计算机图形学基础

胡事民

清华大学计算机科学与技术系

网格（Mesh）细分与网格简化

- 网格相关基本概念
- 网格细分与细分曲面
- 网格简化

网格相关基本概念

- 网格的描述
  - 一系列的面片$F = (f_1, f_2, \ldots, f_n)$
    - 每一个面片都是三角形
  - 一系列的顶点$V = (v_1, v_2, \ldots, v_n)$
  - F中的每个面片是V中顶点的序列组

    e.g.

    $f_1 : (v_1, v_2, v_3)$, $f_2 : (v_4, v_5, v_6)$,

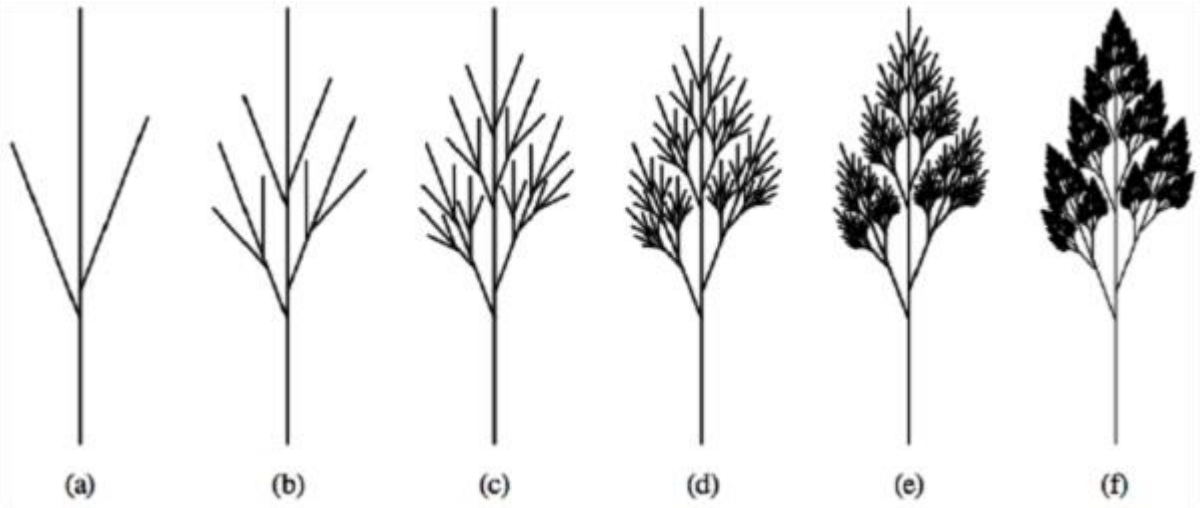
    $f_3 : (v_7, v_8, v_9)$, …

网格表示的由来

- 计算机产生的三维模型和实际获取数据的表示模式不同
  - 图形学中需要一个统一的表示方式
  - 视觉精度和处理速度需要在可以接受的范围之内
  - 由于图形硬件的快速发展, 我们已经能够快速地进行光栅化（rasterize）和渲染（ render ）基于三角网格表示的模型

三维数据的来源

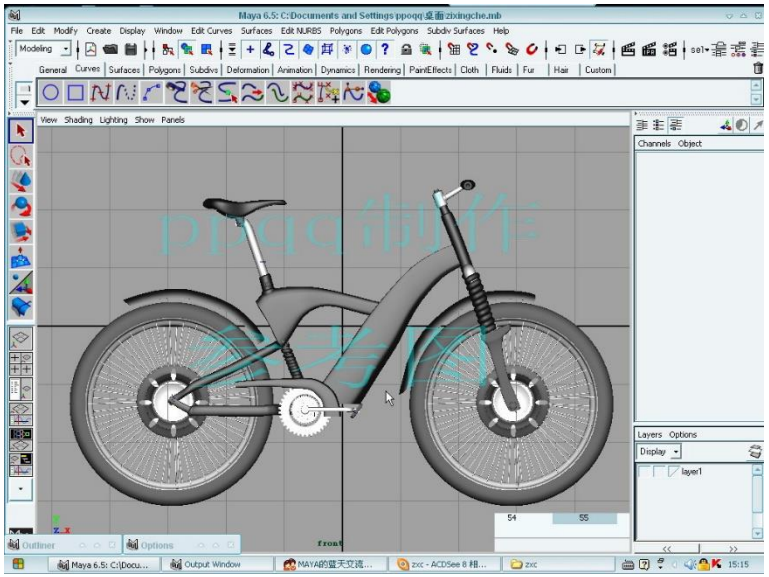
- **模型产生途径：**
  - 直接在几何文件中输入
  - 程序建模（procedure modeling）：通过程序代码进行创建, 比如L-系统，分形几何



三维数据的来源

- **模型产生途径：**
  - 直接在几何文件中输入
  - 程序建模（procedure modeling）：通过程序代码进行创建, 比如L-系统，分形几何
  - 运用建模软件创建，

    比如3DS MAX / MAYA

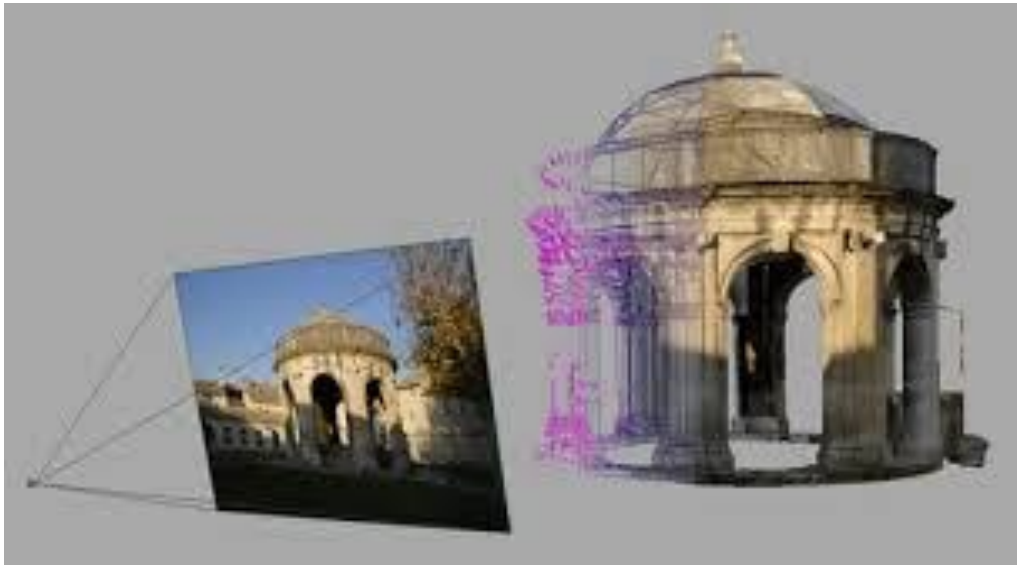


三维数据的来源

- **模型产生途径：**
  - 直接在几何文件中输入
  - 程序建模（procedure modeling）：通过程序代码进行创建, 比如L-系统，分形几何
  - 运用建模软件创建，比如3DS MAX / MAYA
  - 使用3D扫描仪（3D Scanner）获取真实模型

三维数据的来源

- **模型产生途径：**
  - 照片测量法 (photogrammetry)：基于照片进行三维重构(Reconstruction)，例如传统视觉方法



三维数据的来源

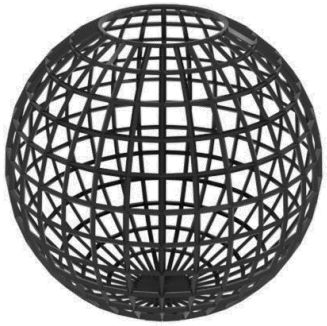
- **模型产生途径：**
  - 照片测量法(photogrammetry)：基于照片进行三维重构(Reconstruction)，例如传统视觉方法
  - 基于RGBD数据：微软Kinect
  - 各类方法的组合

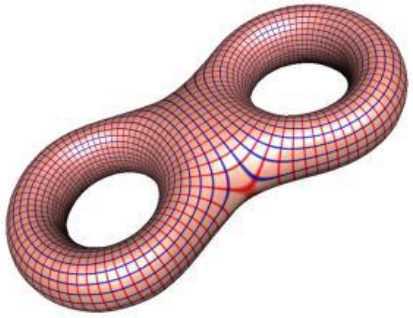
网格化 (Mesh Tessellation)

- 2D网格化
  - 2D网格化在三维网格处理中有重要应用
  - 多边形可以表示成多种不同的形式并且需要分割成更易于处理的图元，比如凸多边形，三角形或者四边形，这种表示过程称之为网格化(Mesh Tessellation)；其中，多边形分割成三角形的形式叫做三角化（ triangulation ）

拓扑结构(Topology)

- 拓扑结构(Topology)
  - 连通多边形网格的结构
  - 亏格(genus)
    - 网格表面孔洞的数目
    - 例如， 一个球体或者立方体的亏格数目为零，而一个环或者咖啡壶的亏格数目是1。
  - 面片、边或顶点的局部拓扑
    - 特征邻域的连通性





二维流形（manifold）

- 拓扑结构(Topology)
  - 二维流形（manifold）
    - 如果局部拓扑处处等价于一个圆盘 (同胚)
    - 在三角网格流形拓扑中, 每一条边属于且仅属于两个三角形，每个三角形分别与三个相邻的三角形有一条共边
  - 带边界的二维流形
    - 确保边界仅属于一个三角形

网格细分

细分(Subdivision)

- 细分(Subdivision)
  - Catmull and Clark, Doo and Sabin 在1978发表的论文标志着模型表面细分的开始
  - 现在，细分已大量应用于电影作品中
  - 细分的简略描述
    - 对一个给定的原始网格进行精细地改进，从而产生更平滑的效果
  - 细分人物：

- 细分人物谱
  - Edwin Catmull： Pixar & Disney, Coons奖
    2020年3月18日获图灵奖。 Catmull-Clark细分曲面
  - Jim Clark:
    - Silicon Graphics

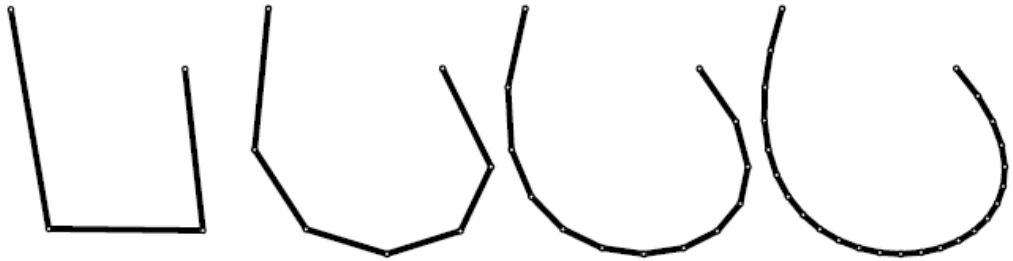
公众号：图形学与几何计算

清华大学图形学实验室

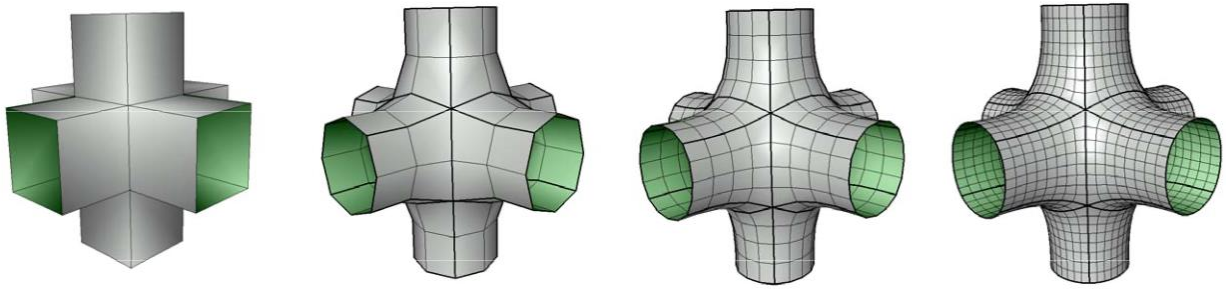
- Malcolm Sabin
  Doo-Sabin曲面

细分(Subdivision)

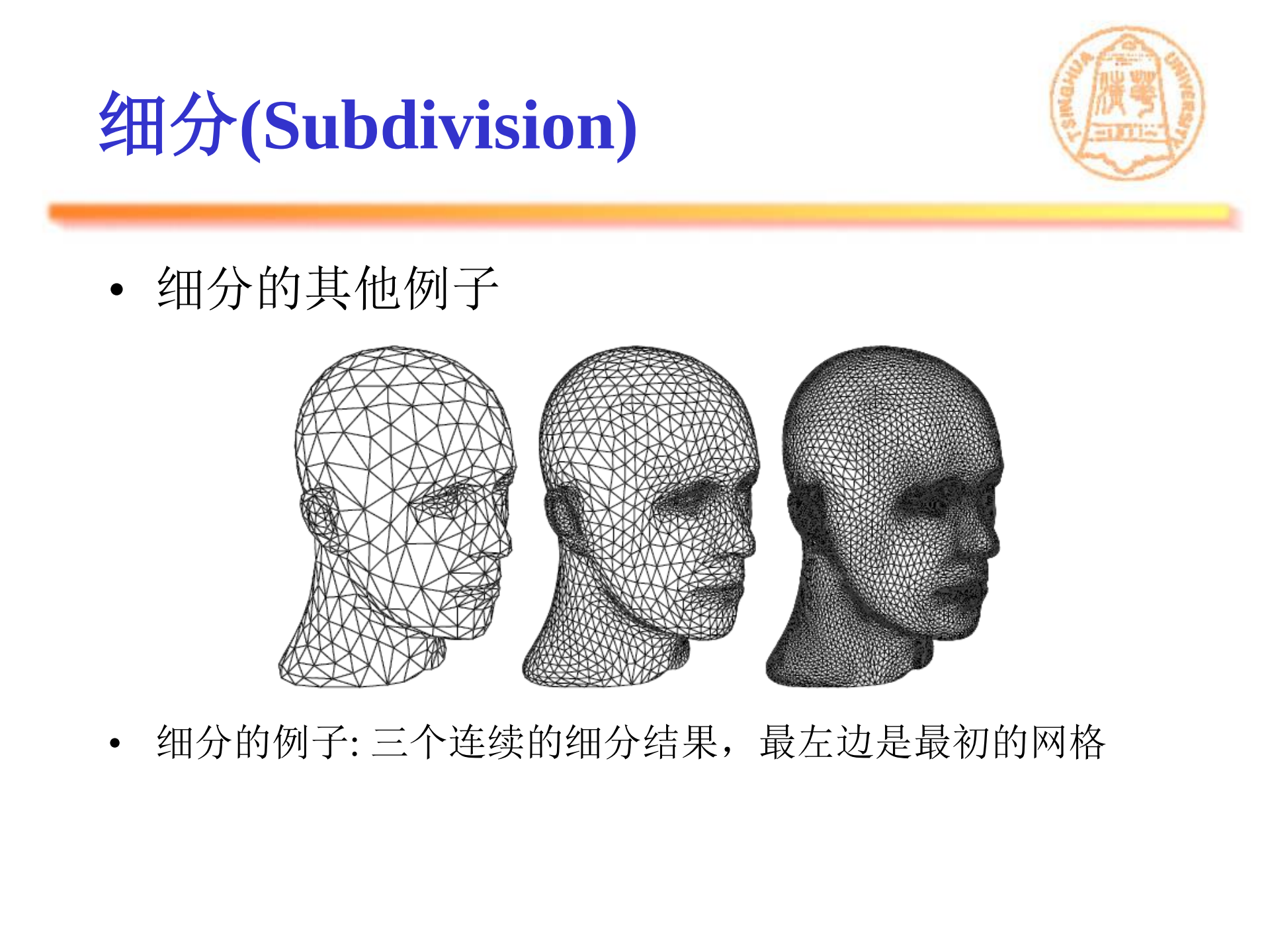
- 一维细分的例子



  - 最左面的4个点由直线段相连
  - 右边一个改进版本: 3个新的点被插入
  - 经过进一步的两轮细分,曲线已经变得相当光滑



细分(Subdivision)



- 细分的其他例子

- 细分的例子: 三个连续的细分结果，最左边是最初的网格

细分(Subdivision)

- 细分(Subdivision)
  - 表面细分可以被看成一个两阶段过程 (最初的网格被称作控制网格)
    - 第一步, 称作细化阶段, 创建新顶点并与先前顶点相连产生新的、更小的三角形
    - 第二步, 称作平滑阶段, 计算(原)顶点的新位置
  - 这两步的细节决定了不同的细分方案
    - 在第一步中, 一个三角形可以由不同的形式进行分割
    - 在第二步中, 新顶点的位置可以由不同的方式插值产生

- 细分方案(Subdivision Schemes)
  - 我们将介绍两个细分方案
    - Loop细分： Charles Loop

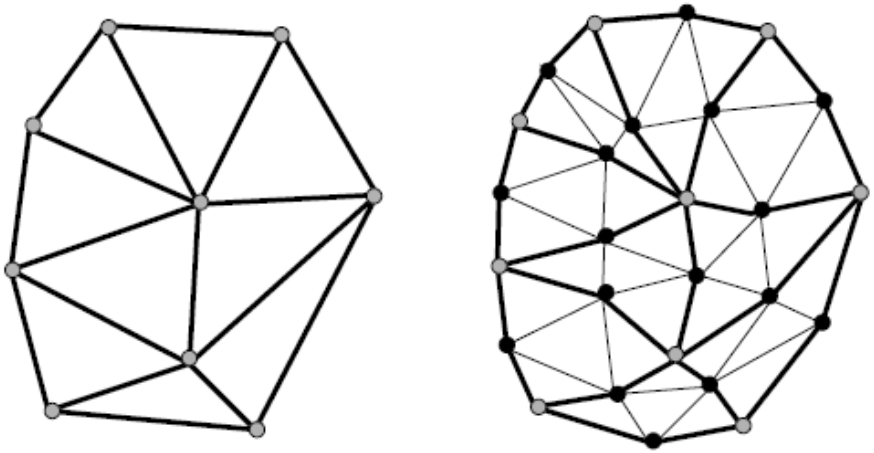
    - $\sqrt{3}$ 细分： Leif Kobbelt

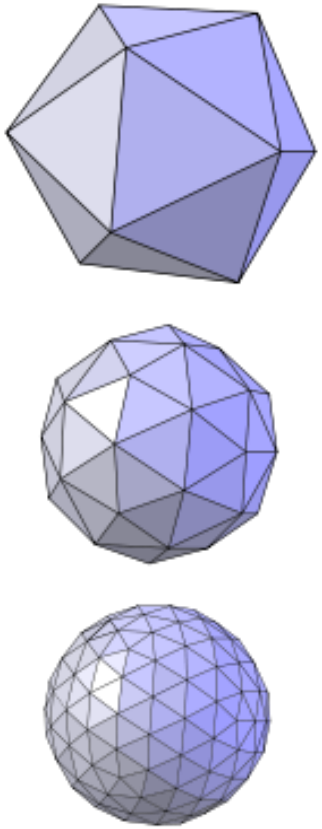
Loop细分(Loop Subdivision)

- Loop细分
  - Loop 细分是第一个基于三角网格的细分方案
  - 它更新每个已有的顶点并对每条边创建一个新的顶点，然后每个三角形被分割成四个新的三角形
  - 因此，经过n步细分，一个三角形被分割成$4n$个三角形

Loop细分(Loop Subdivision)

- Loop细分(Loop Subdivision)





Loop细分的例子，新的顶点以黑色显示，每条边中加入一个新的顶点, 新的顶点被连接起来创建四个新的三角形以替换掉原先的三角网格

Loop细分(Loop Subdivision)

- 关注一个已存在的顶点 p^k , 其中 k是当前已进行的细分步数；按此记法, p^0 即表示控制网格的顶点，一般地，细分过程可以表示为

$$p^0 \to p^1 \to p^2 \to p^3 \ldots$$

- 如果 p^k 的度为 n,那么 p^k 有n个相邻顶点,记做 $p_i^k, i \in \{0,1,\ldots,n-1\}$

Loop细分(Loop Subdivision)

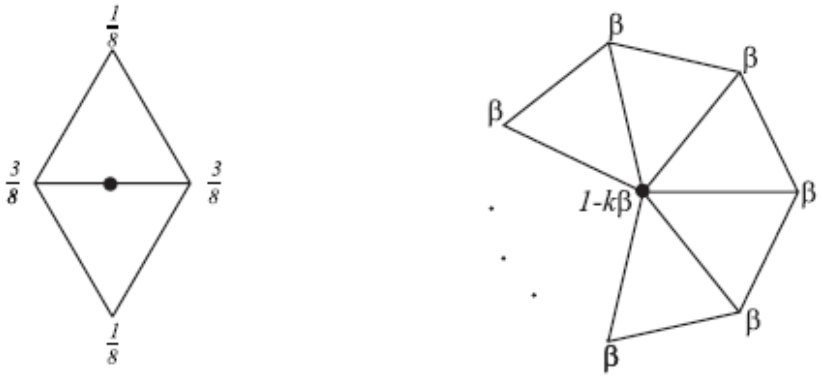
- 细分规则(Subdivision rule)
  - 接下来给出Loop细分方案的规则
  - 第一个公式是将每个顶点 p^k 升级 p^{k+1} 的准则
  - 第二个公式是在边 $p^k p_i^k$ 创建新顶点 p_i^{k+1} 的准则

$$p^{k+1} = (1-n\beta)p^k + \beta(p_0^k + \ldots + p_{n-1}^k)$$

$$p_i^{k+1} = \frac{3p^k + 3p_i^k + p_{i-1}^k + p_{i+1}^k}{8}, i = 0, \ldots, n-1$$

Loop细分(Loop Subdivision)

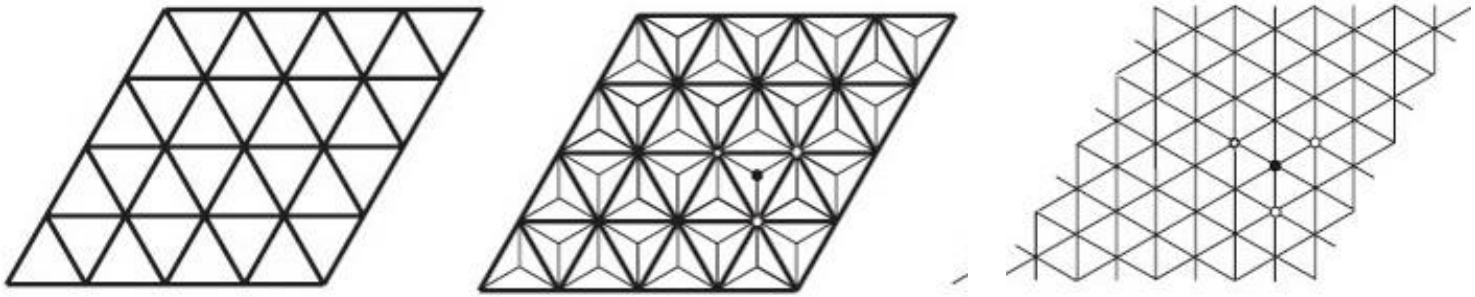
- 细分规则(Subdivision Rule)



  - 这张图给出了插入顶点的规则模板 (左边是在边上加入一个新顶点,右边是对顶点的更新调整)
  - β 是 n 的函数 $\frac{1}{n}(5/8-(\frac{3}{8}+\frac{1}{4}\cos\frac{2\pi}{n})^2)$

$\sqrt{3}$ 细分($\sqrt{3}$ -Subdivision)

- $\sqrt{3}$细分($\sqrt{3}$-Subdivision)
  - 能否将一个三角形分为三个子三角形，构造细分格式？
  - 为了得到更均匀分布的三角形, 每一个原始三角形都被翻转以使之连接两个相邻的中间顶点而不是连接两个已经存在的顶点



$\sqrt{3}$ 细分的过程

$\sqrt{3}$ 细分($\sqrt{3}$ -Subdivision)

- 细分规则(Subdivision rule)
  - 在下面第一行公式中, p_m表示中间顶点,由计算三角形三个顶点 p_a, p_b, p_c 的平均值得来

    对于每一个存在的顶点, p^k 用第二行公式来做更新,其中p_i^k表示p^k 的相邻顶点；在细分方案中, n是 p^k 的度，k是细分步数

$$p_m^{k+1} = (p_a^k + p_b^k + p_c^k)/3$$

$$p^{k+1} = (1 - n\beta)p^k + \beta\sum_{i=0}^{n-1} p_i^k$$

$\sqrt{3}$ 细分($\sqrt{3}$-Subdivision)

- 细分规则
  - 其中，β是一个度n的函数 $\beta(n) = \dfrac{4 - 2\cos(2\pi / n)}{9n}$
  - 这种函数保证细分之后产生的曲面，至少为一阶连续

网格模型简化

背景: 图形学转折期的新方向

- 真实感的**实时绘制**问题变的日益突出，为了达到实时绘制，需要简化模型
- 92年开始，面片简化、压缩、信号处理、纹理合成成为SIGGRAPH会议的一个热门话题；如今网格模型的处理，如网格参数化、网格重剖、细节建模、模型编辑仍然是研究热点；已经演化为数字几何处理这一研究方向
- Eurographics Symposium on Geometry Processing

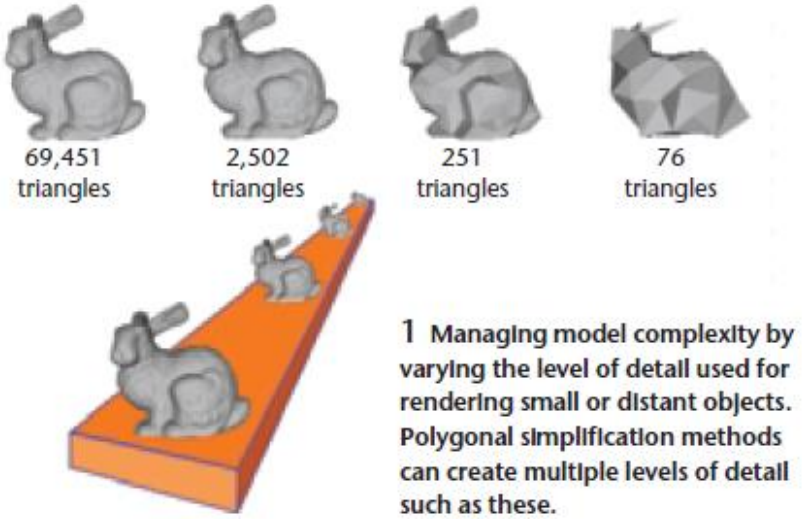
  From 2003, and the 21th symposium will be held in Genova.

简化 (Simplification)

- 网格简化的概念
  - 用一些相对简单但维持几何特征的表示来近似一个给定的网格
- 网格简化的作用
  - 移除几何冗余
    - 如一个有很多共面小三角形的平坦区域，将这些三角形融合成大的多边形能降低模型的复杂度
    - 降低存储或传输的规模
  - 提高运行性能
    - 简化对于高效的渲染和编辑具有重要作用

简化 (Simplification)

- 网格简化的作用
  - 产生场景中物体的层次细节 (level of details，简称 LOD)
    - 较远的物体用较低的层次细节呈现，较近的物体用较高的
    - 例子



1 Managing model complexity by varying the level of detail used for rendering small or distant objects. Polygonal simplification methods can create multiple levels of detail such as these.

简化 (Simplification)

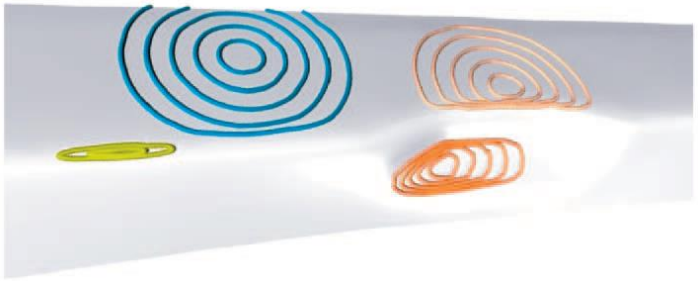
- 技术方法: 如何简化?
  - 几乎每一个简化技术都使用下面四种基本多边形移除技巧的组合或者变形:
    - 采样（sampling）
    - 自适应细分（adaptive subdivision）
    - 去除（decimation）
    - 顶点合并（vertex merging）

    下面详细解释这四种基本操作

- 采样(Sampling)
  - 采样算法通过选取模型表面的点简单地对模型进行几何取样，这些方法通常是复杂的并且难于编程实现
  - 由于这些方法难于精确地获取高频特征，通常在没有尖角的光滑表面上取得的效果较好

在特征丰富的表面上？引进：特征敏感度量。

特征敏感的再分片（Feature sensitive Re-meshing）



- 自适应细分(Adaptive subdivision)
  - 自适应细分算法通过寻找一个可以递归细分的基底网格来近似最初的模型，该算法在基底模型易于获取的情况下能取得很好的效果
    - 例如, 一般地形模型的基底模型是一个四边形
  - 自适应细分算法能够保持表面拓扑细节, 因此有可能会大大提高对于大规模数据处理的能力

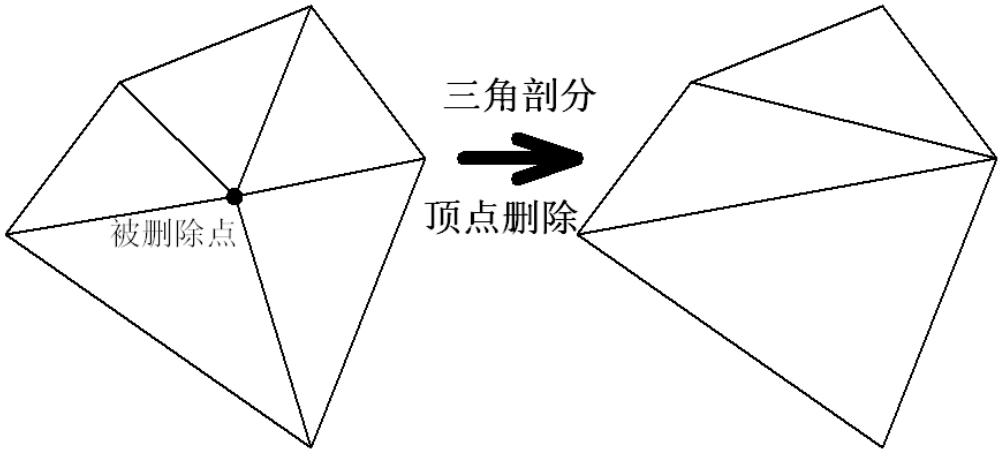
- 去除(Decimation)
  - 去除方法迭代地移除网格上的顶点或面片，并三角化每一步移除操作后留下的孔洞
  - 这类方法相对简单、易于编程实现且运行效率高
  - 这类算法尤其精于移除网格的几何冗余，比如共面多边形

- 顶点合并 (Vertex-merging)
  - 顶点合并方案通过将断裂三角化模型的两个或更多顶点合并为一个顶点，转而可以使之与其他顶点合并，该算法需要采用多种技术来决定以何种顺序合并顶点
  - 边坍塌算法(Edge-collapse)是一种最常用的合并方法，该方法总是将共边的顶点合并, 因此易于保持局部拓扑；更一般的顶点合并算法还支持修改拓扑并且聚合物体

网格化简的基本操作（1）

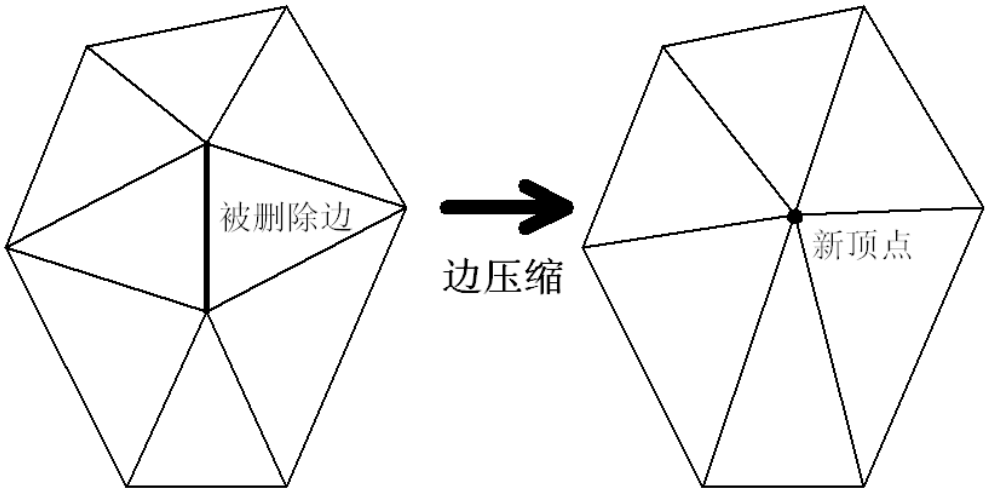
- 三种不同的基本化简操作：

1. **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



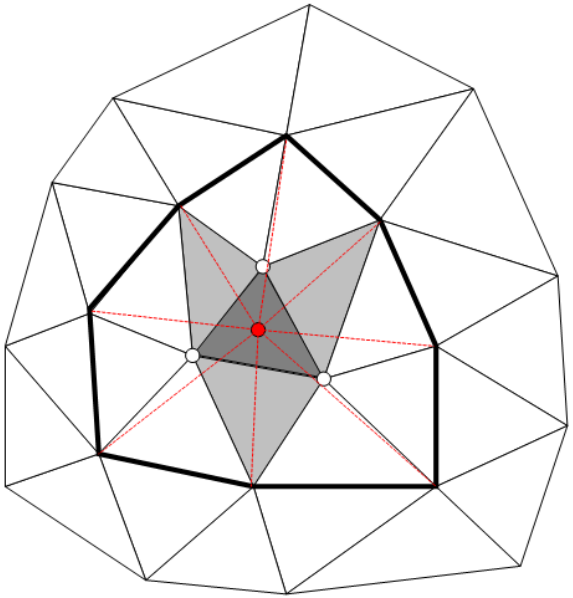
网格化简的基本操作（2）

2. **边压缩操作：**网格上的一条边压缩为一个顶点，与该边相邻的两个三角形的退化



网格化简的基本操作（3）

3. **面片收缩操作：**网格上的一个面片收缩为一个顶点，该三角形本身和与其相邻的三个三角形都退化



基于长方体滤波的多面体简化

- 1993年，Rossignac和Borrel提出一个实用的、能实时建立LOD模型的多面体简化算法

长方体滤波的步骤（1）

- 给定一个多面体 M，记 K 为其拓扑，假设 M 已三角化，算法首先建立 M 的长方体包围盒，并将该包围盒所包围的空间均匀剖分成一系列的小长方体子空间，然后采用各长方体子空间对景物顶点进行聚类合并，位于同一长方体空间的顶点被归于同一类(Cluster)

长方体滤波的步骤（2）

- 最后，属于同一类的顶点被合并为一代表点，而这些代表顶点为原多面体所示景物的重新采样，基于原多面体的拓扑结构和这些这些采样点可重新产生一多面体，所得到的多面体即为保持一定层次细节的模型
- 原多面体的包围盒剖分生成的子空间越小，所得到的层次模型就越逼近于原多面体

长方体滤波的缺点

- 方法的主要缺点是顶点的合并导致了一些**重要高频细节的丢失**
- 参考文献
  - Rossignac J, Borrel P, Mutli-Resolution 3D approximations for rendering complex scences, in Modeling in Computer Graphics, edit by B Falcidieno and T L Kunii, spring-Verleg, 1993, pp 455-465.

顶点删除技术

- 想法: 设法减少景物表面的采样点数目
  - 假设景物表面已离散为一系列三角形，顶点删除算法首先从原始模型的顶点集中删除一些不重要的顶点，同时从其面片集中删除与这些顶点相连的所有面片
  - 经上述操作后在原景物表面留下了一些空洞，算法再对这些空洞进行局部三角剖分
- 如何判定不重要的顶点？局部判别准则

局部判别准则

- 基于相邻面片和边界的局部平坦性原则
  - **Schroeder** 92
- 采用等距面来限定简化模型顶点的变化范围
  - **Cohen** 96

Schroeder 的局部判别准则

该判别准则分两种情况：

1） 对网格内部顶点 v，记其周围相邻面片集为 S，则该点的平坦性标准由下述的距离来描述：

$$d = \left|\mathbf{N}\cdot(v - C)\right| \qquad (6.1)$$

其中 $\mathbf{N}$ 为向量 $\dfrac{\sum_{f\in S}\mathbf{n}(f)A(f)}{\sum_{f\in S}A(f)}$ 的单位向量，$C = \dfrac{\sum_{f\in S}c(f)A(f)}{\sum_{f\in S}A(f)}$，

这里 $A(f)$，$c(f)$，$\mathbf{n}(f)$ 分别为三角面片的面积、中心和法向量

2） 对边界顶点 v ， 记与它相邻的两个边界顶为 v_1 , v_2 ，则其平坦性标准定义为 v 到 v_1 与 v_2 连线的距离

- **参考文献**
  - Schroeder W, Zarge J A, and Lorensen W E, Decimation of triangle meshes, Computer Graphics, 1992, 26-2, 65-70.

Cohen 的局部判别准则

- 多面体的包络（envelope）的概念

  多边形网格表面 P 可看作一张分片线性参数曲面

$$r(u,v) = (r_x(u,v), r_y(u,v), r_z(u,v)) \quad (6.2)$$

其单位法向量为

$$\mathbf{n}(u,v) = (n_x(u,v), n_y(u,v), n_z(u,v))$$

- 对给定的 $\varepsilon > 0$ ，P 的三维 ε 等距面定义为

$$r^{\varepsilon}(u,v) = r(u,v) + \varepsilon \mathbf{n}(u,v)$$

近似地定义原始多边形网格 P 沿其正、负法向的 ε 等距面 $P(+\varepsilon)$ 和 $P(-\varepsilon)$

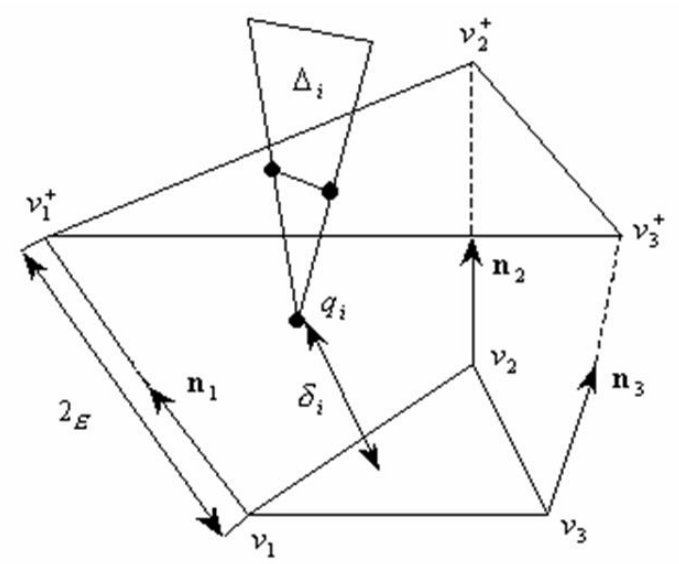
ε 等距面 $P(+\varepsilon)$ 和 $P(-\varepsilon)$ 上对应顶点 v_i^+ , v_i^- 及其法向量可分别表示为

$$v_i^+ = v_i + \varepsilon \mathbf{n}_i \qquad v_i^- = v_i - \varepsilon \mathbf{n}_i \qquad \mathbf{n}_i^+ = \mathbf{n}_i^- = \mathbf{n}_i$$

上述方法生成的 $P(\pm\varepsilon)$ 可能出现自交现象

- 我们需要计算一个最大的不存在自交的 ε

- 我们用解析法计算ε
  现来考察 P 上的任一
  三角形 $\Delta v_1 v_2 v_3$



- 对 P 上每个与三角形 $\Delta v_1 v_2 v_3$ 不相邻的三角面片Δ_i，判别Δ_i 是否与 $\Delta v_1 v_2 v_3$的基本柱体相交；可计算得到q_i到 $\Delta v_1 v_2 v_3$ 的距离δ_i

Cohen的点删除流程

- 采用贪心搜索策略，将原表面P的所有顶点列入待处理的顶点队列
- 对当前待处理顶点队列中的一顶点，算法尝试从P上删除该顶点及与该顶点直接相邻的三角面片，并试图用三角剖分方法来填补顶点删除后在表面P上形成的空洞

- 若空洞位于P的包络内且能被成功填补，则从当前队列中删除该顶点，并重构原来与该顶点相连接的各顶点的拓扑关系；
- 否则，该顶点从当前待处理队列中退出，表面P保持不变
- 重复处理，直到待处理顶点队列变为空
- 参考文献
  - Cohen J, Varshney A, Manocha D, and Turner D, Simplification Evvelopes, Computer Graphics, 1996, 119-128.

渐进的网格简化技术

- 渐进网格算法中，任一网格 $\hat{M}$ 均可表示为基本网格及n个 逐步细化网格 $M^i\{i = 1,2,\dots,n\}$的变换，且有 $\hat{M} = M^n$
- 一张网格 M 可定义为1个二元组 (K,V)

  其中 K 是一个单纯复形(simplicial complex)，它表示了 M 的顶点、边和面的邻接关系；

  $\mathrm{V} = \{v_i \in R^3 | \mathrm{i} = 1,2,\dots,\mathrm{m}\}$ 是 M 的顶点位置向量集，它定义了网格 M在 R^3中的形状

- 单纯复形 K 由顶点集 $\{i = 1,2,\ldots,m\}$ 及其称之为单形的非空子集组成：

  0-单形 $\{i\} \in K$ 即为顶点

  1-单形 $\{i, j\} \in K$ 为一条边

  2-单形 $\{i, j, k\} \in K$ 为一个面

单选题 1分

设置

顶点集{1,2,3,4}构成的单纯复形，1-单形最多几个?

A 4

B 6

C 8

D 12

提交

拓扑实现的概念

- 值得注意的是，单纯复形 K 并不包含点集 $\{i = 1,2, \ldots, m\}$ 的所有子集，仅包含了构造网格 M 所有面、边、顶点的子集
- 为在结构上刻划单纯复形，我们引进拓扑实现(topological realization) $|K|$ 的概念

- 若将顶点 {i}$(i = 1,2, \dots, m)$ 看成为R^m 中的基向量

  $e_i = \{0, \dots, 0, i, 0, \dots, 0\}$

- 则定义在 R^m 中的集合$|K|$为 K 的拓扑实现：

$$|K| = \bigcup_{s \in K} |s|$$

  其中s 为 K 的一个单形，$|S|$为s在 R^m 空间中的顶点的凸包

几何实现的概念

- 为此我们记 $\phi_V := \phi_{|K|} : |K| \to R^3$ ，称为$\Delta v_1 v_2 v_3$在R^3中的几何实现(geometric realization)
- 若$\phi_V(|K|)$不自交，则 ϕ_V为1-1映射。此时，ϕ_V为一嵌入映射，即对 $\forall p \in \phi_V(|K|)$，存在唯一m维向量 $b \in |K|$，使得 $p = \phi_V(b)$，我们将 b 称为 p 关于单纯复形的重心坐标向量(barycentric coordinate vector) 事实上，b可表示为:

$$b = \sum_{i=1}^{m} b_i e_i$$

容易知道，当 M为一三角片网格时，$\phi_v(|K|)$上的任一点的重心坐标向量 b 中至多只有三个分量非零

显式能量函数度量

- 有了上述定义，Hoppe采用显式能量函数$E(M)$来度量简化网格与原始网格的逼近度([HOPP96])：

$$E(M)=E_{dist}(M)+E_{spring}(M)+E_{scalar}(M)+E_{disc}(M)$$

- 其中$E_{dist}(M)$为M 的距离能量，定义为点集 $X=\{x_{1,\cdots,}x_n\}$ 到网格M 的距离平方：

$$E_{dist}(M)=\sum_{i=1}^{n}d^2(x_i,\phi_V(|K|))$$

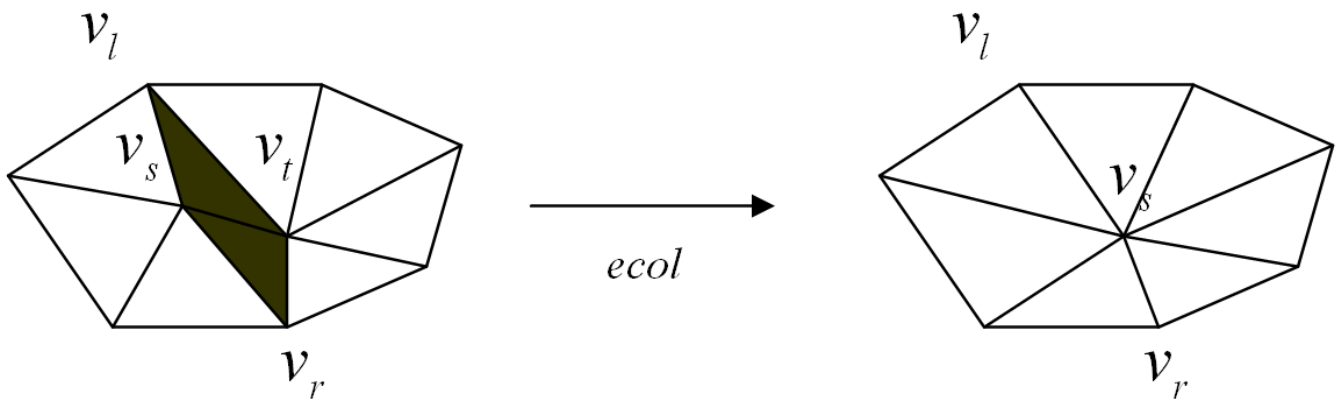
- E_{spring} 为 M 的弹性能量，这相当于在 M 的每条边上均放置一条弹性系数为 k 的弹簧，即

$$E_{spring}(M) = \sum_{\{i,j\}\in K} k\left\|v_i - v_j\right\|^2$$

- $E_{scalar}(M)$ 度量 M 的标量属性的精度，而 $E_{disc}(M)$ 则度量了上视觉不连续的特征线（如边界线、侧影轮廓线等）的几何精度。

边收缩变换

- Hoppe利用边收缩变换 (edge collapse transformation) 来逐步迭代计算上述能量的优化过程
- 下图给出了一个边收缩变换的过程，它将该边的两个端点 (v_s, v_t) 收缩为一个顶点 v_s，经过这个变形后，其相邻两个面 $\{v_s, v_t, v_l\}$ 和 $\{v_t, v_s, v_r\}$ 均退化为一条边



- 因此，初始网格 $\hat{M} = M^n$ 可经过 n 组的边收缩变形后简化为 M^0：

$$\hat{M} = M^n \xrightarrow{ecol_{n-1}} \cdots \xrightarrow{ecol_1} M^1 \xrightarrow{ecol_0} M^0$$

顶点分裂变换

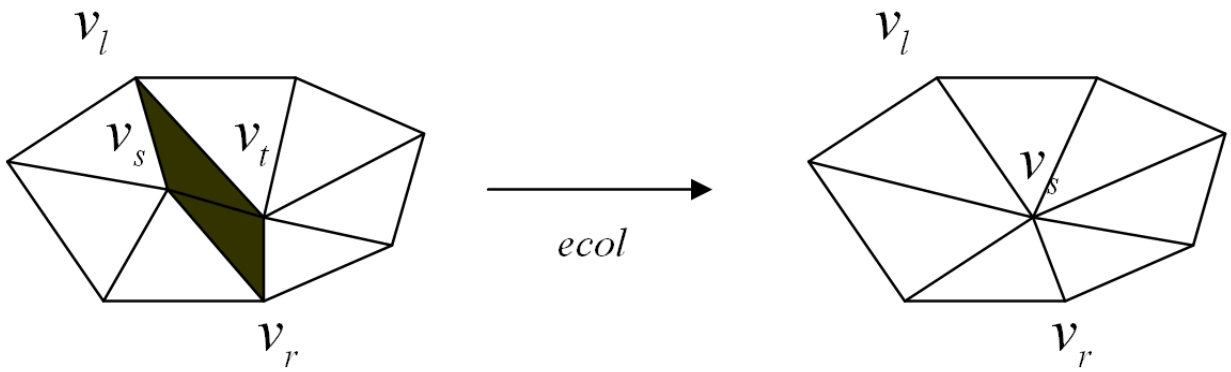
- 顶点分裂变换(vertex split transformation)
  - 由于边收缩变换 $ecol$ 是一可逆变换，我们称其逆变换为顶点分裂变换
- Hoppe利用刚刚提到的能量函数来选择这些变换，使变换前后两网格间的能量差 ΔE 达到最小

$$E(M) = E_{dist}(M) + E_{spring}(M) + E_{scalar}(M) + E_{disc}(M)$$

- 参考文献
  - Hoppe H, DeRose T, Duchamp T, Mesh Optimization, Computer Graphics, 1993,27-4,19-26.
  - Hoppe H, Progressive Meshes, Computer Graphics, 1996, 30-4, 99-108.

基于二次误差度量的简化技术

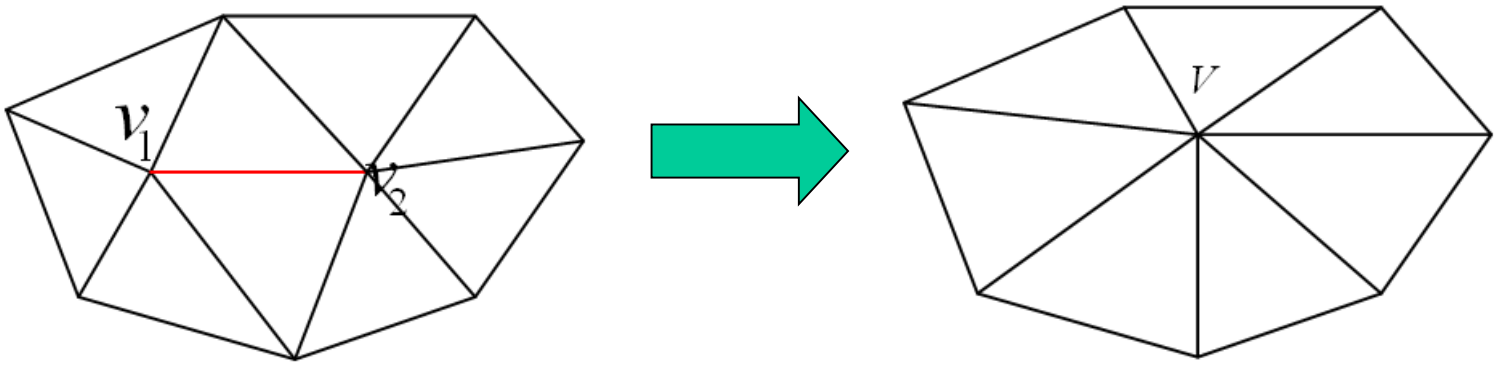
- Hoppe的边收缩(edge collapse)操作可推广为一般的顶点合并变换来描述 $(v_1, v_2) \to v$，其含义是将场景中的两个顶点 v_1, v_2 移到一新的位置 v，将连向 v_1, v_2 的所有边都连向 v，并删除所有退化的边和面片



点对合并（1）

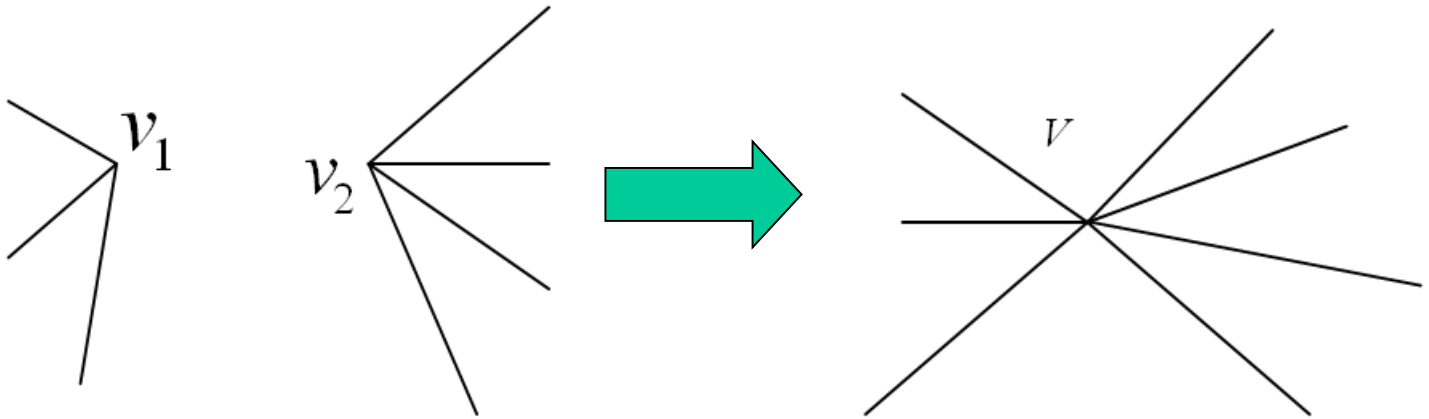
- 点对(v_1, v_2)合并的原则

  （1）v_1, v_2 为某一表面上的相邻点，即 v_1, v_2 为一条边



点对合并（2）

（2）$\|v_1 - v_2\| < t$，t 为用户给定的阈值参数



二次误差度量

- Garland和Heckbert引进了二次误差度量来刻划每一个顶点移动后引起的误差，对表面上的每一个顶点 v_a 均有许多三角面片与之相邻，记 $plane(v_a)$ 为这些三角形所在的平面方程所构成的集合，即

$$plane(v_a)=\left\{(a,b,c,d)\,\middle|\,\begin{array}{c} ax+by+cz+d=0,(a^2+b^2+c^2=1) \\ \text{is coefficients of the adjacent plane of } v_a \end{array}\right\}$$

- 则我们采用如下的二次函数来度量 v_a 移动到 v 时产生的误差：

$$\Delta(v_a \to v) = \sum_{p \in plane(v_a)} (pv^T)^2 \qquad (1)$$

其中 v=(x, y, z, 1) 为齐次坐标，展开上式得到

$$\Delta(v_a \to v) = \sum_{p \in plane(v_a)} (pv^T)^2 = v\left(\sum_{p \in plane(v_a)} K_p\right)v^T = vQ(v_a)v^T \qquad (2)$$

$$Q(v_a) = \sum_{p \in plane(v_a)} K_p$$

- （2）式中

$$K_p = p^T p = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix} \qquad (3)$$

- 这样，对每一顶点 v_a，在预处理时，我们均可按上述方法计算矩阵 $Q(v_a)$，进而就可对其移动进行误差度量了。但由于每次合并时，需同时移动两点，故必须考虑同时移动多个顶点后形成的误差

多点移动的误差

- Garland和Heckbert简单地采用加法规则来刻划多点移动而形成的误差，对点对合并 $(v_1, v_2) \to v$ ，其误差为

$$\Delta(v) = \Delta(v_1 \to v) + \Delta(v_2 \to v) = v(Q(v_1) + Q(v_2))v^T = vQv^T$$

其中 $Q = Q(v_1) + Q(v_2)$

因而，应选取$\boldsymbol{v}$使误差达到最小

- 由极值的性质知，v满足系统方程：

$$\frac{\partial \Delta(v)}{\partial x} = \frac{\partial \Delta(v)}{\partial y} = \frac{\partial \Delta(v)}{\partial z} = 0$$

即

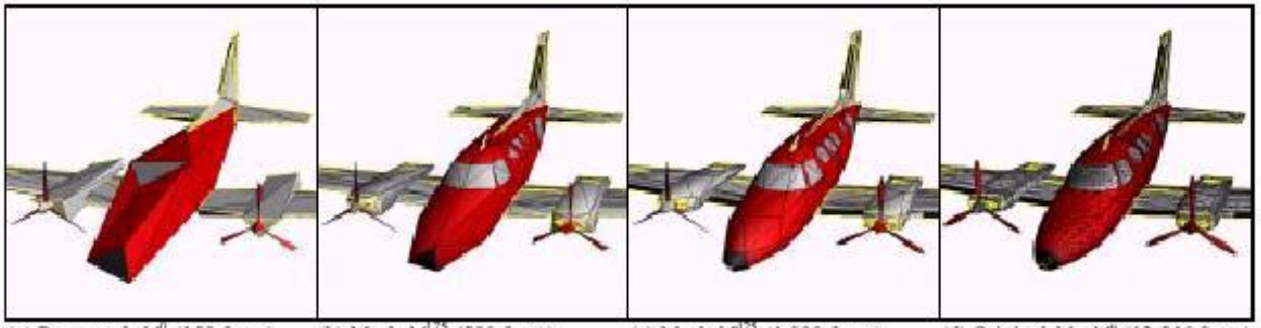
$$\begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} v = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

- 若上式有唯一解，则其解为v的最优解；否则，用伪逆技术求v
- 若伪逆技术失败，则简单地选取v为v_1，v_2或$\frac{v_1 + v_2}{2}$中的任何一个

- 参考文献
  - Garland M., Heckbert P S., Surface simplification using quadric error matrix, Computer Graphics, 1997, 209-216.

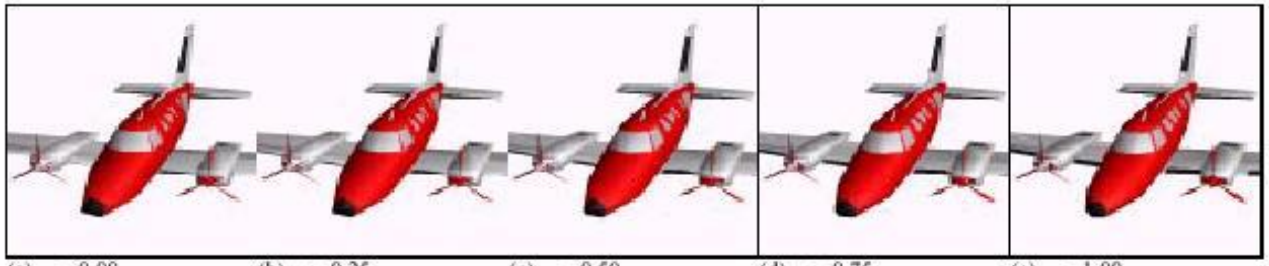
例子

1. Hoppe 的Progressive Meshes:



(a) Base mesh M^0 (150 faces) (b) Mesh M^{175} (500 faces) (c) Mesh M^{425} (1,000 faces) (d) Original $M = M^n$ (13,546 faces)

Figure 5: The PM representation of an arbitrary mesh M captures a continuous-resolution family of approximating meshes M^0 ___ $M^n = M$.



(a) _ = 0_00 (b) _ = 0_25 (c) _ = 0_50 (d) _ = 0_75 (e) _ = 1_00

Figure 6: Example of a geomorph $M^G(\_)$ defined between $M^G(0) = M^{175}$ (with 500 faces) and $M^G(1) = M^{425}$ (with 1,000 faces).

2. Garland的牛

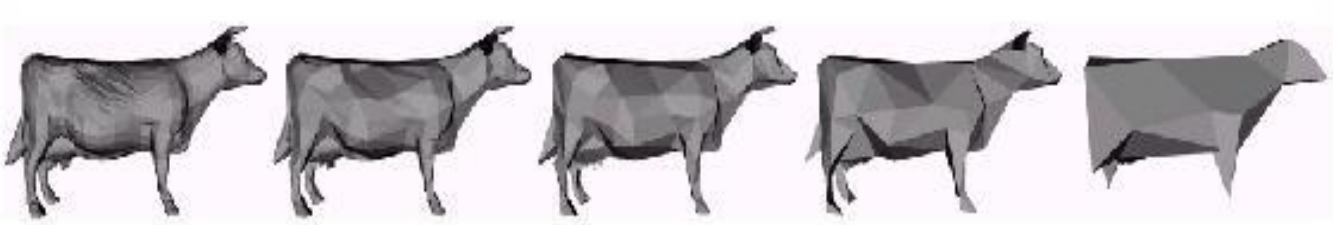
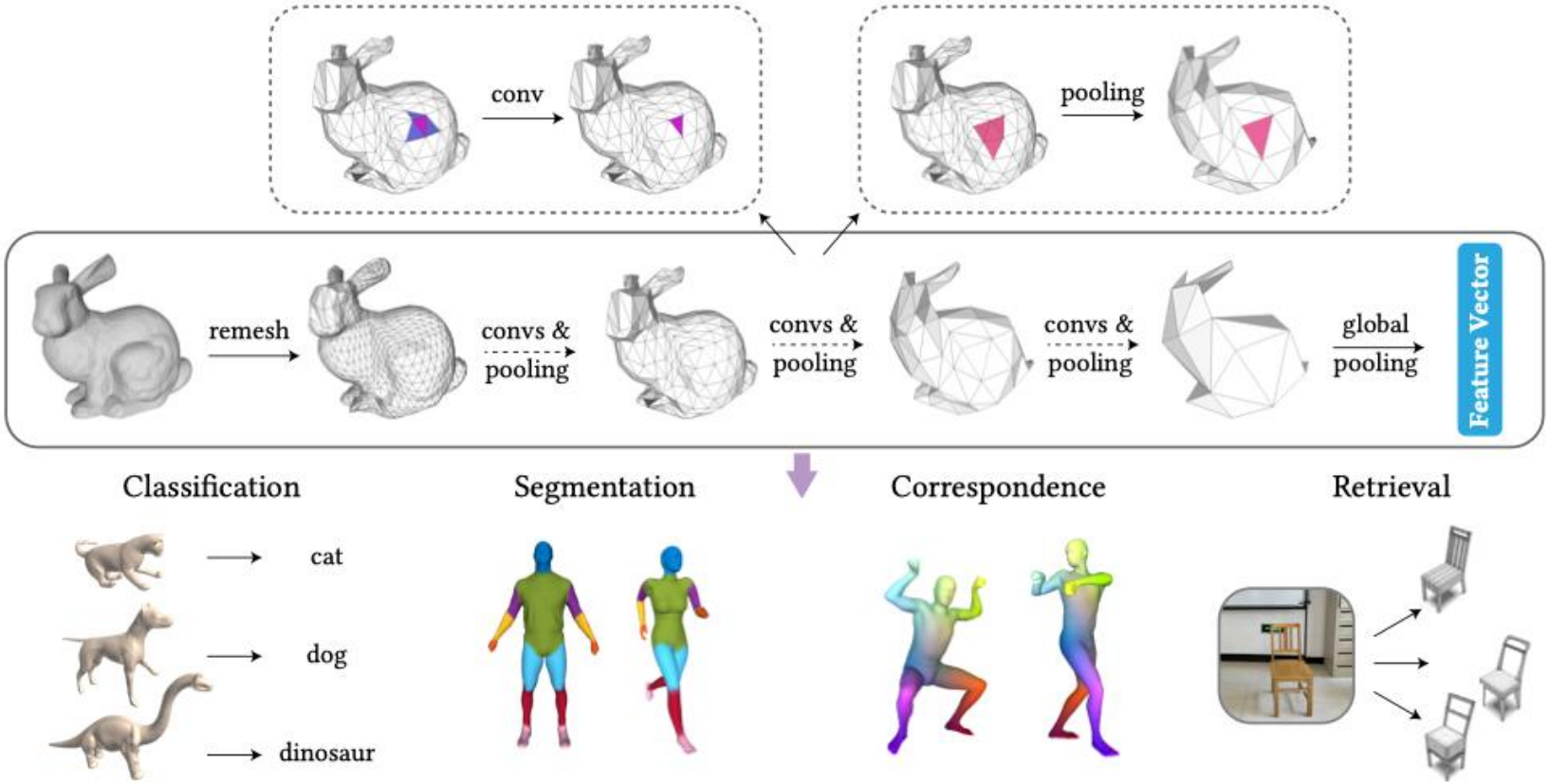


Figure 4: A sequence of approximations generated using our algorithm. The original model on the left has 5,804 faces. The approximations to the right have 994, 532, 248, and 64 faces respectively. Note that features such as horns and hooves continue to exist through many simplifications. Only at extremely low levels of detail do they begin to disappear.

网格简化和细分的应用： SubdivNet



- Hu S M, Liu Z N, Guo M H, et al. Subdivision-based mesh convolution networks. ACM Transactions on Graphics (TOG), 2022, 41(3): 1-16.

展望：基于细分的网格卷积网络(SubdivNet)

三角卷积

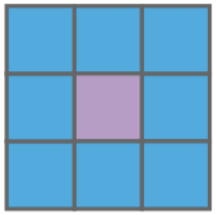
a).

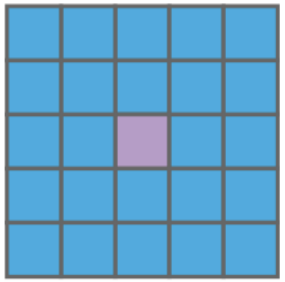
k=3, d=1

k=5, d=1

图像卷积

b).





k=3, d=1

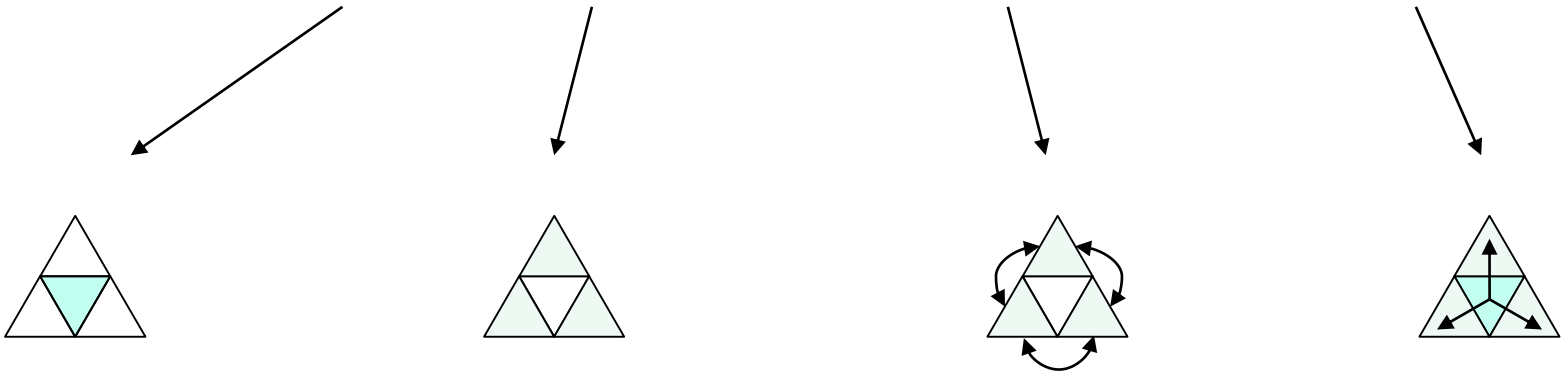
k=5, d=1

展望：基于细分的网格卷积网络(SubdivNet)

三角面片没有顺序，因此需要排列不变性:

构造顺序无关的中间特征，再进行加权

$$\mathrm{Conv}(f_i) = w_0 e_i + w_1 \sum_{j=1}^{n} e_j + w_2 \sum_{j=1}^{n} |e_{j+1} - e_j| + w_3 \sum_{j=1}^{n} |e_i - e_j|$$



中心特征　　邻域特征的和　　邻域差分的和　　中心与邻域的差的和

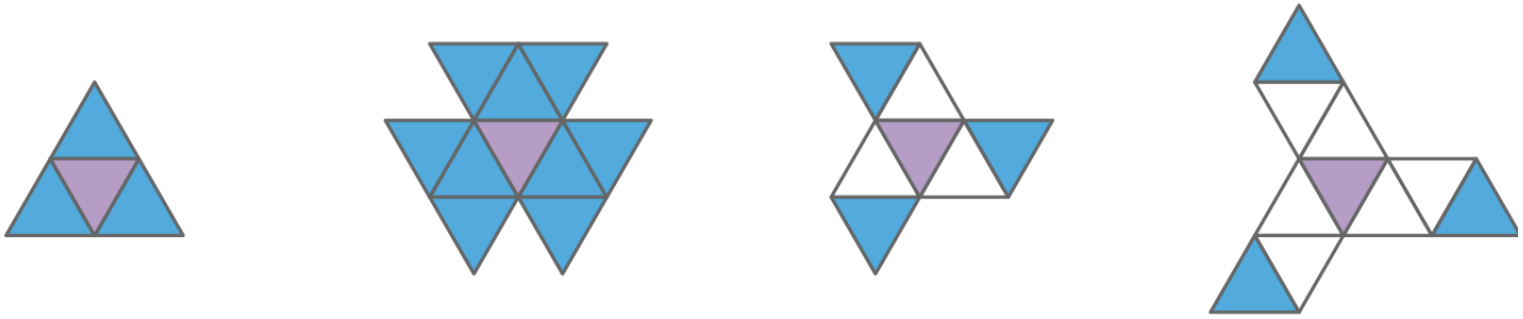
SubdivNet

卷积核大小 = k：距中心 $(k+1)/2$ 的面片

卷积核空洞 = d：d 次 zig-zag 后的面片

三角卷积

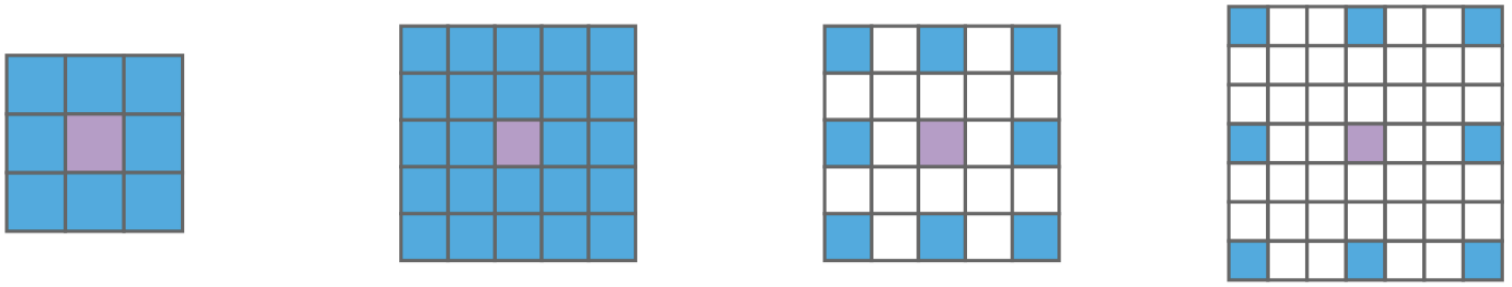
a).



k=3, d=1　k=5, d=1　k=3, d=2　k=3, d=3

图像卷积

b).



k=3, d=1　k=5, d=1　k=3, d=2　k=3, d=3

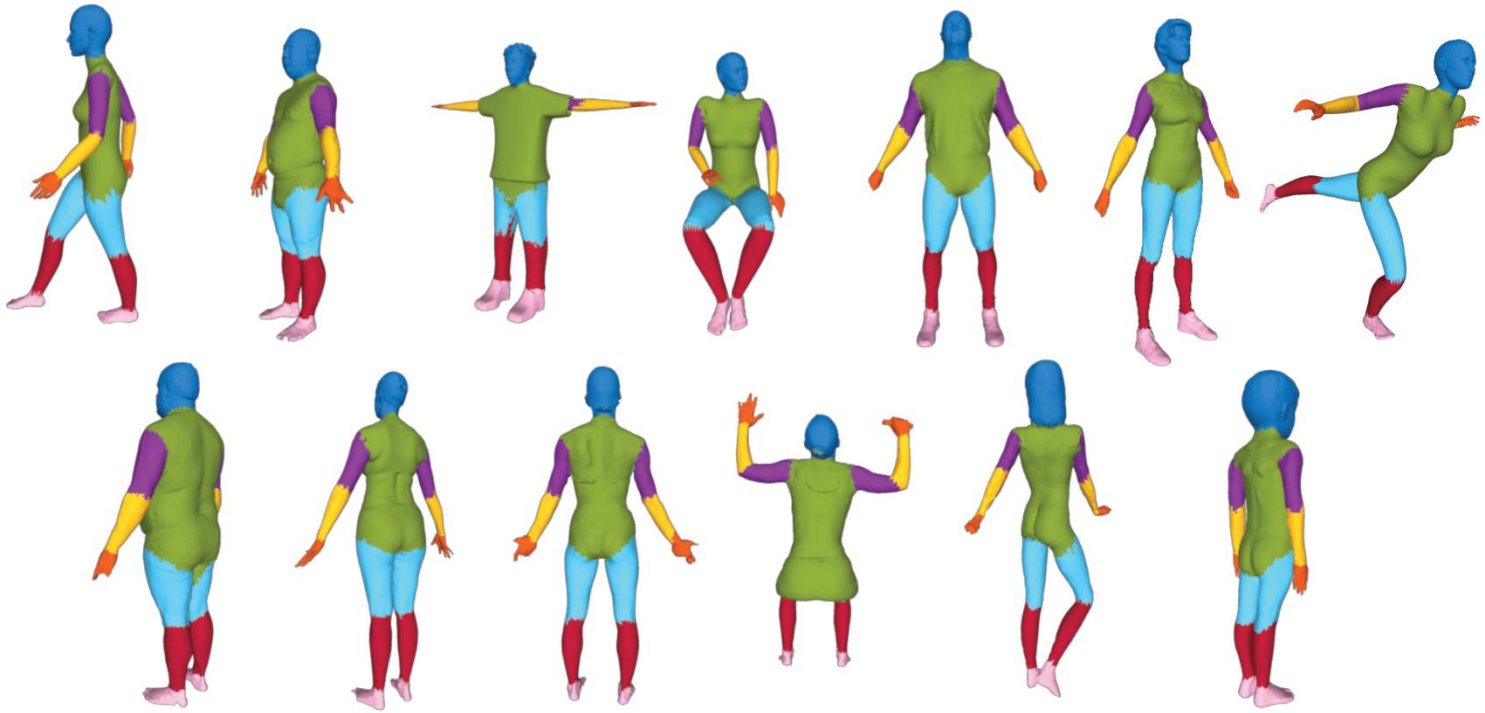
SubdivNet：网格分类

- 三个公开数据集上首次取得100%正确率
- Manifold40:
  - 基于 ModelNet40 构建的流形网格数据集，包含 12311 组数据；SubdivNet 的分类准确率超过其他网格方法。

方法	数据集			
	SHREC11 (split 16)	SHREC11 (split 10)	Cube Engraving	Manifold40
MeshCNN [TOG 2019]	98.6%	91.0%	92.2%	-
PD-MeshNet [NIPS 2020]	99.7%	99.1%	94.4%	-
MeshWalker [TOG 2020]	98.6%	97.1%	98.6%	90.3%
SubdivNet	**100%**	**100%**	**100%**	**91.4%**

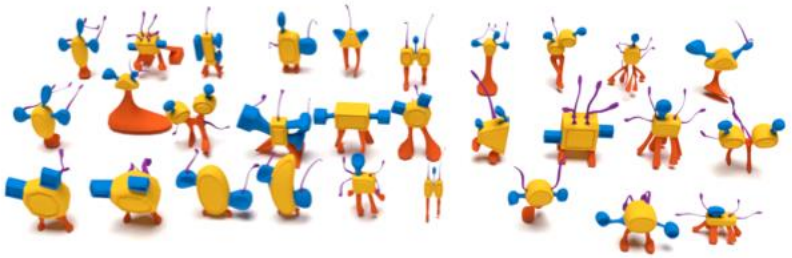
SubdivNet：网格语义分割(1)

- 人体模型语义分割
  - SubdivNet: 93.0% vs. MeshCNN: 87.7%
  - 目前准确率最高



SubdivNet：网格语义分割(2)

- COSEG数据集：目前准确率最高



外星人：97. 3%

椅子：96. 7%

瓷器：96. 7%

今日人物：Hugues Hoppe

- Hugues Hoppe, Google
  - 论文118篇，引用43756，h-index 68, 22篇论文引用超过500次
  - 1994年华盛顿大学博士
  - 导师：Tony DeRose
  - 成名作 Progressive Meshes引用5015次
  - 2004获ACM SIGGRAPH成就奖
  - 2011年获选ACM Fellow
  - ACM Transactions on Graphics主编 (2009-2011)

谢 谢！