



中国科学技术大学

University of Science and Technology of China

博士研究生课程报告

# 多边形网格处理

**答辩人：周航 BA17033001**

**指导老师：俞能海教授 张卫明教授**

**2019.5.23**

# CONTENTS

## 目录

1

选题说明和研究背景

---

2

微分几何

---

3

平滑和滤波

---

4

参数化

---

5

网格重划分

---

6

小结与展望

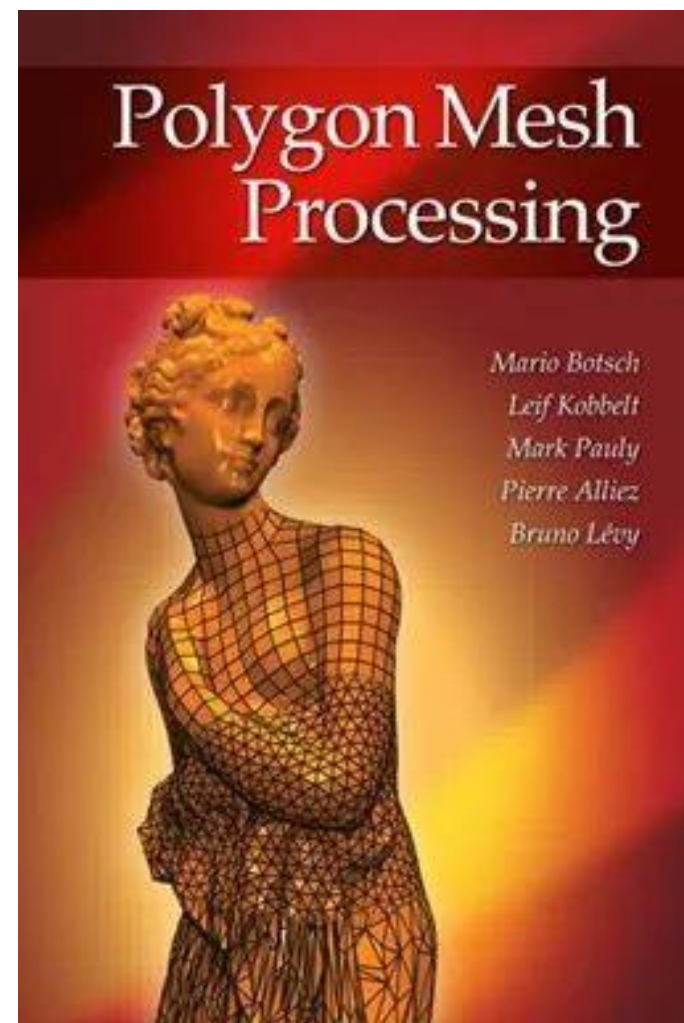
---

# 1

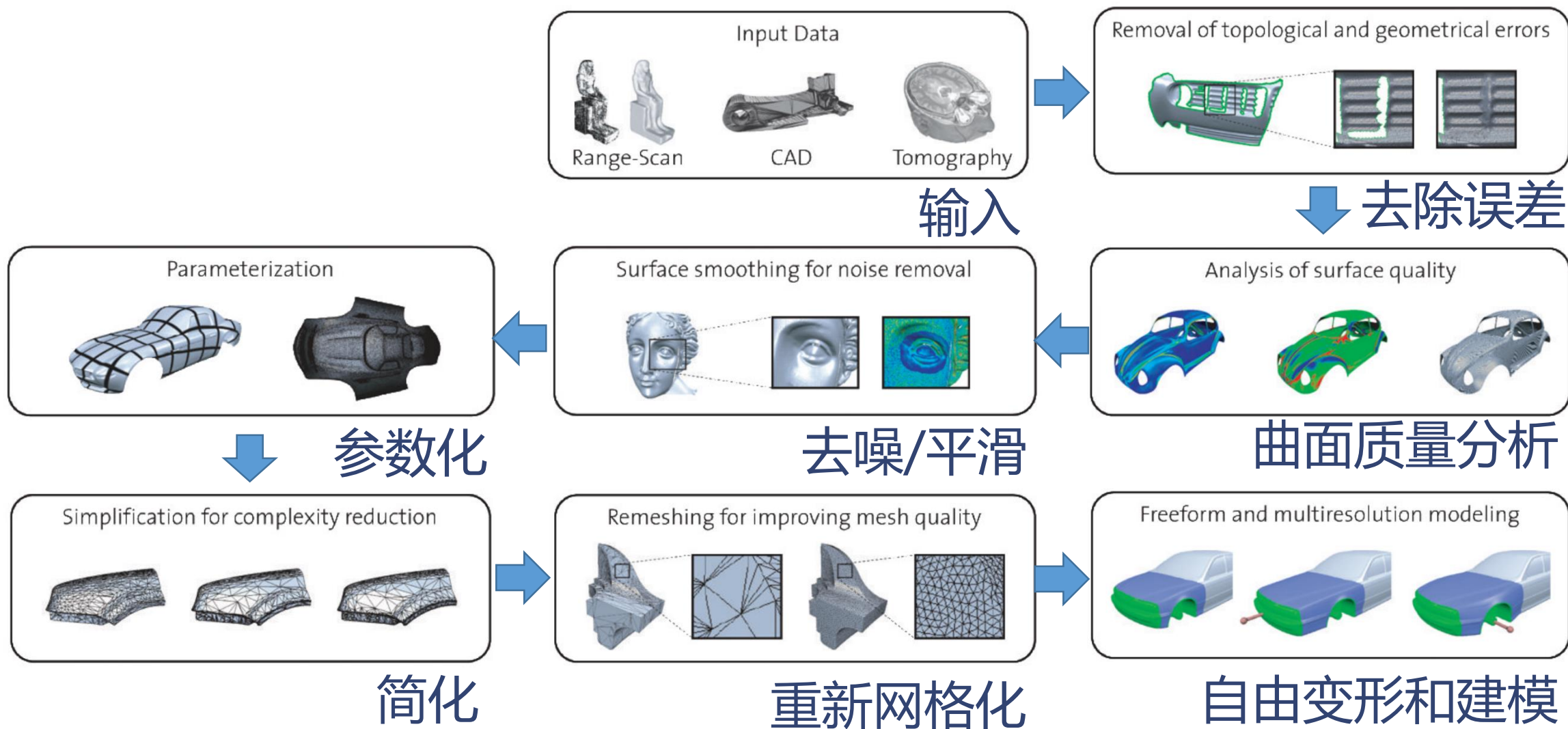
## 选题说明和研究背景

- 阅读材料
- 几何处理流程
- 研究方向储备
- 3D基本格式

- 图形学研究包含：图形学基础原理、渲染基础与算法、计算几何与**几何处理**、表观建模等；
- 《多边形网格处理》 (*Polygon Mesh Processing*) 涵盖了几何处理的多个研究方向；
- 作者：Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, Bruno Levy。



# 几何处理流程



- 3D隐写与隐写分析 (滤波、曲率、法向量、...)
- 3D可逆隐藏

## 2.1 3D Mesh Steganography

Cayre and Marcq [2] embed data by modulating the position of the orthogonal projection of a triangle summit on the opposite side. Wang and Cheng [3] improve the modulation by sliding, extending and rotating levels to embed data; they also combine both the spatial domain and the representation domain [4] to increase the embedding capacity. Other follow-ups with small embedding capacities mainly focus on perfecting visual distortions caused by modifications [5, 6].

High-capacity based 3D mesh steganography can be classified into two categories: distortionless steganography based on order permutation [18, 19, 20], and distorted steganography based on vertex shifting [7, 8, 9, 10, 11, 12]. For distortionless steganography, Bogomjakov *et al.* [18] propose to hide messages in the indexed representation of a mesh by permuting the order in which faces and vertices are stored. Tan *et al.* [19] consider the vertex index as a message block, and propose three embedding strategies: vertex index embedding, dynamic-length bit-string mapping, and repeated bits embedding to enlarge embedding capacity. Tu and Tai [20] propose to use maximum expected level tree based on a message probability model to embed data, which owns larger capacity than previous methods. For distorted

### 2.2.1 Calibration and Smoothing

Before feature extraction, it is necessary to preprocess vertices into the canonical version: the mesh object is rotated and aligned according to its first and second principal axes, which are given by the principal component analysis (PCA) algorithm. The guidance from image steganalysis indicates that the distinction between a stego image and its smoothed image is more significant than the distinction between a cover image and its smoothed image [24, 25]; Analogically, it is expected that the differences follow the same rule. The smoothed reference mesh  $\mathcal{M}'$  is produced by applying the umbrella operator or one iteration of Laplacian smoothing on the original mesh  $\mathcal{M}$ , which changes the vertex  $\mathbf{v}_i$  to  $\mathbf{v}'_i$  for  $i = 1, \dots, N$ , as follows [26]:

$$\mathbf{v}'_i = \mathbf{v}_i + \frac{\tau}{\sum_{\mathbf{v}_j \in \mathcal{N}(\mathbf{v}_i)} w_{ij}} \sum_{\mathbf{v}_j \in \mathcal{N}(\mathbf{v}_i)} w_{ij} (\mathbf{v}_j - \mathbf{v}_i), \quad (1)$$

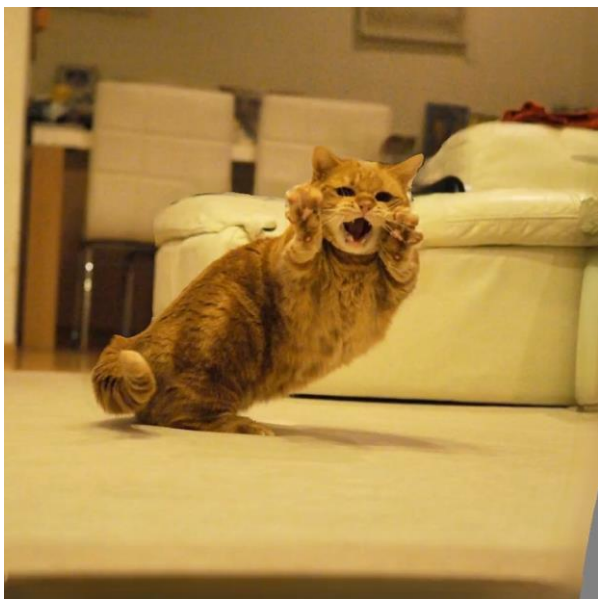
where  $\tau$  is a scalar factor and  $w_{ij}$  is the weight defined by

$$w_{ij} = \begin{cases} 1 & \text{if } \mathbf{v}_j \in \mathcal{N}(\mathbf{v}_i) \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

### 2.2.2 Existing Features

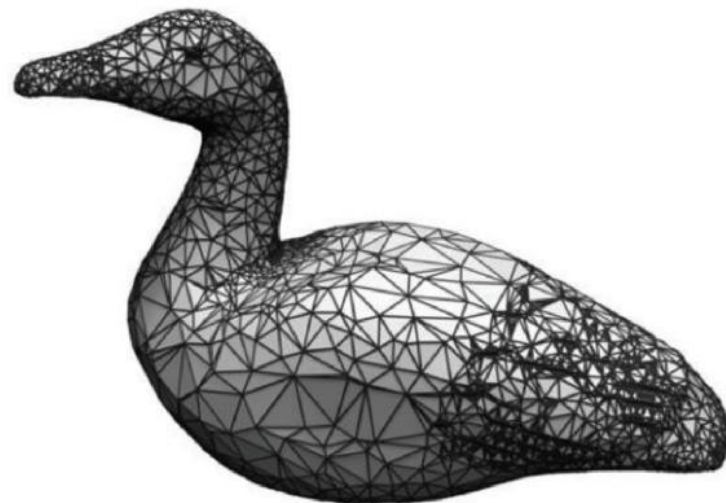
# 研究方向储备

- 3D照片课题

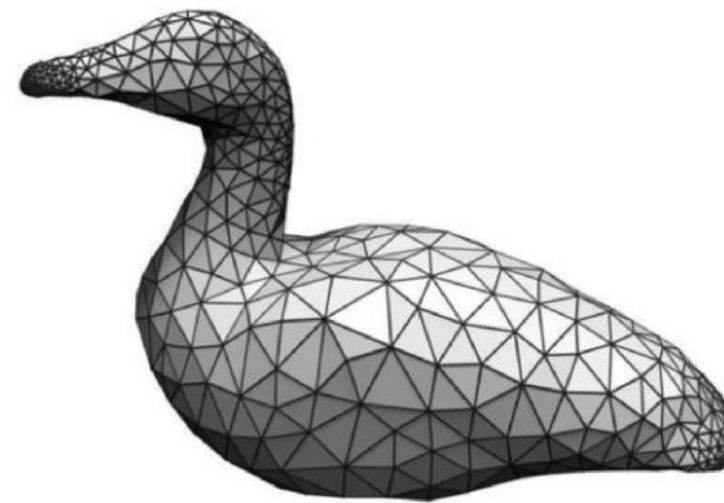


- 3D打印

- 网格重划分攻击等



原模型



网格重划分后模型



# 3D基本格式

顶点序列:  $\mathcal{V} = \{v_1, \dots, v_V\}$

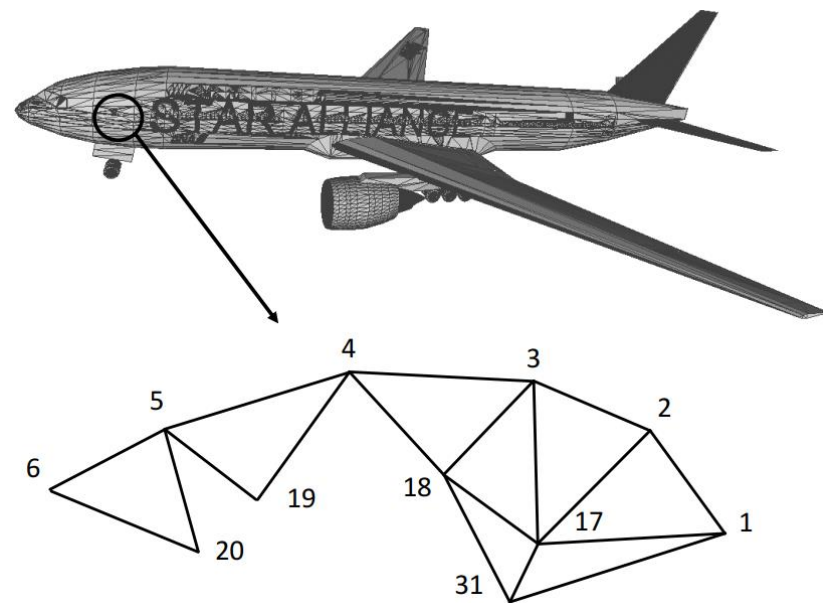
三角面片序列:  $\mathcal{F} = \{f_1, \dots, f_F\}$ ,  $f_i \in \mathcal{V} \times \mathcal{V} \times \mathcal{V}$

边序列:  $\mathcal{E} = \{e_1, \dots, e_E\}$ ,  $e_i \in \mathcal{V} \times \mathcal{V}$

坐标:  $\mathcal{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_V\}$ ,  $\mathbf{p}_i := \mathbf{p}(v_i) = \begin{pmatrix} x(v_i) \\ y(v_i) \\ z(v_i) \end{pmatrix} \in \mathbb{R}^3$

欧拉公式: 在一个闭合连接网格内, 点、边和面的数量存在以下关系:

$$V - E + F = 2(1 - g)$$



# 2

## 微分几何

---



曲面

离散微分算子

- 局部平均算子 (Local Averaging Region)
- 离散拉普拉斯算子

(Discrete Laplace-Beltrami Operator)

# 曲面 (Surfaces)

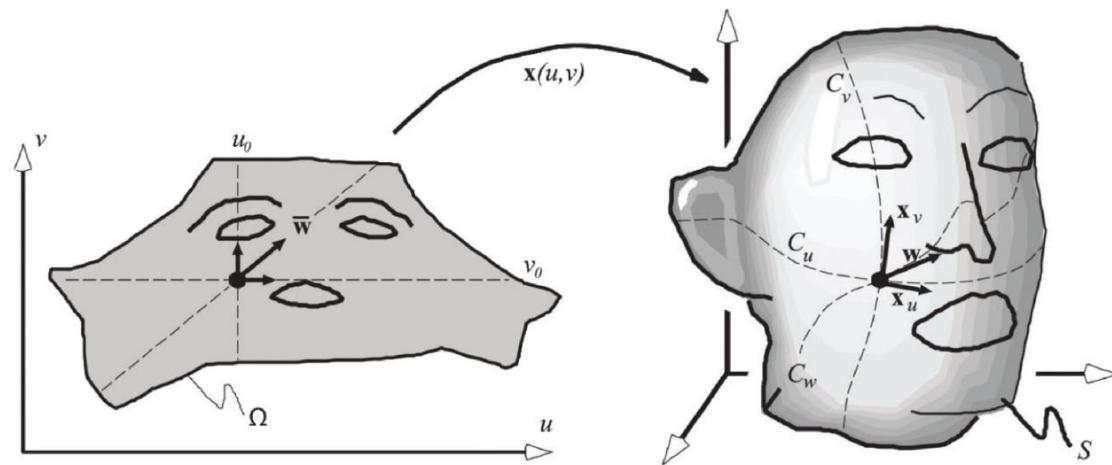
曲面参数方程:  $\mathbf{x}(u, v) = \begin{pmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{pmatrix}, (u, v) \in \Omega \subset \mathbb{R}^2$

曲面的第一基本型:  $\mathbf{I} = \mathbf{J}^T \mathbf{J} = \begin{bmatrix} E & F \\ F & G \end{bmatrix} := \begin{bmatrix} \mathbf{x}_u^T \mathbf{x}_u & \mathbf{x}_u^T \mathbf{x}_v \\ \mathbf{x}_u^T \mathbf{x}_v & \mathbf{x}_v^T \mathbf{x}_v \end{bmatrix}$

曲面的第二基本型:  $\mathbf{II} = \begin{bmatrix} e & f \\ f & g \end{bmatrix} := \begin{bmatrix} \mathbf{x}_{uu}^T \mathbf{n} & \mathbf{x}_{uv}^T \mathbf{n} \\ \mathbf{x}_{uv}^T \mathbf{n} & \mathbf{x}_{vv}^T \mathbf{n} \end{bmatrix}$

用于计算曲率

曲率、平均曲率、高斯曲率



## 内蕴几何学 (Intrinsic geometry) :

- 只依赖于第一基本型的属性被称为是内蕴的。
- 直观上来说它们可以仅仅通过曲面二维特征来导出。例如曲面上曲线的长度, 角度等都是内蕴的。
- 内蕴通常被用来表示参数的独立性。

# 局部平均算子

拉普拉斯算子的定义:

函数梯度的散度

$$\Delta f = \operatorname{div} \nabla f = \operatorname{div} \begin{pmatrix} f_u \\ f_v \end{pmatrix} = f_{uu} + f_{vv}$$

二阶流形表面上的拉普拉斯算子,  
称为**拉普拉斯—贝尔特拉米算子**:

$$\Delta_S f = \operatorname{div}_S \nabla_S f$$

$$\Delta_S \mathbf{x} = -2H\mathbf{n}$$

局部平均算子

计算网格某个点以及与其相邻点的微分属性的平均值

# 离散拉普拉斯算子

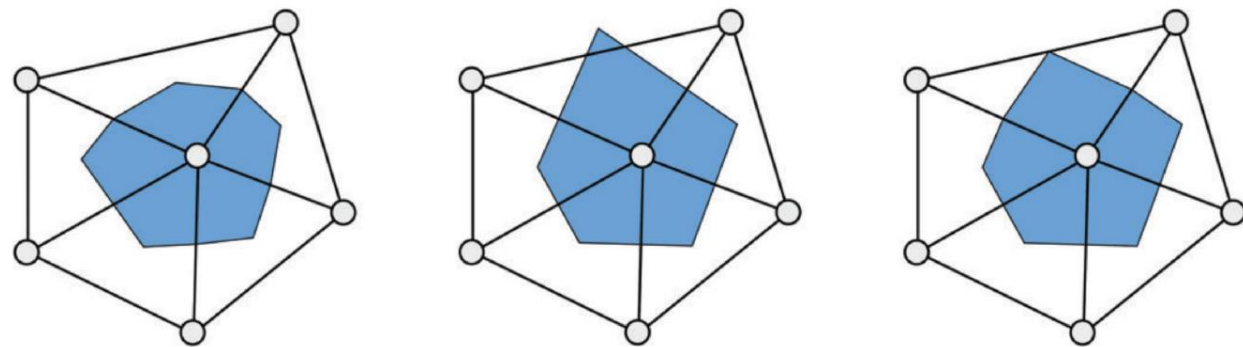
**均匀形式:** 
$$\Delta f(v_i) = \frac{1}{|\mathcal{N}_1(v_i)|} \sum_{v_j \in \mathcal{N}_1(v_i)} (f_j - f_i)$$

以中心点  $i$  为起点, 相邻顶点平均值为终点的向量

**余切形式:** 
$$\Delta f(v_i) := \frac{1}{2A_i} \sum_{v_j \in \mathcal{N}_1(v_i)} (\cot \alpha_{i,j} + \cot \beta_{i,j}) (f_j - f_i)$$

**平均曲率:** 
$$H(v_i) = \frac{1}{2} \|\Delta \mathbf{x}_i\|$$

**高斯曲率:** 
$$K(v_i) = \frac{1}{A_i} \left( 2\pi - \sum_{v_j \in \mathcal{N}_1(v_i)} \theta_j \right)$$



重心 (左) , 外心 (中) , 混合点 (右)

离散散度、曲率张量等参数

# 3

## 平滑和滤波

---

- 傅里叶变换与流形谐波
- 扩散流
- 平均

- 去噪 (de-noising)

一般是去掉凸出曲面的部分（高频部分），而保留和曲面相当的部分（低频部分）。即需要一个在离散三角形网格表面上的低通滤波器。包含**傅里叶变换**和**扩散流**。

- 抹平 (fairing)

在抹平的过程中，所做的不仅仅只是去除高频的部分。抹平的过程相当于是对曲面做了一个变换，使其从各个角度（曲率、高阶导数）上看尽可能的光滑。

# 傅里叶变换与流形谐波

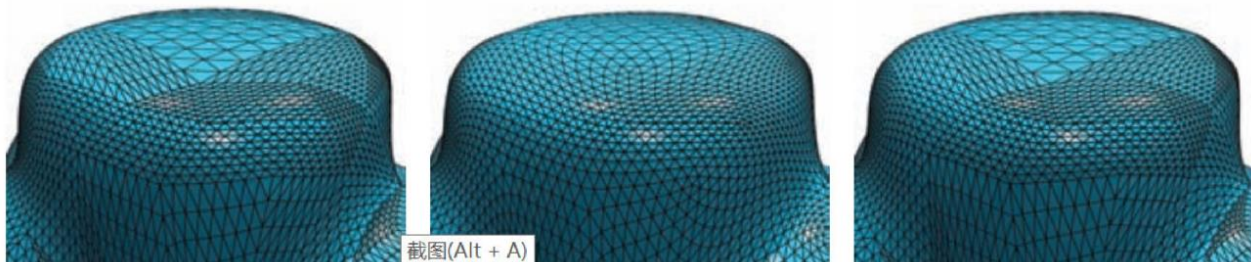
## 连续信号滤掉高频

一维傅里叶变换:  $f(x) = \sum_{\omega=-\infty}^{\infty} \langle f, e_{\omega} \rangle e_{\omega}$

基的变换。通过将向量  $f$  投影到不同频率的基向量上, 然后对其进行累加操作, 这样就完成了从空间域到频域的转变。

去除高频部分:  $|\omega| < \omega_{\max}$

$$\tilde{f}(x) = \int_{-\omega_{\max}}^{\omega_{\max}} \langle f, e_{\omega} \rangle e_{\omega} d\omega$$



原模型 (左), 均匀拉普拉斯 (中), 余切拉普拉斯 (右)

## 3D网格离散信号滤掉高频

离散的三角形网格进行拉普拉斯—贝尔特拉米算子变换:

$$\begin{pmatrix} \Delta f(v_1) \\ \vdots \\ \Delta f(v_n) \end{pmatrix} = \mathbf{L} \begin{pmatrix} f(v_1) \\ \vdots \\ f(v_n) \end{pmatrix}$$

每一行的运算:  $\Delta f(v_i) = \sum_{v_j \in \mathcal{N}_1(v_i)} w_{ij} (f(v_j) - f(v_i))$

函数  $e_{\omega}$  是拉普拉斯算子的特征函数

$$\Delta (e^{2\pi i \omega x}) = \frac{d^2}{dx^2} e^{2\pi i \omega x} = -(2\pi \omega)^2 e^{2\pi i \omega x}$$

$$\Delta e_i = \lambda_i e_i$$

对于矩阵的特征向量  $\mathbf{L}$  中的特征向量  $e_i$ :

$$f = \sum_{i=1}^n \langle e_i, f \rangle e_i \quad \longrightarrow \quad \tilde{f} = \sum_{i=1}^m \langle e_i, f \rangle e_i$$



# 扩散流 (Diffusion Flow)

## 基于傅里叶变换的频域 高频滤除方法的缺陷:

诸如热扩散和布朗运动之类的物理过程 (扩散过程):

应用于曲面网格:

用上式更新网格的每一个定点:

为了得到特征向量需要对拉普拉斯矩阵进行特征分解, 而当模型的顶点比较多时, 计算代价很高。

$$\frac{\partial f(\mathbf{x}, t)}{\partial t} = \lambda \Delta f(\mathbf{x}, t)$$

$$\frac{\partial}{\partial t} f(v_i, t) = \lambda \Delta f(v_i, t), \quad i = 1, \dots, n$$



$$(\mathbf{Id} - h\lambda\mathbf{L})\mathbf{f}(t+h) = \mathbf{f}(t)$$

$$\mathbf{x}_i \leftarrow \mathbf{x}_i + h\lambda\Delta\mathbf{x}_i$$

$$\Delta_S\mathbf{x} = -2H\mathbf{n}$$



原模型 (左) 迭代10次 (中) 迭代100次 (右)

本质上是让每一个定点沿着其法向量的方向移动, 移动距离由这一点的平均曲率H决定

**扩散流方法比傅里叶变换来说计算量更低, 但是其主要的思想仍然是移除高频噪音 (模型上不平滑的地方) 而保留低频部分。**

# 抹平 (Fairing)

## 总思路:

1. 定义一个量来衡量曲面的不平滑程度
2. 更新曲面以最小化曲面的不平滑程度

① 曲面面积:

$$E_M(\mathbf{x}) = \iint_{\Omega} \sqrt{\det(\mathbf{I})} du dv$$

② 主曲率:

$$E_{TP}(\mathbf{x}) = \iint_{\Omega} \kappa_1^2 + \kappa_2^2 du dv$$

③ 主曲率相对切向量的变化率:  $\iint_{\Omega} \left( \frac{\partial \kappa_1}{\partial \mathbf{t}_1} \right)^2 + \left( \frac{\partial \kappa_2}{\partial \mathbf{t}_2} \right)^2 du dv$



不同抹平参数下的平滑效果。

# 4

## 参数化

---

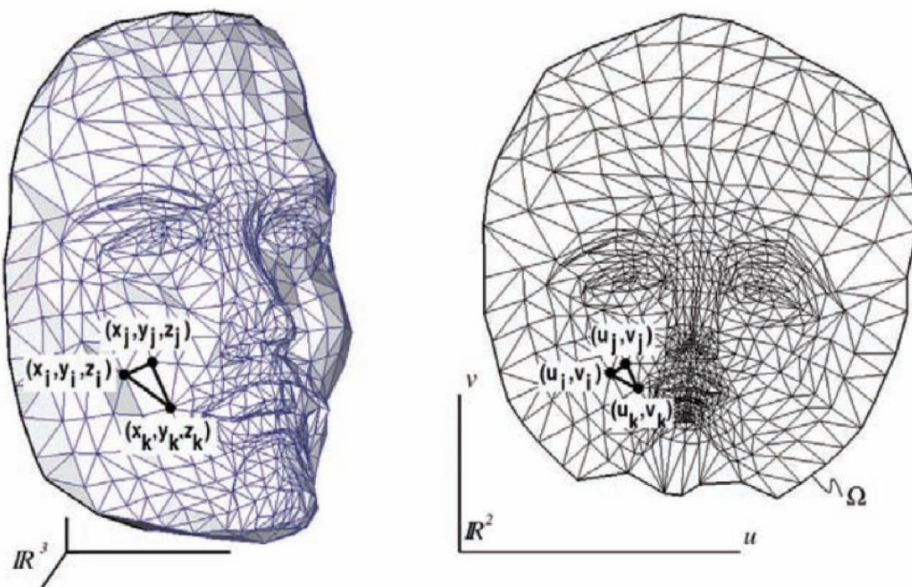
- 重心映射 (Barycentric Mapping)
- 保角映射 (Conformal Mapping)
- 形变分析 (Distortion Analysis)

# 参数化 (Parameterization)

## 主要目标:

将复杂的3维模型转换到2维空间上。对于一个三角网格模型，即把它拍平到一个平面上。

数学解释：三角形网格参数化的过程就是寻找一种映射，即把网格的每一个顶点  $i$  映射到  $(u_i, v_i)$  上。经过映射后，任意两个三角形公共的部分只有可能是：一条边，一个顶点或者不存在。



表面参数化示意图

# 重心映射

## 前提:

三角形网格与圆盘是同胚的，并且网格的边界在一个凸多边形上，同时内部的顶点是其周围顶点的凸组合。

## 更新方法:

$$\begin{pmatrix} u_i \\ v_i \end{pmatrix} \leftarrow \frac{1}{a_{i,i}} \sum_{j \neq i} a_{i,j} \begin{pmatrix} u_j \\ v_j \end{pmatrix}$$

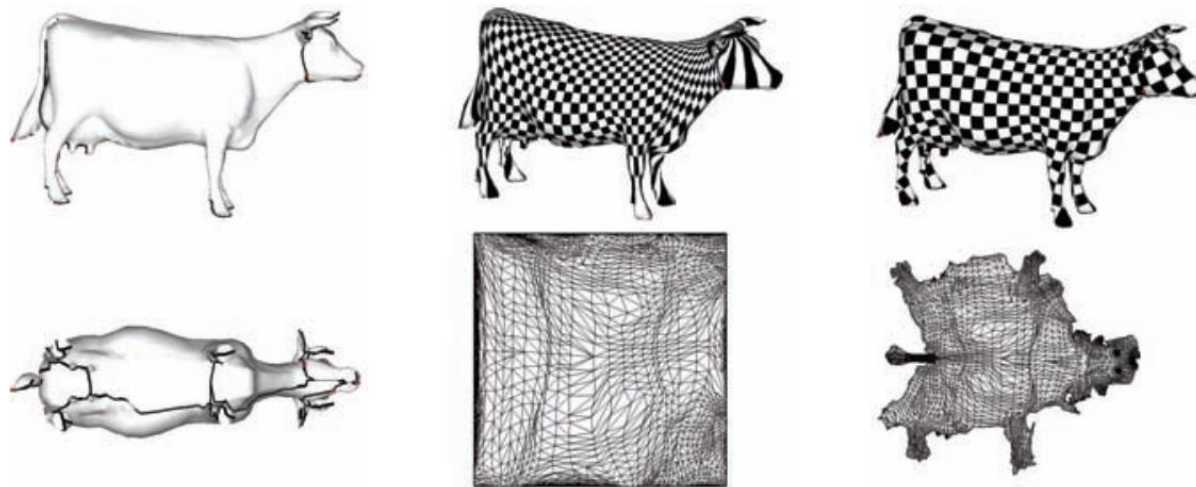
①简单模式:  $a_{i,j} = \begin{cases} 1 & \text{如果 } i \text{ 和 } j \text{ 通过边相连} \\ -|N_i| & \text{其他} \end{cases}$

## ②离散拉普拉斯:

$$a_{i,j} = \frac{1}{\|\mathbf{x}_i - \mathbf{x}_j\|} \left( \tan\left(\frac{\delta_{i,j}}{2}\right) + \tan\left(\frac{\gamma_{i,j}}{2}\right) \right),$$

$$a_{i,i} = - \sum_{j \neq i} a_{i,j},$$

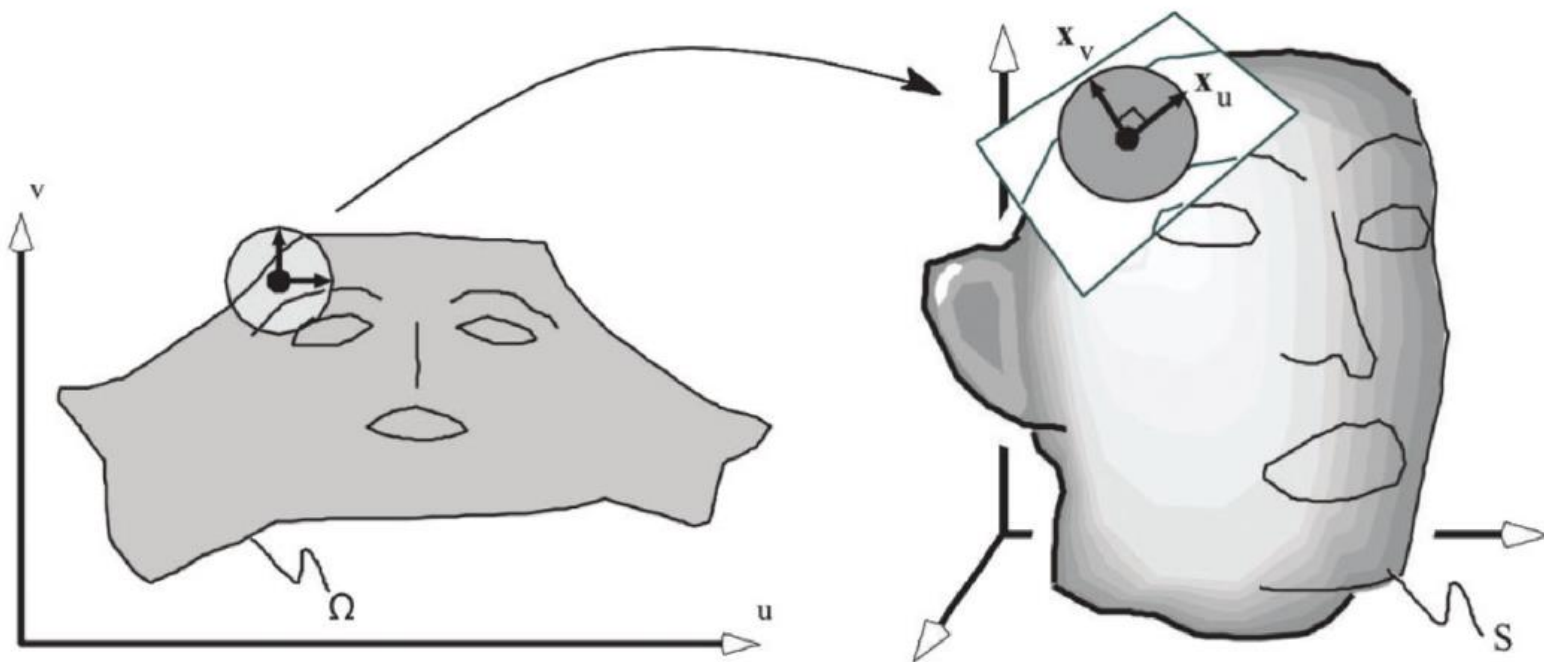
边界区域设置。圆盘（左），矩形（中），自由边界（右）。



## 局限:

无论使用上面哪种方法，都需要手动指定并固定网格的边界，但是对于一个模型往往选取边界不是一件容易的事情。边界的选取与固定会影响到参数化的结果。

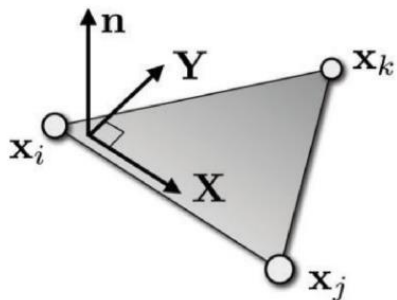
# 保角映射



假设参数空间  $(u, v)$  上存在一个小圆，使用保角映射将它变换到 3 维坐标空间之后，其沿着  $u$  和  $v$  方向的切向量  $\mathbf{x}_u$  和  $\mathbf{x}_v$  的模长应该同样地是相等并且  $\mathbf{x}_u$  和  $\mathbf{x}_v$  正交。对于此变化的逆变化，即从 3 维坐标空间变换为 2 维参数空间也同样成立。

# 保角映射

三角形局部X、Y基:



$$\mathbf{X} = \frac{\mathbf{x}_j - \mathbf{x}_i}{\|\mathbf{x}_j - \mathbf{x}_i\|}$$

$$\mathbf{n} = \frac{\mathbf{X} \times (\mathbf{x}_k - \mathbf{x}_i)}{\|\mathbf{X} \times (\mathbf{x}_k - \mathbf{x}_i)\|}$$

$$\mathbf{Y} = \mathbf{n} \times \mathbf{X}$$

函数梯度:

$$\nabla u = \begin{bmatrix} \partial u / \partial X \\ \partial u / \partial Y \end{bmatrix} = \frac{1}{2A_T} \underbrace{\begin{bmatrix} Y_j - Y_k & Y_k - Y_i & Y_i - Y_j \\ X_k - X_j & X_i - X_k & X_j - X_i \end{bmatrix}}_{=\mathbf{M}_T} \begin{pmatrix} u_i \\ u_j \\ u_k \end{pmatrix}$$

参数空间下  $u$  和  $v$  分量的基向量垂直:

$$\nabla v = \mathbf{n} \times \nabla u \quad \longrightarrow \quad \nabla v = (\nabla u)^\perp = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \nabla u$$

上式代入:

$$\mathbf{M}_T \begin{pmatrix} v_i \\ v_j \\ v_k \end{pmatrix} - \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \mathbf{M}_T \begin{pmatrix} u_i \\ u_j \\ u_k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

求解 (保角能量):

$$E_{\text{LSCM}} = \sum_{T=(i,j,k)} A_T \left\| \mathbf{M}_T \begin{pmatrix} v_i \\ v_j \\ v_k \end{pmatrix} - \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \mathbf{M}_T \begin{pmatrix} u_i \\ u_j \\ u_k \end{pmatrix} \right\|^2$$

# 形变分析

**背景：** 纹理映射时，如何减小形变？

**保角：** 当第一基本型是单位矩阵的倍数或者奇异值相同的时候，向量的角度得到了保留。

**保面：** 第一基本型行列式值为1或者所有奇异值相乘值为1时，圆形面积在变换后不变。

**等距：** 第一基本型是单位矩阵或者奇异值为1。

求解方法：

- ①. 格林—拉格朗日变形张量法
- ②. 最等距参数表面法
- ③. 专用信号参数化法



# 5

## 网格重划分

---

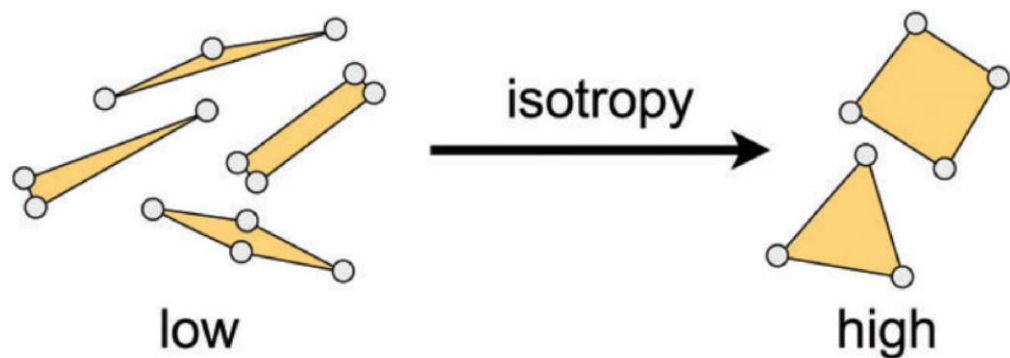
- 沃洛诺伊图
- 德洛内三角剖分
- 网格重划分算法

# 网格重划分 (Remeshing)

**定义:** 输入一个 3D 网格, 通过计算得到另一个和输入大致相同且满足一定质量要求网格。

- 目标:
- ①. 根据需求减少曲面的复杂度
  - ②. 改善曲面的质量

曲面质量: 非拓扑属性  
(采样密度, 正则性, 大小, 方位, 对齐性, 曲面网格形状)



**正则:** 三角形网格上非边界上的顶点周围顶点的数量为 6 或边界上的顶点周围顶点的数量为 4 时。

网格形状: 各向同性/各项异性。

# 网格重划分 (Remeshing)

**一致性：** 所有网格重划分算法都会在曲面上计算点的位置。大部分算法甚至还会进行额外的迭代来修正顶点的位置以改善网格的质量。在网格重划分的过程中一个关键的问题就是要保证计算前后顶点的一致性。

**解决方法：**

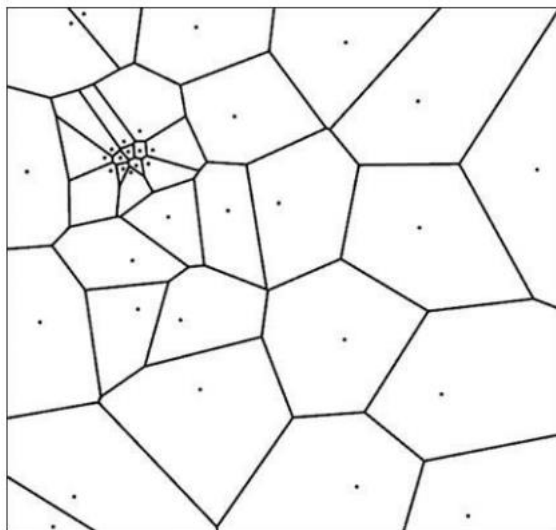
- **全局参数化：** 将输入的模型整个参数化到一个2维参数域上，然后在这个参数域对采样点的分布和位置进行调整，最后再将其还原到3维空间中。
- **局部参数化：** 算法只保留某个点局部的参数化信息，当采样点变换的时候，再去计算那个点的局部参数化信息。
- **投影：** 将采样点投影到输入模型上对应最近的顶点、边或者三角形上。

**优缺点：**

- **全局参数化：** 当模型被切开成一个圆盘时，计算比较耗时。
- **投影：** 当采样点离曲面太远时，可能会导致重影现象。
- **局部参数化：** 相对稳定，效果较好，不过需要不断的跟踪记录并重新计算局部的参数化信息。

# 沃洛诺伊图 (Voronoi Diagrams)

**概念:** 基于一组特定点将空间分割成不同的区域，而每一个区域都只包含这些点中的一个，并且该区域内的任意点到这个特定点的距离小于这个任意点到空间中其它特定点的距离。其中这些被分割的区域称作沃罗诺伊区域。



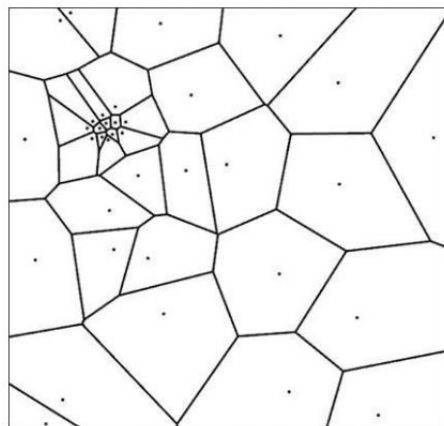
2D沃洛诺伊图

给定任意维空间  $\mathbb{R}^d$  上的一个点的集合  $\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$ ，点  $\mathbf{p}_i$  的沃罗诺伊区域  $V(\mathbf{p}_i)$  是：

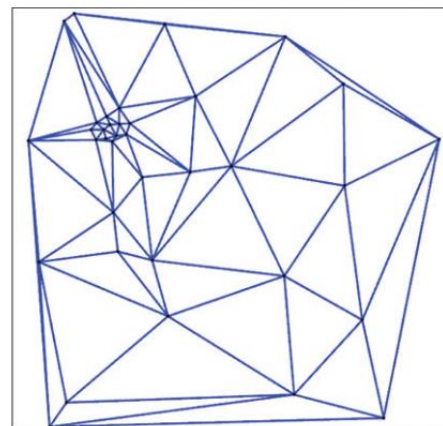
$$V(\mathbf{p}_i) = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x} - \mathbf{p}_i\| \leq \|\mathbf{x} - \mathbf{p}_j\|, \forall j \neq i\}$$

# 德洛内三角剖分 (Delauney Triangulations)

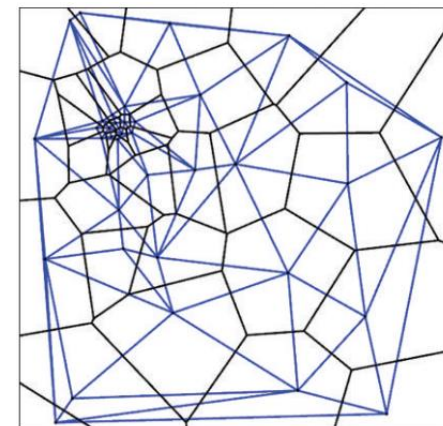
**概念：** 沃罗诺伊图的**对偶结构**被称为德洛内三角剖分。



沃洛诺伊图



德洛内三角剖分



混合图

德洛内三角剖分其实并不是一种算法，它只是给出了一个“好的”三角网格的定义，它的优秀特性是空圆特性和最大化最小角特性，这两个特性避免了狭长三角形的产生。

# 网格重划分算法

常见算法:

- ①. 贪心算法
- ②. **变分算法**
- ③. 增量算法

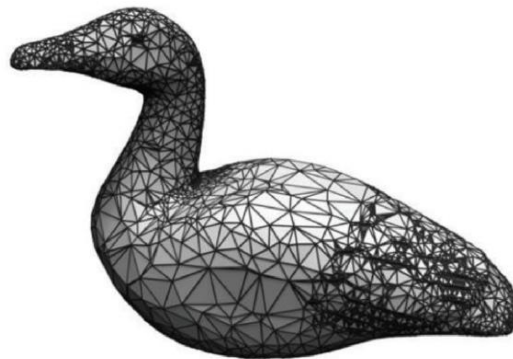
## 介绍变分算法:

- **优化的标准是什么?** 一些和形状、三角大小等相关的几何量。
- **自由度是多少?** 尽可能的少。

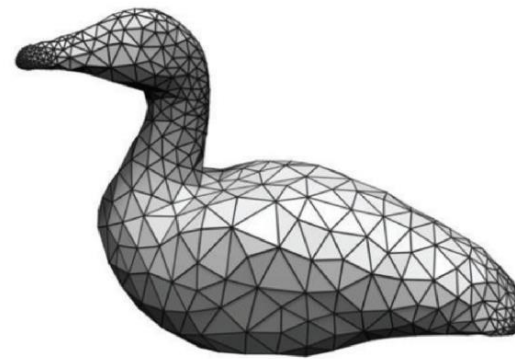
**主要思想:** 将一系列的点尽可能平均地放置到输入的网格上。

构建**质心沃洛诺伊图镶嵌** (Centroidal Voronoi Tessellation, 简称CVT) :

给定一个边界域 $\Omega$ , 如果存在一个被 $\Omega$ 限制的沃洛诺伊划分, 其上的任意一个沃洛诺伊区域中的顶点刚好是这个区域的质心, 那么称这个这个划分为CVT。



原模型



网格重划分后模型

# 变分算法

实际采用**洛依德算法** (Lloyd' s Algorithm) 构建CVT。其中, 每个沃洛诺伊图的重心:

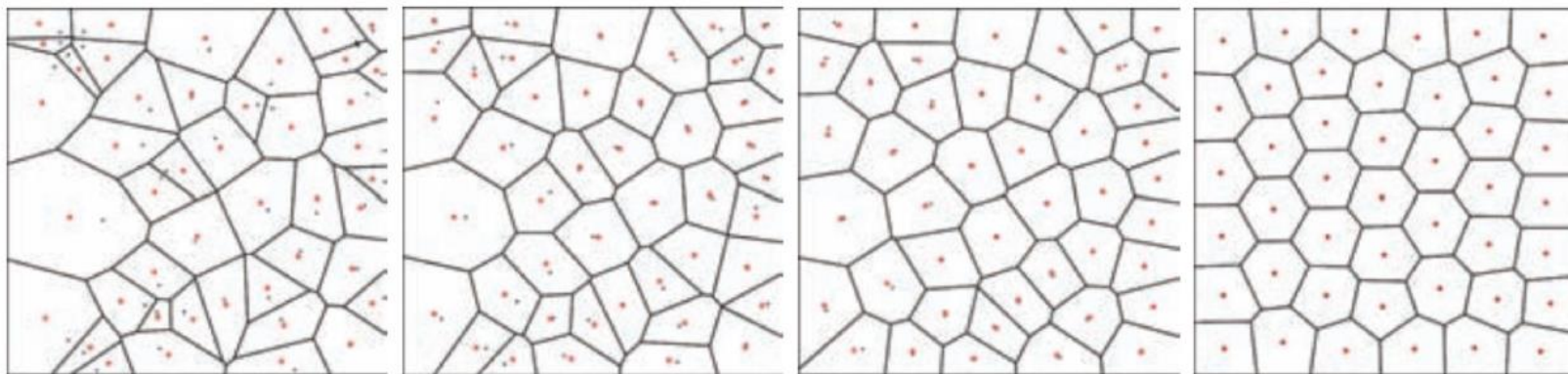
$$\mathbf{c}_i = \frac{\int_{V_i} \mathbf{x} \cdot \rho(\mathbf{x}) d\mathbf{x}}{\int_{V_i} \rho(\mathbf{x}) d\mathbf{x}}$$

优化过程等价于固定 $\mathbf{p}_i$ , 不断地迭代最小化以下**能量函数**:

$$E(\mathbf{p}_1, \dots, \mathbf{p}_n, V_1, \dots, V_n) = \sum_{i=1}^n \int_{V_i} \rho(\mathbf{x}) \|\mathbf{x} - \mathbf{p}_i\|^2 d\mathbf{x}$$

算法步骤:

1. 根据坐标点点集 $\mathbf{p}_i$ 建立沃洛诺伊划分 $V_i$ ;
2. 计算每一个沃洛诺伊区域的重心 $\mathbf{c}_i$ , 然后将坐标点 $\mathbf{p}_i$ 移动到 $\mathbf{c}_i$ 的位置;
3. 重复执行(1)(2)直到满足收敛条件。



从左至右: 沃洛诺伊图  
迭代过程

# 6

## 小结与展望

---

- 3D隐写分析 (滤波/平滑)
- 3D照片 (纹理映射: 网格参数化)
- 3D打印 (网格重划分)





中国科学技术大学

University of Science and Technology of China

**谢 谢！**