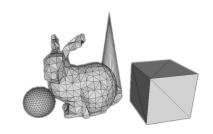
Introduction to Computer Graphics

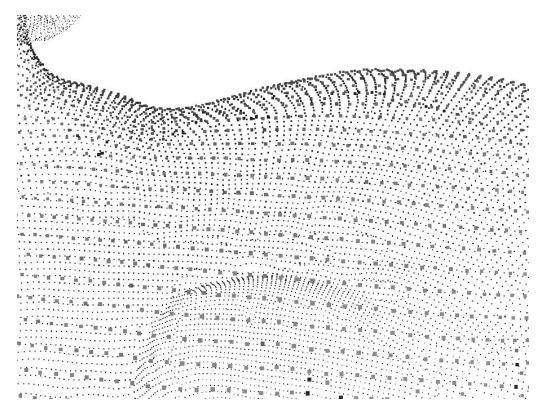


Geometry 2 几何 2

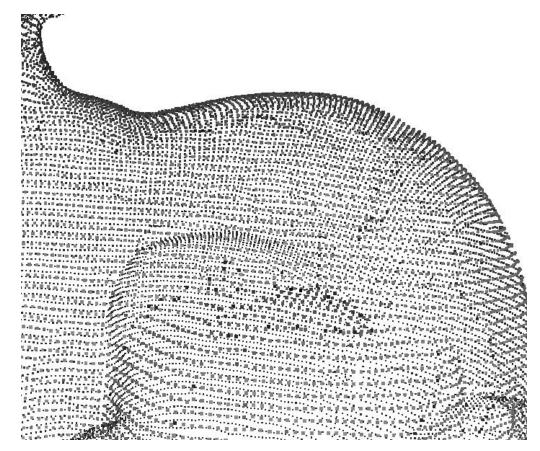
第八章 几何

- •几何的表示形式
 - 隐式表示
 - 显式表示
- 参数化曲线曲面
 - 贝塞尔曲线
 - 贝塞尔曲面
- 点云表示
- 体表示
- 网格表示
- 网格处理

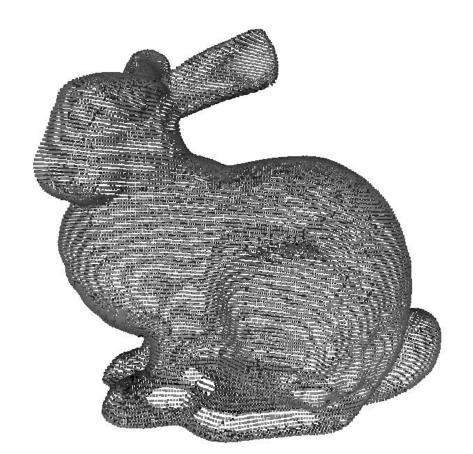
- •点的集合P={P1, P2, P3, ..., Pn}
- Pi={xi, yi, zi}



- 点的集合P={P1, P2, P3, ..., Pn}
- Pi={xi, yi, zi}



- •点的集合P={P1, P2, P3, ..., Pn}
- Pi={xi, yi, zi}



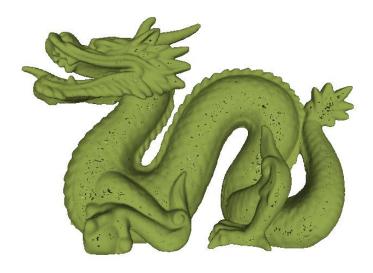
- 点的集合P={P1, P2, P3, ..., Pn}
- Pi={xi, yi, zi}



- 点的集合P={P1, P2, P3, ..., Pn}
- Pi={xi, yi, zi}



Armadillo



dragon

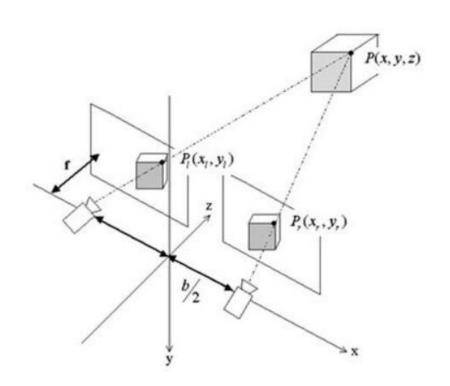


Happy buddha

点云的获取

•三维重建

- 主要思想: 双目视觉三角化



激光条带



结构光



Kinect

3D Reconstruction with Kinect

Augmented Vision DFKI 2010

多视图重建

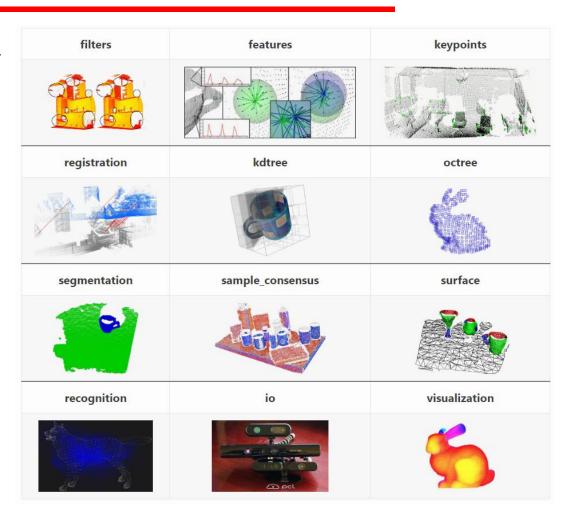


点云的处理与分析

•点云库 PCL

- 点云去噪
- 点云下采样
- •估计法向
- 点云分割

•



点云的注册

- Registration
- 将多个点云拼接成完整的点云
- 求最优的刚体变换, 使得两个点云可以重合



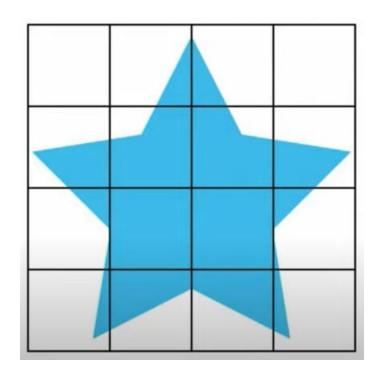
体表示

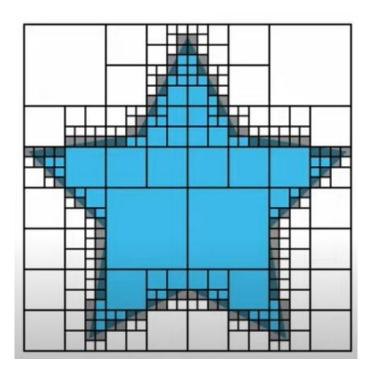
• 像素的扩展



体表示及稀疏结构

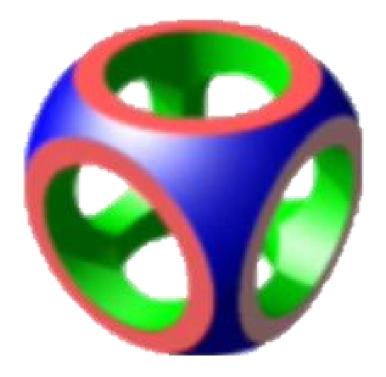
- •存储方式:1表示体内,0表示体外
 - 蓝色表示1, 白色为0 (既有白色也有蓝色可以任取一个)





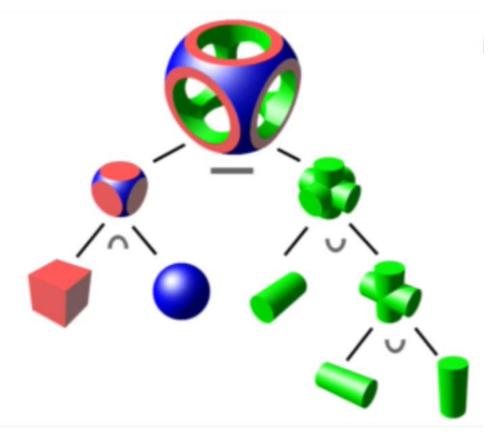
CSG

Constructive Solid Geometry



CSG

- Constructive Solid Geometry
- •体表示+布尔运算



• 符号距离场 (Signed Distance Field)

84.8	54.6	24.4	5.8	-17.0	8.2	33.4	58.2	69.2	80.3	91.7	116.5
49.3	14.8	-15.4	-45.7	-60.2	-35.0	-9.8	9.4	20.5	31.6	46.2	85.5
39.2	-10.3	-55.3	-85.5	-103.4	-78.2	-53.0	-39.4	-28.3	-17.2	21.3	68.6
32.3	-17.2	-66.7	-116.3	-146.6	-121.4	-99.3	-88.1	-77.0	-42.3	5.0	52.3
25.4	-24.1	-73.7	-104.2	-132.7	-161.8	-148.0	-136.9	-105.8	-58.6	-11.3	36.0
24.3	5	-34.6	-63.1	-91.7	-127.3	-170.1	-169.4	-120.5	-71.2	-22.0	27.2
63.5	35.0	6.4	-22.1	-58.4	-102.0	-133.5	-129.2	-111.8	-62.6	-13.3	35.9
104.6	76.0	47.5	9.7	-33.9	-77.5	-83.6	-80.1	-70.2	-53.9	-4.7	44.6
145.6	117.1	77.8	34.2	9.4	-31.1	-33.6	-31.1	-21.2	-11.3	4.0	53.2
186.7	145.9	102.3	58.7	21.9	18.9	16.3	17.9	27.8	37.7	47.6	72.4
214.0	170.4	127.3	90.2	71.4	68.8	66.2	66.9	76.8	86.7	96.6	111.9

• 截断符号距离场 (Truncated Signed Distance Field)

-0.9	-0.3	0.0	0.2	1	1	1	1	1
						1		1
						1	1	1
100	10		THE RESERVE	0.2	400	S S	1	1
	100		100		100	0.9	1	1
-1	-0.7	-0.3	0,8	0.3	0.6	1	1	1
-1	-0.7	-0.4	00	0.2	0.7	0.8	1	1
-0.9			1				1	1
-0.1	0.0	0.0	0.1	0.3	1	1	1	1
0.5	0.3	0.2	0.4	0.8	1	1	1	1

- ·如何从点云表示转成SDF表示呢?
 - 对每个voxel计算其与点云的最近距离
 - 根据点云的法向判断内外

- •如何从SDF转成点云
 - 找到临界点并进行插值

-0.9	-0.3	0.0	0.2	1	1	1	1	1
-1	-0.9	-0.2	9.0	0.2	1	1	1	1
-1	-0.9	-0.3	0.)	0.1	0.9	1	1	1
				0.2			1	1
	100		1	Q.1	707		1	1
-1	-0.7	-0.3	0,8	0.3	0.6	1	1	1
-1	-0.7	-0.4	00	0.2	0.7	0.8	1	1
-0.9	-0.7	-0.2	G O	0.2	0.8	0.9	1	1
-0.1	0.0	0.0	0.1	0.3	1	1	1	1
0.5	0.3	0.2	0.4	0.8	1	1	1	1

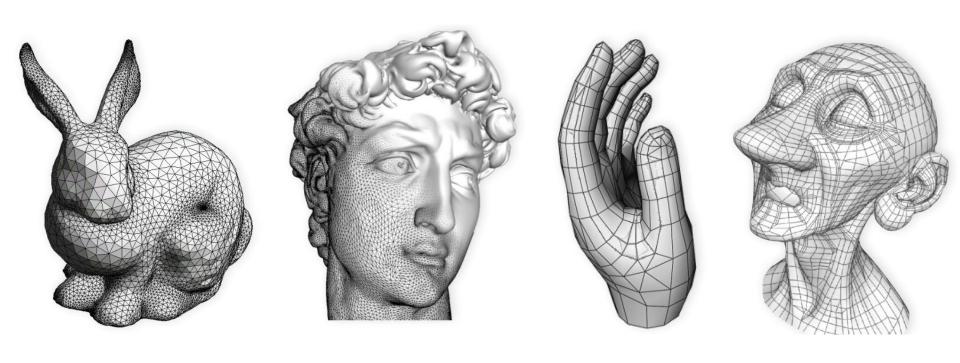
•优点

- 数据表示规整,与像素类似,便于做卷积等操作
- 数据整体性强,不容易受噪声干扰
- 适合用于数据的融合

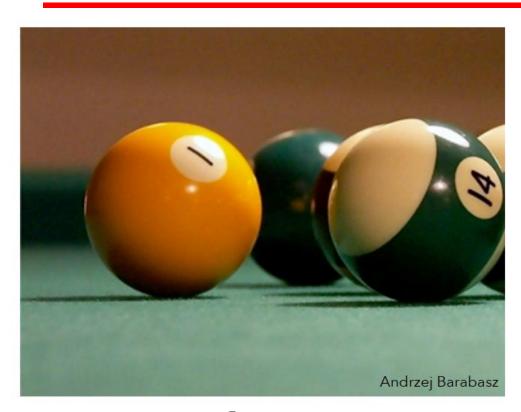
• 缺点:

- 数据量大
- 往往需要稀疏表示

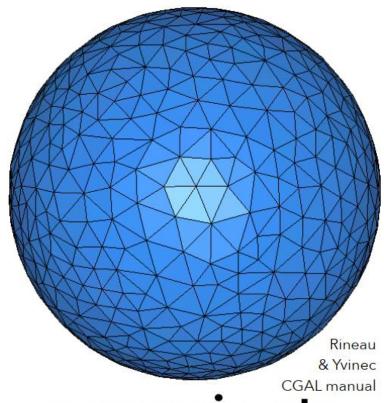
网格表示



网格表示

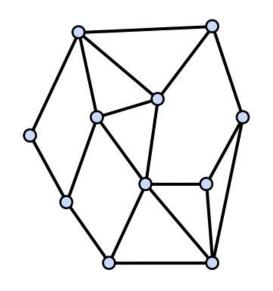


spheres



approximate sphere

网格表示



$$M = \langle V, E, F \rangle$$
 vertices edges faces

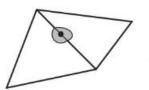
拓扑与几何

- Topology and Geometry
- 拓扑是什么? 顶点的连接关系
- Manifold (流形):流形是局部具有欧几里得空间性质的空间
- 二维流形:一个表面,当用一个小球去切割时,得到一个二维的圆盘。

拓扑与几何

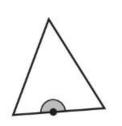
•二维流形:一个表面,当用一个小球去切割时,得到一个二维的圆盘。

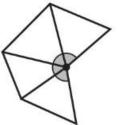
Manifold



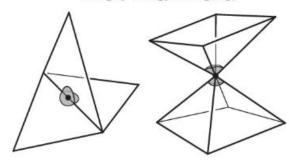


With border

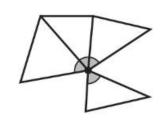




Not manifold

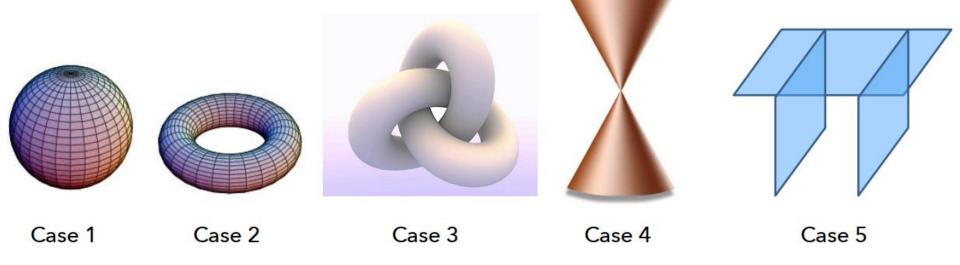


With border



拓扑与几何

•哪些是流形?



网格存储格式

• PLY, OBJ, STL, OFF等

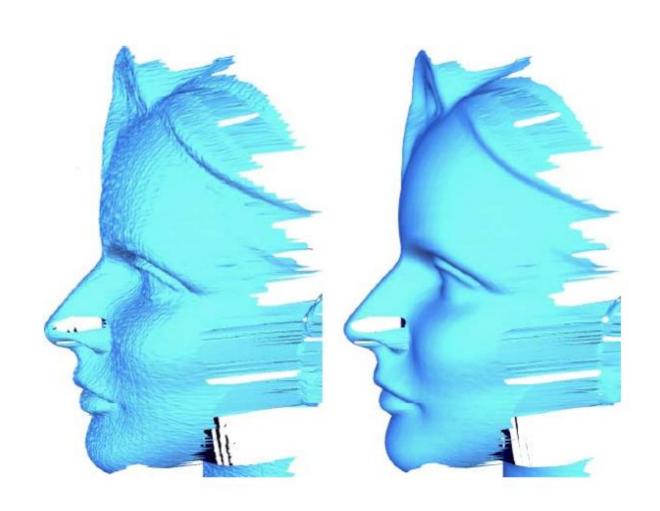
Triangles						
0	x0	y0	z0			
1	x1	y1	z1			
2	x2	y2	z2			
3	х3	у3	z3			
4	x4	у4	z4			
5	x5	y5	z5			
6	x6	у6	z6			
•••	•••	•••	•••			

T	riar	gle	s
t0	v0	v1	v2
t1	v0	v1	v3
t2	v2	v4	v3
t3	v5	v2	v6
•••	•••	•••	•••

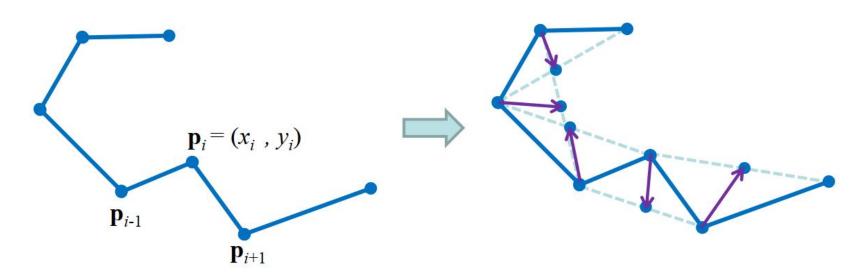
Vertices						
v0	x0	y0	z0			
v1	x1	x1	z1			
v2	x 2	y2	z2			
v3	х3	у3	z3			
v4	x4	y4	z4			
v5	x5	y5	z5			
v6	x6	у6	z6			
•••	•••	•••	•••			

三角形直接存顶点

三角形存顶点的索引



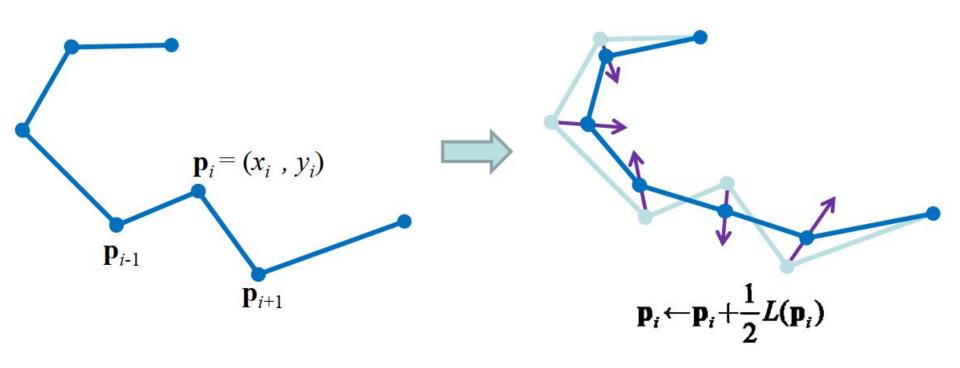
· 先以曲线为例,定义Laplacian算子



$$(\mathbf{p}_{i-1} + \mathbf{p}_{i+1})/2 - \mathbf{p}_i$$

$$L(\mathbf{p}_i) = \frac{1}{2} (\mathbf{p}_{i+1} - \mathbf{p}_i) + \frac{1}{2} (\mathbf{p}_{i-1} - \mathbf{p}_i)$$

· 每个顶点沿着Laplacian算子方向移动



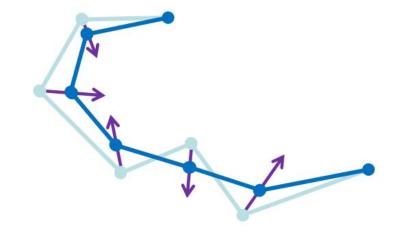
Algorithm:

Repeat for *m* iterations (for non boundary points):

$$\mathbf{p}_i \leftarrow \mathbf{p}_i + \lambda L(\mathbf{p}_i)$$

For which λ ?

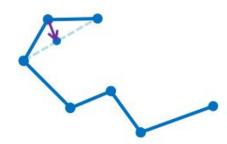
$$0 < \lambda < 1$$



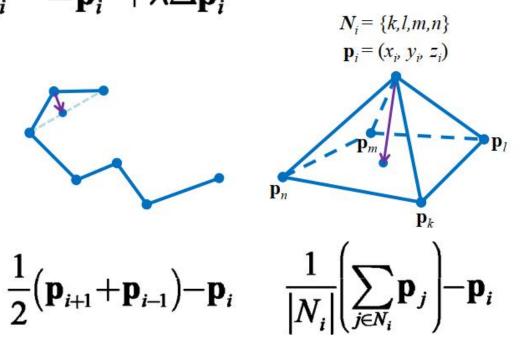
Same as for curves:

$$\mathbf{p}_{i}^{(t+1)} = \mathbf{p}_{i}^{(t)} + \lambda \Delta \mathbf{p}_{i}^{(t)}$$

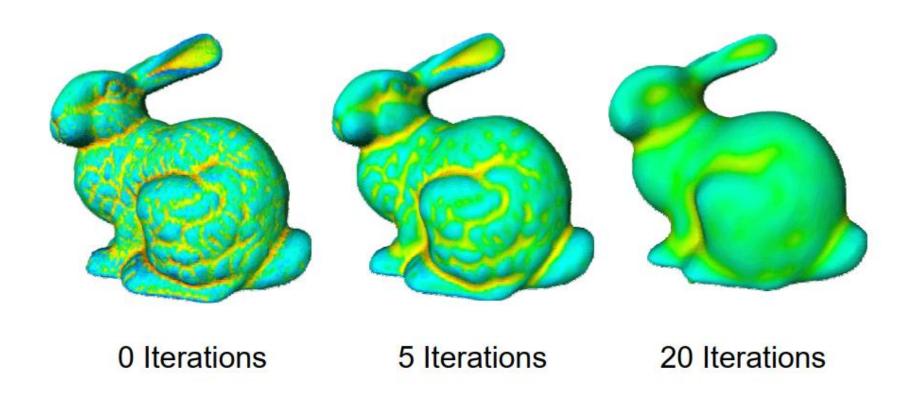
What is $\Delta \mathbf{p}_i$?



$$\frac{1}{2}(\mathbf{p}_{i+1}+\mathbf{p}_{i-1})-\mathbf{p}_i$$

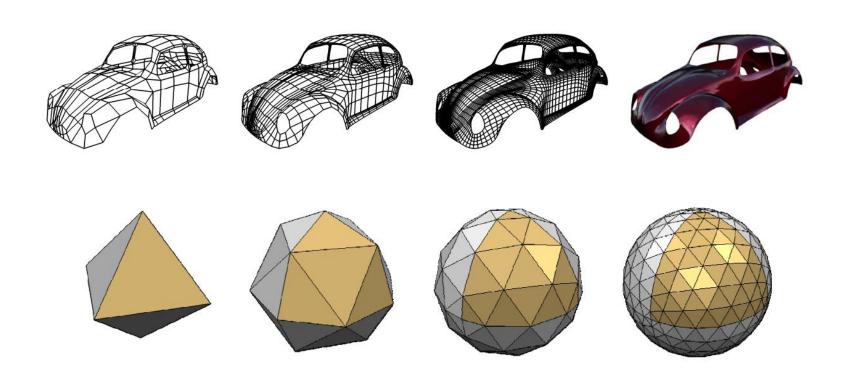


Result of Laplacian Smoothing



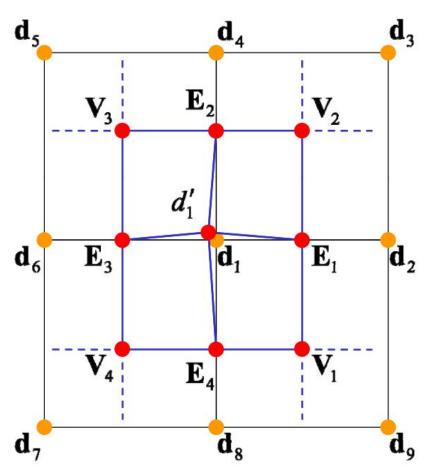
网格细分 (Subdivision)

- 增加更多控制点,引入更多细节
- 往往会改变形状



网格细分

Catmull-Clark Subdivision



$$\mathbf{V}_2 = \frac{1}{n} \times \sum_{j=1}^n \mathbf{d}_j$$

$$\mathbf{E}_i = \frac{1}{4} \left(\mathbf{d}_1 + \mathbf{d}_{2i} + \mathbf{V}_i + \mathbf{V}_{i+1} \right)$$

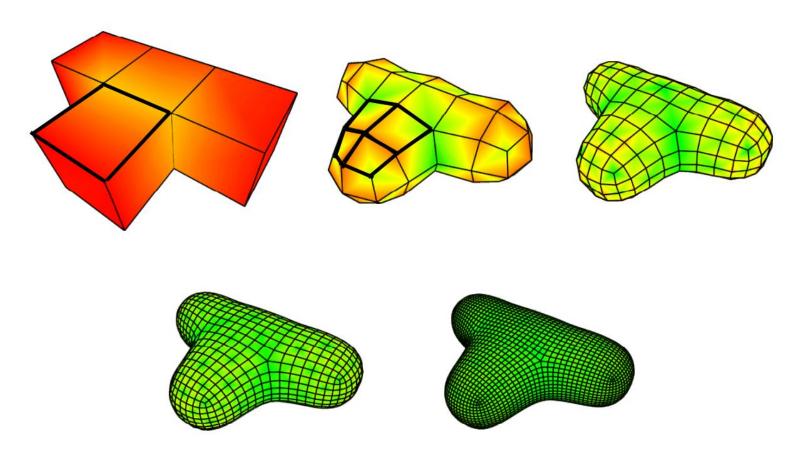
$$\mathbf{d}_{1}' = \frac{(n-3)}{n}\mathbf{d}_{1} + \frac{2}{n}\mathbf{R} + \frac{1}{n}\mathbf{S}$$

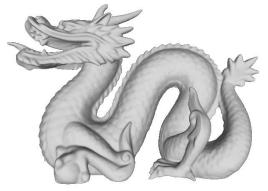
$$\mathbf{R} = \frac{1}{m}\sum_{i=1}^{m}\mathbf{E}_{i} \quad \mathbf{S} = \frac{1}{m}\sum_{i=1}^{m}\mathbf{V}_{i}$$

n is the number of faces containing d1

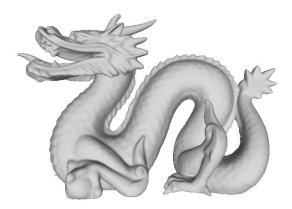
网格细分

Catmull-Clark Subdivision





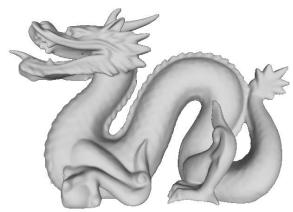
Vertex: 437645 Face: 871414



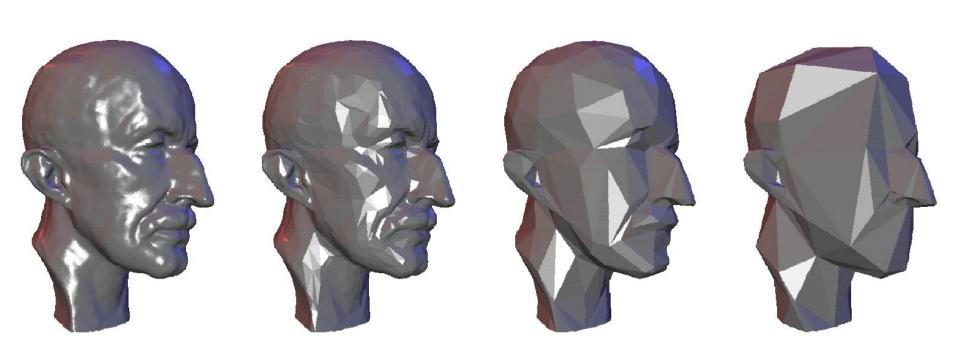
Vertex: 50073 Face: 99999



Vertex: 218121 Face: 435671



Vertex: 25000 Face: 49999



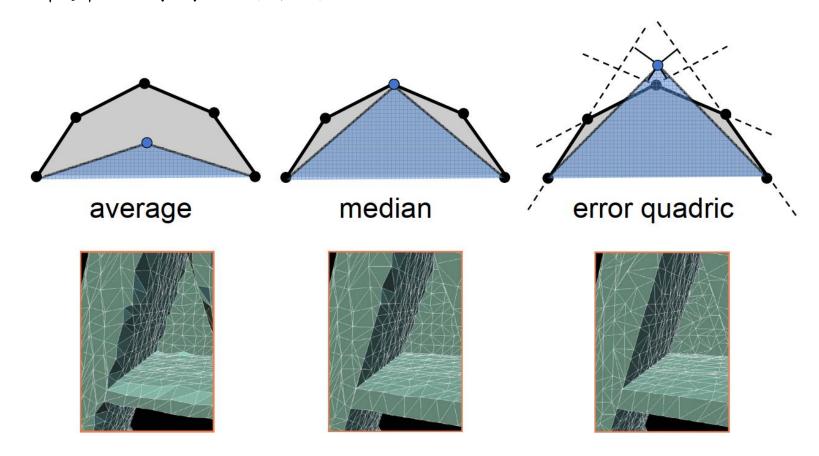
•基本思路:

- 顶点聚类, 把相近的顶点聚成一个代表点
- 计算代表点的位置
- 重新连接边



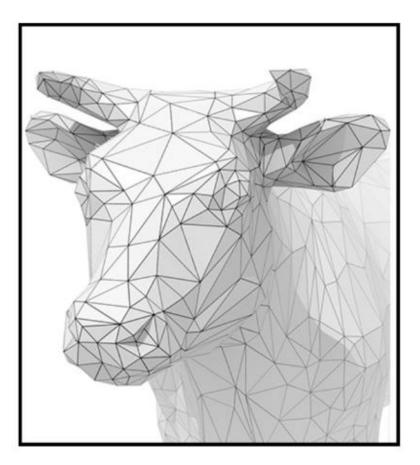


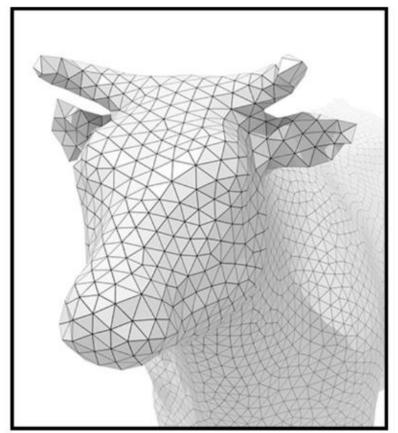
•计算代表点的位置



网格网格正则化 (规范化)

· Mesh Regularization, 改变网格的分布





Meshlab

http://www.meshlab.net/

