3.9 三角网格

- 形体表示方法
  - 线框模型；
  - 实体模型；
  - 表面模型：
    - 基于连续参数曲面的表示方法：样条等；
    - 离散表面表示方法：三角网格。

3.9 三角网格

- 线框模型
  - 只能表示一些简单的模型。
- 实体模型
  - 能够完整地、无歧义地表达三维形状；
  - 不适合物体表面的运算：光照计算、阴影计算和光线求交计算等。
- 连续的表面模型
  - 容易表达工业造型中较光滑的曲线曲面；
  - 不能有效地表达真实世界中的物体，如兔子模型等。

3.9 三角网格

- 三角网格的概念
- 三角网格的半边表示
- 网格处理概述
- 网格简化
- 网格细分
- 特征敏感网格重剖

3.9.1 三角网格的概念

- 三角网格 delaunay
  - 用三角形组成的面片列表近似三维模型；
  - 用于表达物体的原因：
    - 容易通过三维扫描技术大量获取原始数据；
    - 采用足够多的面片时可以任意精度逼近复杂的曲面；
    - 网格模型的数据结构简单、适合于光照计算等处理；
    - 适合硬件进行并行处理。

Velodyne

随着图形硬件不断增强，处理三角网格的性能也以指数级别的速度不断增长，因此已成为最为广泛使用的模型表达方法。

3.9.1 三角网格的概念

三角网格模型

3.9.1 三角网格的概念

- 三角网格描述
  - 顶点几何信息：
    - n个顶点V=(v_1, v_2, …, v_n)。
  - 三角网络拓扑连接的信息
    - 一维的边：对应顶点的二元组, 如(v_1, v_2)；
    - 二维的面：对应顶点的有序三元组（逆时针），如(v_1, v_2, v_3)。
  - 附加属性：
    - 与某顶点、边或面关联的法向、纹理坐标等信息。

3.9.1 三角网格的概念

- 三角网格模型的存储
  - 可通过不同的方法获得：
    - 三维扫描仪，采用特定方法生成三角网格模型；
    - 三维动画和造型软件。
  - 文件格式：
    - 文本：易于阅读和理解；(OBJ和PLY文件格式)
    - 二进制：具有较高的编码效率。

3.9.1 三角网格的概念

- OBJ格式：Alias|Wavefront公司开发的标准3D模型文件格式，适用于3D软件模型之间的互导。
  - # 注解；
  - v 指定顶点；
  - vt 指定纹理坐标:每个三角形面的三个顶点，均需指定一个纹理坐标；
  - vn 指定法线向量：每个三角形面的三个顶点，均需指定一个法向量；
  - f 指定表面：面的顶点、纹理坐标、法向量的索引。

    例子：obj文件描述了一个立方体 。

3.9.1 三角网格的概念

- PLY格式(Polygon File Format)：

  90年代中期，由斯坦福大学CG实验室的Marc Levoy，Greg Turk等人开发出来（Stanford Triangle Format）。

  - 代表：Stanford Bunny, Dragon, Happy Buddha。
  - 典型的PLY文件结构：

    头部
    顶点列表
    面片列表
    （其他元素列表）

3.9.1 三角网格的概念

- 头文件：

```
ply                          { ply格式标识}
format ascii 1.0  {  ascii/binary，格式版本号  }
comment made by anonymous  {  注释关键词说明，像其他行一样  }
element vertex 8  {  定义“vertex”（顶点）元素，在文件中有8个  }
property float32 x  {  顶点包含浮点坐标“x”}
property float32 y  {  y  坐标同样是一个顶点属性  }
property float32 z  {  z  也是坐标  }
element face 6  {  在文件里有6个“face”（面片）  }
property list uint8 vertex_index { 面顶点索引属性，类型为uint8的列表 }
end_header  {  划定头部结尾  }
```

- 顶点/面片：见文件！

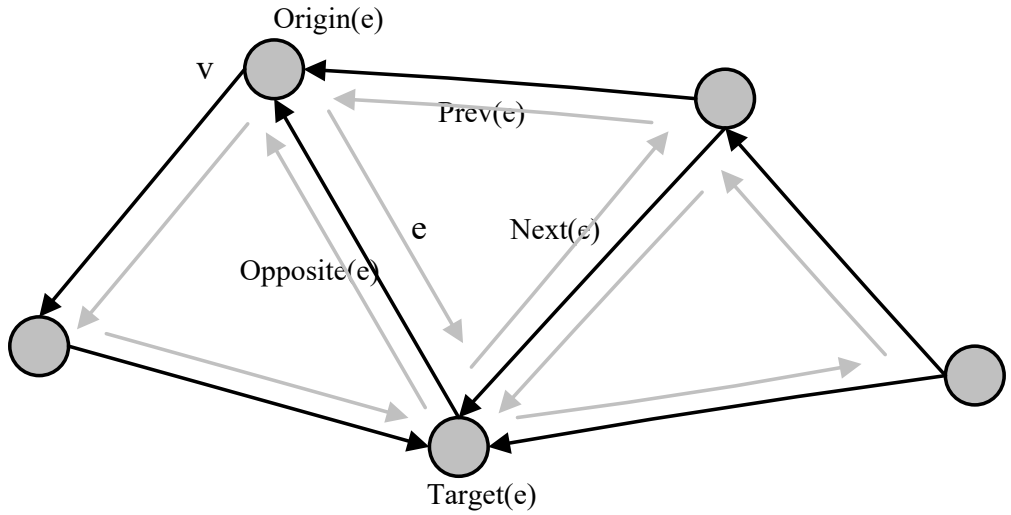
3.9.2 三角网格的半边表示

- 精确的表示三角网格：
  - 只需存储顶点和每个面所包含的顶点。
- 对网格拓扑进行检索和遍历：

  按顺时针顺序遍历包含某顶点的所有面。
  - **原始的存储结构**：需要遍历整个模型；
  - **改进的存储结构**：增加少量的存储量，提供一系列访问的便利，例如三角网格的半边表示结构。

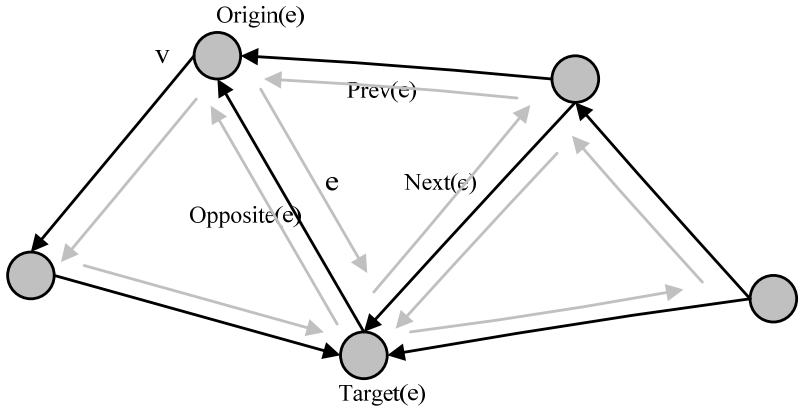
3.9.2 三角网格的半边表示

- 半边结构（双向链接表）：把一条无向的边拆分成两条有向的“半边”，规定半边的方向总是沿着逆时针方向。
  - 半边e和半边Opposite(e)对应同一条边；
  - 除顶点、边和面外，存储额外信息。



3.9.2 三角网格的半边表示

- 存储的信息
  - 顶点的几何信息，即空间坐标(v)；
  - 一条从该顶点出发的半边(e)：
    - 该半边的源顶点Origin(e)
    - 该半边在同一三角形中的下一半边Next(e)
    - 与该半边同属一条边的对边Opposite(e)
    - 该半边所属面IncFace(e)



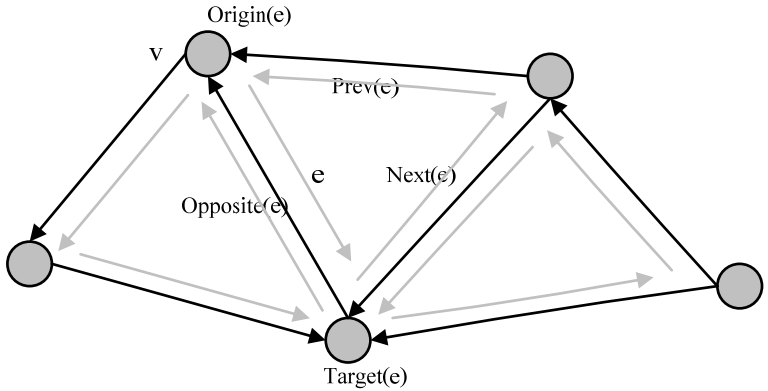
3.9.2 三角网格的半边表示

- 例子：此存储结构便于各种网格上的遍历操作。
  - 获得半边e的上一半边：

    Prev(e)= Next(Next(e))。
  - 半边e指向的顶点：

    Target(e)= Origin(Next(e))。
  - 获取与当前面相邻的其它面：

    Next(e)+Opposite()+IncFace()
  - 枚举顶点v周围的所有面：
    - 获得v出发的一条半边e, IncFace(e)；
    - Opposite(e)+IncFace()；
    - Prev/Next(e)+Opposite()+IncFace()。



3.9.2 三角网格的半边表示

- 半边表示结构：
  - 需要存储一些额外的信息，当网格结构发生变化时，需要进行相应的更新，以保证数据的一致性。
  - 可为网格上的访问操作提供便利，并提高处理效率。

3.9.3 网格处理概述

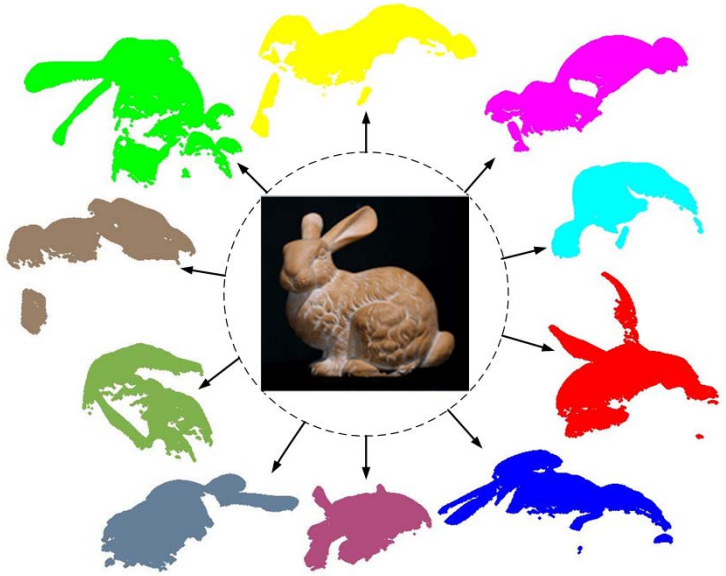
- **背景**：出现大量高精度的三角网格模型，为了获得符合人们需要的数据，需要进行相关处理。
- 从应用目标出发，网格处理包括
  - **简化**：用较少的面片表示几何，提高绘制效率；
  - **细分**：以原始网格为基础，按一定规则生成包含更多面片的几何；
  - **重剖**：为了获得更规则的网格模型，可能具有更少或更多的面片；
  - **光顺**：与原始网格基本一致，但更光顺，以去除不需要的几何细节或噪声；

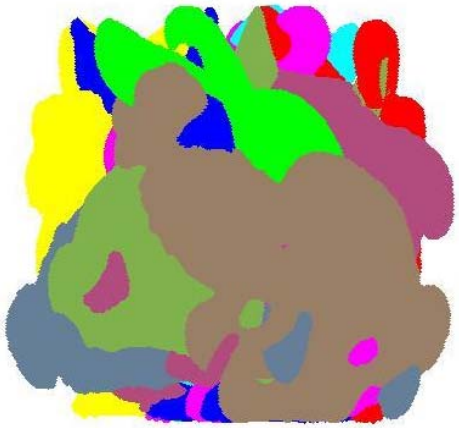
3.9.3 网格处理概述

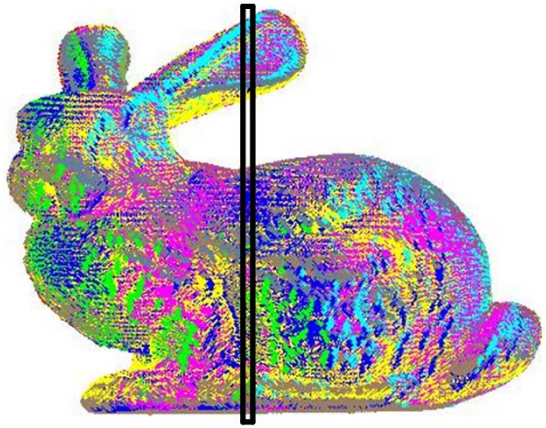
- **参数化**：将网格映射到更简单的参数域上；
- **网格修复**
  - 由于数据采集和配准存在误差，使得重构算法生成的网格：
    - **几何缺陷**：网格表面含有自相交、面片重合等几何位置缺失或重叠；
    - **拓扑缺陷**：重构的网格经常含有环、孤岛和空洞、极大地影响了网格简化、参数化等处理算法。
  - 而网格处理的算法效果很大程度上依赖于模型的质量；
  - 可采用基于网格面或基于体数据的方法进行处理。

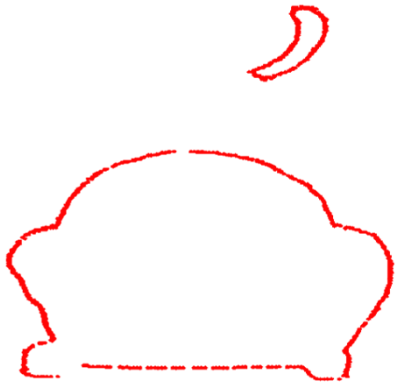
3.9.3 网格处理概述

- 配准









3.9.3 网格处理概述

- **网格变形**：在不改变网络拓扑连接关系的同时改变网格顶点的坐标，使原来网络模型的姿态发生变化。
- **网格分析**
  - 网格中三角形重要度不一致（尖锐和光滑的边沿区域）；
  - 对特征区域进行有效分析，可改进处理算法的效率和效果，且有助于对模型的理解。
- **网格分割**：将模型分解成若干有意义的部分，可视作网格分析的基本处理工具。
- **模型检索**：根据模型的内在特征衡量模型之间的相似性，进而在模型库中检索与输入模型相似的模型。

3.9.4 网格简化

- 定义：减少已有网格的面片数量，仍能表达原三维模型的过程
  - 在当前的真实感图形学中，需要绘制的三维场景包含大量多边形，不易实时绘制：
    - 复杂的多面体网格由Lidar扫描真实三维物体而得到；
    - 为真实反映原物体的表面变化，所采样点非常稠密，导致**存储、传输及绘制**带来极大地困难。
  - 实时生成真实感图像过程中，得到特定的视觉效果：
    - 选择合适的图形算法；
    - **通过减少场景的复杂度提高图形绘制的速度**。
  - 方法：层次细节网格简化技术 Level of Detail

3.9.4 网格简化

- 层次细节网格简化技术(LOD)
  - 为简化采样密集的多面体网格物体而设计的算法：
    - 在基本不影响画面视觉效果的条件下，通过逐次简化景物的表面细节来减少场景的几何复杂性，从而提高绘制算法的效率。
  - **实质**：通过降低显示三维场景模型的复杂度，损失一定的图形质量，达到实时绘制真实感图象的目的。

3.9.4 网格简化

- LOD技术的基本原理：
  - 对一个原始多面体模型，建立几个不同逼近程度的几何模型：
    - 每个模型均保留一定层次的细节；
    - 从近处观察物体时，采用精细的模型；
    - 从远处观察物体时，采用较粗糙的模型；
    - 运动或运动较快的物体，用较粗的模型表示；
    - 静止或运动较慢的物体，用较精细的模型表示。

3.9.4 网格简化

- 研究的主要问题
  - 如何建立原始网格模型的不同层次细节的模型；
  - 如何建立相邻层次的多边形网格模型之间的**几何形状过渡**：
    - 当视点连续变化时，在两个不同层次的模型之间存在明显的跳跃，必须在相邻层次的模型之间形成光滑的视觉过渡，使生成的真实感图象序列是视觉光滑的。

3.9.4 网格简化

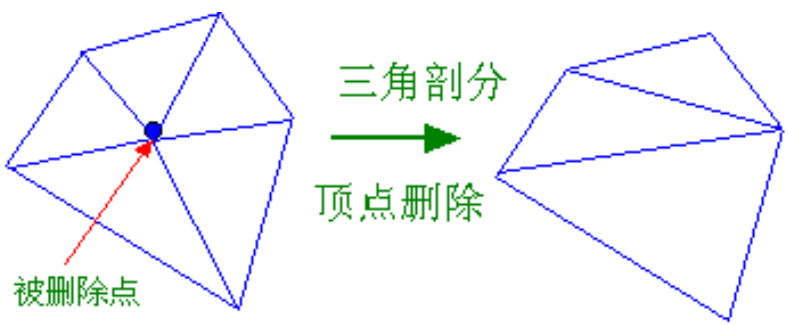
- LOD算法
  - 生成不同细节的LOD模型
    - 从原始精细模型中删除重要性小的点、边和面，简化网格；
    - 相对于原始网格，它们之间的误差是逐步递增的。
  - 基于某种原则选取某个LOD模型
    - 视点、视线相关的。
  - 实现几何形状光滑过渡
    - 建立相邻层次的多边形网格模型之间的几何形状过渡：采用某种方法从一种LOD模型切换到另一种模型；
    - 插值对应网格基本元素的位置，实现平滑过渡，避免突变。

3.9.4 网格简化

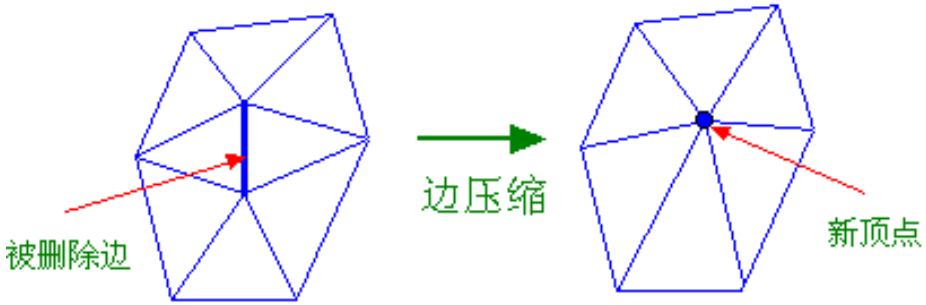
- 如何建立原始网格模型的不同层次细节的模型
  - 假设场景的模型都是三角形网格
  - LOD模型的化简操作
    - 顶点删除操作；
    - 边压缩操作；
    - 面片收缩操作。

3.9.4 网格简化

- 顶点删除操作
  - 删除网格中的一个顶点，然后对它的相邻三角形形成的空洞作三角剖分，以保持网格的拓扑一致性。
- 边压缩操作
  - 把网格上的一条边压缩为顶点，与该边相邻的两个三角形的退化，两个顶点融合为一个新的顶点。



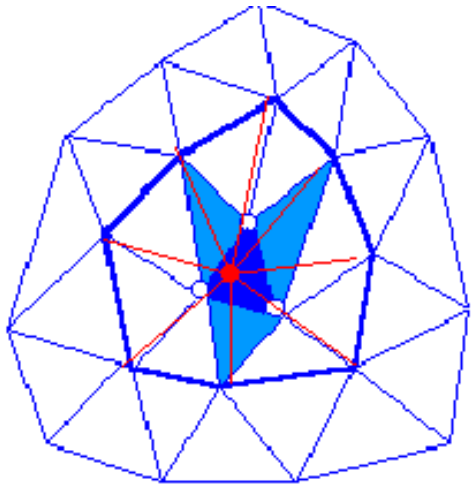
(A) 顶点删除操作



(B) 边压缩操作

3.9.4 网格简化

- 面片收缩操作
  - 把网格上面片收缩为顶点，该三角形本身和与其相邻三个三角形都退化，三个顶点收缩为一个新顶点。



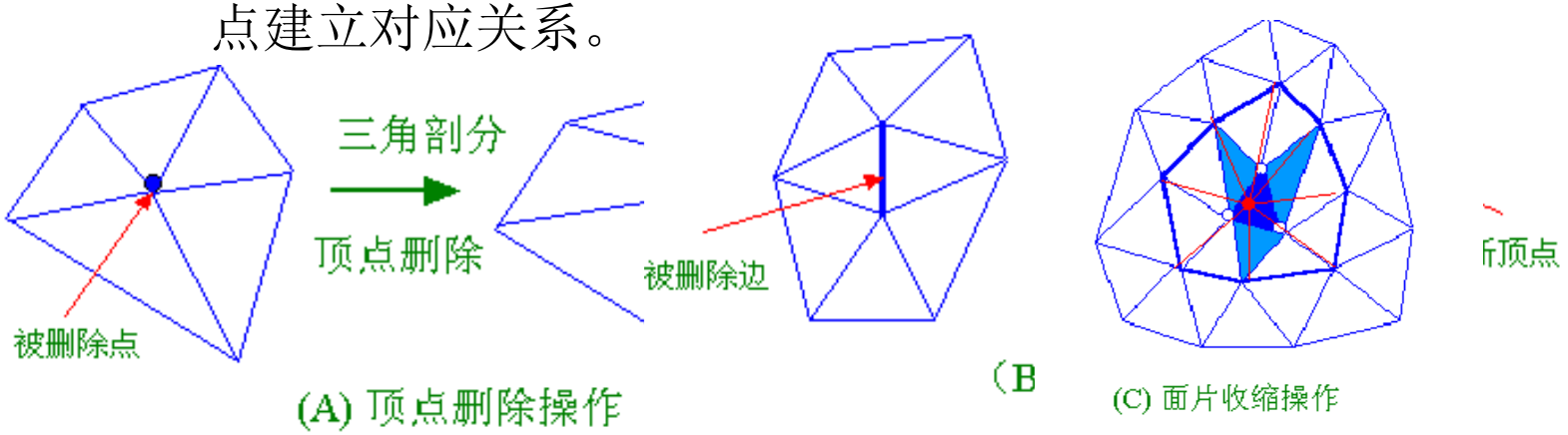
(C) 面片收缩操作

3.9.4 网格简化

- LOD模型生成
  - 基于上述操作，确定每次操作给网格场景带来的误差，用此误差代替原始网格上的每个基本元素的误差作为权值，插入到按权值增序排列的队列中；
  - 开始循环进行网格基本简化操作：
    - 选取队首权值最小的操作，执行之；
    - 更新变化的网格信息，并重新计算改变了的网格基本元素的误差，插入到队列；
    - 再开始下一个循环，直到队列的最小误差达到用户设定的阀值或者用户希望的化简网格数目已经达到。

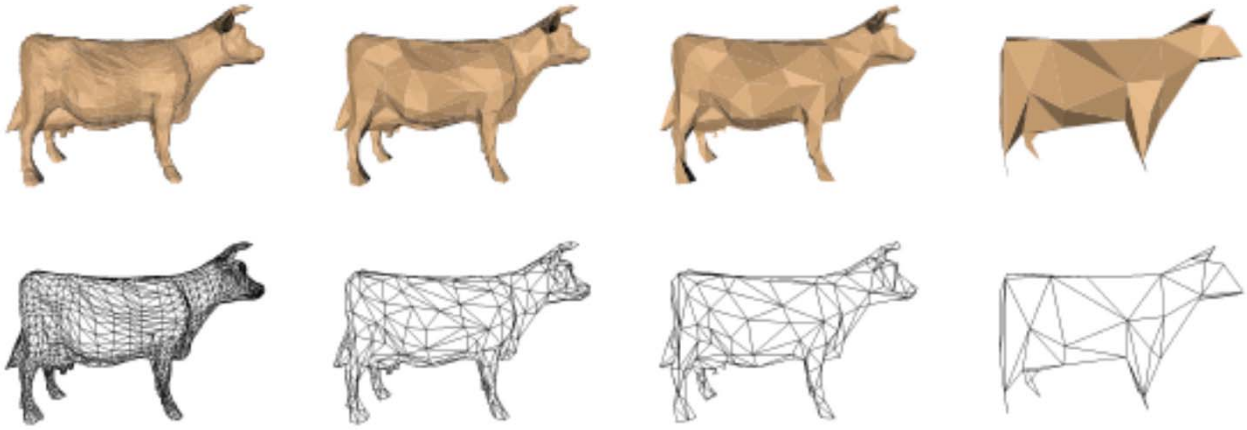
3.9.4 网格简化

- 几何形状过渡
  - 通过插值对应网格基本元素的位置来实现光滑过渡；
  - 如何得到两个相邻层次的多边形网格模型的基本元素之间的对应关系：
    - 顶点删除和面片收缩操作：用被操作的对象与其相邻的基本元素之间建立对应关系；
    - 边压缩操作：只要简单的把压缩边上的两点与压缩后的新点建立对应关系。



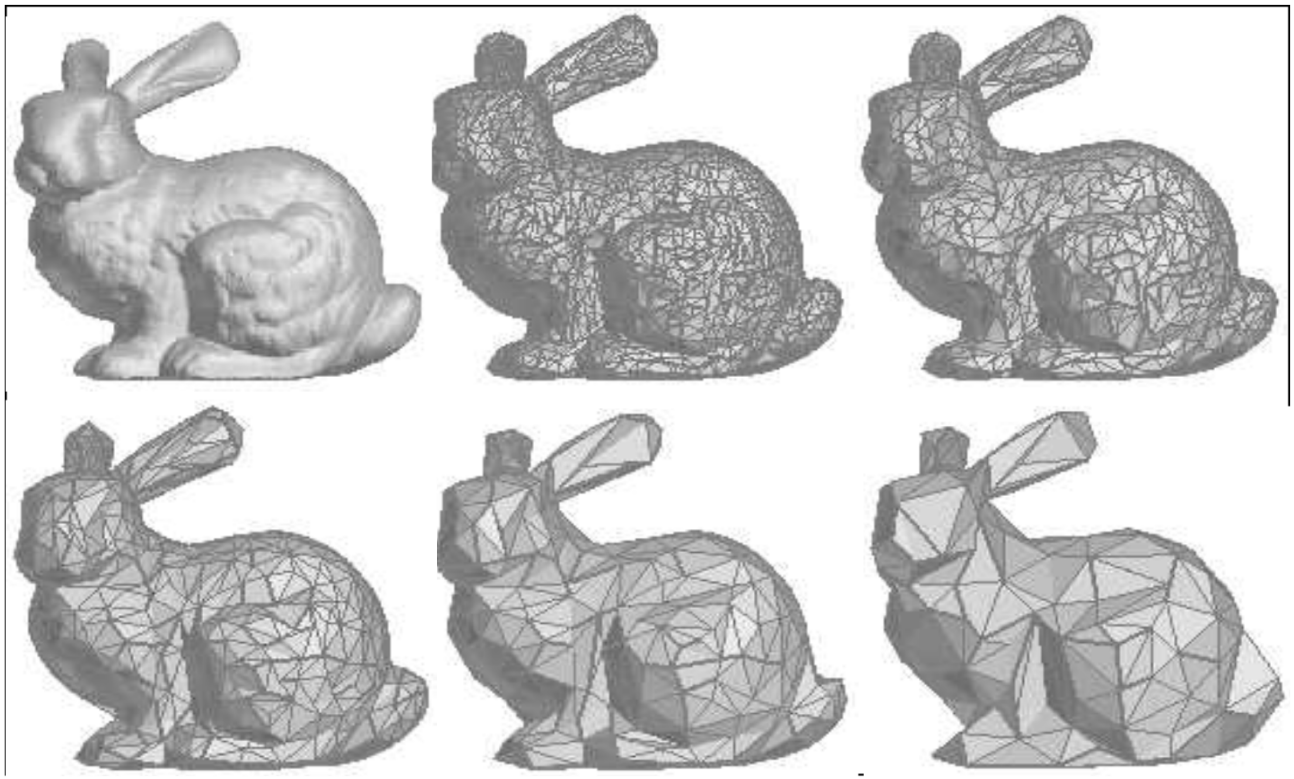
3.9.4 网格简化

- LOD技术：
  - 广泛地应用于实时真实感图形学中；
  - 简化场景复杂度，满足某些关键任务的实时性要求：
    - 采用不同分辨率的模型显示复杂场景的不同物体，保证真实感图象质量满足要求的前提下，实时地产生真实感图象。



牛模型的层次细节简化模型

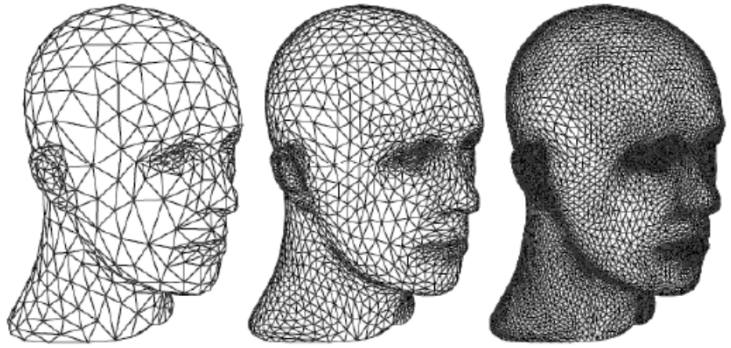
3.9.4 网格简化



兔子的层次细节简化模型

3.9.5 网格细分

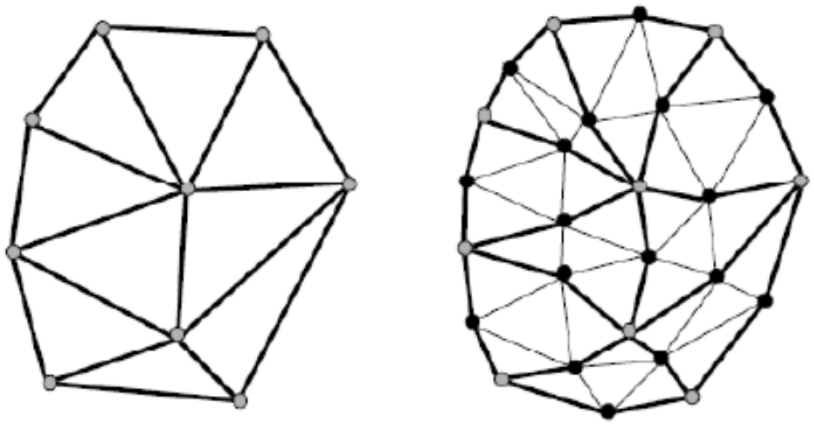
- 网格细分：通过按一定规则给网格增加顶点和面片数量，让网格模型变得更加光滑。
  - Loop细分法；
  - $\sqrt{3}$ 细分法；
  - Butterfly细分。



细分示意图

3.9.5 网格细分

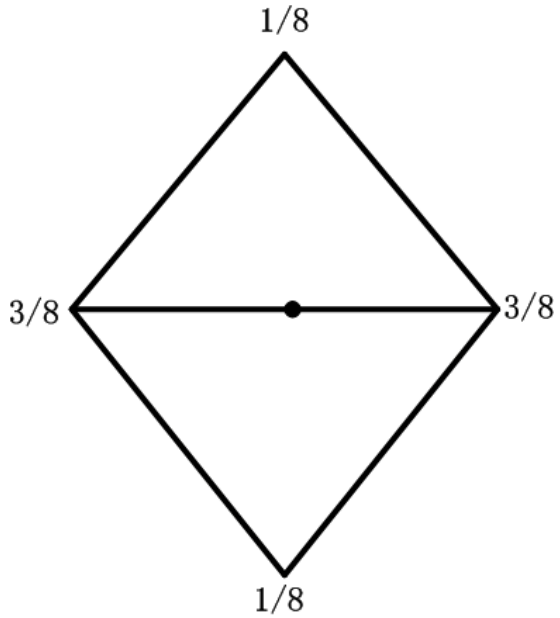
- Loop细分法
  - 最早的基于三角网络的细分法；
  - 步骤：
    - 增加顶点：调增网格的拓扑结构；
    - 顶点位置调整：平滑网格的粗糙程度。
  - 生成两类顶点，E-顶点和V-顶点。



3.9.5 网格细分

- Loop细分法
  - 如内部边有两个顶点v_0和v_1，共享此边的两个三角形为（v_0, v_1, v_2）和（v_0, v_1, v_3），则新生成的E顶点为:

$$v_E = \frac{3}{8}(v_0 + v_1) + \frac{1}{8}(v_2 + v_3)$$



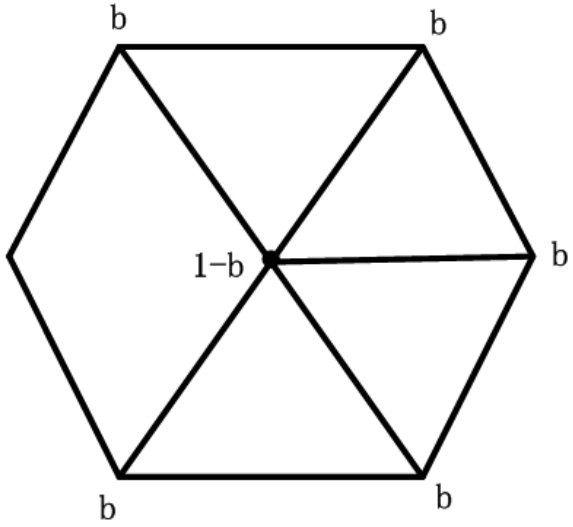
3.9.5 网格细分

- Loop细分法（续）：
  - 若内部顶点v的1-邻域顶点为v_i(i=1, 2, …, n)，则新生的V顶点为：

$$v_V = (1-n\beta)v + \beta(v_1 + v_2 + ... + v_n)$$

即是顶点本身与其所有相邻顶点的加权和，它本身的权值为$(1-n\beta)$，而邻点权值为：

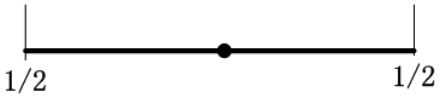
$$\beta = \frac{1}{n}\left(\frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{n}\right)^2\right)$$



3.9.5 网格细分

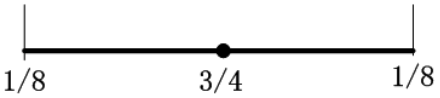
- Loop细分法（续）：
  - 若边界边的两个顶点为v_0和v_1，则新生成的E-顶点为：

$$v_E = \frac{1}{2}(v_0 + v_1)$$



  - 若边界顶点v在边界上的两相邻顶点为v_0和v_1，则新生成的V-顶点为：

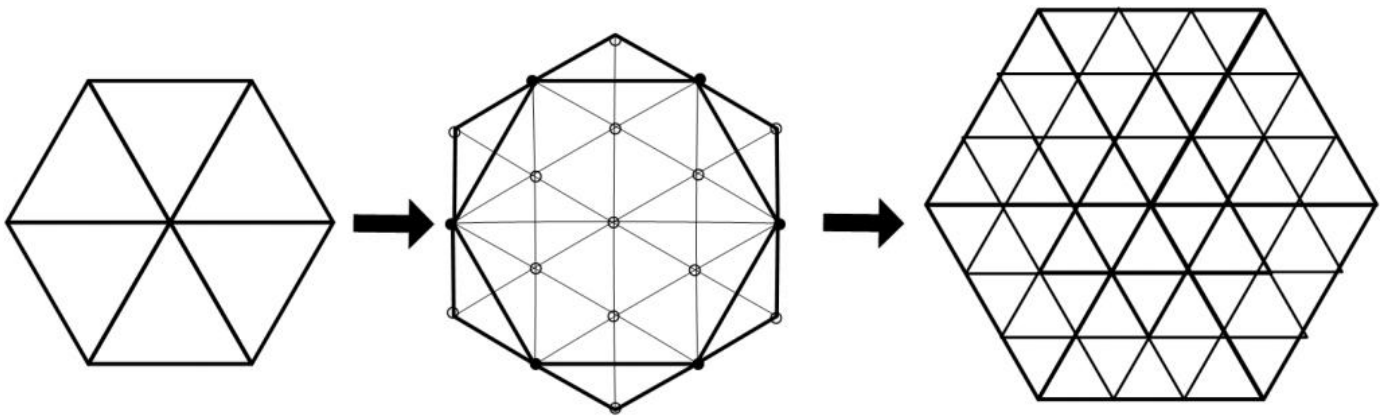
$$v_V = \frac{1}{8}v_0 + \frac{1}{8}v_1 + \frac{3}{4}v$$



3.9.5 网格细分

- $\sqrt{3}$细分法：

  在每次细分过程中，在每个三角形几何中心增加一个顶点，并调整原有顶点的坐标值，从而使得一个三角形在每次细化过程中将变成三个。

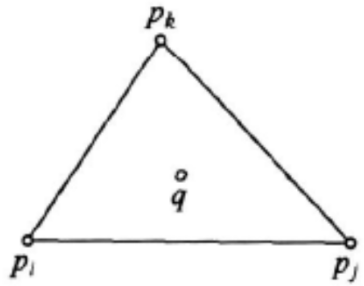


根号3细分法过程图

3.9.5 网格细分

- $\sqrt{3}$ 细分法：
  - 对于每一个面，计算机一个新面点：

$$q = \frac{p_i + p_j + p_k}{3}$$



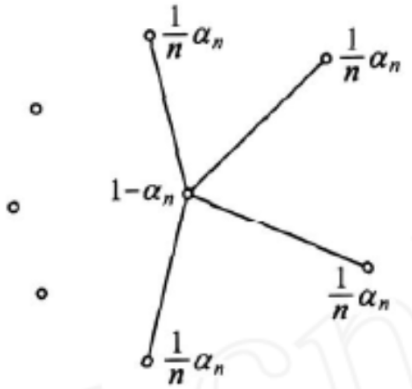
a 新面点

  - 对于每一个顶点，计算一个新顶点：

$$p' = (1-\alpha_n)p + \frac{\alpha_n}{n}\sum_{i=1}^{n} p_i$$

其中

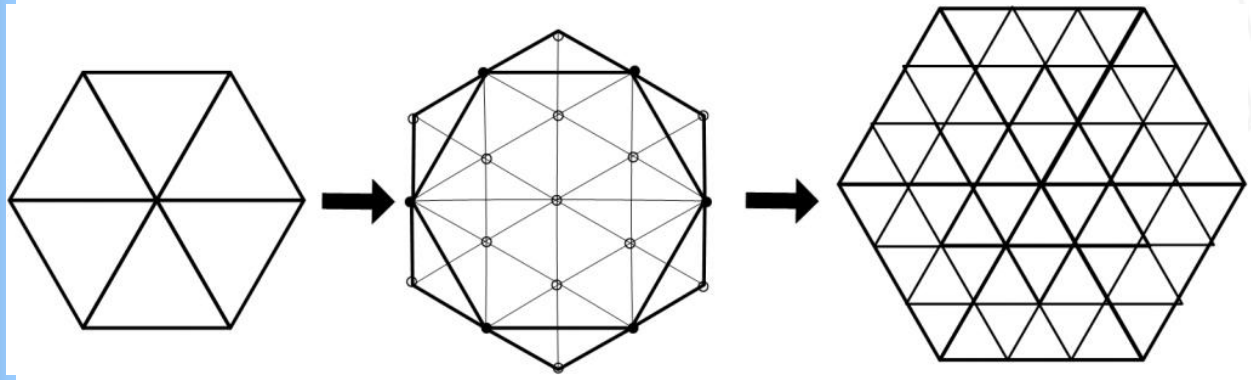
$$\alpha_n = \frac{4 - 2\cos(2\pi / n)}{9n}$$

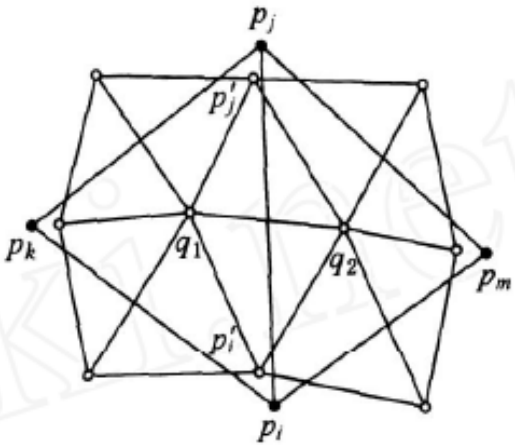


b 新顶点

3.9.5 网格细分

- $\sqrt{3}$ 细分法：
  - 对于每一个新面点，与其所在平面的3个顶点对应的3个新顶点连成3条边，与其有公共边的3个面对应的新面点连成3条边；
  - 每3个互连的点组成了如图c所示的新面。

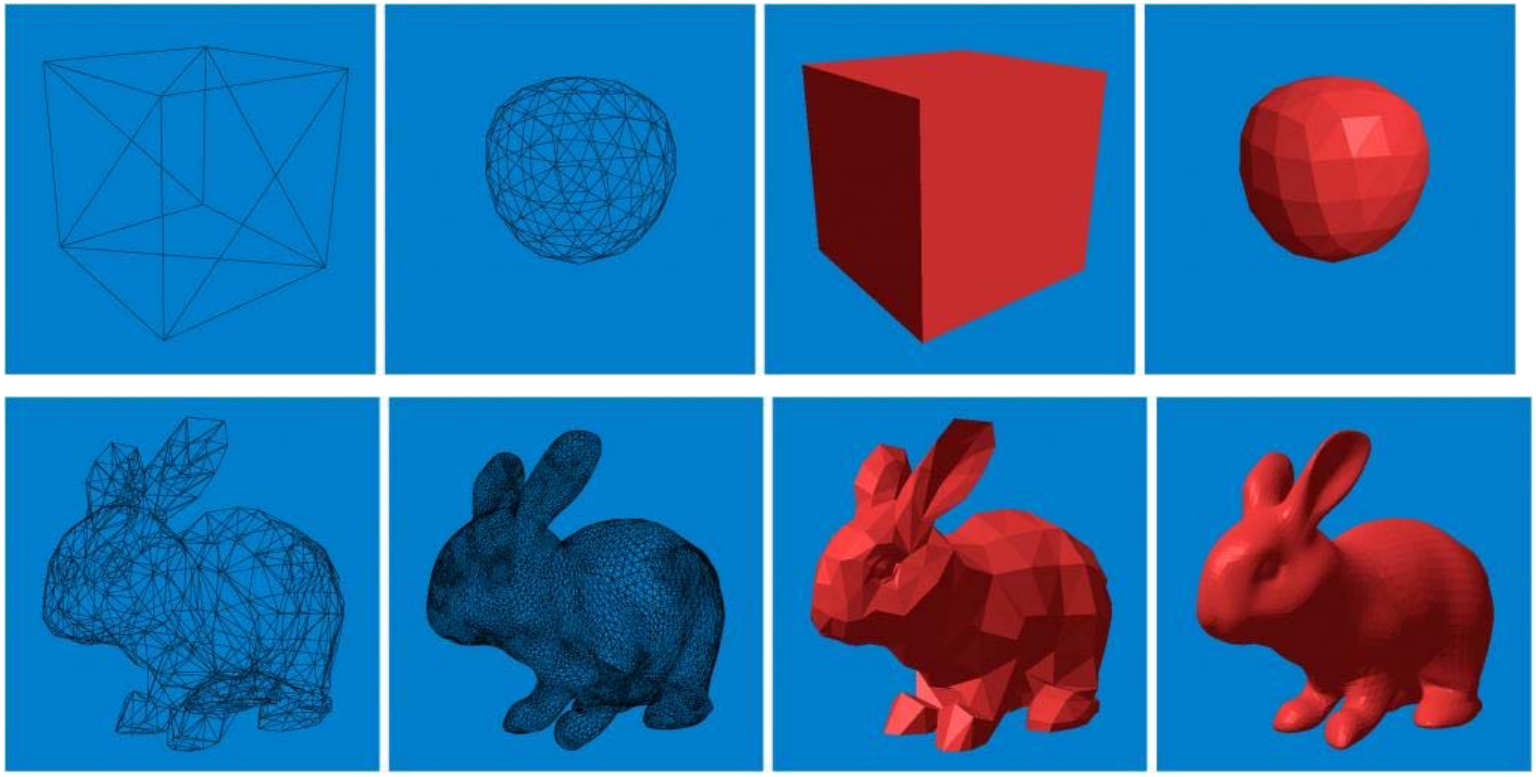




c 拓扑连接关系

3.9.5 网格细分

- 模型细分的一些示例图：



3.9.6 特征敏感网格重剖

- 特征敏感网格重剖
  - 在网格简化后的模型中，三角形形状的**规则性**难以有较好的保证；
  - 三角网格模型的规则性指标：
    - 网格中顶点的度数尽可能接近于6；
    - 每个三角面的顶角尽可能的接近于60度；
    - 构成网格的各边的边长尽可能相近。
  - **网格重剖**：生成较规则的网格模型，并尽可能与原网格在几何上相近。

3.9.6 特征敏感网格重剖

- 特征敏感度量
  - **特阵区域**：具有至少一个较大主曲率的区域
    - 对于几何模型的外观及表达尤为重要；
    - 特征区域曲面上单位法向发生变化：尖锐特征，平滑特征，平坦区域。
  - 特阵敏感度量：通过改变度量，为与特阵相关的网格处理提供一种统一的解决方案
    - 欧氏度量；
    - 马氏距离。

3.9.6 特征敏感网格重剖

- 特阵敏感距离
  - 把网格视作连续曲面的近似，设曲面 Φ 上一点 $x \in \Phi$ 处的单位法向量为 $n(x)$，可以将曲面的上任意一点 x 映射到空间 R^6 中的一点 $x_f = (x, wn)$，其中 w 是一个非负的权重；
  - 假设网格是连续曲面足够好的近似，那么网格上两个顶点 v_1 和 v_2，坐标和法向量分别是 (x_1, n_1) 和 (x_2, n_2)，则特征敏感度量下的距离：

$$fs\_dist(v_1, v_2) = \sqrt{\|x_1 - x_2\|^2 + w^2 \|n_1 - n_2\|^2}$$

3.9.6 特征敏感网格重剖

– 特阵敏感距离

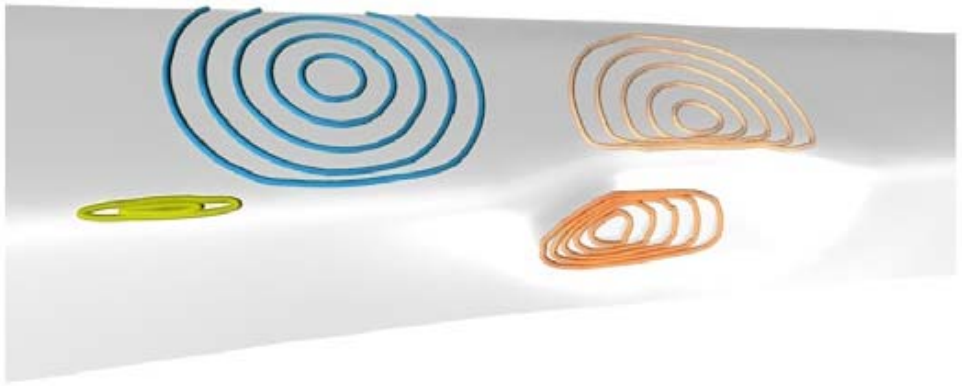
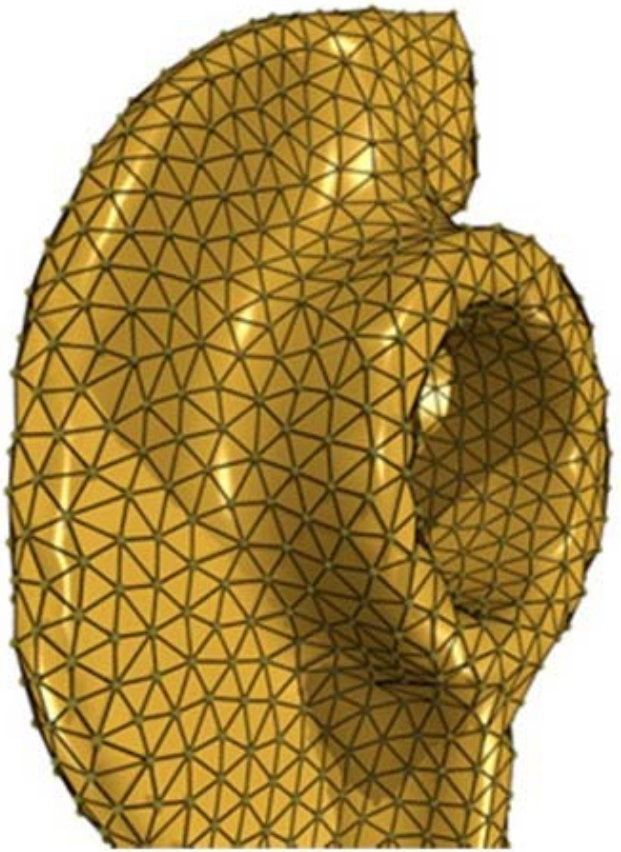


图3.70 特征敏感度量下到中心点等距离点的连线

- 很多网格处理算法或多或少依赖于度量：网格重剖、特阵分析和编辑等；
- 与欧氏度量相比，综合考虑了位置和法向的变化，有助于更好地处理和保持特征。

3.9.6 特征敏感网格重剖

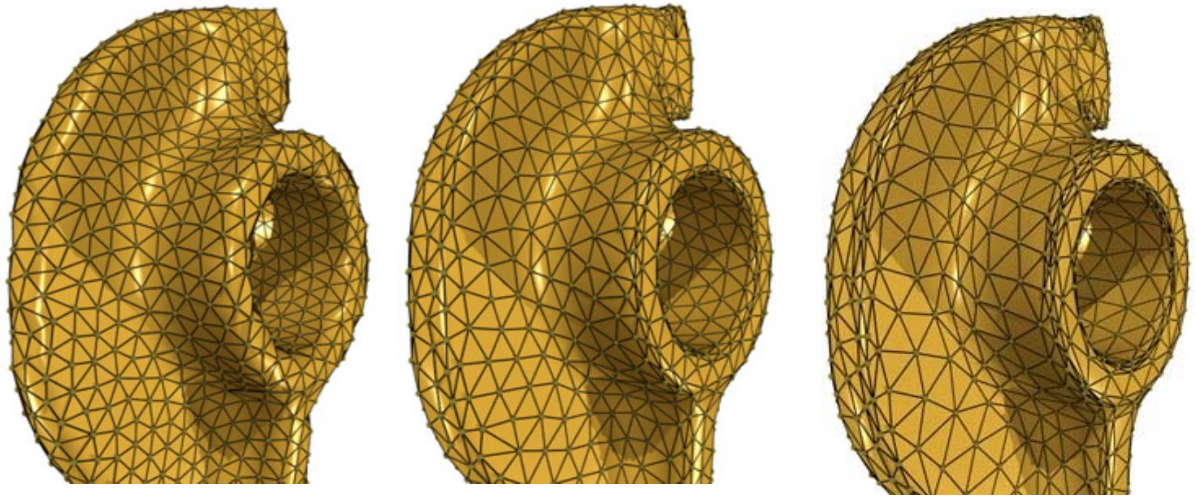
- 例子：特征敏感重剖
  - 各向同性重剖：生成尽可能均匀的模型：
    - 首先调整一系列采样点的位置，使他们在模型上均匀分布，然后利用局部参数化将局部区域映射到平面上，并使用带约束的Delaunay参数化来恢复网格的连接关系；
    - 虽然三角形形状比较好，但是特征区域由于和其它区域一样进行采样，因而变得模糊。



特征敏感网格重剖

特征敏感网格重剖

- 基于特征敏感度量的重剖：
  - 调整 w 的取值，可以改变生成网格对于特征的保持程度；
  - 在特征敏感度量下仍是均匀的，虽然在三维空间中看起来，三角形的形状沿特征方向拉伸，但三角形的度仍然接近6，且改进了特征区域的保持，用较少的面片比较精确地表示几何模型。



特征敏感网格重剖，从左到右w=0,0.1,0.2