中国科学技术大学
University of Science and Technology of China

GAMES 102在线课程

几何建模与处理基础

刘利刚
中国科学技术大学

中国科学技术大学

University of Science and Technology of China

GAMES102

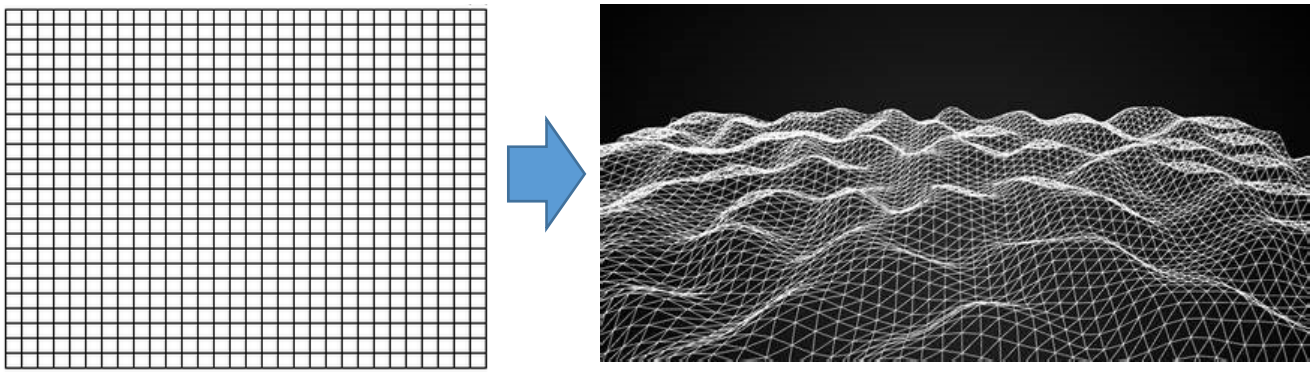
GAMES 102在线课程：几何建模与处理基础

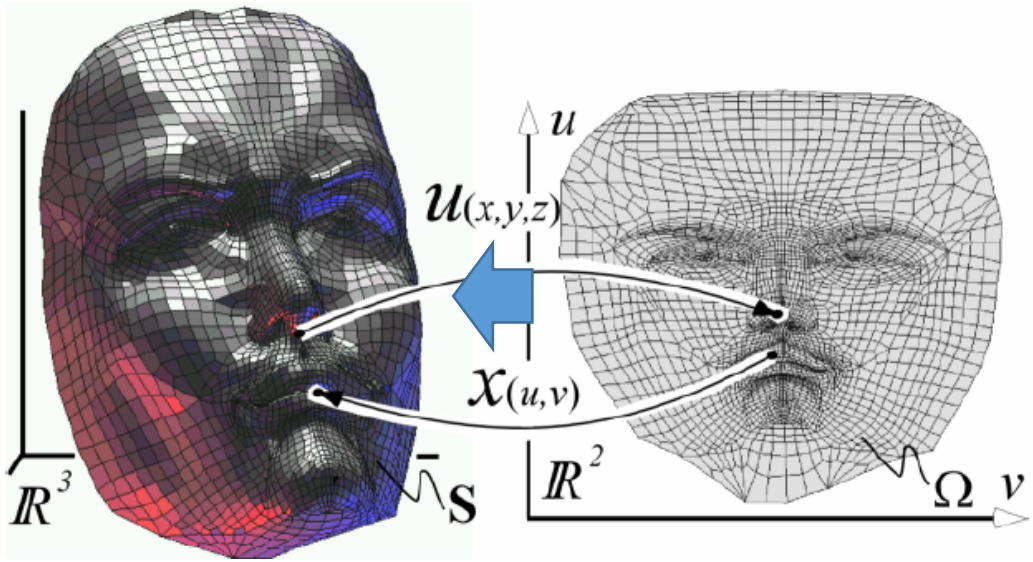
微分坐标

(Laplace坐标)

3D网格曲面：二维流形曲面的离散

- 平面图在3D空间中的嵌入 $f: R^2 \rightarrow R^3$





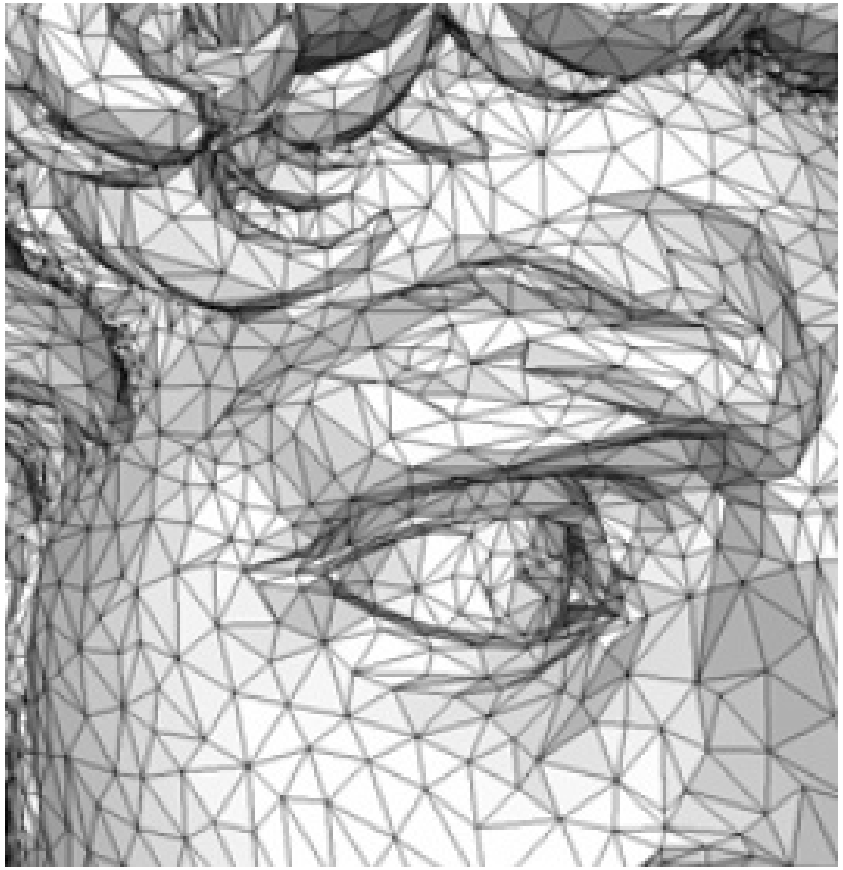
Cardinal Coordinates

- (x,y,z) coordinates

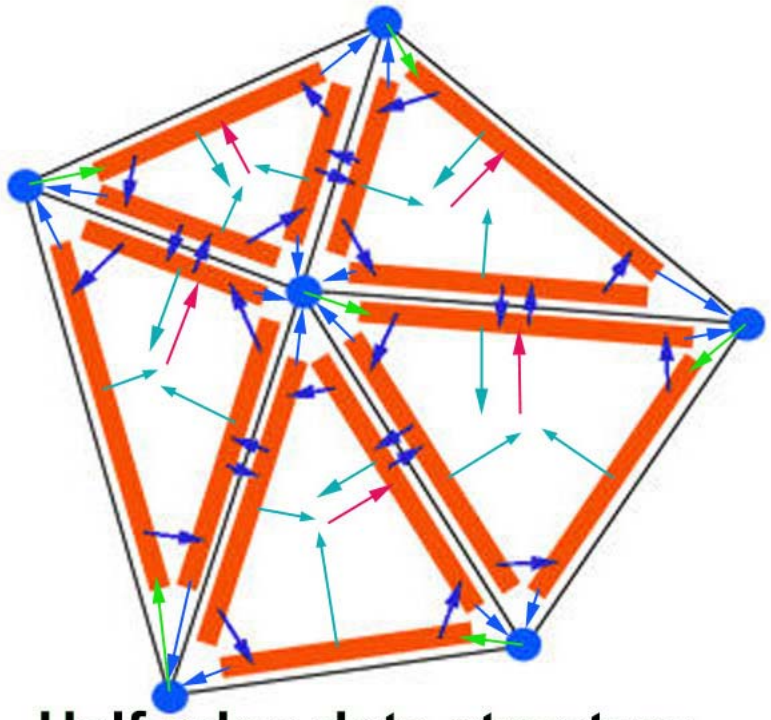
```
v 1.5 -0.960751 -1.2232
v 0.81 -0.891238 -3.74258
v 0.16 -0.233535 -2.28405
v 1.49 -2.44325 -3.6962
v 1.59 -2.98815 -4.15761
v 1.66 -2.81016 -4.10777
v 1.41 -1.14861 -1.92823
v 1 -1.40023 -3.80159
v 0.88 -1.33122 -3.83517
…

f 1 2 3
f 2 4 5
…
```

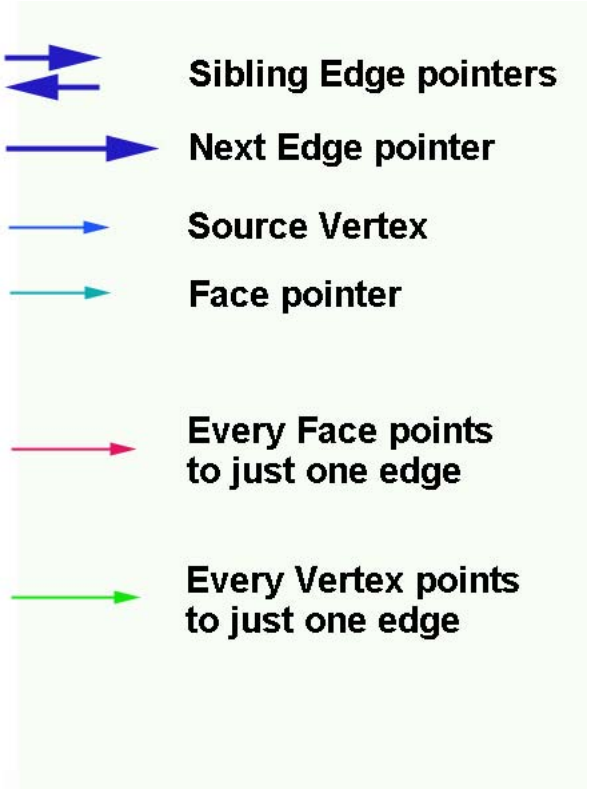
图的连接关系：点、边、面



存储数据结构：半边结构

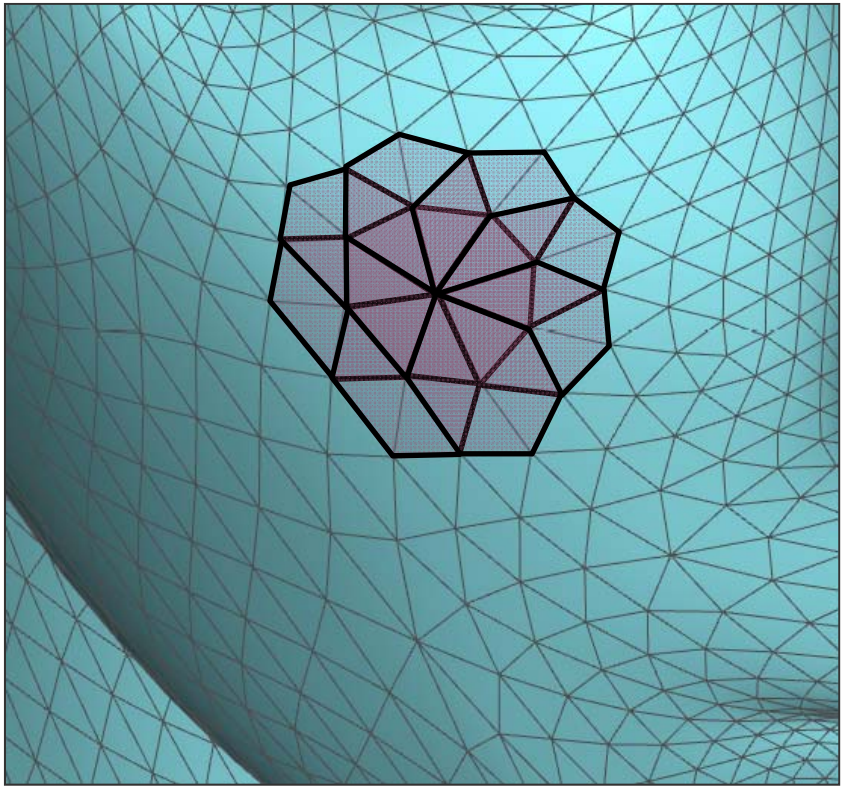


Half-edge data structure



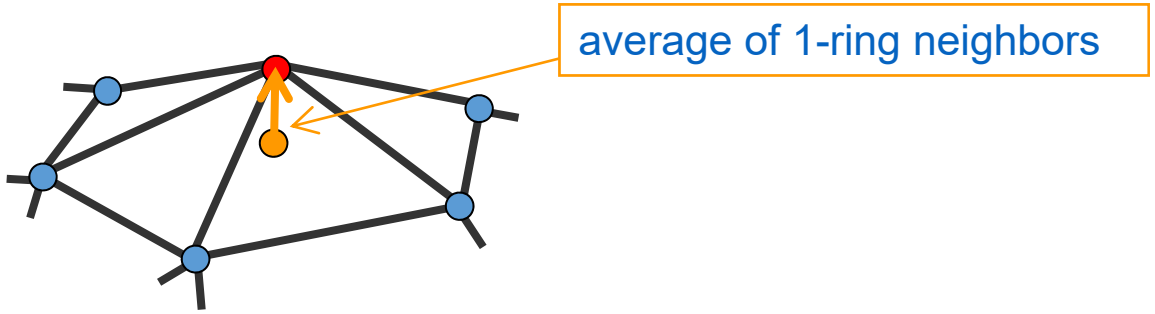
Local Structure

- Small-sized Cells
  - 1-ring neighborhood （1-邻域）
  - 一般“流形”结构也是通过局部邻域来定义



局部特征度量： 1-邻域

- Detail = surface – smooth (surface)
- Smoothing = averaging



Laplace算子(operator)

- n维欧几里得空间的二阶微分算子（椭圆型算子）
- 梯度∇f的散度$\nabla \cdot f$

$$\Delta f = \nabla \cdot \nabla f = \nabla^2 f$$

- 在笛卡尔坐标系中，为所有非混合二阶偏导数:

$$\Delta f = \sum_{i=1}^{n} \frac{\partial^2 f}{\partial x_i^2}$$

- 特别地，对二元实函数$f(x, y)$:

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

Laplace-Beltrami 算子

- 推广：定义在黎曼流形上的椭圆型算子

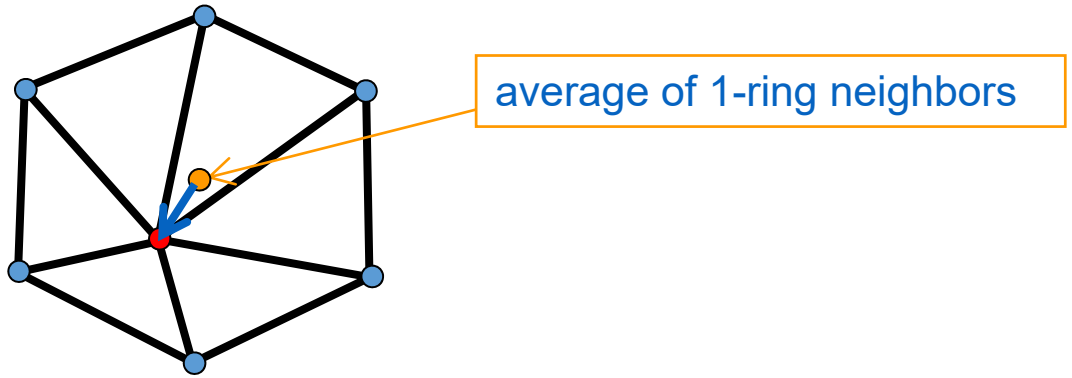
$$\nabla^2 f = \nabla \cdot \nabla f.$$

$$\nabla^2 f = \frac{1}{\sqrt{|g|}} \partial_i \left(\sqrt{|g|} g^{ij} \partial_j f \right).$$

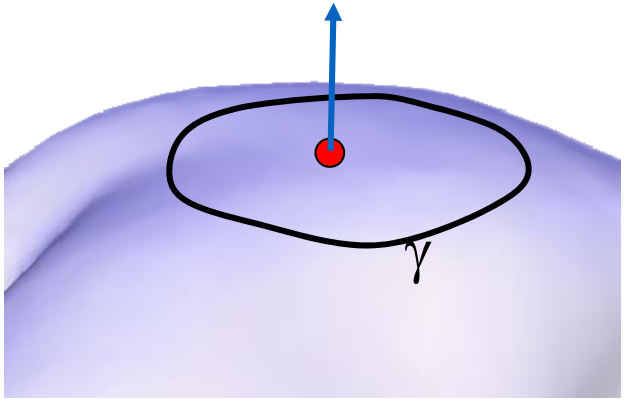
Differential Coordinates （微分坐标）

- 离散形式的 Laplacian 算子 (Umbrella Operator，伞型算子):

$$\delta_i = v_i - \sum_{j \in N(i)} w_j v_j$$

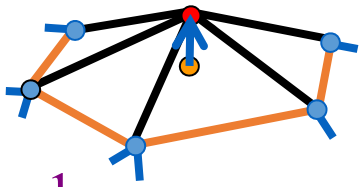


平均曲率流定理



$$\frac{1}{len(\gamma)}\int_{\mathbf{v}\in\gamma}(\mathbf{v_i}-\mathbf{v})ds$$

离散形式：

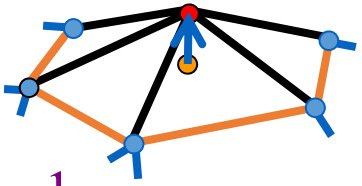


$$\boldsymbol{\delta}_\mathbf{i}=\frac{1}{d_i}\sum_{\mathbf{v}\in N(i)}(\mathbf{v_i}-\mathbf{v})$$

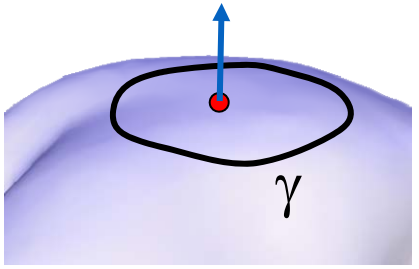
$$\lim_{len(\gamma)\to 0}\frac{1}{len(\gamma)}\int_{\mathbf{v}\in\gamma}(\mathbf{v_i}-\mathbf{v})ds = H(\mathbf{v_i})\mathbf{n_i}$$

Geometric Meaning

- DCs represent the **local** detail / local shape description
  - The direction approximates the normal
  - The size approximates the mean curvature



$$\boldsymbol{\delta}_i = \frac{1}{d_i} \sum_{\mathbf{v} \in N(i)} (\mathbf{v}_i - \mathbf{v})$$



$$\frac{1}{len(\gamma)} \int_{\mathbf{v} \in \gamma} (\mathbf{v}_i - \mathbf{v}) ds$$

$$\lim_{len(\gamma) \to 0} \frac{1}{len(\gamma)} \int_{\mathbf{v} \in \gamma} (\mathbf{v}_i - \mathbf{v}) ds = H(\mathbf{v}_i)\mathbf{n}_i$$

Weighting Schemes (Barycentric coordinates)

- Uniform weight (geometry oblivious)

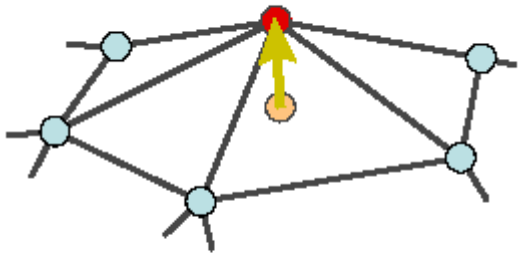
$$w_j = 1$$

- Cotangent weight (geometry aware)

$$w_j = (\cot\alpha + \cot\beta)$$

- Normalization

$$w_j = \frac{w_j}{\sum_j w_j}$$

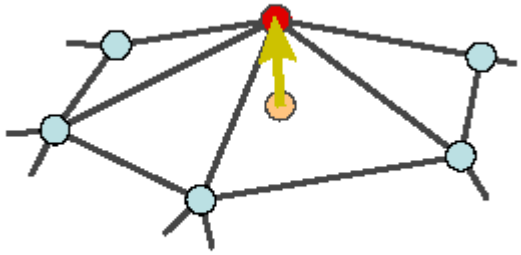


$$\delta_i = \frac{1}{d_i}\sum_{j\in N(i)}(\mathbf{v}_i - \mathbf{v}_j)$$

Local Laplacian Smoothing

Differential Coordinates (Laplace Coordinates)

- Represent local detail at each surface point
  - better describe the shape
- Linear transition from global to differential
- Useful for operations on surfaces where surface details are important

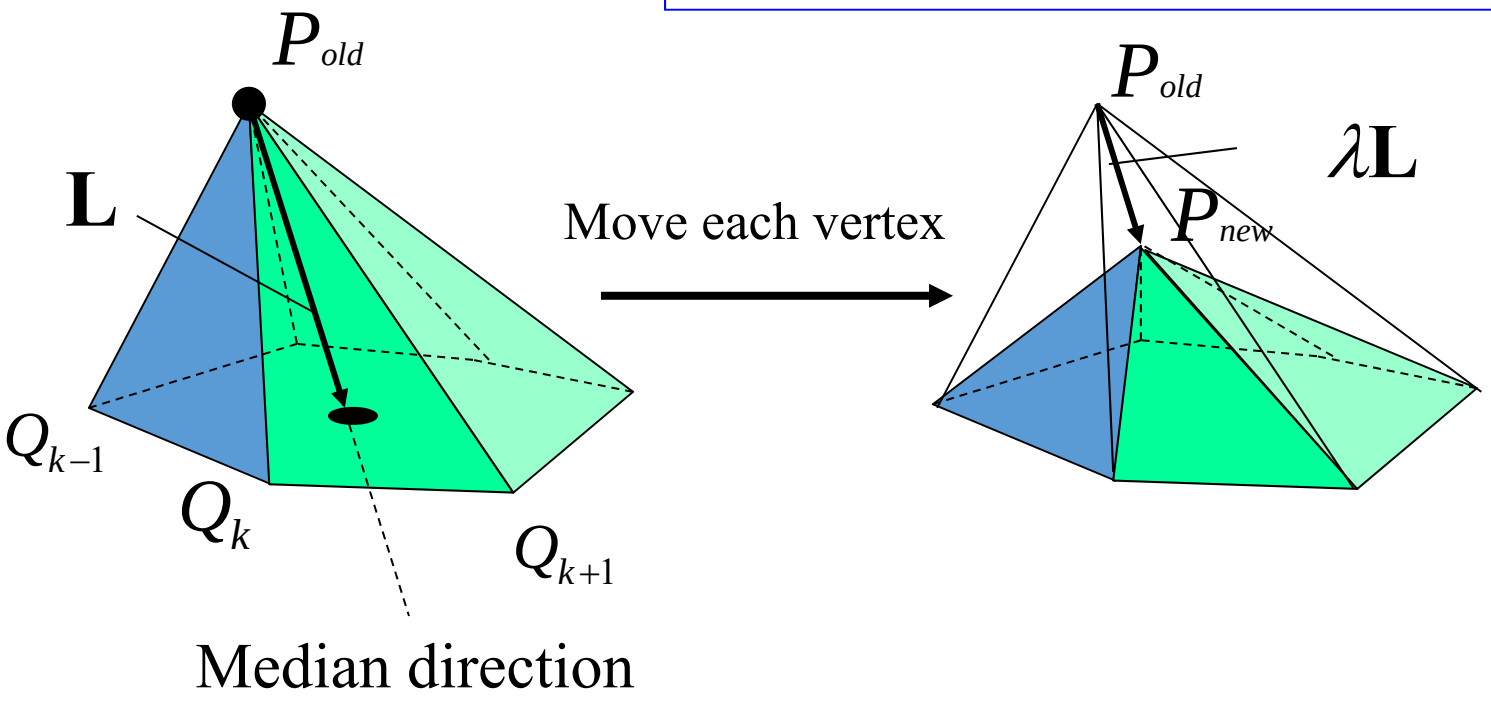


$$\delta_i = \frac{1}{d_i} \sum_{j \in N(i)} (\mathbf{v}_i - \mathbf{v}_j)$$

Laplacian Smoothing Flow

$$P_{new} \leftarrow P_{old} + \lambda \mathbf{L}(P_{old})$$

Average of the vectors to neighboring vertices

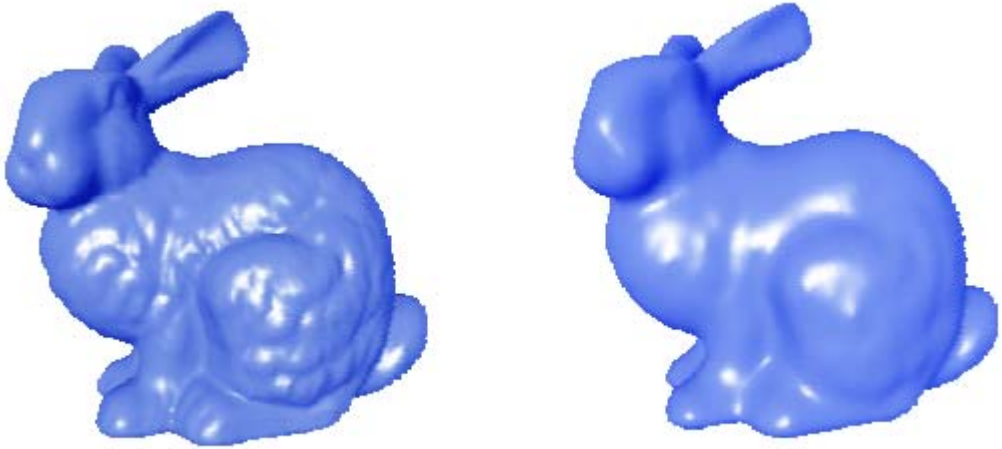


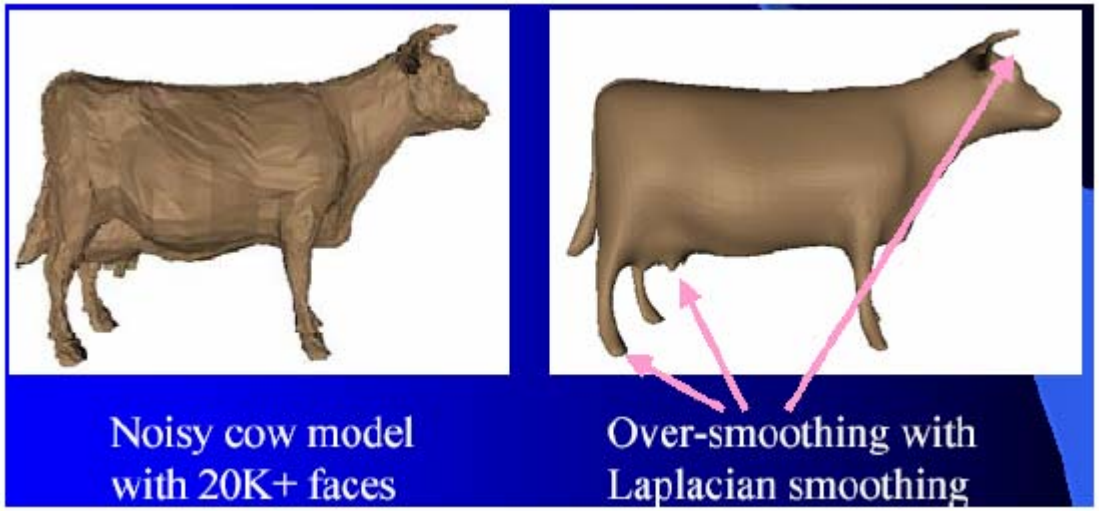
Laplacian Smoothing

$$P^{new} = P^{old} + \lambda L\left(P^{old}\right)$$

- Equivalent to box filter in signal processing
- Apply to all vertices on mesh
- Typically repeat several times
- Can describe as energy minimization
  - Energy = sum of squared edge lengths in mesh
  - Parameter $\lambda > 0$ controls convergence "speed"

Laplacian Smoothing – Example

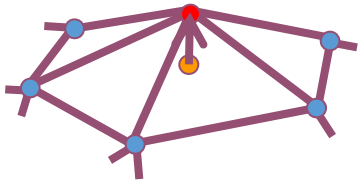


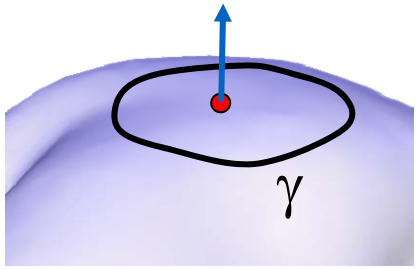


Problem of Over-smoothing

How to find appropriate λ and number of iterations?

Mean Curvature Flow





$$\boldsymbol{\delta}_\mathbf{i} = \frac{1}{d_i} \sum_{\mathbf{v} \in N(i)} \left(\mathbf{v}_\mathbf{i} - \mathbf{v} \right)$$

$$\frac{1}{len(\gamma)} \int_{\mathbf{v} \in \gamma} \left(\mathbf{v}_\mathbf{i} - \mathbf{v} \right) ds$$

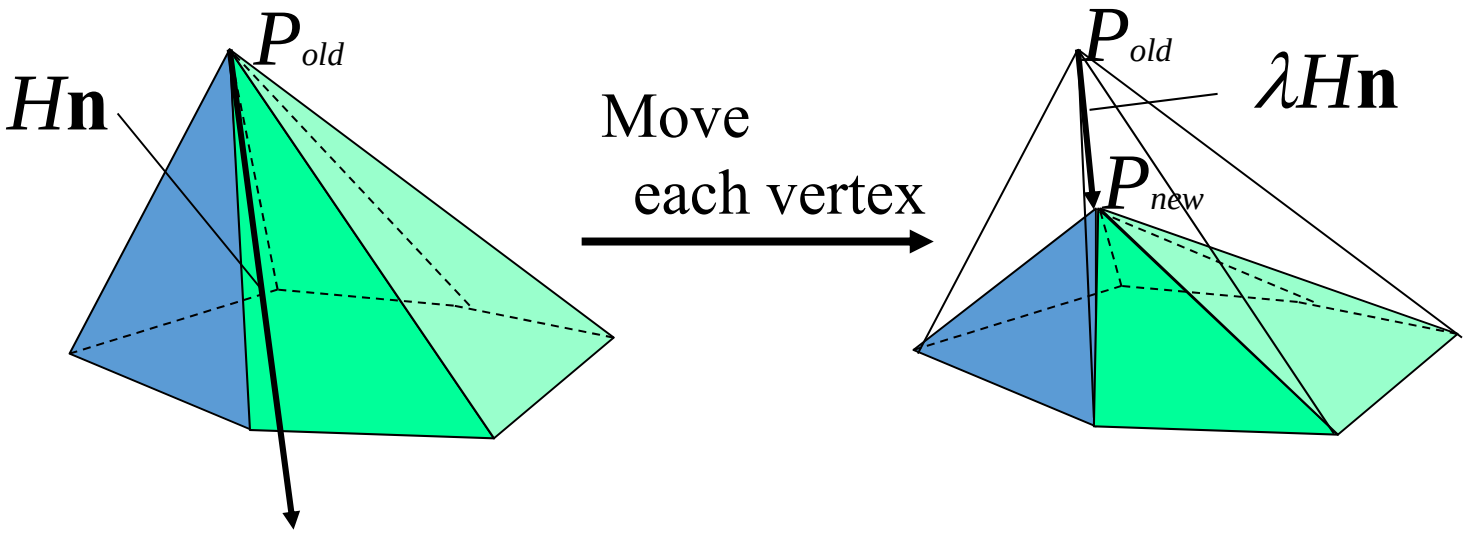
$$\lim_{len(\gamma) \to 0} \frac{1}{len(\gamma)} \int_{\mathbf{v} \in \gamma} \left(\mathbf{v}_\mathbf{i} - \mathbf{v} \right) ds \;=\; H(\mathbf{v}_\mathbf{i}) \mathbf{n}_\mathbf{i}$$

Discrete Mean Curvature Flow

$$P_{new} \leftarrow P_{old} + \lambda H(P_{old})\,\mathbf{n}(P_{old})$$

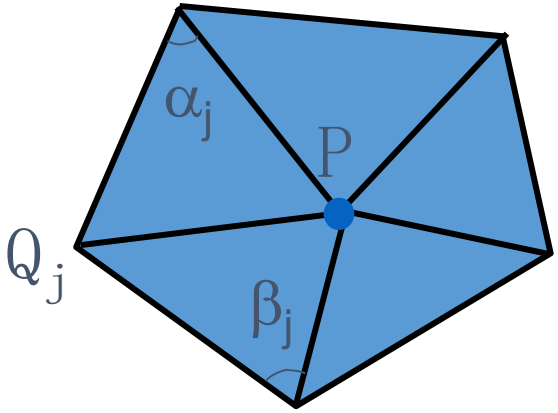
Speed = discrete mean curvature

Direction = normal



Discrete Mean Curvature

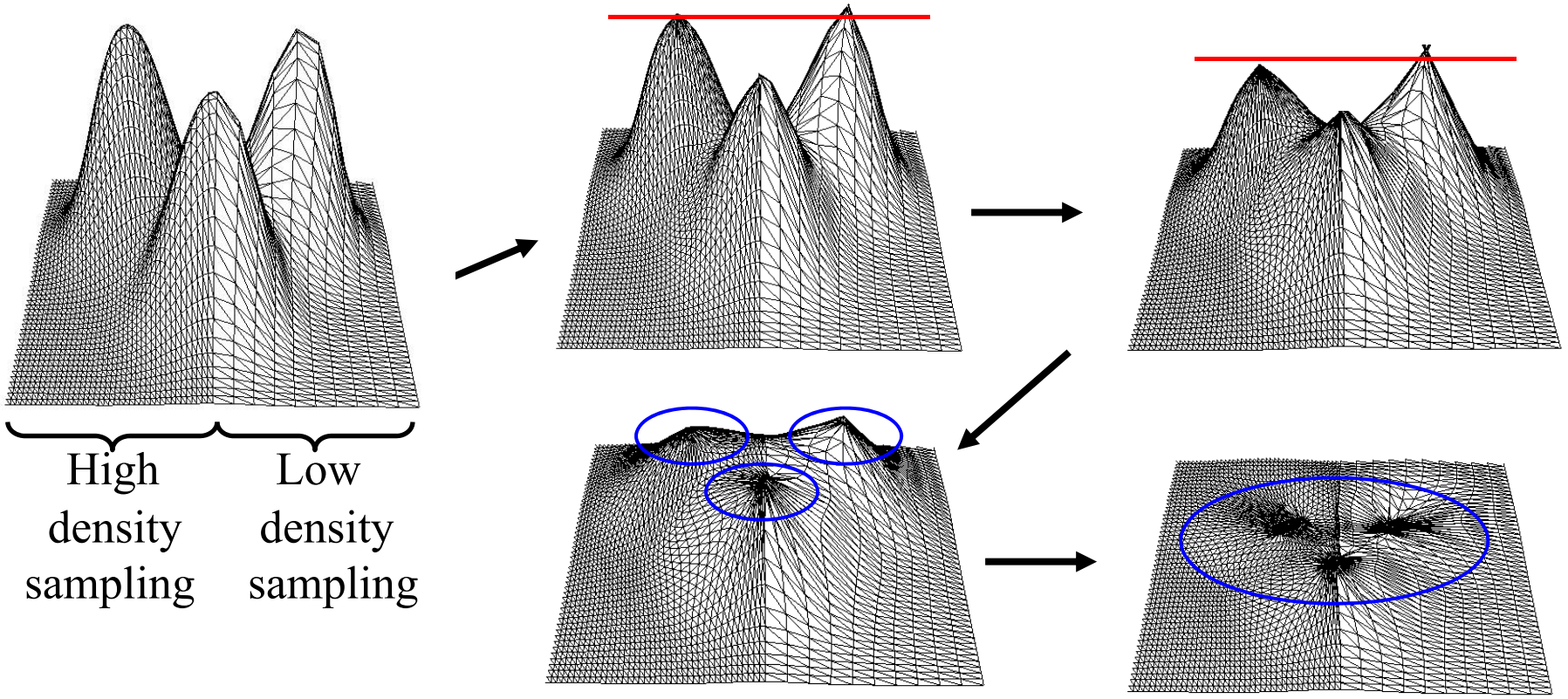
$$H\mathbf{n} = \frac{\nabla_P \mathrm{A}}{2\mathrm{A}}$$



$$H\mathbf{n} = \frac{1}{4\mathrm{A}} \sum_j (\cot \alpha_j + \cot \beta_j)(\mathbf{P} - \mathbf{Q}_j)$$

Properties of Mean Curvature Flow

- Increases mesh irregularity.
- Doesn't develop unnatural deformations



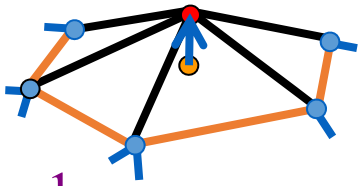
Global Laplacian Smoothing

极小曲面(minimal surface)

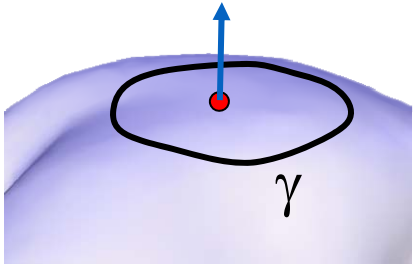
- 平均曲率处处为0

$$H(\mathbf{v}_i) = 0, \qquad \forall i$$

➡ 微分坐标=0



$$\delta_i = \frac{1}{d_i} \sum_{\mathbf{v} \in N(i)} (\mathbf{v}_i - \mathbf{v})$$



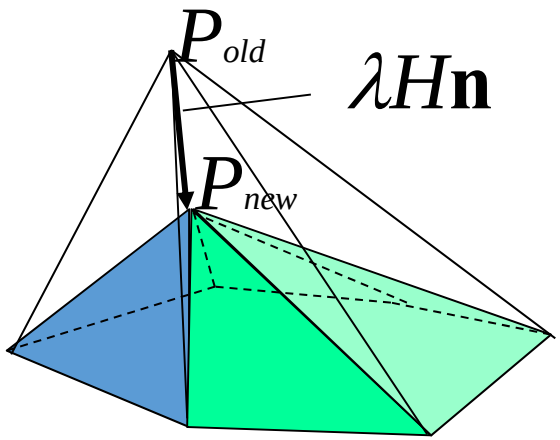
$$\frac{1}{len(\gamma)} \int_{\mathbf{v} \in \gamma} (\mathbf{v}_i - \mathbf{v}) ds$$

$$\lim_{len(\gamma) \to 0} \frac{1}{len(\gamma)} \int_{\mathbf{v} \in \gamma} (\mathbf{v}_i - \mathbf{v}) ds = H(\mathbf{v}_i)\mathbf{n}_i$$

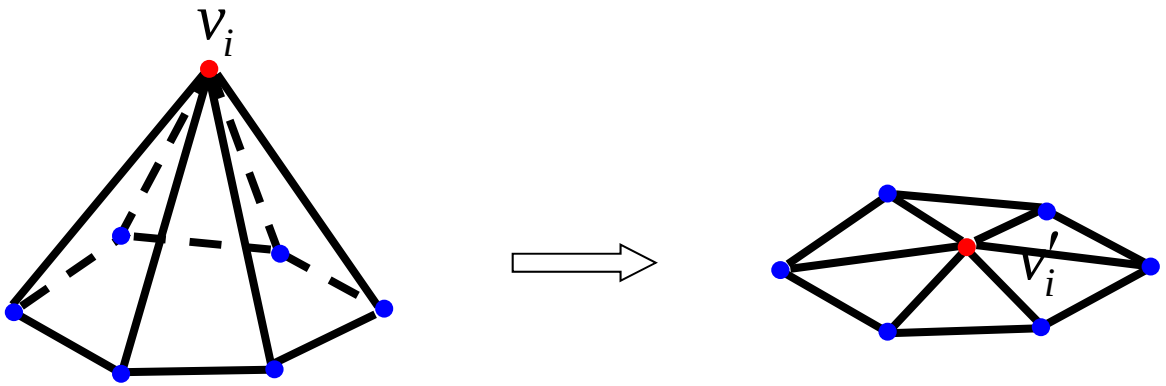
局部迭代光顺方法的问题?

- 注：通过作业6中已发现

$$P_{new} \leftarrow P_{old} + \lambda \boxed{H(P_{old})} \boxed{\mathbf{n}(P_{old})}$$



微分坐标一致为0



$$L(v_i) = v_i - \sum_{j \in N(i)} \omega_{ij} v_j = 0$$

$\omega_{cotangent}$: $\omega_{ij} = \cot \alpha_{ij} + \cot \beta_{ij}$

- 所有顶点的方程联立，得到网格曲面的整体Laplacian方程：

$$Ax = 0$$

Laplacian Matrix

- The transition between the δ and *xyz* is linear:

$$\begin{pmatrix} \\ \mathrm{L} \\ \\ \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} \delta_1^{(x)} \\ \delta_2^{(x)} \\ \vdots \\ \vdots \\ \delta_n^{(x)} \end{pmatrix}$$

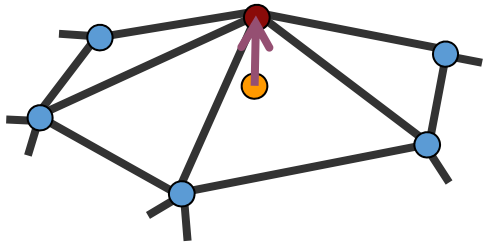
$$A_{ij} = \begin{cases} 1 & i \in N(j) \\ 0 & otherwise \end{cases}$$

$$D_{ij} = \begin{cases} d_i & i = j \\ 0 & otherwise \end{cases}$$

$$\mathrm{L} = \mathrm{I} - \mathrm{D}^{-1}\mathrm{A}$$

Laplacian matrix

- The transition between the δ and *xyz* is linear:



$$\boldsymbol{\delta}_i = \sum_{j \in N(i)} w_{ij} \left(\mathbf{v}_i - \mathbf{v}_j \right)$$

$$\mathbf{L}\,\mathbf{v}_x = \boldsymbol{\delta}_x$$

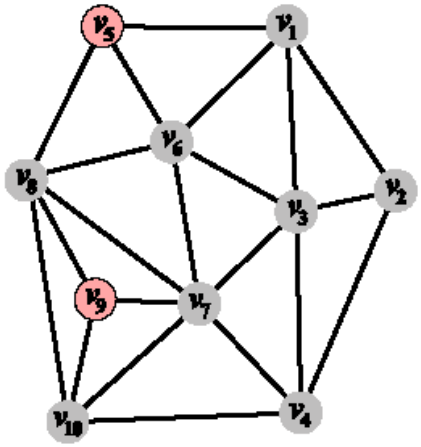
$$\mathbf{L}\,\mathbf{v}_y = \boldsymbol{\delta}_y$$

$$\mathbf{L}\,\mathbf{v}_z = \boldsymbol{\delta}_z$$

Reconstruction

- From relative coordinates to absolute coordinates.
- Solving a sparse linear system

$$Lv = \delta$$



The mesh

4	−1	−1		−1	−1				
−1	3	−1	−1						
−1	−1	5	−1		−1	−1			
	−1	−1	4			−1			−1
−1				3	−1		−1		
−1		−1			4	−1	−1		
		−1	−1		−1	6	−1	−1	−1
				−1	−1	−1	6	−1	−1
						−1	−1	3	−1
			−1			−1	−1	−1	4

The symmetric Laplacian L_s

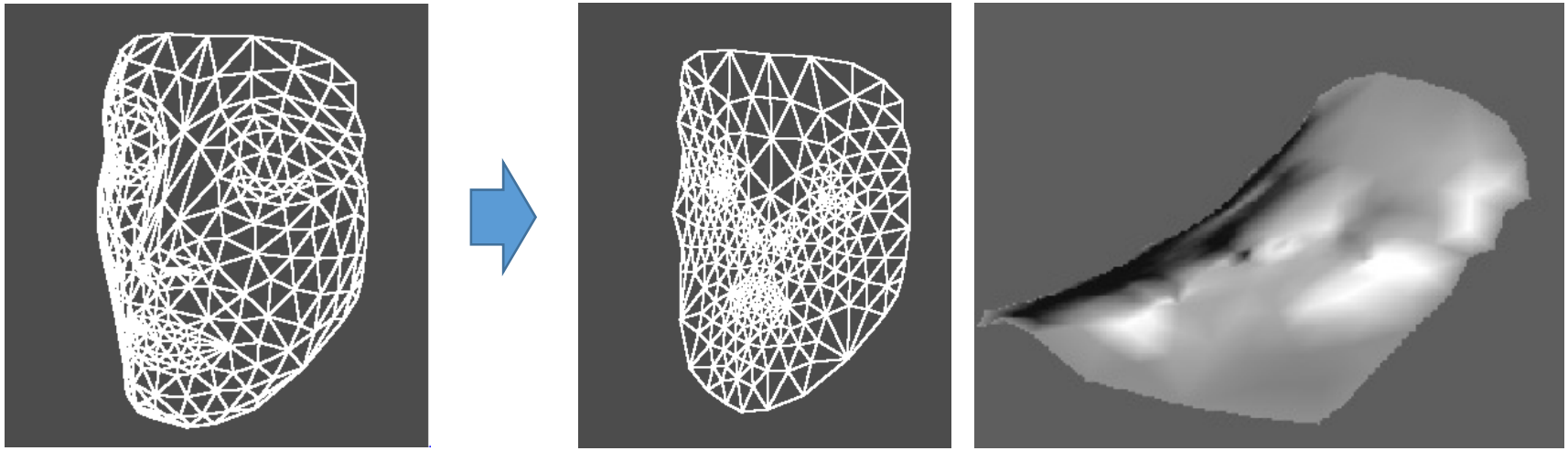
Basic properties

- Rank(L) = n-c (n-1 for connected meshes)
- We can reconstruct the xyz geometry from delta up to translation

$$L\mathbf{x} = \boldsymbol{\delta}$$

极小曲面生成的全局方法

- 检测边界，固定边界
- 构建稀疏方程组 ($\delta = 0$)
- 求解稀疏方程组
  - 【注：有高效的求解方法，且有成熟的数学库可使用】
- 更新内部顶点坐标

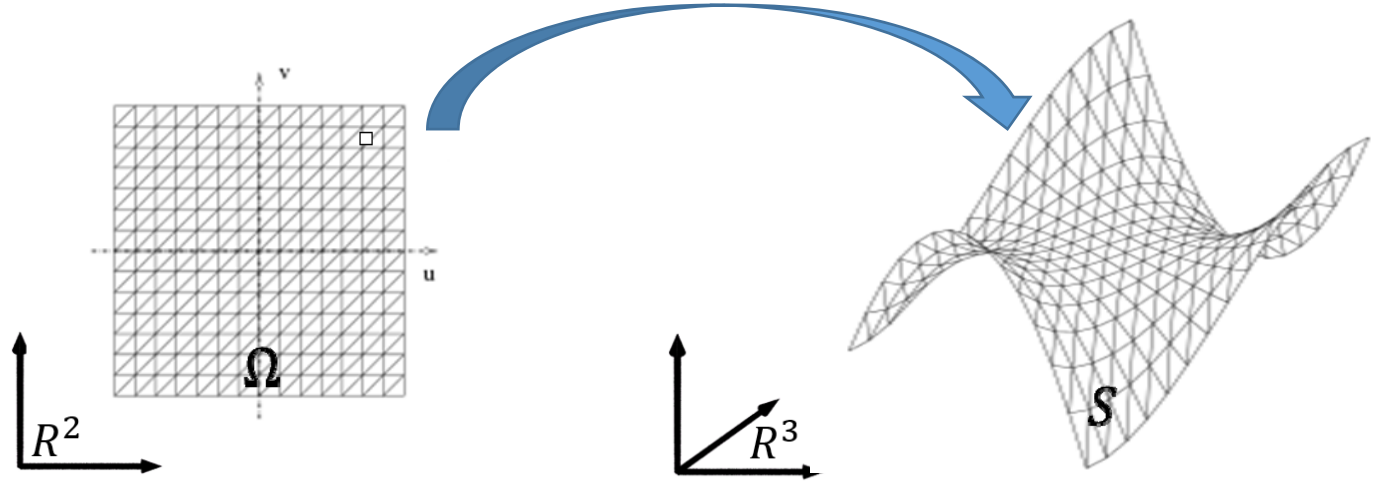


Mesh Parameterization (Mesh Flattening)

曲面参数化

2D Manifold Surfaces in R^3

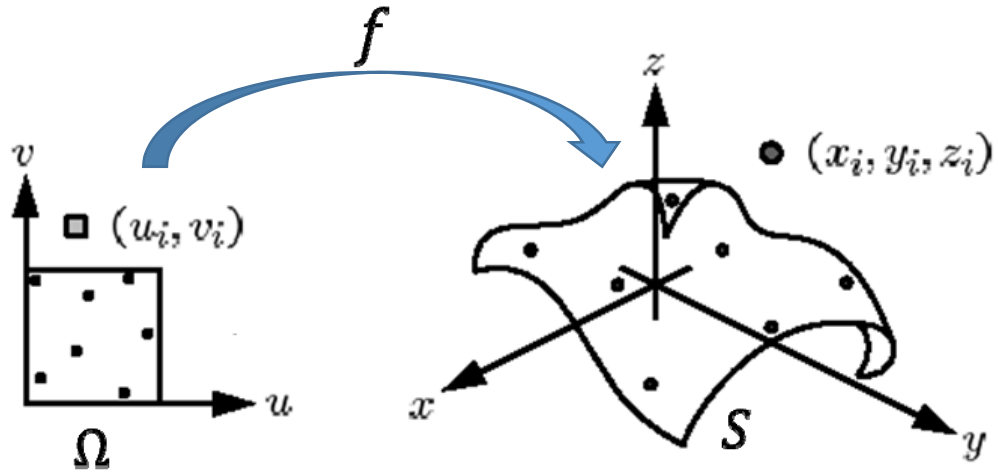
- A surface S in R^3 has an intrinsic dimension of 2D
  - a patch Ω in R^2 is embedded into R^3 (each point in Ω is assigned a position in R^3)



Parameter domain

Embedded (ambient) space

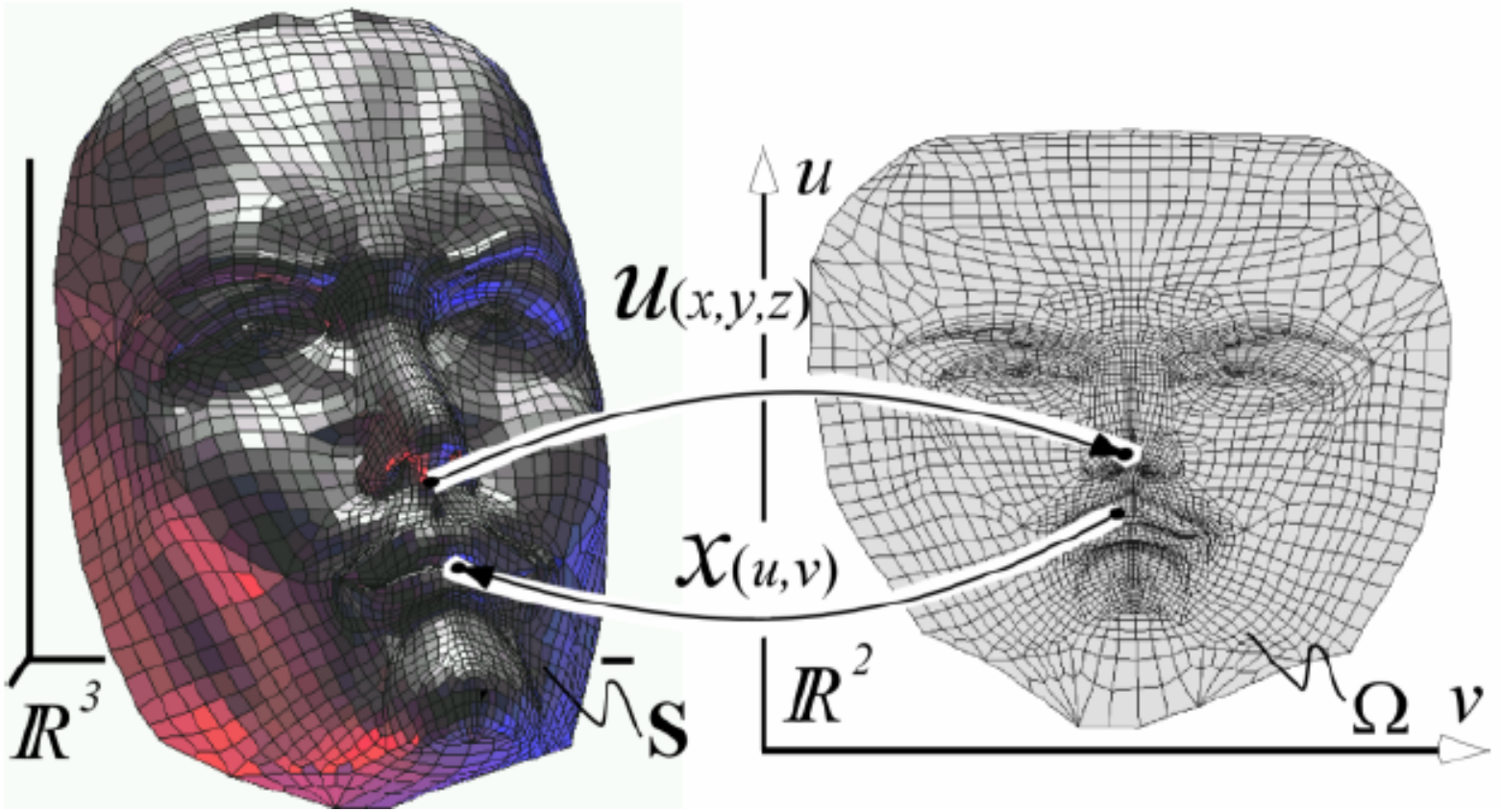
Parametric Surfaces



$$f: \Omega \to S$$

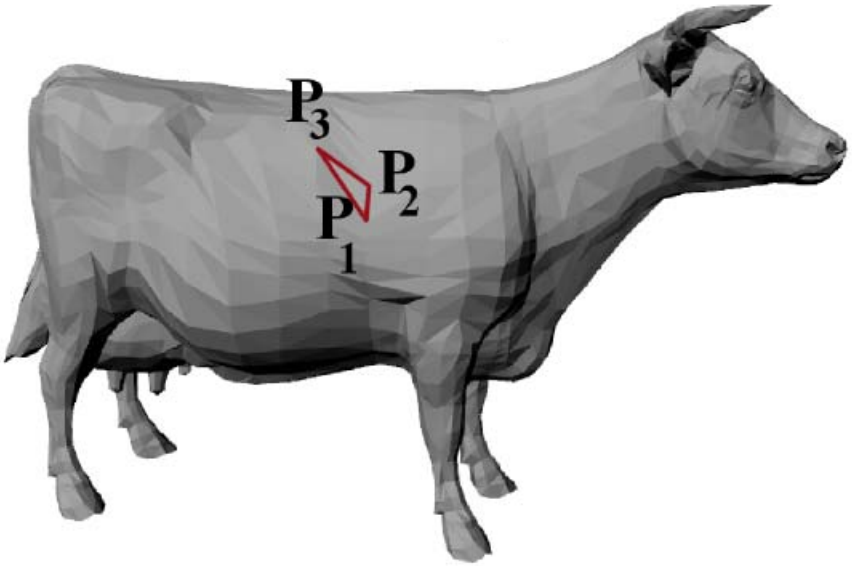
$$(u, v) \mapsto \begin{cases} x = x(u, v), \\ y = y(u, v), \\ z = z(u, v), \end{cases}$$

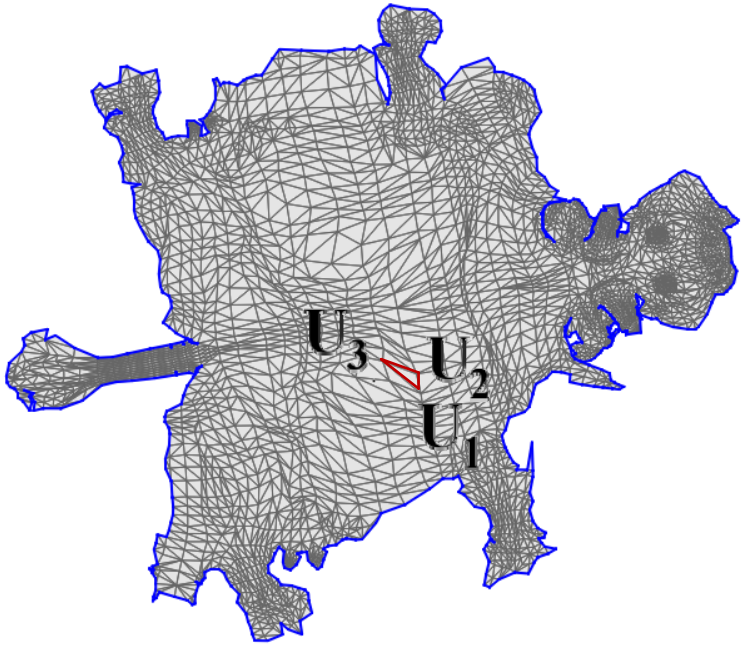
曲面展开（参数化）



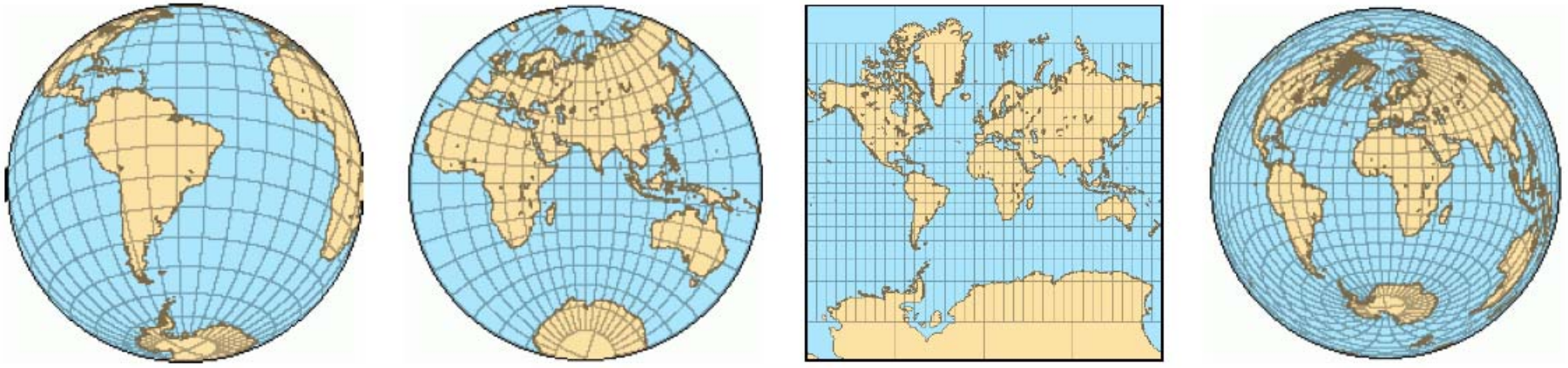
参数化：将曲面展开成平面

- 每个3D顶点(x,y,z)对应一个2D点(u,v)
  - (u,v) 称为 (x,y,z) 的参数（2D流形曲面的本征维数）





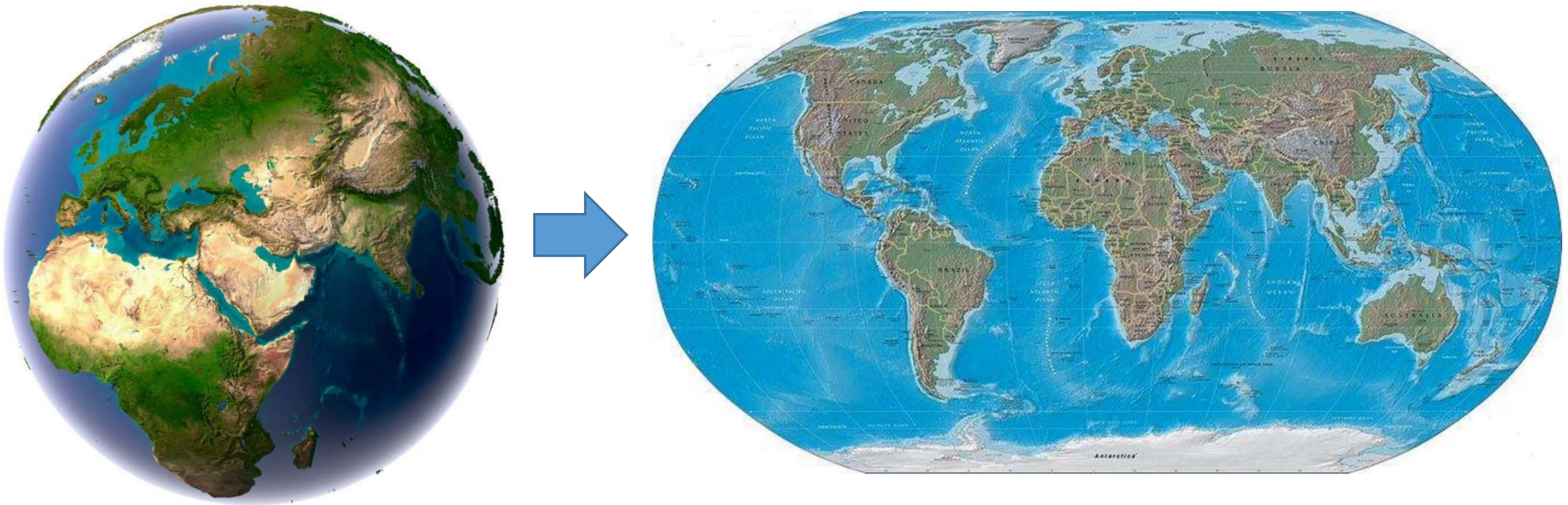
History



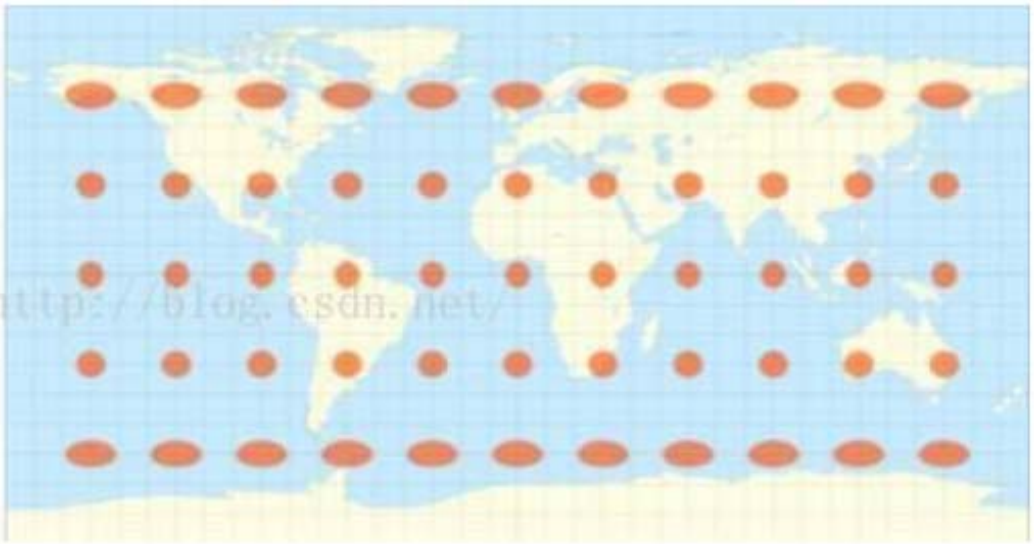
(a) (b) (c) (d)

地图：将地球“展平”成平面

- 3D地球表面和2D地图的点有一一对应



但，球面是不可展的，必有形变



思考：
1. 地图上的两点之间的距离是真实距离吗？哪些地方的距离可信度更高？
2. 地图上看，哪些区域的面积被放大了？

思考：北京和纽约的最短路径是什么？

将球面割开展平的不同方式

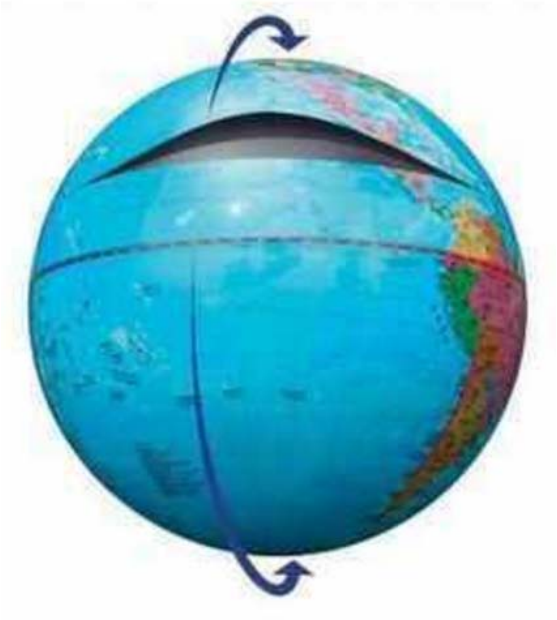
沿经线切

沿北纬15度的纬线切

传统横板地图

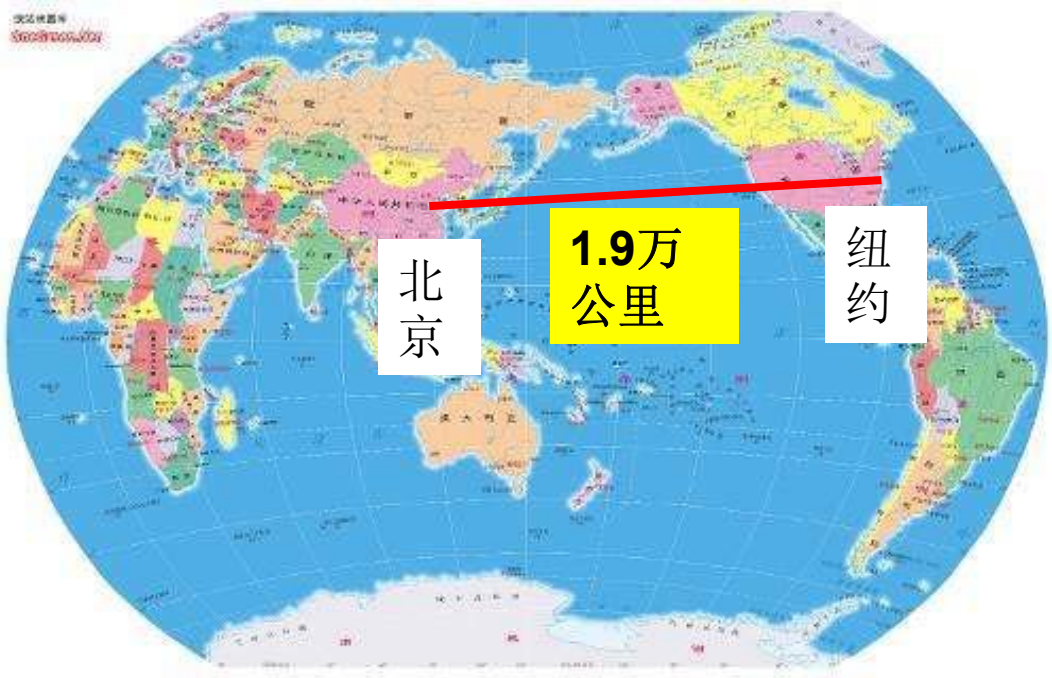
沿经线切开

新型竖板地图



沿北纬15度的纬线切开

北京-纽约的最短路径

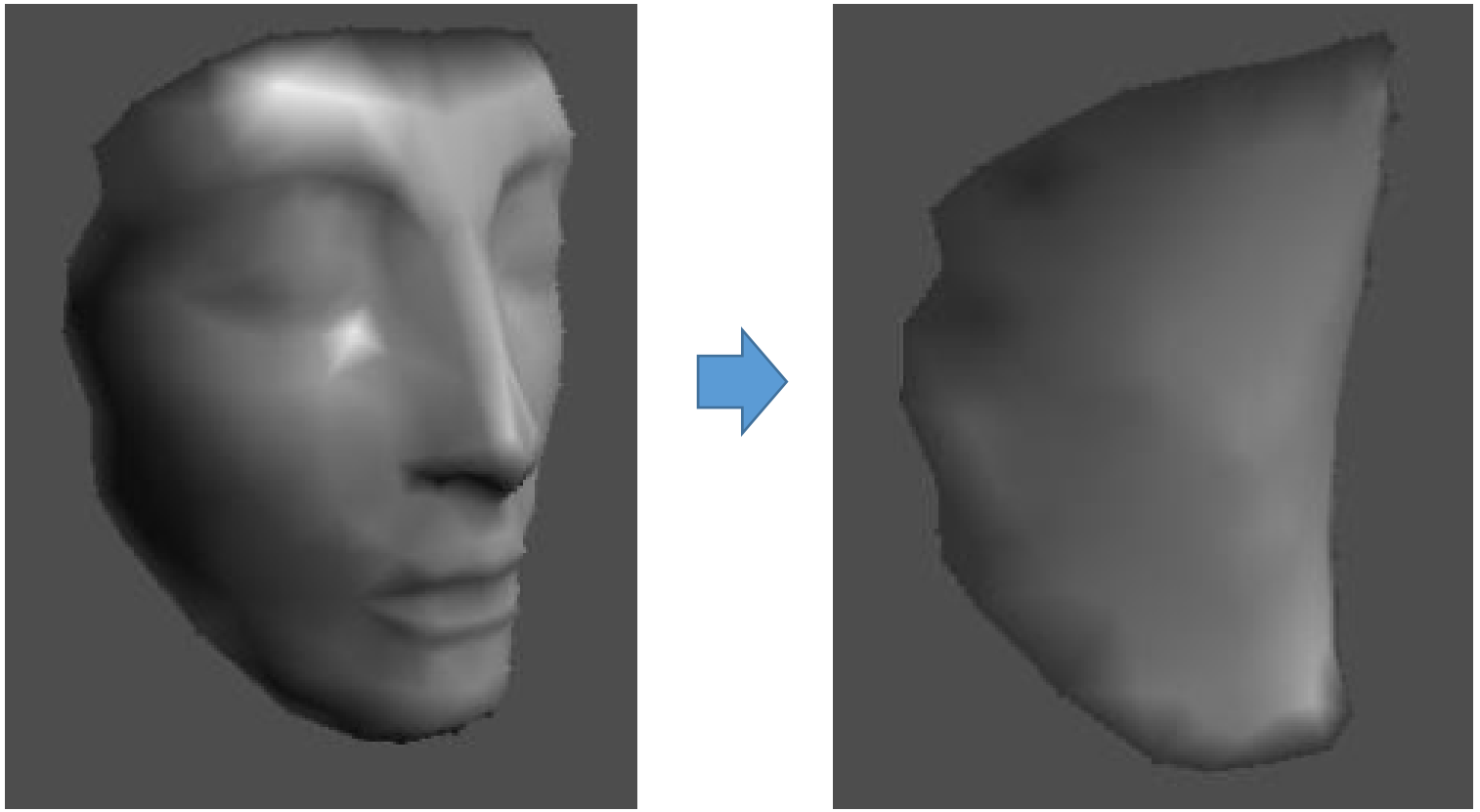


参数化是几何处理中的基本问题

- 提供了三维曲面每个点的一个二维参数
  - 本征维数参数
- 在低维来处理高维问题，减少复杂度
  - 降维
- 三维曲面之间的相关问题可通过参数化空间来处理

给定边界的极小曲面

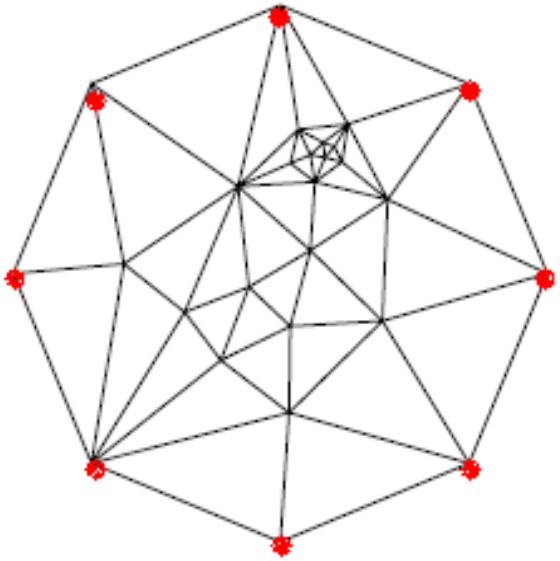
- 如果边界刚好在一个平面上（共面）？



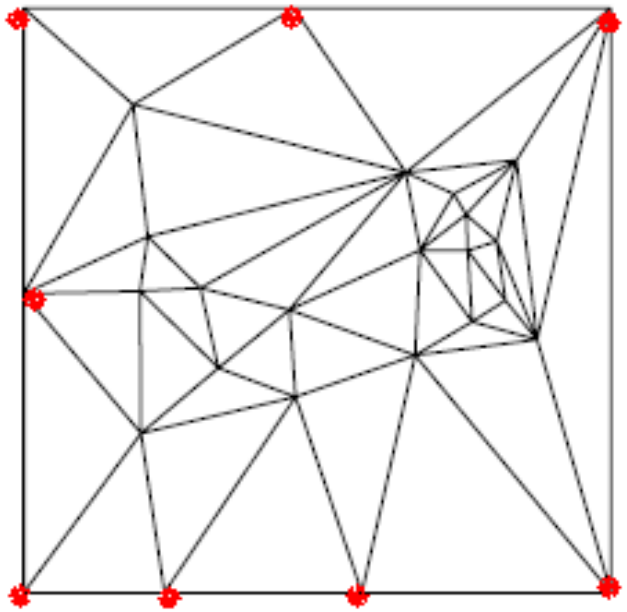
将边界映射到平面的凸多边形上

[Floater 97']

- Fixing the boundary of the mesh onto



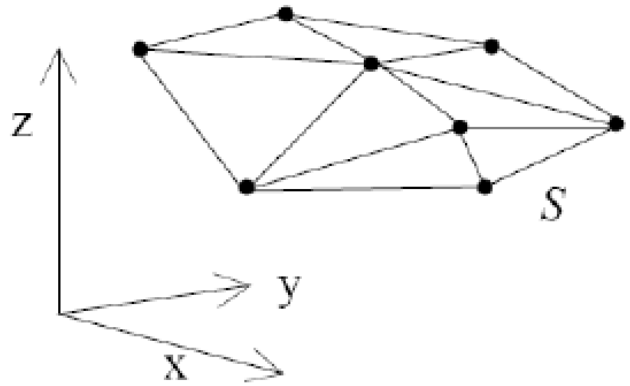
an unit circle

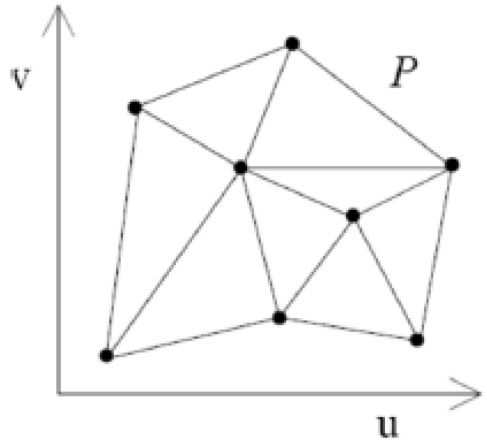


an unit square

M. Floater. Parametrization and smooth approximation of surface triangulations. CAGD, 1997.
http://www.cs.jhu.edu/~misha/Fall09/Floater97.pdf

方法：求解稀疏方程组





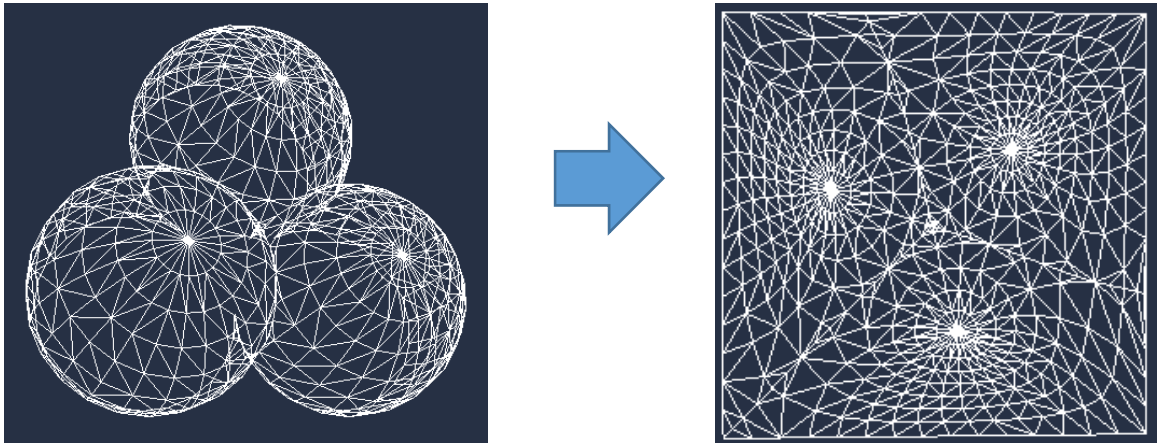
For interior mesh points:

$$p_i = \sum_{\{j:(i,j)\in \text{edges}\}} \lambda_{i,j_k} p_k, \qquad \sum_{k=1}^{d_i} \lambda_{i,j_k} = 1, \qquad \lambda_{i,j_k} > 0$$

⇒ Forming a sparse linear system

Tutte's Method: Why it Works

- Theorem [Tutte,63], [Maxwel,1864] :
  - If G=<V,E> is a 3-connected planar graph (triangular mesh) then any barycentric embedding provides a valid parameterization



如果边界位于凸多边形上，则三角形一定不会发生翻转！

参数化：操作步骤

- 检测边界
- 将边界映射到正方形边界或圆边界（凸边界）
- 构建稀疏方程组
- 求解稀疏方程组
- 更新顶点坐标
- 连接纹理图像，更新显示

Floater参数化方法

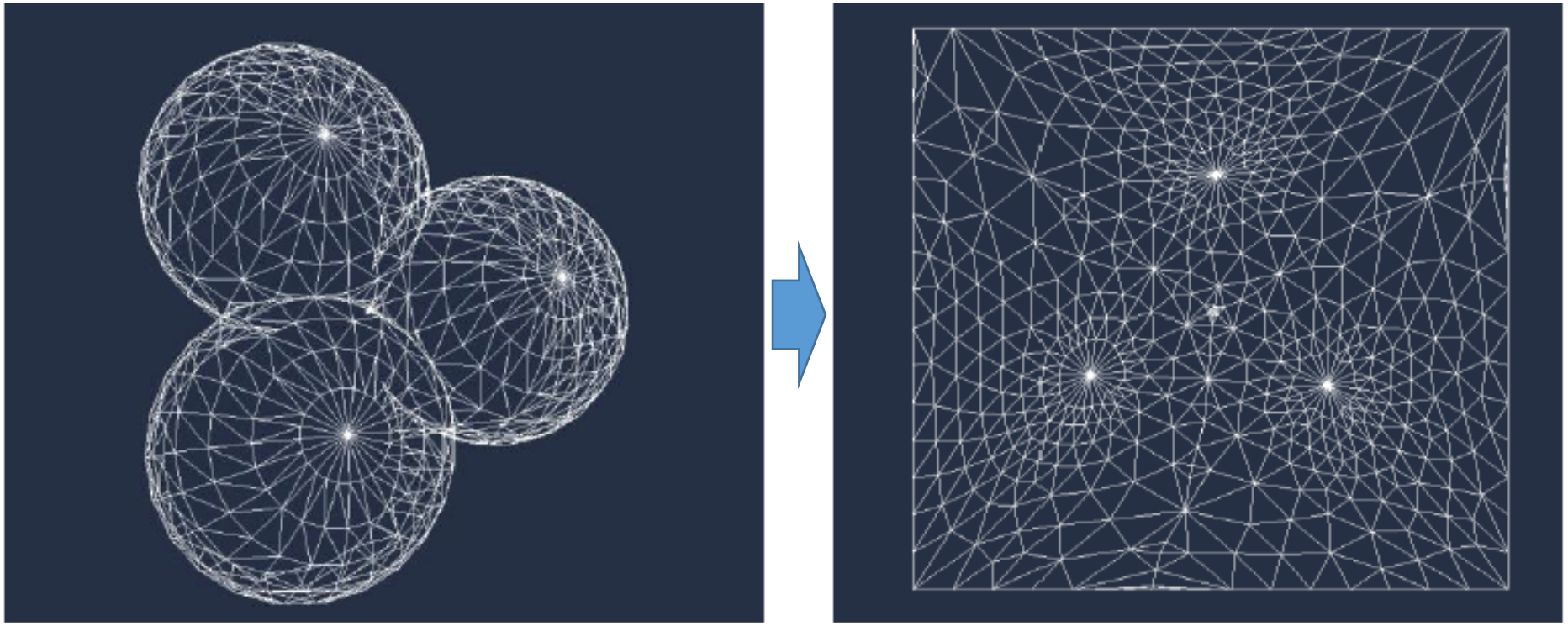
[Floater 97']

- Uniform parametrization
- Weighted least squares parametrization
- Shape-preserving parametrization

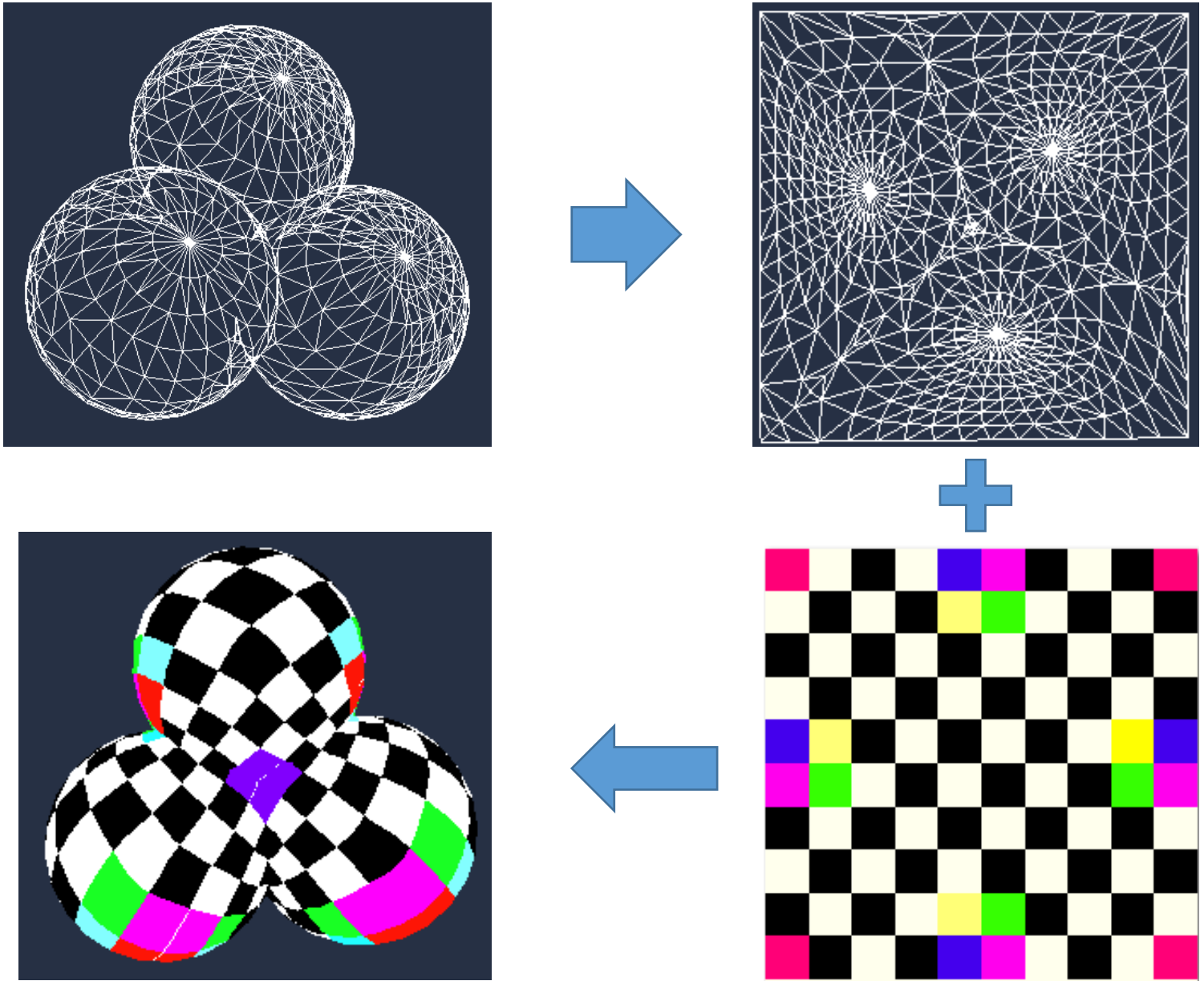
- 如何判断哪个参数化方法更好?

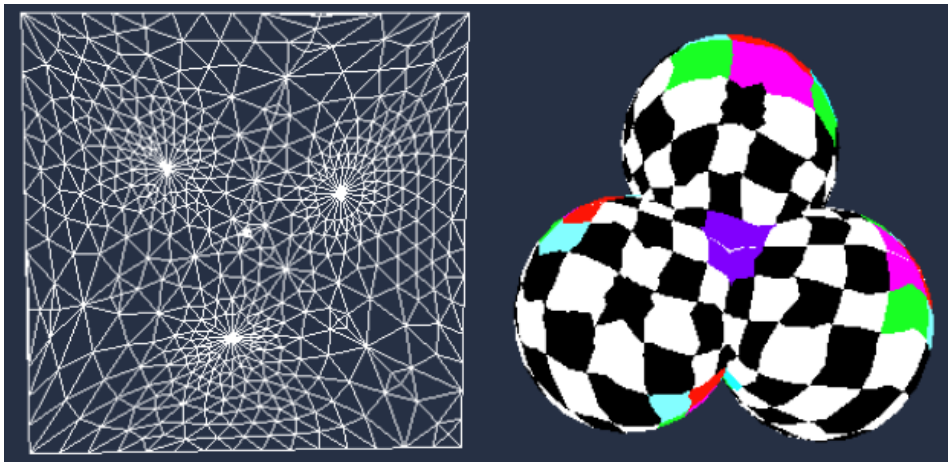
M. Floater. Parametrization and smooth approximation of surface triangulations. CAGD, 1997.
http://www.cs.jhu.edu/~misha/Fall09/Floater97.pdf

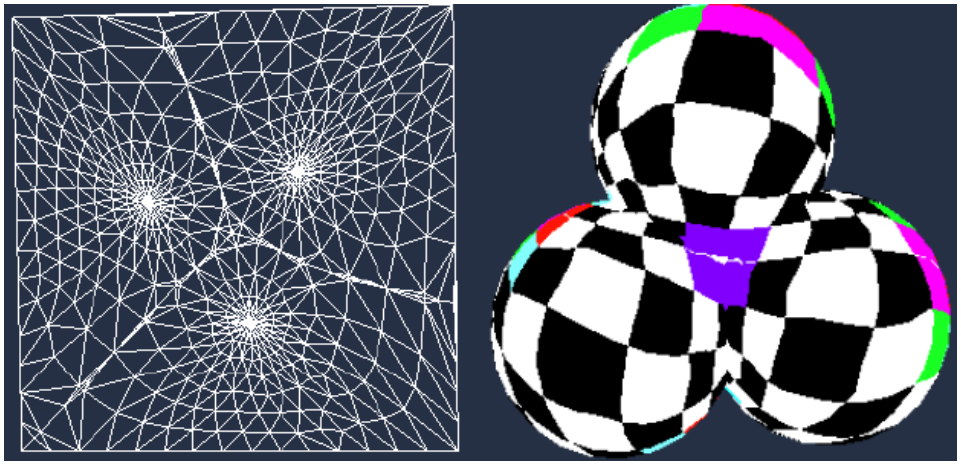
曲面展开（参数化）

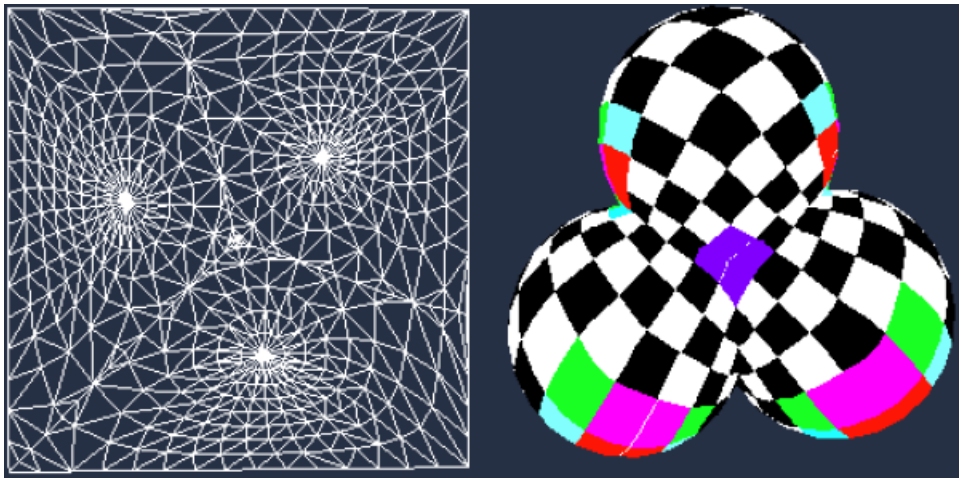


纹理映射

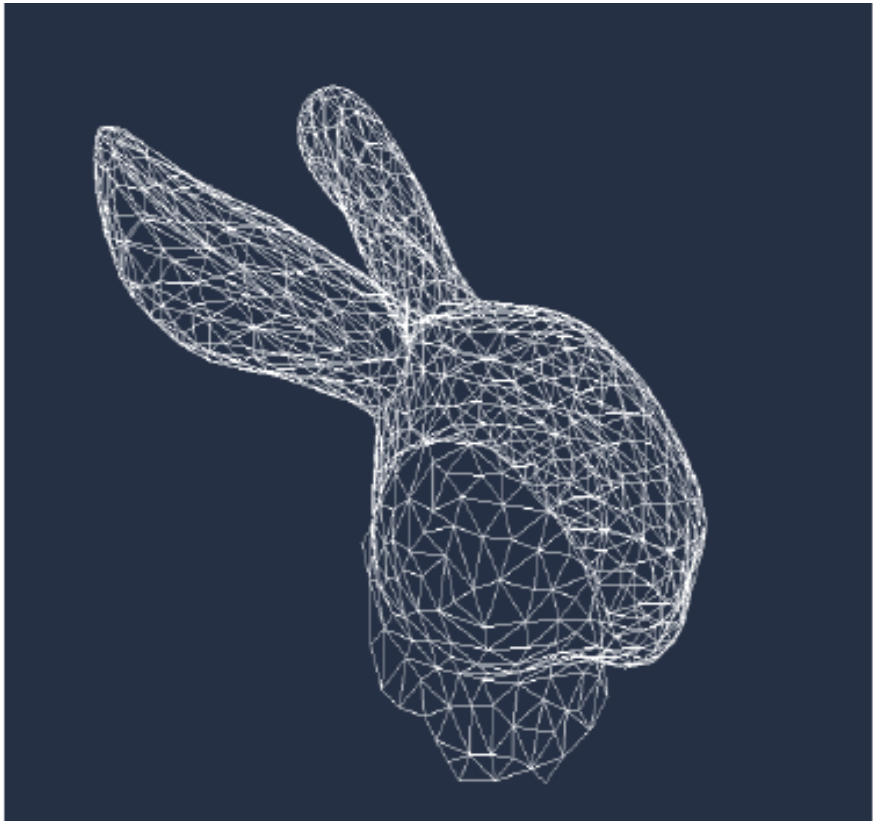


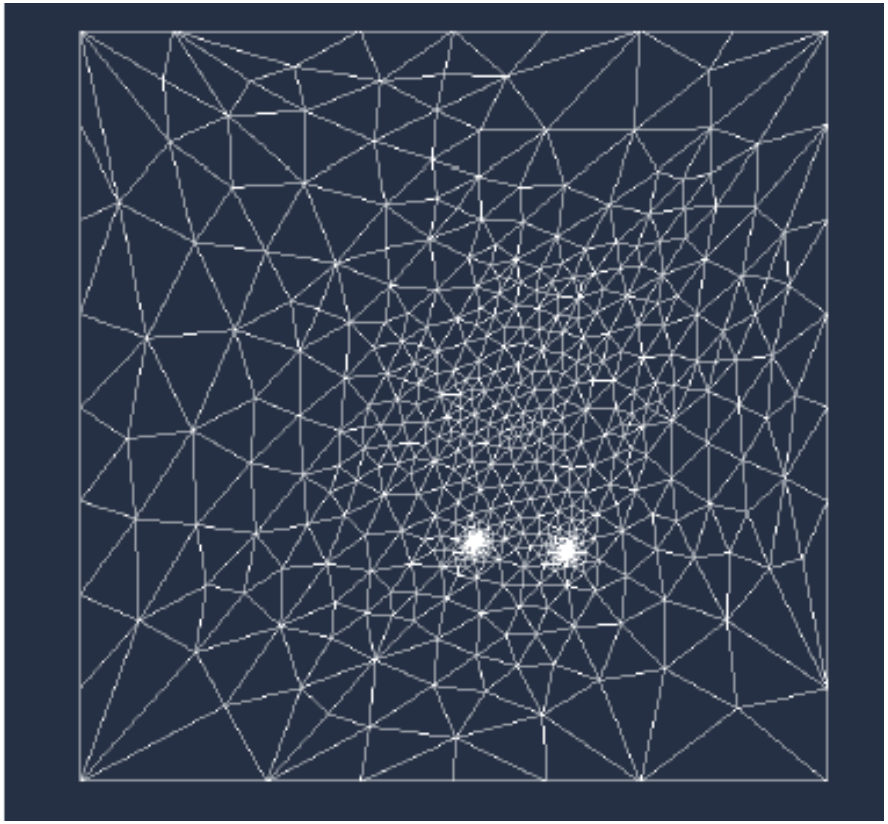




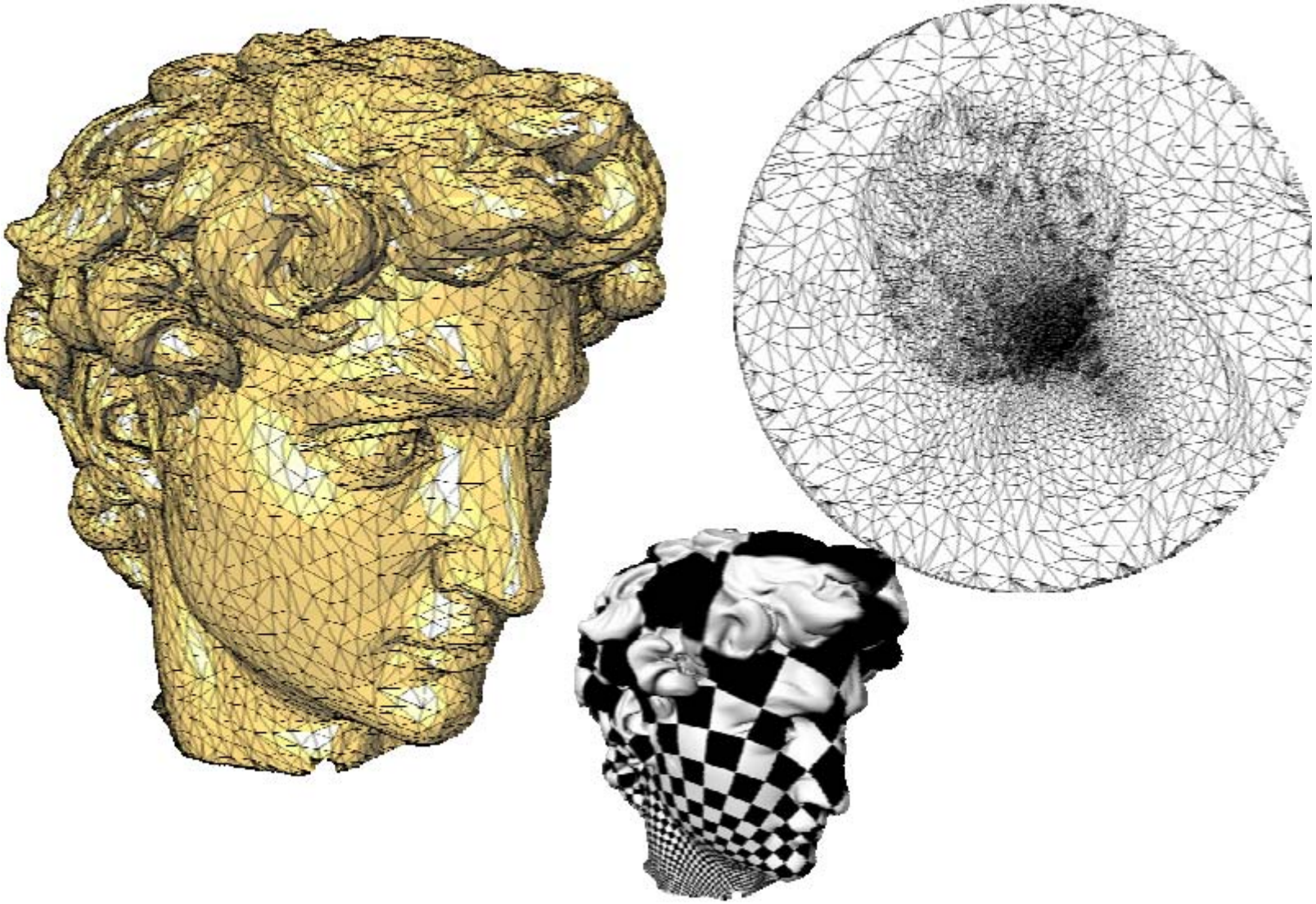


例子

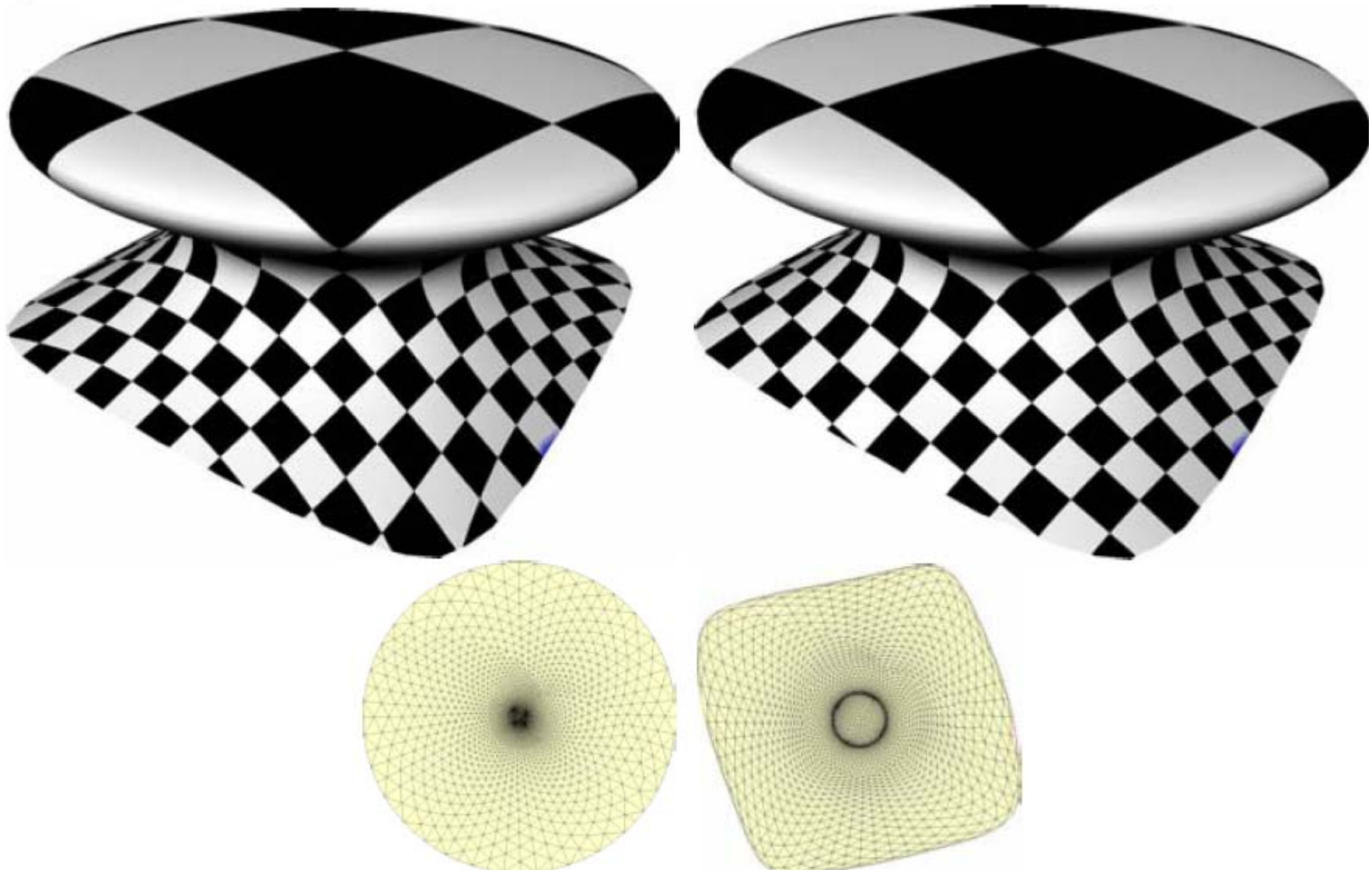




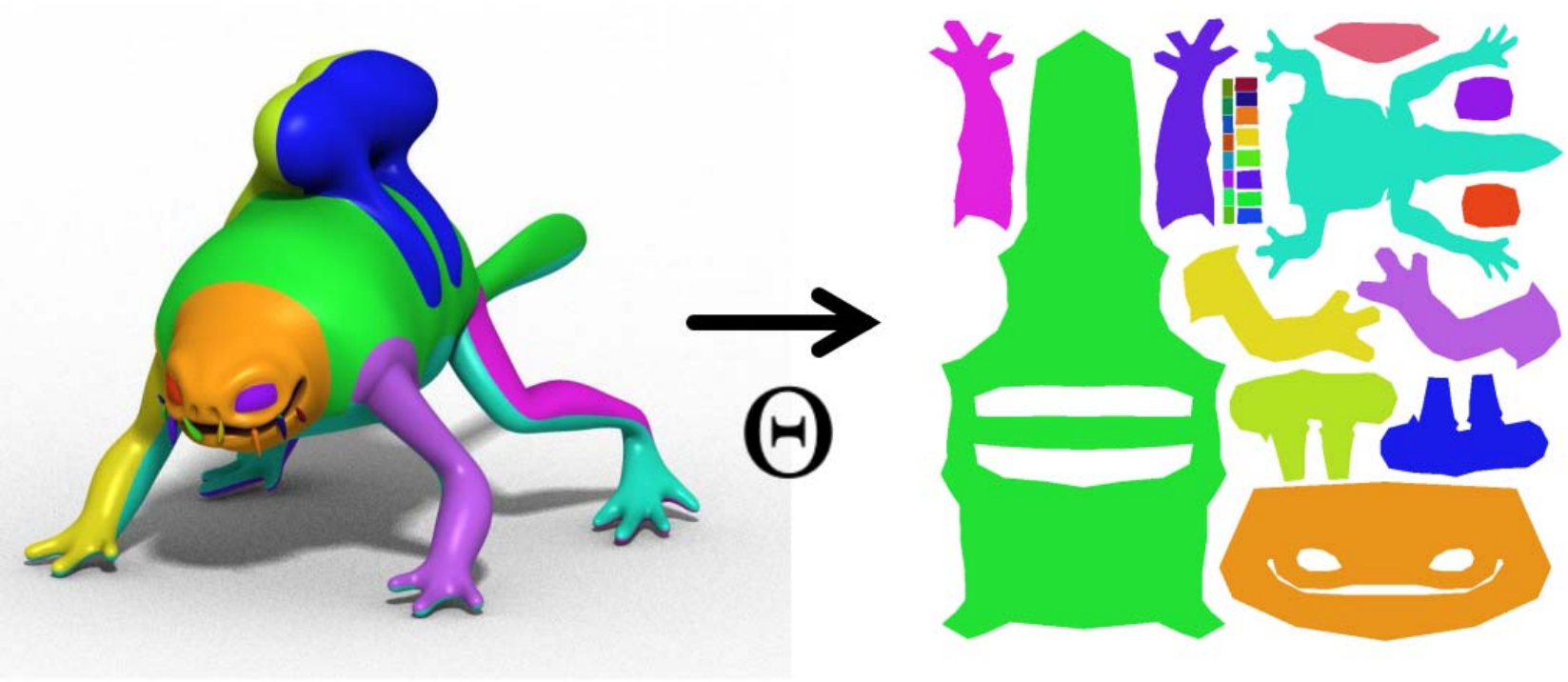
例子



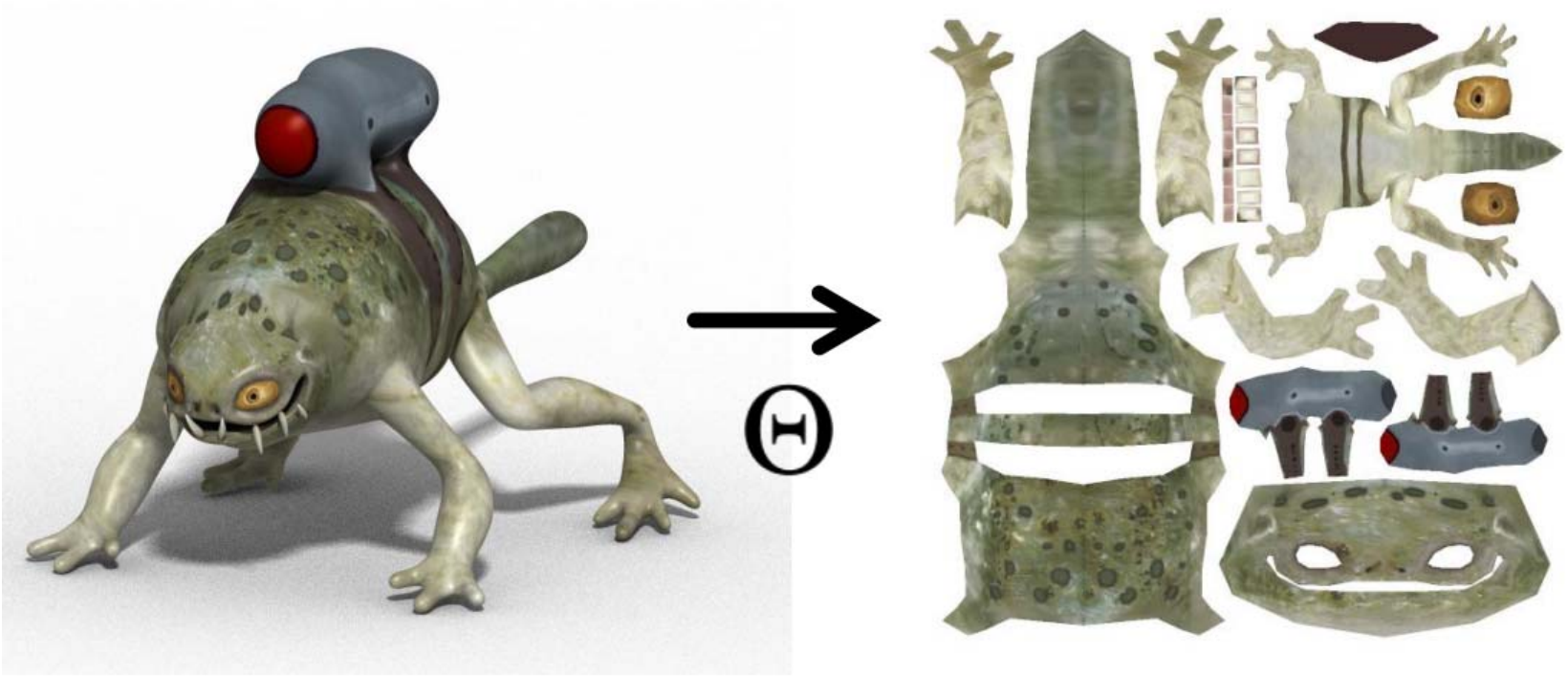
例子



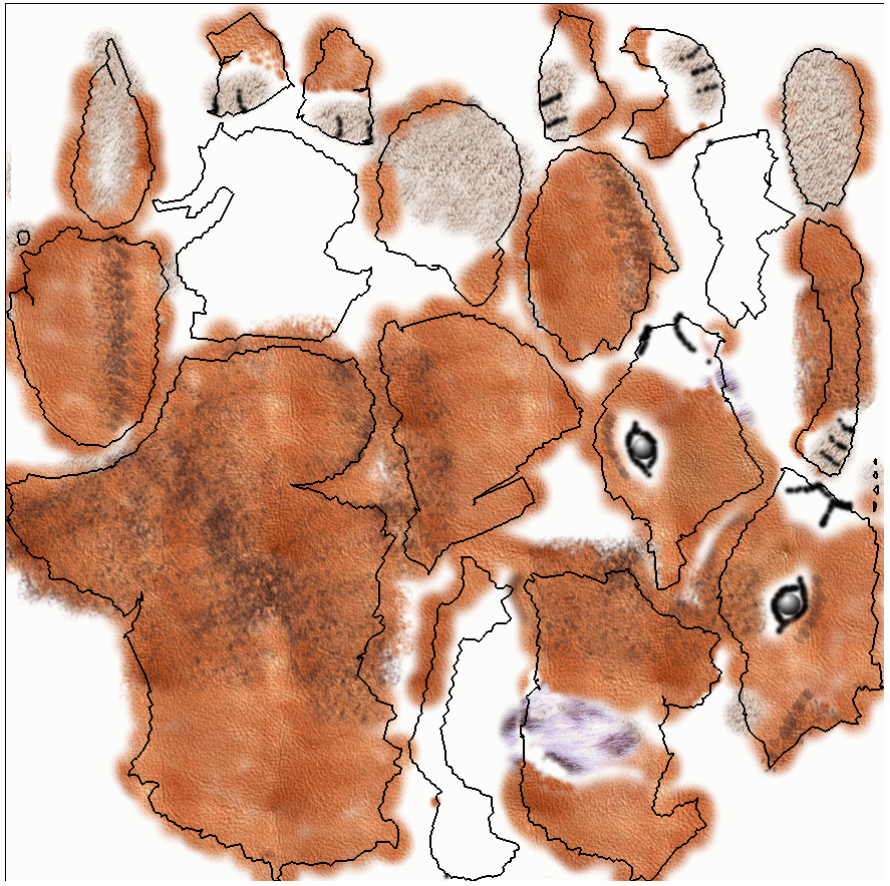
Parameterization



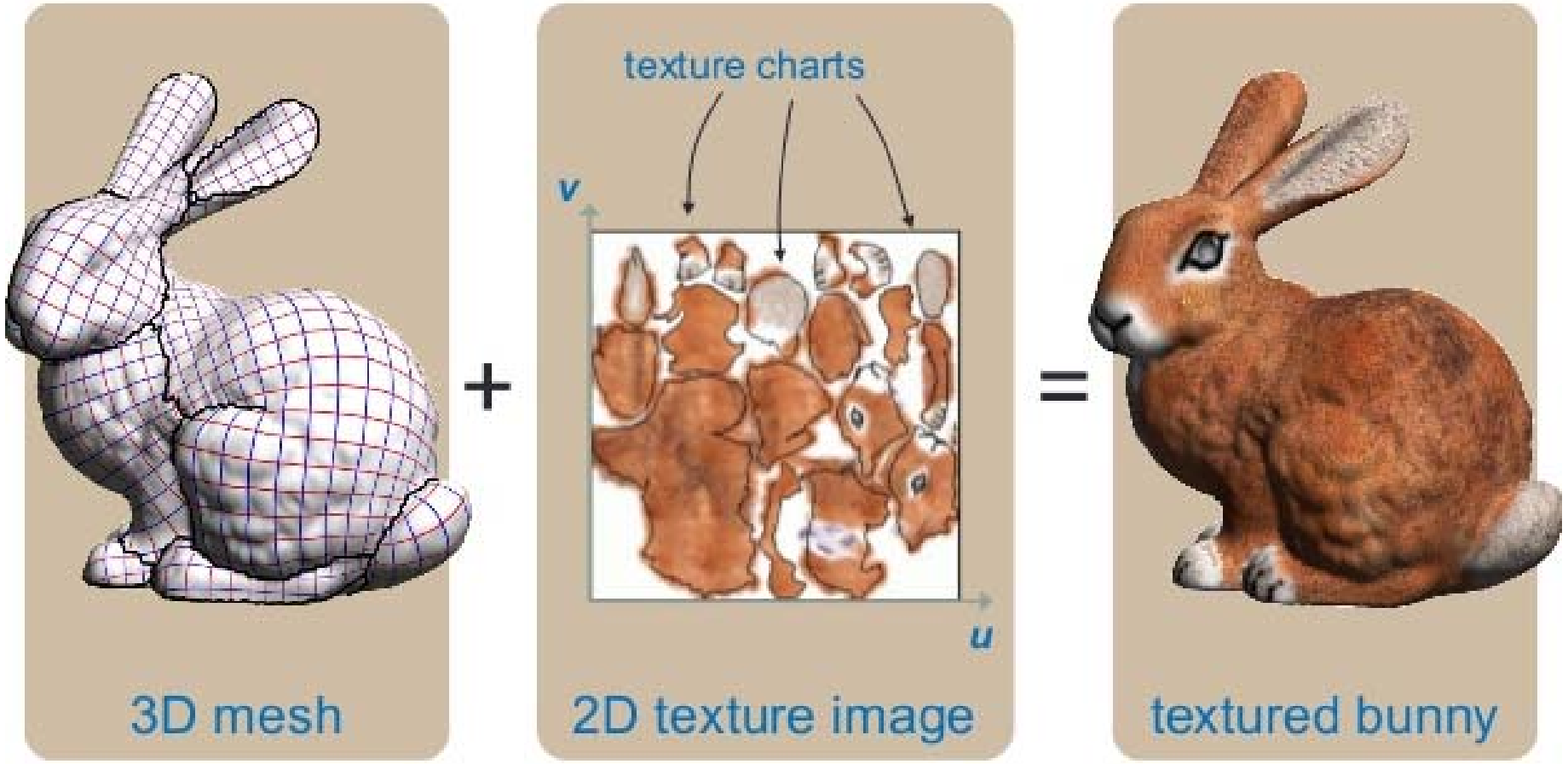
Parameterization



曲面纹理的保存：2D图像



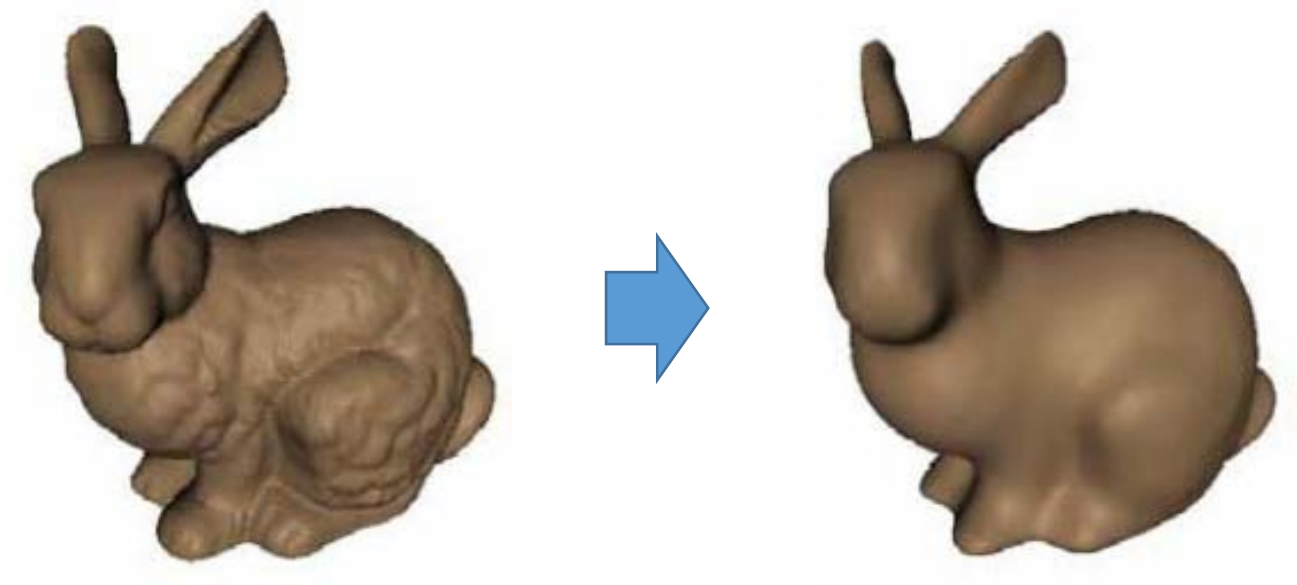
参数化应用：纹理映射



Constrained (Feature-Preserving) Global Laplacian Smoothing

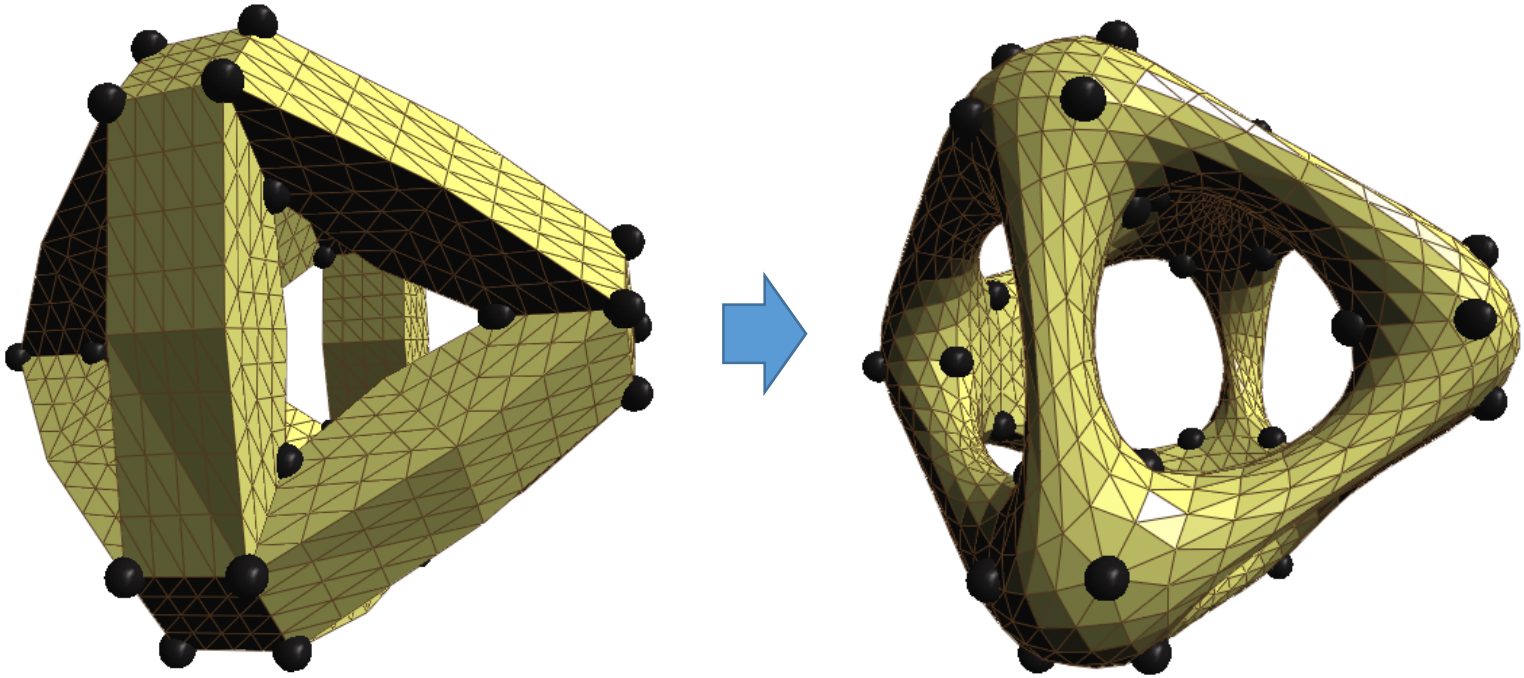
Shrinkage

- Global Laplacian smoothing results in shrinkages



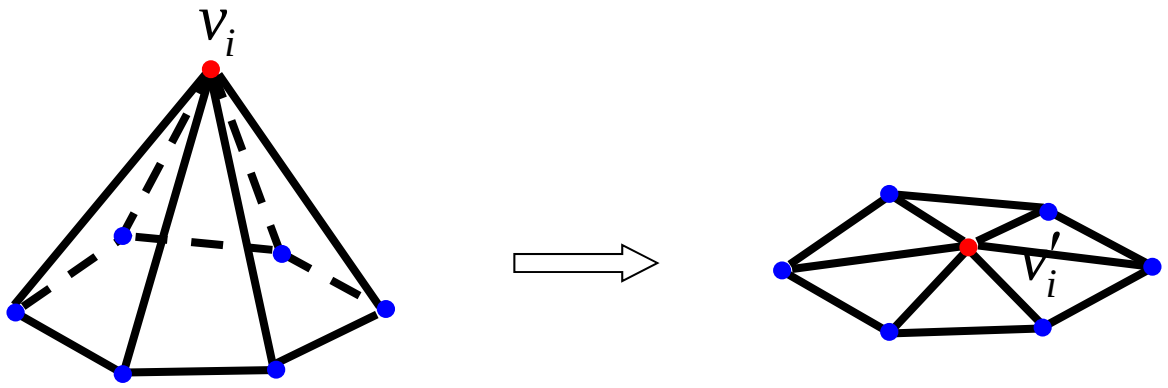
Ideas: add constraints

- 如何处理各种约束条件?
  - 顶点约束、面约束...



Soft Lapacian Smoothness

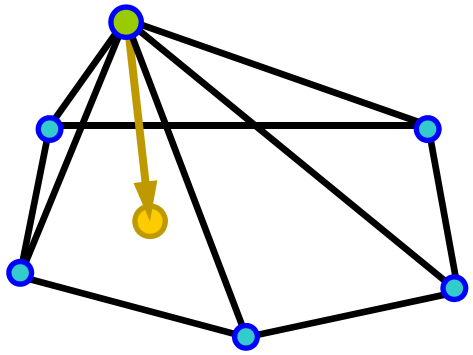
- Interpolation (hard constraints) → Approximation (soft constraints)



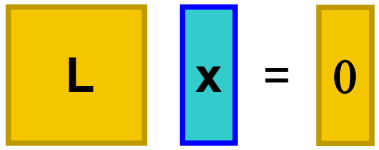
$$L(v_i) = v_i - \sum_{j \in N(i)} \omega_{ij} v_j = 0$$

Laplacian of Mesh

- Discrete Laplacians



$$L(v_i) = v_i - \sum_{j \in N(i)} \omega_{ij} v_j = 0$$

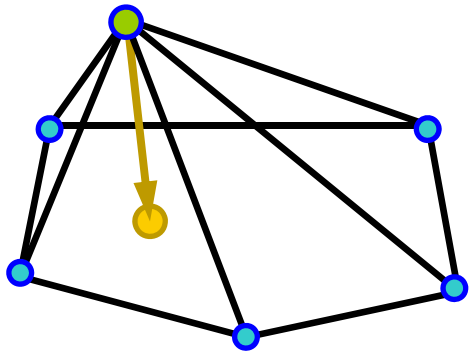


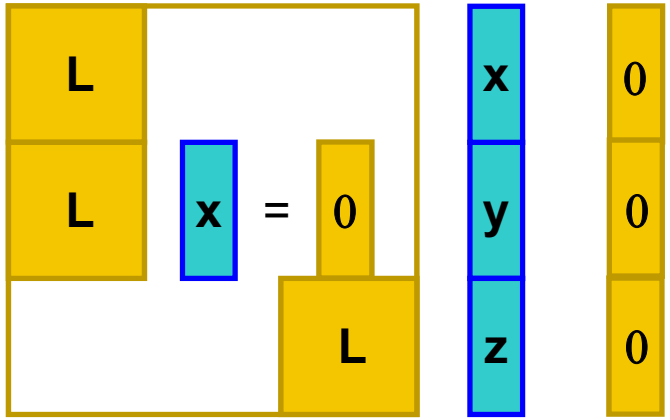
$$L_{ij} = \begin{cases} 1, & i = j, \\ -\omega_{ij}, & (i,j) \in E, \\ 0, & \text{other}. \end{cases}$$

- Laplacian of the mesh

Laplacian of Mesh

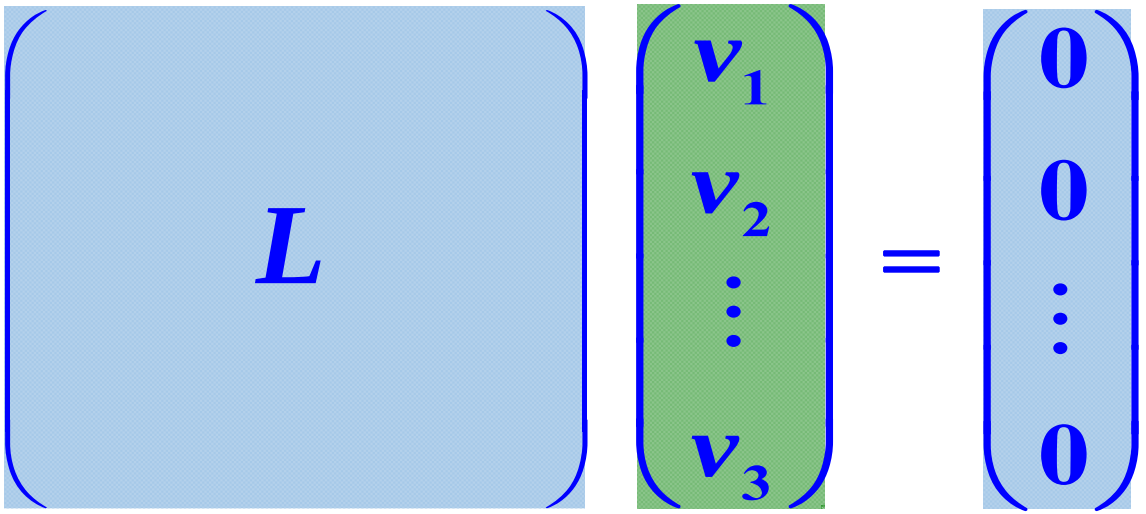
- Surface reconstruction





$$L(v_i) = v_i - \sum_{j \in N(i)} \omega_{ij} v_j = 0$$

Properties of Laplacian

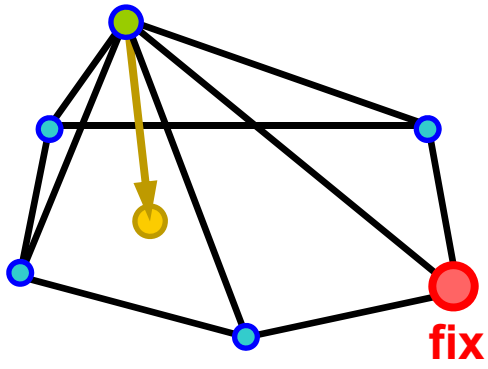


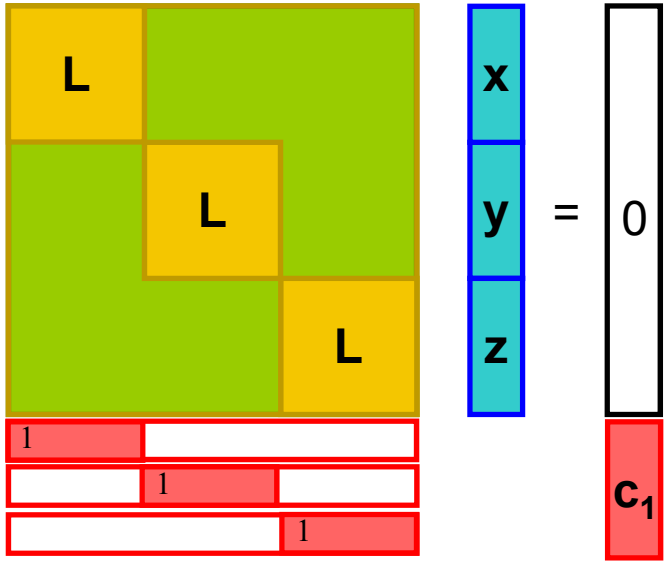
Rank(L) = n-k

- k is the number of connected components of the mesh
- Need to add some constraints

Vertex Constraints

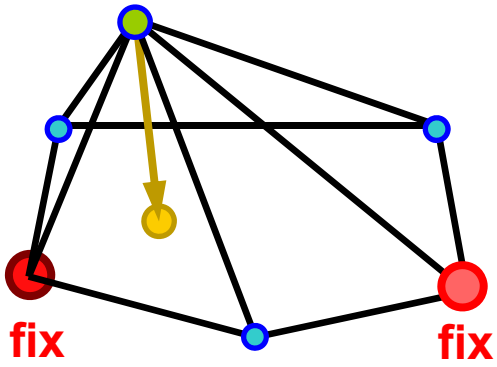
- Add position constraint for one vertex

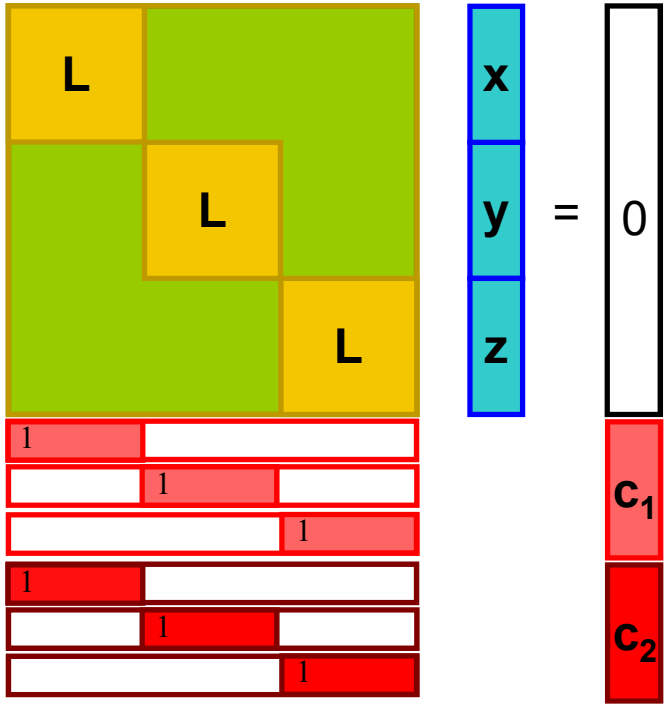




Vertex Constraints

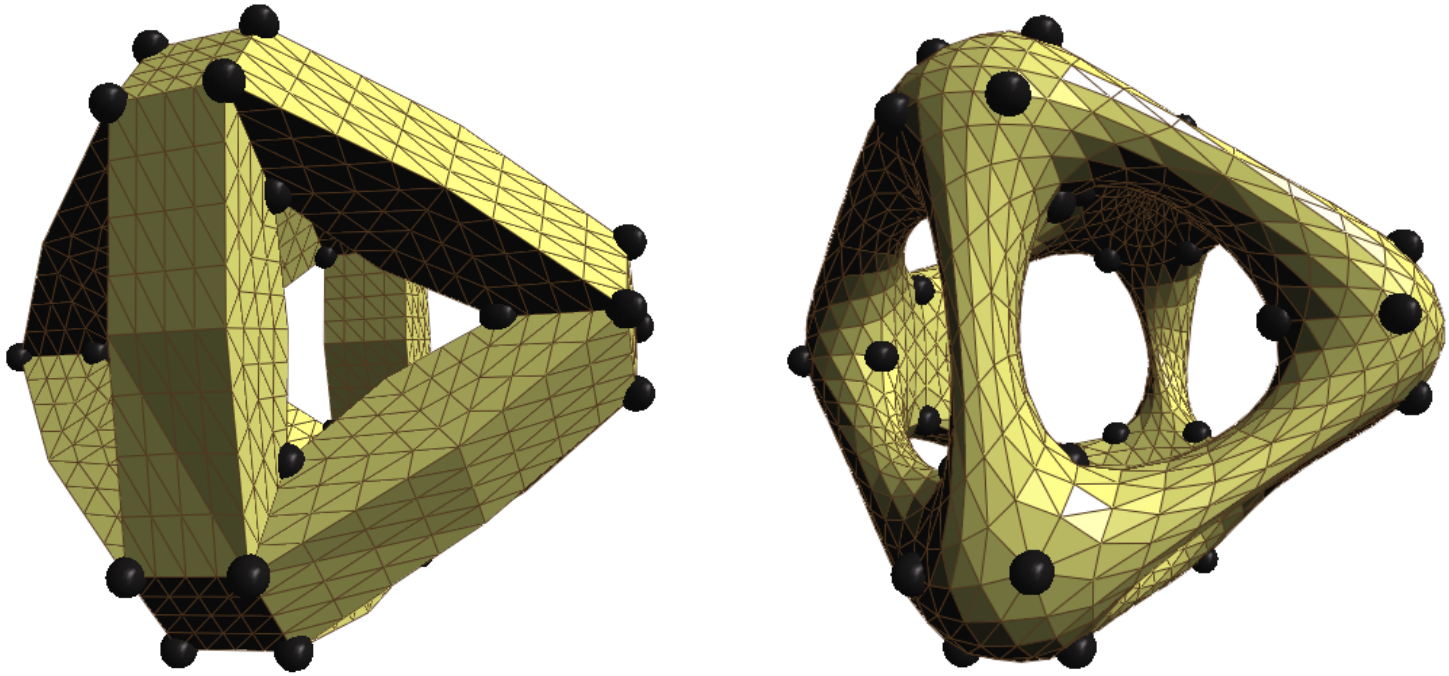
- Add position constraints for more vertices





Adding Vertex Constraints

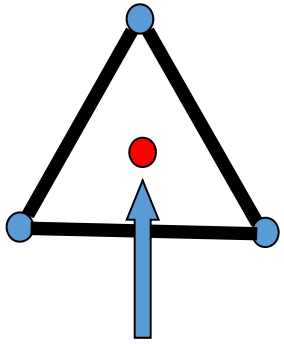
$$\min_{X'} \{ \| LX' \|^2 + \mu^2 \sum_{i \in C} \left| v_i' - v_i \right|^2 \}$$



Face Constraints

$$\begin{pmatrix} L \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ c_1 \\ c_2 \\ t_1 \\ t_2 \end{pmatrix}$$

$A \qquad x \qquad = \qquad b$

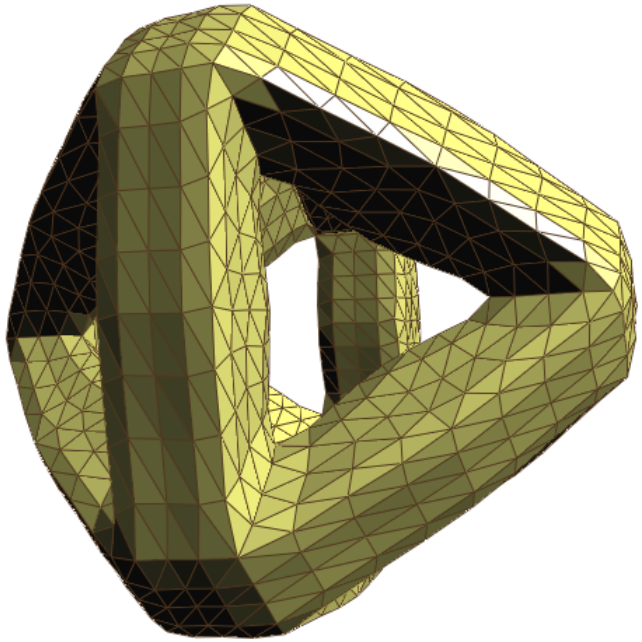


barycenter

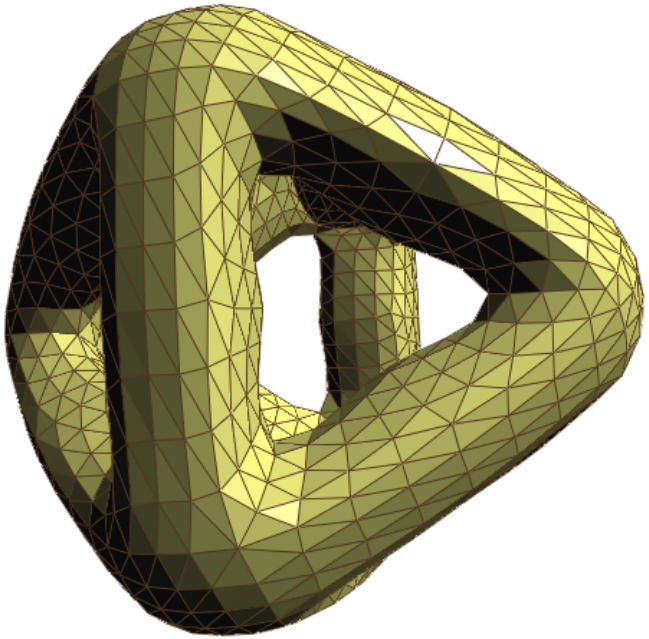
$$v_{center} = \frac{1}{3}(v_i + v_j + v_k)$$

Adding Face Constraints

$$\min_{X'}\{\|LX'\|^2 + \sum_{<i,j,k>\in T} \lambda^2 \left|(v_i' + v_j' + v_k') - (v_i + v_j + v_k)\right|^2\}$$



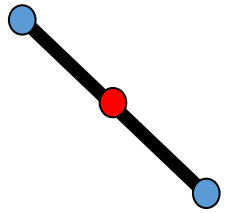
λ=0.5



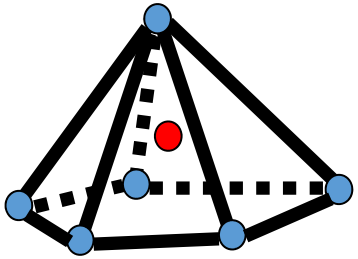
λ=0.3

Other Constraints

- Edge constraints



- 1-ring barycenter constraints



- Other linear constraints

Minimizing Energy

$$\min_{X'}\left\{\left\|LX'\right\|^2+\sum_{i\in C}\mu^2\left|v_i'-v_i\right|^2+\sum_{<i,j,k>\in T}\lambda^2\left|(v_i'+v_j'+v_k')-(v_i+v_j+v_k)\right|^2\right\}$$

$$A\mathbf{x}=\mathbf{b}$$

Least Square Solution

- An over-determined system:

$$\mathbf{A}\,\mathbf{x} = \mathbf{b}$$

- Normal equation:

$$\mathbf{A}^T\mathbf{A}\,\mathbf{x} = \mathbf{A}^T\mathbf{b}$$

$$\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\,\mathbf{A}^T\mathbf{b}$$

One Channel Solution

- Very efficient solution by Cholesky factorization of A^TA:

$$\boldsymbol{A^T A = R^T R}$$

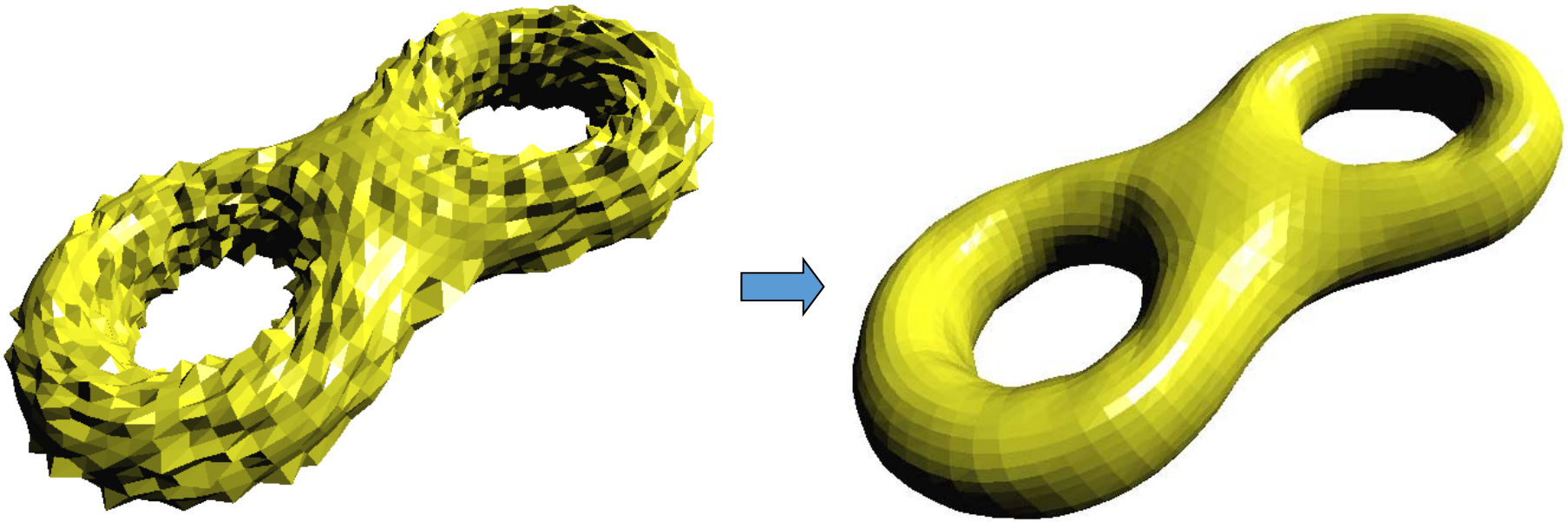
R is upper-triangular and sparse

Once R is computed, solving for **x**, **y**, **z** by back-substitution:

$$\boldsymbol{R^T \xi = A^T b}$$

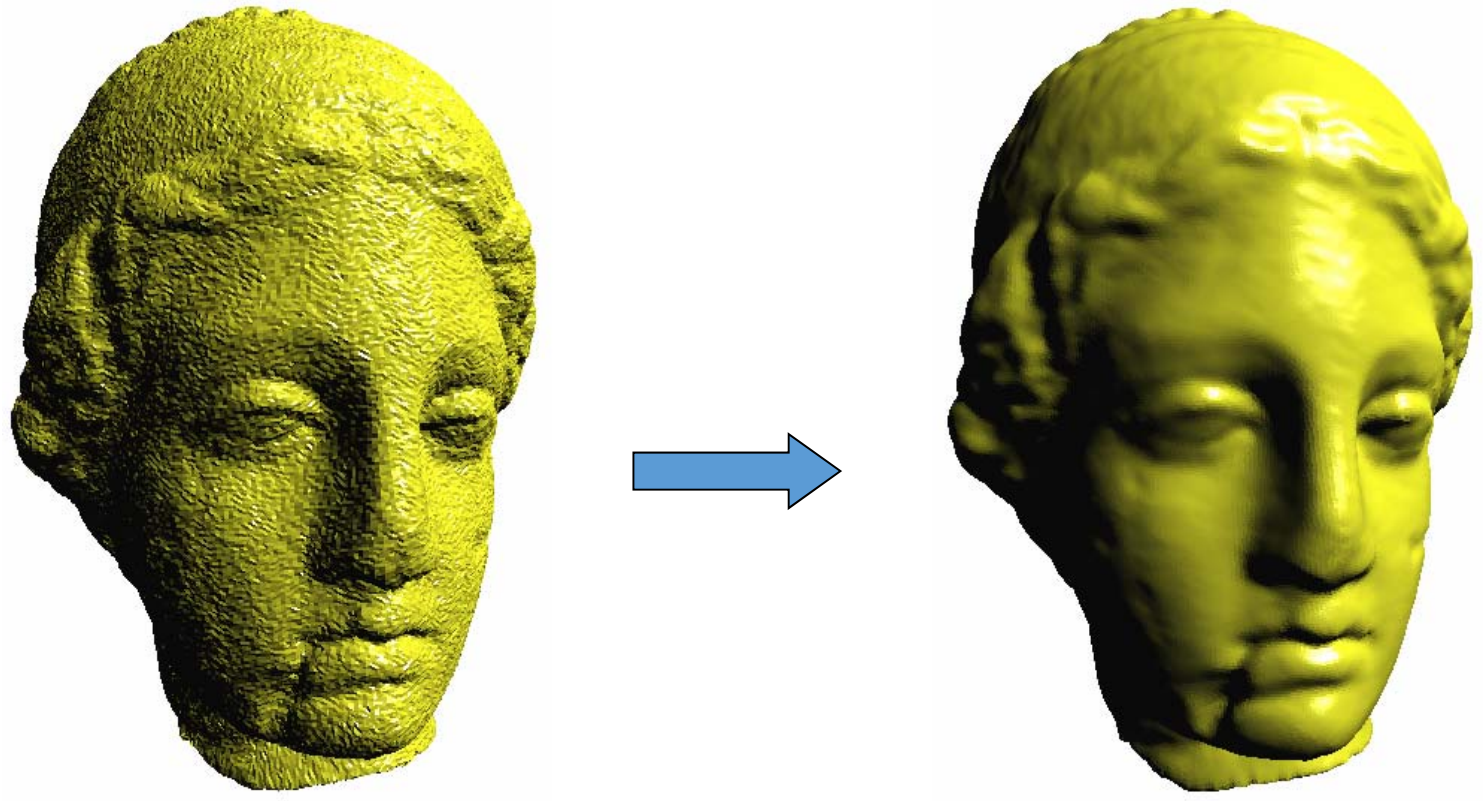
$$\boldsymbol{R x = \xi}$$

Results



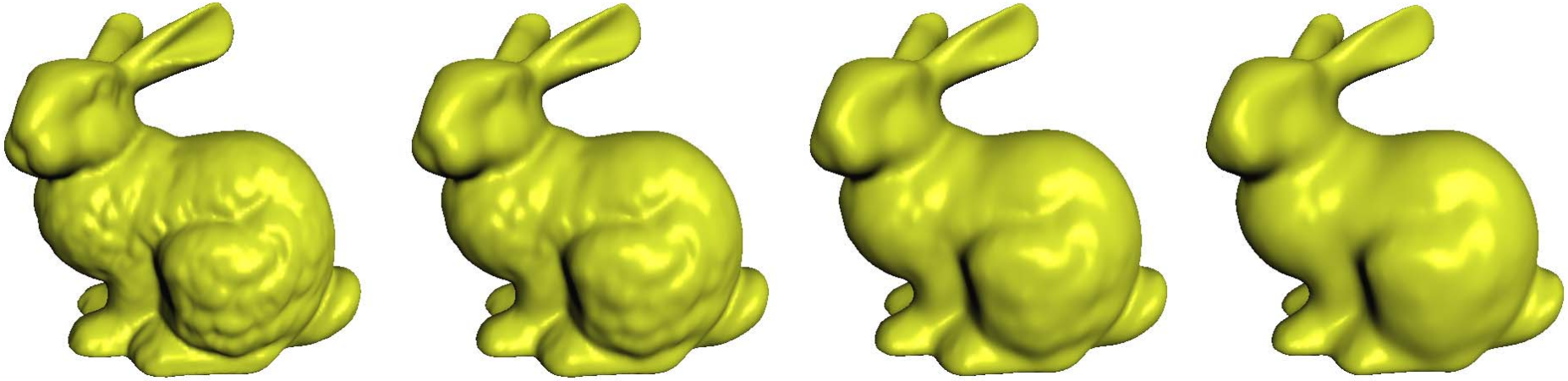
‘8’-like mesh model
3070 vertices, 6144 triangles

Results



Venus head model
134359 vertices, 268714 triangles

LoD Smoothing



Applying our algorithm to the bunny model with different parameters.

Laplacian Editing

[Sorkine et al. SGP 2004]

Cardinal Coordinates

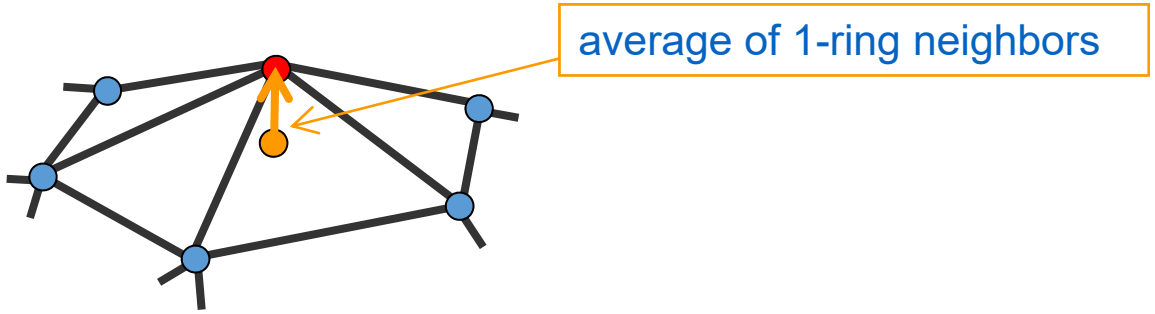
- (x,y,z) coordinates

```
v 1.5 -0.960751 -1.2232
v 0.81 -0.891238 -3.74258
v 0.16 -0.233535 -2.28405
v 1.49 -2.44325 -3.6962
v 1.59 -2.98815 -4.15761
v 1.66 -2.81016 -4.10777
v 1.41 -1.14861 -1.92823
v 1 -1.40023 -3.80159
v 0.88 -1.33122 -3.83517
v 1.69 -2.60816 -4.12133
v 1.68 -2.36516 -4.13078
…
```

Pros and Cons?

What's are Details?

- Detail = surface – smooth (surface)
- Smoothing = averaging



What's the Difference?

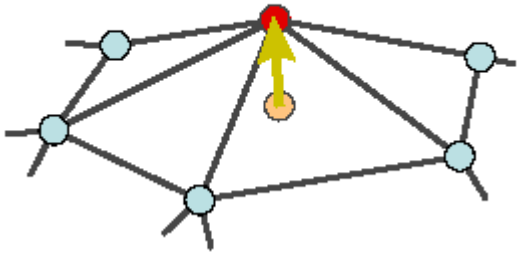
- Absolute Coordinate

$$v_i = (x_i, y_i, z_i)$$

?

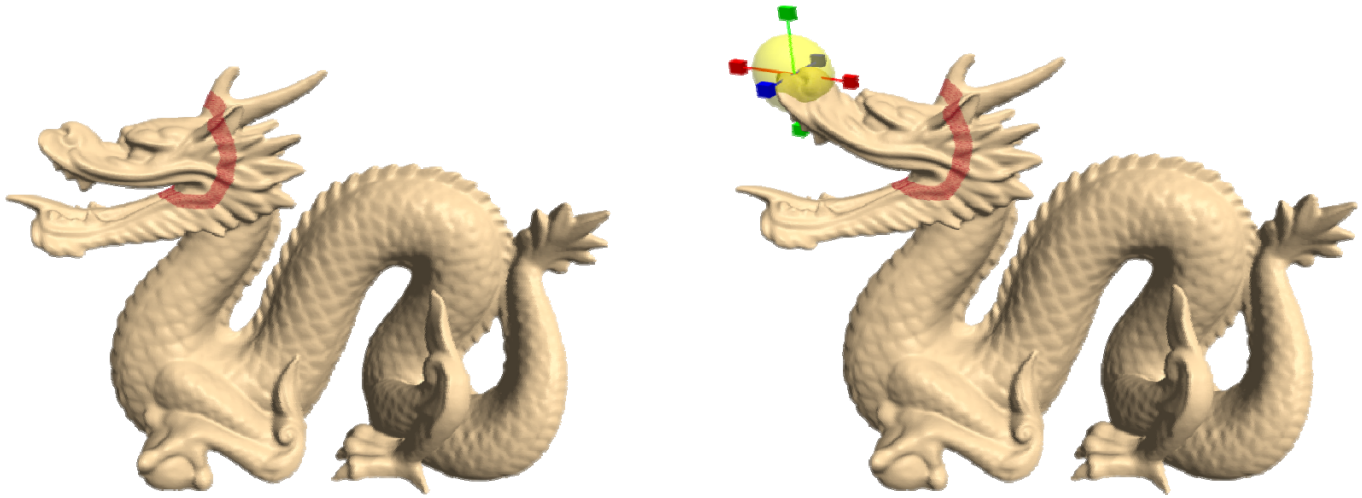
- Relative Coordinate

$$v_i = \sum_{j \in N(i)} w_j v_j + \delta_i$$



Laplacian Editing

- Local detail representation – enables detail preservation through various modeling tasks
- Representation with sparse matrices
- Efficient linear surface reconstruction



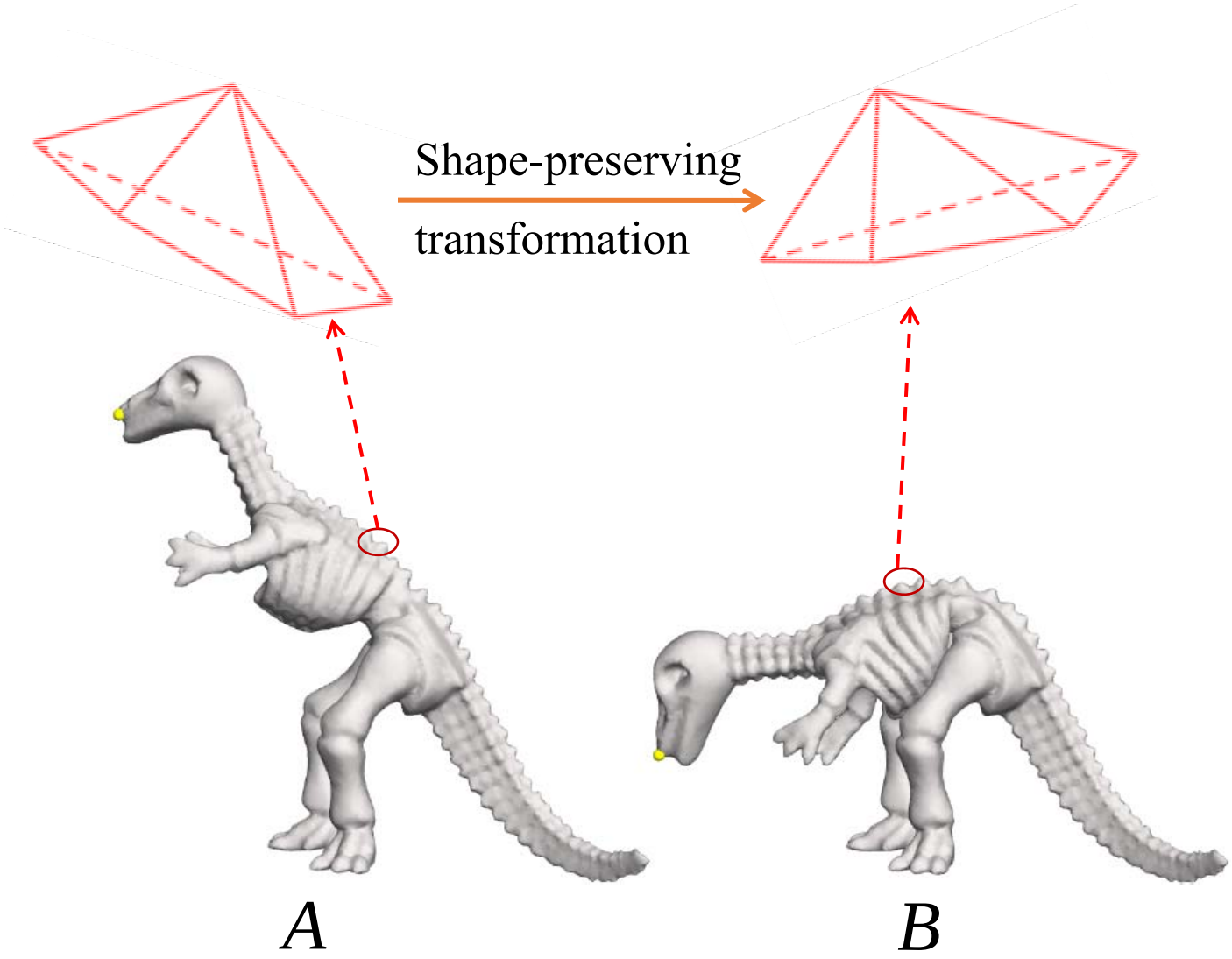
Editing framework

- The spatial constraints will serve as modeling constraints
- Reconstruct the surface every time the modeling constraints are changed

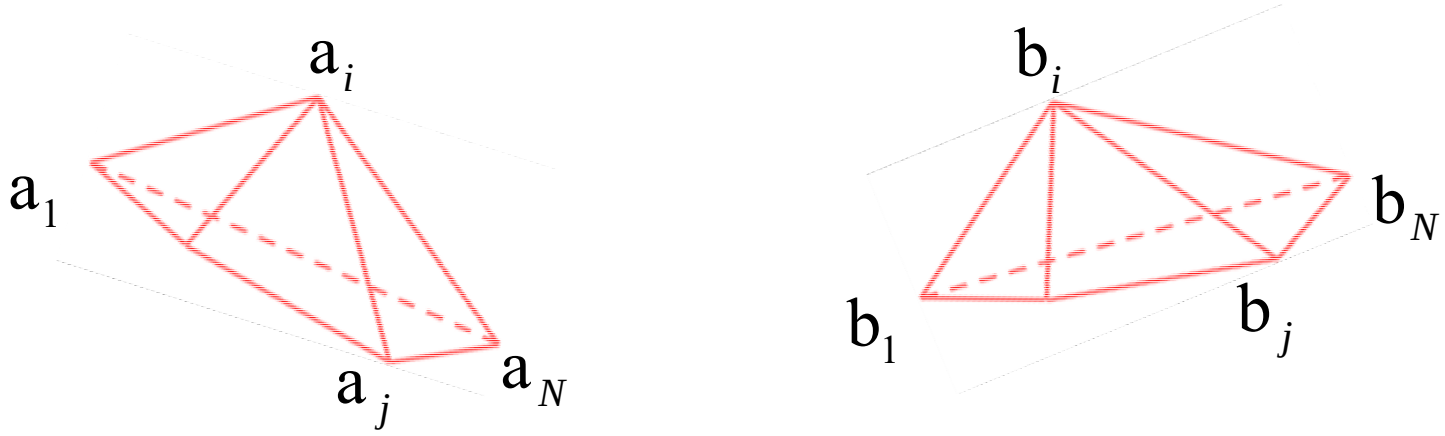
Detail constraints: $L\mathbf{x} = \boldsymbol{\delta}$

Modeling constraints: $x_j = c_j, \quad j \in \{j_1, j_2, \ldots j_k\}$

Direct Detail Preserving



Rotation Transformation

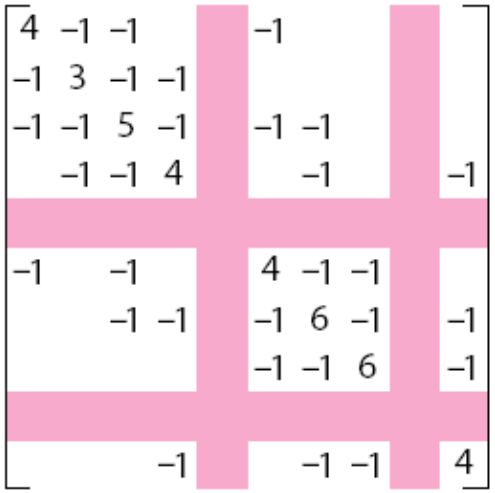


$$\begin{pmatrix} b_1 - b_i \\ \vdots \\ b_N - b_i \end{pmatrix} = \begin{pmatrix} a_1 - a_i \\ \vdots \\ a_N - a_i \end{pmatrix} \mathbf{R}_i$$

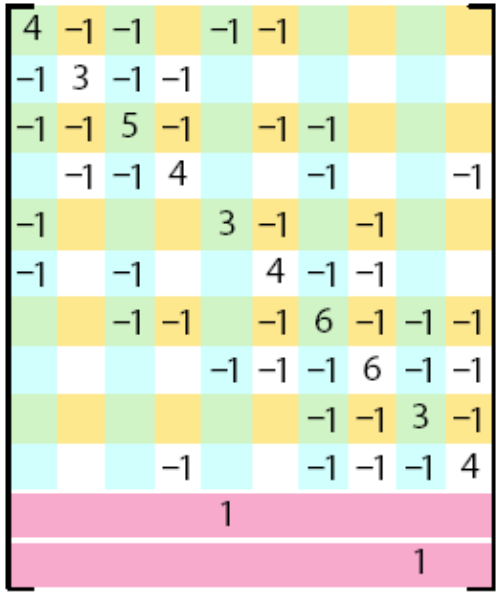
Reconstruction

- Soft constraints

$$L^T L v = L^T \delta$$



Invertible Laplacian



2-anchor $\tilde{L}$

Variational Viewpoint

- Laplacian Approximation

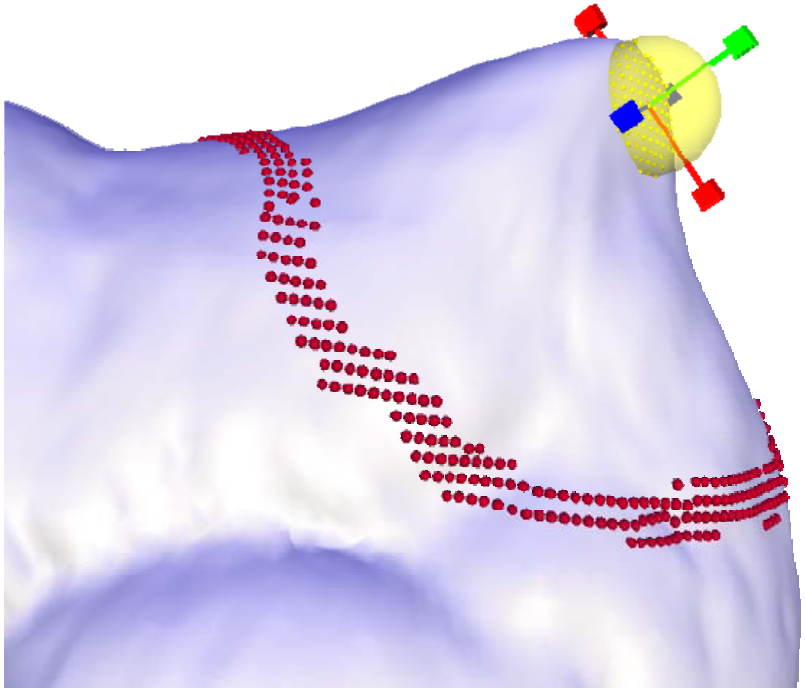
$$\tilde{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \left(\|L\mathbf{x} - \delta^{(x)}\|^2 + \sum_{j \in C} \omega^2 \|x_j - c_j\|^2 \right).$$

- Gradient Approximation

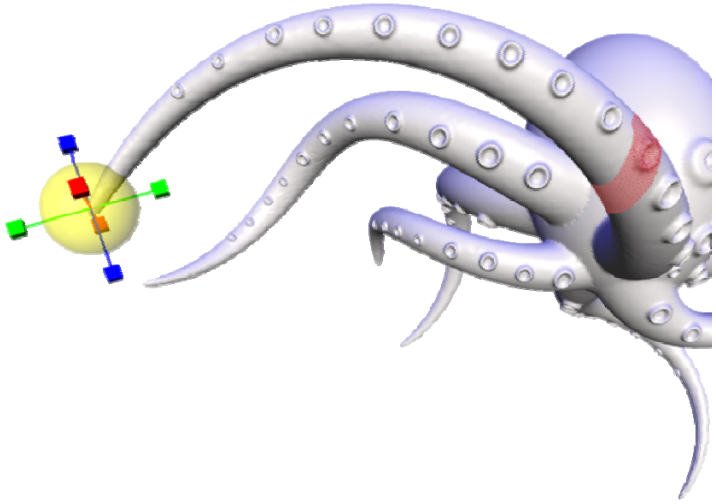
$$\min_{\phi} \int \int_{\Omega} \|\nabla \phi - \mathbf{w}\|^2 dA,$$

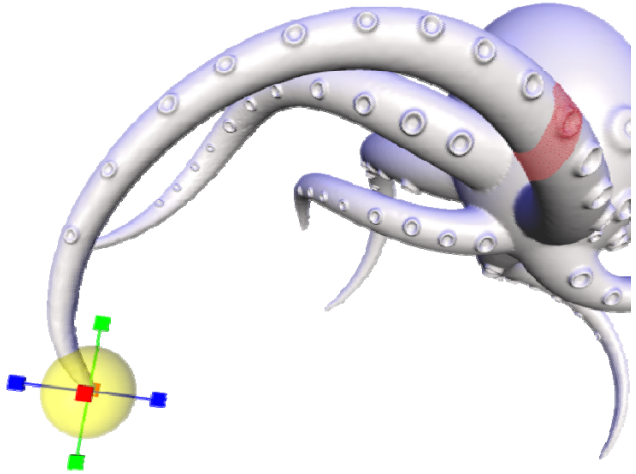
User Interfaces

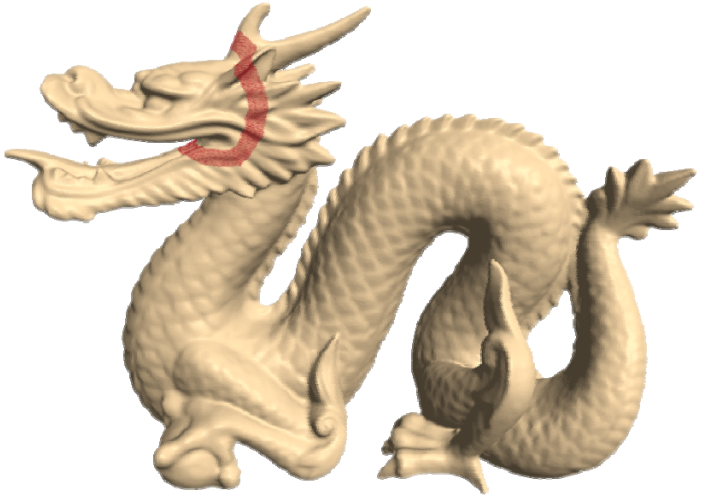
- ROI is bounded by a belt (static anchors)
- Manipulation through handle(s)

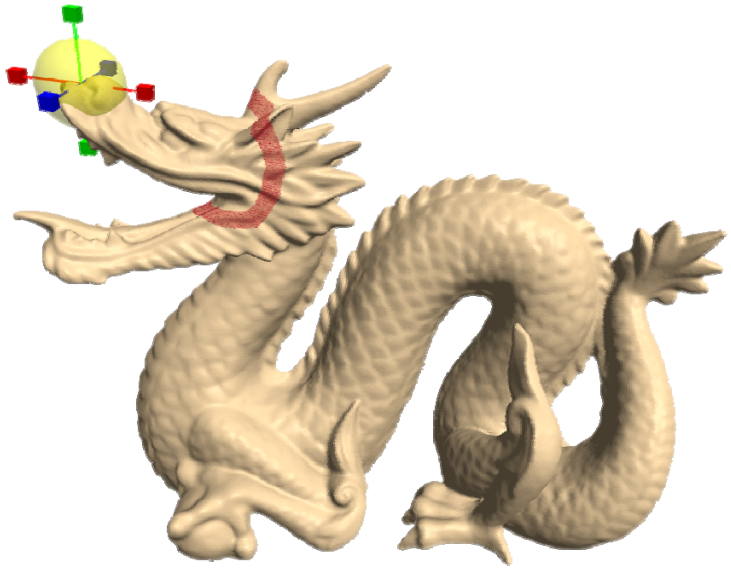


Results

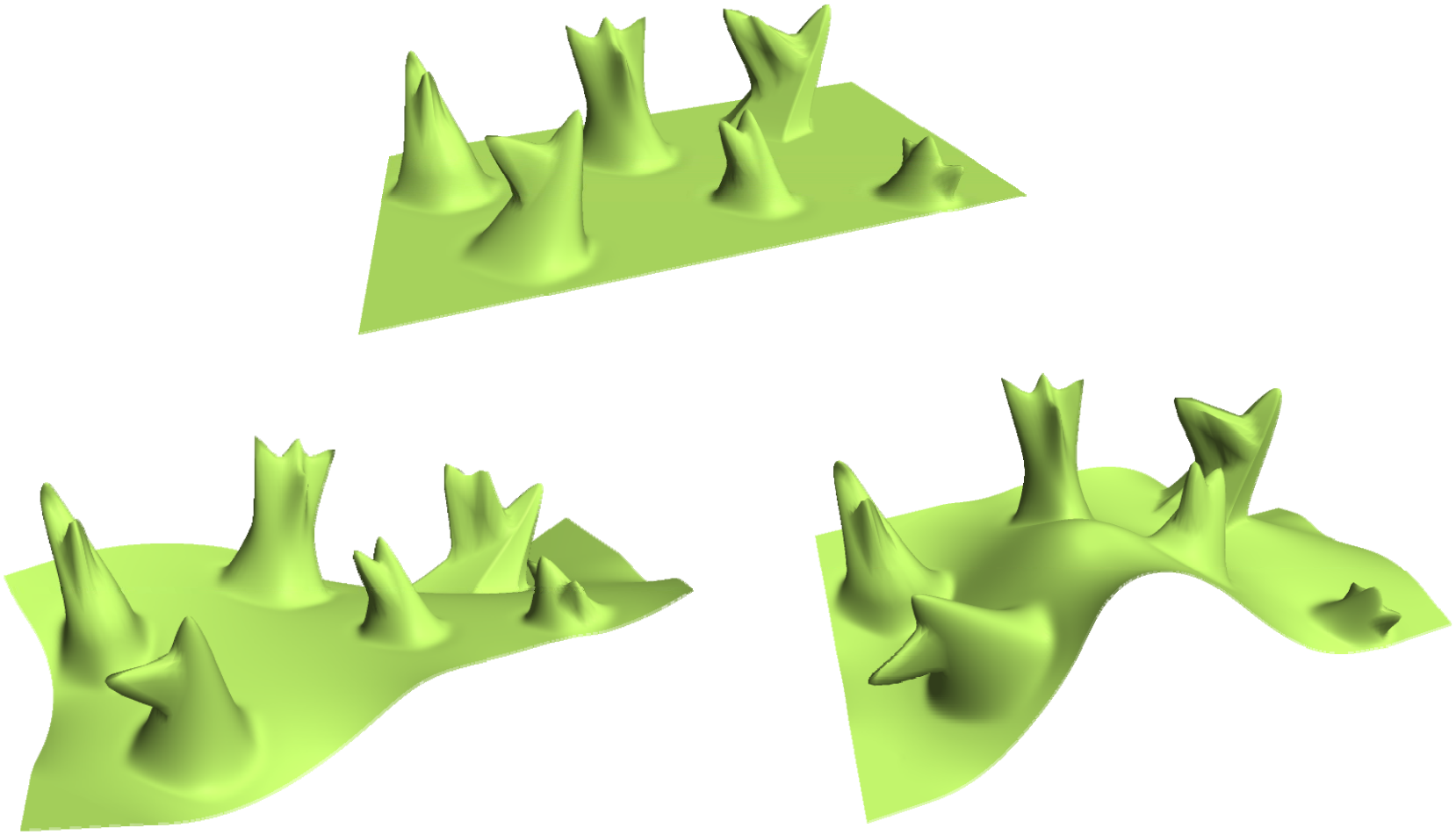






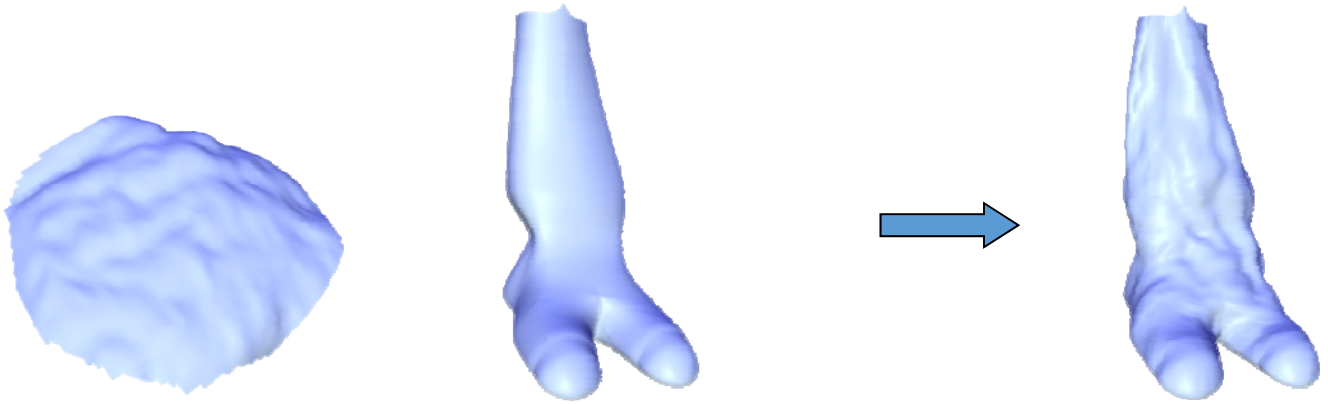


Results



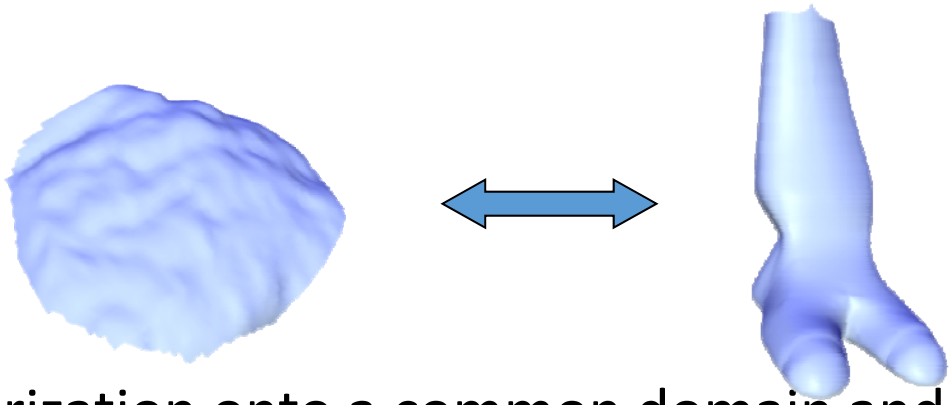
Detail transfer and mixing

- “Peel“ the coating of one surface and transfer to another



Detail transfer and mixing

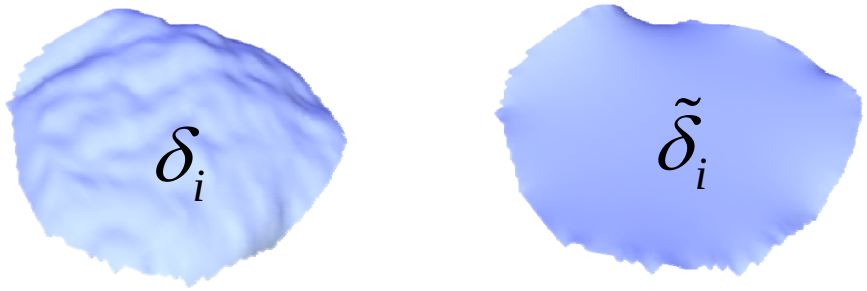
- Correspondence:



  - Parameterization onto a common domain and elastic warp to align the features, if needed

Detail transfer and mixing

- Detail peeling:

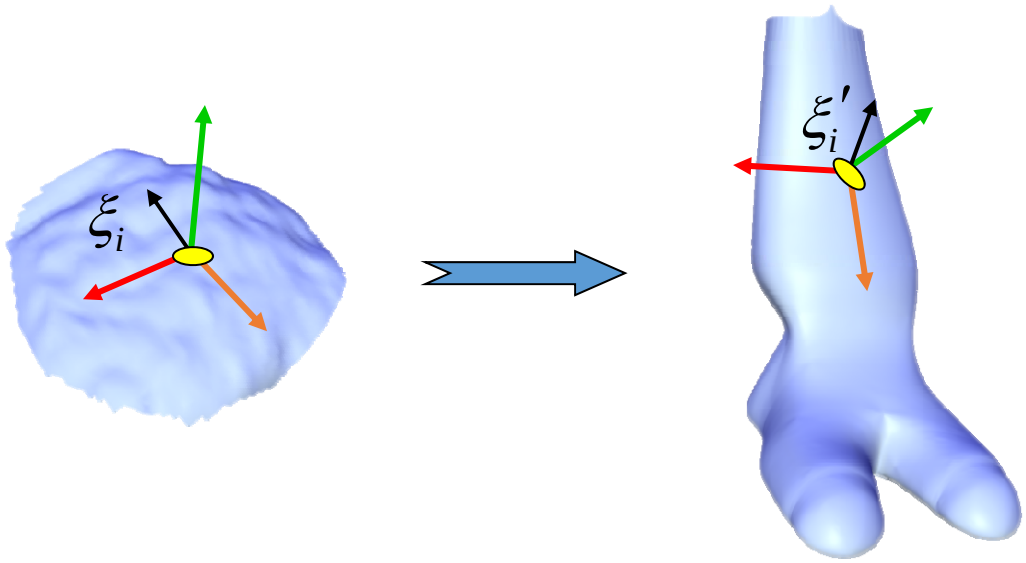


Smoothing by
[Desbrun et al.99]

$$\xi_i = \delta_i - \tilde{\delta}_i$$

Detail transfer and mixing

- Changing local frames:

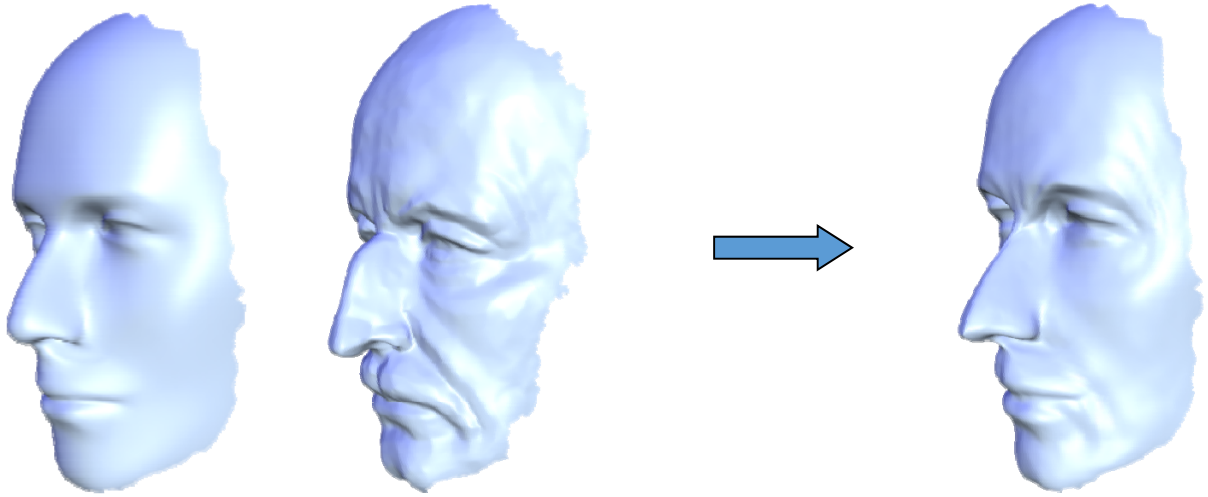


Detail transfer and mixing

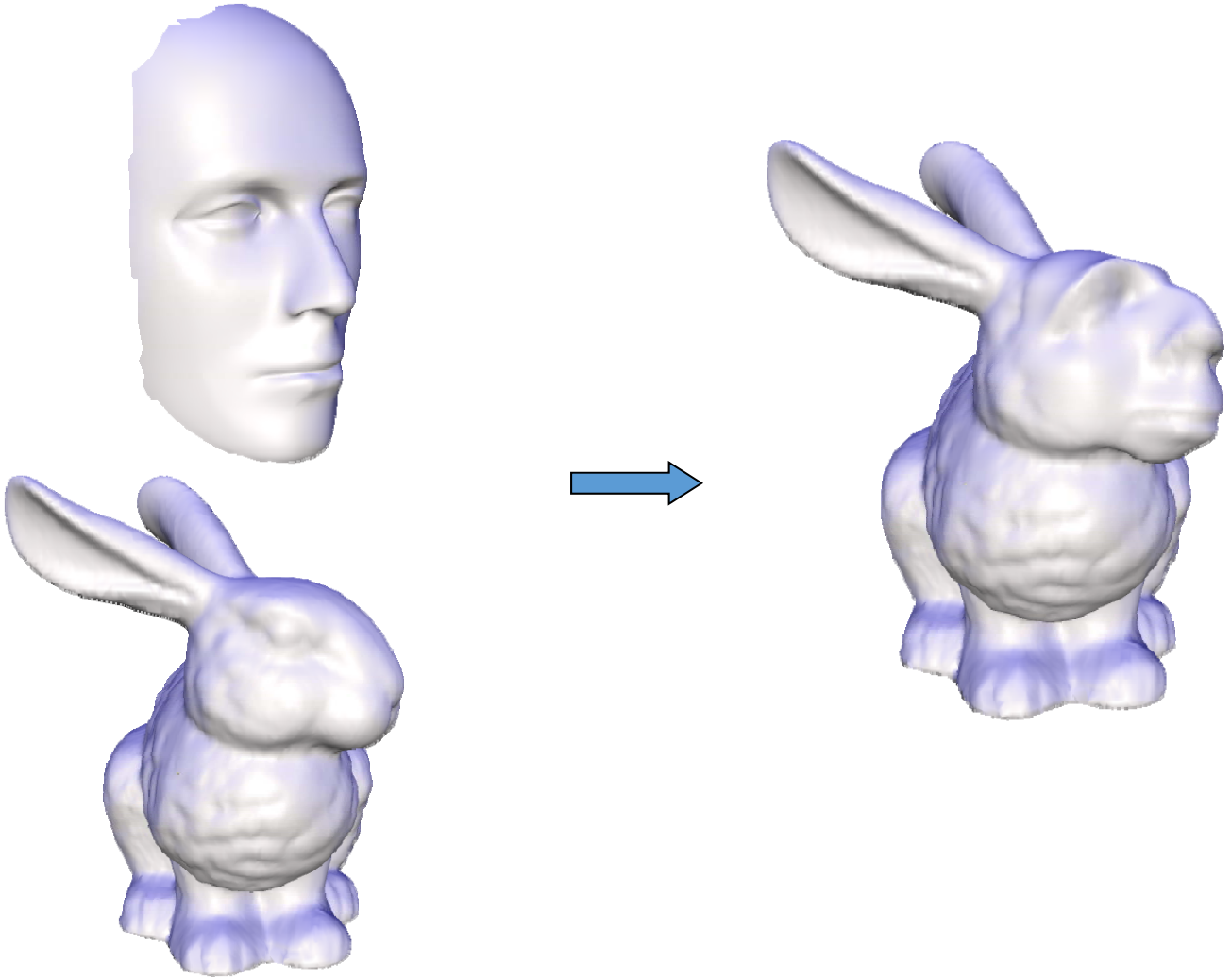
- Reconstruction of target surface from: δ_{target}

$$\delta_{\text{target}} = \delta_i' + \xi_i'$$

Examples

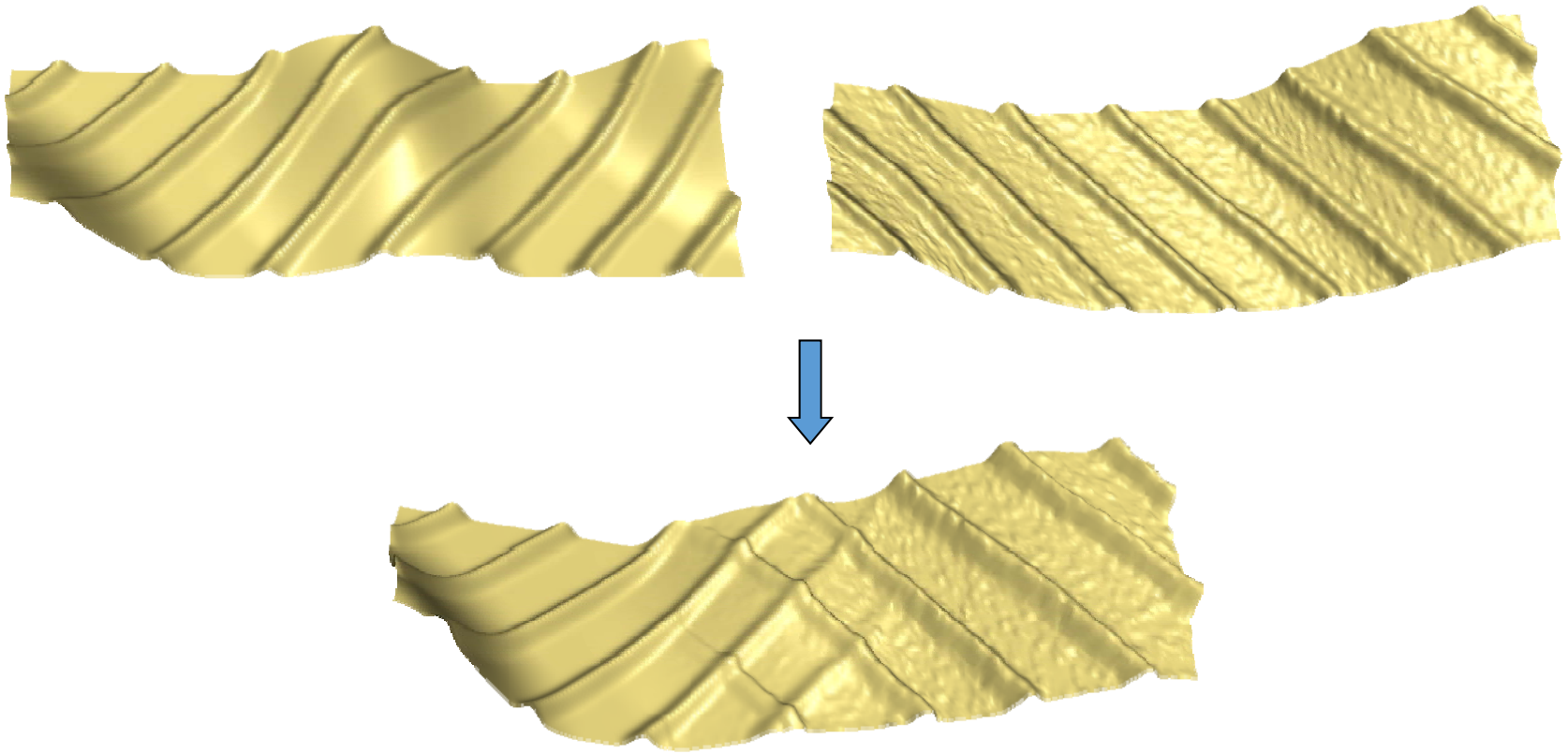


Examples



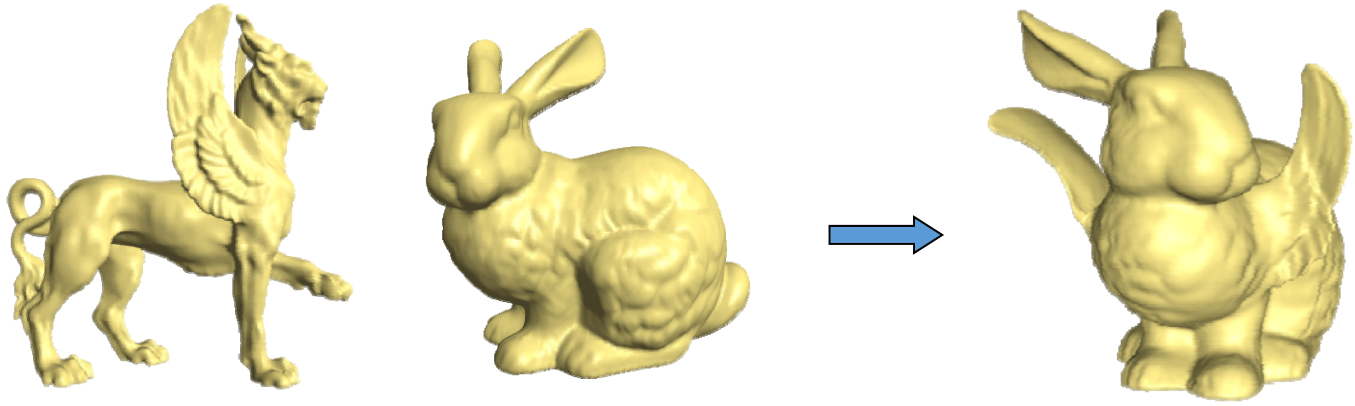
Mixing Laplacians

- Taking weighted average of δ_i and δ'_i



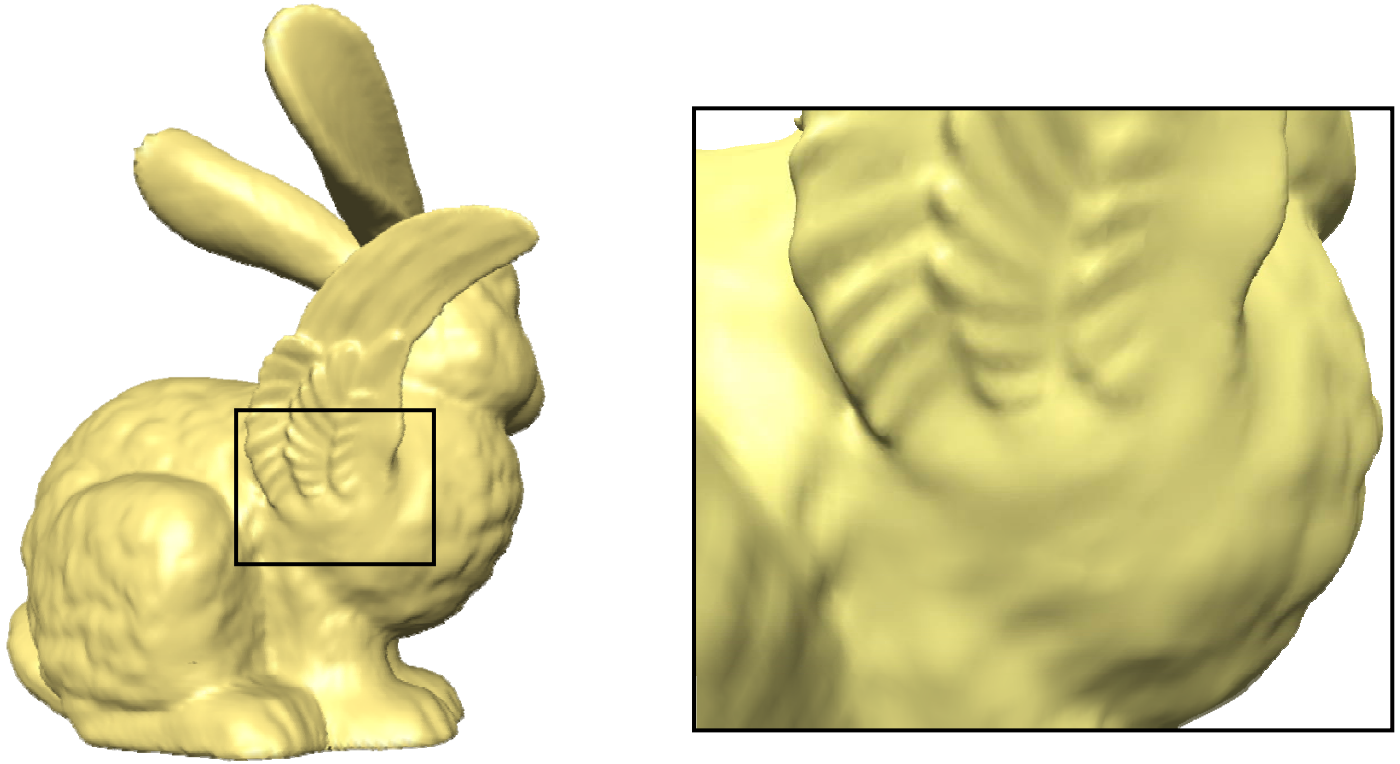
Mesh transplanting

- The user defines
  - Part to transplant
  - Where to transplant
  - Spatial orientation and scale
- Topological stitching
- Geometrical stitching via Laplacian mixing



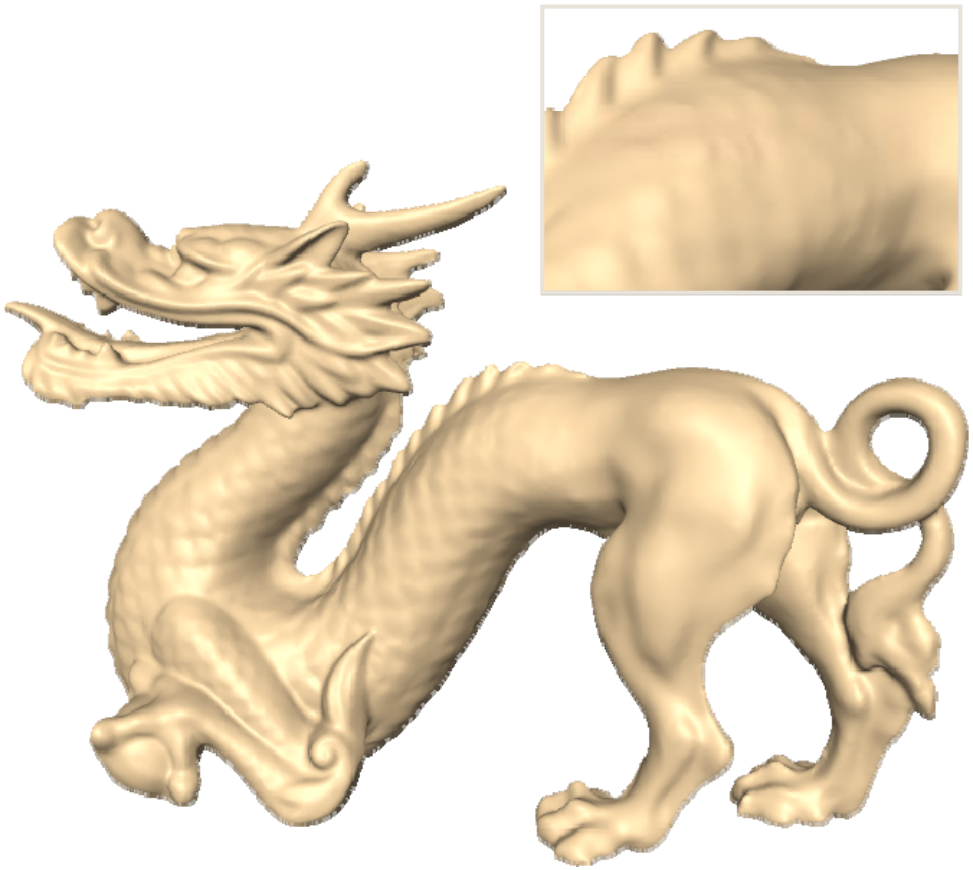
Mesh transplanting

- Details gradually change in the transition area



Mesh transplanting

- Details gradually change in the transition area



作业7

- 任务：
  - 实现极小曲面的全局方法：边界固定，求解方程组
  - 实现曲面参数化：边界映射到平面，求解方程组
    - 只要实现Floater1997论文中的一种方法（cot权）即可，其他的可选
- 目的
  - 学习使用数学库（推荐Eigen库）求解稀疏线性方程组
- 数据
  - 带一条边界的网格曲面（暂不处理复杂曲面）
- Deadline：2020 年12 月12日晚

中国科学技术大学

University of Science and Technology of China

谢　谢　！