Lecture 12

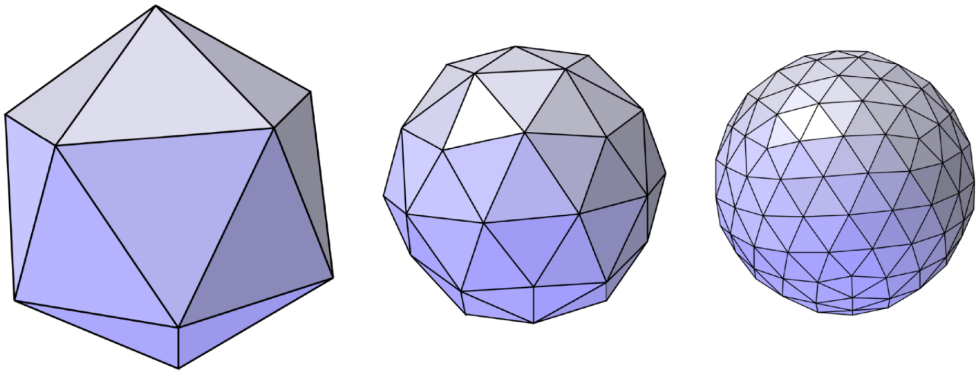
Lecture 11 提到了网格的三种操作

- subdivision 细分 - 增加三角形，增加分辨率
- simplification 简化 - 降低分辨率并保持外形
- regularization 正则化 - 修改三角形分布提升外形质量

1.subdivision 细分

在细分中，通常操作步骤分为两部

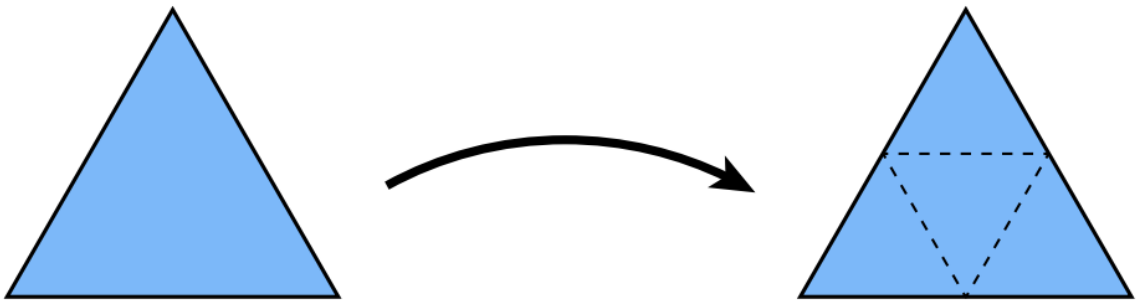
- 增加三角形
- 调整三角形位置



1.Loop 细分

Loop 细分

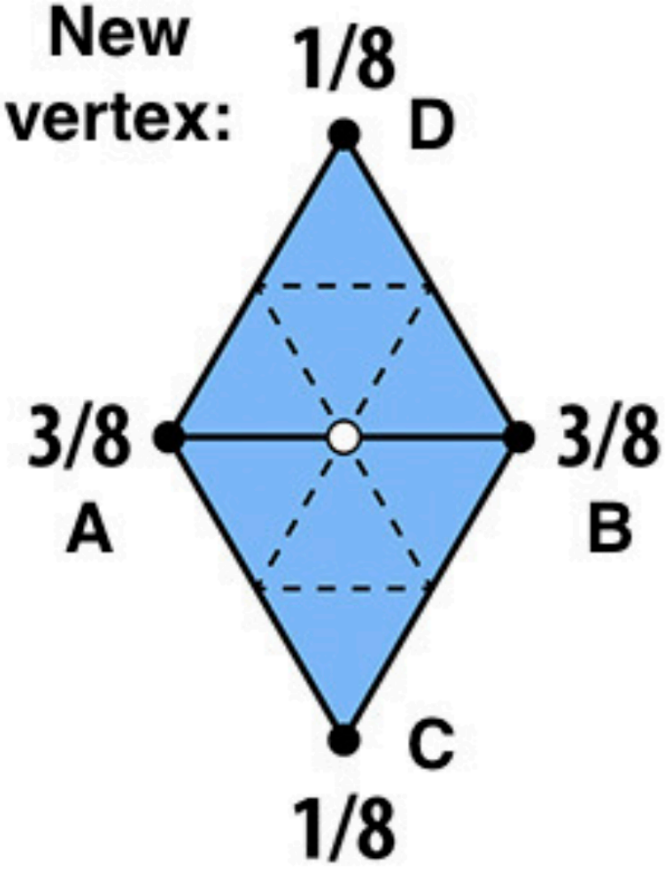
- 对原有三角形进行拆分，从1个三角形，拆分成4个三角形。



- 对 **新顶点，旧顶点** 的位置根据权重进行更新。

新顶点

- $\frac{3}{8} * (A + B) + \frac{1}{8} * (C + D)$

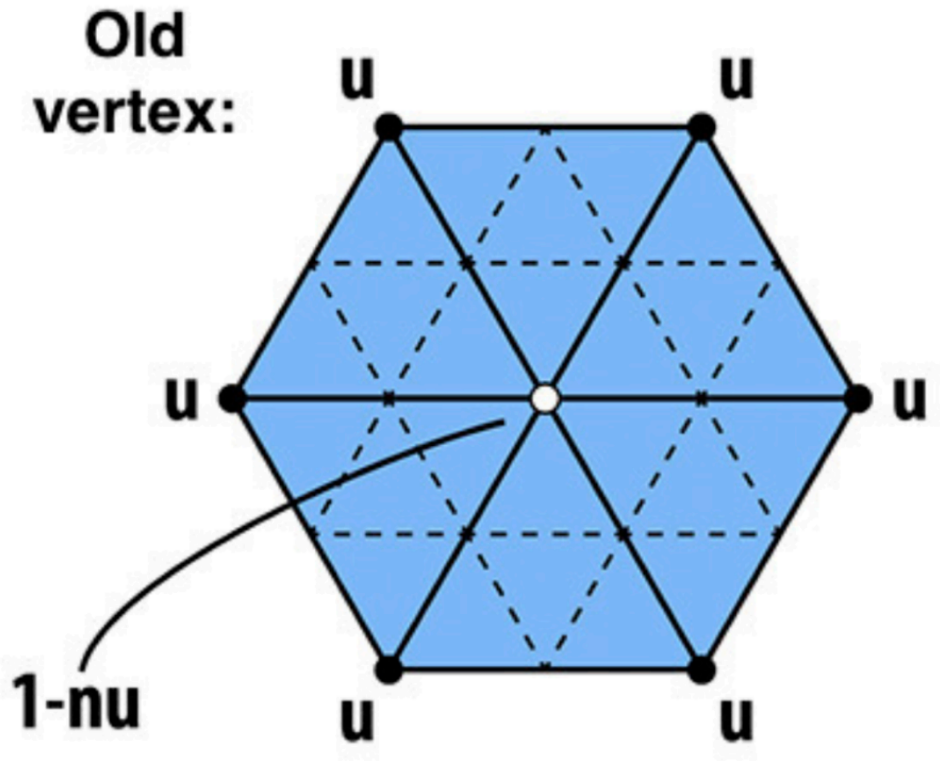


旧顶点

$$(1 - n * u) * position_{origin} + u * position_{neighborsum}$$

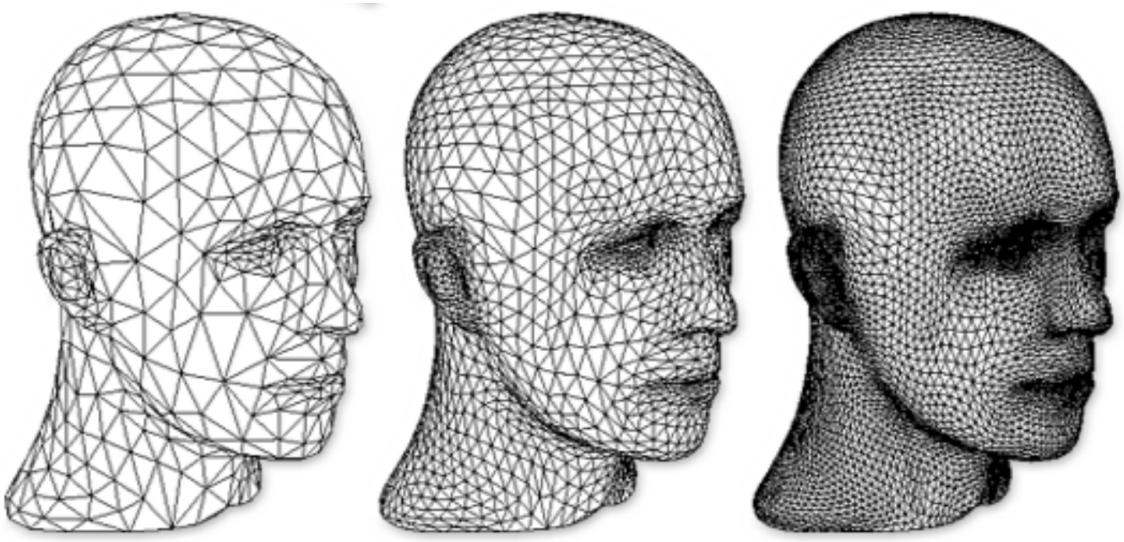
$$n : vertext\ degree$$

$$u : \frac{3}{16}\ if\ n = 3, \quad \frac{3}{8n}\ otherwise$$



Loop细分结果

- 只能用于三角形的细分



2.Catmull-Clark 细分

该细分将顶点分为两类

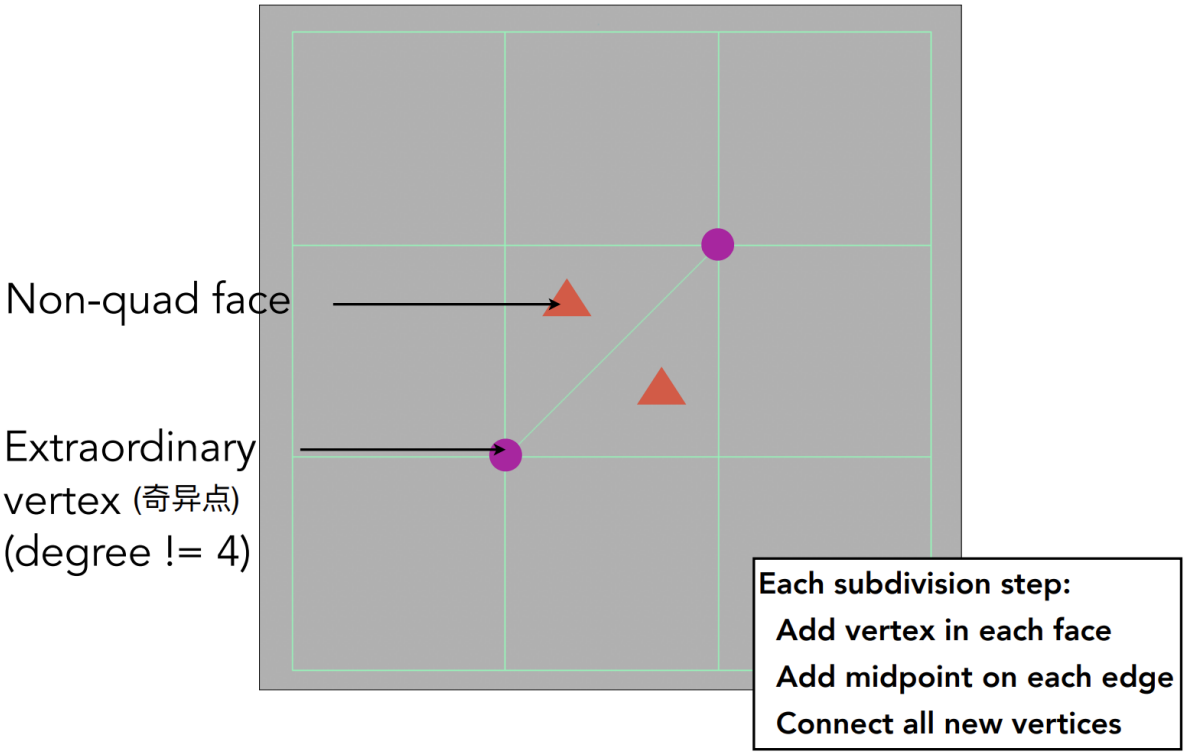
- 奇异点：顶点度不为4
- 其他点

将面分为两类

- 非四边形面
- 四边形面

Catmull-Clark 细分步骤

- 在每个面上添加新顶点
- 将新顶点连接到所在面各边的中点



- **性质**：
  - 在第一次细分后，奇异点不再增加，因为所有非四边形在第一次细分之后都变成了四边形。
  - 新增加奇异点的度数与该非四边形的边数相等。
  - 第一次细分增加的奇异点数目与非四边形的个数相等。

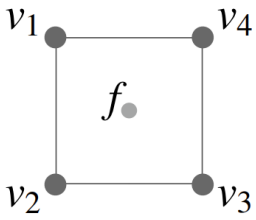
Catmull-Clark 顶点计算步骤 在四边形中计算

- 面点 f

- 边点 m
- 平均面点 p
- 平均边点 v

FYI: Catmull-Clark Vertex Update Rules (Quad Mesh)

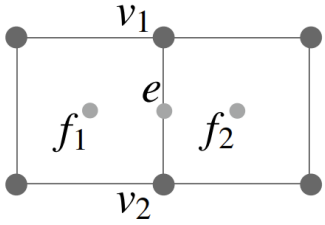
Face point

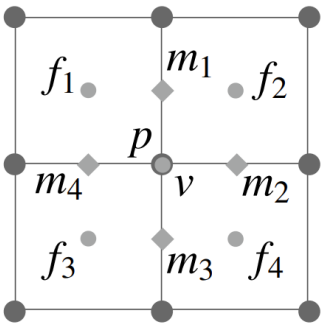


$$f = \frac{v_1 + v_2 + v_3 + v_4}{4}$$

$$e = \frac{v_1 + v_2 + f_1 + f_2}{4}$$

Edge point



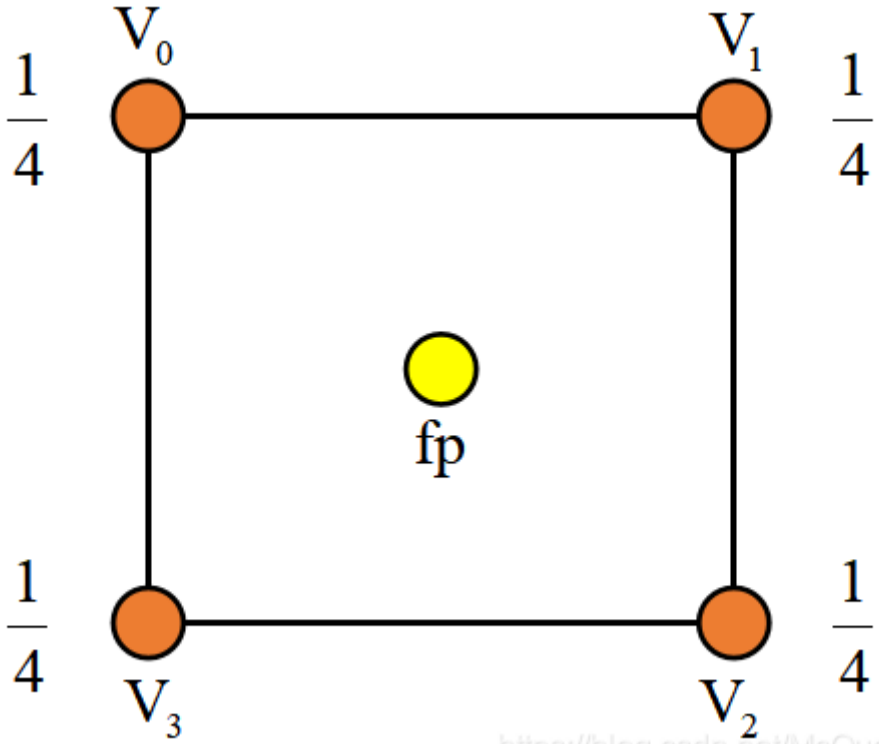


Vertex point

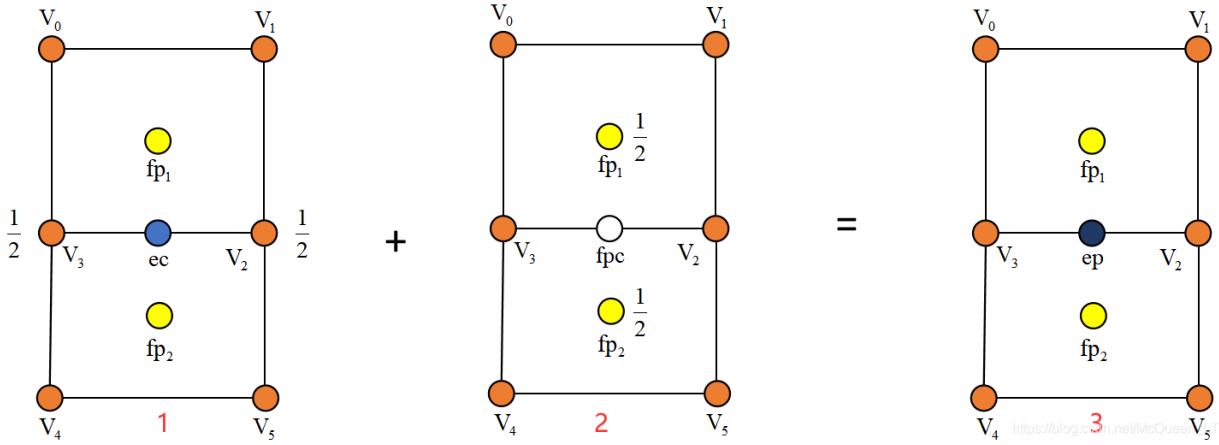
$$v = \frac{f_1 + f_2 + f_3 + f_4 + 2(m_1 + m_2 + m_3 + m_4) + 4p}{16}$$

m midpoint of edge
p old "vertex point"

- 面点：$fp = \frac{1}{4}(V_0 + V_1 + V_2 + V_3)$



- 边点：

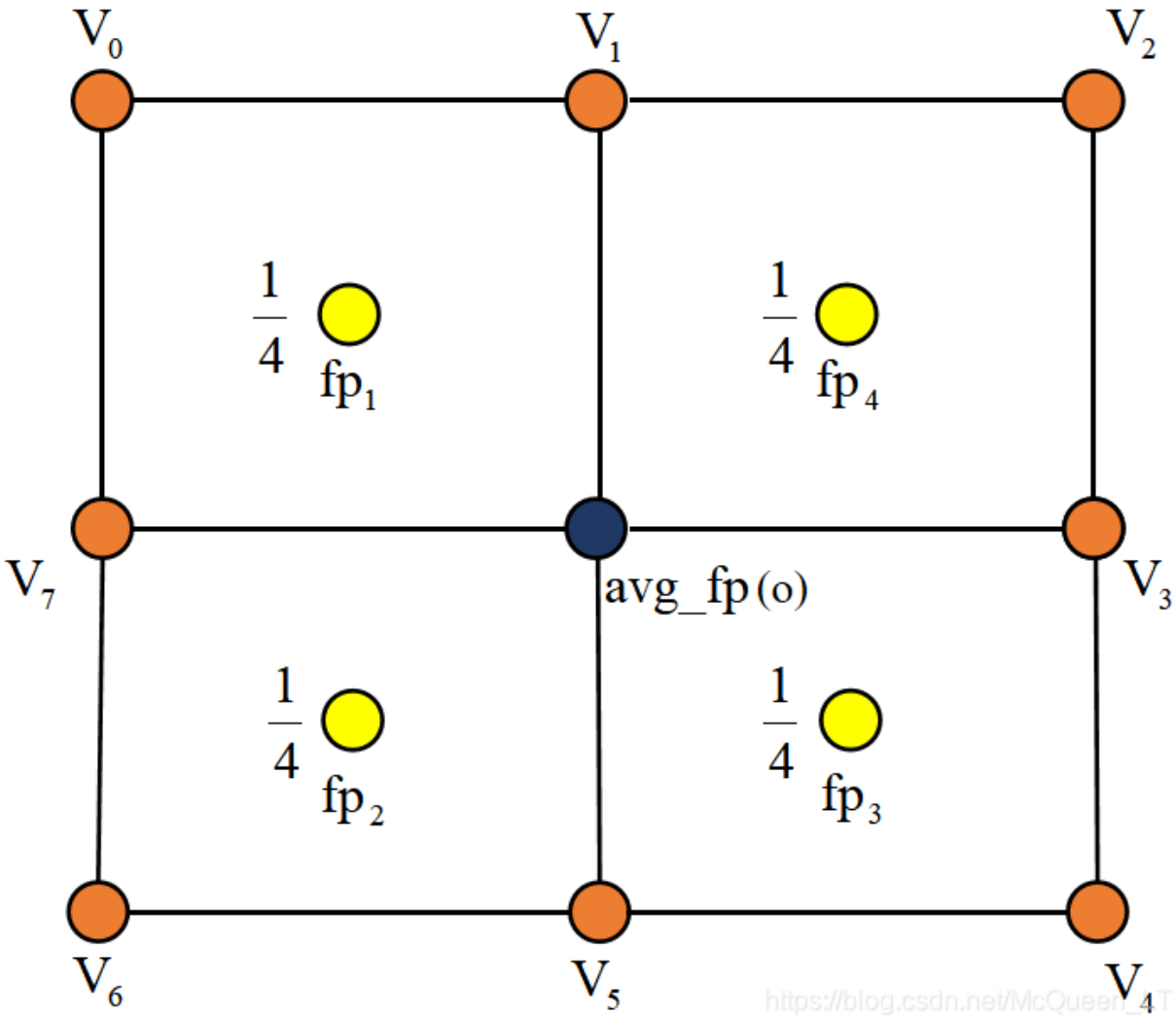


$$ep = \frac{1}{2}(ec + fpc)$$

$$ec = \frac{1}{2}(V_3 + V_2)$$

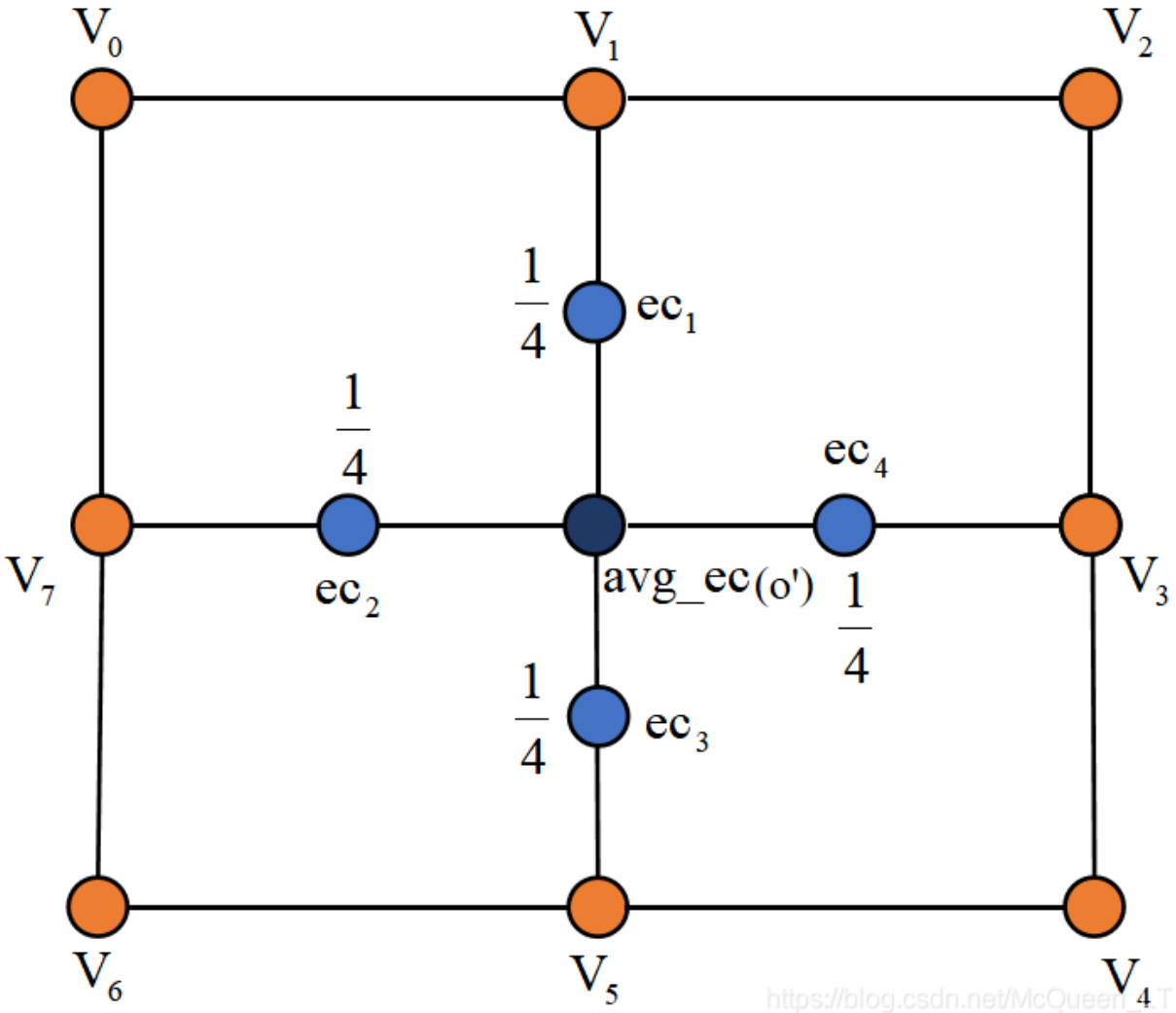
$$fpc = \frac{1}{2}(fp_1 + fp_2)$$

- 平均面点 O 相邻的面有 f_1, f_2, f_3, f_4 一共四个，如下图所示，计算面中点公示



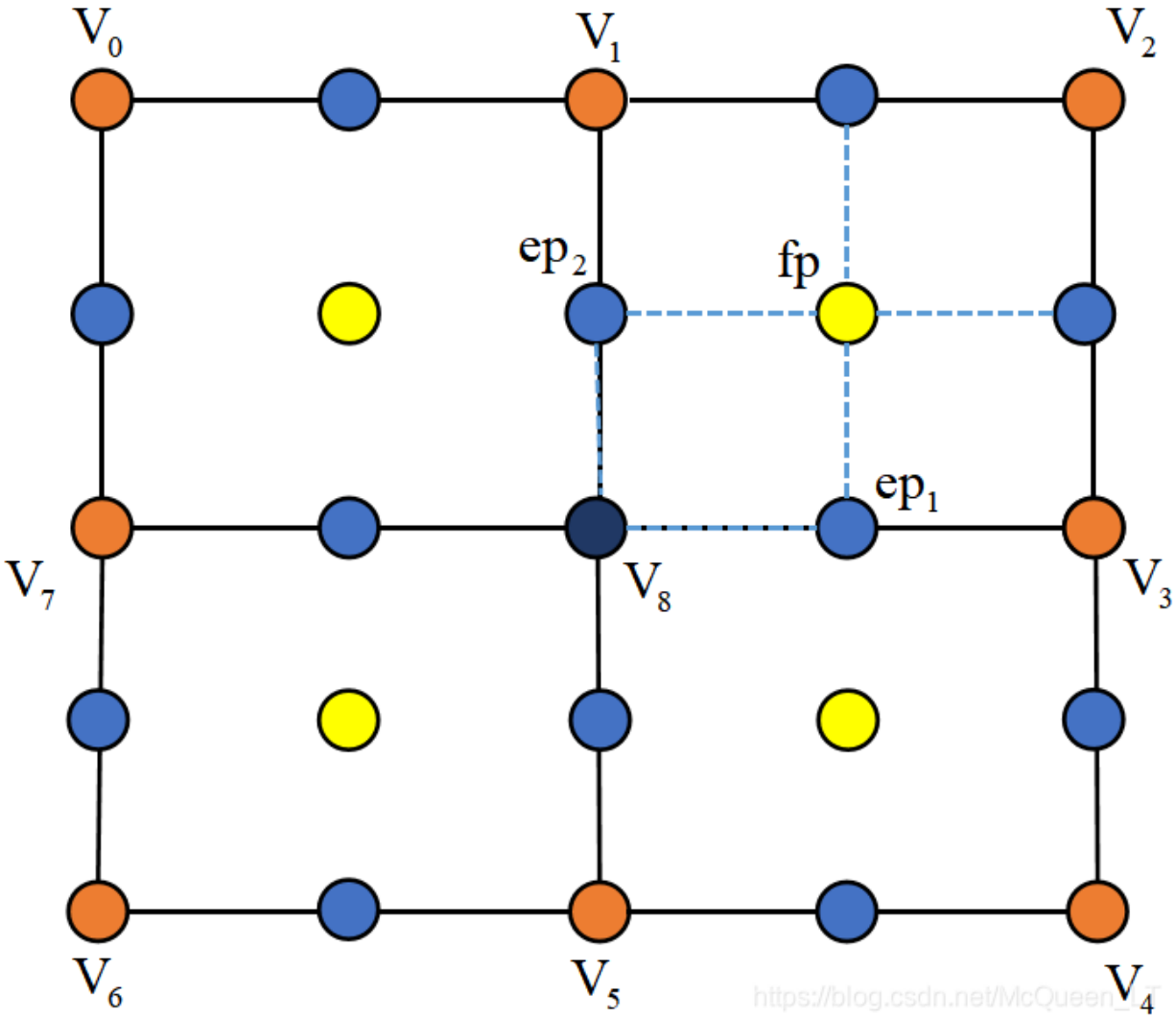
$$avg_{fp} = \frac{1}{4}(fp_1 + fp_2 + fp_3 + fp_4)$$

- 平均边中点 O‘ 有四个相邻面，计算这四个面边中点的平均值。



$$avg_{ec} = \frac{1}{4}(ec_1 + ec_2 + ec_3 + ec_4)$$

- 细分结果 将每个面细分出面点，边点，以及与其他面相交边上的边中点与四个面相交的面点，可以得出细分结果



Loop细分与Catmull-Clark细分之间的区分

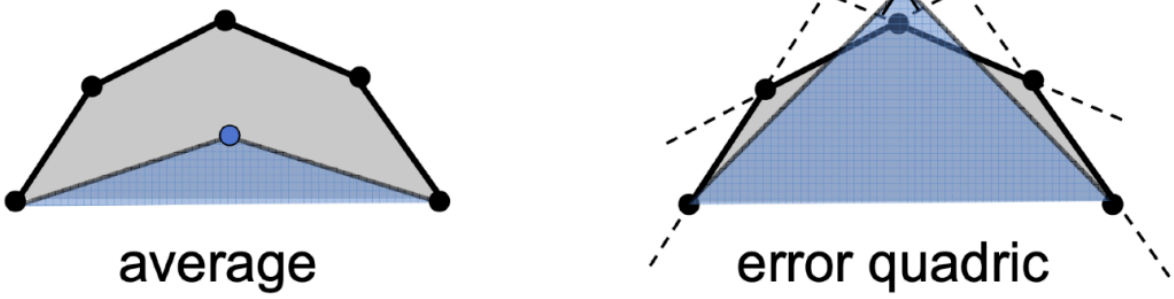
- Loop细分只能用于三角形的细分
- 而Catmull-Clark细分多用于四边形的细分。

3.Mesh Simplification 简化

网格简化的目的旨在减少三角形数量的同时维持物体外形不变。 本节使用 **edgecollapsing 边坍缩** 来进行网格简化

1.Quadric Error Metrics 二次误差度量

与机器学习中的 L2误差 概念相同，网格简化中对边坍缩过后的点的评估使用与周边点距离平方和。

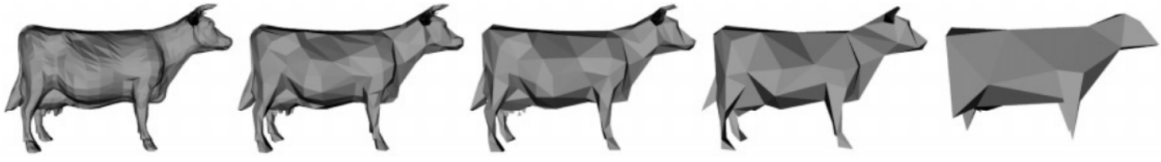


简化算法

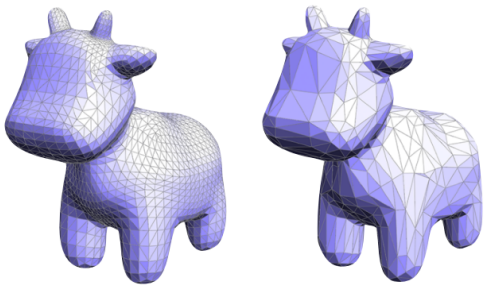
- 需要选取 L2误差 最小的点进行简化，
- 并且需要动态更新其他受影响点的 L2 误差
- 此处需要 **堆/优先队列** 的数据结构实现。

简化结果

- 使用二次误差度量的简化中，需要多个三角形构成的结构，在简化中通常不会被简化掉，比如下图中奶牛的背部。



Garland and Heckbert '97



4.Shadow Mapping

阴影的产生需要根据光源，物体，物体之间位置关系来确定，总结为

- 在能被摄像机和光源同时看到的点不会产生阴影。

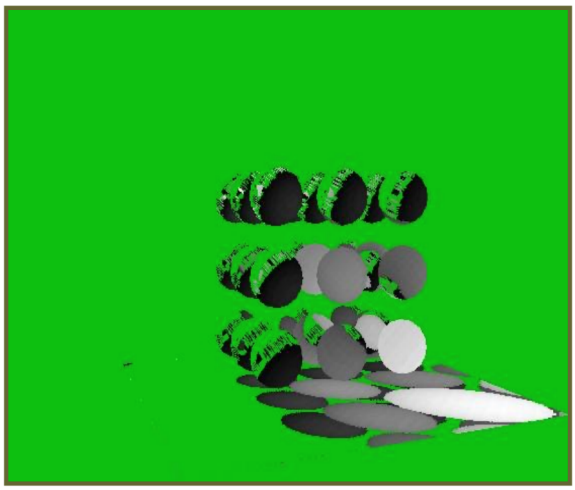
阴影算法

- 从光源视角获取深度图
- 从摄影机获取不同点，并投影回光源，获取新的深度图
- 比较两个深度图，如果两者一致说明 **可视**，反之 **不可视**

该算法的问题

- 高质量的阴影需要高分辨率的 shadow map
- 在判断两个深度图相等时，浮点数会给相等造成麻烦，如下图气球上绿色并没有覆盖完全。

Green is where the distance(light, shading point) ≈ depth on the shadow map



Non-green is where shadows should be