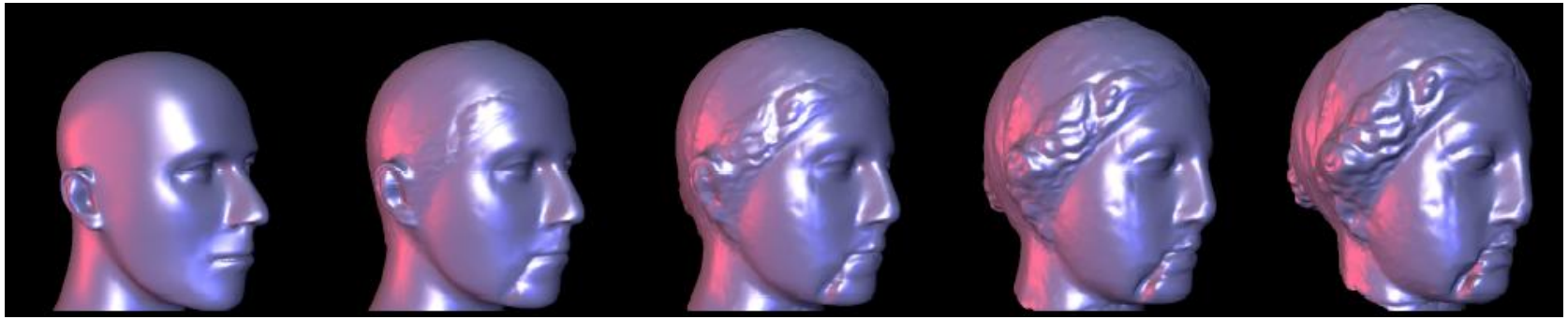
三维morphing技术



金小刚

Email: jin@cad.zju.edu.cn

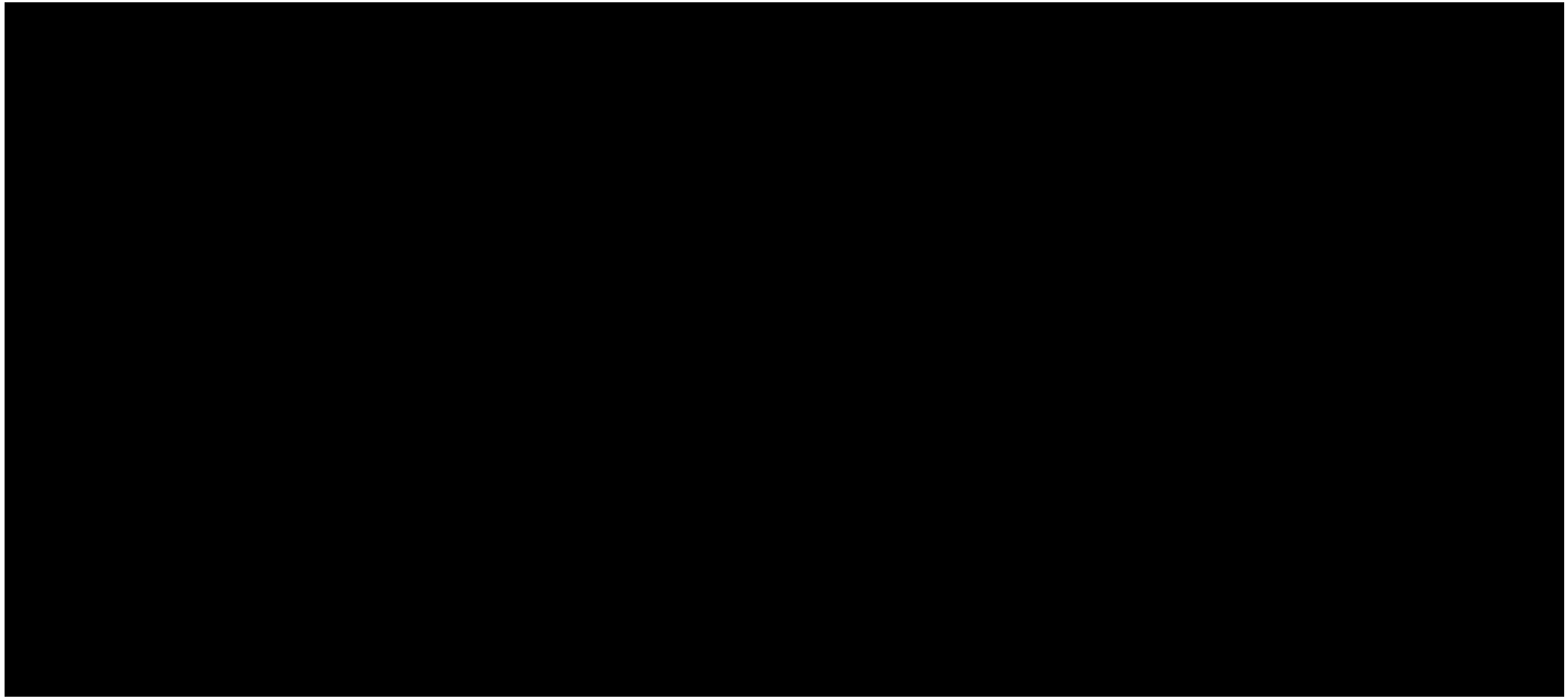
浙江大学CAD&CG国家重点实验室

紫金港校区蒙民伟楼512

演示——Demonic Chick Morphing

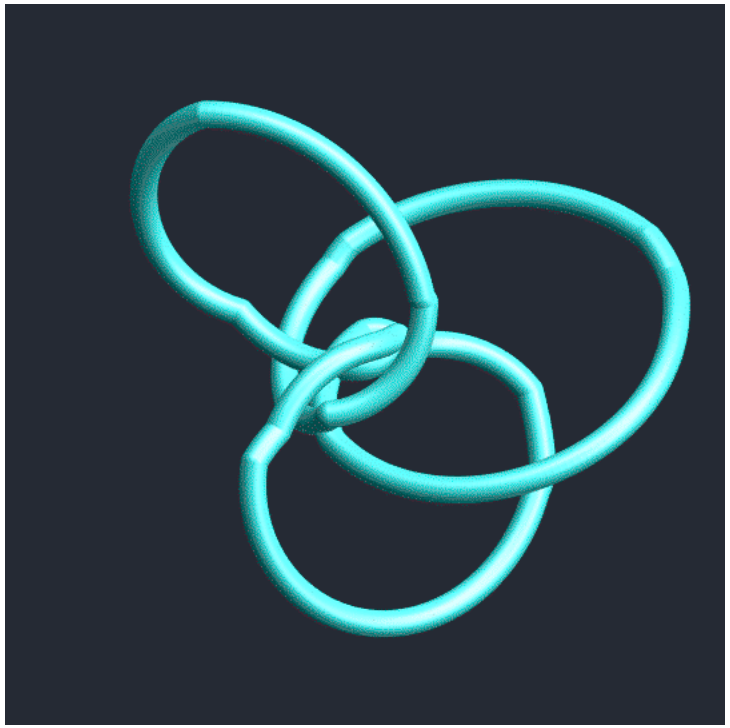
演示——Volkswagen People's Car Project Morphing

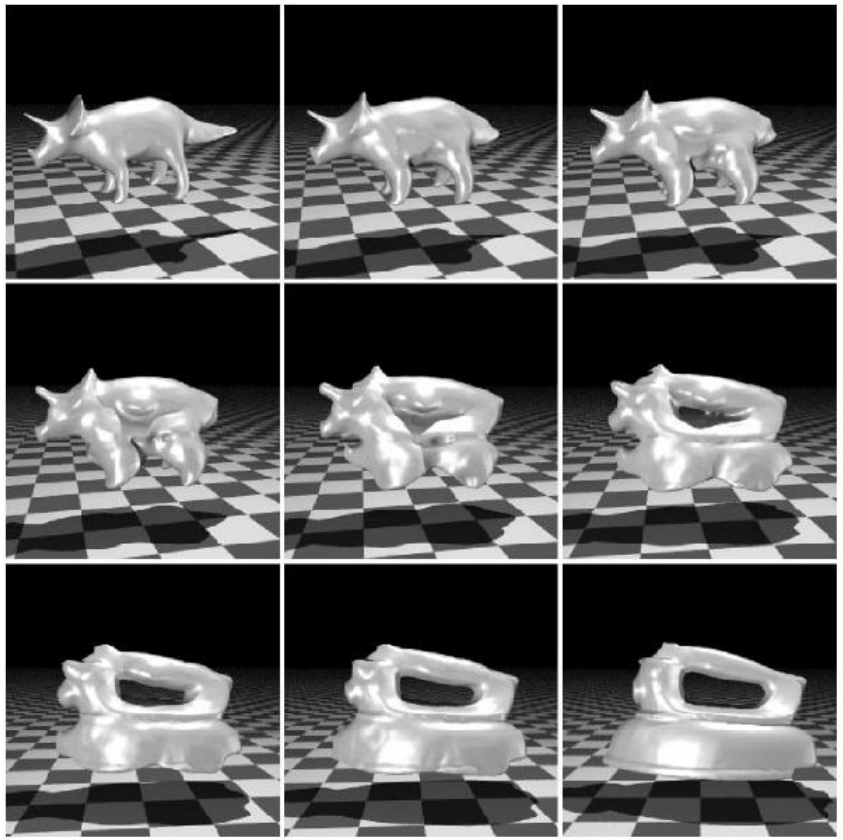
演示——Terminator II



三维morphing技术

- 所谓三维morphing，是指将一个三维物体光滑连续地变换为另一个三维物体。



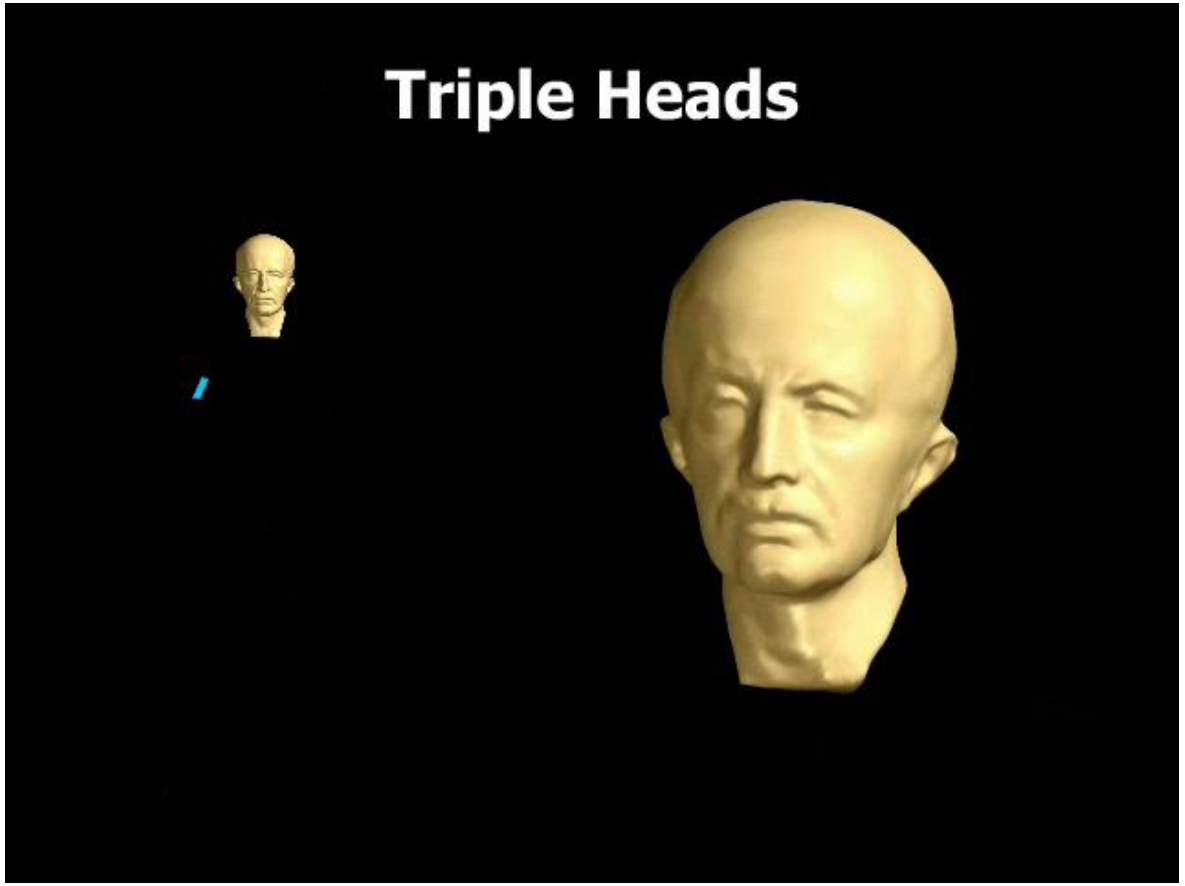


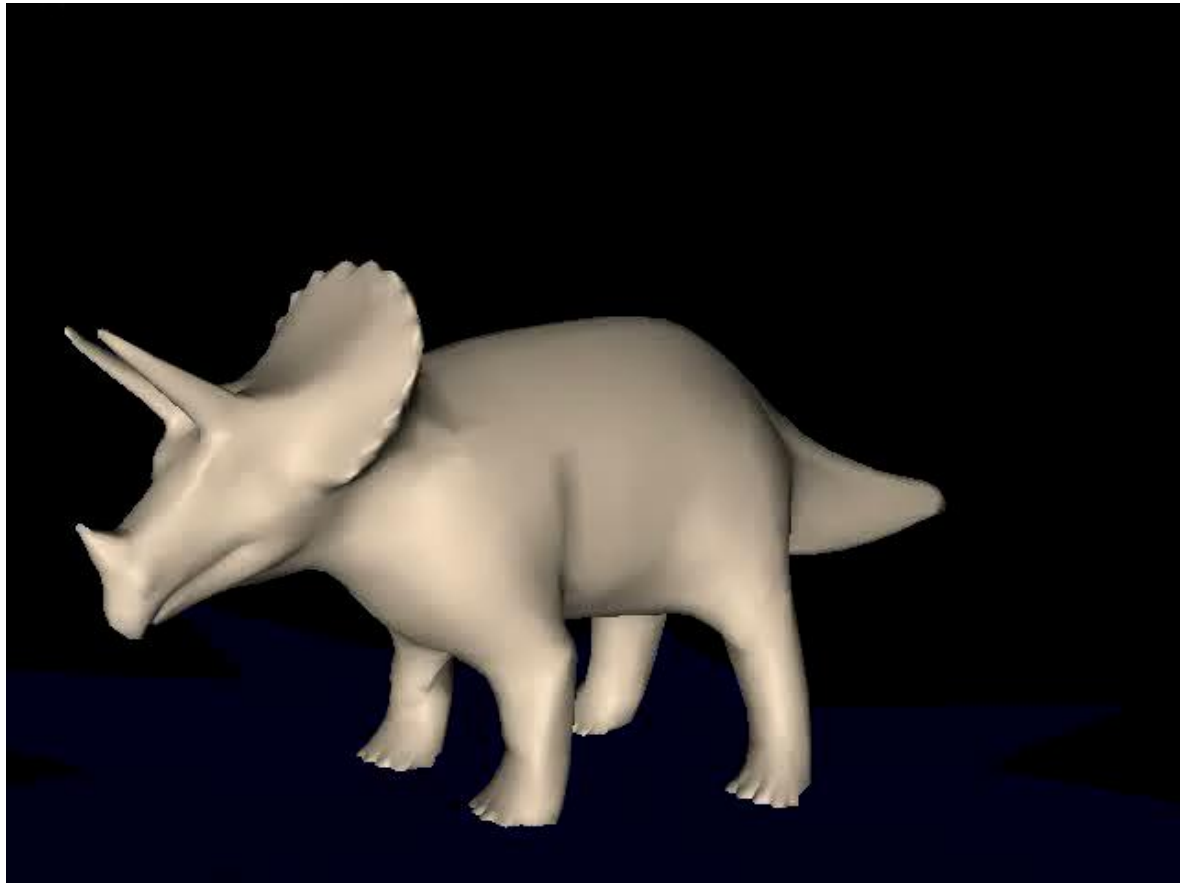
三维morphing，从一头恐龙变为电熨斗

三维morphing技术

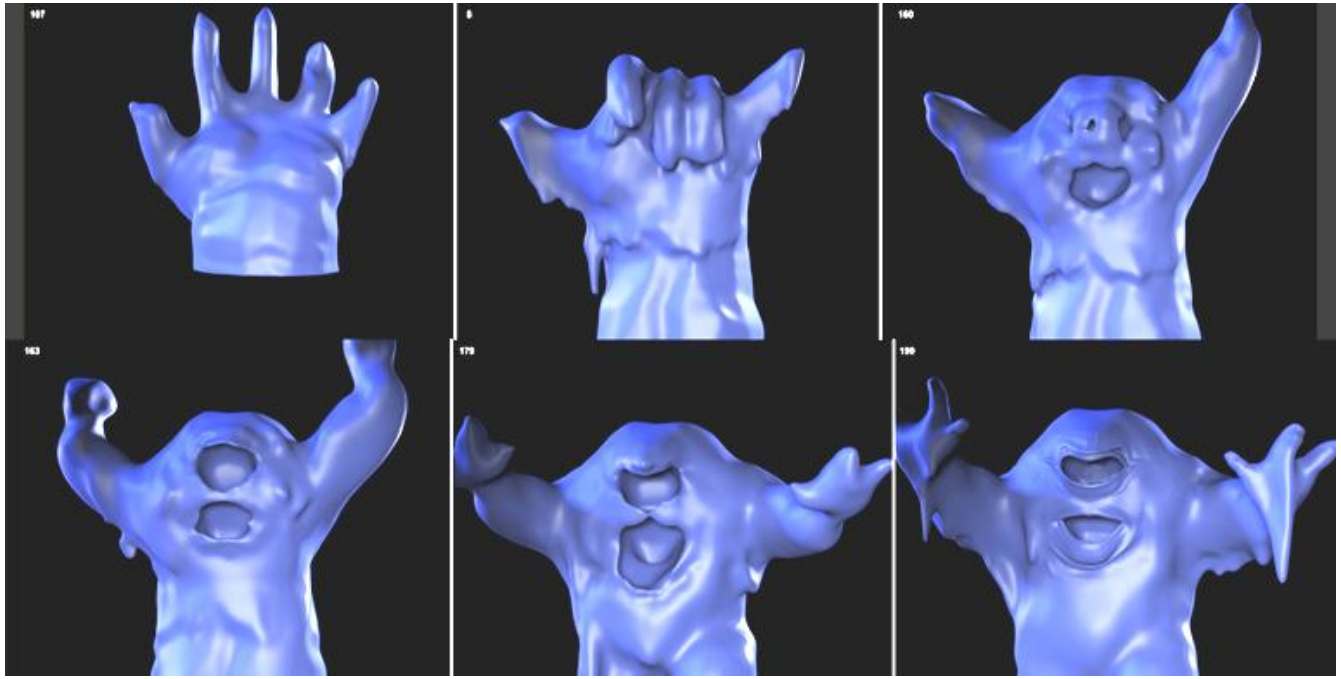
- 虽然二维morphing技术在影视特技、广告等行业中取得了广泛的应用，但二维图象对应的物体存在没有三维几何信息这一很严重的局限性，故它不能象其他三维物体一样进行几何变换，从而使摄象机的动画受到了很大的限制。
- 尽管三维morphing比二维图象morphing要复杂得多，但由于能生成更逼真和生动的特技效果，所以它还是吸引了许多研究者。
- 与二维图象morphing相比，三维morphing得到的中间帧是物体的模型而不是图象，所以三维morphing的结果与视点和光照参数无关，并能够生成精确的光照和阴影效果。在三维morphing中，一旦得到中间帧物体序列，就可以用不同的摄象机角度和光照条件来对它们进行重新绘制，也可以把它们与其它的三维场景相结合进行绘制。

DEMO



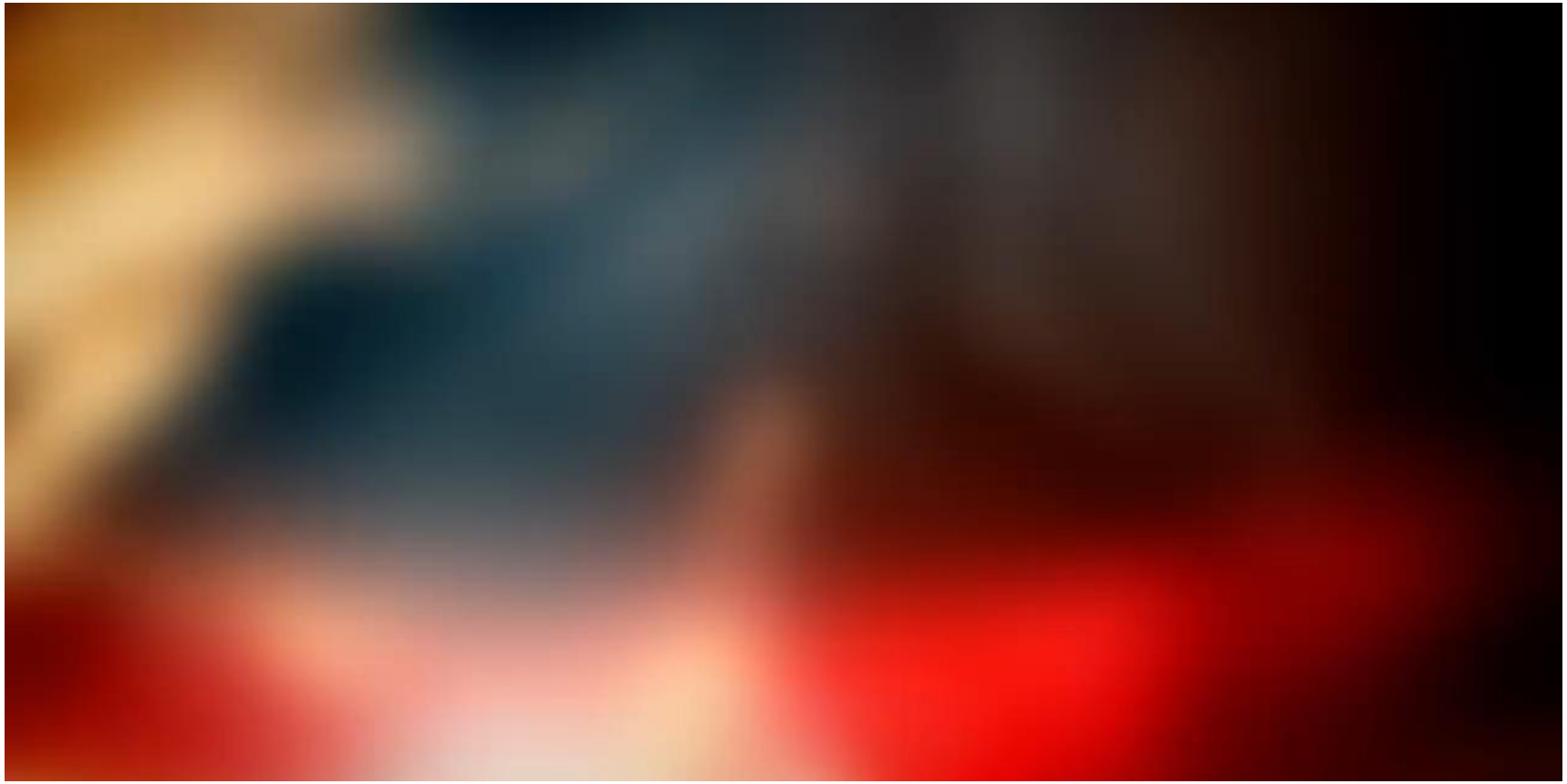


应用实例——《史酷比2：怪兽偷跑》



应用实例——《西游•降魔篇》

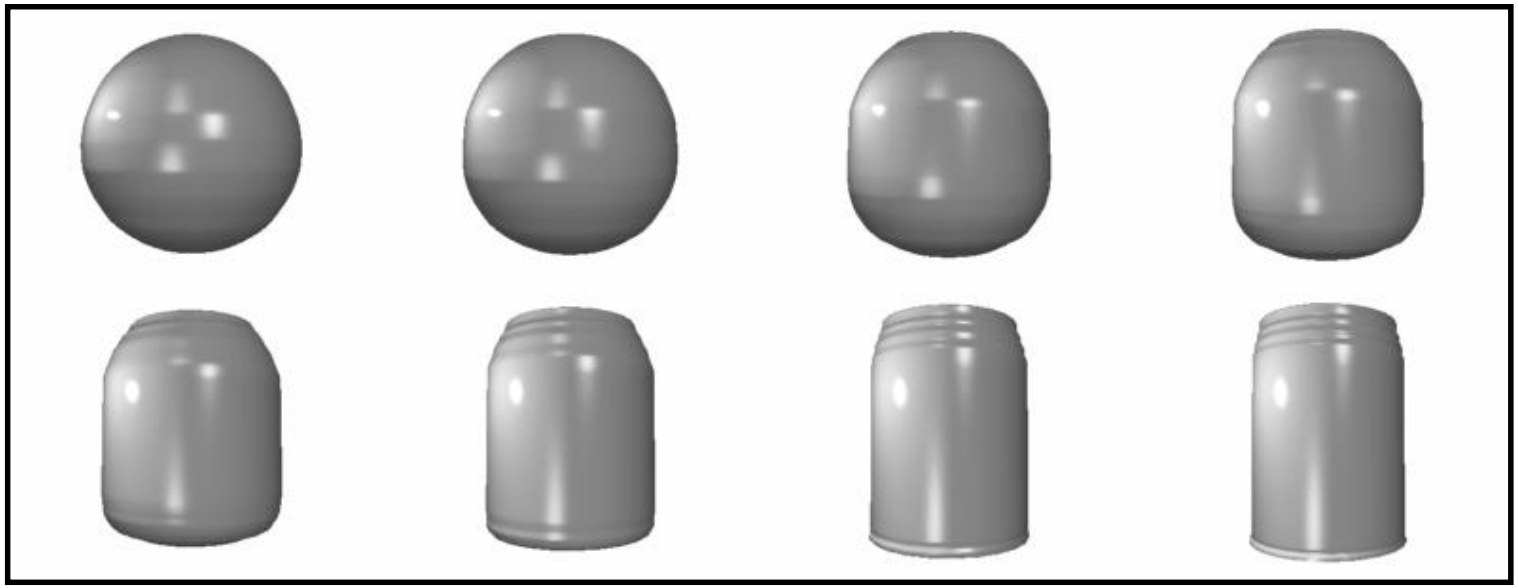
演示——Transformer 7(变形金刚7:超能勇士崛起)



三维morphing技术分类

- 基于表面(surface)表示的
  - 顶点与顶点的对应关系
  - 顶点之间的插值
  - 通常对模型具有拓扑一致性的限制
- 基于体(volume)表示的
  - 要把表面表示转化为体表示（有可能带来失真）
  - 计算量较费

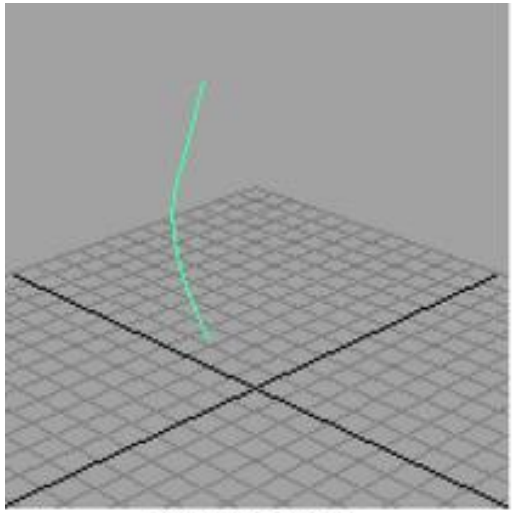
一个简单的3D Morphing例子



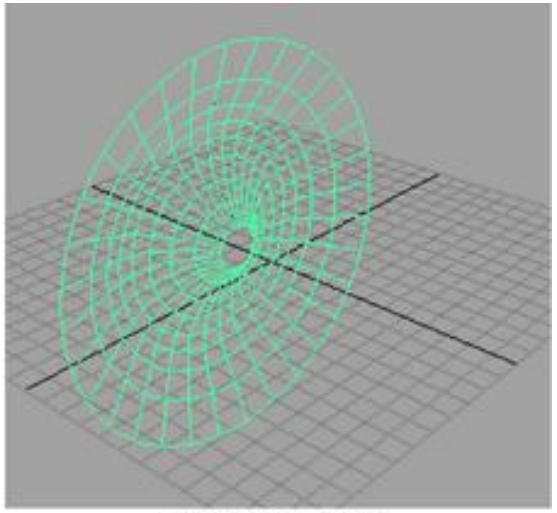
球到易拉罐的三维morphing过程

- 当给定两个物体的**顶点数和拓扑结构都相同时**，只需对对应顶点进行插值便可实现三维morphing过程。
- 但通常这个条件并不满足，因而需要寻找一些更一般的三维morphing方法。

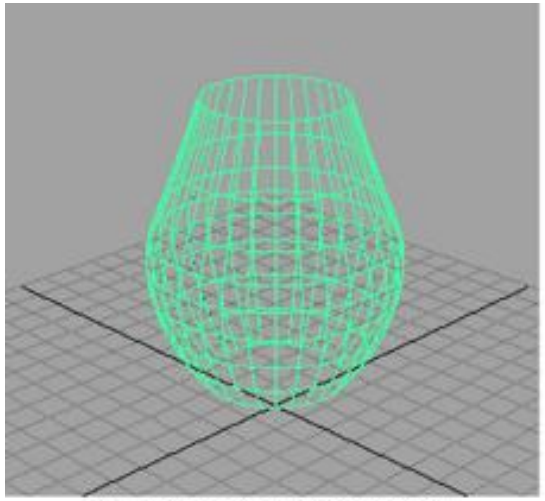
一个简单的3D Morphing例子（旋转体构建）



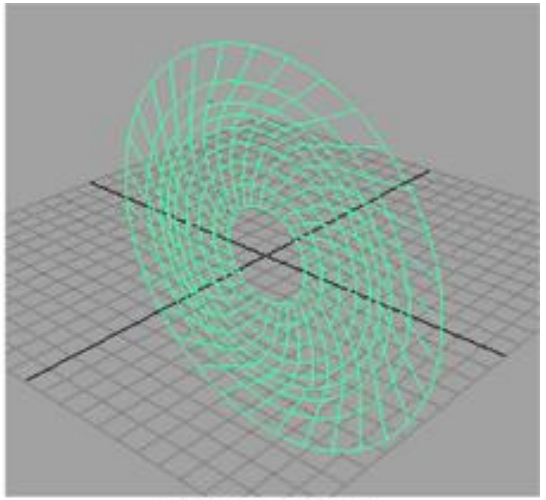
Profile curve



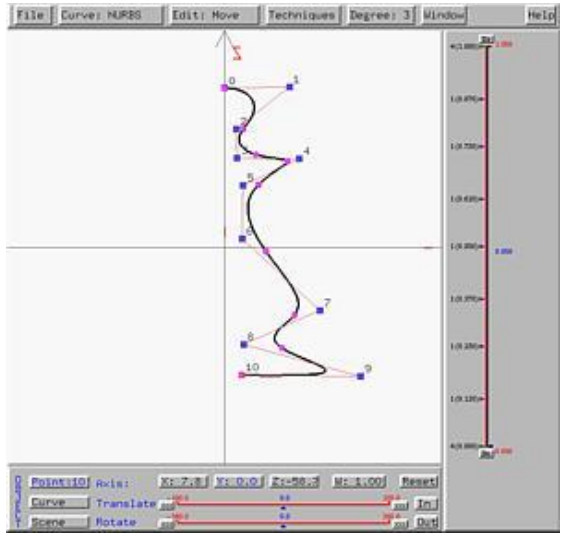
Revolved in X

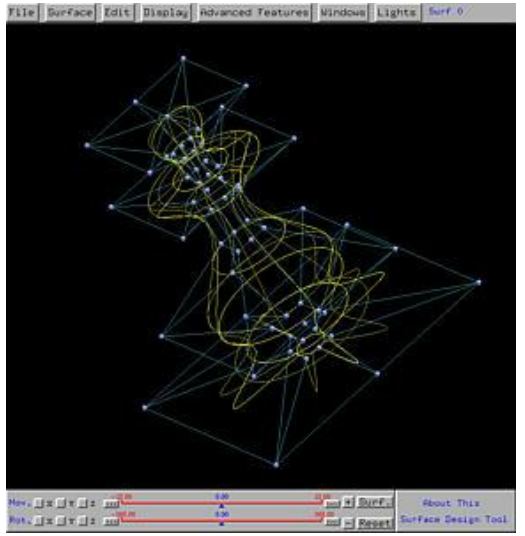


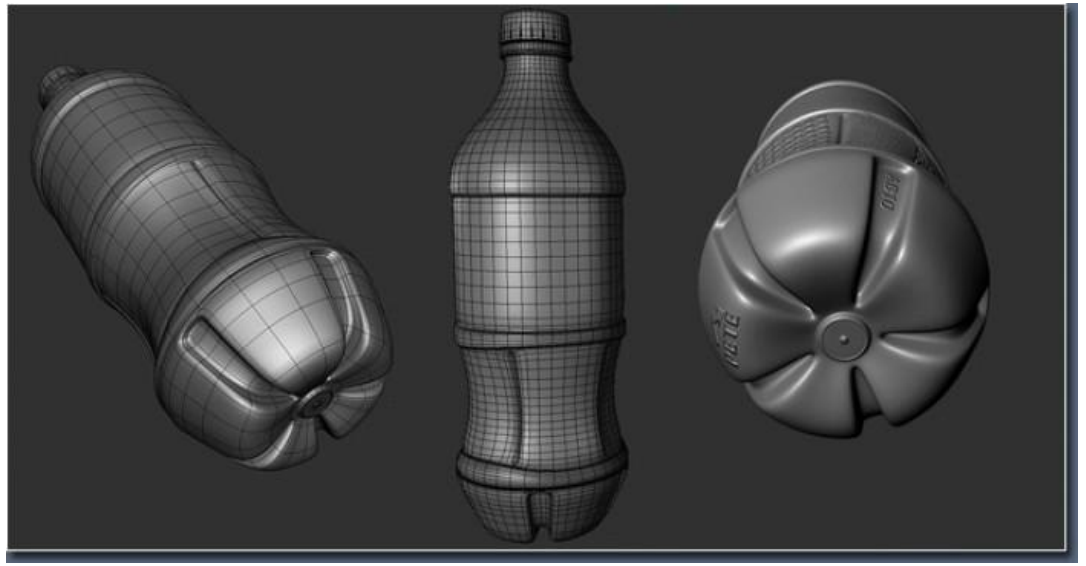
Revolved in Y (default)



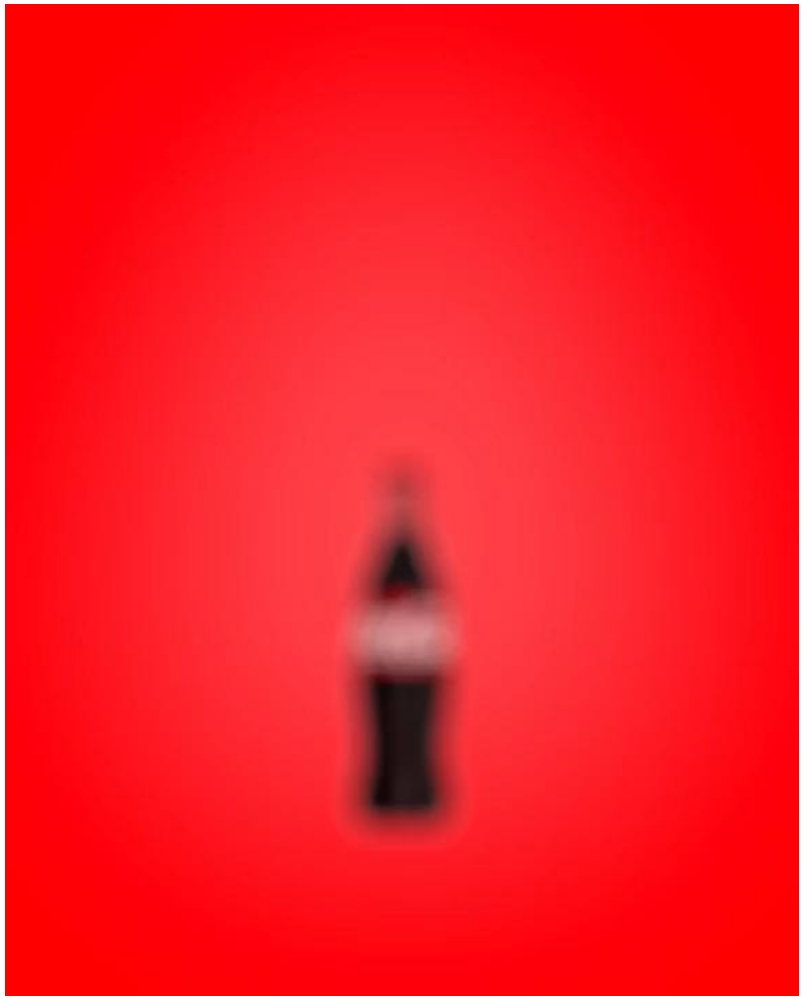
Revolved in Z







一个简单的3D Morphing例子（旋转体构建）



基于星形物体的多面体morphing方法

- 1992年，Kent、Carlson和Parent提出了三维多面体物体之间的morphing方法。

- 该方法通过合并一对三维多面体物体模型的拓扑结构，使得它们具有相同的顶点/边/面网络结构，然后对相应的顶点进行插值。

- 参考文献： Kent J R, Carlson W E, Parent R E. Shape transformation for polyhedral objects. ACM Computer Graphics, 1992, 26(2):47~54

基本概念

- 为了讨论方便，我们引进以下概念。
  - 物体（object）：具有三维表面几何的实体。
  - 形状（shape）：在物体空间构成物体表面的点的集合。
  - 模型（model）：对物体形状的完整描述。显然，一个物体可以有很多不同的模型来描述它的形状。
  - 拓扑（topology）：指一个模型的顶点/边/面结构。
  - 几何（geometry）：拓扑的一种范例（instance），这可通过指定顶点的坐标得到。

基本概念

- 拓扑元素：顶点、边、面。
- 同构（homeomorphic）或拓扑等价（topologically equivalent）：两个物体被称为是同构的如果两个物体表面上的点存在一个连续、可逆的一一映射。
- 欧拉有效（Euler Valid）：如果一个物体的拓扑满足广义欧拉公式 $V-E+F=2-2G$，则称该物体为欧拉有效。其中V、E、F分别为拓扑网络结构的顶点、边和面数目，G为物体的亏格（genus）。
  - 对于一个四面体，$V=4$，$E=6$，$F=4$，$G=0$，显然它是欧拉有效的。

基本思想

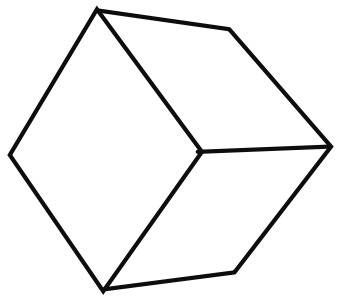
- 通过合并拓扑结构将一个物体的形状变换为另外一个物体的形状通常可分为以下两步：

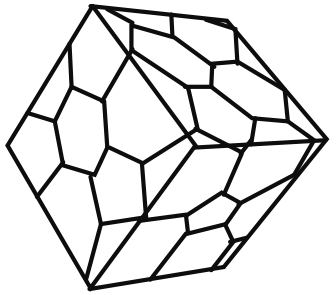
  1. 建立源物体表面上的点与目标物体上的点对应关系；
  2. 插值对应的点。

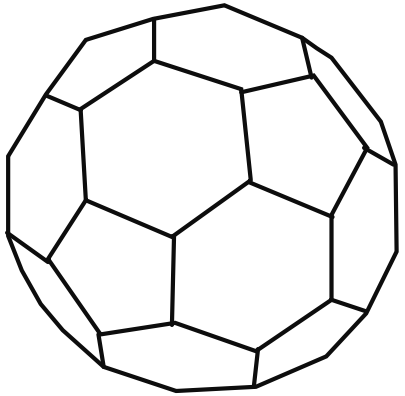
- 其中第一步称为对应问题，第二步称为插值问题。
- 在三维morphing中，建立点的对应关系是难点，而插值问题相对来说比较简单。

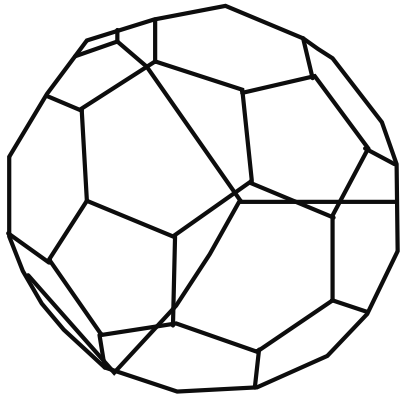
- 我们介绍一将种欧拉有效、亏格为0的多面体之间的形状渐变方法。显然，对于多面体物体，我们只需指定顶点的对应关系，而不必指定物体表面上每一点的对应关系。

建立顶点对应关系









(a) 原始模型　　(b) 合并拓扑后的模型

顶点对应关系算法的一个例子

建立顶点对应关系

- 对于亏格为零的两个多面体，显然它们都同构于球。

- 设想它们象气球一样可充气，我们把它们都充成球形，则物体表面上的点与球面上的点存在着一一对应关系（**一种直观的参数化方法！**）。

- 如果来自一个物体的点V_1与来自另一个物体的点V_2映射到球面上的同一点，则设定V_1与V_2相对应。这就是合并拓扑结构的基本思想。

建立顶点对应关系

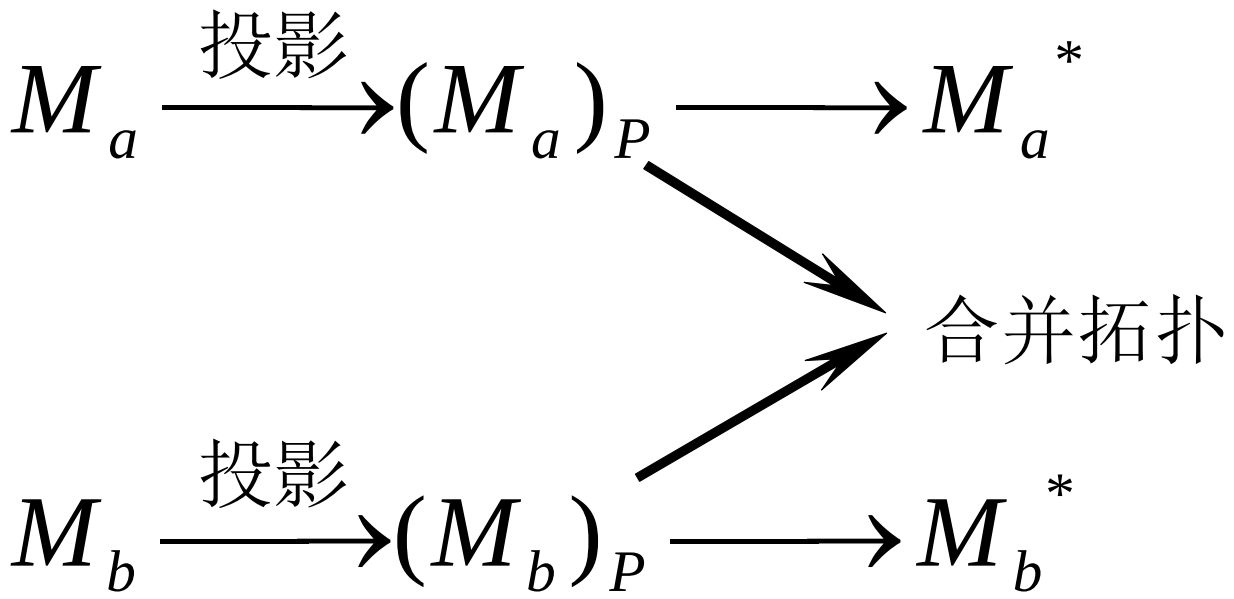
- 基于上述思想，可采取下面的三个步骤来建立亏格为零的多面体之间的对应关系：
  - 把两个多面体投影到单位球面上；
  - 将投影在单位球面上的两个拓扑结构合并在一起构成一个新的拓扑结构；
  - 将新的拓扑结构映射回原来的两个多面体。

- 这样得到的两个新的模型与原来的模型具有相同的形状，且共享相同的拓扑结构，也就是说建立好了相应的顶点对应关系。

记号

- A、B：原始物体
- M_a、M_b：原始多面体物体的模型
- V_a、E_a、F_a、V_b、E_b、F_b：顶点、边、面
- $(M_a)_P$、$(M_b)_P$：M_a、M_b到单位球面的投影
- $(V_a)_P$、$(V_b)_P$：$(M_a)_P$、$(M_b)_P$的顶点
- 并用小写字母表示特定的拓扑元素，如e_a表示M_a的某条边。

拓扑合并过程

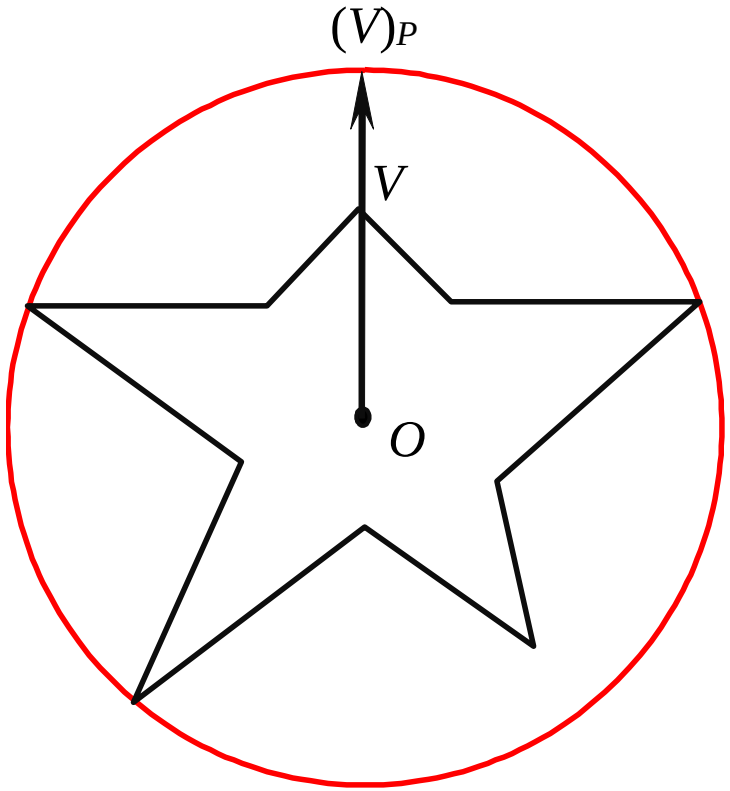


$(M_a)_P$、$(M_b)_P$合并拓扑后，M_a^*和M_b^*具有相同的拓扑结构。

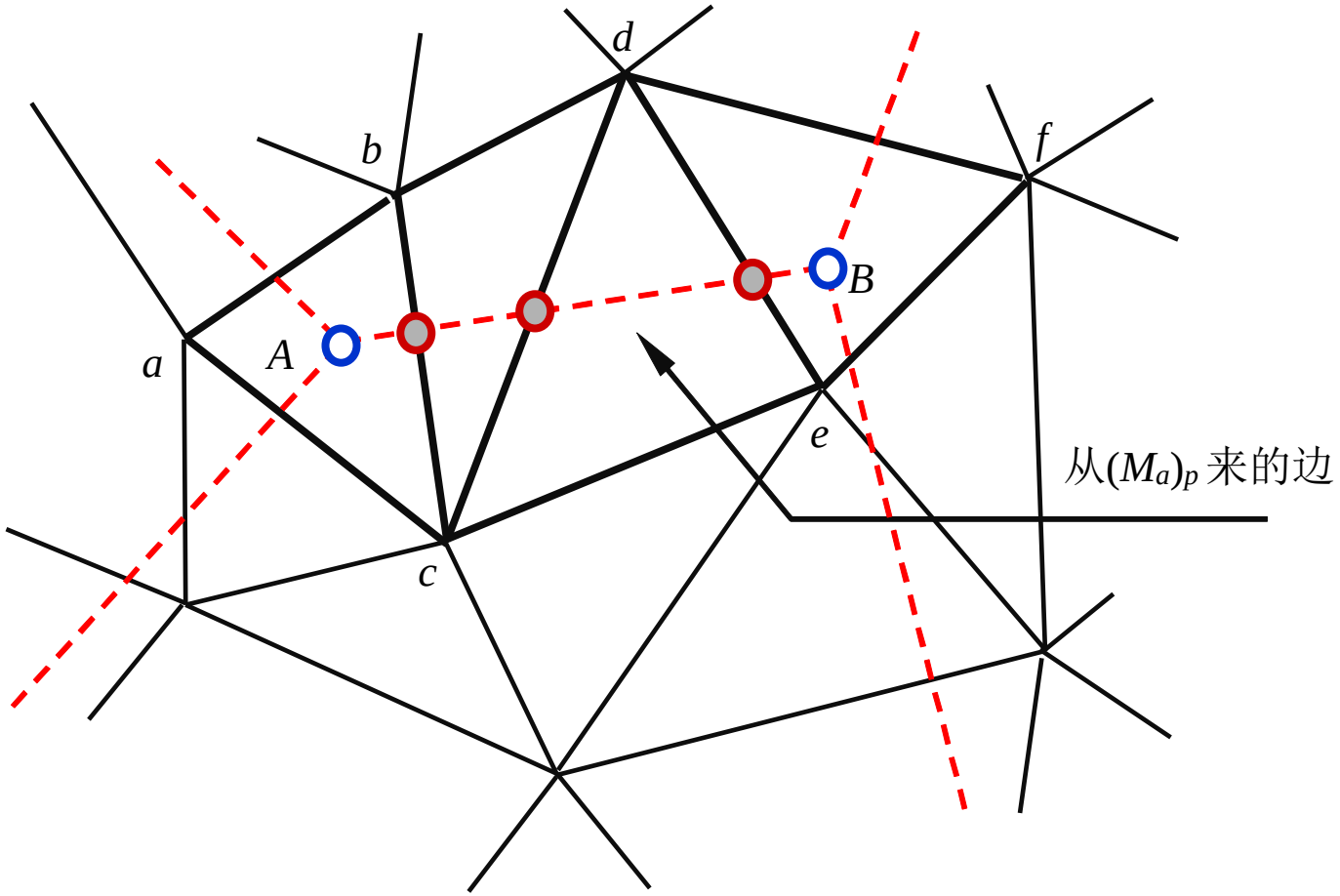
投影方法

- 上述合并拓扑中的投影方法必须足两个条件：
  1. 投影方法必须是一一对应的，即物体表面上的点和单位球面上的点是一一对应。
  2. 投影方法必须是连续的，即很小半经区域内的点投影到很小半经区域内点。
- 在这里，我们讨论凸体和星形物体的投影。所谓星形物体，是指物体中至少有一点，从该点可见多面体的所有顶点。显然，凸体是星形物体的特例，而星形物体则可以是凹体。
- 上述定义实际上提供了一种投影的方法。在物体内选一内点，连接O和物体的顶点V，射线OV和单位球面的交点即为V的投影点$(V)_P$。也就是说，从O点出发，沿OV方向距O点单位距离的点即为V的投影点$(V)_P$。

星形物体的投影方法



拓扑合并算法

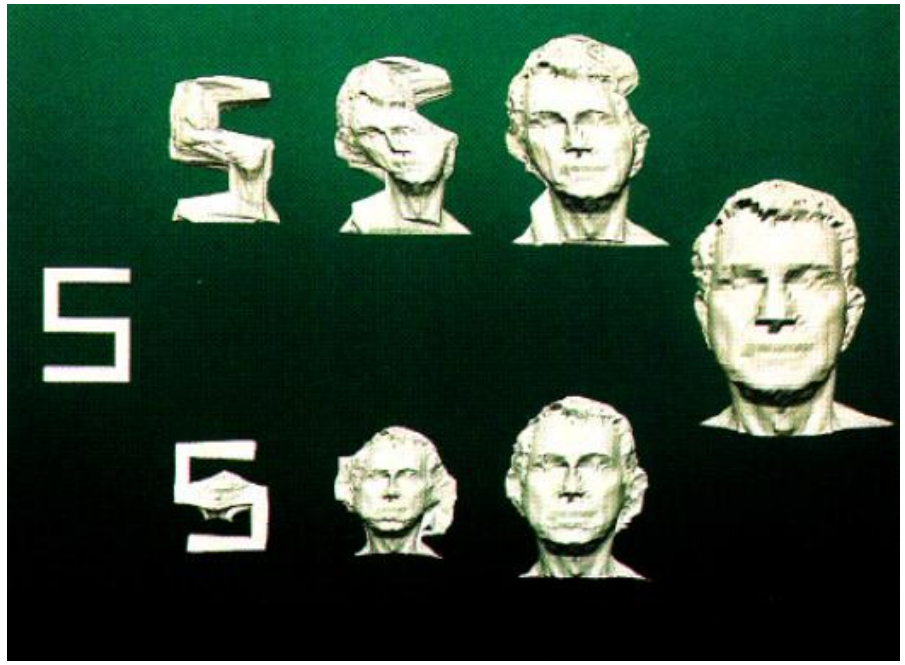


计算一条边的交点

插值问题

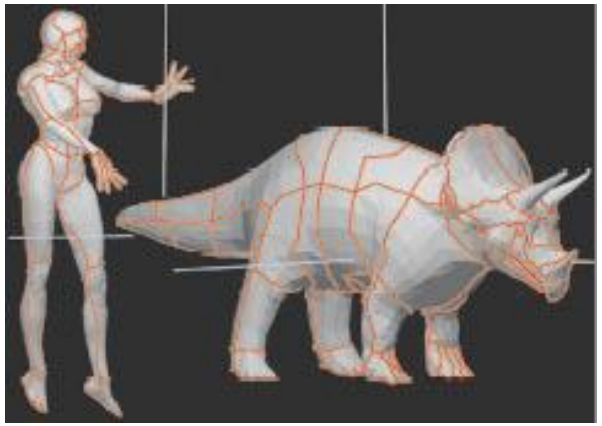
- 一旦顶点对应关系建立以后，通常采用线性插值或Hermite插值来插值相应的顶点。
- 但插值时可能会带来一些问题：
  - 对于边数大于3的面，在顶点插值过程中，原来共面的面有可能变成不共面。一个解决方法是在插值前对合并后的面进行三角化。
  - 插值过程中物体有可能自交。
- 对于如颜色、纹理、透明度等非几何属性的插值，可采用如下方法：
  - 对于中间帧模型中的某一点，计算该点在相应面的面积坐标。
  - 根据该面积坐标，计算出该点在原关键帧物体中相应的点。
  - 插值原关键帧物体中相应点的非几何属性。

结果展示



一般Mesh模型的Morphing

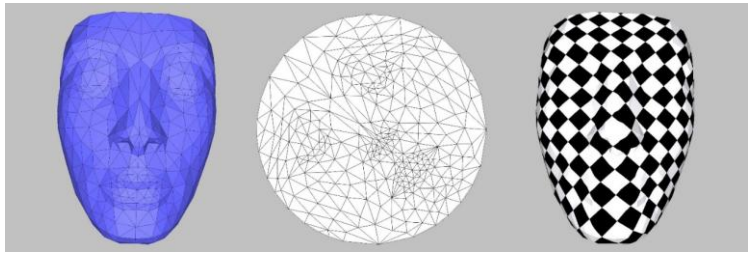
Mesh是最常用的模型。如何提供更一般的参数化方法?



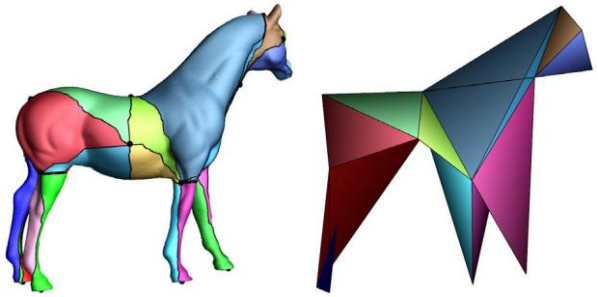
Parameterization（参数化）

- Spherical Parameterization

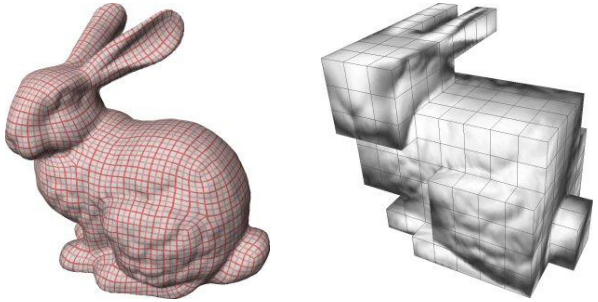
- Planar Parameterization



- Simplicial Parameterization

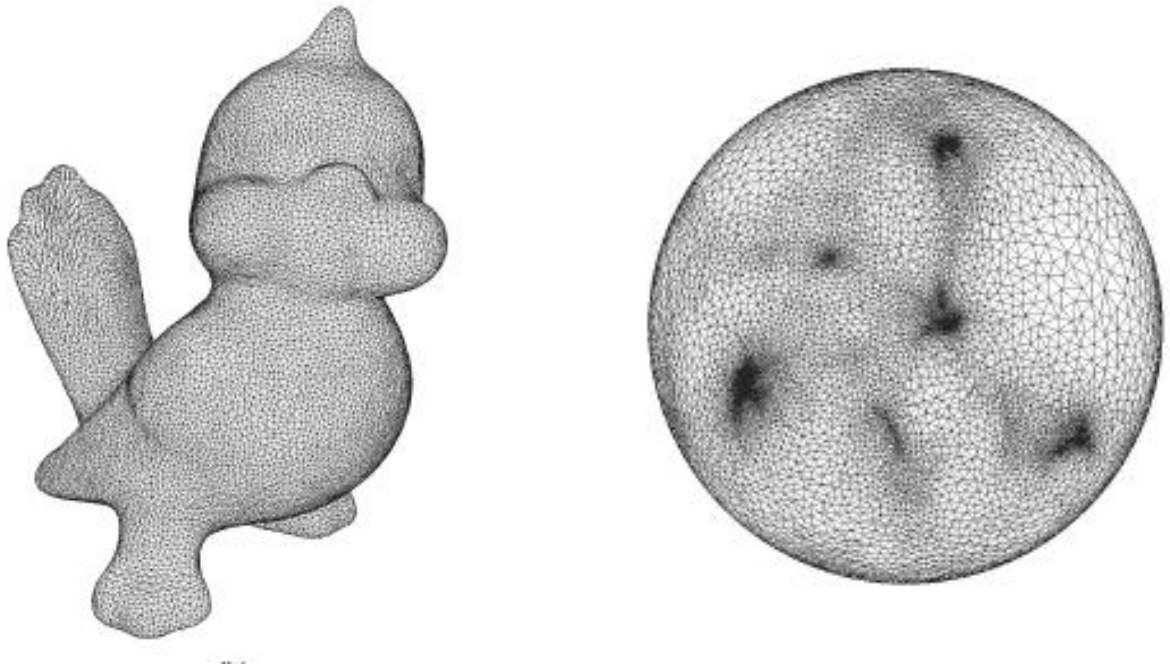


- Polycube Maps



亏格=0的物体：Spherical parameterization

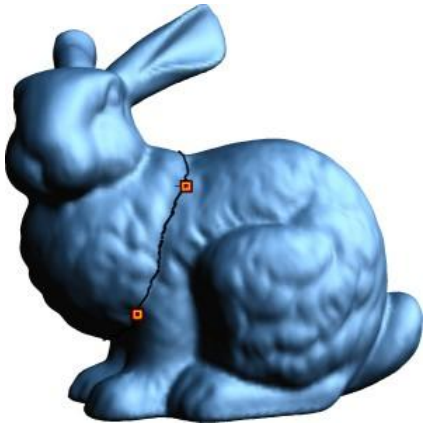
Genus=0

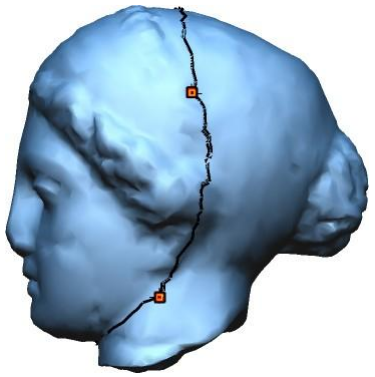


Spherical parameterization

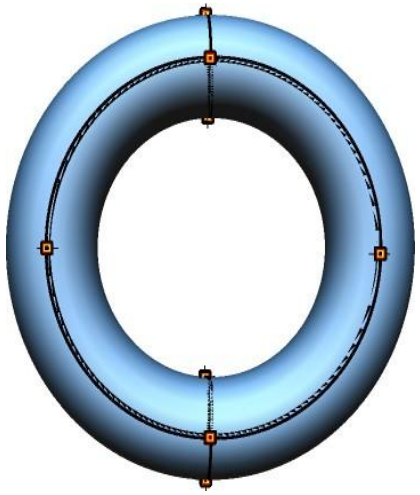
亏格>0的物体： Patch parameterization

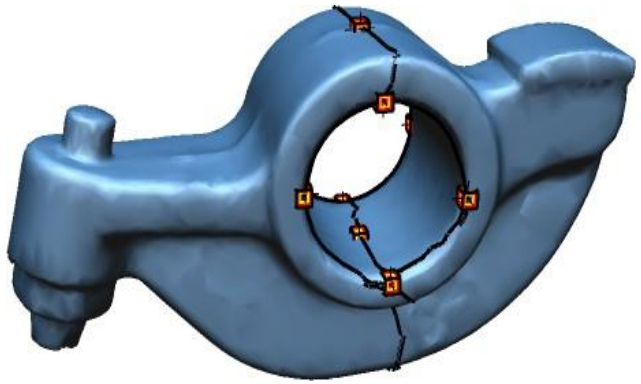
Genus=0

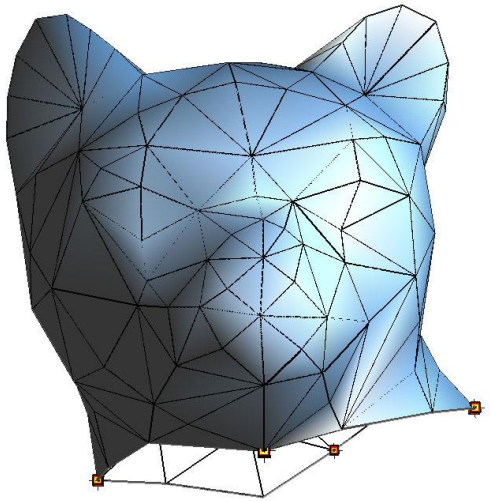


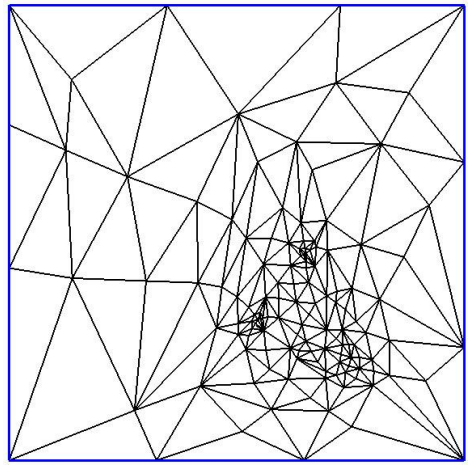


Genus=1



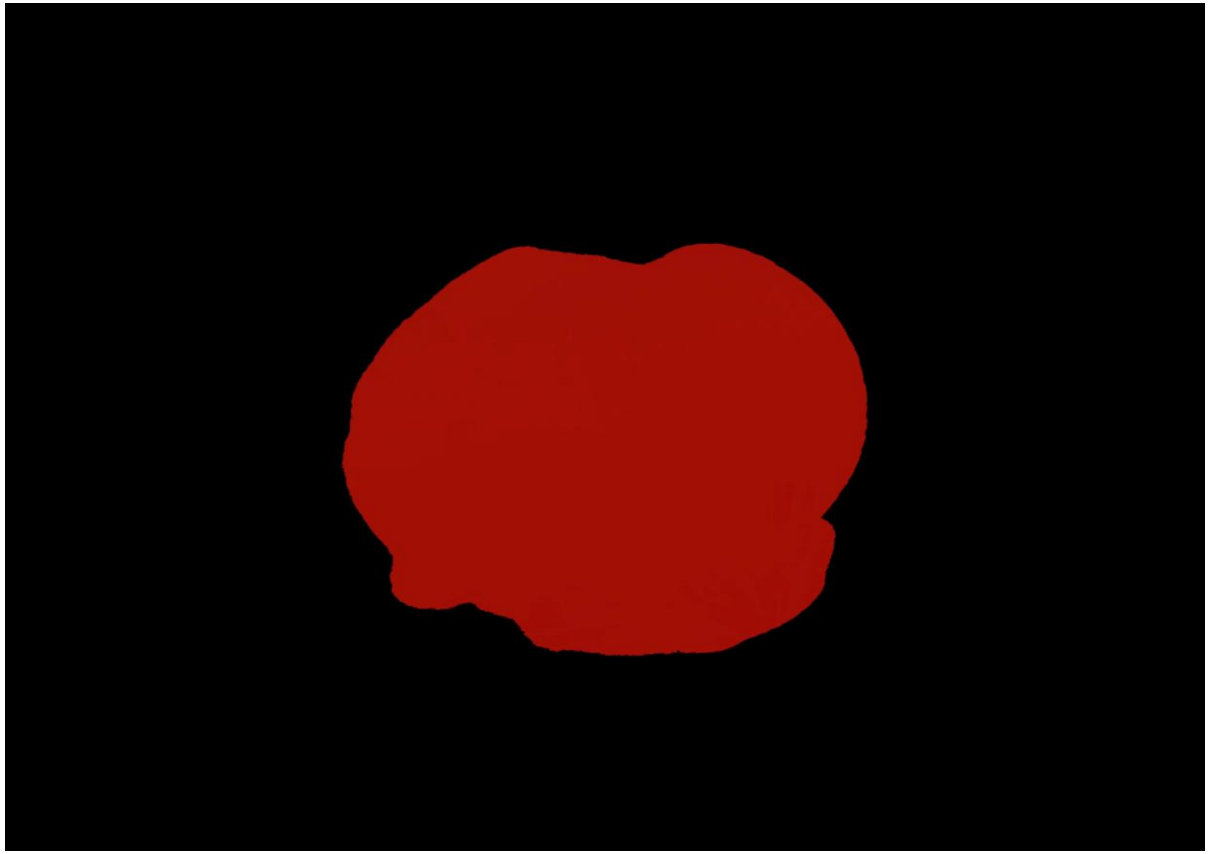




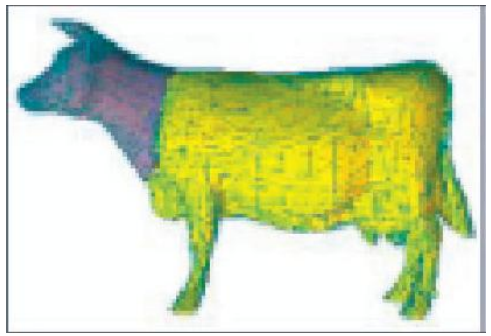


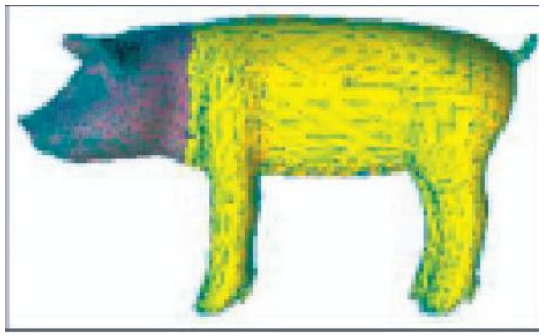
Patch parameterization

Patch parameterization

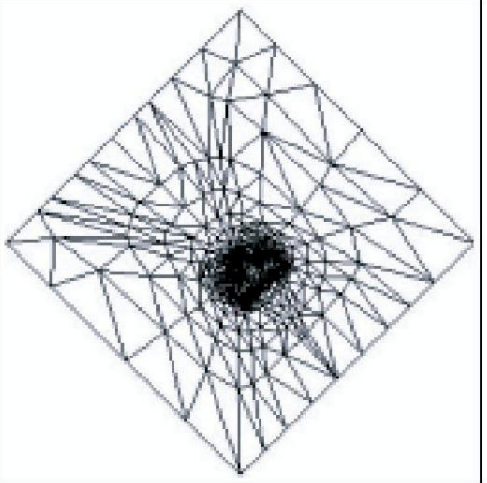
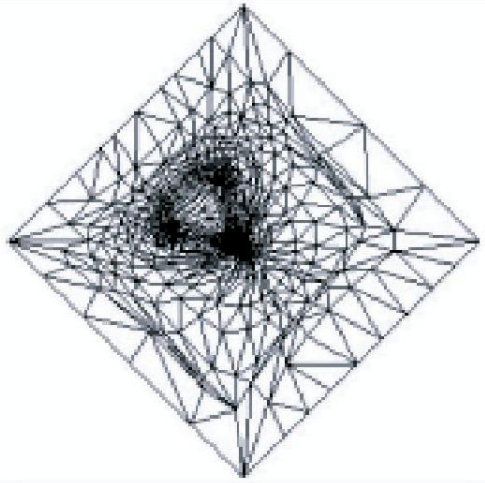
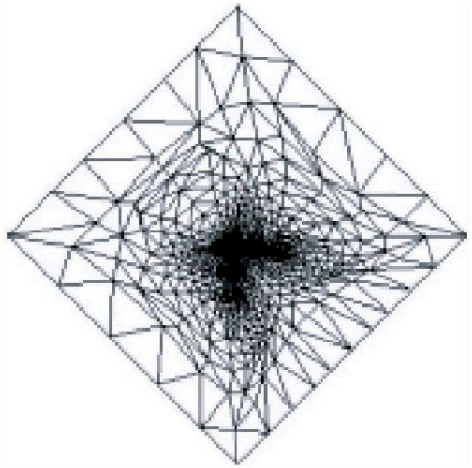
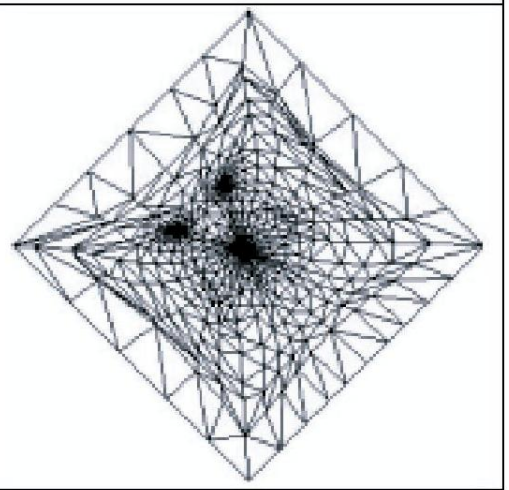
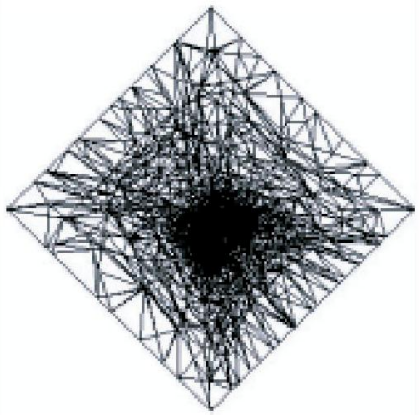
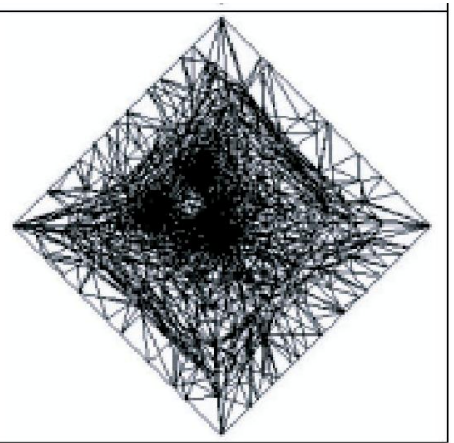


Merging

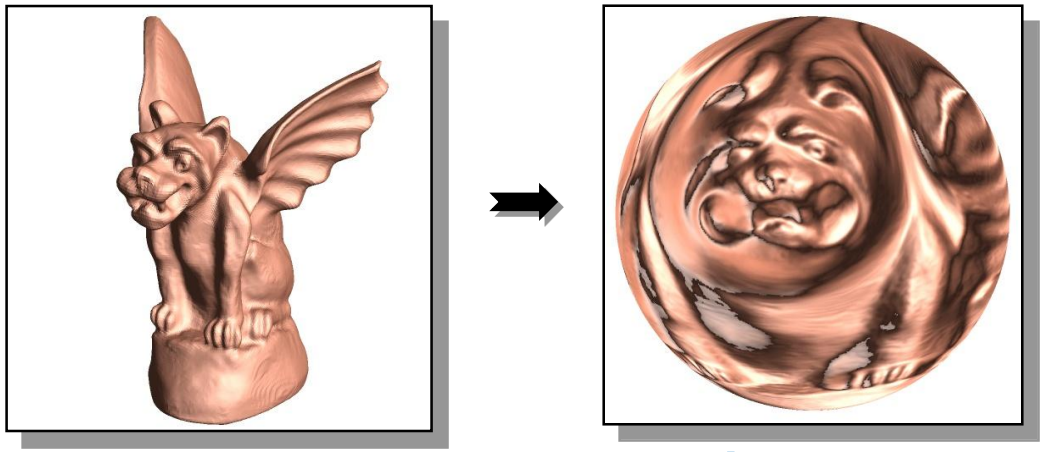




Cow ***pig***

	Head	Body
Cow		
Pig		
Merging		

从粗到细的球面参数化策略



为了将模型*M*参数化到球面上*S*，我们需要为每个**顶点**分配球面上的**坐标**。模型的**边**将对应于**大圆弧**，**网格三角形**将对应于**球面三角形**。这种映射有两个主要目标：

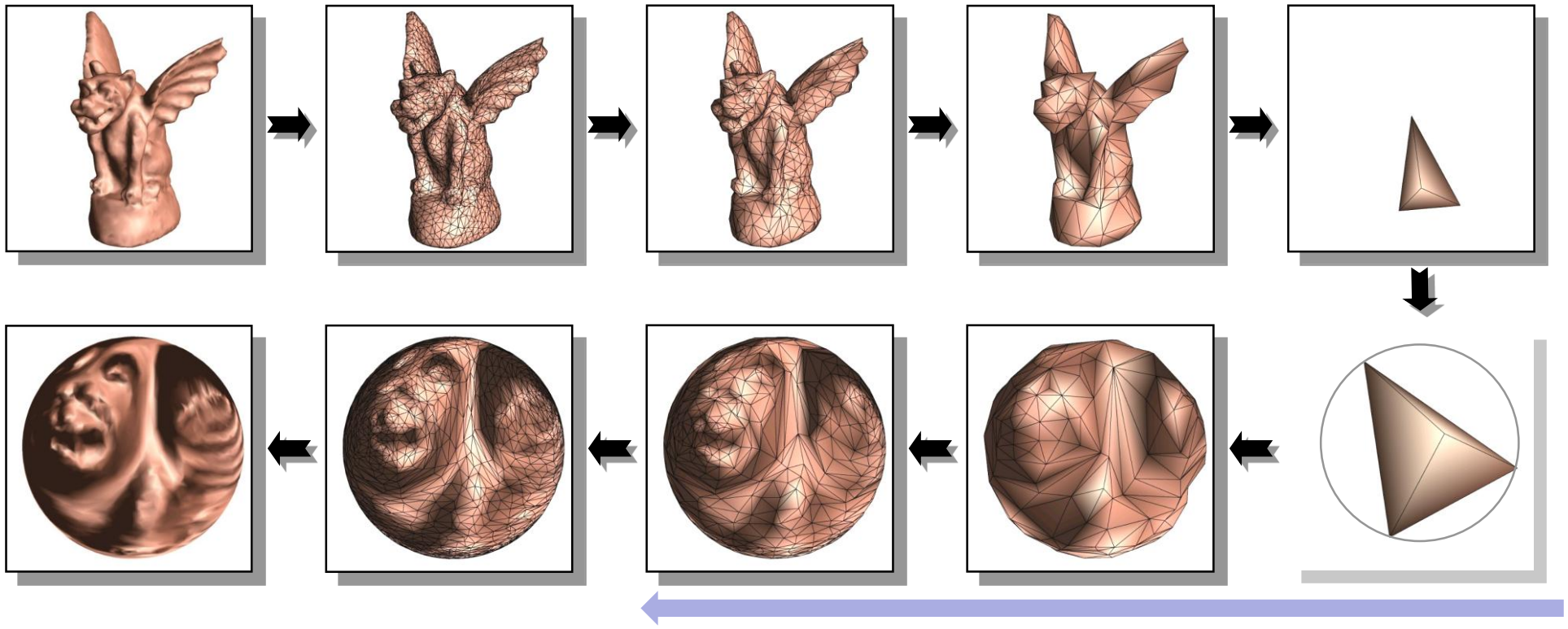
1）**鲁棒性**；**鲁棒性**意味着生成的映射保证是一一对应的，这意味着球体上没有三角形出现**重叠（折叠）**。

2）**良好的采样**。良好的采样率意味着模型的**感兴趣的特征**在球体上获得足够的空间，以便进行准确采样。我们通过使用从粗到细的策略来实现这些目标，并使用 Sander 等人[2001]的拉伸度量最小化映射失真。

* SANDER, P., SNYDER, J., GORTLER, S., AND HOPPE, H. 2001. Texture mapping progressive meshes. ACM SIGGRAPH 2001, pp. 409-416.

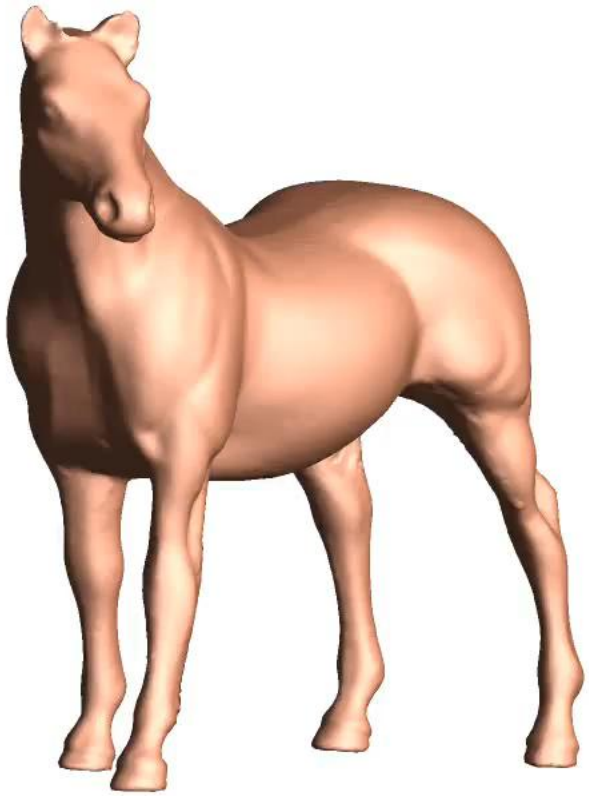
从粗到细的球面参数化策略

把输入模型转化为渐进网格(Progressive Mesh)

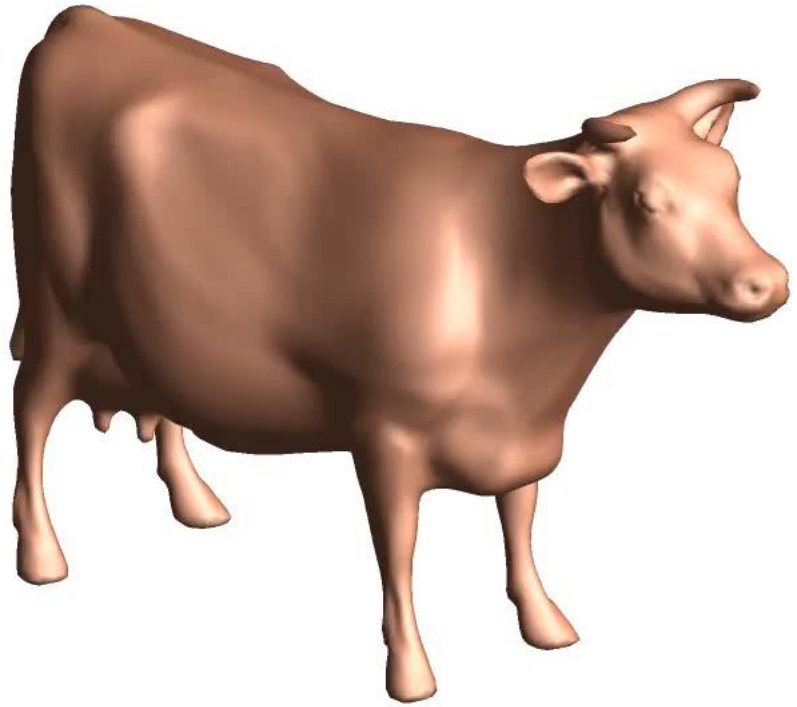


从粗到细参数化，保持嵌入并尽量减少拉伸

球面参数化及其Morphing应用



马模型的球面参数化



牛模型的球面参数化

球面参数化及其Morphing应用

滴水兽到兔子模型的Morphing动画

球面参数化及其Morphing应用

Mesh模型的平面参数化

- **RiemannMapper : A Mesh Parameterization Toolkit**
- https://www3.cs.stonybrook.edu/~gu/software/RiemannMapper/

Mesh模型的平面参数化

- Mesh模型的平面参数化在三维Morphing和纹理映射中都是一个非常有用的操作。

Mesh模型的平面参数化

- 我们讨论如何把一个R^3中的物体 M 如何映射为R^2中的单位园?

- 调和映射(Harmonic Map), h : $M \rightarrow H$ is one of the mapping that realizes embedding from a topological disk to a planar graph.

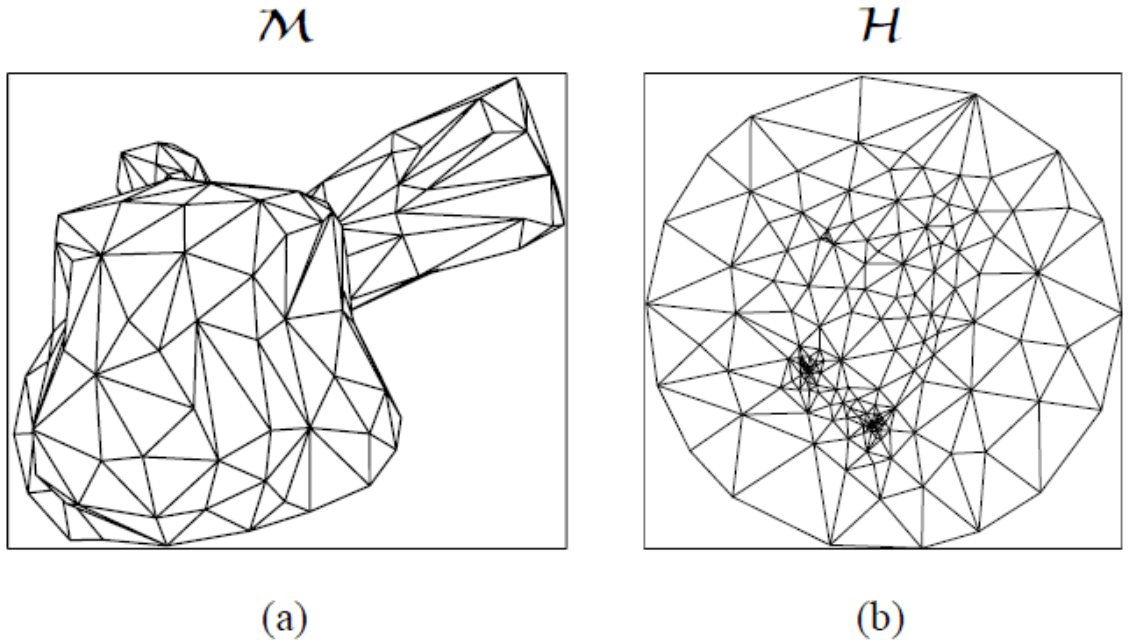


Figure 2. Harmonic Map: (a) a bunny model. (b) Harmonic Map.

Mesh模型的平面参数化

- 首先，我们把构成边界H的n个顶点放置到中心为原点的圆盘上（R^2空间），并确保边界点之间的**夹角**与连接这两个顶点的边长成正比。其余顶点的位置则根据使总能量 E_{harm} 最小化来计算：

$$E_{harm}(\mathbf{v}) = \frac{1}{2} \sum_{\{i,j\} \in Edges(\mathrm{H})} k_{i,j} \left\{ \left\| v_i - v_j \right\| \right\}^2$$

- 其中，i, j 表示顶点，v_i, v_j表示它们在H的位置，$\mathbf{v}$为顶点v_i的集合。$Edges(H)$表示H中边的集合，$\{i, j\}$表示连接顶点 i 和j的边。$k_{i,j}$表示边$\{i, j\}$的弹簧常数。

Mesh模型的平面参数化

- 对于每个面$\{i, j, k\}$, 假设$A_{i,j,k}$表示该面在M 的面积。对于关联到面$\{i, j, k_1\}$, $\{i, j, k_2\}$的每条内边$\{i, j\}$:

$$k_{i,j} = \frac{l_{i,k_1}{}^2 + l_{j,k_1}{}^2 - l_{i,j}{}^2}{A_{i,j,k_1}} + \frac{l_{i,k_2}{}^2 + l_{j,k_2}{}^2 - l_{i,j}{}^2}{A_{i,j,k_2}}$$

Mesh模型的平面参数化

- 通过求解一个稀疏的线性系统：

$$\nabla E_{harm} = 0$$

其中，∇E_{harm} 为E_{harm}对**v**的梯度。因为E_{harm}是一个关于H 中顶点位置的正二次函数，我们可以求得一个使E_{harm}极小化的唯一解。

- 为了求得E_{harm}的梯度， 我们把**v**的x, y分量按顺序排列，从而得到一个2N(N为顶点的数目)维的向量**V**：

$$\mathbf{V} \equiv \left(v_{1x}, v_{1y}, v_{2x}, v_{2y}, \cdots, v_{Nx}, v_{Ny}\right)$$

Mesh模型的平面参数化

- E_{harm} 是关于**V**中每个分量的二次型，它可以表示为 $\nabla E_{harm} = \mathbf{V}^T \mathbf{H} \mathbf{V}$

- 因此， E_{harm} 的梯度可以表示为：$\nabla E_{harm} = \partial E_{harm} / \partial \mathbf{V}$
- 而且，边界上的点是固定的。为了求解线性系统，我们把变量向量**V**分成两部分：自由部分$\mathbf{V}_{\alpha}$和固定部分$\mathbf{V}_{\beta}$ ， 常数矩阵**H**也可进行相应分解。因此能量函数E_{harm}可重写为：

$$\nabla E_{harm} = \begin{bmatrix} \mathbf{V}^{\alpha T} & \mathbf{V}^{\beta T} \end{bmatrix} \begin{bmatrix} \mathbf{H}^{\alpha\alpha} & \mathbf{H}^{\alpha\beta} \\ \mathbf{H}^{\beta\alpha} & \mathbf{H}^{\beta\beta} \end{bmatrix} \begin{bmatrix} \mathbf{V}^{\alpha} \\ \mathbf{V}^{\beta} \end{bmatrix}$$

Mesh模型的平面参数化

- 对于固定部分，这个能量函数是常量。因此，我们只需计算∇E_{harm}

  对自由部分$\mathbf{V}_{\alpha}$的偏导来极小化E_{harm}：

$$\nabla E_{harm} = \frac{\partial E_{harm}}{\partial \mathbf{V}^{\alpha}} = 2\mathbf{H}^{\alpha\alpha}\mathbf{V}^{\alpha} + 2\mathbf{H}^{\alpha\beta}\mathbf{V}^{\beta} = 0$$

解这个线性系统，我们可得到自由部分$\mathbf{V}_{\alpha}$的位置。

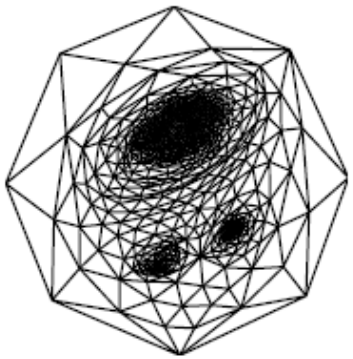
3D objects

$\mathcal{M}_1$

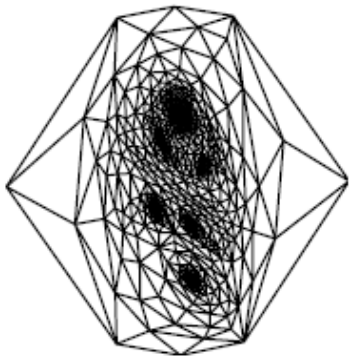
$\mathcal{M}_2$

Harmonic Map

Embeddings



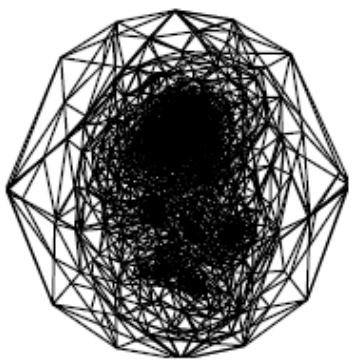
$\mathcal{H}_1$



$\mathcal{H}_2$

Merge

A merged embedding



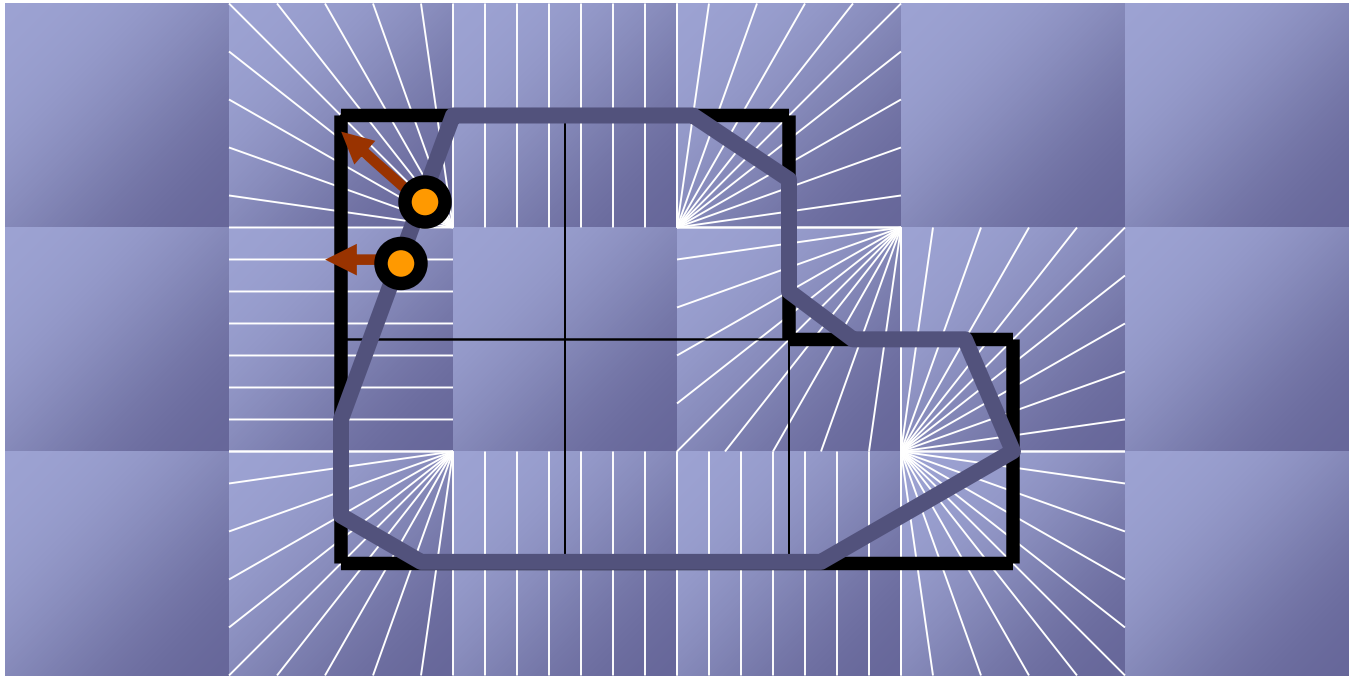
$\mathcal{H}_c$

Interpolation from $\mathcal{M}_1$ to $\mathcal{M}_2$

Mesh模型的Polycube参数化

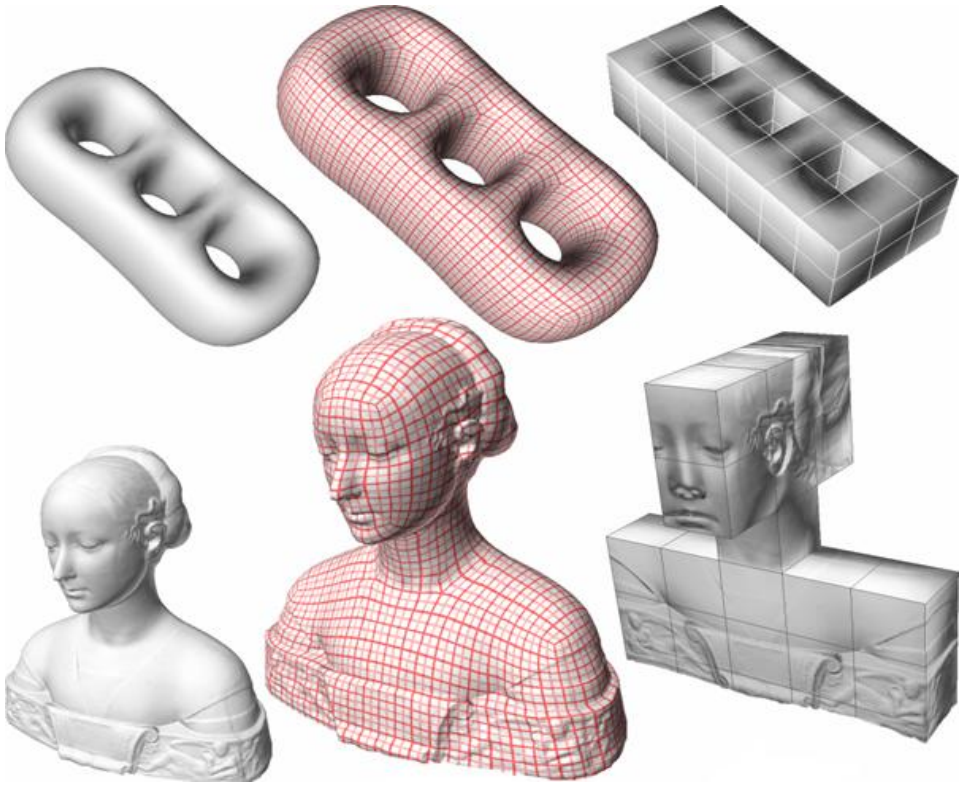
Polycube Maps

2D analogue

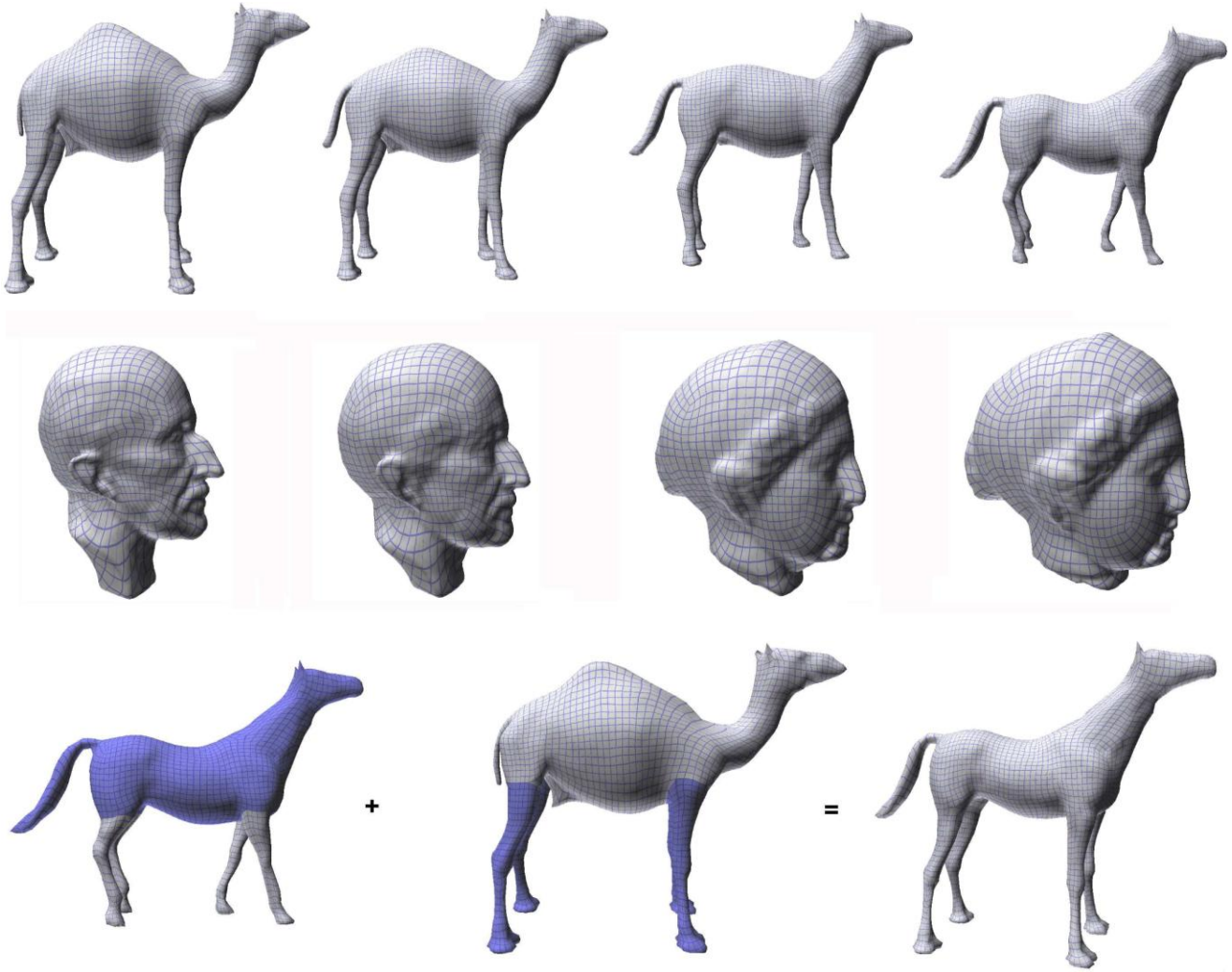


dual grid

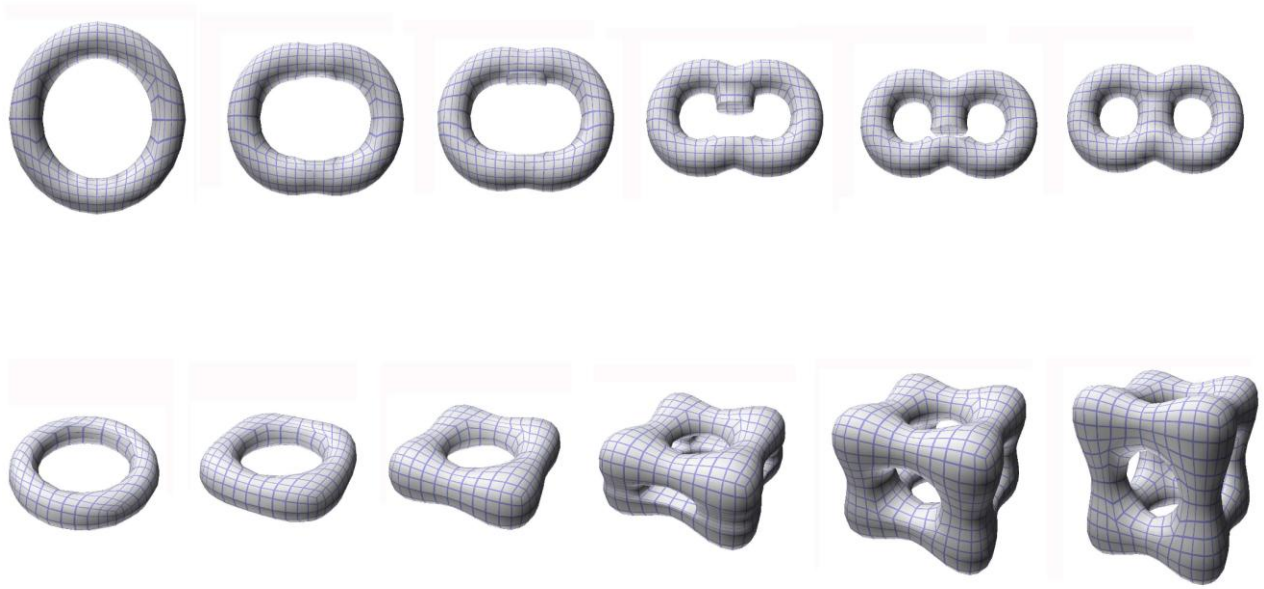
polycube

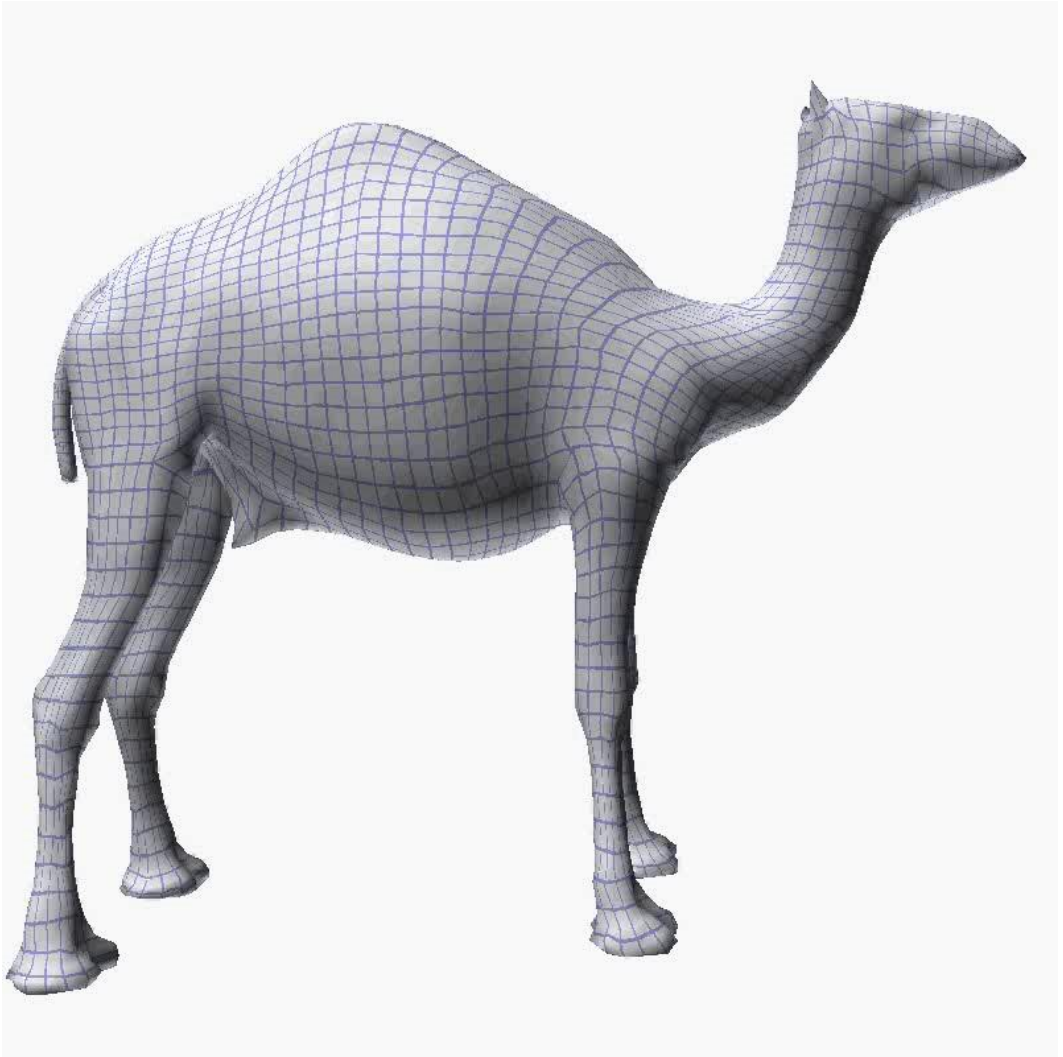


Mesh模型的Polycube参数化



Mesh模型的Polycube参数化与Morphing

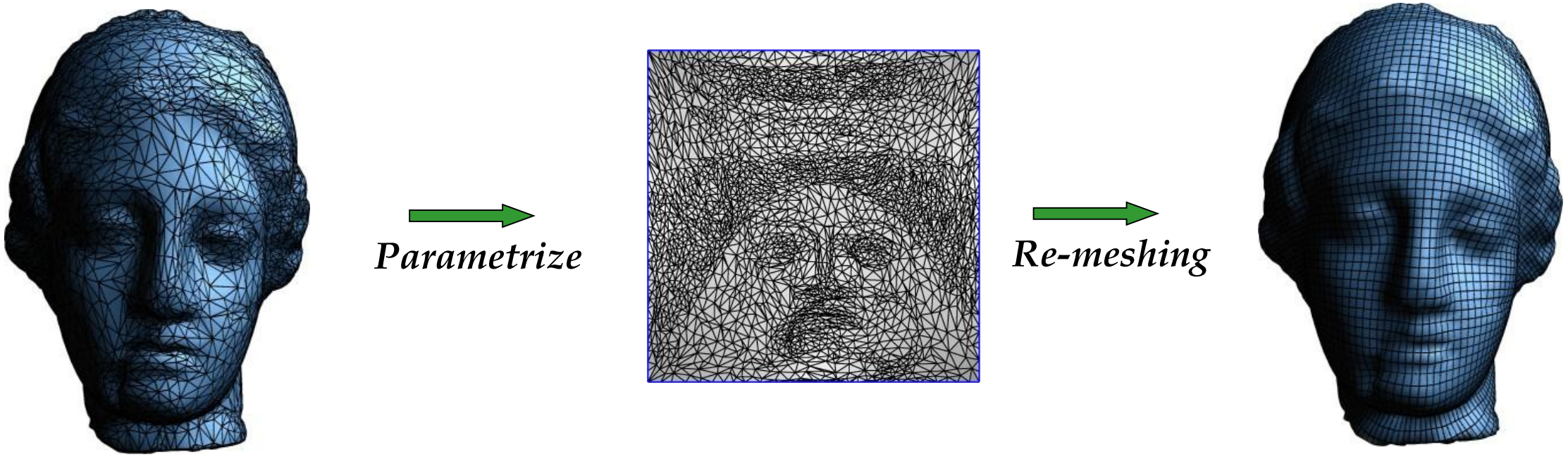




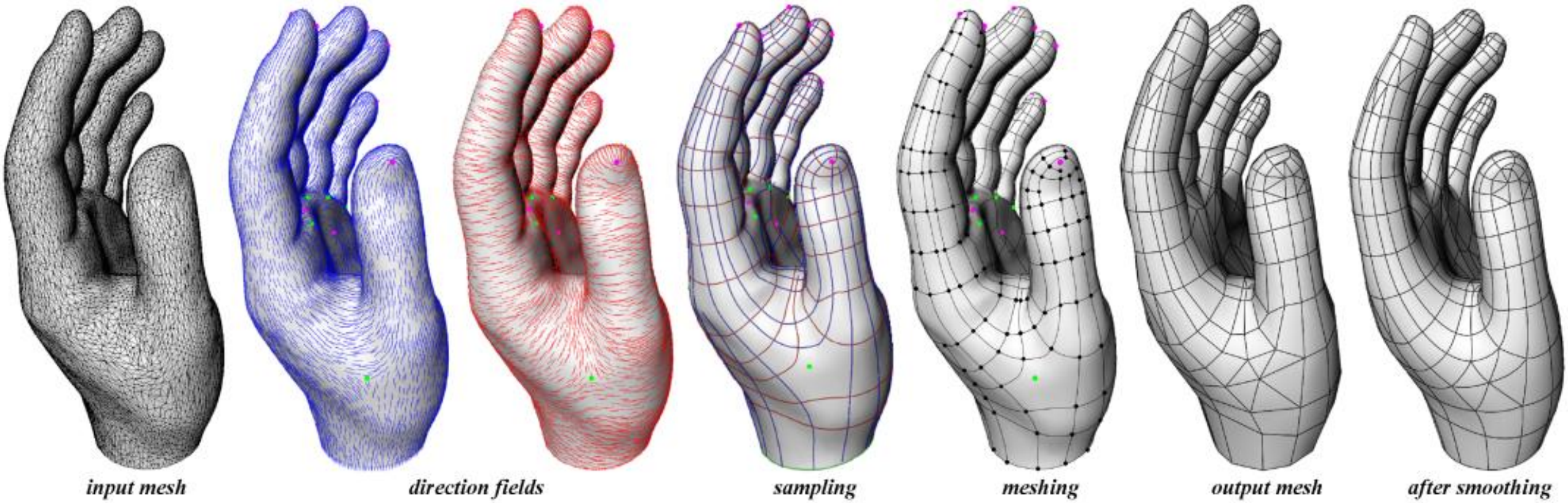
Zhengwen Fan, Xiaogang Jin, Jieqing Feng, Hanqiu Sun: Mesh morphing using polycube-based cross-parameterization. Comput. Animat. Virtual Worlds 16(3-4): 499-508 (2005)

Patch Parameterization and Re-meshing

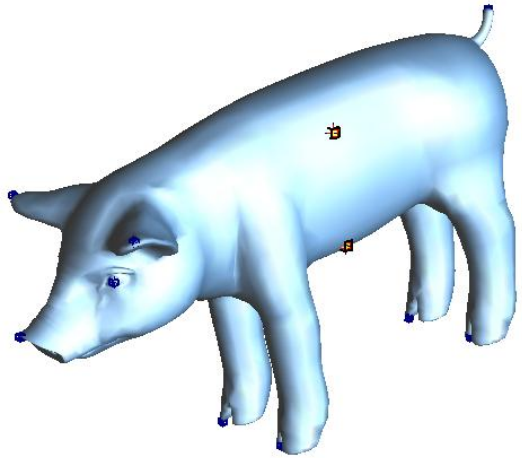
- Modify the sampling and connectivity of a geometry
- Convert an irregular mesh to a (semi-)regular mesh

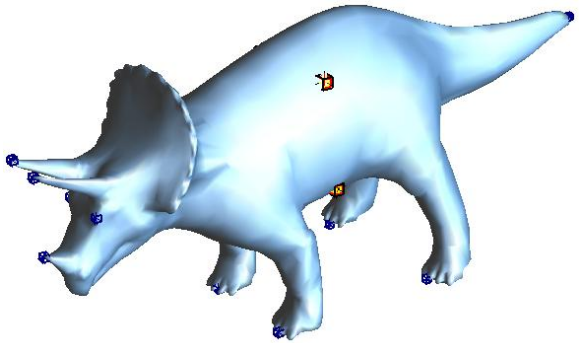


Patch Parameterization and Re-meshing



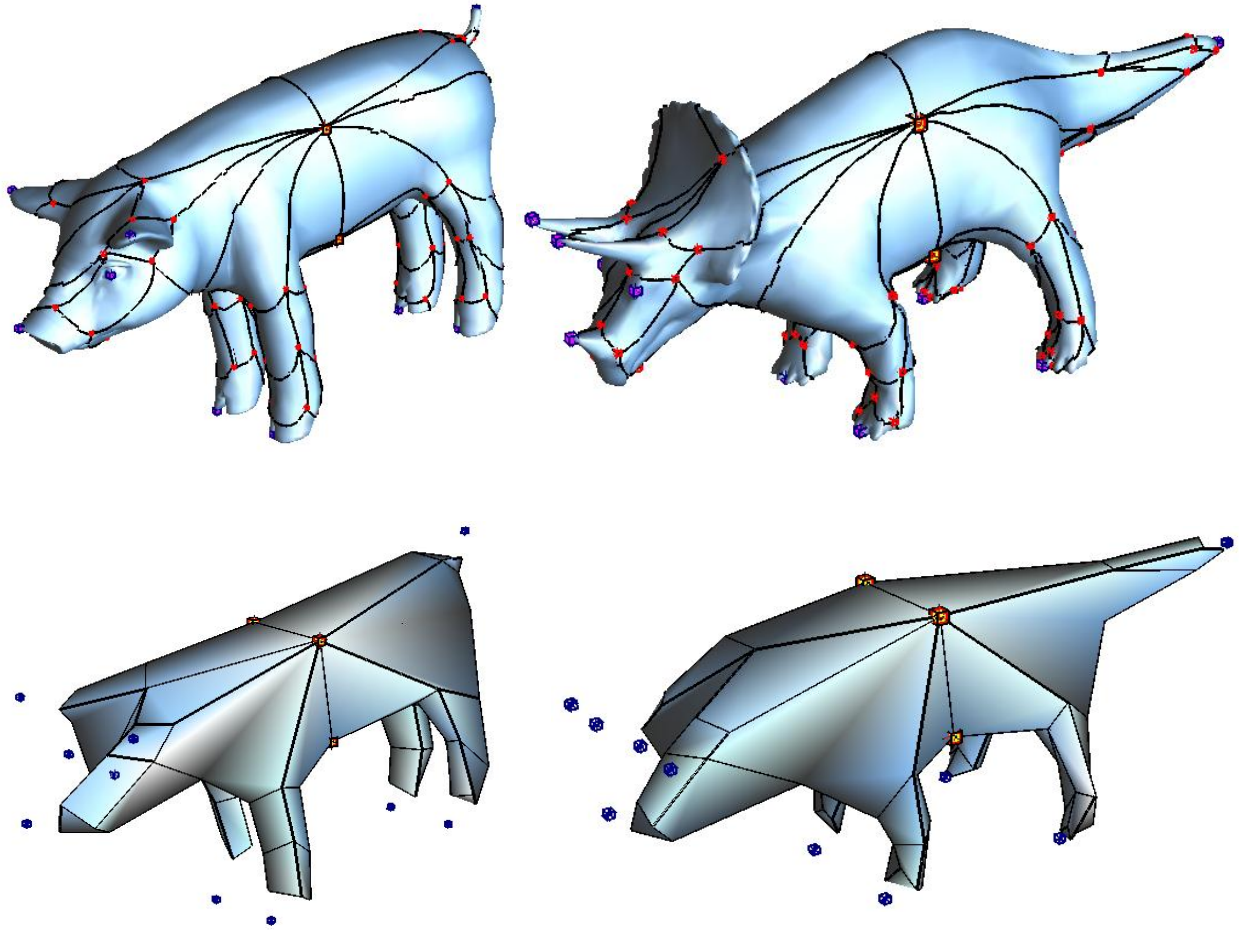
Patch Parameterization and Re-meshing





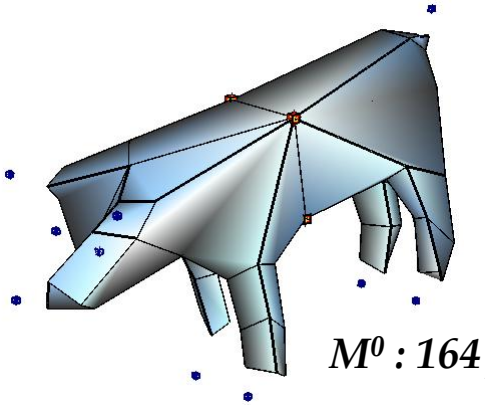
User-specified feature points

- 4 basic points
- Additional points

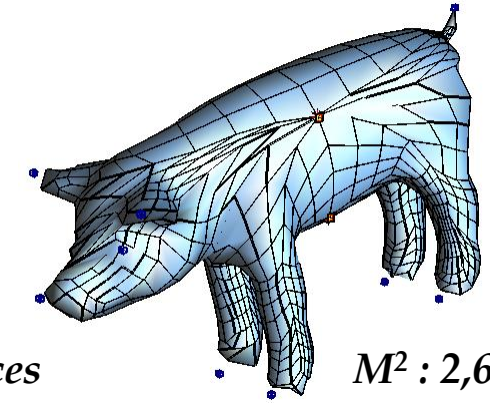


Consistent common dissection of pig and triceratops models

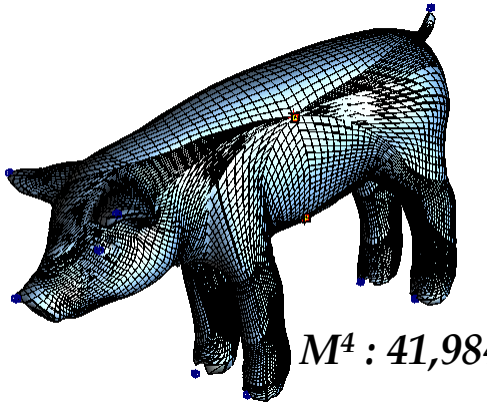
Patch Parameterization and Re-meshing



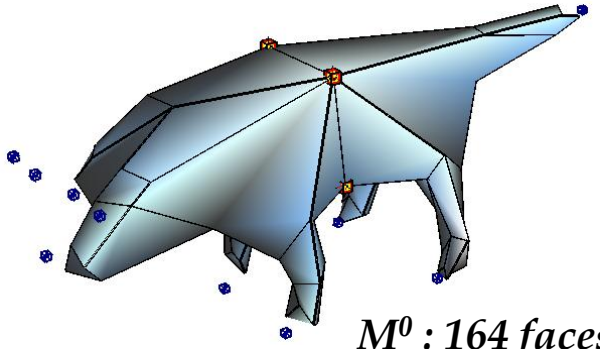
M^0 : 164 faces



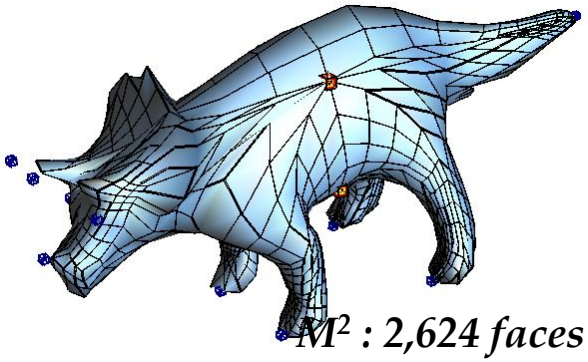
M^2 : 2,624 faces



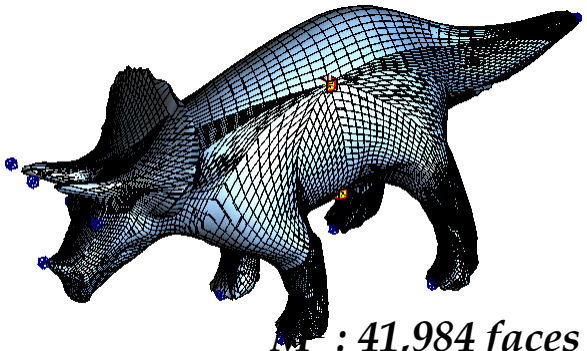
M^4 : 41,984 faces



M^0 : 164 faces

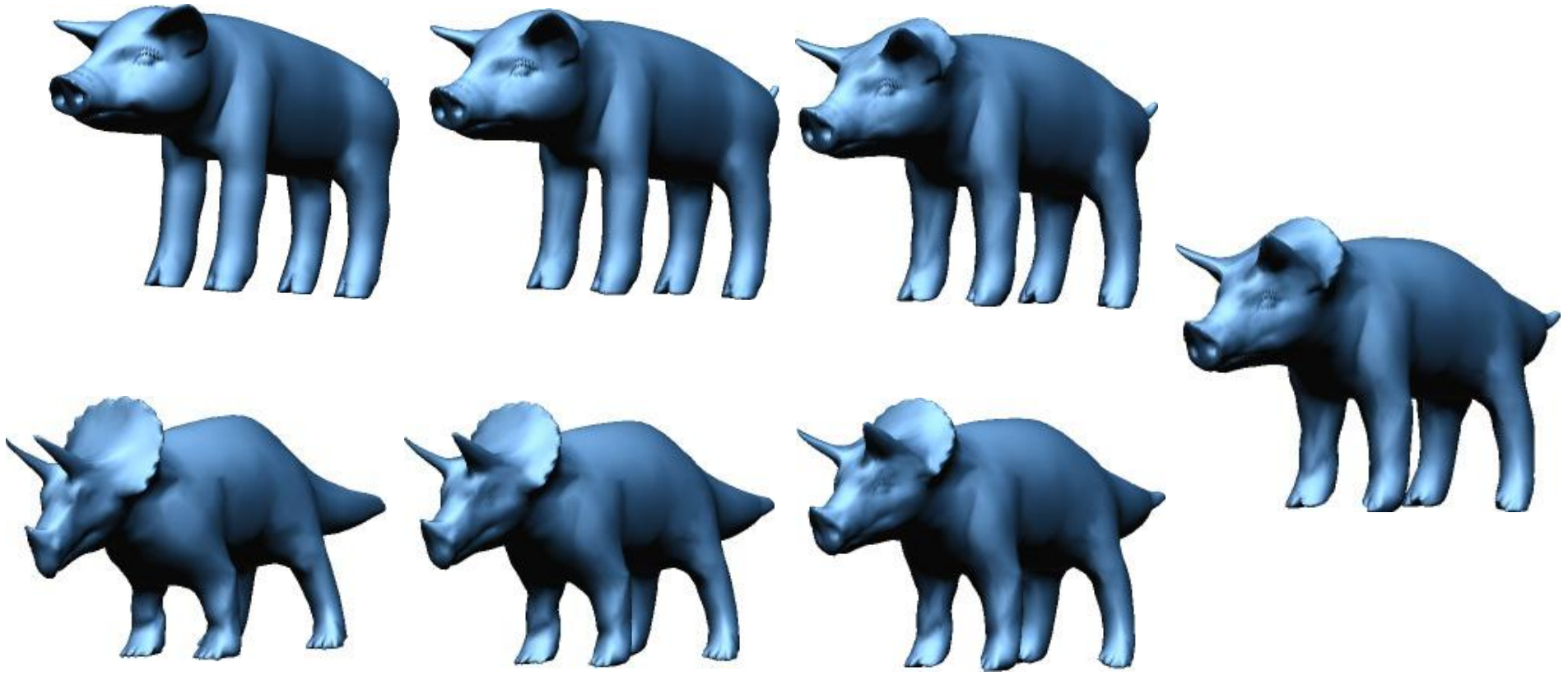


M^2 : 2,624 faces



M^4 : 41,984 faces

Patch Parameterization and Re-meshing



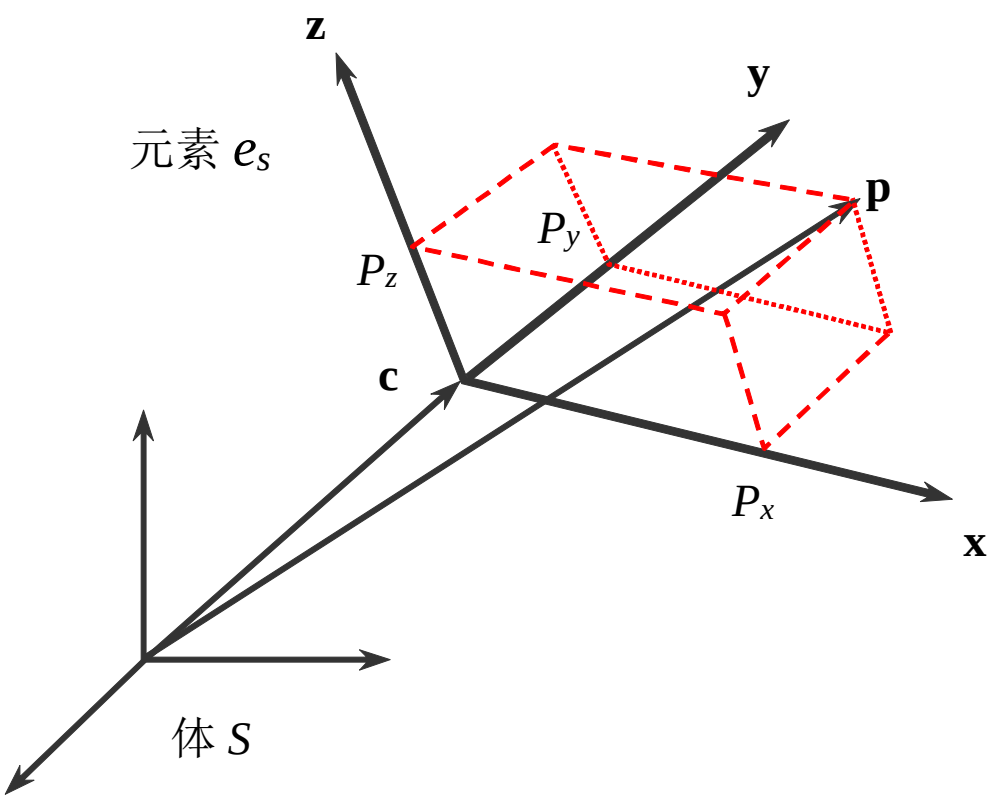
基于体表示的三维morphing方法

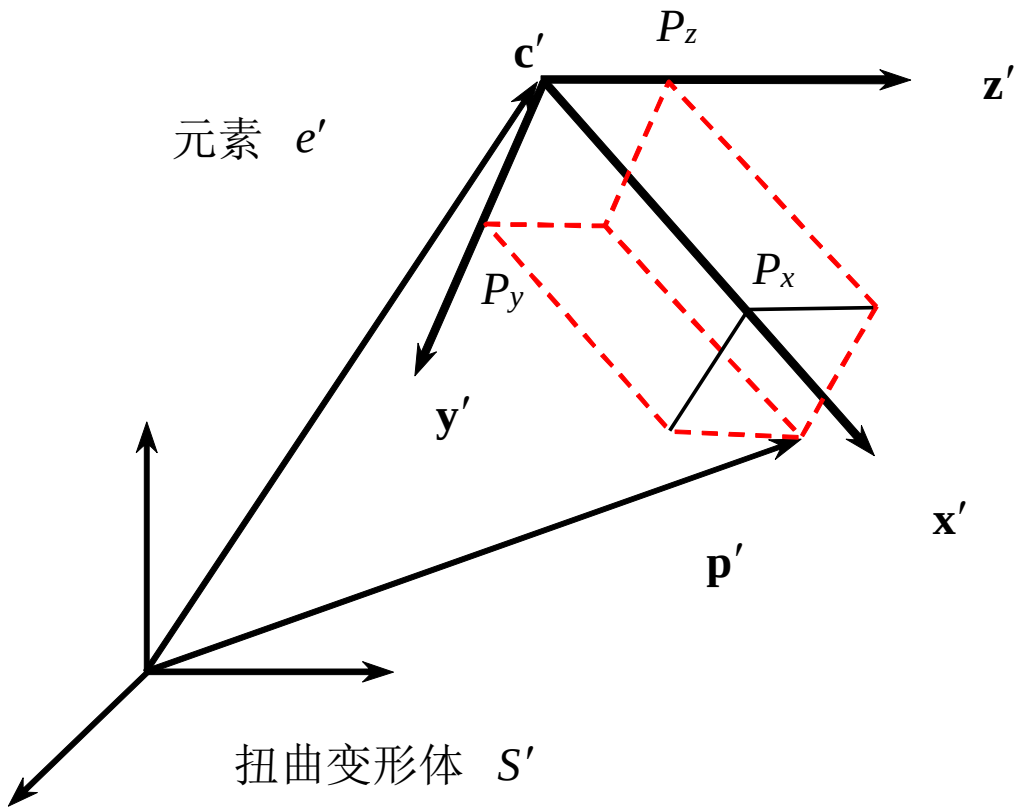
- 1995年，Lerios等人通过把Berier和Neely的二维图象morphing方法推广到三维，提出了基于体表示（volume-based）的三维morphing方法。
- 思想：给定源体S和目标体T，与二维图象morphing的思想类似，该方法首先根据指定的对应特征生成一空间变换，该变换使给定的两个体扭曲变形（warp）成S′和T′，达到几何对齐的目的；然后对得到的两个扭曲变形体S′和T′进行混合。
- 体数据可以用多种方法来获取，如扫描CT或核磁共振数据、对几何模型进行扫描转换、体数据造型方法、过程定义等。

* Lerios A, Garfinkle C D, Levoy M. Feature based volume metamorphosis. Computer Graphics, 1995, 29(3):449~456

几何对齐

- 在基于体表示的三维morphing中，首先要使给定的两个体扭曲变形，达到几何对齐。
- 为了实现几何对齐，动画师在源体和目标体中指定特征元素对(e_s, e_t) ,特征元素包括点、线、长方形和长方体。
- 为了便于在三维空间指定特征，Lerios把特征的位置和朝向编码为一个包含四个向量的特征局部坐标系，这四个向量包括一个位置向量c和三个定义坐标轴方向的正交单位向量x, y, z。
- 另外，特征元素还包括比例缩放因子属性s_x, s_y, s_z，它们定义了特征沿每个轴的伸展程度。
- 在morphing过程中，这些特征彼此变换到对方。这些变换把S中的特征平移、旋转、和比例缩放后，分别与T中对应特征的位置、朝向和大小相匹配。





一对特征元素

Warping函数的计算

- 插值。对对应特征元素e_s和e_t的局部坐标系和比例缩放因子插值，得到插值的局部坐标系x’, y’, z’和比例缩放因子s_x’, s_y’, s_z’，它们决定了特征元素e’。
- 逆向映射。对s’中的每一点P’，计算它在特征元素e’的局部坐标系中的坐标：

$$p_x = \frac{(\mathbf{p}' - \mathbf{c}') \bullet \mathbf{x}'}{s_x'}, \quad p_y = \frac{(\mathbf{p}' - \mathbf{c}') \bullet \mathbf{y}'}{s_y'}, \quad p_z = \frac{(\mathbf{p}' - \mathbf{c}') \bullet \mathbf{z}'}{s_z'}$$

- 由于P’在e’中的局部坐标与P在e_s中的局部坐标相同(**局部参数化**)，故P’在S中的对应点P为:

$$\mathbf{p} = \mathbf{c} + p_x s_x \mathbf{x} + p_y s_y \mathbf{y} + p_z s_z \mathbf{z}$$

这样，求得S扭曲变形后的体S’。

- 用e_t取代e_s，同理可求得T’中的每一点P’在T中对应点P，从而求得扭曲变形后的体T’。

多对特征元素的处理

- 有多对特征元素时，由于每对特征元素都决定了P’在S中的对应点$\mathbf{P}_i$，我们采用加权平均的方法来决定P’在S中的最终对应点P，

$$\mathbf{p} = \sum_{i=1}^{n} w_i \mathbf{p}_i \Big/ \sum_{i=1}^{n} w_i$$

- 其中权函数w_i依赖于P’到插值特征元素$\mathbf{e}_i$’的距离d_i，它与距离的平方成反比，

$$w_i = \frac{1}{(d_i + \varepsilon)^2}$$

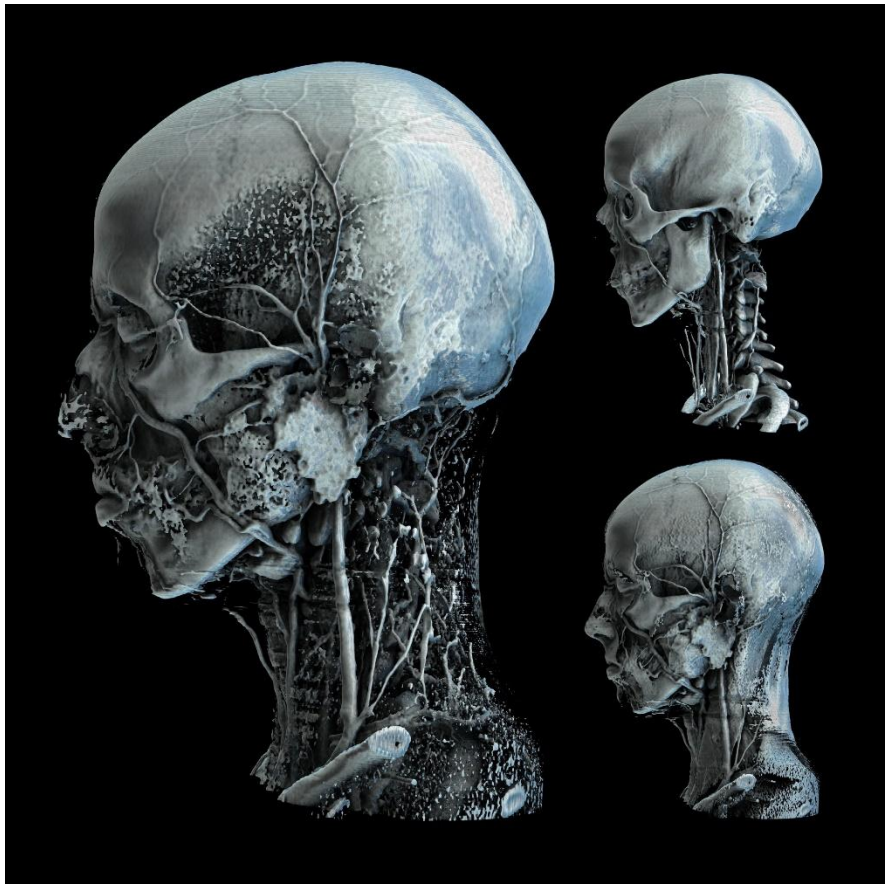
- 上式中的ε是一为了防止分母为零而设的很小的数，可设置为ε=0.001。

距离d_i的计算方法

- 特征元素的$\mathbf{e}_i$'类型不同，距离的计算方法也不同
  - 当$\mathbf{e}_i$'为一个点时，d_i为**P**'与点$\mathbf{e}_i$'之间的距离;
  - 当$\mathbf{e}_i$'为一条线段时， d_i为**P**'到$\mathbf{e}_i$'的最短距离，这与Beier基于线对的图象morphing中采用的方法一样。
  - 当$\mathbf{e}_i$'为长方形时，若**P**'位于长方形内，则距离为零；否则，距离为**P**'到该长方形边的最短距离。
  - 当$\mathbf{e}_i$'为长方体时，若**P**'位于长方体内，则距离为零；否则，距离为**P**'到该长方体的所有表面的最短距离。

混合(Blending)方法

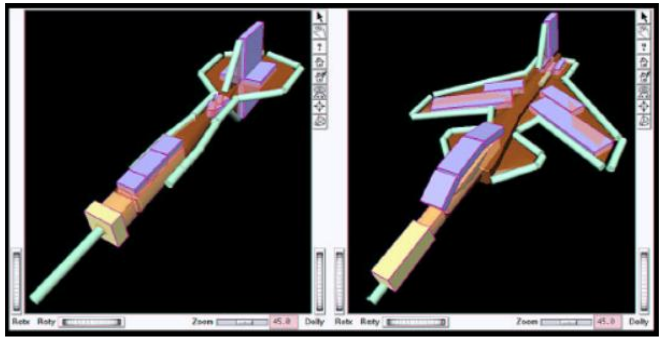
- 当Warping操作生成两个扭曲变形的体和后，我们可以采用多种方法对它们进行混合：
  - 先对S’和T’进行体绘制，然后把绘制好的图象进行交溶处理。该方法的缺点是生成结果的光照和遮挡情况不正确，且缺乏真实的三维morphing效果。
  - 对S’和T’体素的颜色和不透明度值进行交溶处理，然后对混合后的体素进行体绘制(volume rendering)。通常，后一种方法可以得到更好的三维morphing效果。

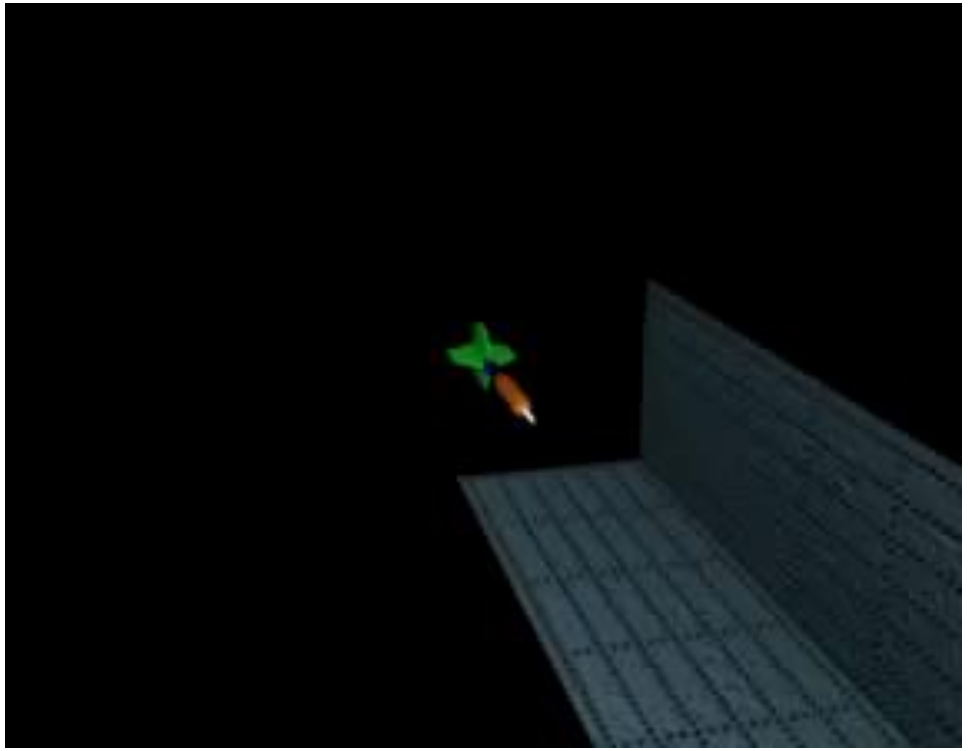


Volume rendering

实验例子1

用户交互和特征指定

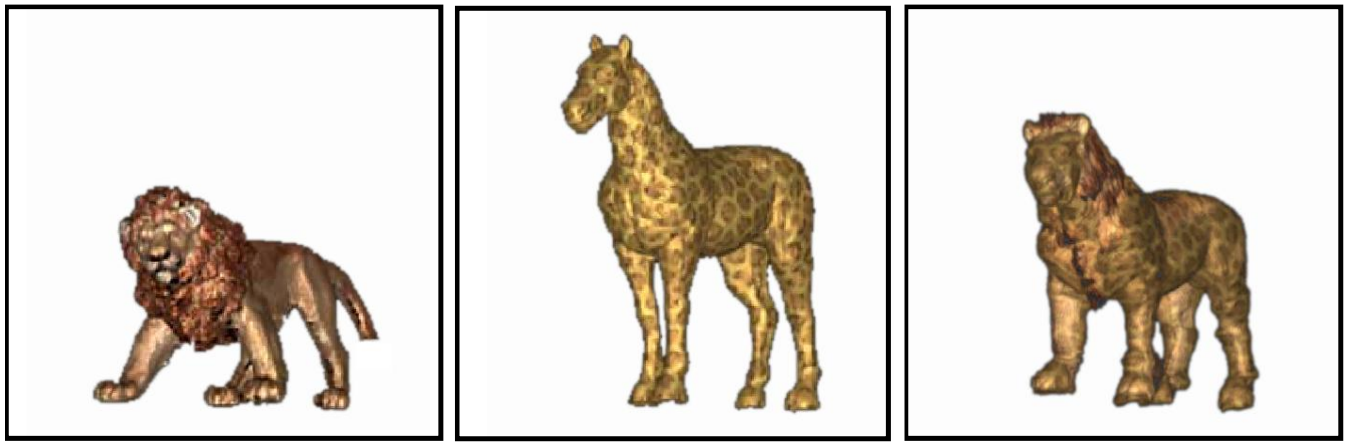




(a) 扫描转换飞镖得到的体　(b) 扫描转换X-29飞机得到的体　(c) morphing的中间帧

图　飞镖到X-29飞机的三维体morphing

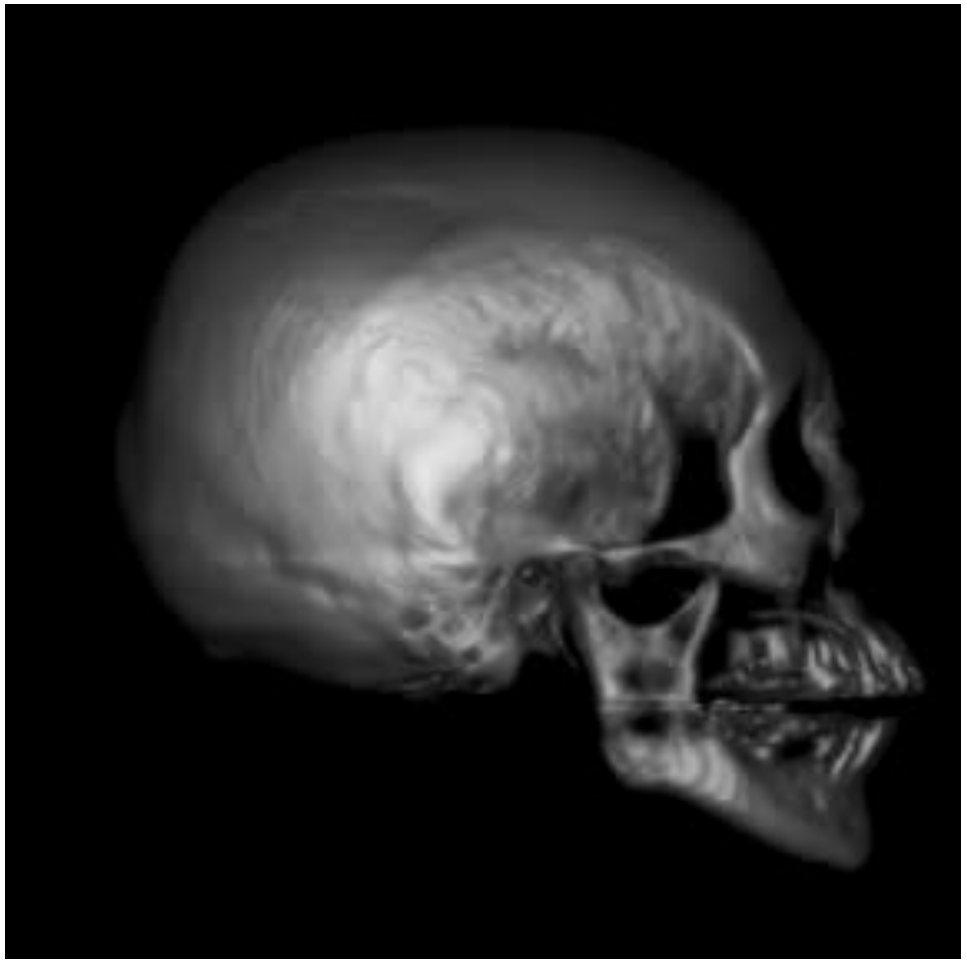
实验例子2



(a) 扫描转换狮子模型得到的体 (b) 扫描转换豹马模型得到的体 (c) morphing的中间帧

图 狮子到豹马的三维体morphing

实验例子3



Volume方法的优缺点

优点

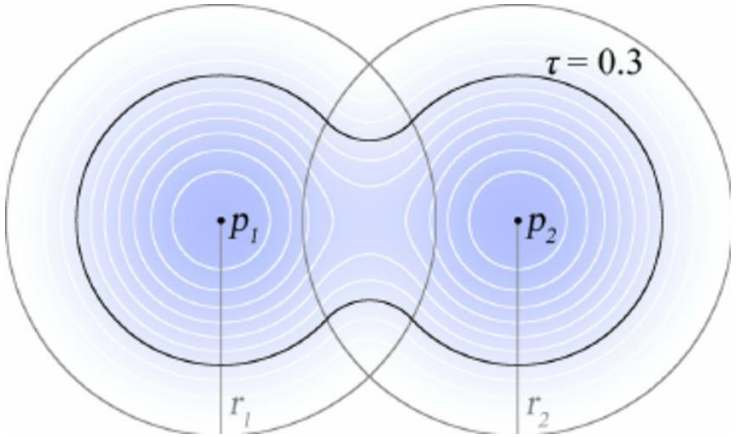
- 基于体的三维morphing方法与物体的几何和拓扑结构无关；
- 由于多面体、NURBS等几何表示的物体可以转化为体表示，基于体表示的morphing提供了一种统一处理方法。

缺点

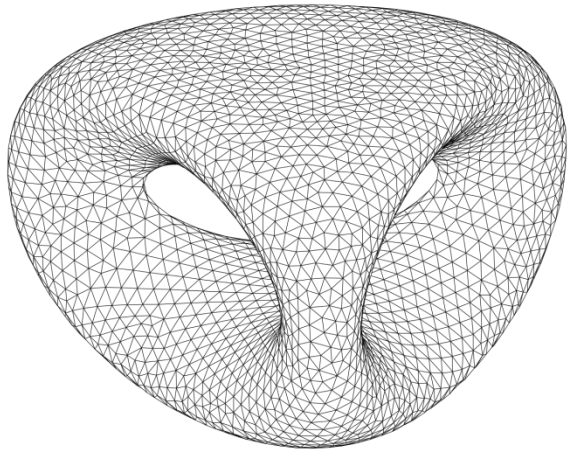
- 走样现象较为严重，结果的精度通常没有基于几何表示的三维morphing方法好；
- 另外，为了获得较高精度的体素表示，几何模型需要扫描转换为300×300×300或者更高的体素表示，计算时间较费。

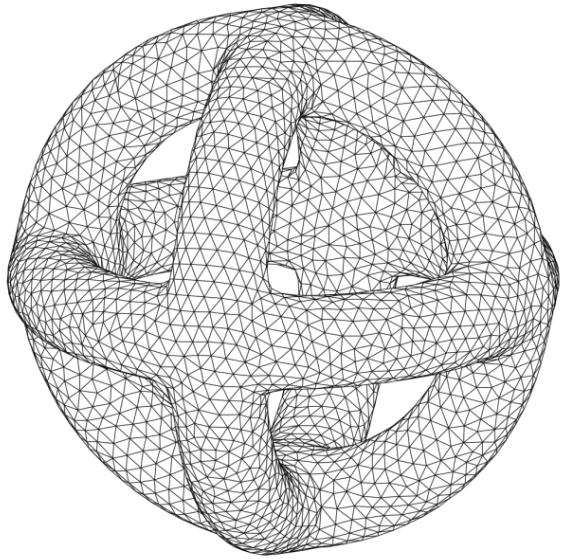
基于变分隐式曲面的三维Morphing

- Morphing可以采用隐式曲面的方法：



$$f(x, y, z) = T$$





1). 建立源物体和目标物体所对应的隐式曲面:

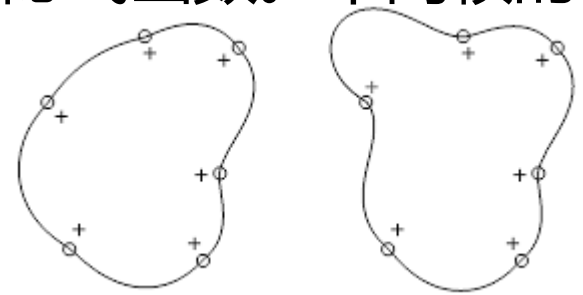
$f(x,y,z)=0;$ $g(x,y,z,)=0;$

2). 插值这两个函数:

$(1-t)*\ f(x,y,z)+t*g(x,y,z,)=0;\ 0<=t<=1;$

基于变分隐式曲面的三维Morphing思路

- Turk和O'brien提出了一种把这两步结合成一步的方法。他们把两个N维物体的Morphing转化为在$N+1$维空间的散乱数据插值问题。
- **2D情形**：对于二维形状，他们把所有的数据约束放在两个平面上，一个平面对应一个形状。然后把这两个平面平行地放在三维空间中。在形状的边界上指定0值约束，而在指向形状中心的边界法向方向指定正值约束。然后采用变分插值技术（薄板插值的三维推广），得到一个三维空间的隐式函数。中间帧的形状为这个三维函数的二维切片（零值轮廓）。
- **3D情形**：三维Morphing则可通过4D插值问题来进行。



优点：即使对于不同拓扑的三维物体，改方法仍能生成平滑自然的中间帧结果。

* Ref: Turk, Greg, and James F. O'brien. "Shape transformation using variational implicit functions." SIGGRAPH 1999: 335-342.

变分插值(Variational Interpolation)

- 采用散乱数据插值(scattered data interpolation)来解决形状的变换问题。
- 散乱数据插值：给定一系列数据点，构建一个光滑函数，通过给定的这些点。
- 2D散乱数据插值问题：给定k个约束点$\{\mathbf{c}_1,\mathbf{c}_2,\ldots,\mathbf{c}_k\}$，它们为平面上的散乱点，并给定在这些点的标量值$\{h_1, h_2,\ldots, h_k\}$，构建一个光滑曲面函数$f(\mathbf{x})$，使得曲面在给定约束点的高度值为给定的值，即$f(\mathbf{c}_i)= h_i$, i=1,2,3,…,k。
- 上述问题可通过变分法进行求解。通过构建度量插值函数质量的能量函数，在插值给定点的同时，使得能量函数极小。在二维情形，当采用如下能量函数E时: $E = \int_\Omega f_{xx}^2(\mathbf{x}) + 2f_{xy}^2(\mathbf{x}) + f_{yy}^2(\mathbf{x})$, 薄板插值即为其变分解。

基于变分隐式曲面的三维Morphing思路

- 能量函数E的极小值可通过如下径向基函数(**R**adial **B**asis **F**unction, **RBF**):

$$\emptyset(\mathbf{x}) = |\mathbf{x}|^2 \log(|\mathbf{x}|)$$

的加权和来求解，插值函数解$f(\mathbf{x})$为：$f(\mathbf{x}) = \sum_{j=1}^{n} d_j \emptyset(\mathbf{x} - \mathbf{c}_j) + P(\mathbf{x})$;

其中，$\mathbf{c}_j$为约束点的位置，$P(\mathbf{x})$为函数$f(\mathbf{x})$的线性和常量部分。

- 为了求解d_j，利用约束条件$f(\mathbf{c}_i) = h_i$,我们得到:

$$h_i = \sum_{j=1}^{n} d_j \emptyset(\mathbf{c}_i - \mathbf{c}_j) + P(\mathbf{c}_i);$$

这是一个关于d_j和$P(\mathbf{x})$ 系数的**线性方程组**。对于三维情形，假设

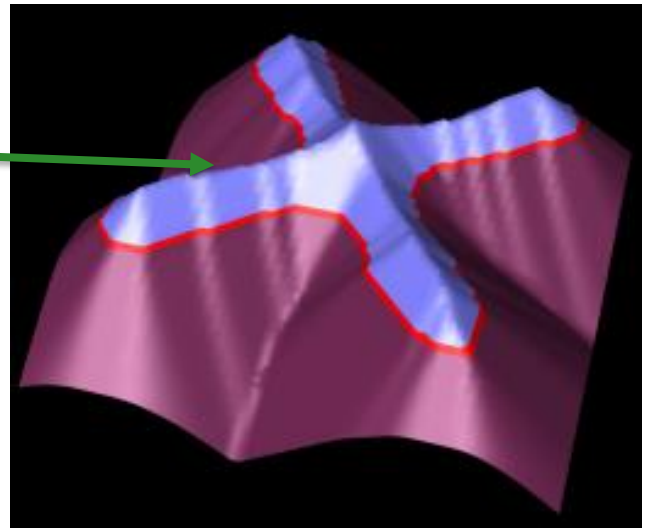
$\mathbf{c}_i = (\mathbf{c}_i{}^x, \mathbf{c}_i{}^y, \mathbf{c}_i{}^z)$, $\emptyset_{ij} = \emptyset(\mathbf{c}_i - \mathbf{c}_j)$, 则上述线性方程组可写成:

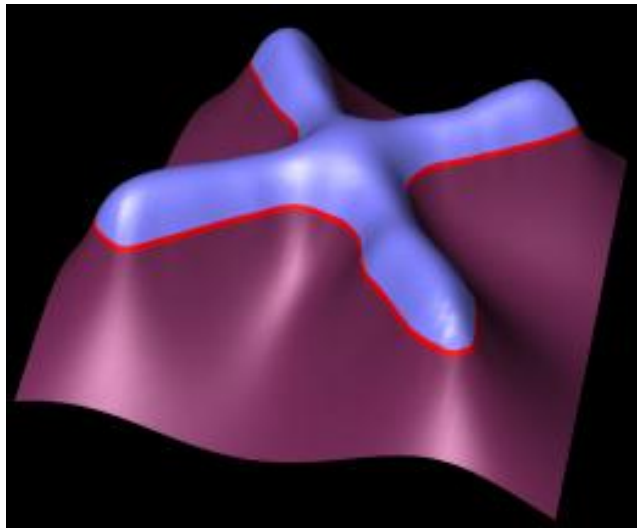
$$\begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1k} & 1 & c_1^x & c_1^y & c_1^z \\ \phi_{21} & \phi_{22} & \cdots & \phi_{2k} & 1 & c_2^x & c_2^y & c_2^z \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi_{k1} & \phi_{k2} & \cdots & \phi_{kk} & 1 & c_k^x & c_k^y & c_k^z \\ 1 & 1 & \cdots & 1 & 0 & 0 & 0 & 0 \\ c_1^x & c_2^x & \cdots & c_k^x & 0 & 0 & 0 & 0 \\ c_1^y & c_2^y & \cdots & c_k^y & 0 & 0 & 0 & 0 \\ c_1^z & c_2^z & \cdots & c_k^z & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_k \\ p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_k \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

该方程为对称半正定的(symmetric and positive semi-definite)，所以存在关于d_j和p_j的唯一解。对于小于数千个约束的系统，可用对称LU分解法求解 (symmetric LU decomposition)

2D形状的隐式曲面

sharp ridges



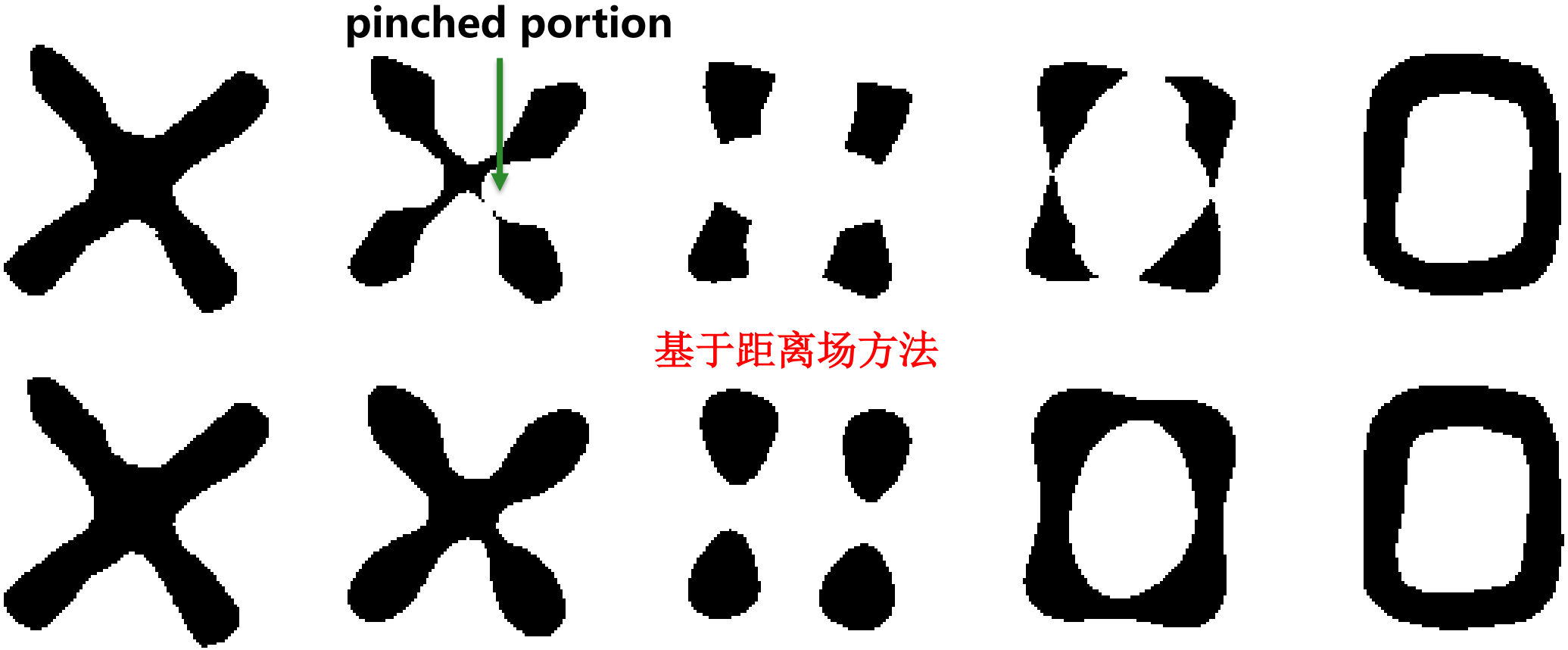


基于距离场方法　　基于变分隐式曲面

X-shape的隐式曲面

- 会生成sharp ridges (the medial axis)
- 使得中间帧形状呈现不光滑、挤压效果 (pinched portion)
- 可以生成光滑的结果

2D形状的隐式曲面



基于距离场方法

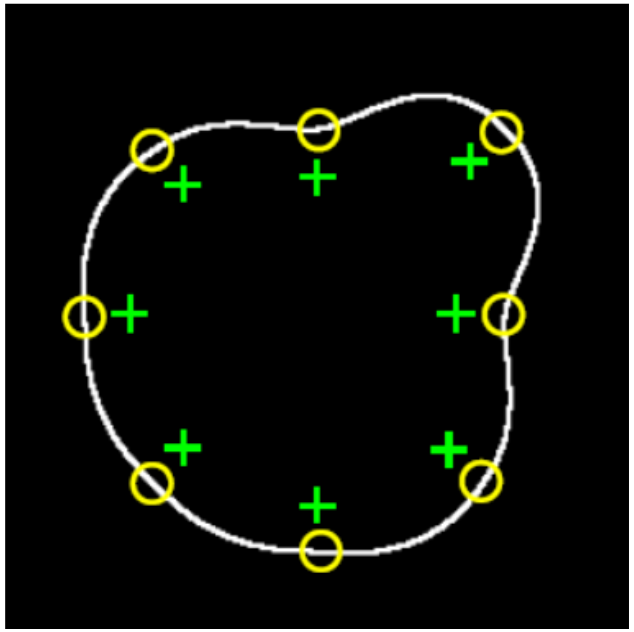
基于变分隐式曲面方法

X-shape到O-shape的形状变换

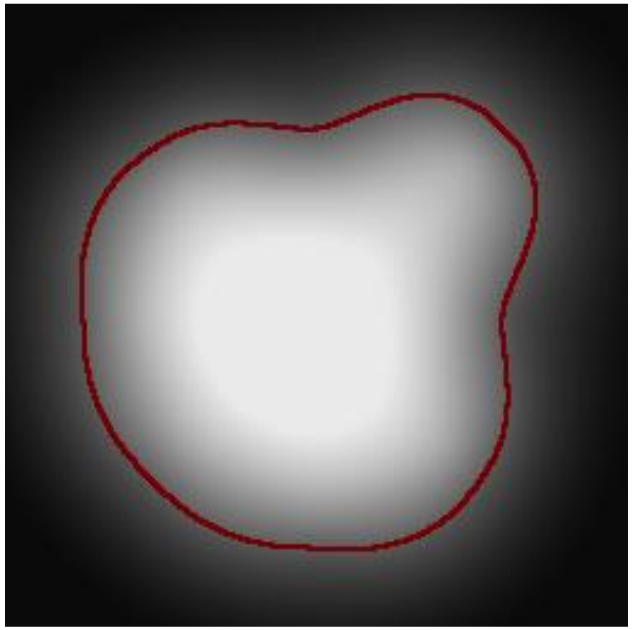
2D变分隐式曲面的构建

“O”约束：值为0，表示在曲面上，在2D Shape的边界上；

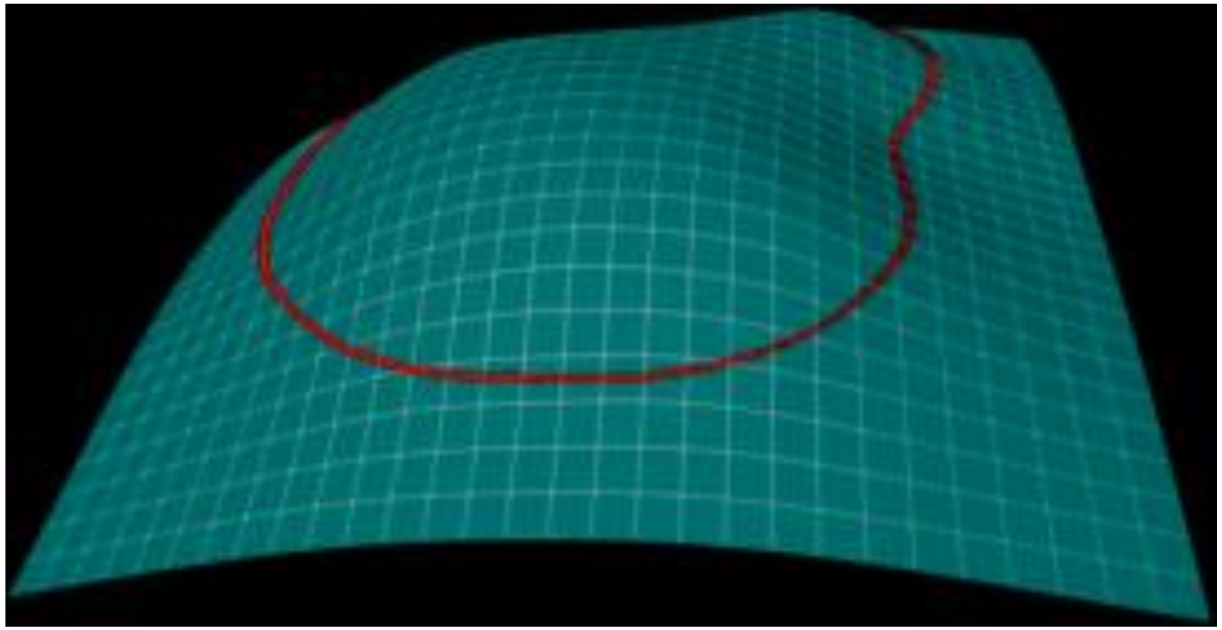
“+”约束：值为1，表示在曲面内部，沿边界点的法向向内偏移一定距离的点；



成对的边界(O)和法向(+)约束



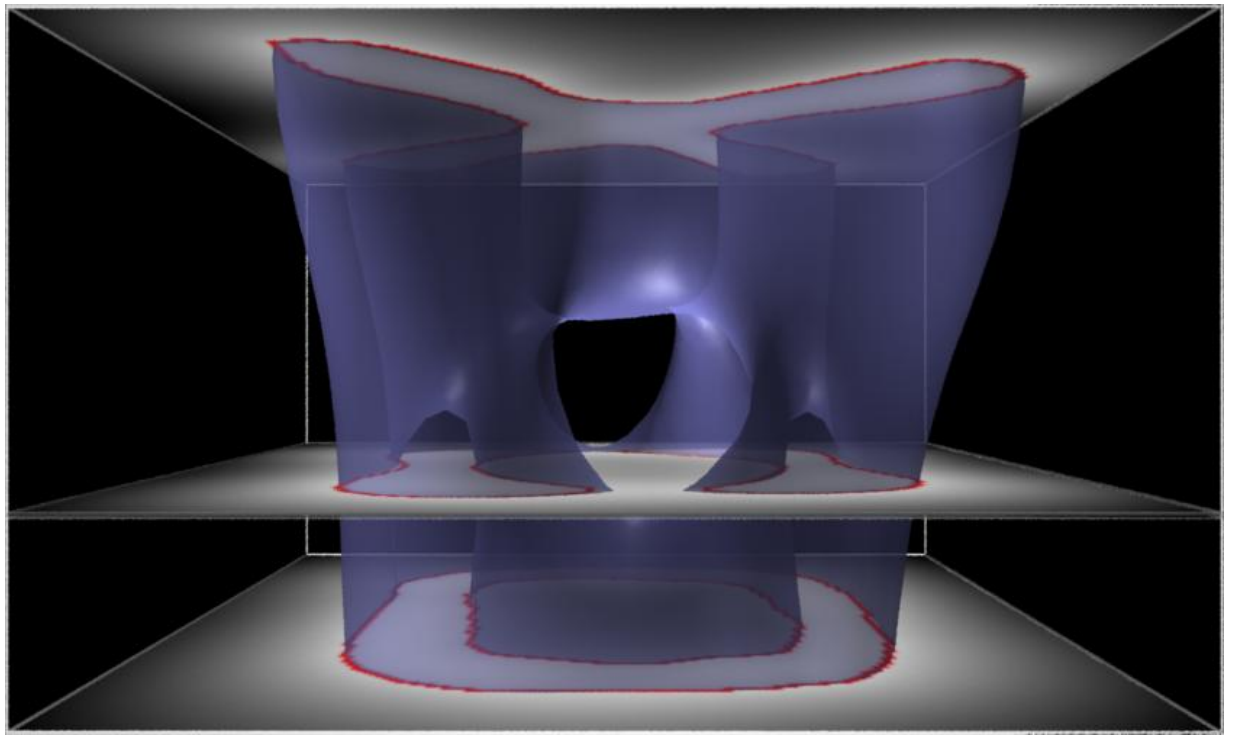
变分曲面的亮度表示



变分曲面的高度场表示

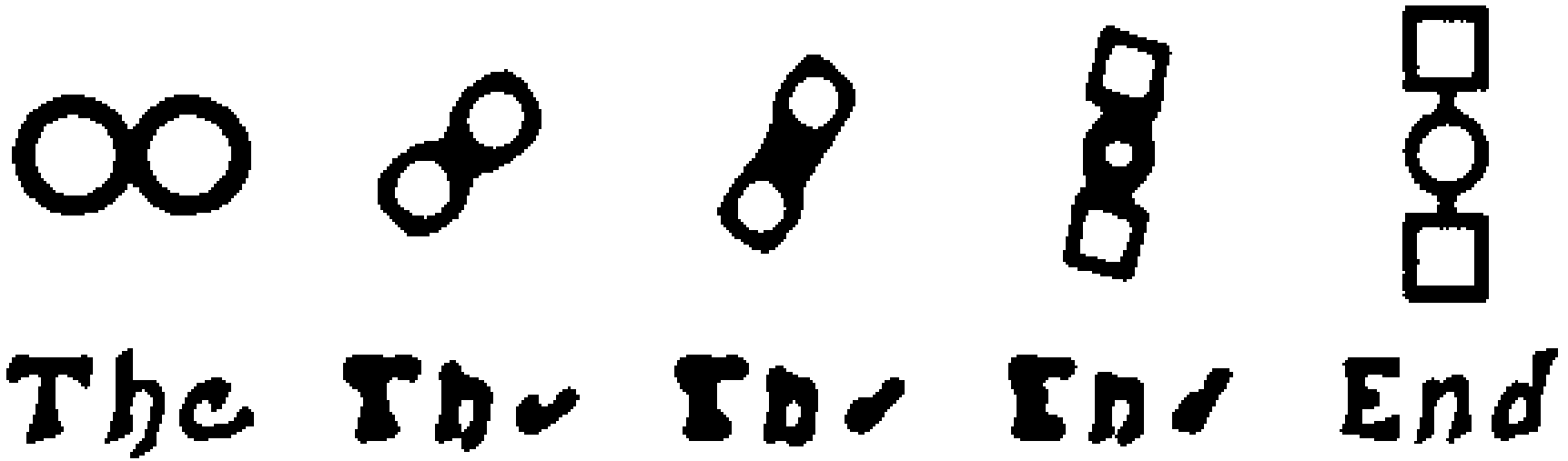
函数构建和插值的统一
(Unifying Function Creation and Interpolation)

- 采用该方法的关键是把整个形状序列**用一个隐式函数来表示**。
- 为此，我们在给定形状**更高维的空间（原有维度+1）**考虑问题：
  - 对于二维形状，在三维空间构建隐式函数，该函数在两个平行面上表示了给定的两个形状。



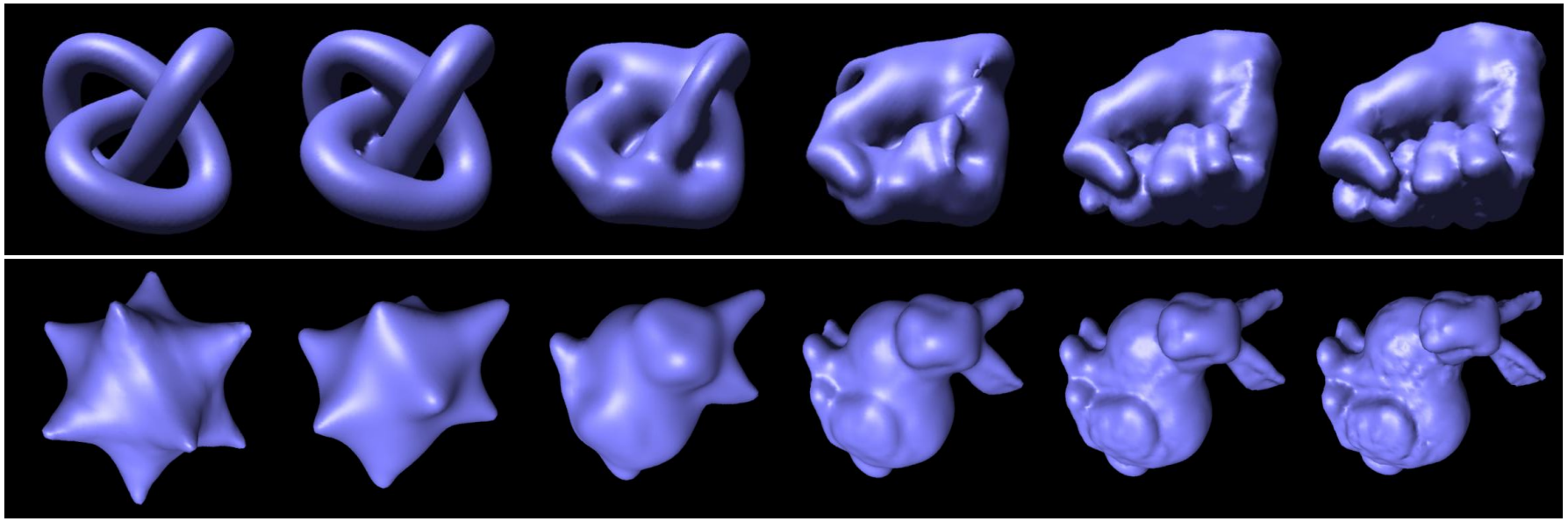
“X”和“O” 形状的变换。上平面和下平面包含了两个形状的约束。半透明曲面为三维变分隐式曲面的等值面。截面为中间形状。

函数构建和插值的统一——二维情形



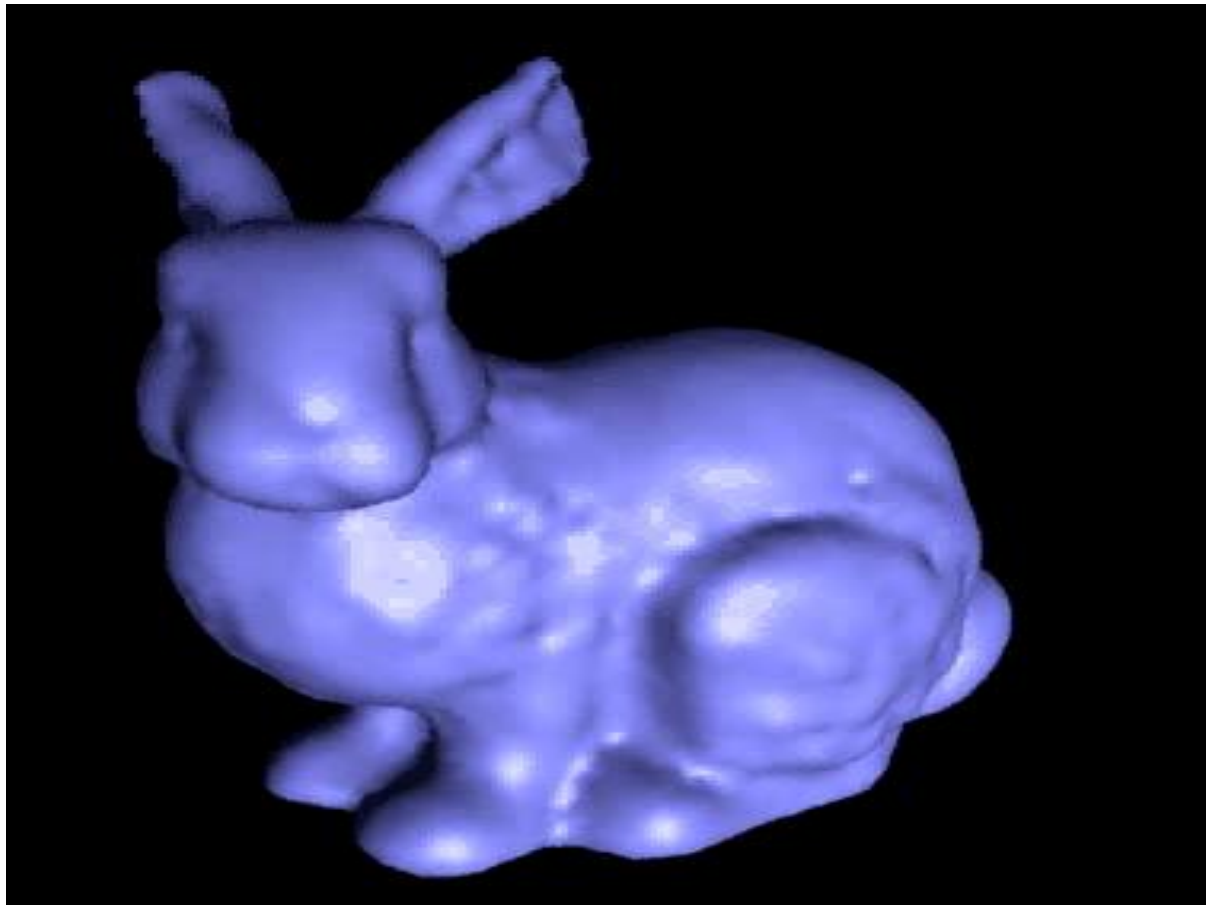
- 在边界(O)和法向(+)约束的位置中，**加入第三个坐标**。即，(x, y) --->(x, y, t)。对于第一个形状的所有约束，把t设成0；对于第二个形状的所有约束，把t设成 t_{max}；创建**三维隐式曲面**后，$0 < t < t_{max}$之间的截面序列，即中间帧形状。

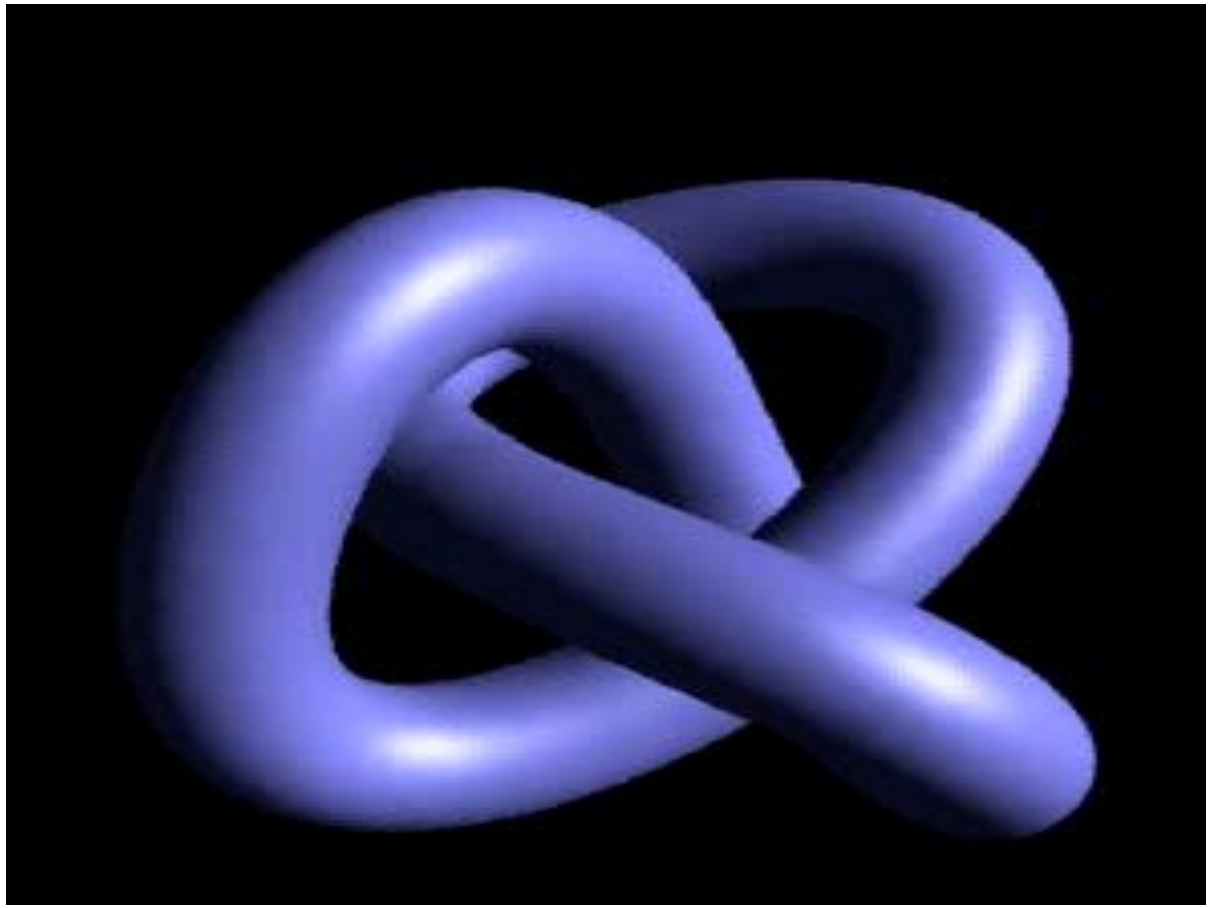
函数构建和插值的统一——三维情形



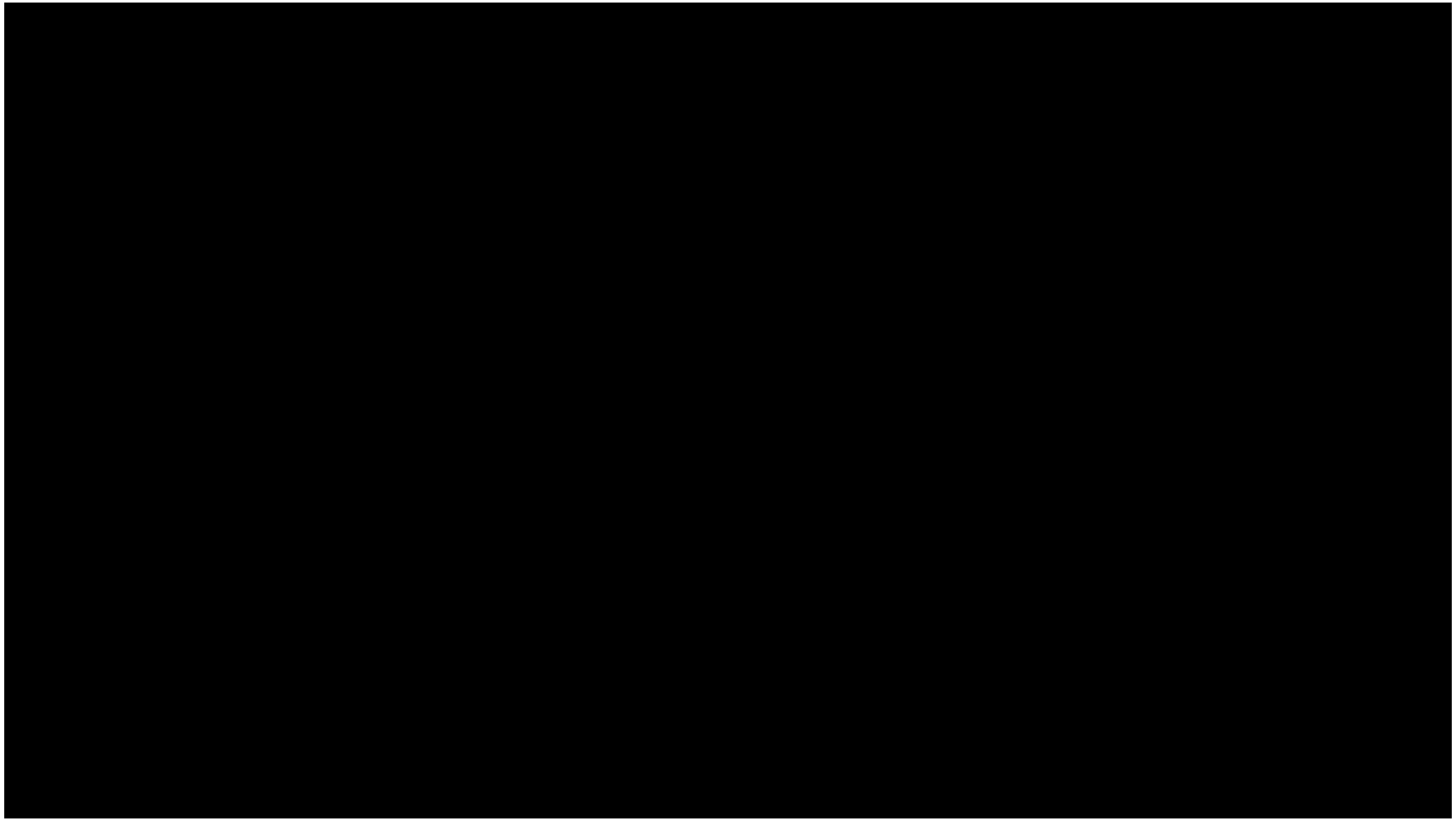
- 在边界(O)和法向(+)约束的位置中，**加入第四个坐标**。即，(x, y, z) --->(x, y, z, t)。对于第一个物体的所有约束，把t设成0；对于第二个物体的所有约束，把t设成 t_{max}。创建**四维隐式曲面**后，则$0 < t < t_{max}$之间的截面序列，即为中间帧物体。

Demo





3D Tree Morphing



Wang, Y., Wang, L., Deng, Z., & Jin, X. (2017). Topologically consistent leafy tree morphing. Computer Animation and Virtual Worlds, 28(3-4), e1761.

总结

- 3D Morphing是电影、电视中一种常用的特效生成方法。
- 每类方法都有自己的局限性。要生成高质量、全自动的3D Morphing效果通常非常困难。

The End