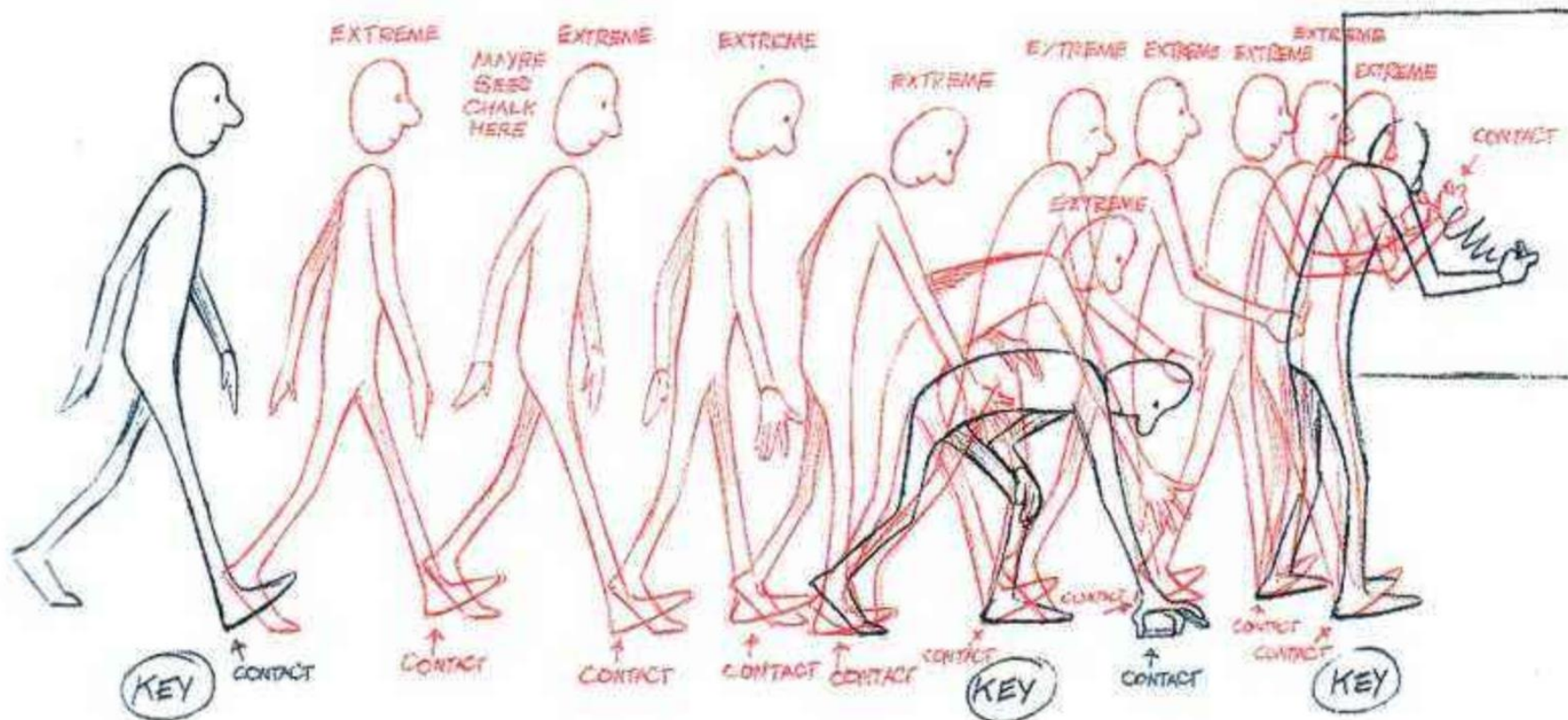


9.2 基于形变的动画

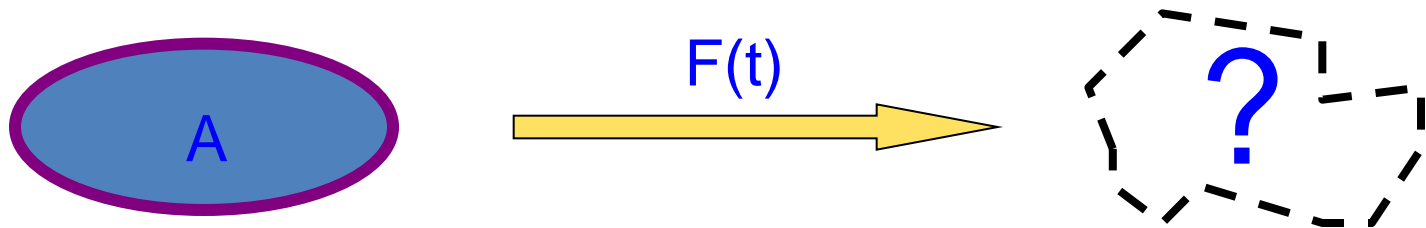
- 关键帧动画
- 关键帧插值
- 三维网格的自由编辑
- 基于约束的编辑
- 骨架驱动的编辑
- 编辑迁移

关键帧动画 (Morphing)

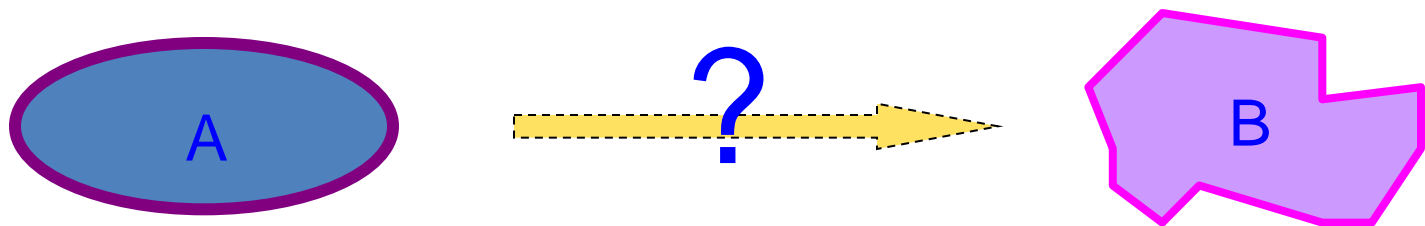


Shape Transformations

- Warping: Unary Op
 - Given **Object A** and $F(t)$, find **Object B**

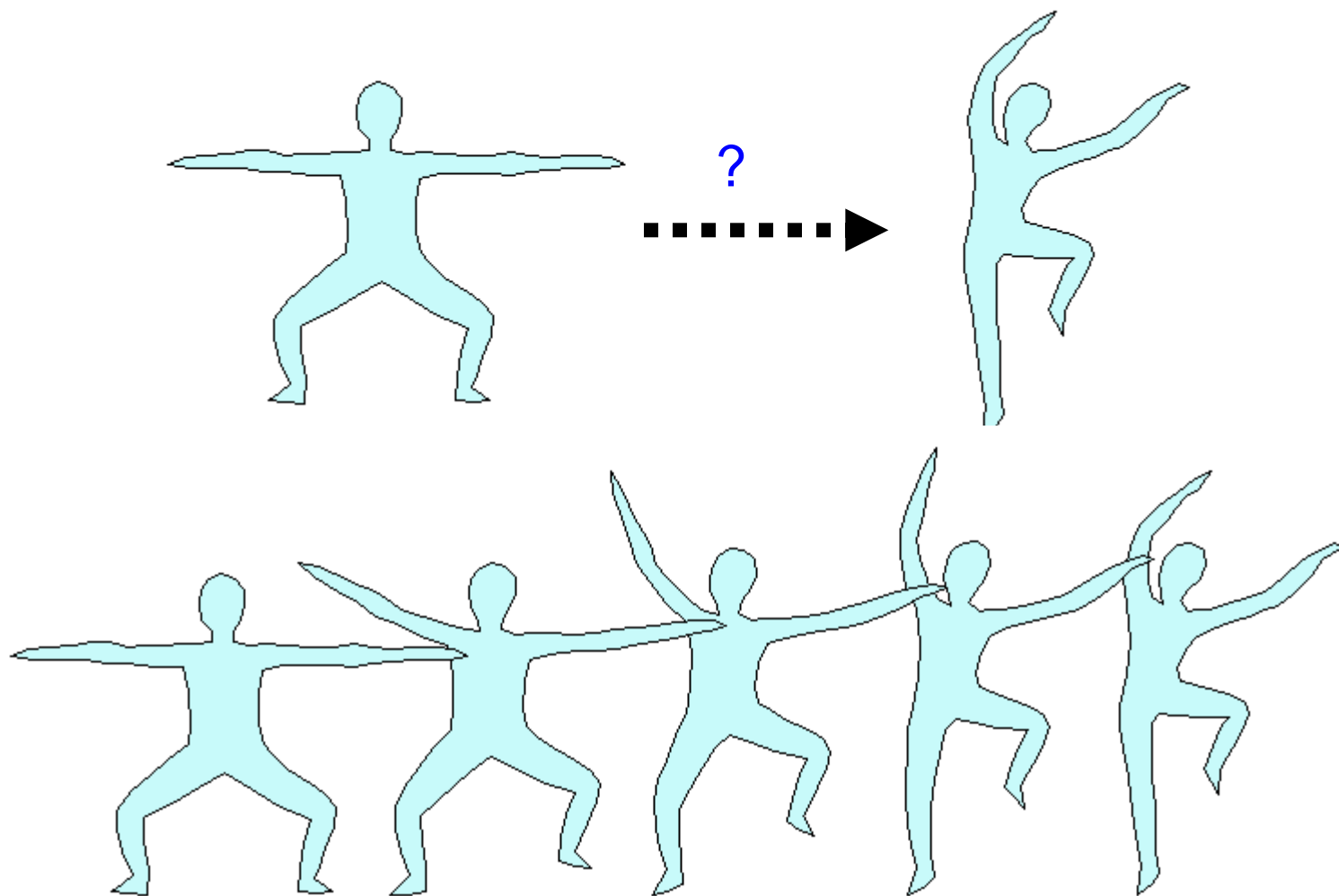


- Morphing: Binary Op
 - Given **Object A** and **Object B**, find $F(t)$

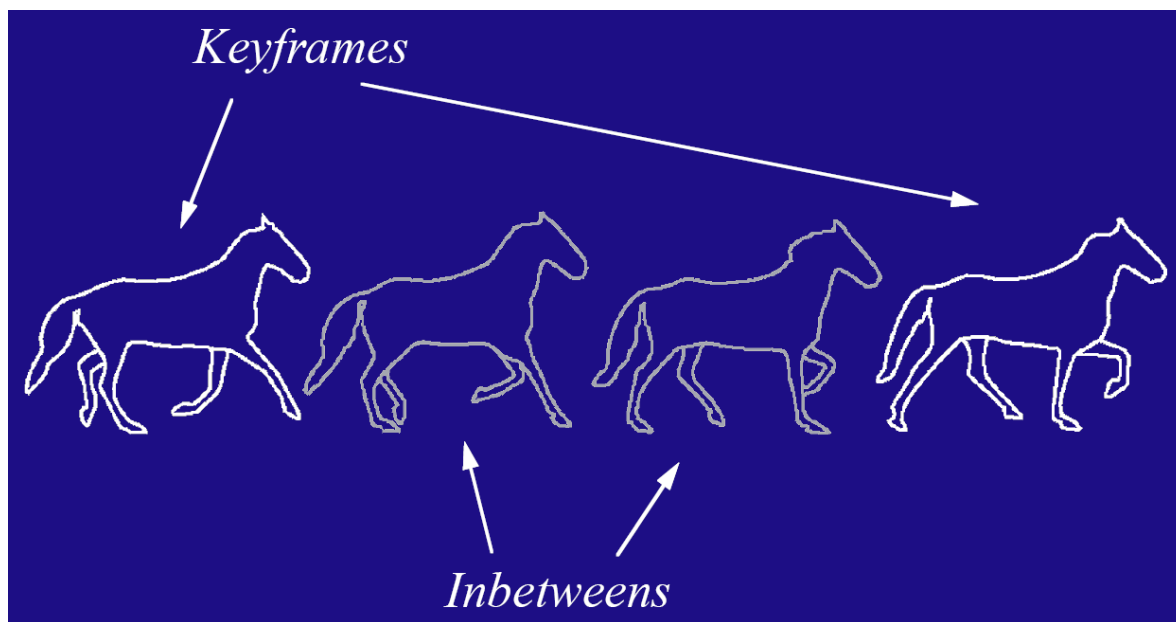
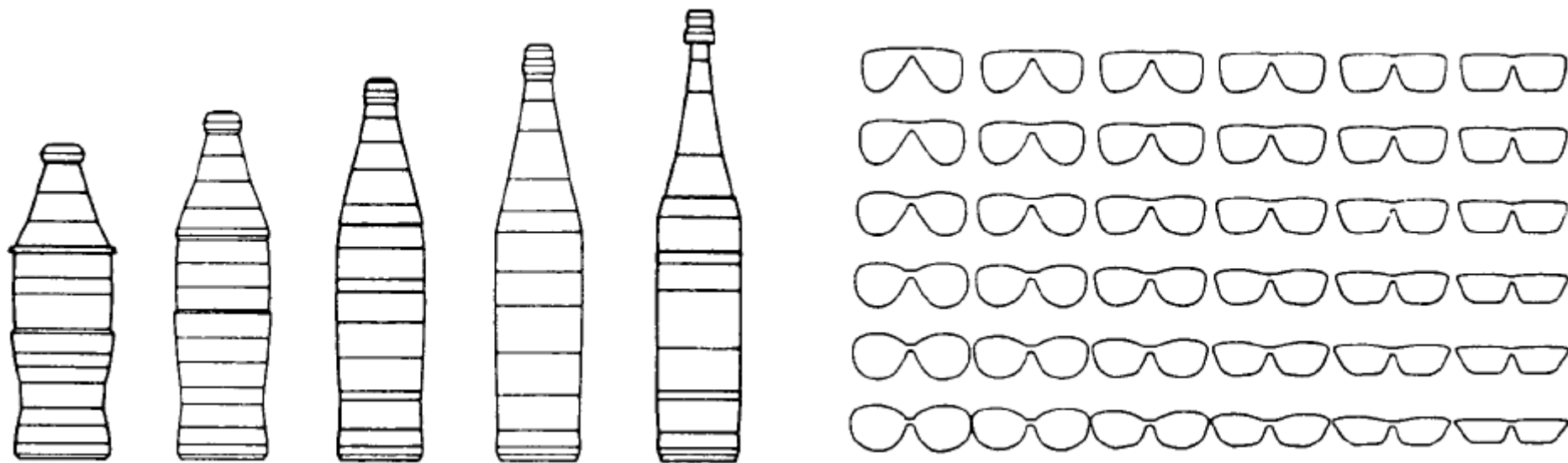


Morphing Problem

关键帧插值



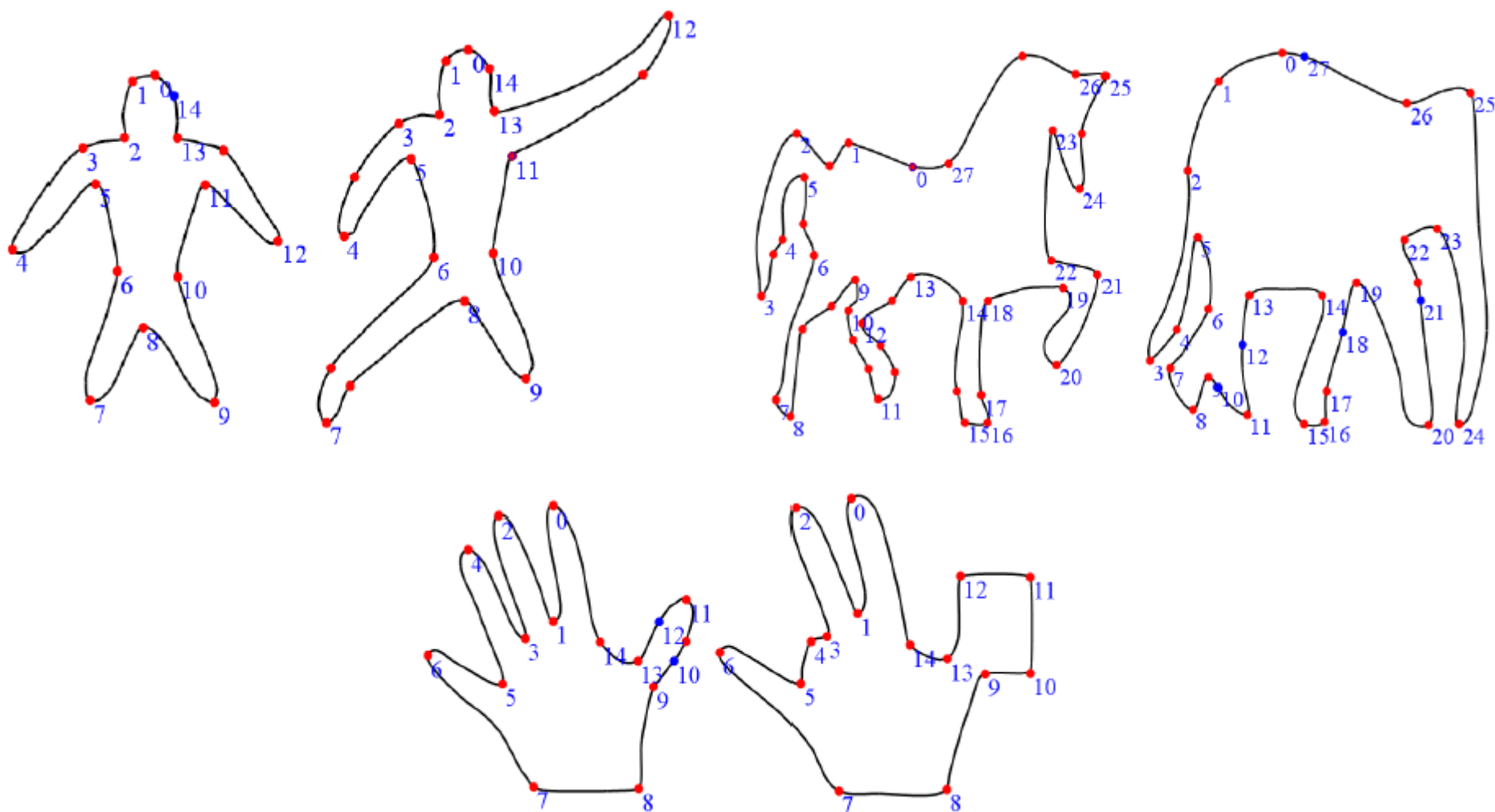
应用



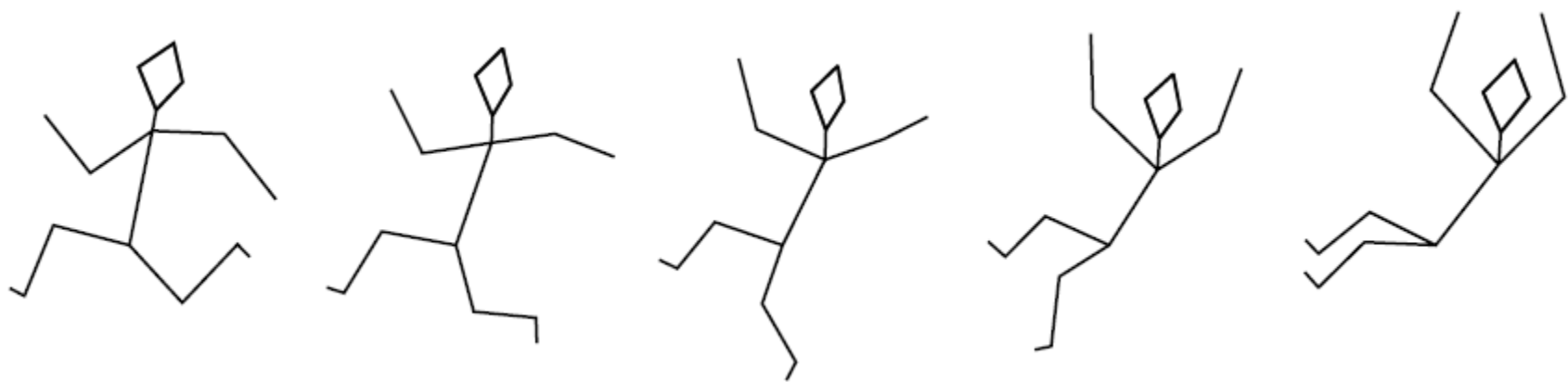
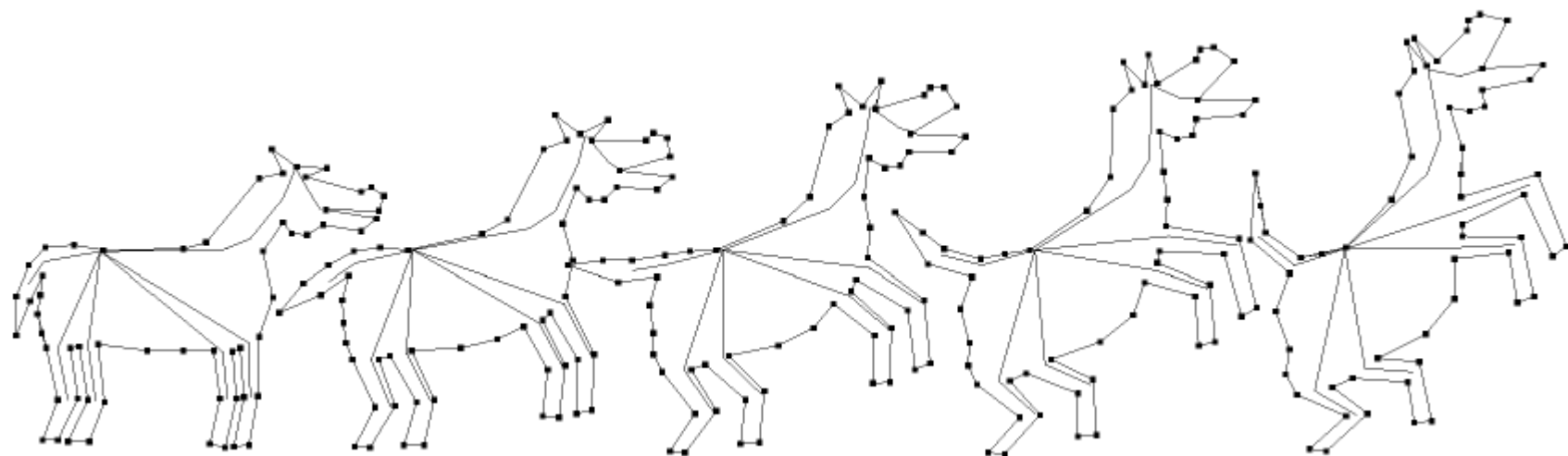
核心问题

- 关键点的匹配 Correspondence problem
 - find a correspondence between vertices of the two shapes
- 轨迹的插值 Path problem
 - find paths that the corresponding vertices traverse during the morphing process
- 这两个问题相互独立

1. 关键点的匹配

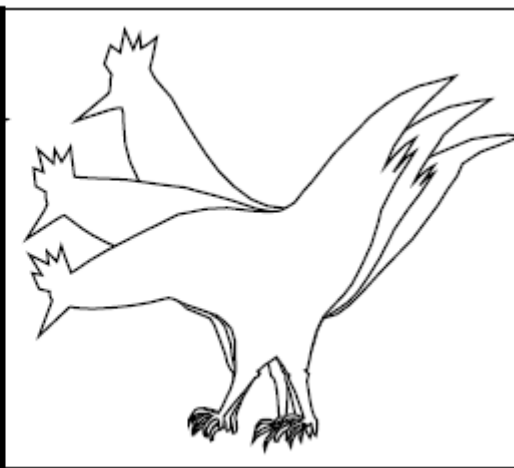
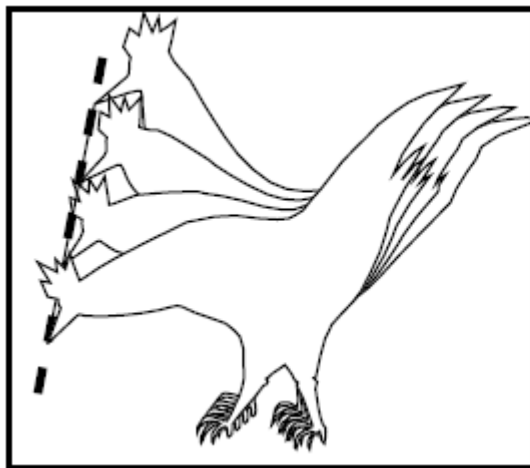
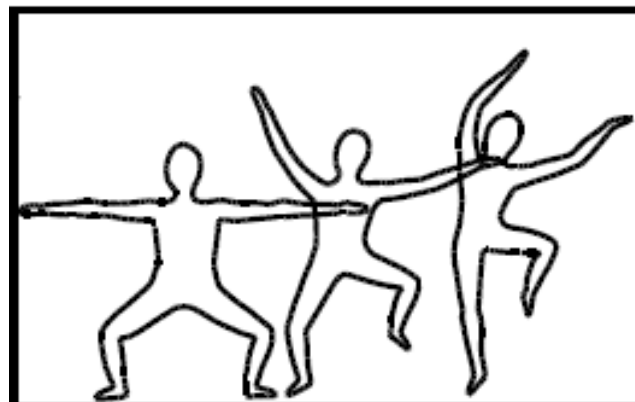
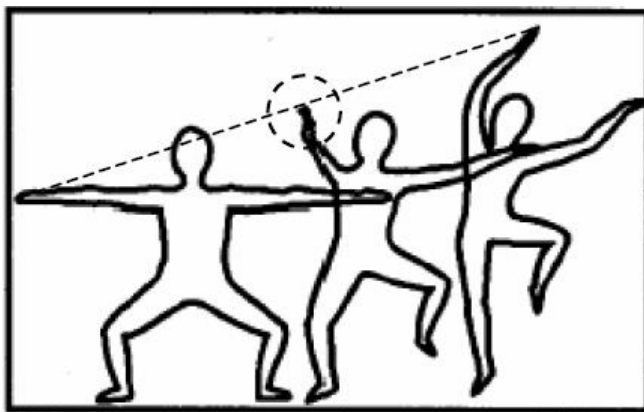


“部件”的匹配

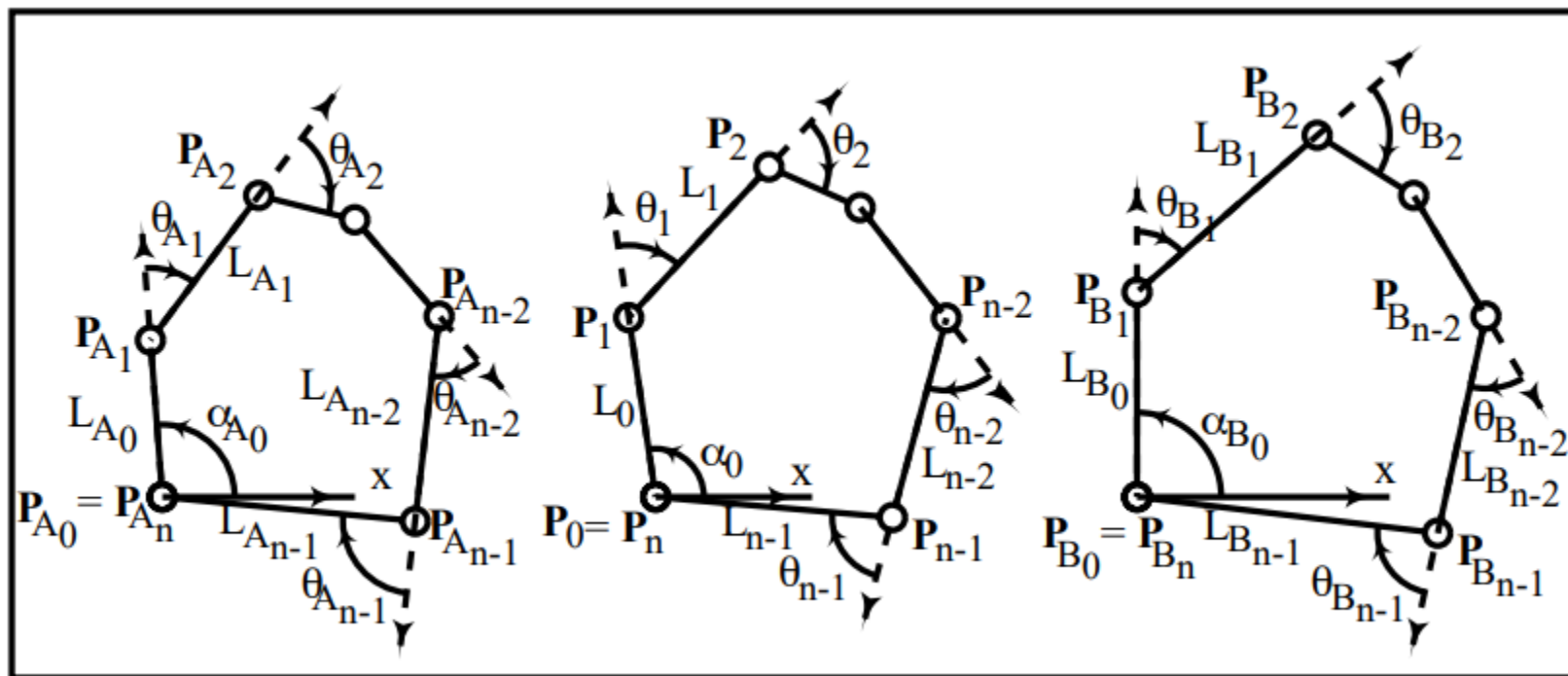


2. 运动路径的插值

Linear VS. Non-linear



Intrinsic shape blending



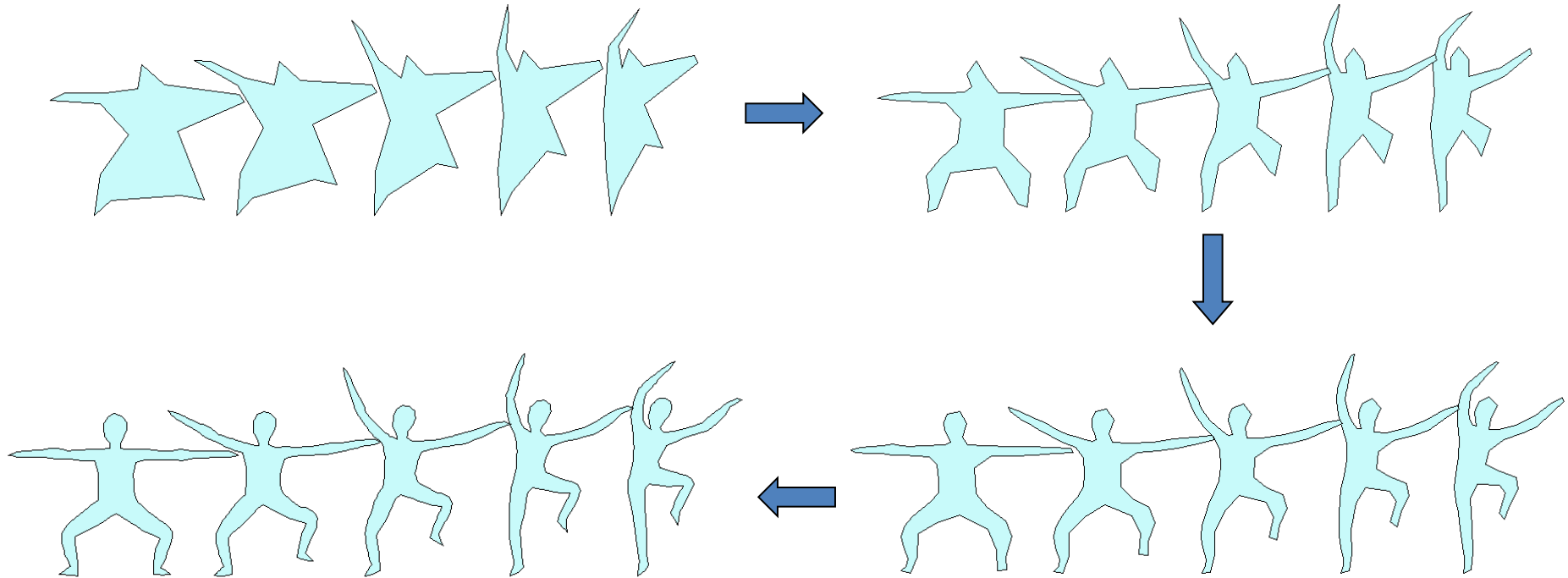
$$\alpha_0 = (1 - t)\alpha_{A_0} + t\alpha_{B_0},$$

$$\theta_i = (1 - t)\theta_{A_i} + t\theta_{B_i}, \quad (i = 1, 2, \dots, m).$$

$$L_i = (1 - t)L_{A_i} + tL_{B_i}, \quad (i = 0, 1, 2, \dots, m).$$

“2-D shape blending: An intrinsic solution to the vertex path problem.1993”

Wavelet Interpolation



- (1) using wavelet decomposition to remove short edges and/or duplicated vertices,
- (2) applying the intrinsic shape-blending algorithm to blend the overall shapes of the polygons
- (3) applying wavelet reconstruction to reconstruct the inbetweening of the original polygons.

“Wavelet shape blending”

Morphing between Different Topologies (拓扑)



Polygonal shape blending with topological evolutions. 2004.

网格编辑

(Mesh editing/deformation)

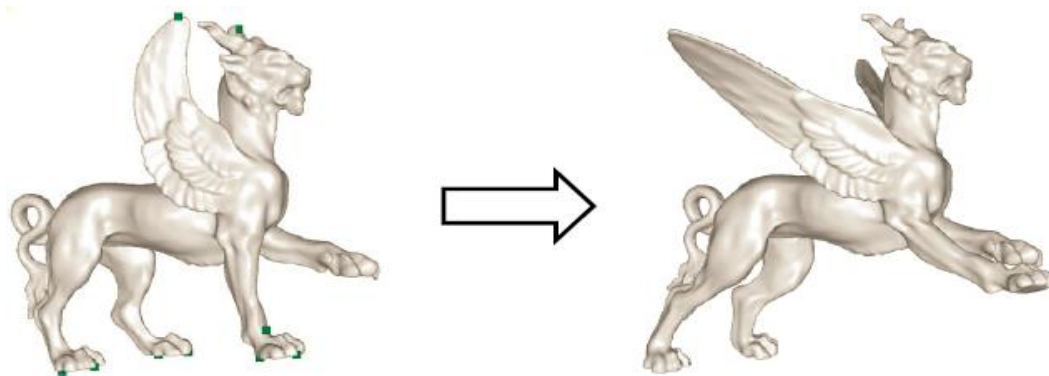
— 定义

- 操纵和修改网格表面的几何形状，同时能够保留原始网格几何细节的操作。



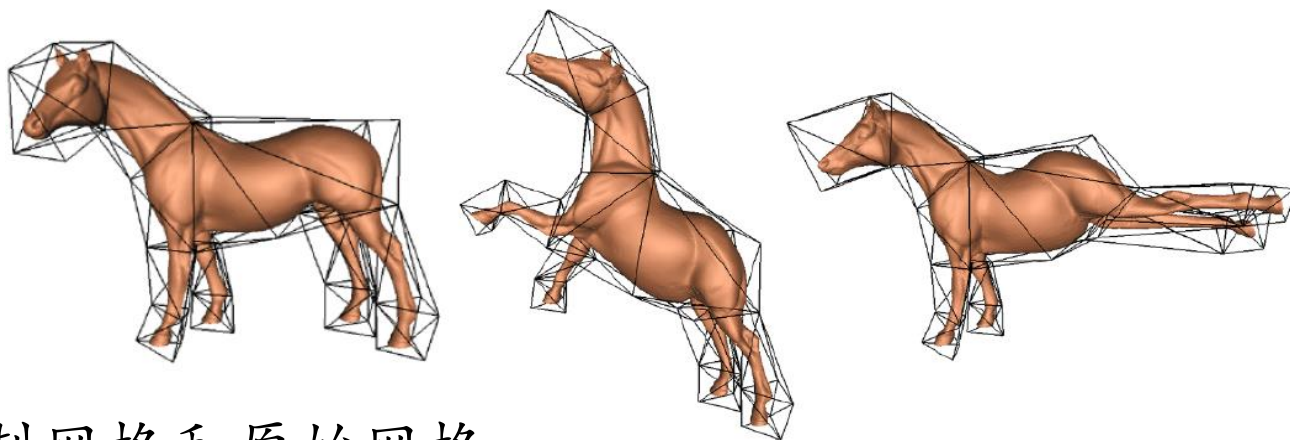
网格编辑问题起源

- 各种应用中需要动态形状，例如，游戏、电影、计算机辅助设计等。但是，每种状态下的形状单独建模比较费时费力。
- 目标
 - 通过内部或者外部作用力改变形状
 - 编辑应该是快速的并且尽可能接近现实物体操作



自由编辑 (Free-form deformation)

- 思想：将网格嵌入到一个简单的控制多面体中
 - 控制多面体比原始形状简单
- 方法：
 - 对控制多面体进行变形
 - 根据变形后的控制多面体更新嵌入网格的形状



控制网格和原始网格

编辑后结果

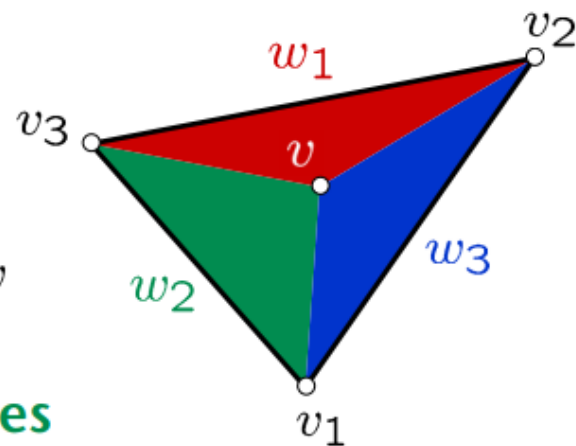
重心坐标

August Ferdinand Möbius [1827]

- v is the **barycentre** of the points v_1, \dots, v_n with **weights** w_1, \dots, w_n if and only if

$$v = \frac{w_1 v_1 + \dots + w_n v_n}{w_1 + \dots + w_n}$$

- w_i are the **barycentric coordinates** of v
- **unique** up to common factor for **triangles**

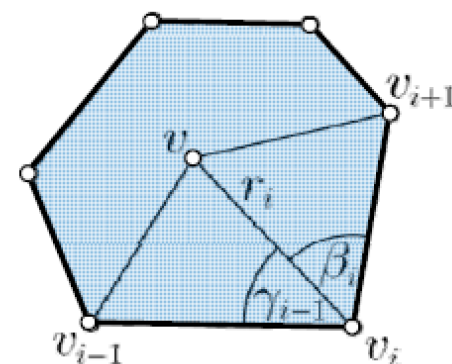


$$v = \frac{w_1 v_1 + w_2 v_2 + w_3 v_3}{w_1 + w_2 + w_3} \iff w_i = \eta A(v, v_{i+1}, v_{i+2})$$

各种重心坐标的计算方法

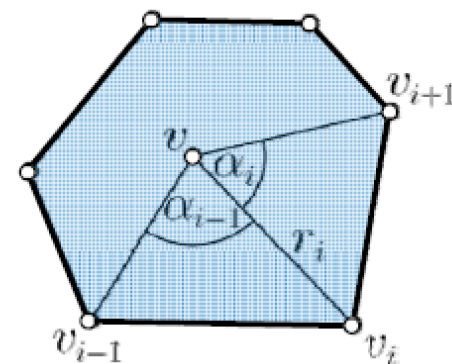
- *Wachspress* (WP) coordinates

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



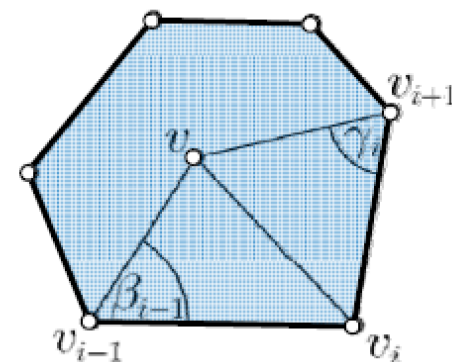
- *mean value* (MV) coordinates

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



- *discrete harmonic* (DH) coordinates

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



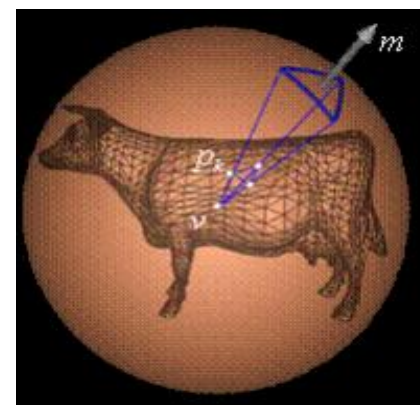
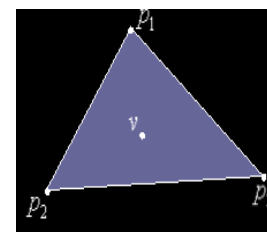
自由编辑 (Free-form deformation)

- 基于重心坐标的编辑

- 三角形重心坐标

$$v = w_1 p_1 + w_2 p_2 + w_3 p_3$$

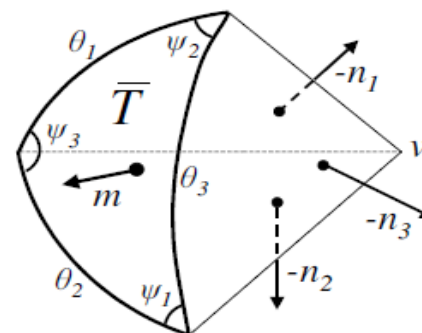
$$w_i = \frac{\text{area}(\Delta v p_j p_k)}{\text{area}(\Delta p_1 p_2 p_3)}$$



- 三角网格重心坐标

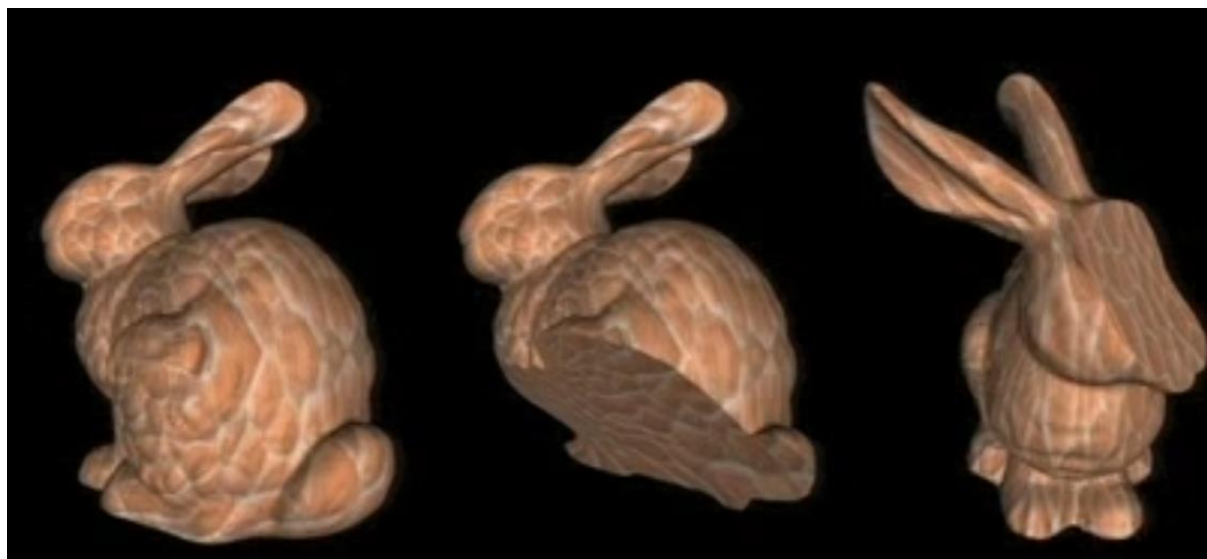
$$v = w_i p_i / \sum w_i$$

$$w_i = \frac{n_i \cdot m}{n_i \cdot (p_i - v)}$$



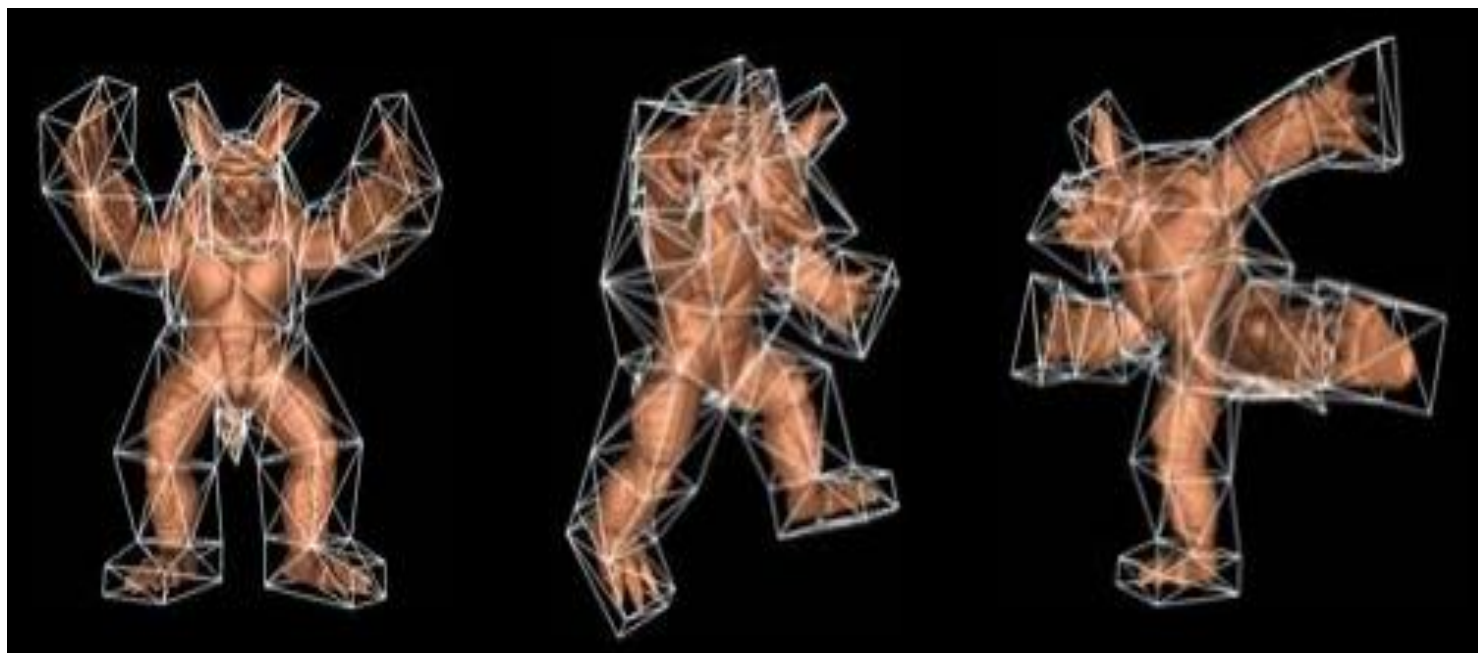
自由编辑 (Free-form deformation)

- 基于重心坐标的编辑
 - 结果



自由编辑 (Free-form deformation)

- 基于重心坐标的编辑
 - 结果



[Demo](#) video1

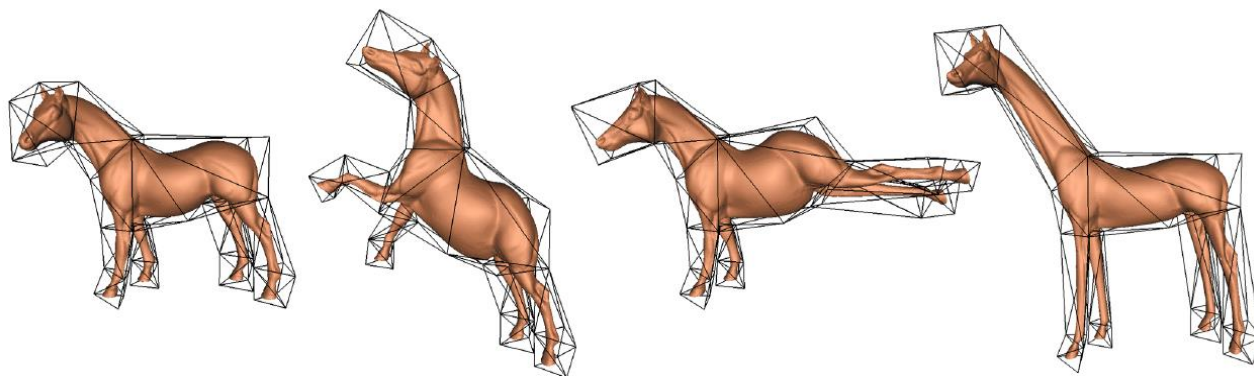
自由编辑 (Free-form deformation)

- 优点

- 任意形状的平滑编辑
- 对编辑的局部控制
- 编辑迅速

- 缺点:

- 难以得到满足物理的编辑效果



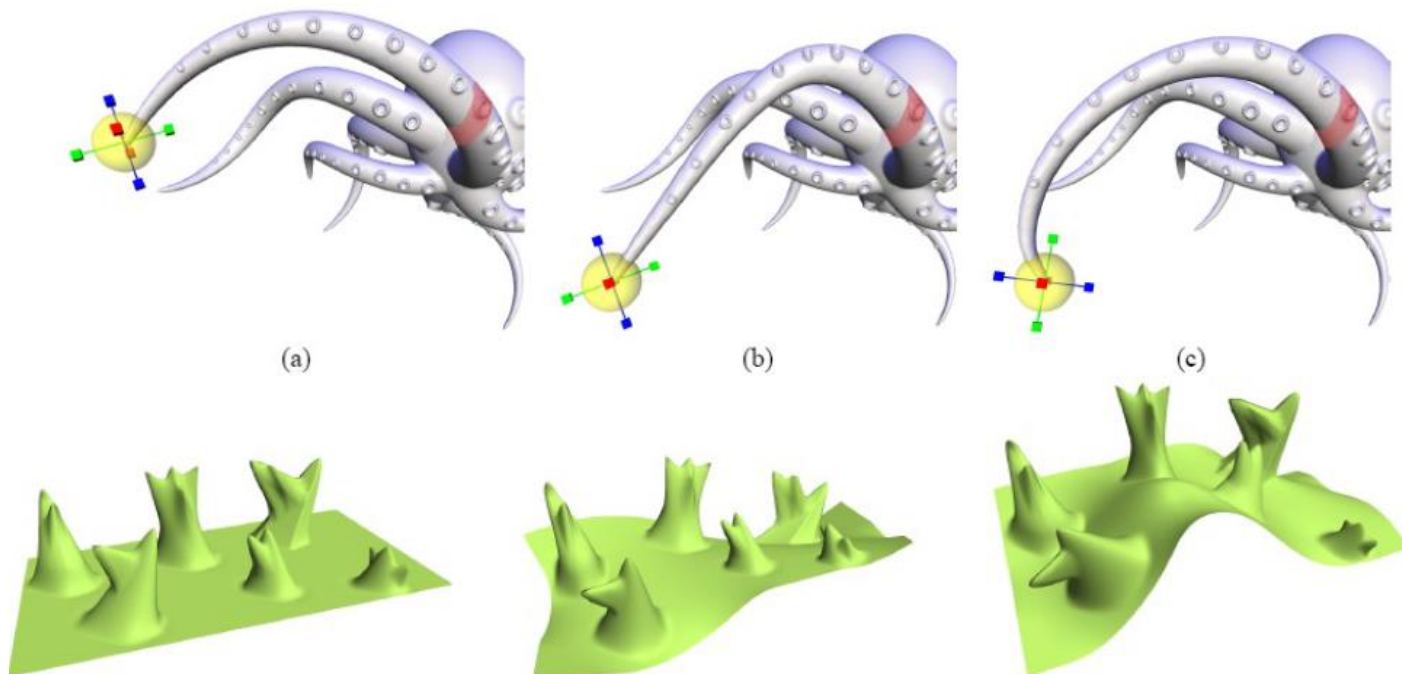
基于约束的编辑

(Constrain-based deformation)

- 思想：交互指定约束类型作为编辑限制
 - 点/区域约束
 - 方向约束
 - 位置约束
- 方法：通过求解满足约束的形状方程计算编辑结果



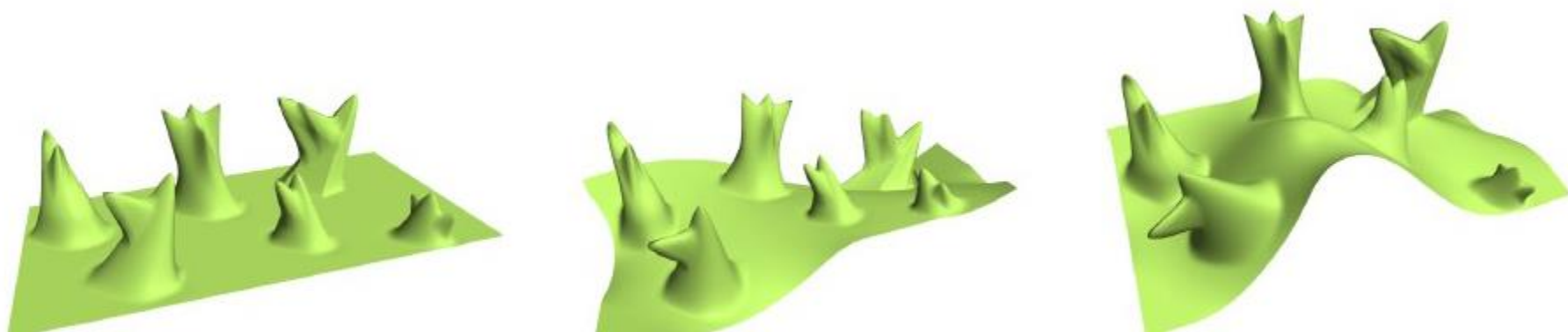
基于约束的编辑



编辑形状

- 一般来讲，编辑形状比重新建模更省力
- 编辑形状要考虑哪些方面的因素？
- 保持局部形状不变

如同对于人脸模型来说，眼睛、鼻子、嘴等部分在形变过程中必须保持原来的形状

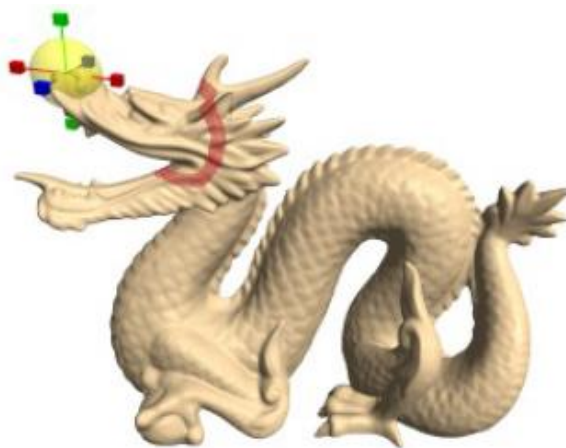


Laplacian 网格编辑

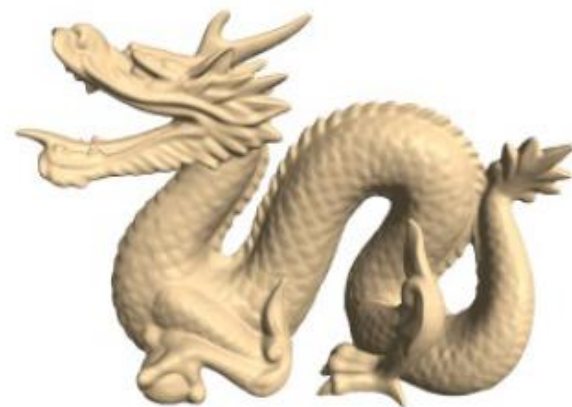
- 用户指定要编辑的位置和区域 (ROI)
- 用户移动模型上的某些点
- ROI上其他部分的位置由最小化误差函数来确定



(a)



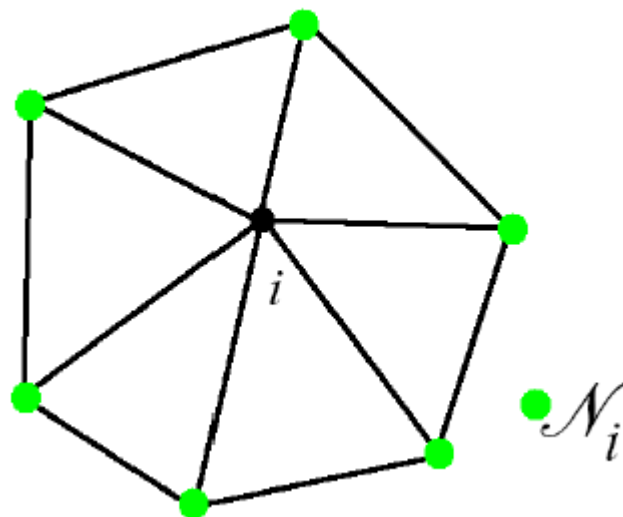
(b)



(c)

Laplacian坐标

- 假设所有的多边形都是三角形
- 每个顶点都被一组邻居节点包围
- Laplacian坐标由 V_i 与邻居顶点的坐标平均值之差来决定

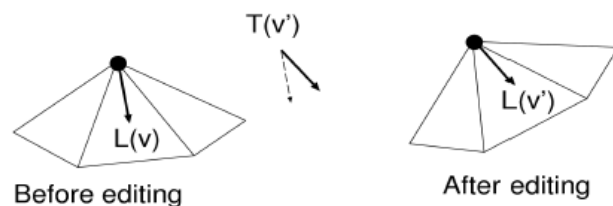


$$\mathcal{L}(\mathbf{v}_i) = \mathbf{v}_i - \frac{1}{d_i} \sum_{j \in \mathcal{N}_i} \mathbf{v}_j$$

Laplacian坐标

- 假设新的位置设为 \mathbf{v}'_i
- 我们希望变换过后该点的Laplacian坐标不变

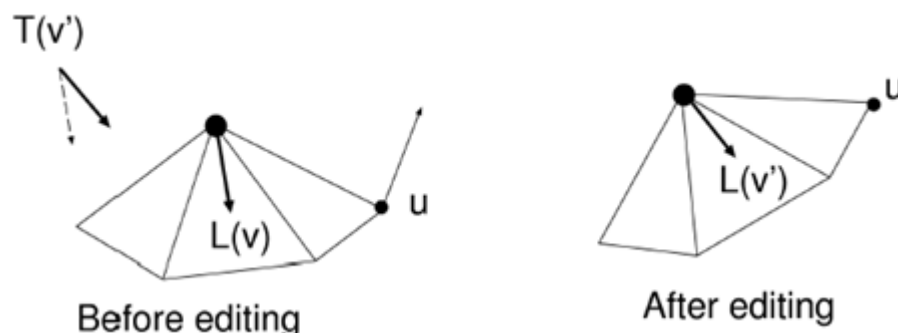
$$\delta_i = \mathcal{L}(\mathbf{v}_i) .$$



$$E(V') = \sum_{i=1}^n \|T_i(V')\delta_i - \mathcal{L}(\mathbf{v}'_i)\|^2$$

- T 是一个齐次变换矩阵

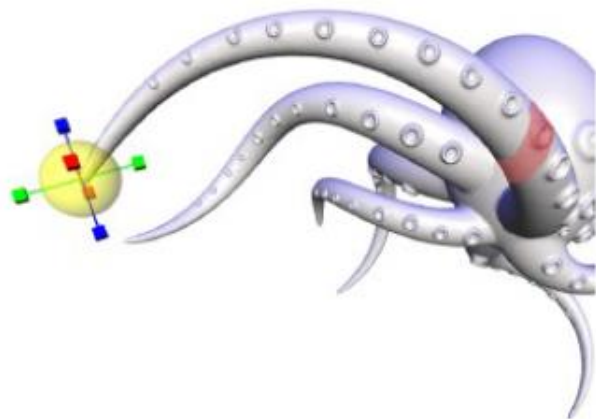
Laplacian坐标



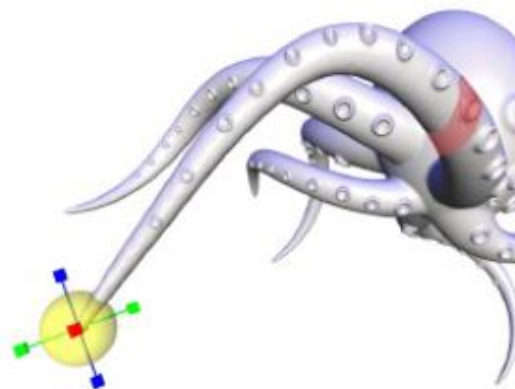
- 我们也希望模型上的某些部分不发生形变
- 保持 v_i' 距离 u_i 不是太远
- 最终我们要求使得下式最小化的 V'

$$E(V') = \sum_{i=1}^n \|T_i(V')\delta_i - \mathcal{L}(\mathbf{v}_i')\|^2 + \sum_{i=m}^n \|\mathbf{v}_i' - \mathbf{u}_i\|^2$$

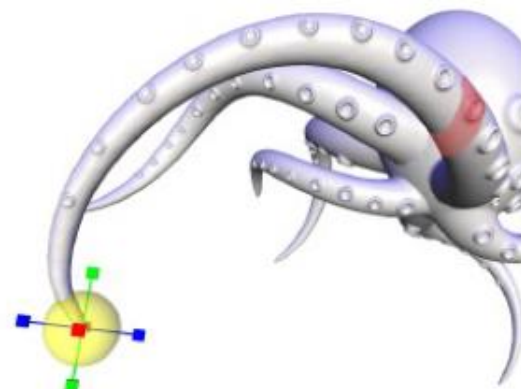
实验结果



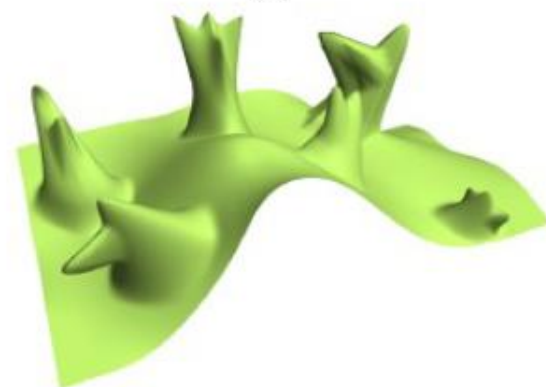
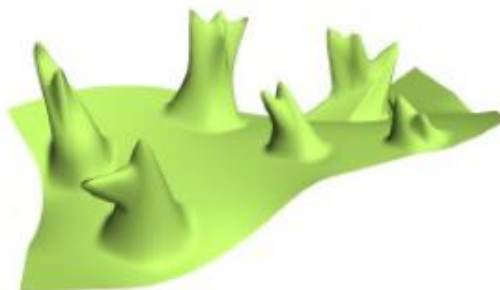
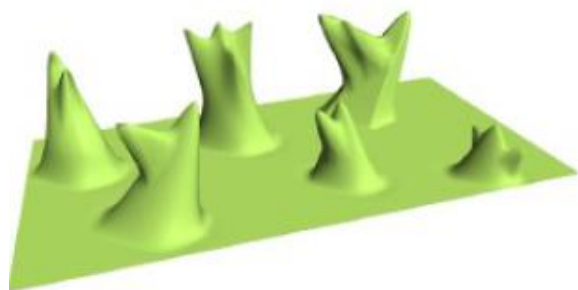
(a)



(b)

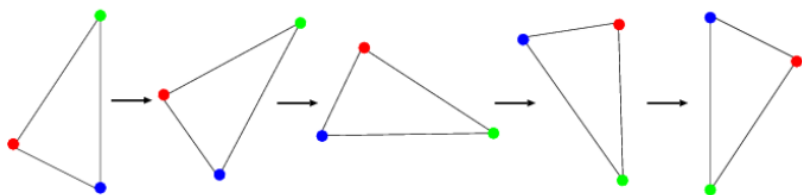
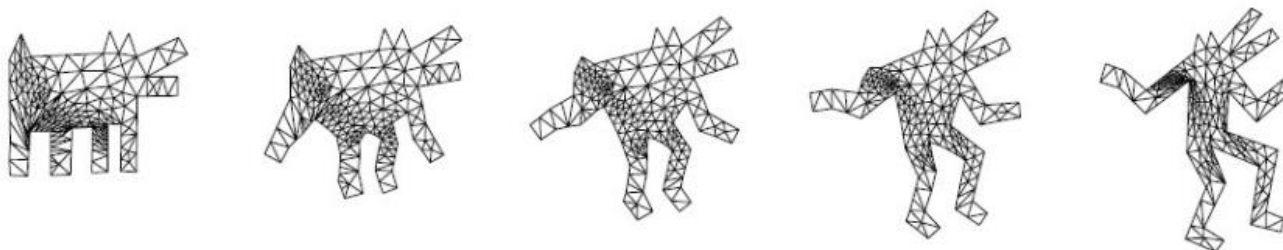


(c)

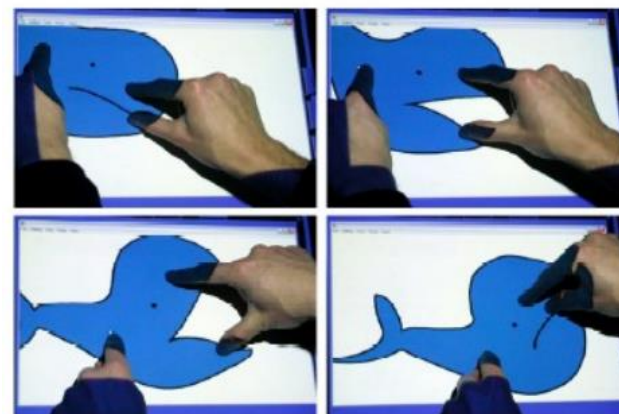


[demo](#)

As rigid as possible 形状保持的图像形变插值



rotation & scaling



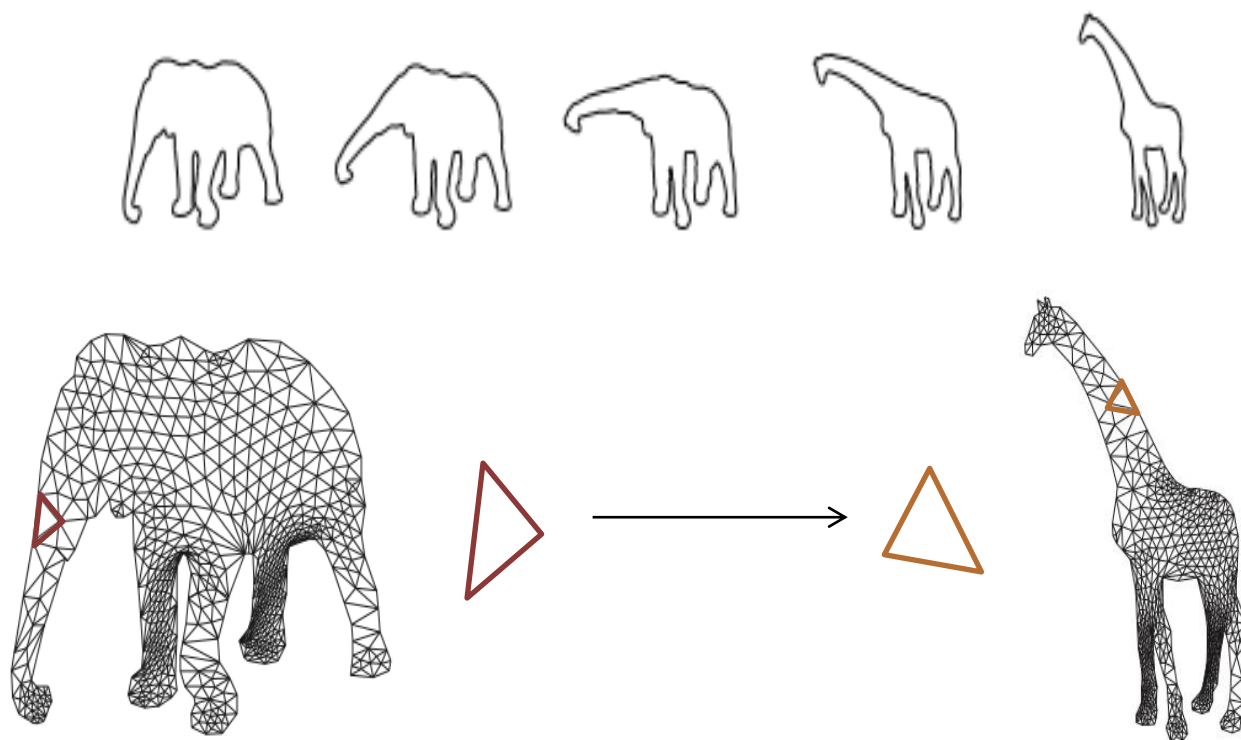
As rigid as possible

- **问题：** 给定起始和终止图像作为关键帧，如何在形变过程中尽可能保持形状较少改变
- **思想：** 利用平面形状的刚性变换进行形变
 - 刚性变换：保持度量（长度+角度）不变的几何变换



As rigid as possible

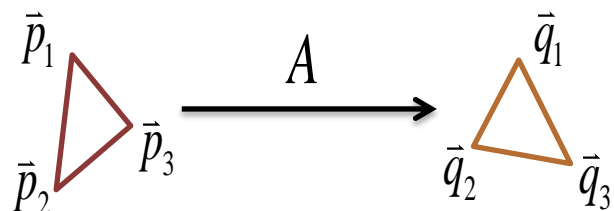
- 方法：在刚性变换空间进行局部矩阵插值
 - 关键帧图像转化为平面三角网格
 - 三角形之间的变换采用刚性变换插值



As rigid as possible

– 方法：在刚性变换空间进行局部矩阵插值

- 通过变换矩阵的极分解计算刚性变换分量和尺度伸缩分量



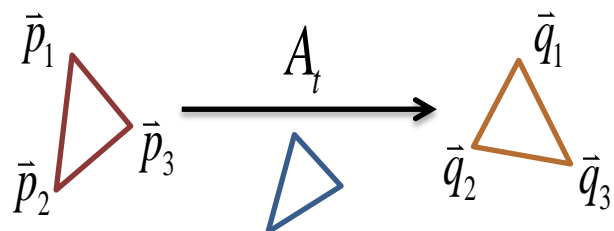
$$\vec{q}_i = A\vec{p}_i + \vec{l} = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \vec{p}_i + \begin{pmatrix} l_x \\ l_y \end{pmatrix}$$

$$A = R_\alpha D R_\beta = (R_\alpha R_\beta) (R_\beta^T D R_\beta) = R_\gamma S$$

As rigid as possible

– 方法：在刚性变换空间进行局部矩阵插值

- 线性插值旋转角度和尺度因子，生成中间变换矩阵



$$A_t = R_{t\gamma} \left((1-t)I + tS \right)$$

顶点线
性插值

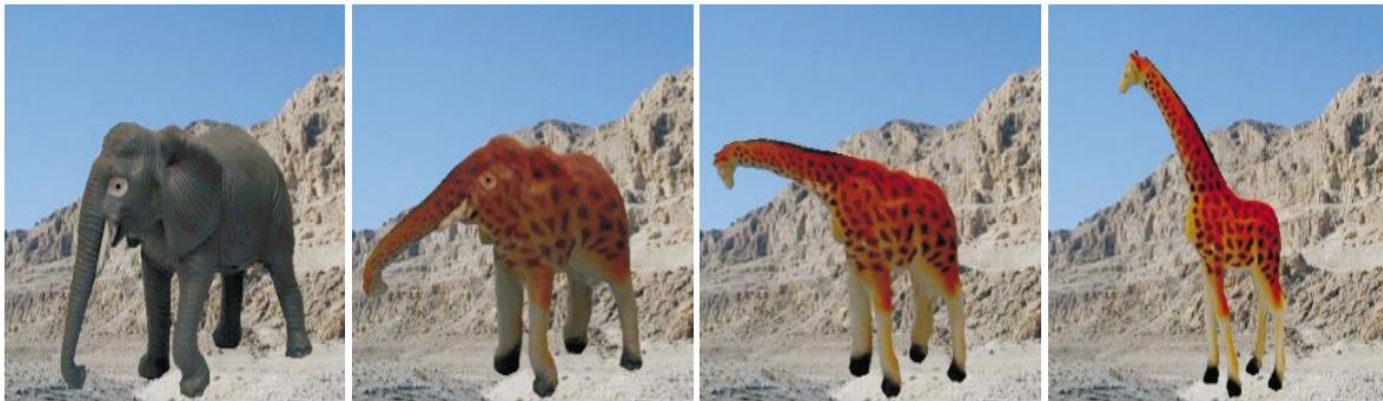


刚性变
换插值



As rigid as possible

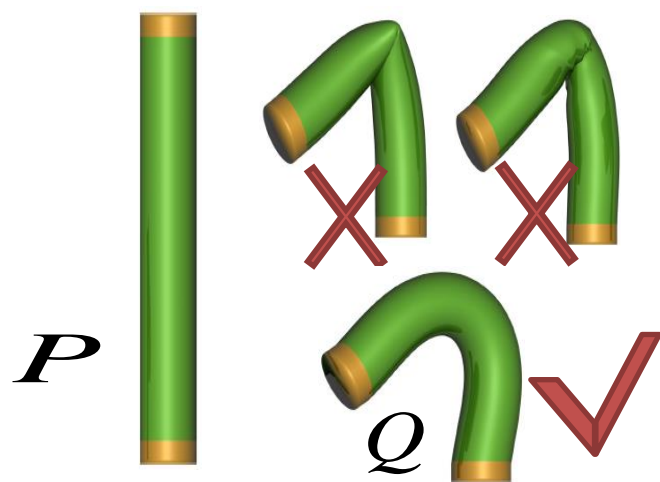
— 结果



[demo](#)

As rigid as possible for 3D shape

- **问题：** 三维网格发生形变时局部保持形状尽可能小的扭曲，以及形变的连续性
- **思想：** 采用As-rigid-as-possible (ARAP) 能量函数，驱使三维网格发生形变时局部遵循刚性变换（旋转）约束

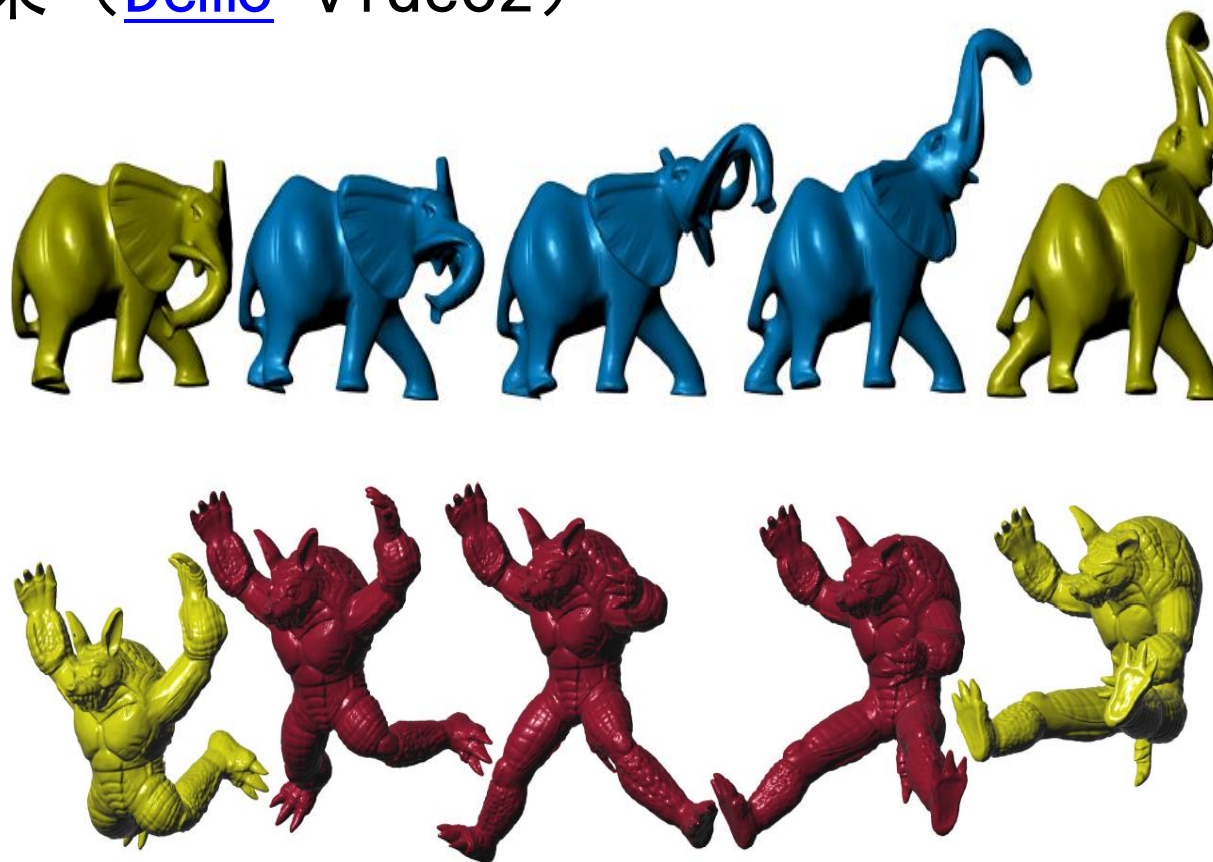


ARAP能量函数形式

$$E(P, Q) = \sum_{k=1}^m \min_{R_k \in SO(3)} \sum_{(i,j) \in \mathcal{E}_k} c_{ijk} \|q_{ij} - R_k p_{ij}\|^2$$

As rigid as possible

- 形状保持的三维网格形变插值
 - 结果 ([Demo](#) video2)

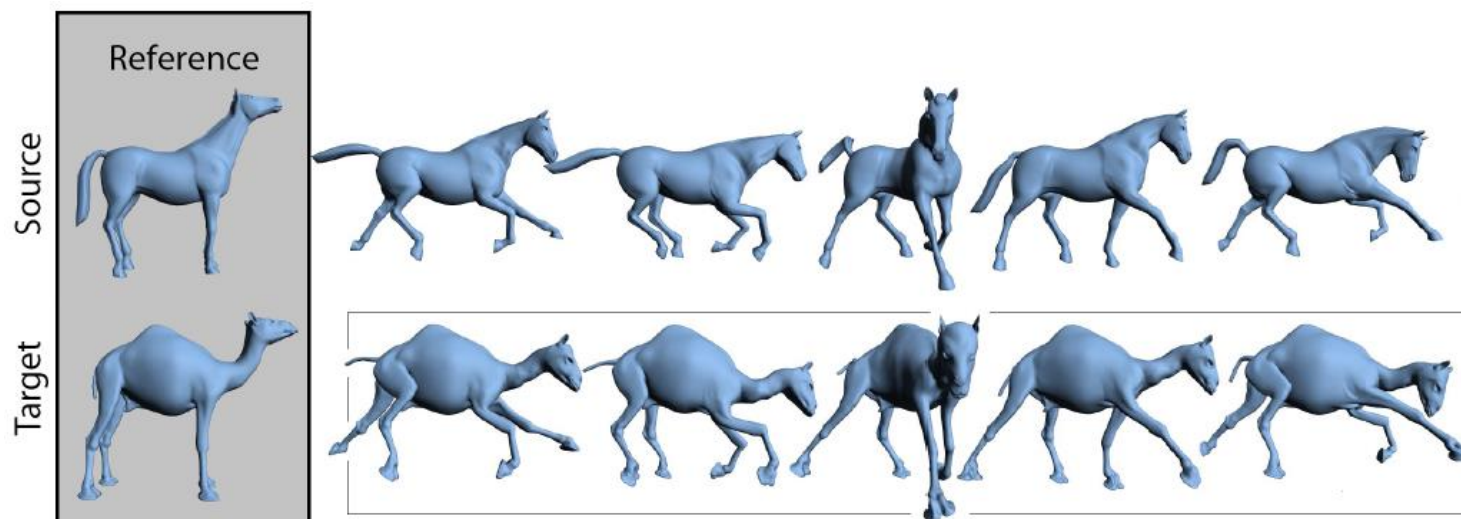


编辑迁移 (Deformation transfer)

— 思想：将网格编辑过程迁移到其他网格

— 方法

- 在源对象和目标之间建立对应的映射
- 将对源对象的编辑迁移到目标，快速生成目标的编辑效果



编辑迁移 (Deformation transfer)

– 思想：将网格编辑过程迁移到其他网格

– 方法

- 在源对象和目标之间建立对应的映射
- 将对源对象的编辑迁移到目标，快速生成目标的编辑效果

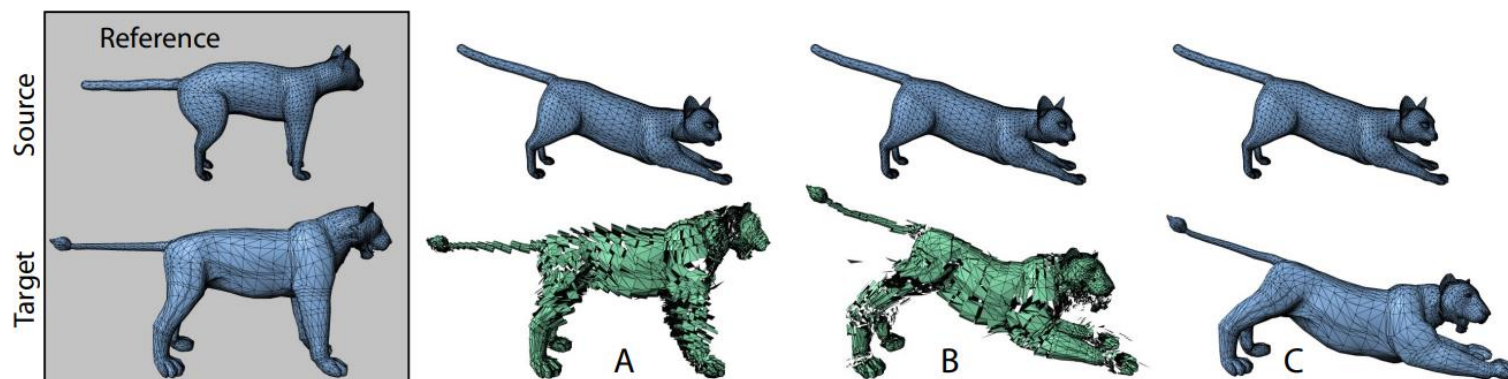


Figure 2: We encode a source deformation with an affine transformation for each source triangle and relate the transformations to the target through a user supplied triangle correspondence. (A) Using only the non-translational component of the source transformations transfers the change in orientation and scale to the target triangles but does not position them appropriately relative to their neighbors. (B) Using the source displacements gives a disconnected shape since consistency requirements are not enforced. (C) Deformation transfer solves a constrained optimization problem for a new set of target transformations that are as close as possible to the source transformations while enforcing the consistency requirements: shared vertices must be transformed to the same place.

骨架驱动编辑

(Skeleton-driven deformation)

- 通过改变骨架驱动网格形状变化
- 方便动画师更加灵活地控制具有骨架结构的物体变形



[Demo](#) video3