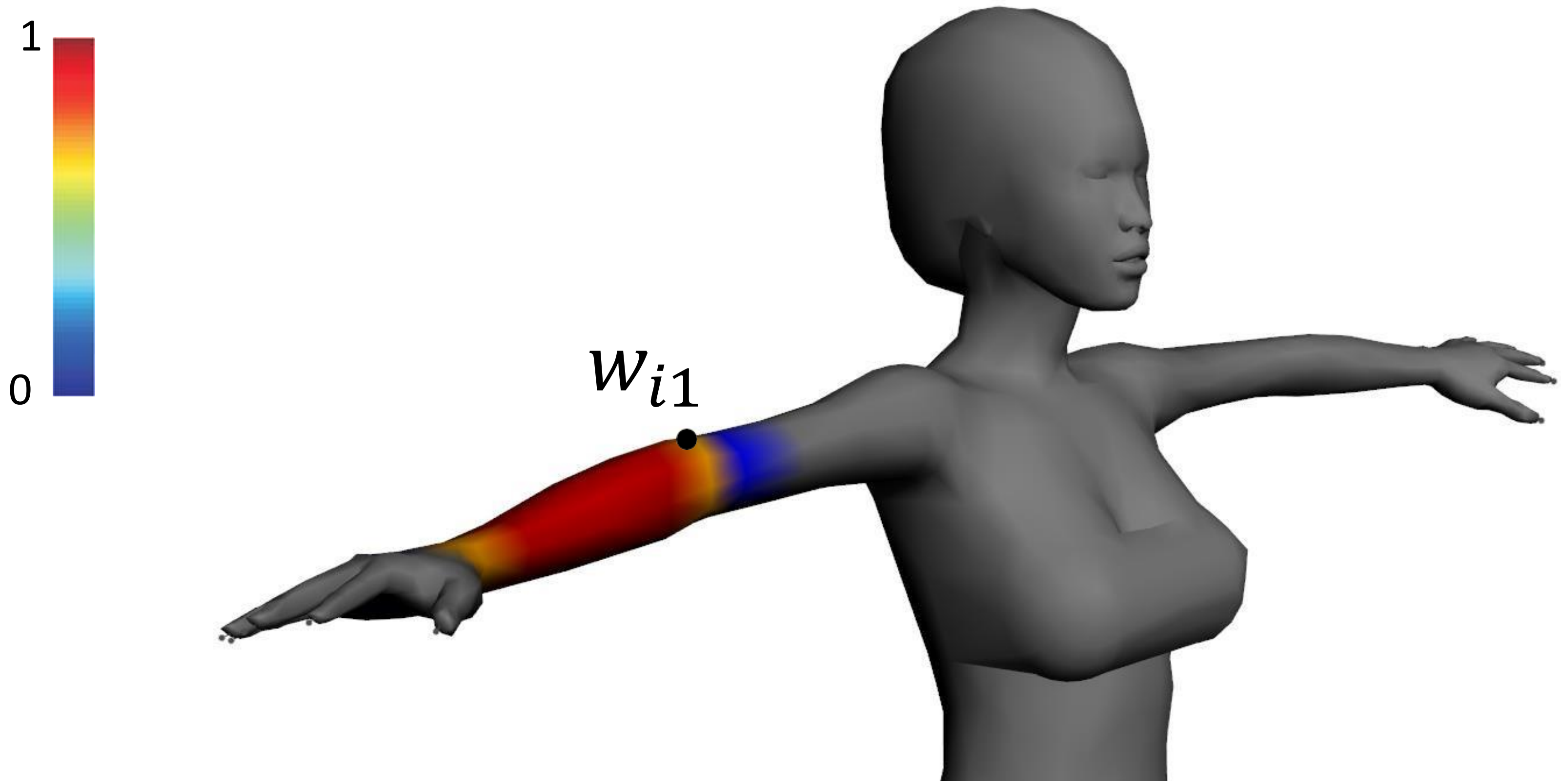
- Rest pose x_0
- Skinning weights ω
- Skinning transformation T

$$x_i' = \sum_{j=1}^{m} \omega_{ij} T_j x_i$$

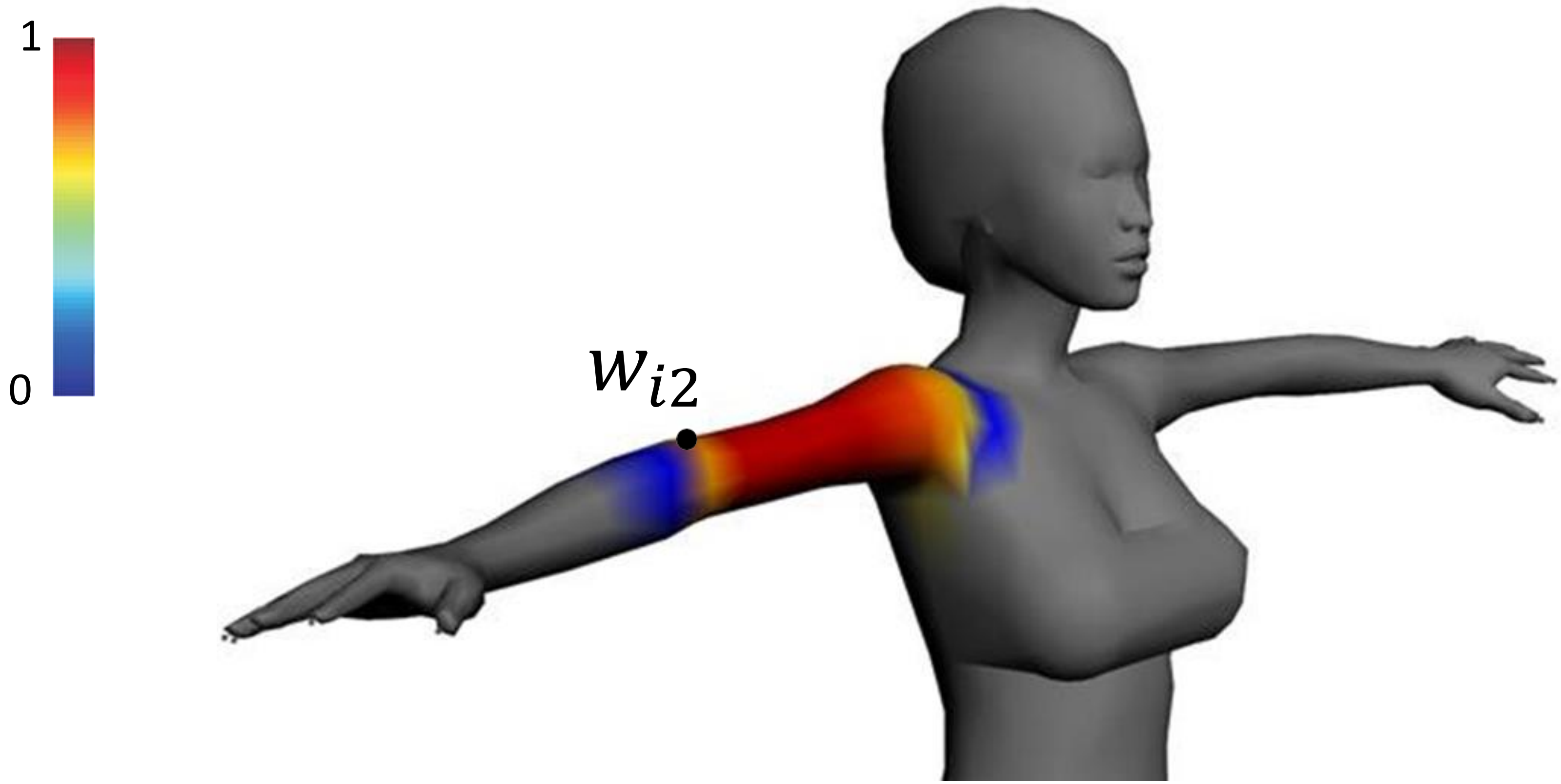
Basic Setup: Rest Pose/Bind Pose



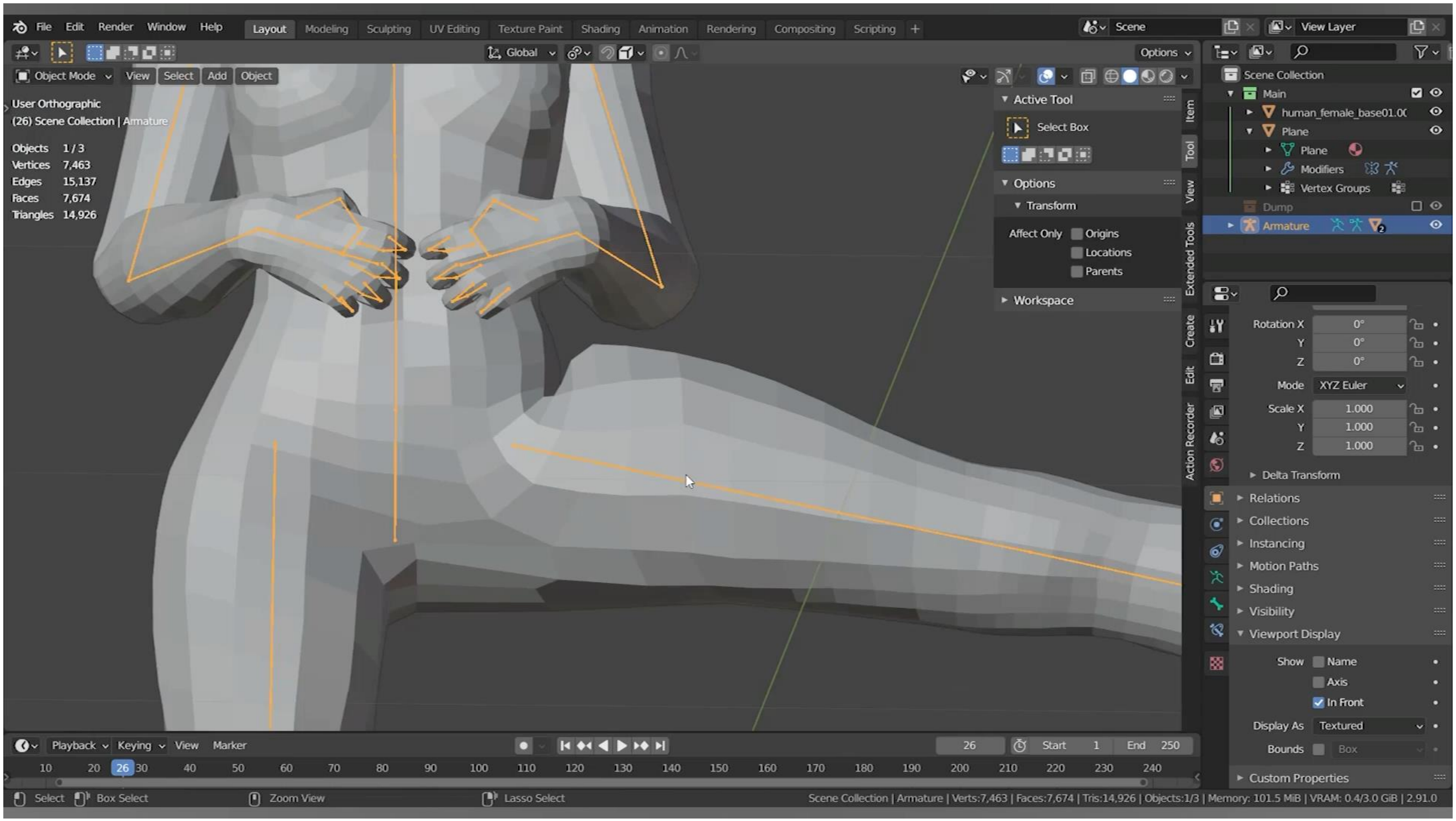
Basic Setup: Skinning Weights



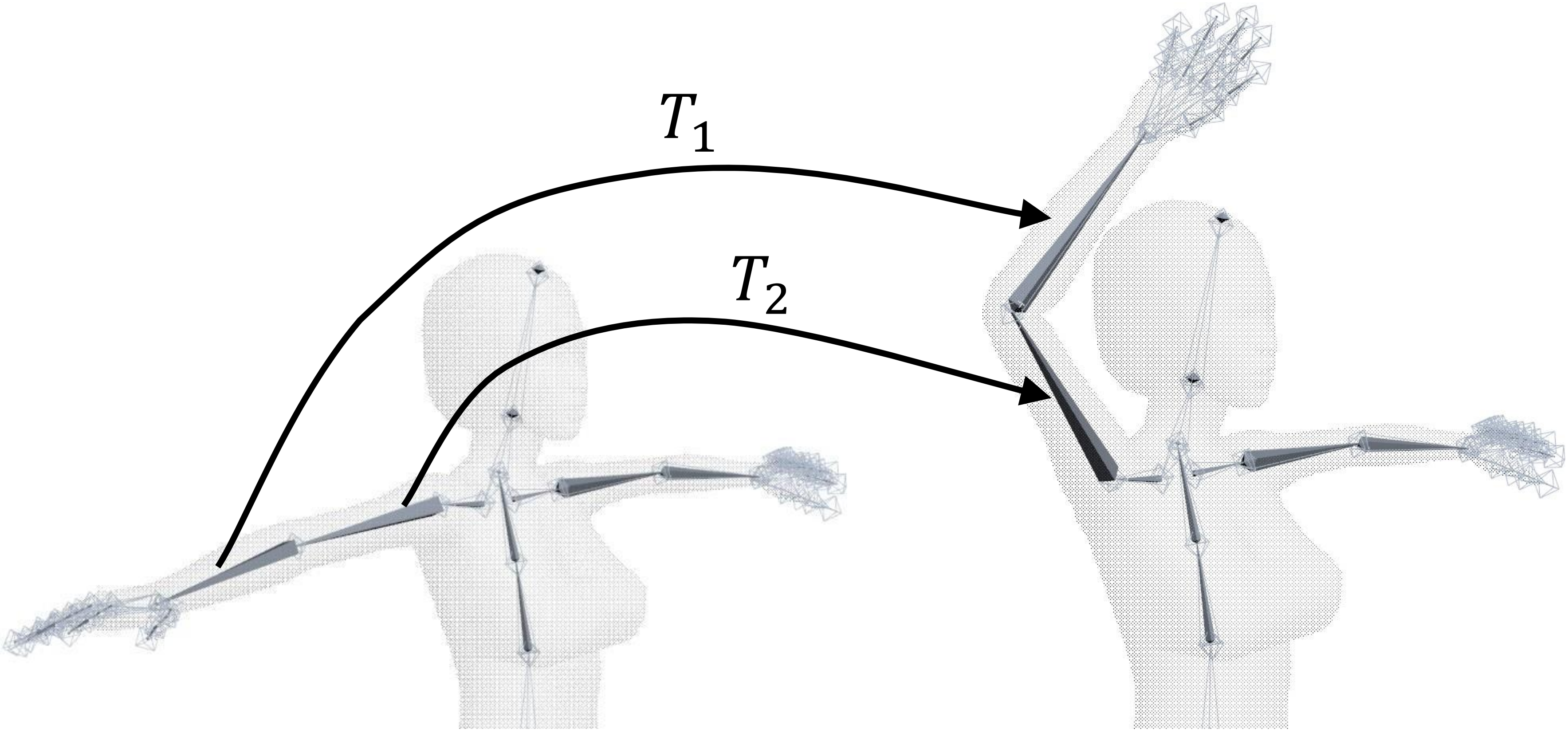
Basic Setup: Skinning Weights



Basic Setup: Skinning Weights

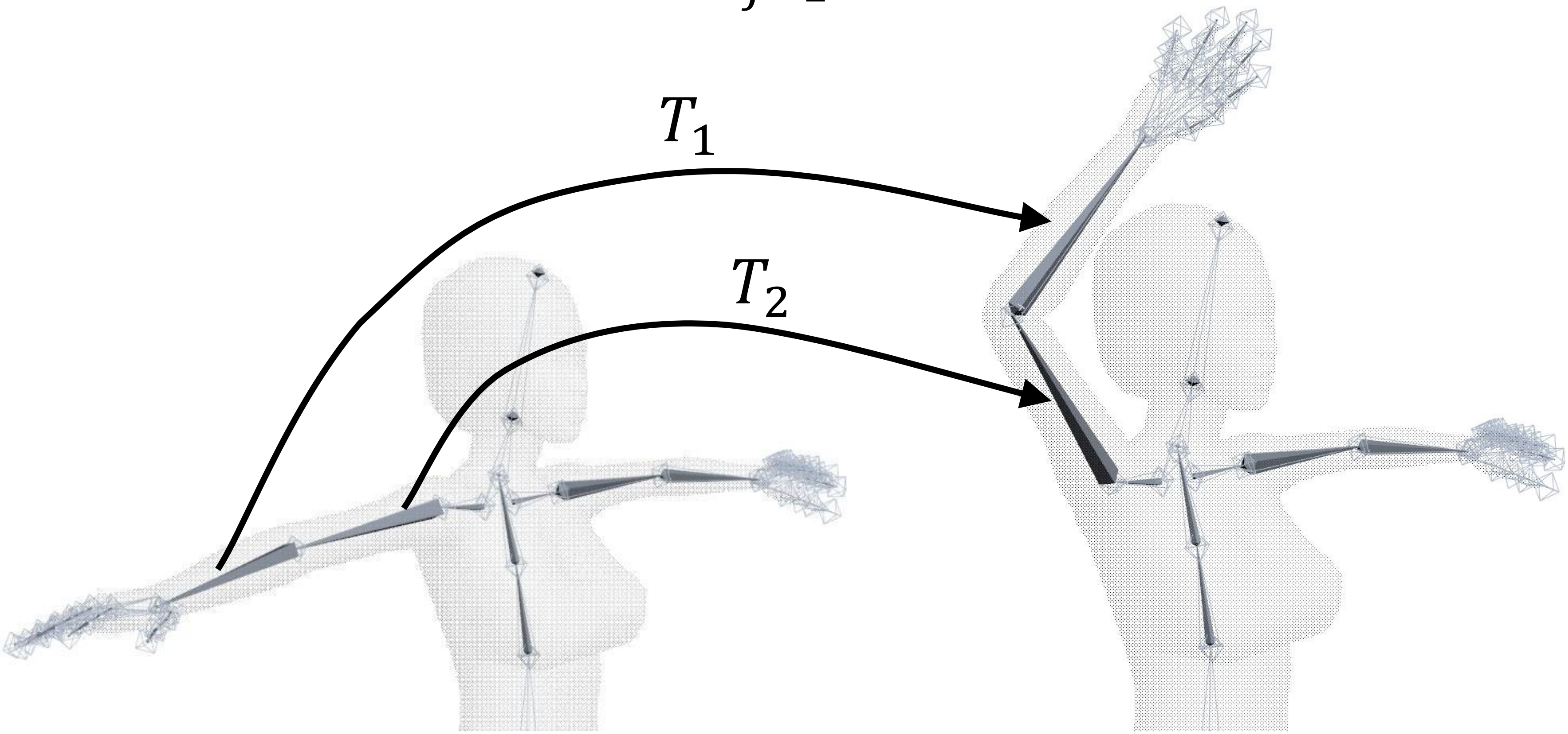


Basic Setup: Skinning Transformation



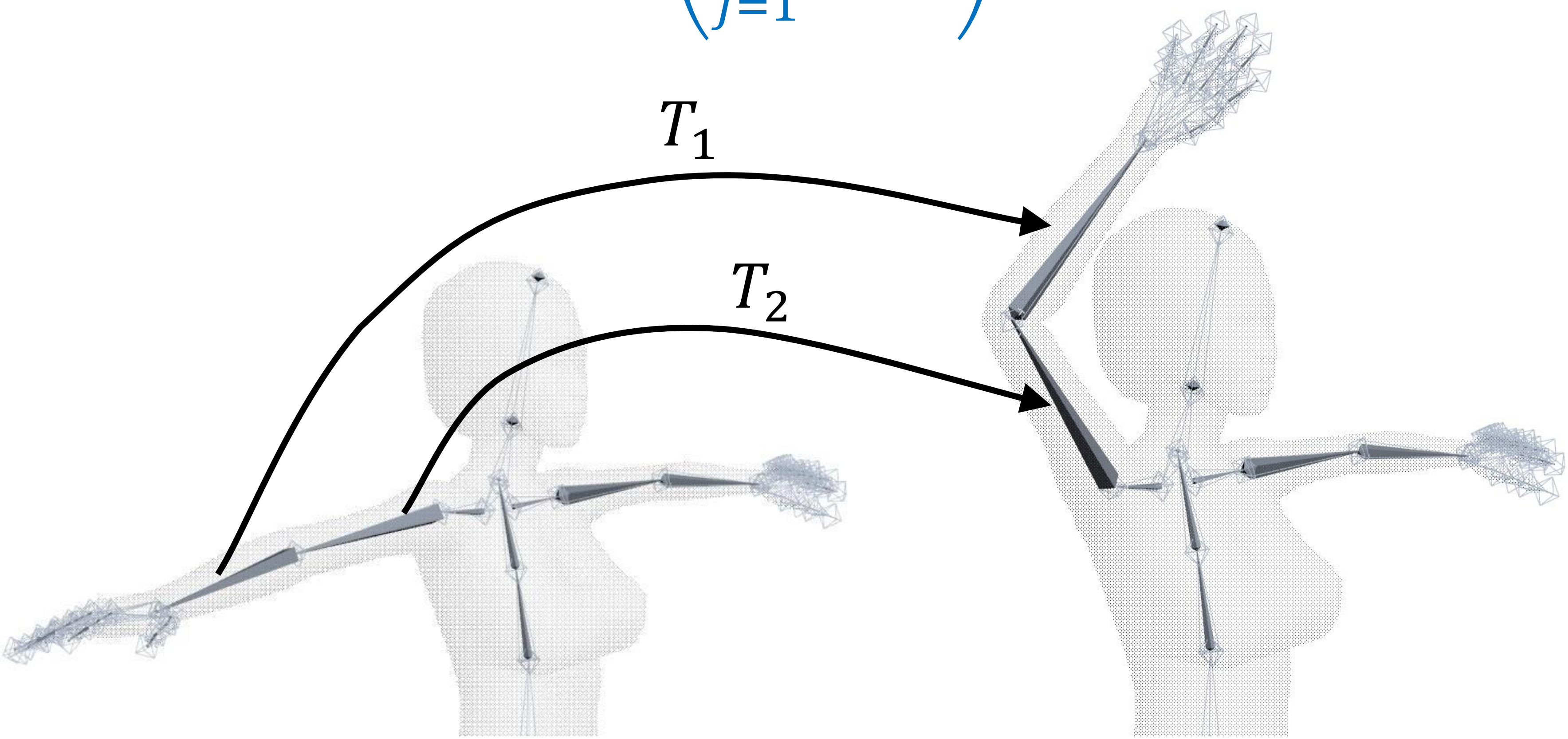
Linear Blend Skinning (LBS)

$$x_i' = \sum_{j=1}^{m} \omega_{ij} T_j x_i$$



Linear Blend Skinning (LBS)

$$x_i' = \left(\sum_{j=1}^{m} \omega_{ij} T_j \right) x_i$$



Linear Blend Skinning (LBS)

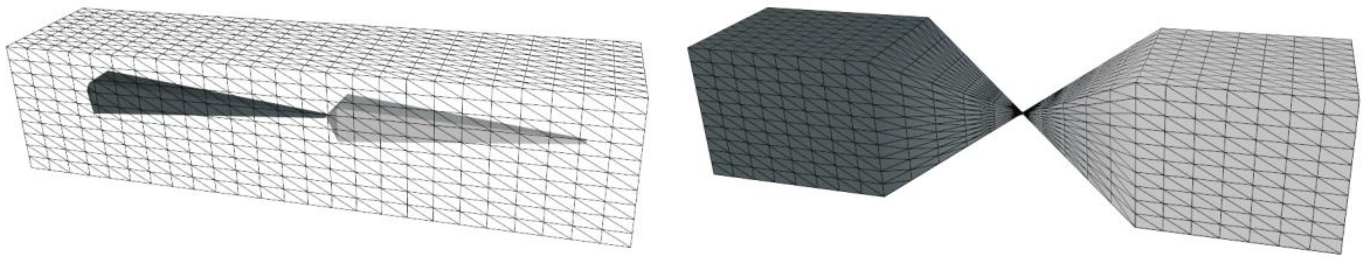
$$x_i' = \left(\sum_{j=1}^{m} \omega_{ij} T_j \right) x_i$$

- Used widely in the industry
- Efficient and GPU-friendly
  - Games like it

基于GPU约束，优化权重，减少权重的使用（即节点受多少关节的影响）

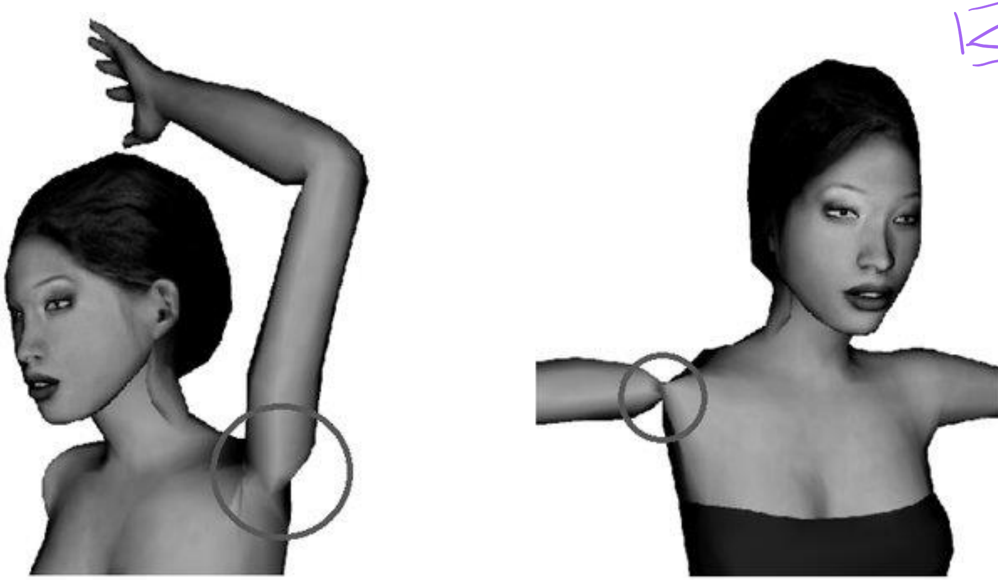
Halo 3

Candy-Wrapper Artifact



Rest pose

Linear blend skinning



Candy-Wrapper Artifact

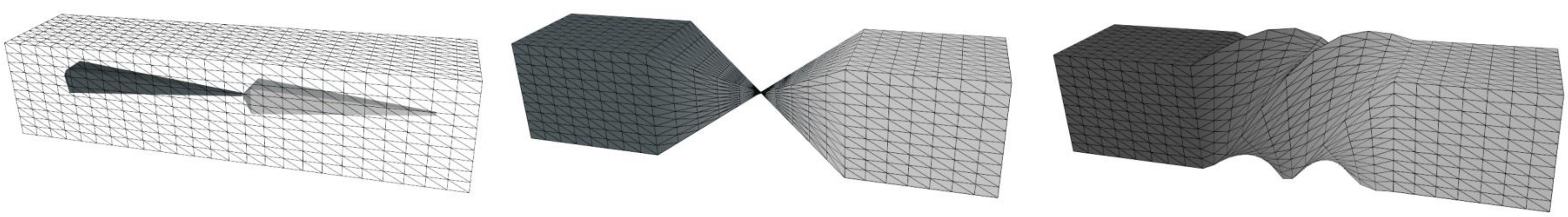
$$x_i' = \left(\sum_{j=1}^{m} \omega_{ij} T_j \right) x_i$$

Consider $T = (R \quad t)$ and

$$\mathbf{R}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{R}_2 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Advanced skinning methods

- Multi-linear Skinning
- Nonlinear Skinning
  - Dual-quaternion Skinning (DQS)



Rest pose Linear blend skinning Dual quaternion skinning

Multi-linear Skinning

- Multi-weight enveloping [Wang and Phillips 2002]
- Animation Space [Merry et al. 2006]

Multi-linear Skinning

$$x' = Tx$$

$$T = [R \mid u] \in \mathbb{R}^{3\times 4}$$

$$v' = Xt$$

$v' \in \mathbb{R}^{3n}$: stack of all vertices x'

$t \in \mathbb{R}^{12m}$: stack of all transformation T

$X \in \mathbb{R}^{3n\times 12m}$: skinning matrix

Multi-linear Skinning

The structure of matrix $\mathbf{X}$

$$\mathbf{X} \in R^{3n \times 12m}$$

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_{11} & \cdots & \mathbf{X}_{1m} \\ \vdots & \ddots & \vdots \\ \mathbf{X}_{n1} & \cdots & \mathbf{X}_{nm} \end{bmatrix}$$

$$\mathbf{X}_{ij} \in R^{3 \times 12}$$

Multi-linear Skinning

Matrix vectorization

$$\mathrm{vec}(\mathbf{A}) = \left(a_{1,1}, a_{1,2}, a_{1,3}, \dots, a_{m,n}\right)^T$$

Kronecker product

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{1,1}\mathbf{B} & \cdots & a_{1,q}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{p,1}\mathbf{B} & \cdots & a_{p,q}\mathbf{B} \end{bmatrix} \in \mathbb{R}^{pr \times qs}$$

B r×s

$$\mathrm{vec}(\mathbf{AB}) = (\mathbf{I}_s \otimes \underbrace{\mathbf{A}}_{m\times n})\,\mathrm{vec}(\mathbf{B}) = (\mathbf{B}^{\mathbf{T}} \otimes \mathbf{I}_p)\,\mathrm{vec}(\mathbf{A})$$

Multi-linear Skinning

$$x_i' = \sum_{j=1}^{m} \omega_{ij} T_j x_i$$

$$\underbrace{x_i'}_{3\times 1} = \mathrm{vec}(x_i') = \sum_{j=1}^{m} \mathrm{vec}\left(T_j \omega_{ij} x_i\right) = \sum_{j=1}^{m} \left(\omega_{ij} \boldsymbol{x}_i \otimes I\right) \mathrm{vec}\left(T_j\right)$$

$$= \sum_{j=1}^{m} X_{ij}^{LBS} t_j \;\; = X_i^{LBS} t$$

LBS in matrix form

$$X_{ij}^{LBS} = [w_{ij}x_{i1}\mathbf{I}_3 \quad w_{ij}x_{i2}\mathbf{I}_3 \quad w_{ij}x_{i3}\mathbf{I}_3 \quad w_{ij}\mathbf{I}_3]$$

$$\mathbf{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathrm{x}_i = \begin{bmatrix} \mathrm{x}_{i1} \\ \mathrm{x}_{i2} \\ \mathrm{x}_{i3} \end{bmatrix}$$

$$\begin{bmatrix} w_{i,j}\mathrm{x}_{i,1} & 0 & 0 & w_{i,j}\mathrm{x}_{i,2} & 0 & 0 & w_{i,j}\mathrm{x}_{i,3} & 0 & 0 & w_{i,j} & 0 & 0 \\ 0 & w_{i,j}\mathrm{x}_{i,1} & 0 & 0 & w_{i,j}\mathrm{x}_{i,2} & 0 & 0 & w_{i,j}\mathrm{x}_{i,3} & 0 & 0 & w_{i,j} & 0 \\ 0 & 0 & w_{i,j}\mathrm{x}_{i,1} & 0 & 0 & w_{i,j}\mathrm{x}_{i,2} & 0 & 0 & w_{i,j}\mathrm{x}_{i,3} & 0 & 0 & w_{i,j} \end{bmatrix}$$

LBS has one weight per vertex/bone pair

Multi-linear Skinning

$$X_{ij} = \begin{vmatrix} * & * & * & * & * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * & * & * & * & * \end{vmatrix}$$

$$\in \mathbb{R}^{3\times12}$$

We can have 36 weights per vertex/bone pair in the most general form

Multi-weight Enveloping [Wang and Phillips 2002]

$$X_{ij}^{MWE} =$$

$$\begin{bmatrix} w_{i,j}^{1}\,\mathrm{x}_{i,1} & 0 & 0 & w_{i,j}^{4}\,\mathrm{x}_{i,2} & 0 & 0 & w_{i,j}^{7}\,\mathrm{x}_{i,3} & 0 & 0 & w_{i,j}^{10} & 0 & 0 \\ 0 & w_{i,j}^{2}\,\mathrm{x}_{i,1} & 0 & 0 & w_{i,j}^{5}\,\mathrm{x}_{i,2} & 0 & 0 & w_{i,j}^{8}\,\mathrm{x}_{i,3} & 0 & 0 & w_{i,j}^{11} & 0 \\ 0 & 0 & w_{i,j}^{3}\,\mathrm{x}_{i,1} & 0 & 0 & w_{i,j}^{6}\,\mathrm{x}_{i,2} & 0 & 0 & w_{i,j}^{9}\,\mathrm{x}_{i,3} & 0 & 0 & w_{i,j}^{12} \end{bmatrix}$$

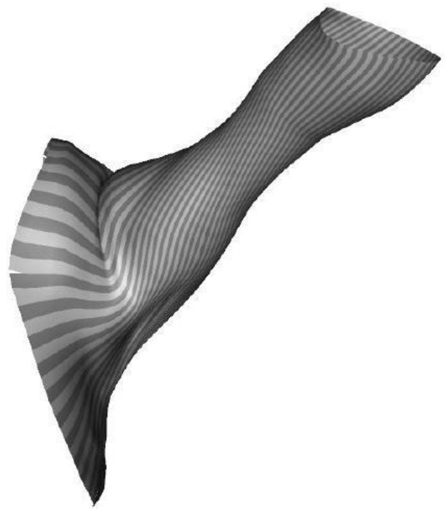
12 weights per vertex/bone

通过样例坐标值进行优化

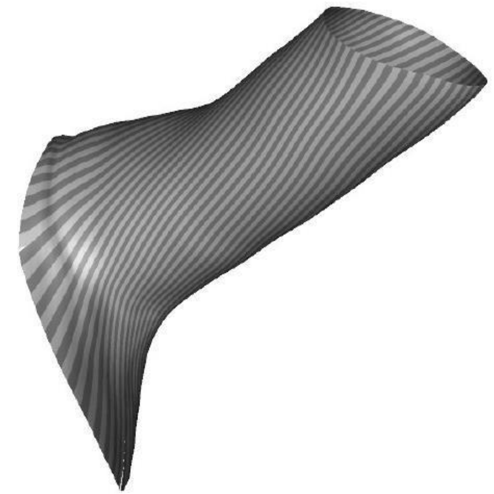
Multi-weight Enveloping [Wang and Phillips 2002]

More powerful

Weights trained from examples



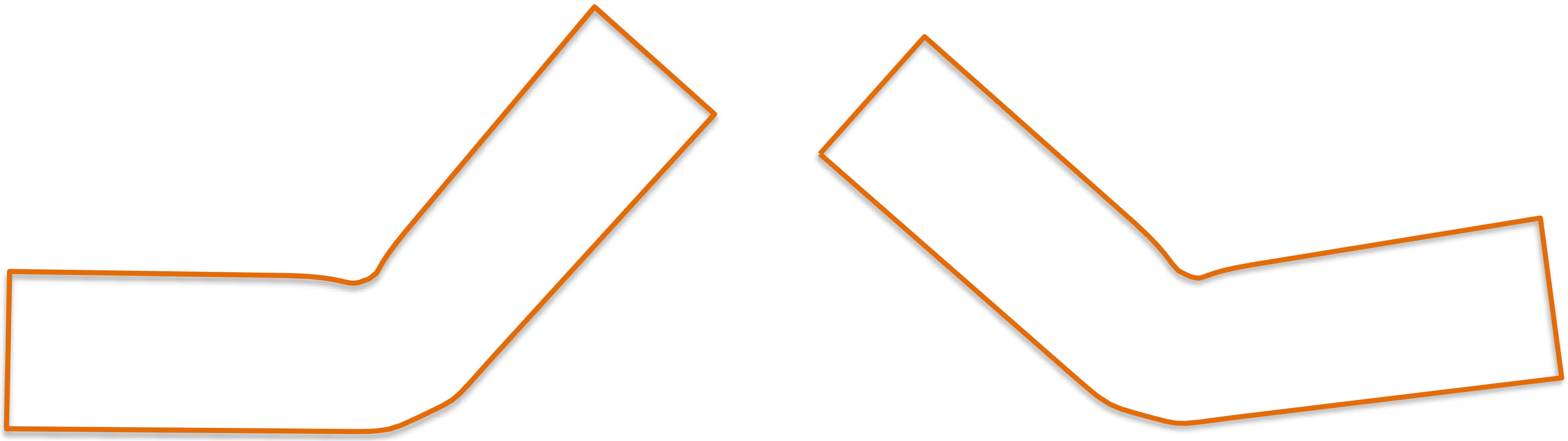
LBS



MWE

Animation Space [Merry et al. 2006]

World-space rotation invariance.



不满足旋转不变性，对应的T不同，外形会
发生改变

Animation Space [Merry et al. 2006]

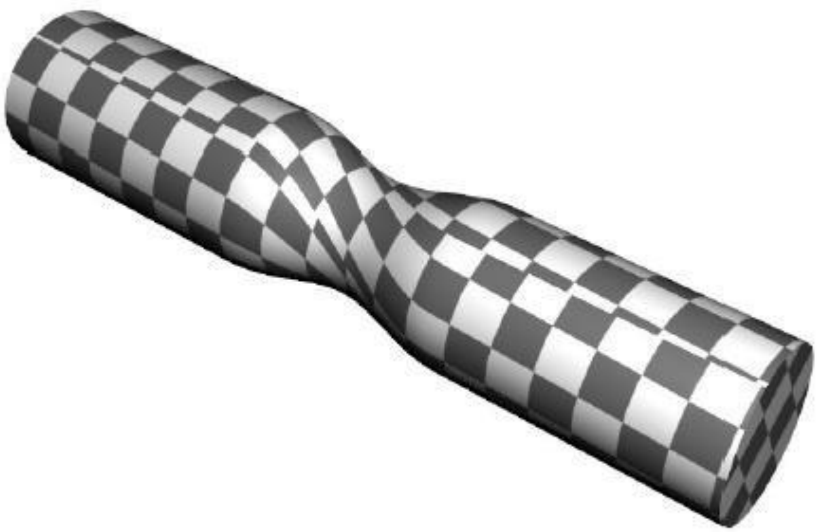
Rotation-invariant results in 4 weights per vertex/bone pair

每个block的weight相同，一共四个

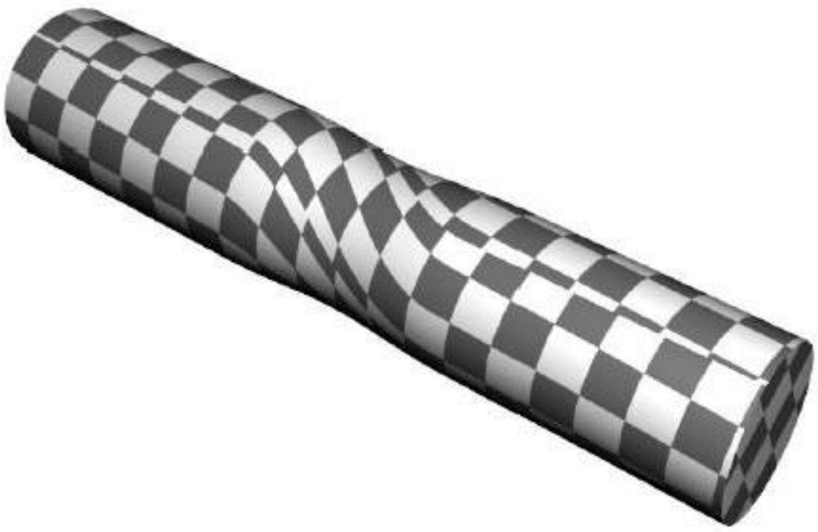
$$X_{ij}^{LBS} = [y_{ij1}\mathbf{I}_3 \quad y_{ij2}\mathbf{I}_3 \quad y_{ij3}\mathbf{I}_3 \quad y_{ij4}\mathbf{I}_3]$$

$$\sum_{j=1}^{m} y_{ij4} = 1$$

Animation Space [Merry et al. 2006]



LBS



Animation space

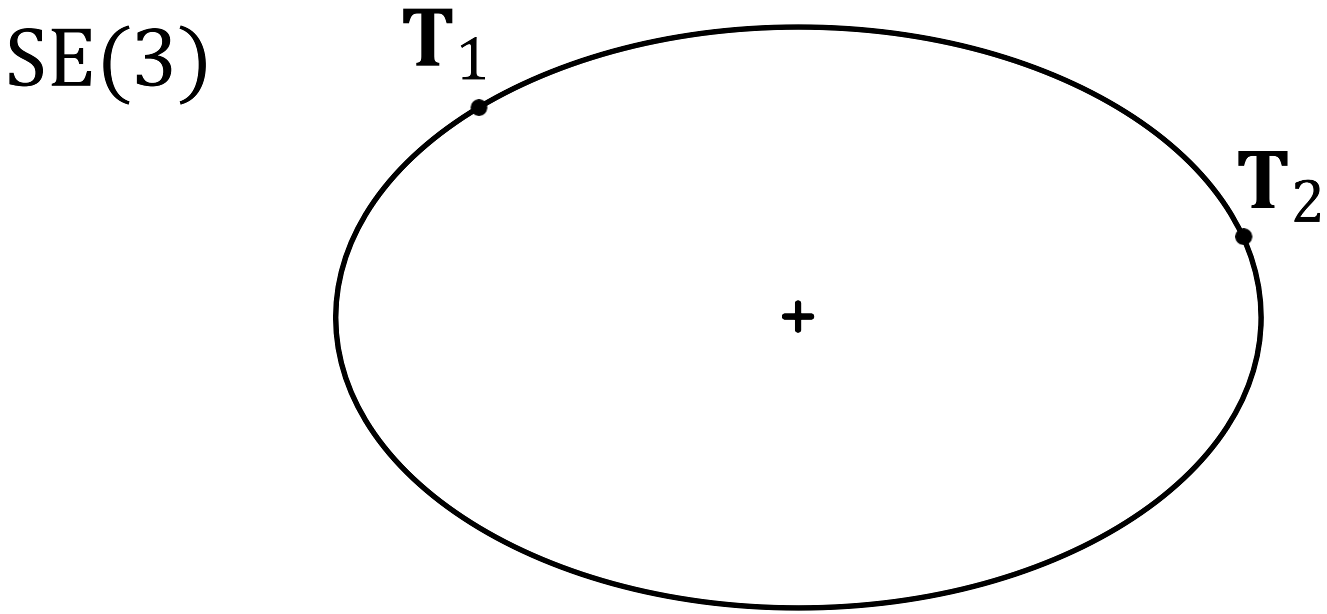
Non-linear Skinning

$$x_i' = \left(\sum_{j=1}^{m} \omega_{ij} T_j \right) x_i$$

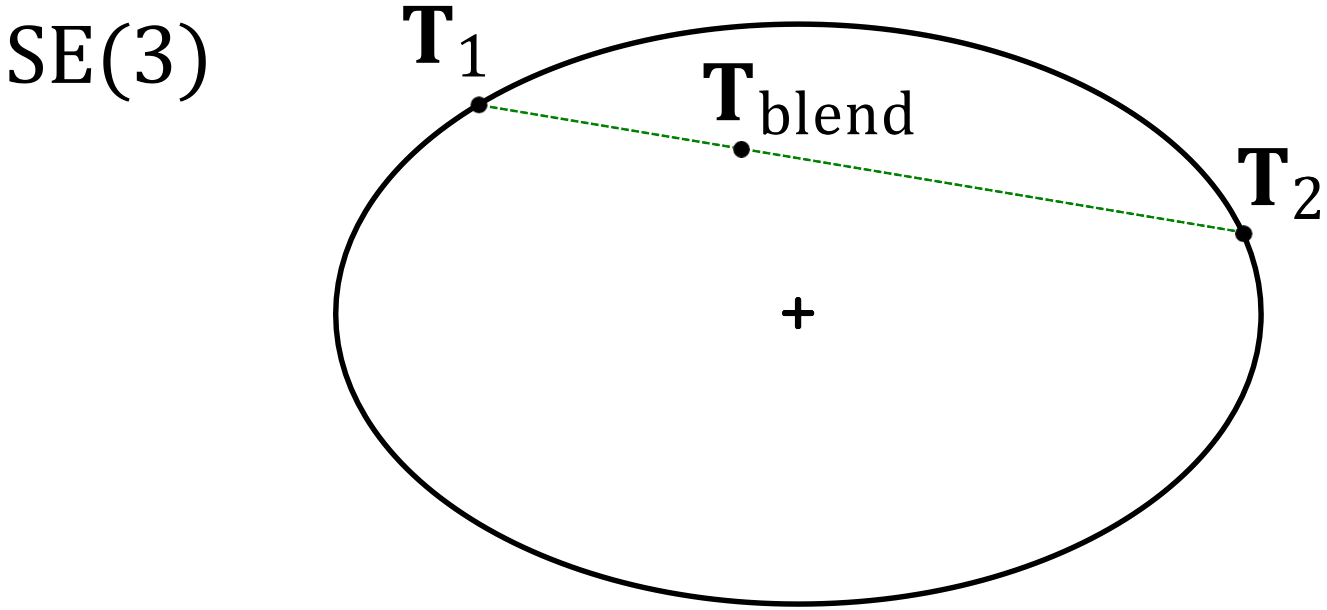
$T_j \in SE(3)$: 刚性变换群

$R \in SO(3)$: 旋转变换群

Interpolation in SE(3)

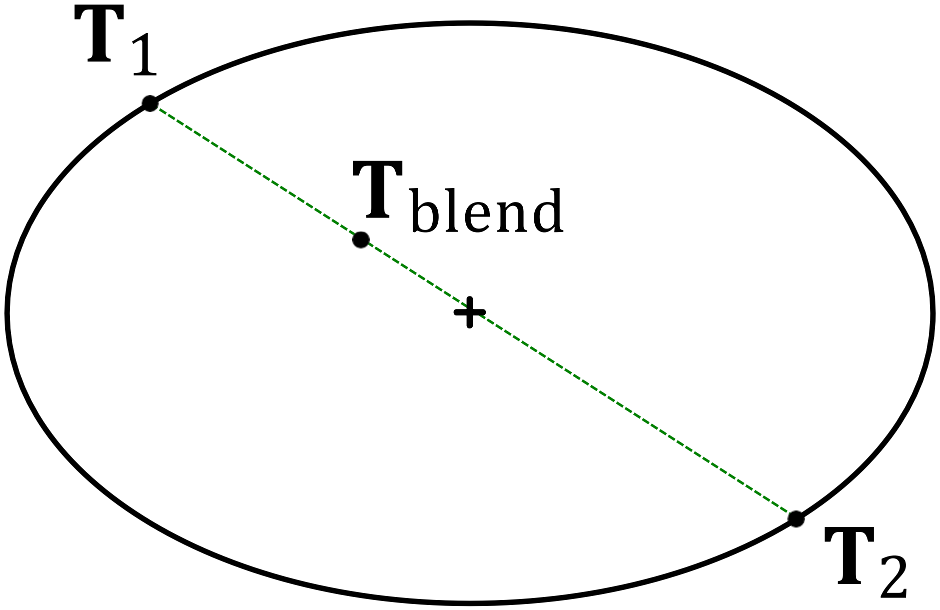


Interpolation in SE(3)

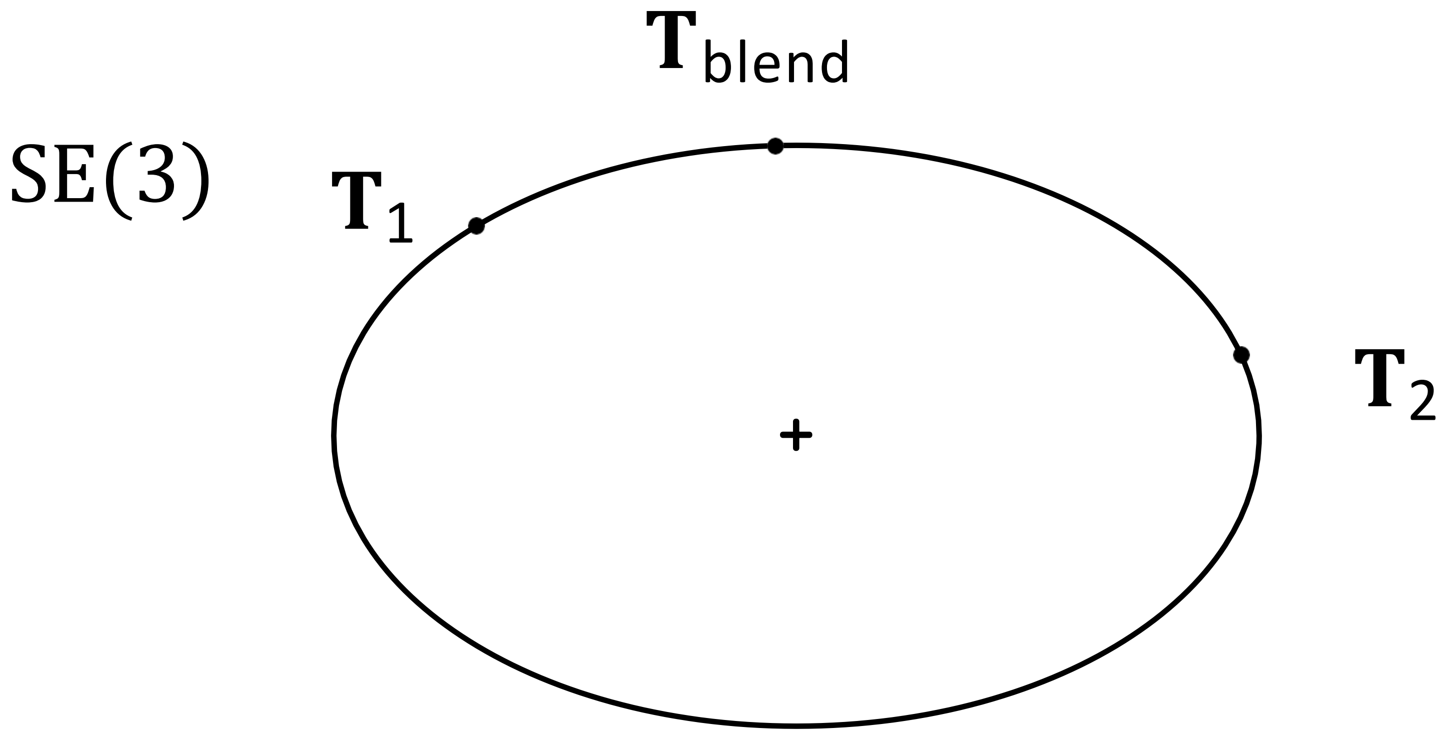


Interpolation in SE(3)

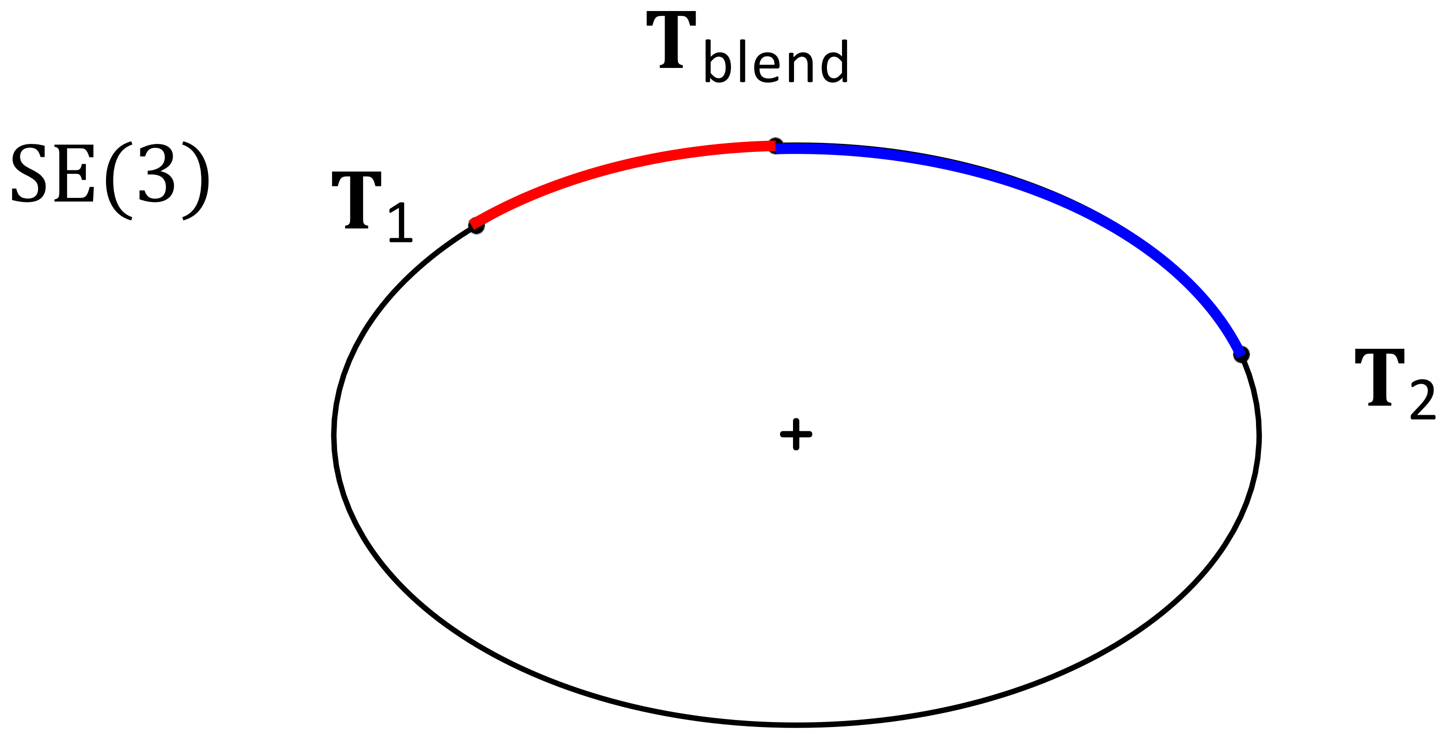
$\mathrm{SE}(3)$



Intrinsic blending



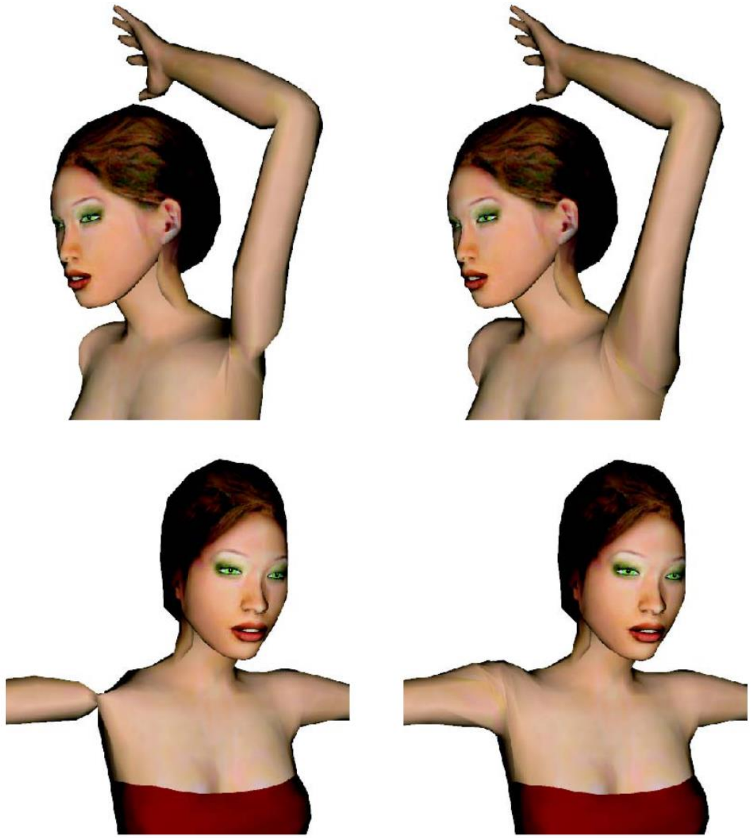
Intrinsic blending



对偶四元数，可以表示平移加旋转

Dual-Quaternion Skinning (DQS)

- Approximation of intrinsic averages in SE(3)



Ladislav Kavan, Steven Collins, Jiri Zara, Carol O‘Sullivan. ***Geometric Skinning with Approximate Dual Quaternion Blending***, ACM Transaction on Graphics, 27(4), 2008.

Dual-Quaternion

- Dual number

$$x = a + b\varepsilon$$

where $\varepsilon^2 = 0$

$y = c + d\varepsilon$

$x \cdot y = ac + ad\varepsilon + bc\varepsilon$

Dual-Quaternion

- Dual quaternion

$$\widehat{\boldsymbol{q}} = \boldsymbol{q}_0 + \varepsilon \boldsymbol{q}_\varepsilon$$

8个变量

- Conjugation

三种不同的共轭

$$\widehat{\boldsymbol{q}}^* = \boldsymbol{q}_0^* + \varepsilon \boldsymbol{q}_\varepsilon^*$$

① q^*
② $-\varepsilon$
③ q^* 同时 $-\varepsilon$

- Norm

$$\|\hat{\mathbf{q}}\| = \sqrt{\hat{\mathbf{q}}^* \hat{\mathbf{q}}} = \|\mathbf{q}_0\| + \varepsilon \frac{\langle \mathbf{q}_0, \mathbf{q}_\varepsilon \rangle}{\|\mathbf{q}_0\|}$$

$(q_0 + \varepsilon q_\varepsilon)(q_0^* + \varepsilon q_\varepsilon^*)$

$= q_0 q_0^* + \varepsilon q_0 q_\varepsilon^* + \varepsilon q_\varepsilon q_0^*$

Dual-Quaternion

Scalar Multiplication

$$s\mathbf{q} = s\mathbf{q}_r + s\mathbf{q}_d \varepsilon$$

Addition

$$\mathbf{q}_1 + \mathbf{q}_2 = \mathbf{q}_{r1} + \mathbf{q}_{r2} + (\mathbf{q}_{d1} + \mathbf{q}_{d2})\varepsilon$$

Multiplication

$$\mathbf{q}_1\mathbf{q}_2 = \mathbf{q}_{r1}\mathbf{q}_{r2} + (\mathbf{q}_{r1}\mathbf{q}_{d2} + \mathbf{q}_{d1}\mathbf{q}_{r2})\varepsilon$$

Dual-Quaternion

- Norm

$$\|\hat{\mathbf{q}}\| = \sqrt{\hat{\mathbf{q}}^* \hat{\mathbf{q}}} = \|\mathbf{q}_0\| + \varepsilon \frac{\langle \mathbf{q}_0, \mathbf{q}_\varepsilon \rangle}{\|\mathbf{q}_0\|}$$

- Unit dual quaternion: $\|\widehat{\boldsymbol{q}}\| = 1$
    - $\|\boldsymbol{q}_{\mathbf{0}}\| = 1$
    - $\boldsymbol{q}_{\mathbf{0}} \cdot \boldsymbol{q}_{\varepsilon} = 0$

Dual-Quaternion

- Like quaternion, any rigid transformation $T \in SE(3)$ can be converted into a dual-quaternion

$$T = [R \mid t] \rightarrow \widehat{\boldsymbol{q}} = \boldsymbol{q}_0 + \varepsilon \boldsymbol{q}_\varepsilon$$

$$\boldsymbol{q}_0 = \overset{R}{\boldsymbol{r}}$$

$$\boldsymbol{q}_\varepsilon = \frac{1}{2}\boldsymbol{t}\boldsymbol{r}$$

Dual-Quaternion

- Transform a vector using unit dual quaternion

$$\widehat{\boldsymbol{v}}' = \widehat{\boldsymbol{q}}\widehat{\boldsymbol{v}}\widehat{\boldsymbol{q}}^* \rightarrow \text{是第三种共轭}$$

Where

$$\widehat{\boldsymbol{v}} = 1 + \varepsilon(0, \boldsymbol{v})$$

$$\bar{v} = [1, 0, 0, 0, 0, \underbrace{x, y, z}_{\text{直接部分}}]$$

Dual-Quaternion

- Like quaternion, any rigid transformation $T \in SE(3)$ can be converted into a dual-quaternion

$Tx = Rx + t$

$$T = [R \mid t] \rightarrow \hat{\boldsymbol{q}} = \boldsymbol{q}_0 + \varepsilon \boldsymbol{q}_\varepsilon$$

$$\boldsymbol{q}_0 = \boldsymbol{r}$$

$$\boldsymbol{q}_\varepsilon = \frac{1}{2}\boldsymbol{t}\boldsymbol{r}$$

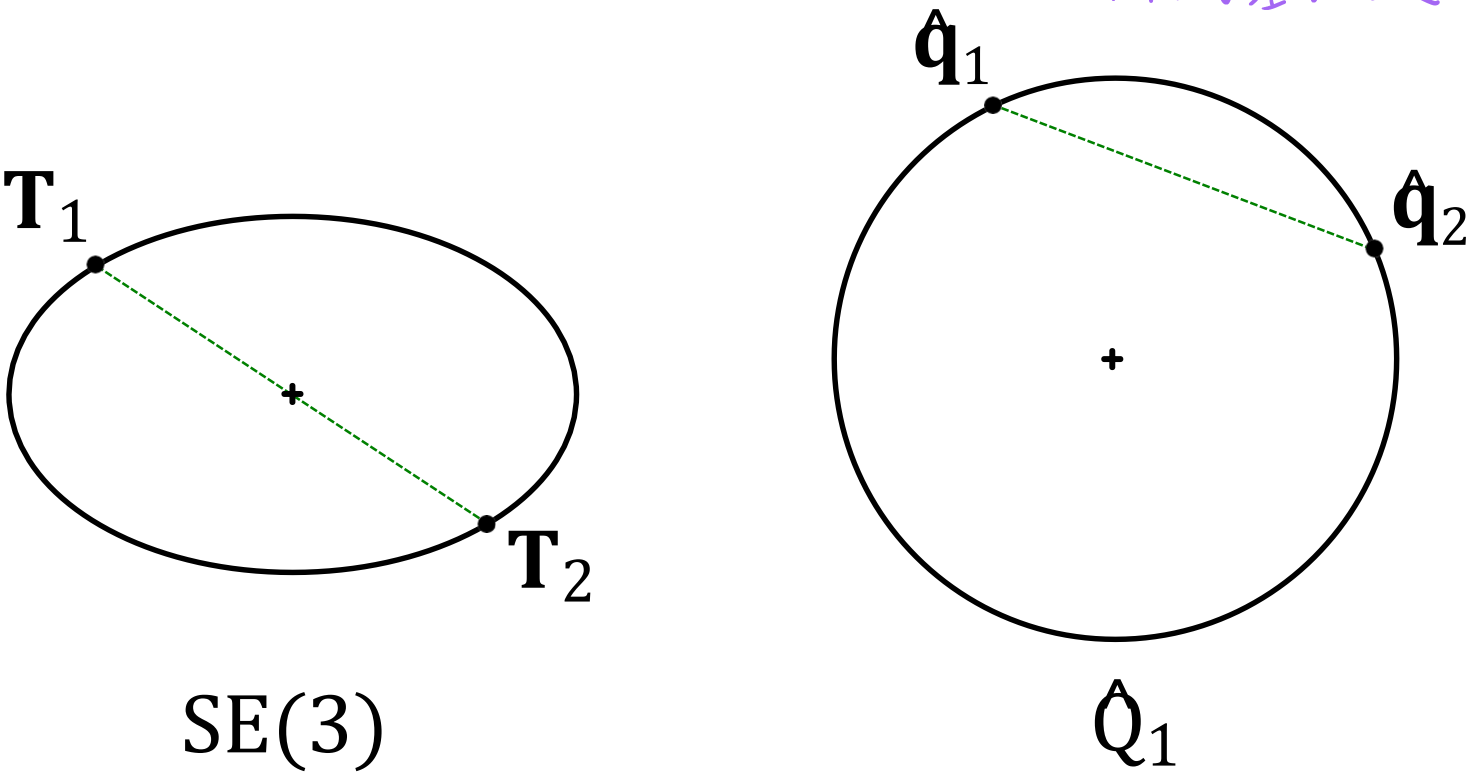
$t = (0, t)$

$\hat{\mathbf{q}}$ and $-\hat{\mathbf{q}}$ represent the same transformation

$\hat{Q}_1$ is a double cover of SE(3)

Double cover visualized

解决了线性插值的基本问题



Interpolating Dual-Quaternion

$$\hat{\boldsymbol{q}} = (\mathbf{1} - \boldsymbol{t})\hat{\boldsymbol{q}}_{\mathbf{0}} + \boldsymbol{t}\hat{\boldsymbol{q}}_{\mathbf{1}}$$

$$\Downarrow$$

$$\hat{\boldsymbol{q}} = \frac{(\mathbf{1} - \boldsymbol{t})\hat{\boldsymbol{q}}_{\mathbf{0}} + \boldsymbol{t}\hat{\boldsymbol{q}}_{\mathbf{1}}}{\|(\mathbf{1} - \boldsymbol{t})\hat{\boldsymbol{q}}_{\mathbf{0}} + \boldsymbol{t}\hat{\boldsymbol{q}}_{\mathbf{1}}\|}$$

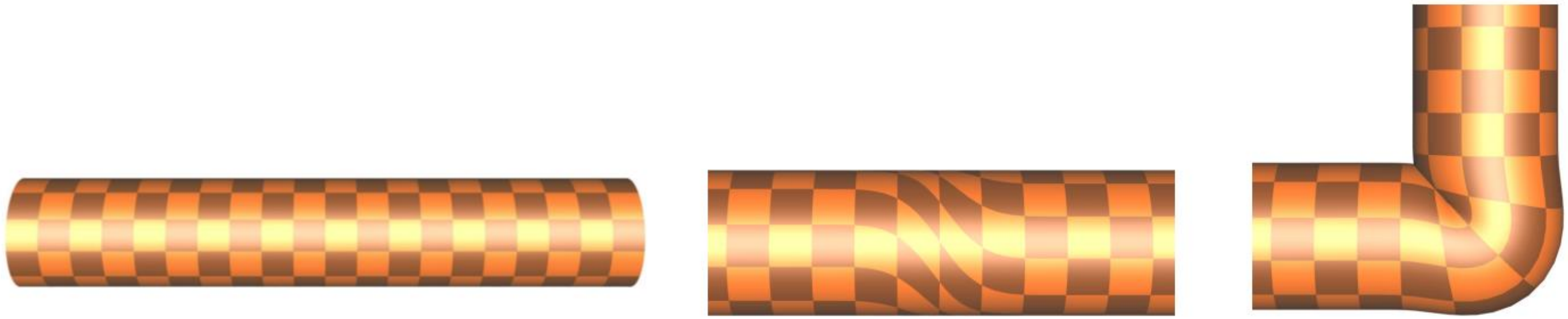
Dual-Quaternion Skinning

$$\boldsymbol{\hat{q}} = (\mathbf{1} - \boldsymbol{t})\boldsymbol{\hat{q}}_{\mathbf{0}} + \boldsymbol{t}\boldsymbol{\hat{q}}_{\mathbf{1}}$$

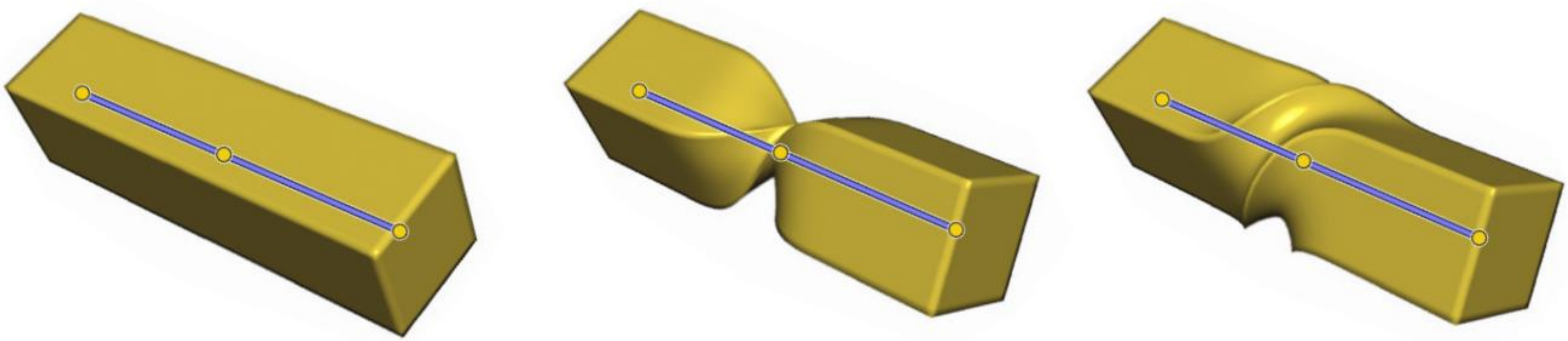
$$\Downarrow$$

$$\boldsymbol{\hat{q}} = \frac{(\mathbf{1} - \boldsymbol{t})\boldsymbol{\hat{q}}_{\mathbf{0}} + \boldsymbol{t}\boldsymbol{\hat{q}}_{\mathbf{1}}}{\|(\mathbf{1} - \boldsymbol{t})\boldsymbol{\hat{q}}_{\mathbf{0}} + \boldsymbol{t}\boldsymbol{\hat{q}}_{\mathbf{1}}\|}$$

Dual-Quaternion Skinning



Rest pose — Dual quaternions: twist — Dual quaternions: bend



Rest pose — Linear blend skinning — Dual quaternion skinning

Non-rigid transformations

Two-phase skinning

Enhanced for production use by Disney

[Lee et al. 2013]

Uncovered Topics

- Deformers
- Skinning weight reduction and compression
- Skeleton Extraction

有没有问题?

Any Questions