第9讲：几何

中国海洋大学 周佳然

上次课程内容

- 几何的多种表示形式
  - 隐式
  - 显式
- 贝塞尔曲线
- 贝塞尔曲面
- 学习体验调查

本次课程内容

- 几何处理(Geometry Processing)
  - 什么是几何处理?
  - 为什么要做几何处理?
  - 几何处理的常见任务有哪些?
- 阴影贴图（Shadow Mapping）
- 实验4发布（包含实验报告要求）
- 实验3提交截止时间延期一周
- 实验1-3报告提交截止时间延期一周

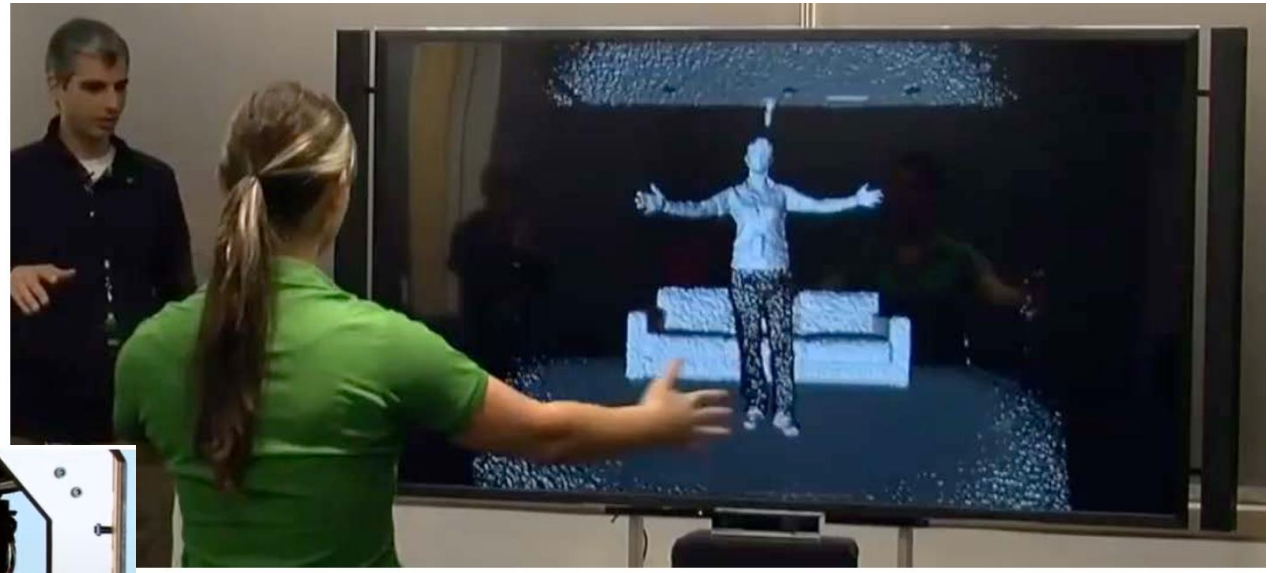
什么是几何处理？

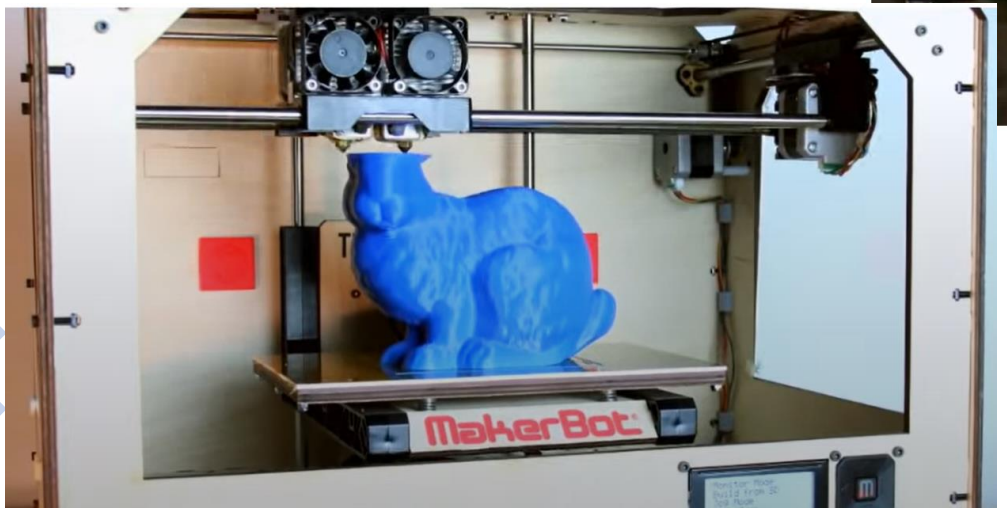
- 对传统数字信号（音频、视频等）处理的扩展，用于对几何信号的处理
  - 上采样/下采样/重采样/过滤……
  - 走样

为什么要做几何处理？

- 几何处理是计算机图形学许多领域（渲染、动画等等）的基石

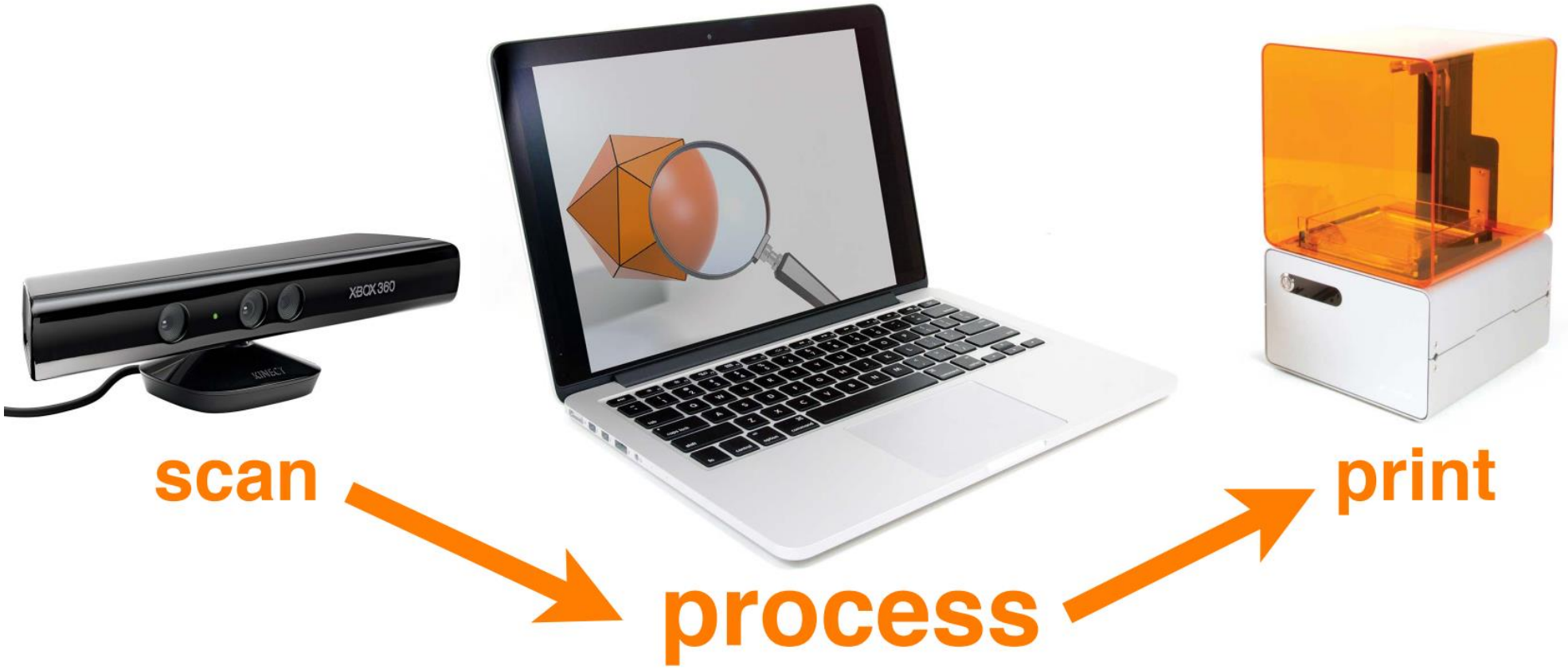
三维扫描（3D Scanning）



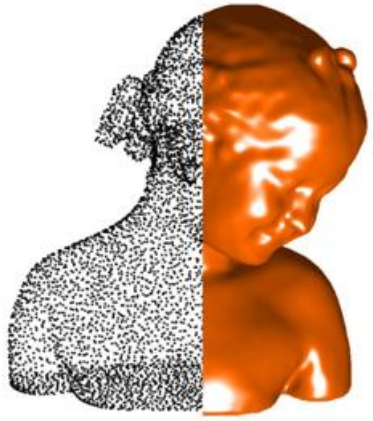


三维打印（3D Printing）

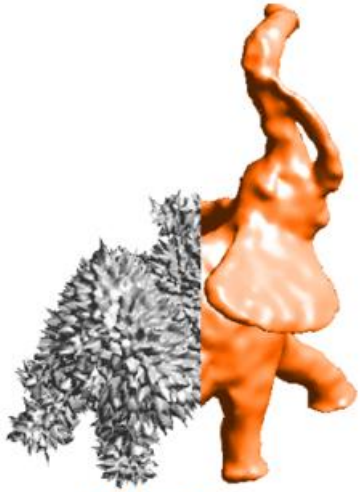
几何处理管线（Geometry Processing Pipeline）



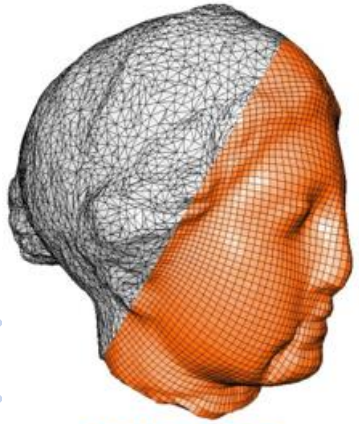
几何处理的常见任务



reconstruction



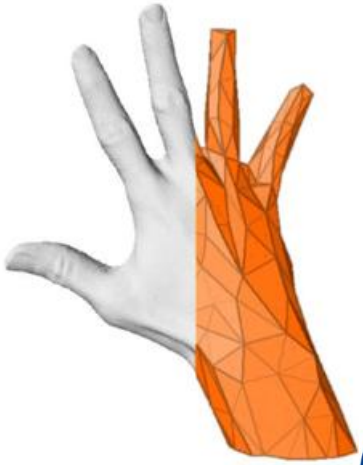
filtering



remeshing

shape analysis

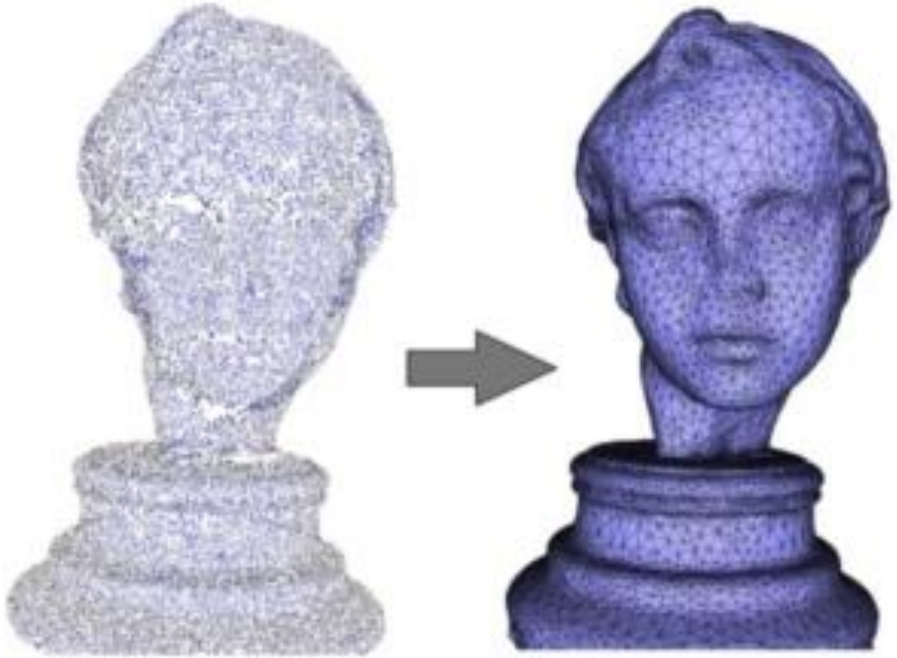
parameterization



compression

几何处理：重建（reconstruction）

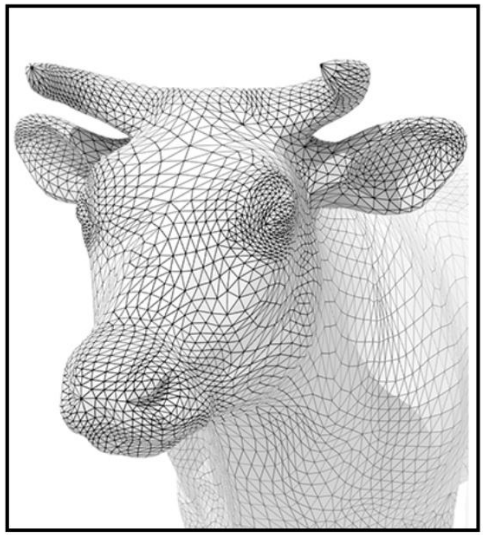
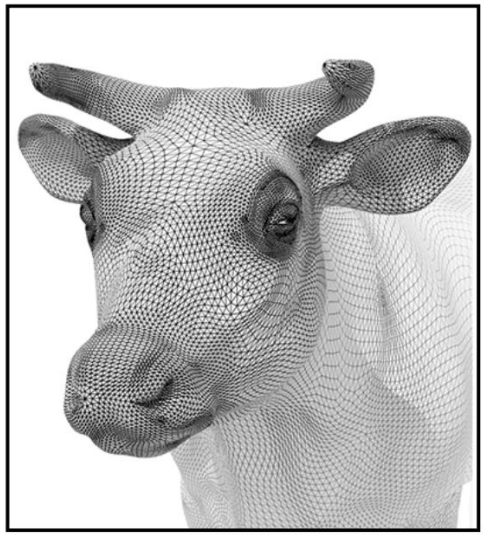
- 给定几何样本，重建表面
- 样本：
  - 点、点和法线
  - 图像的集合（多视图）
  - 线密度积分（MRI、CT扫描）
- 重建表面的技术：
  - 基于轮廓
  - 基于Voronoi
  - 基于PDE
  - 基于等值面



几何处理：上采样（upsampling）

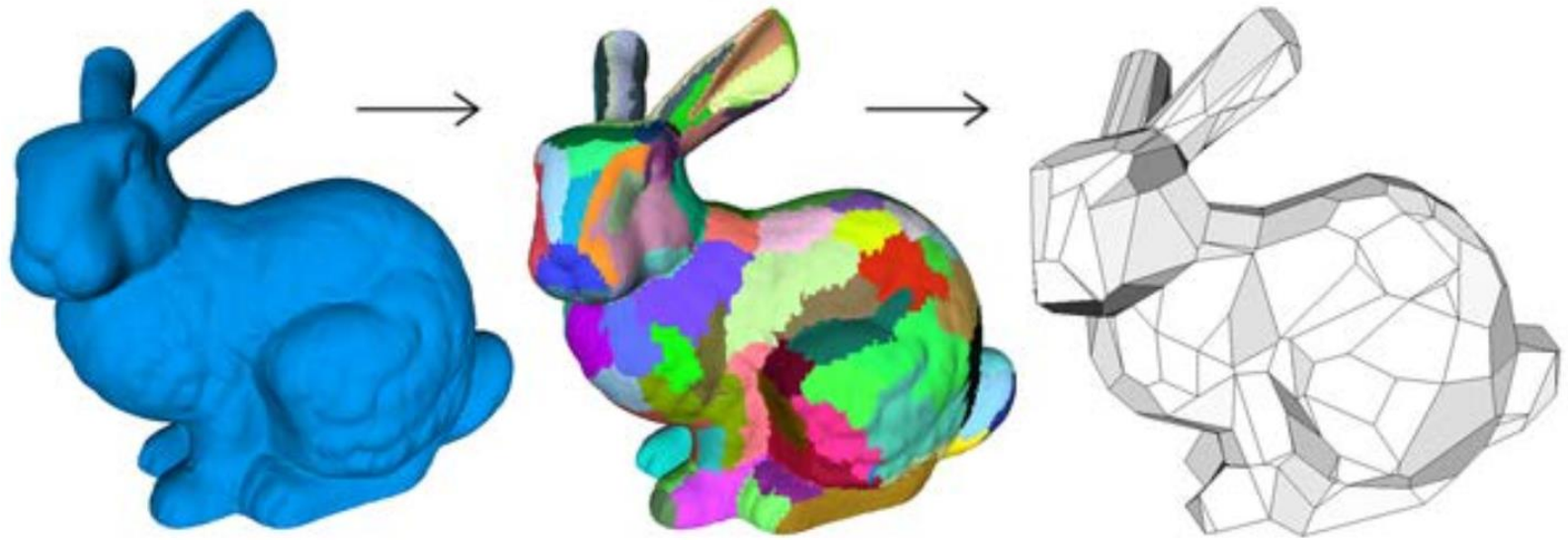
- 通过插值提高分辨率
- 图像：双线性插值、双三次插值
- 网格：网格细分（mesh subdivision）



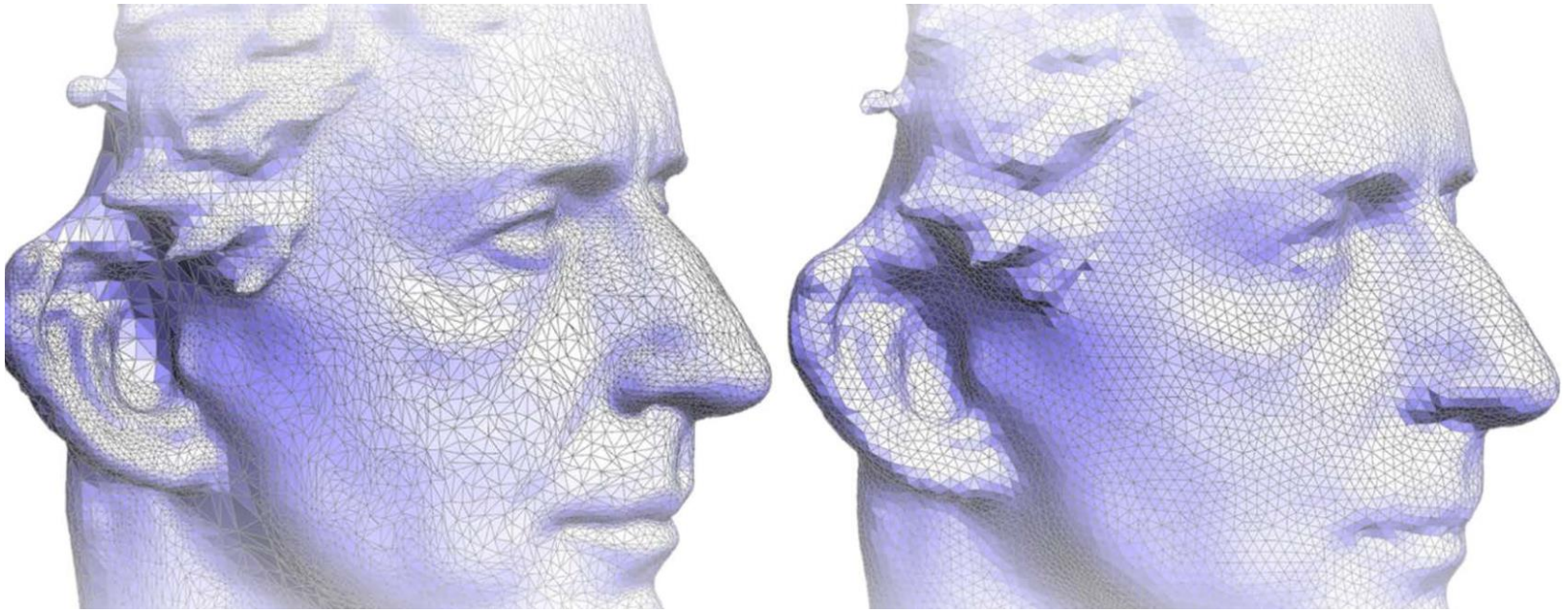
几何处理：下采样(downsampling)

- 降低分辨率，同时保持几何体的形状/外观
- 图像：最近邻、双线性插值、双三次插值
- 点云：二次采样（subsampling）
- 网格：网格简化（mesh simplification）



几何处理：重采样(resampling)

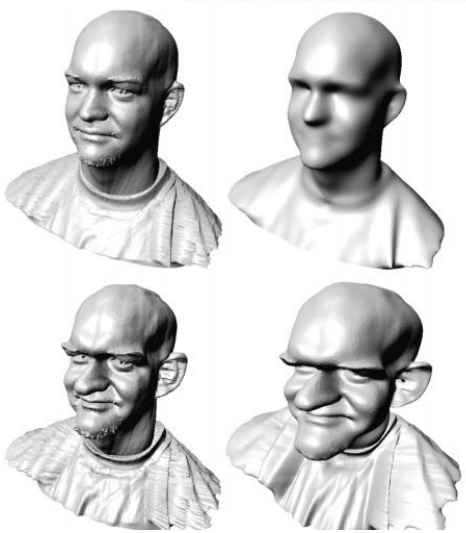
- 修改样本分布以提高质量（保持多边形的数目不变）
- 图像：不是问题！（像素始终存储在规则网格上）
- 网格：多边形的形状十分重要！对于质量的定义依赖具体的任务。

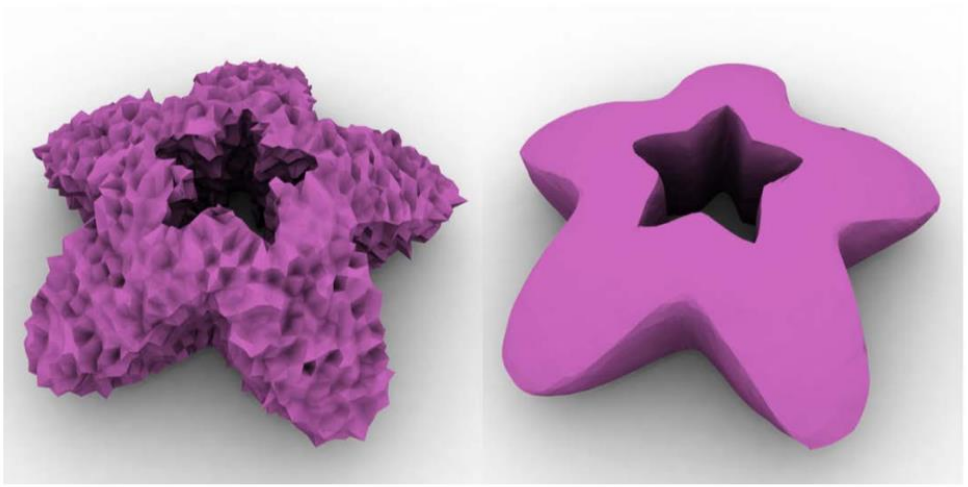


可视化 v.s. 求解方程

几何处理：过滤(filtering)

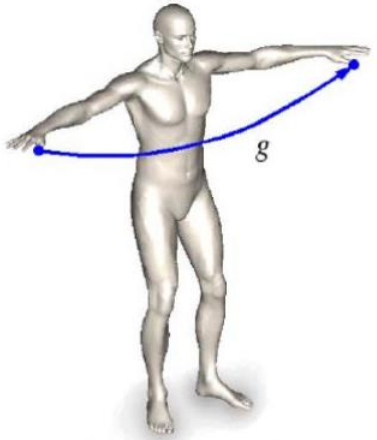
- 消除噪声或强调重要特征





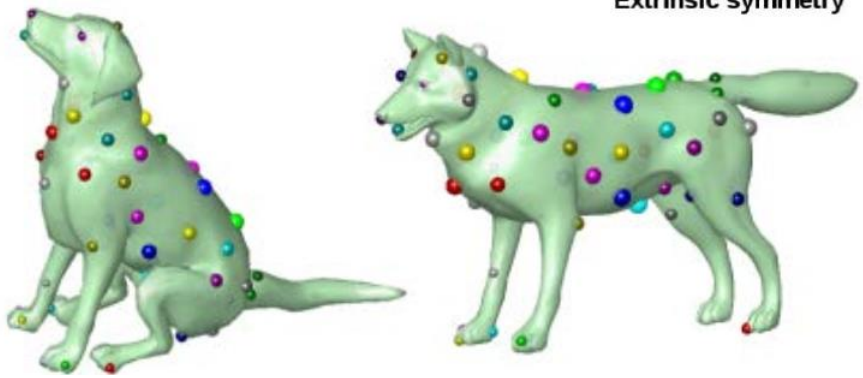
几何处理：形状分析(shape analysis)

- 识别/理解重要的语义特征



Extrinsic symmetry

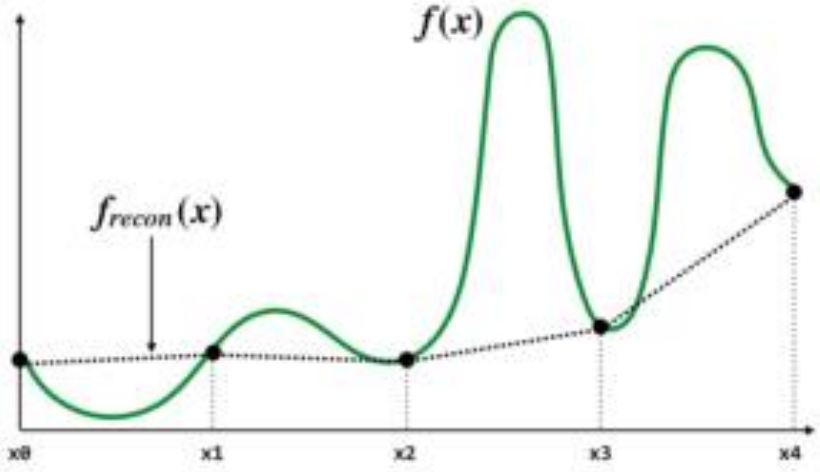
Intrinsic symmetry



Q&A

具体的几何处理方法举例

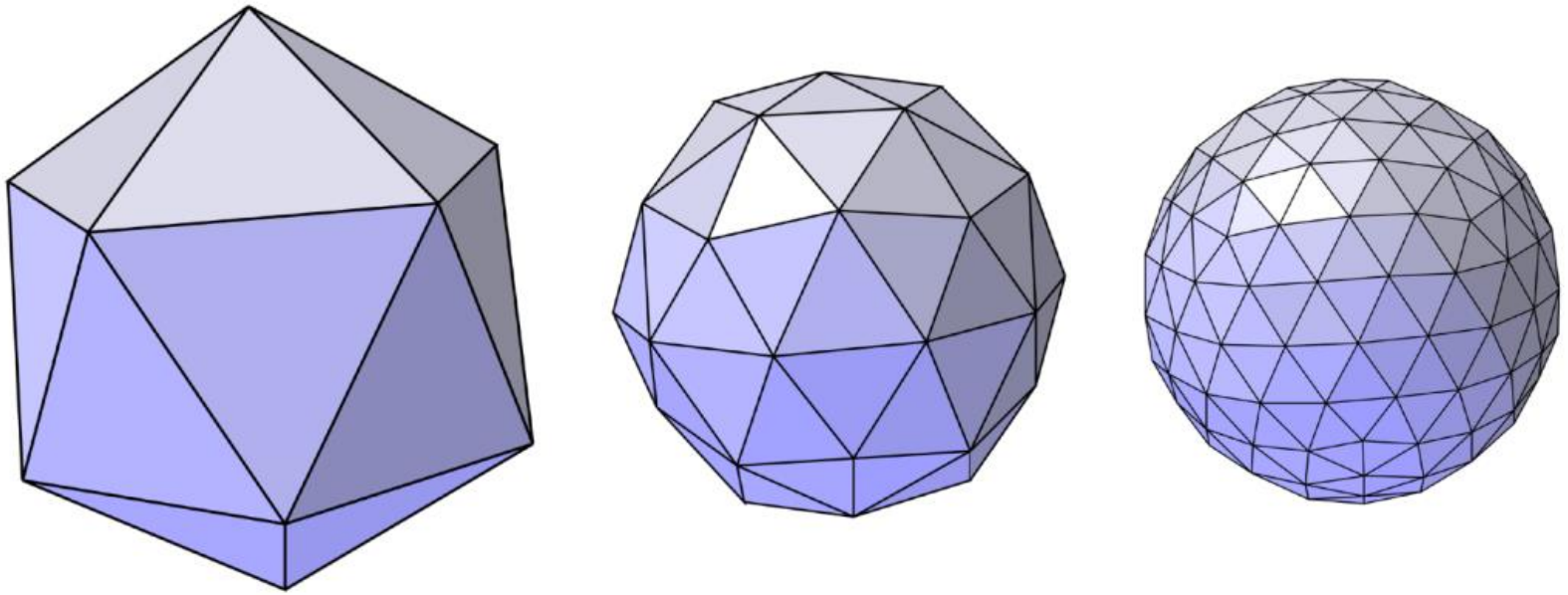
- 回忆我们关于走样的理解
  - 不恰当的采样会使信号看起来与实际不符
  - 例如：欠采样使曲线变得平滑



- 几何上是同样的！
  - 欠采样会破坏特征
  - 过采样影响性能

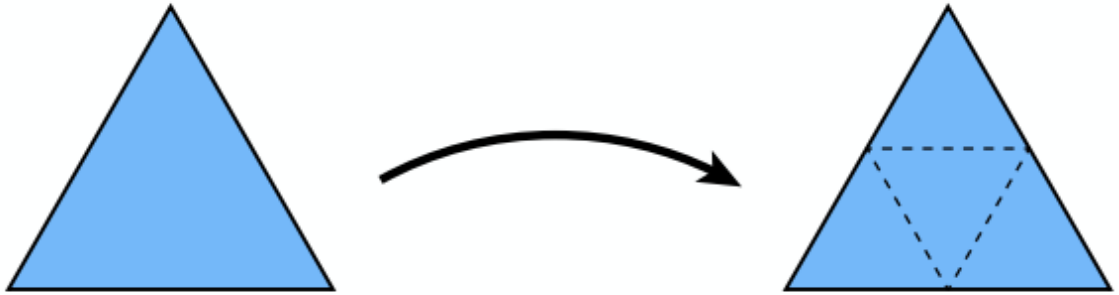
网格细分

- Loop细分
  - 针对三角形网格的常用细分方法
  - 首先增加更多的三角形（顶点）；然后调整顶点的位置

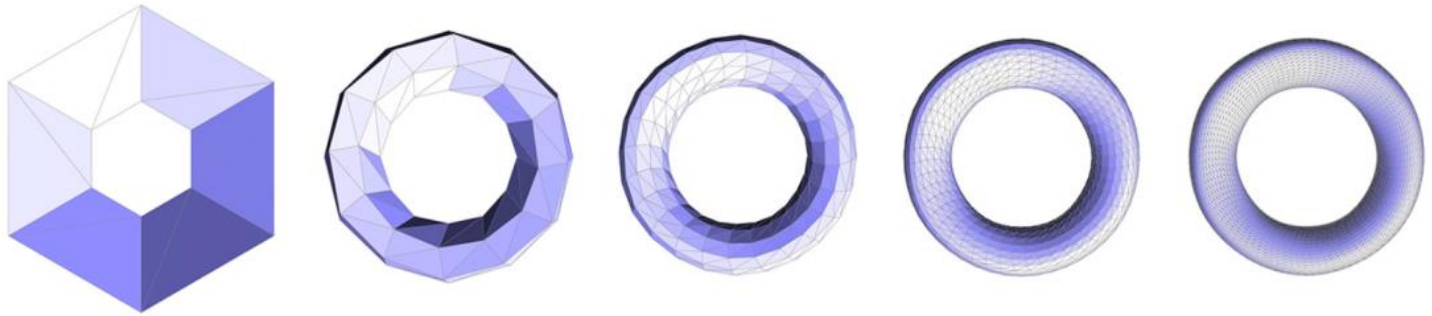


Loop细分

- 把每个三角形一分为四

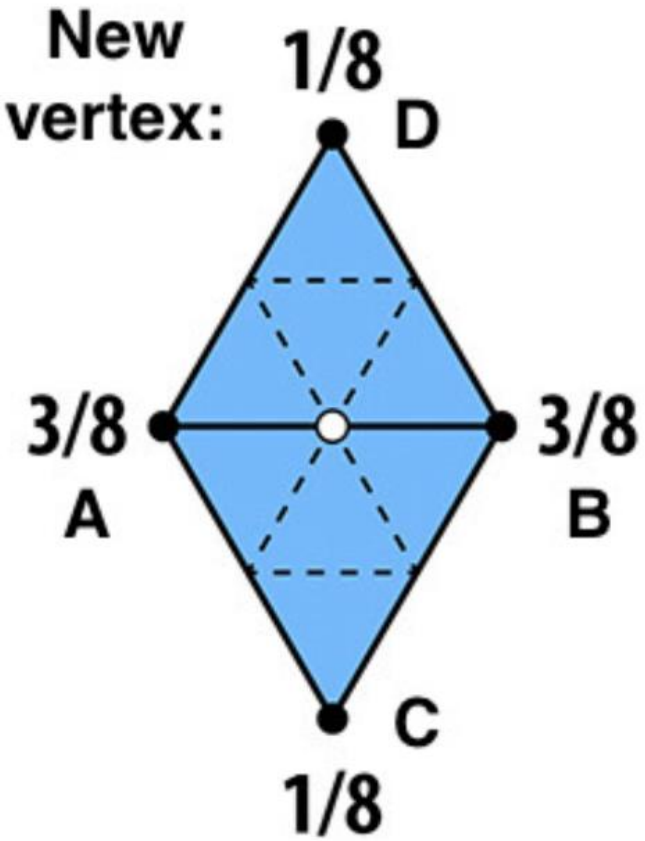


- 根据不同权重分配顶点位置（新旧顶点有区别）



Loop细分

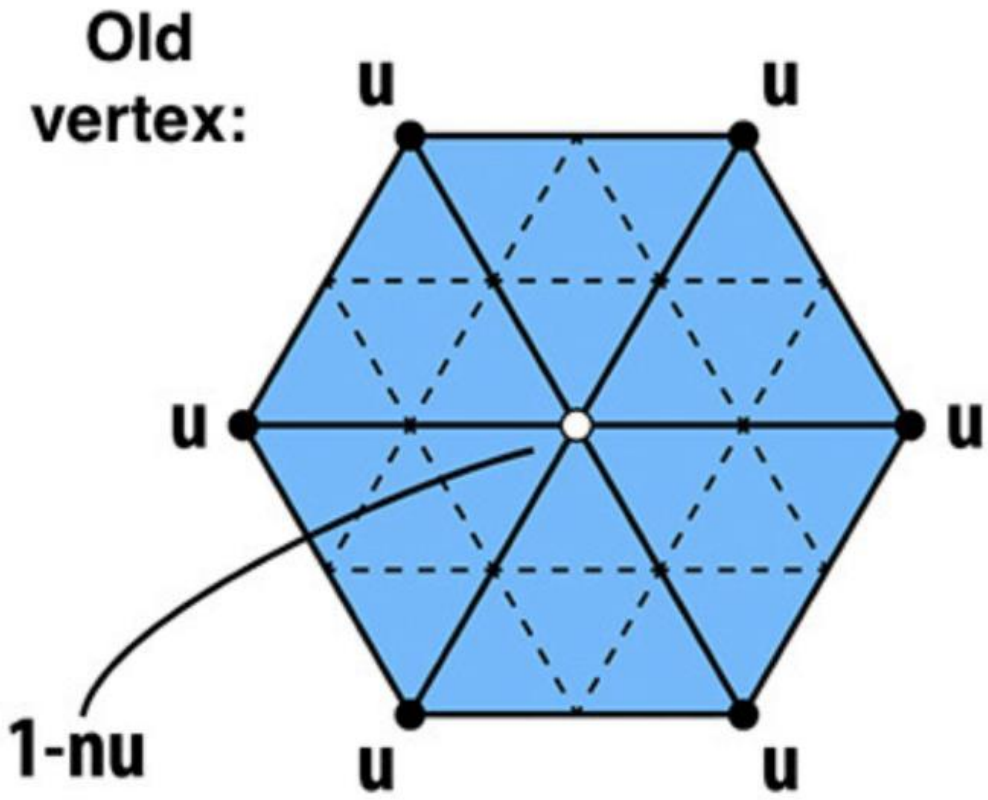
- 对于新顶点



$$3/8 * (A + B) + 1/8 * (C + D)$$

Loop细分

- 对于旧顶点



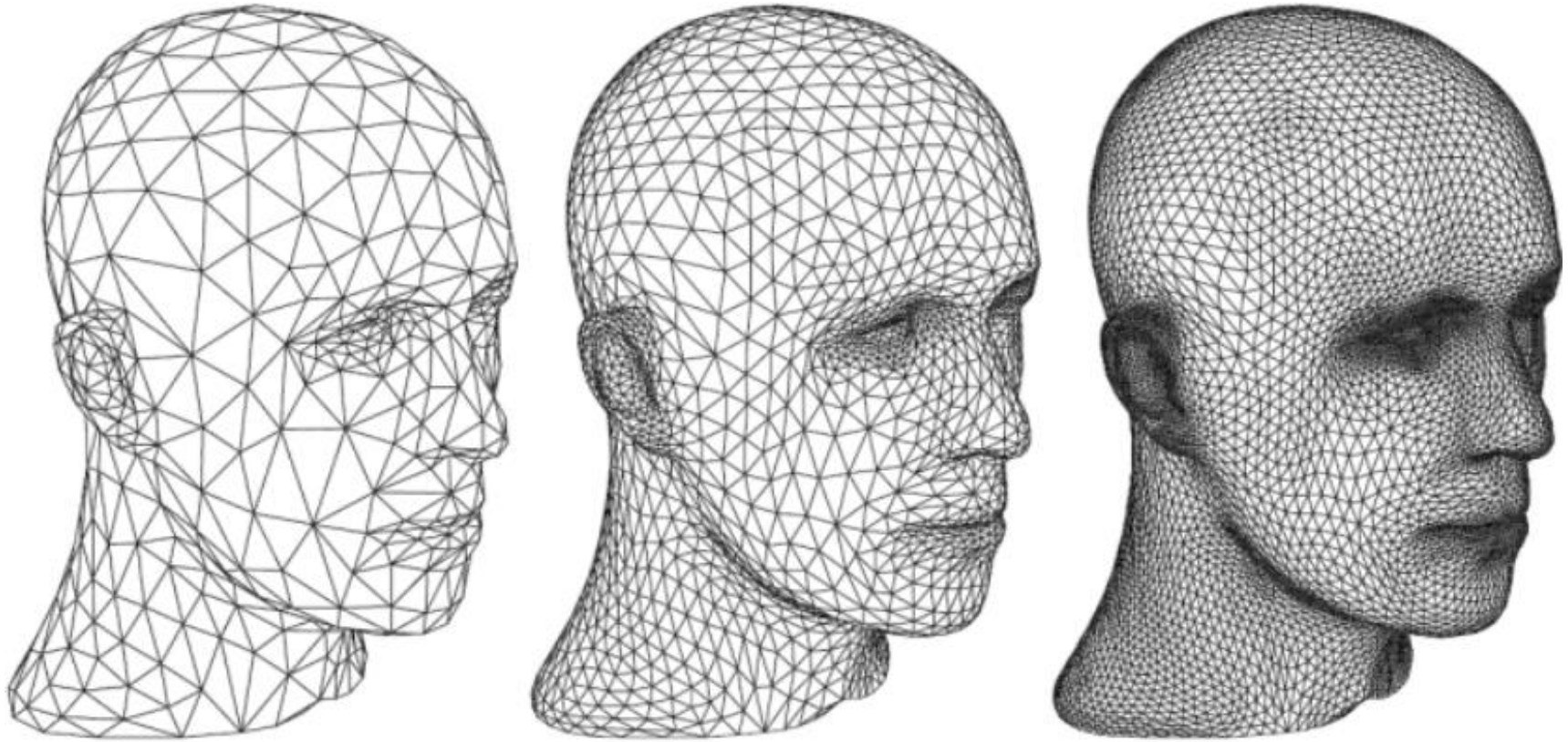
(1 - n*u) * original_position +
u * neighbor_position_sum

n: vertex degree

u: 3/16 if n=3, 3/(8n) otherwise

Loop细分

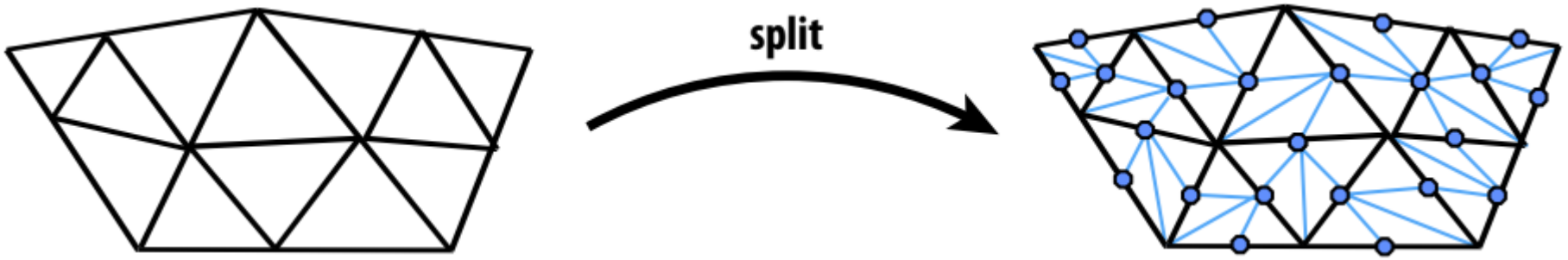
- 效果举例



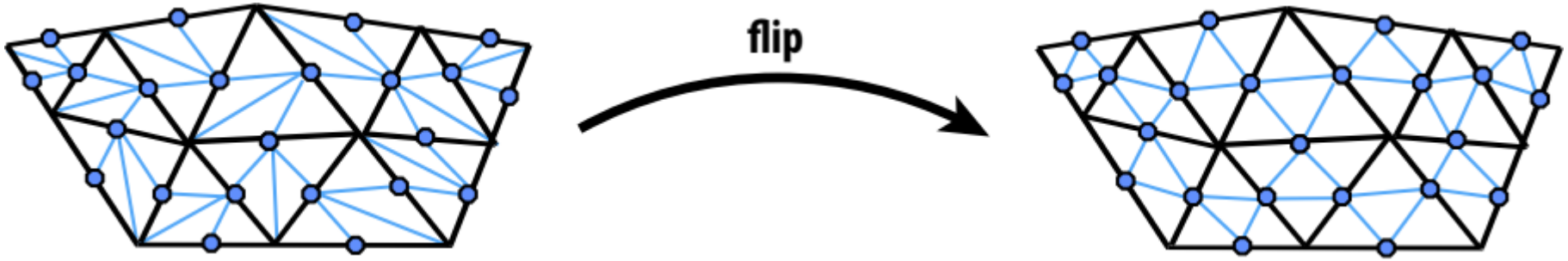
Loop细分

- 通过边操作进行Loop细分

➢ 首先以任意顺序分割原始网格的边



➢ 然后翻转接触新旧顶点的新边



➢ 最后更新顶点位置

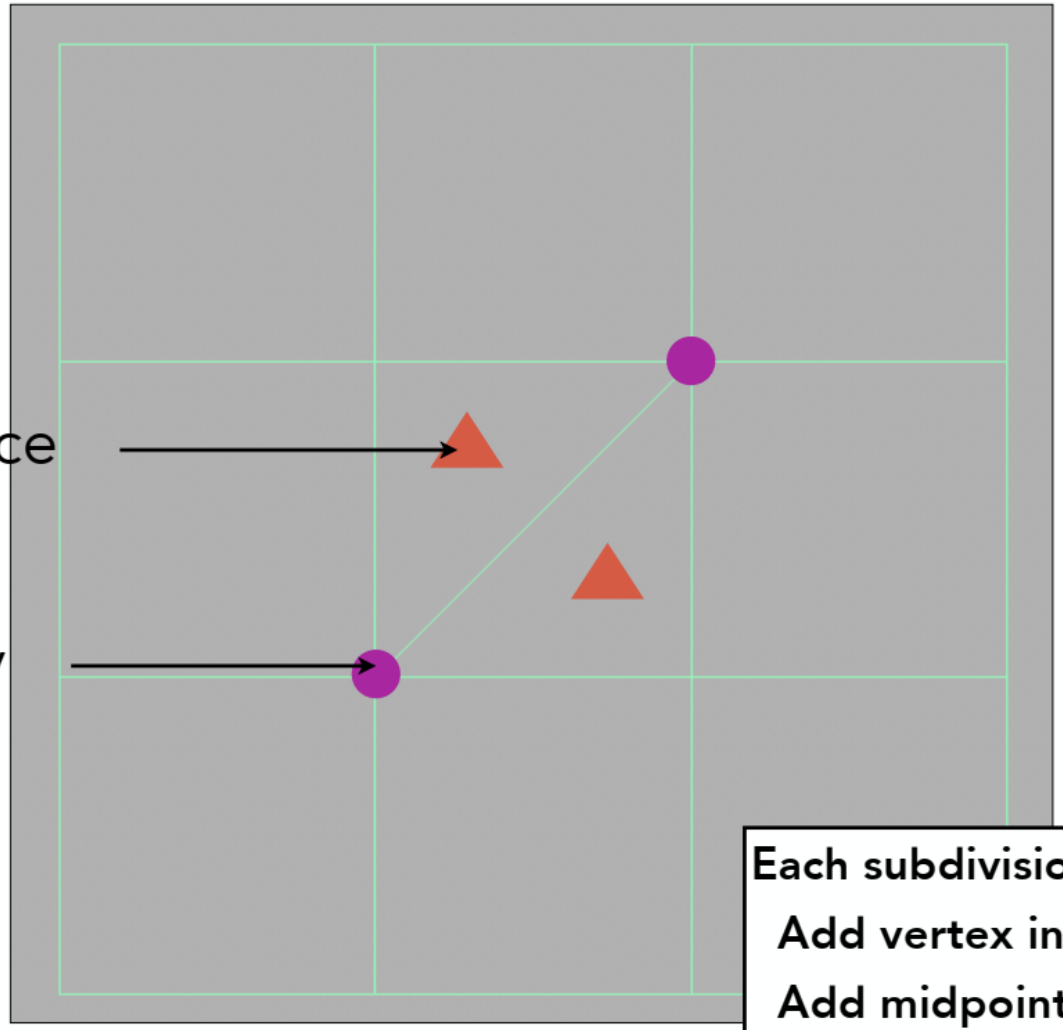
网格细分

- CatMull-Clark细分

➢ 针对多边形网格,多用于四边形网格

Q:为什么用在四边网格上?

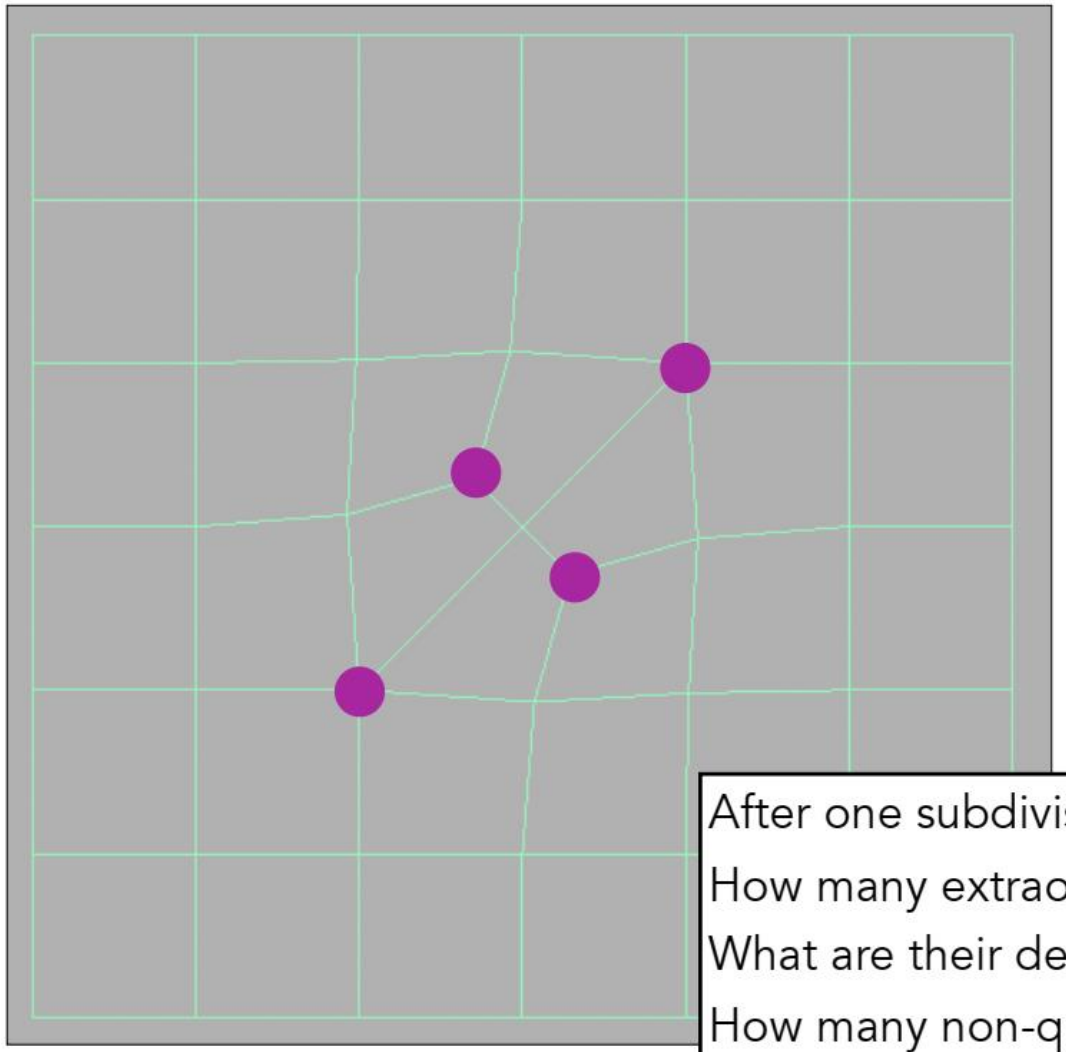
CatMull-Clark细分



Each subdivision step:

- Add vertex in each face
- Add midpoint on each edge
- Connect all new vertices

CatMull-Clark细分



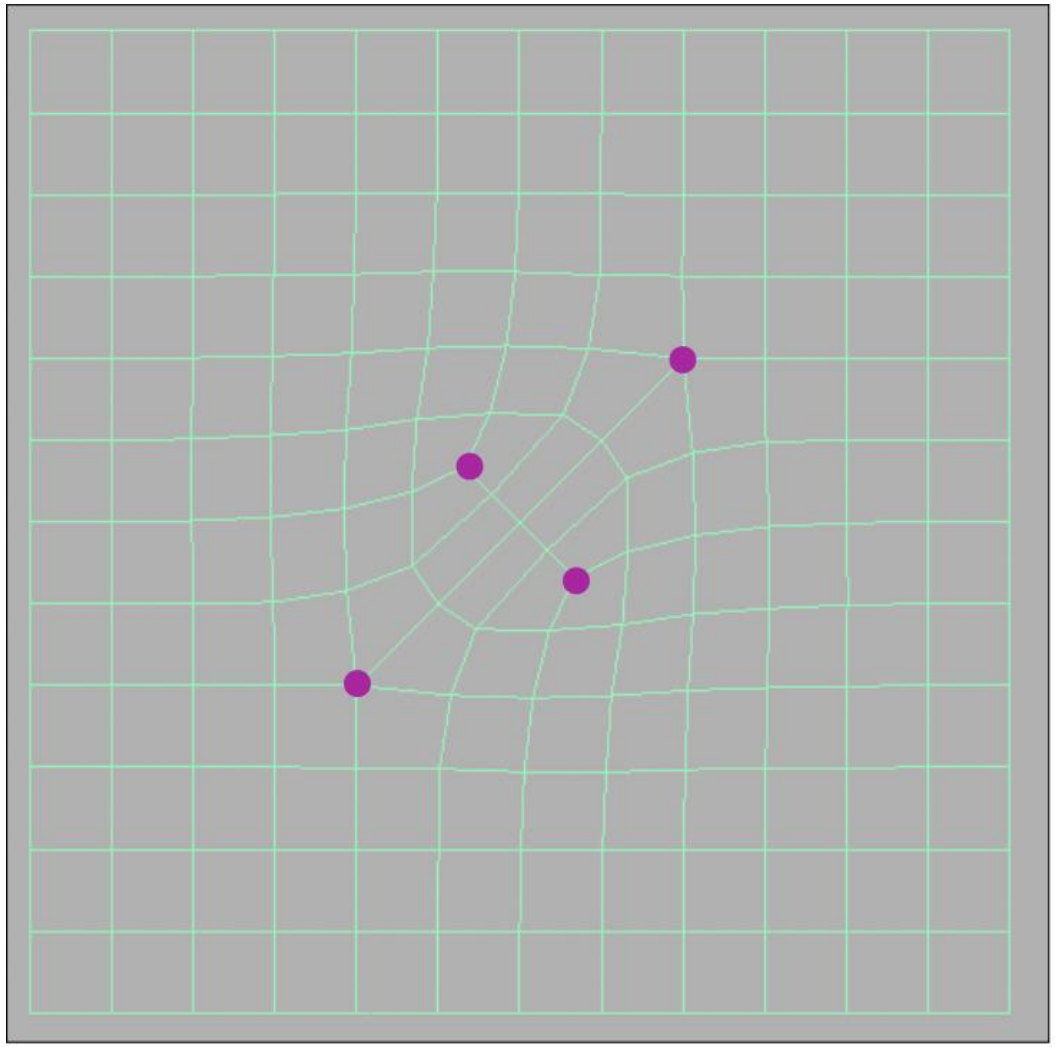
After one subdivision:

How many extraordinary vertices?

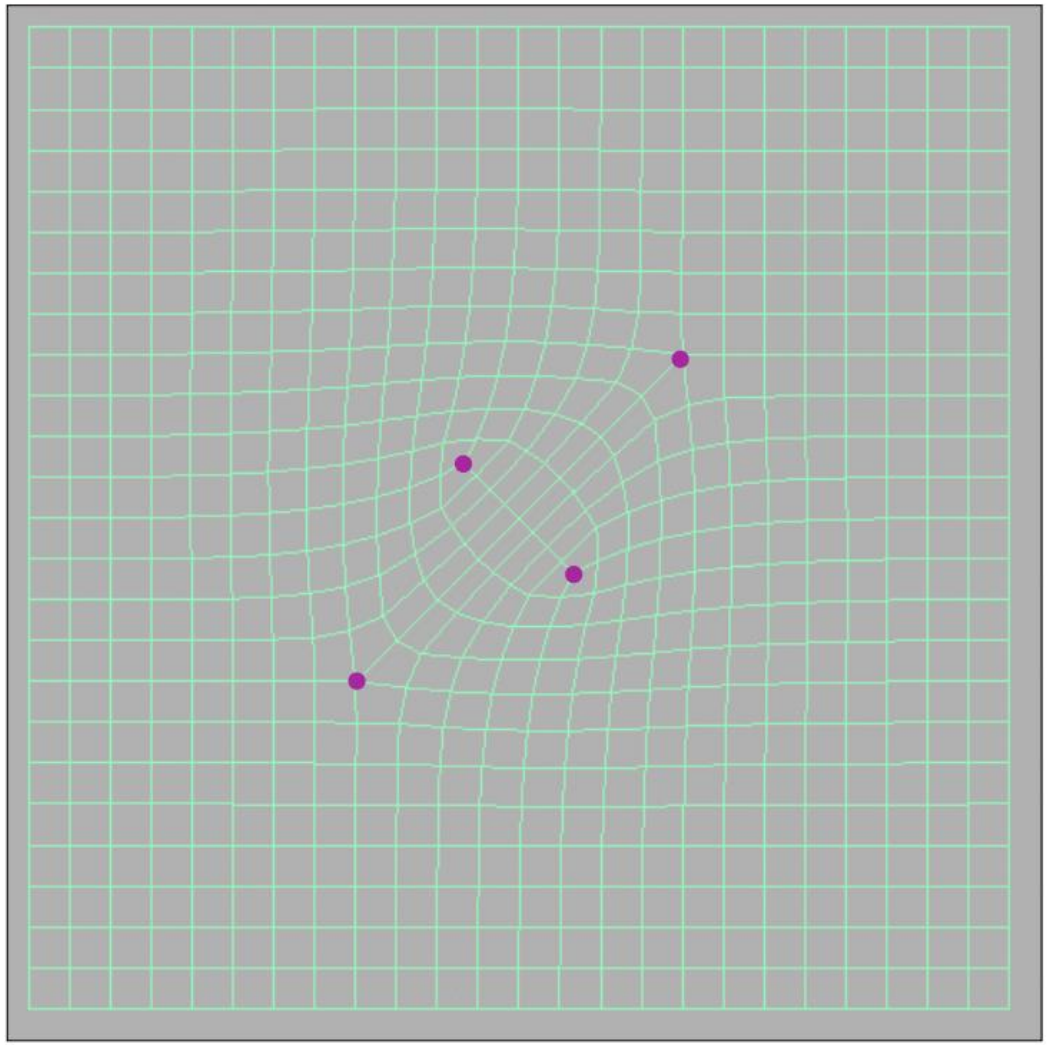
What are their degrees?

How many non-quad faces?

CatMull-Clark细分

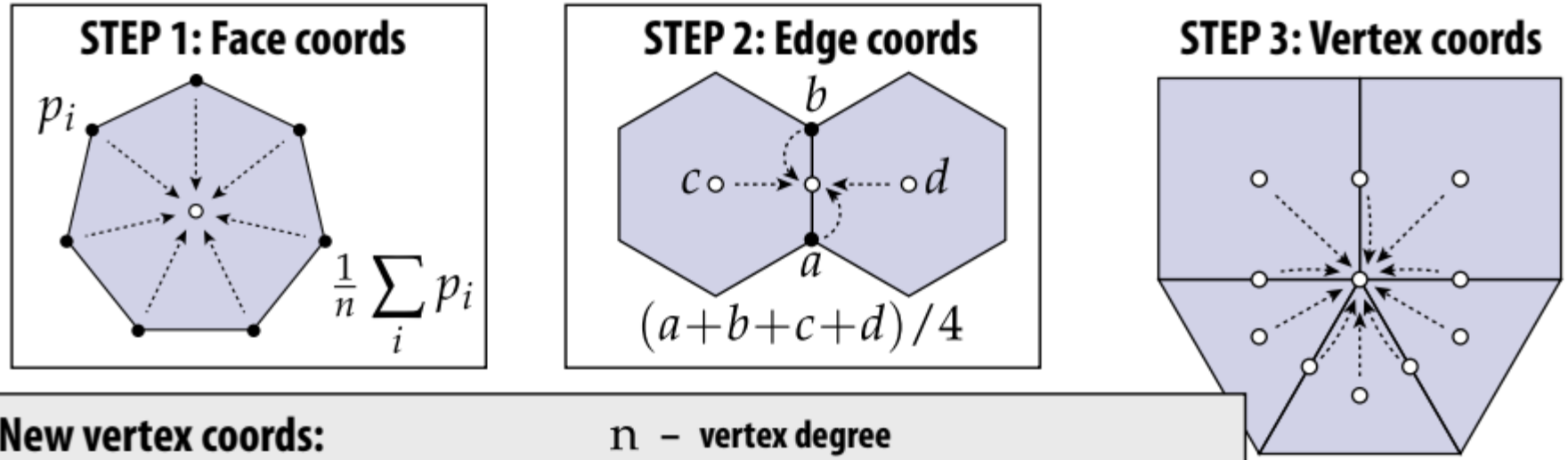


CatMull-Clark细分



CatMull-Clark细分

- 顶点位置



New vertex coords:

$$\frac{Q+2R+(n-3)S}{n}$$

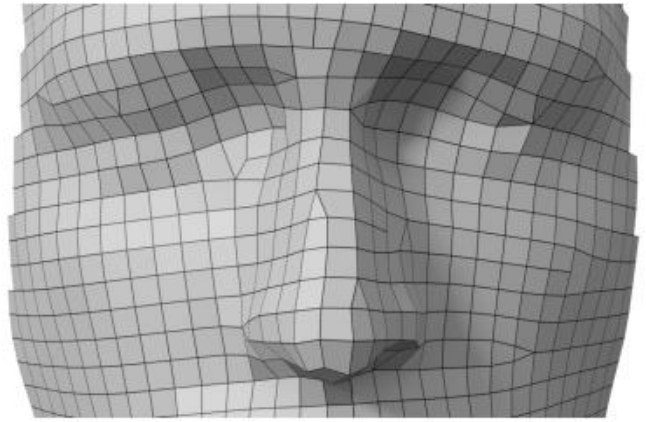
n – vertex degree

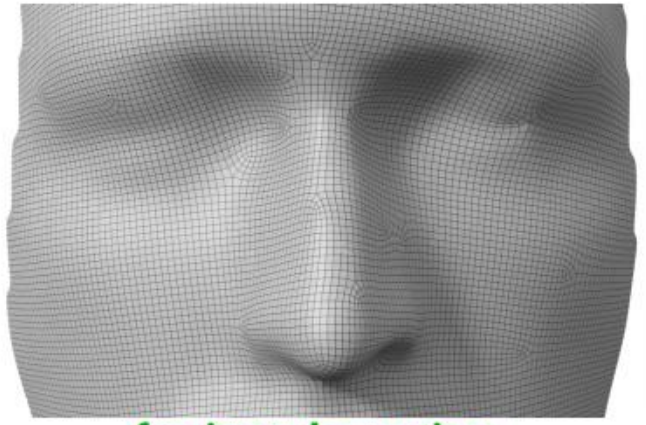
Q – average of face coords around vertex

R – average of edge coords around vertex

S – original vertex position

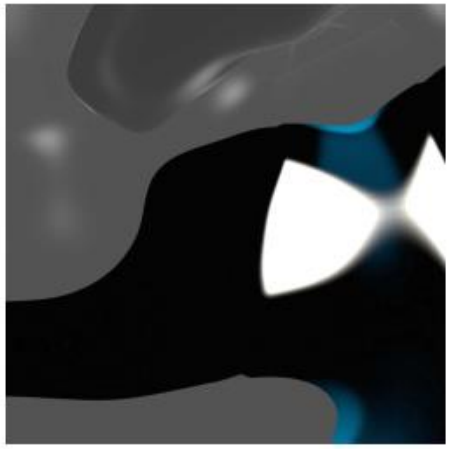
CatMull-Clark细分



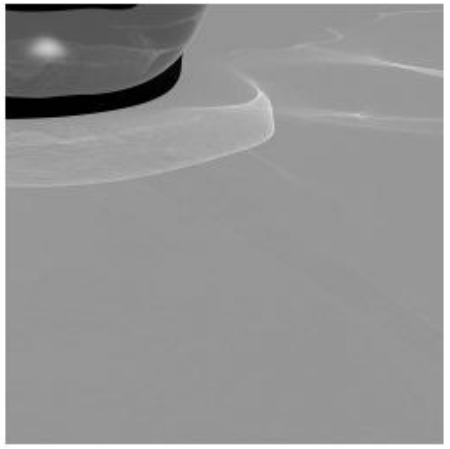


few irregular vertices

$\Longrightarrow$ smoothly-varying surface normals

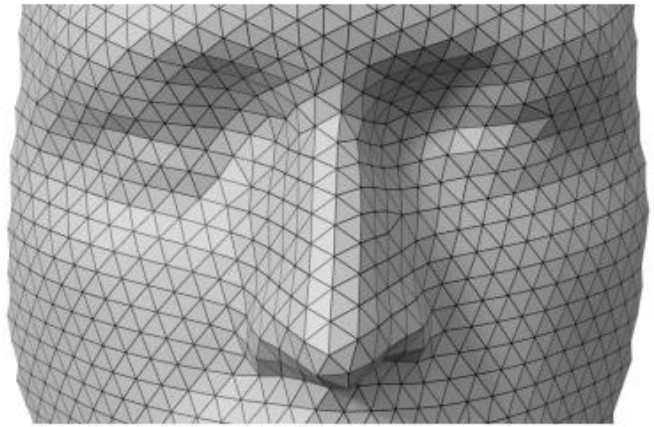


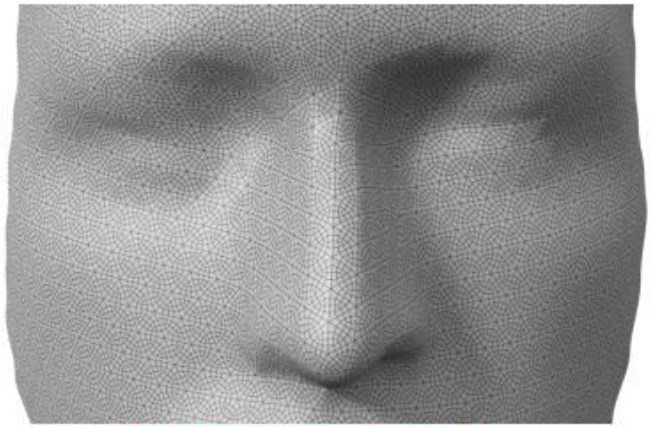
smooth
reflection lines



smooth
caustics

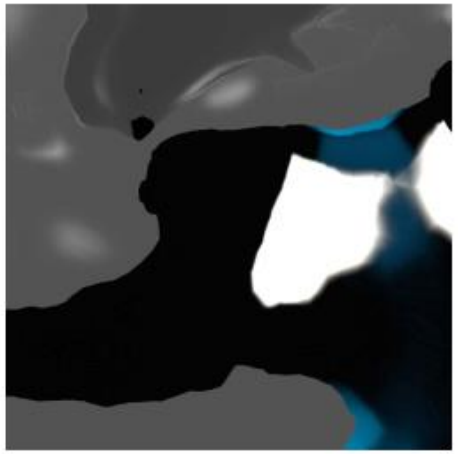
CatMull-Clark细分



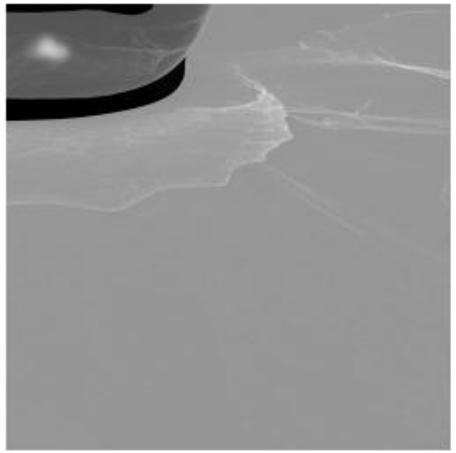


many irregular vertices

$\Longrightarrow$ erratic surface normals



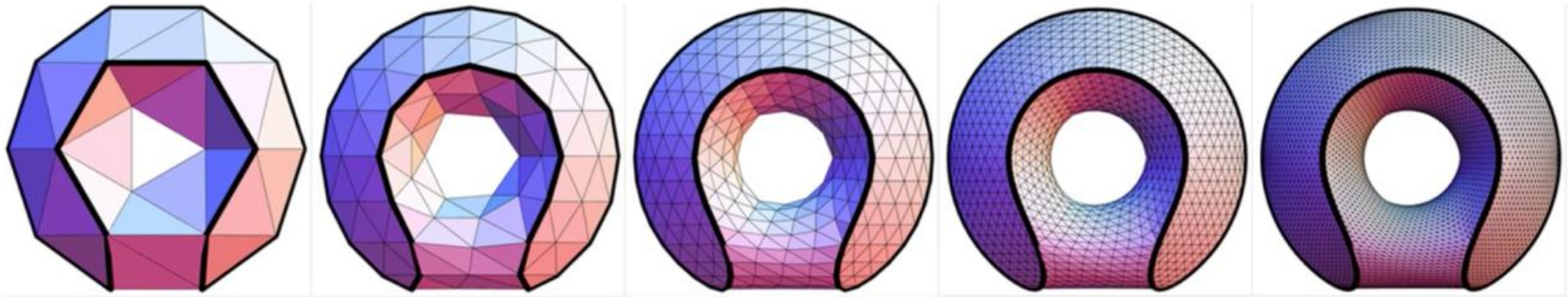
jagged
reflection lines



jagged
caustics

网格细分

Loop with Sharp Creases



Catmull-Clark with Sharp Creases

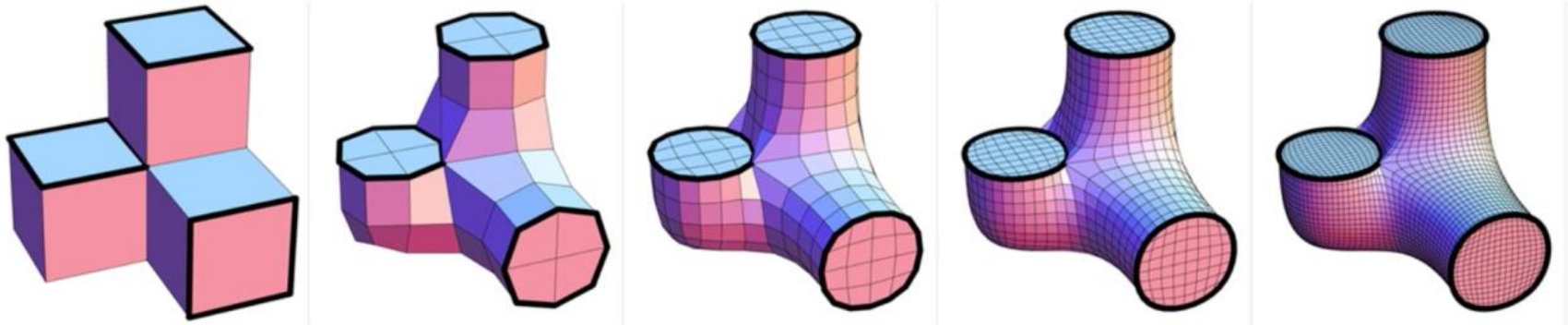


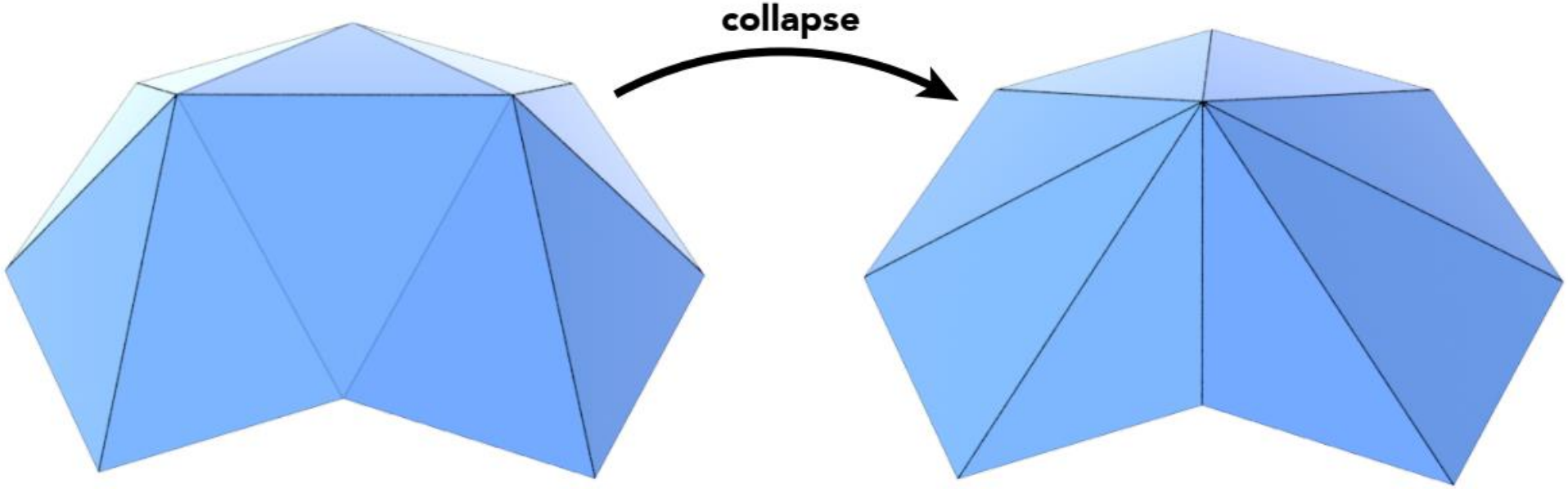
Figure from: Hakenberg et al. Volume Enclosed by Subdivision Surfaces with Sharp Creases

Q&A

网格简化

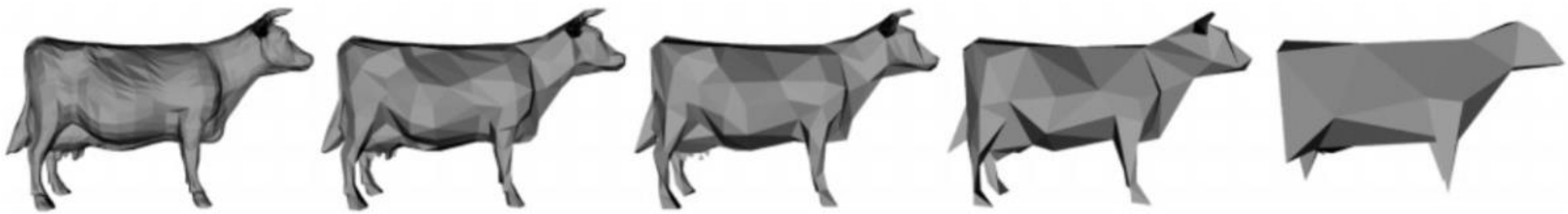
网格简化

- 利用边的塌陷（edge collapsing）实现



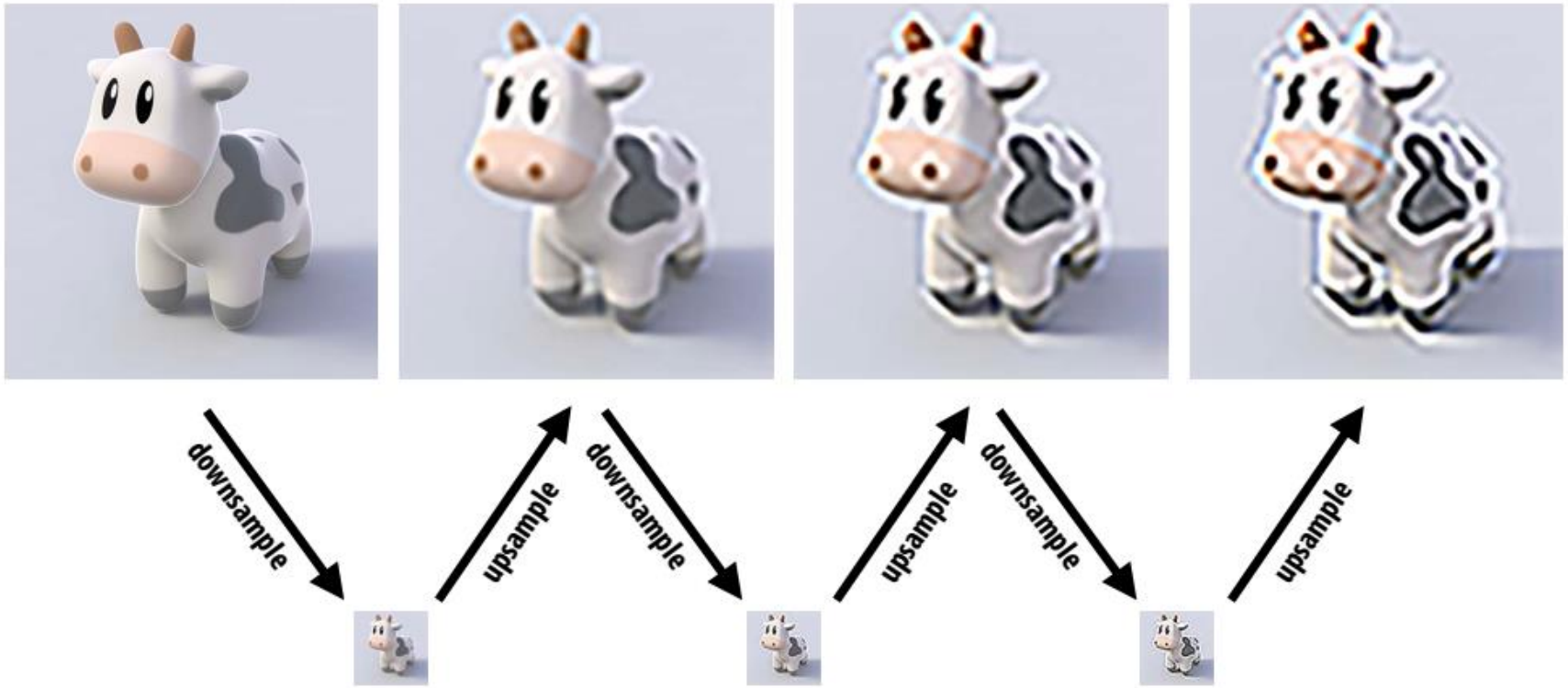
网格简化

- 迭代的进行边塌陷
- 贪心算法
  - 给每条边赋一个成本（cost）
  - 每次对成本最小的边进行边塌陷
  - 重复，直到简化到指定的元素（点/边/面）数目
- 常用的有效的成本函数：二次误差度量（quadric error metric）



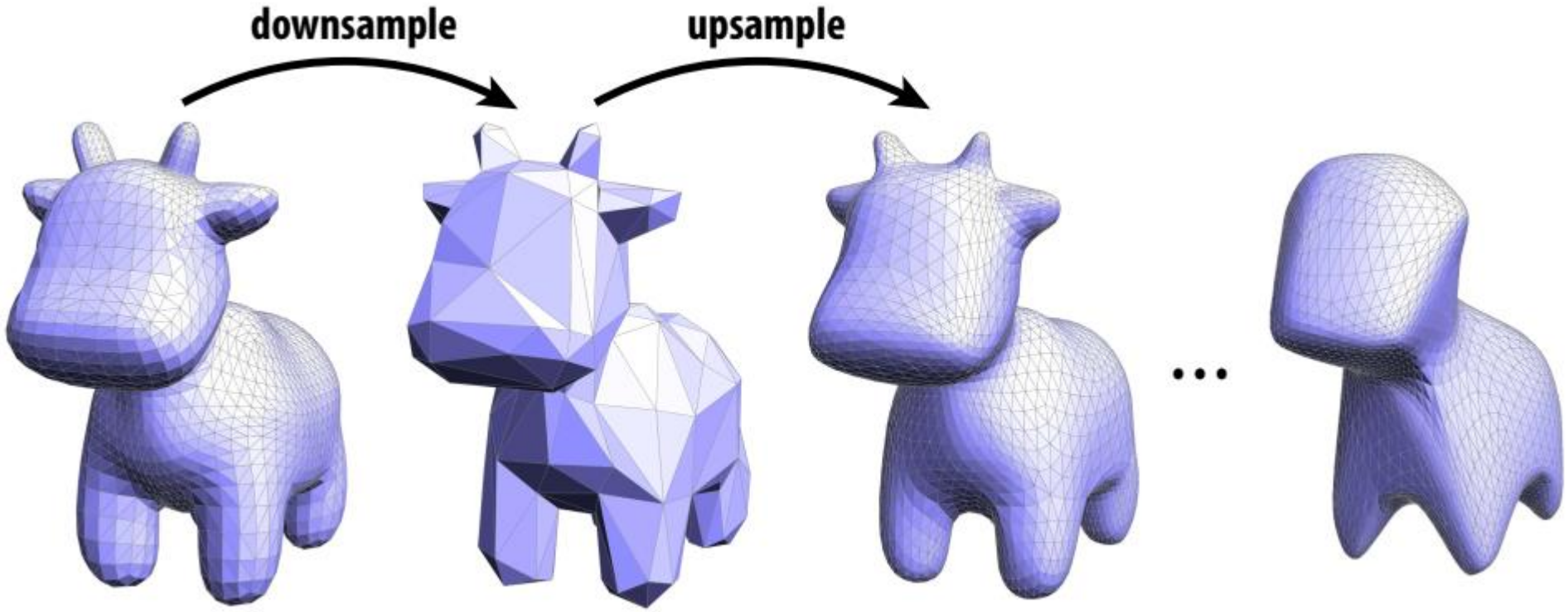
重复重采样

Q:不断地重复重采样一个图片会出现什么问题?



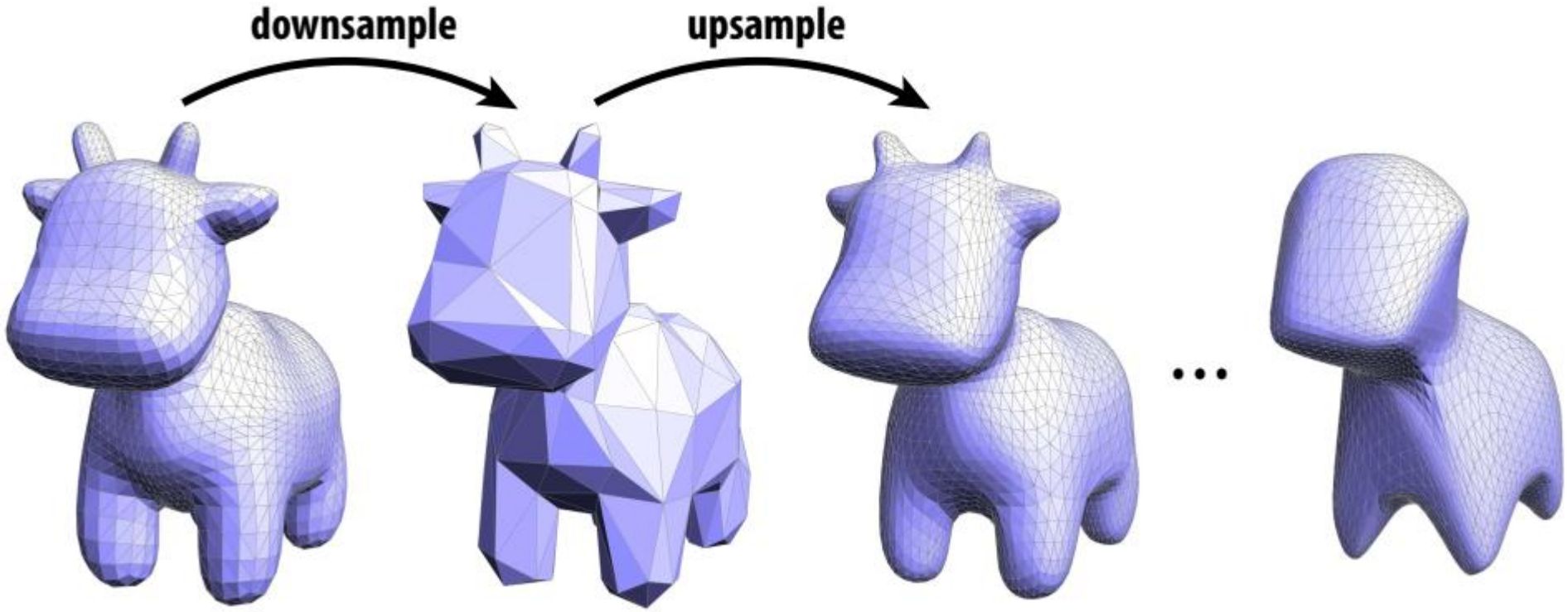
重复重采样

Q:不断地重复重采样一个网格会出现什么问题?



重复重采样

Q:如何解决？



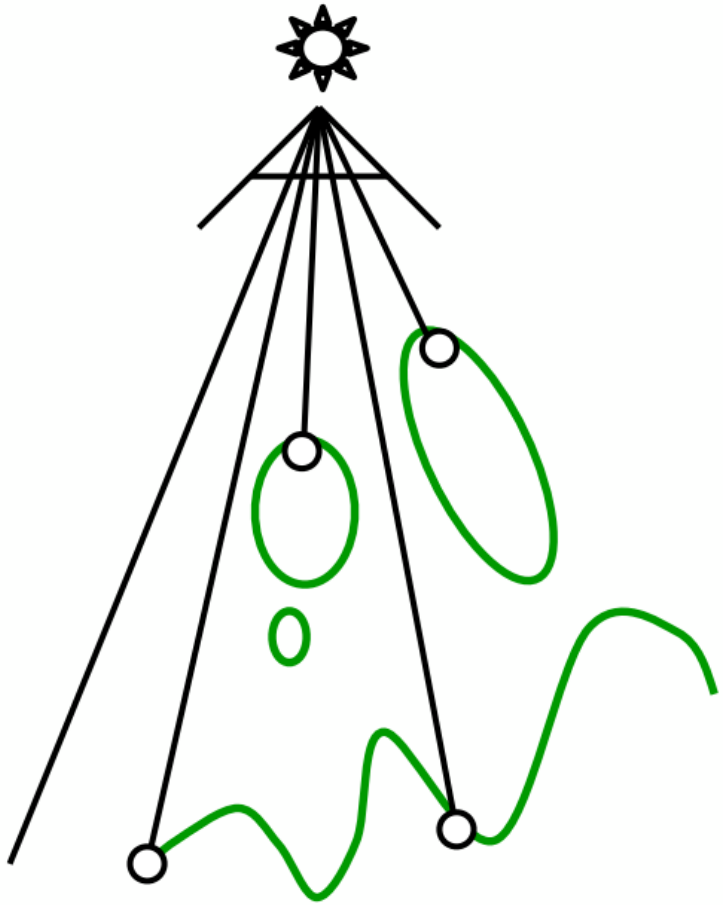
Q&A

如何使用光栅化绘制阴影？

- 阴影贴图（shadow mapping）
  - 图像空间算法
  - 关键思想：不在阴影中的点必须同时被光源和相机看到

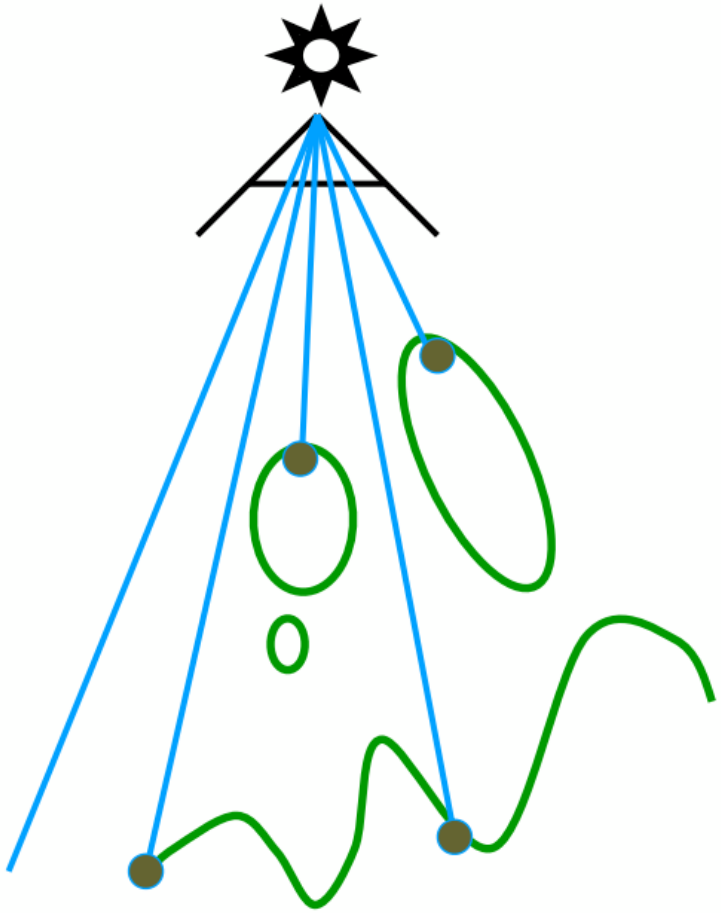
阴影贴图

- 从光源渲染，计算深度图



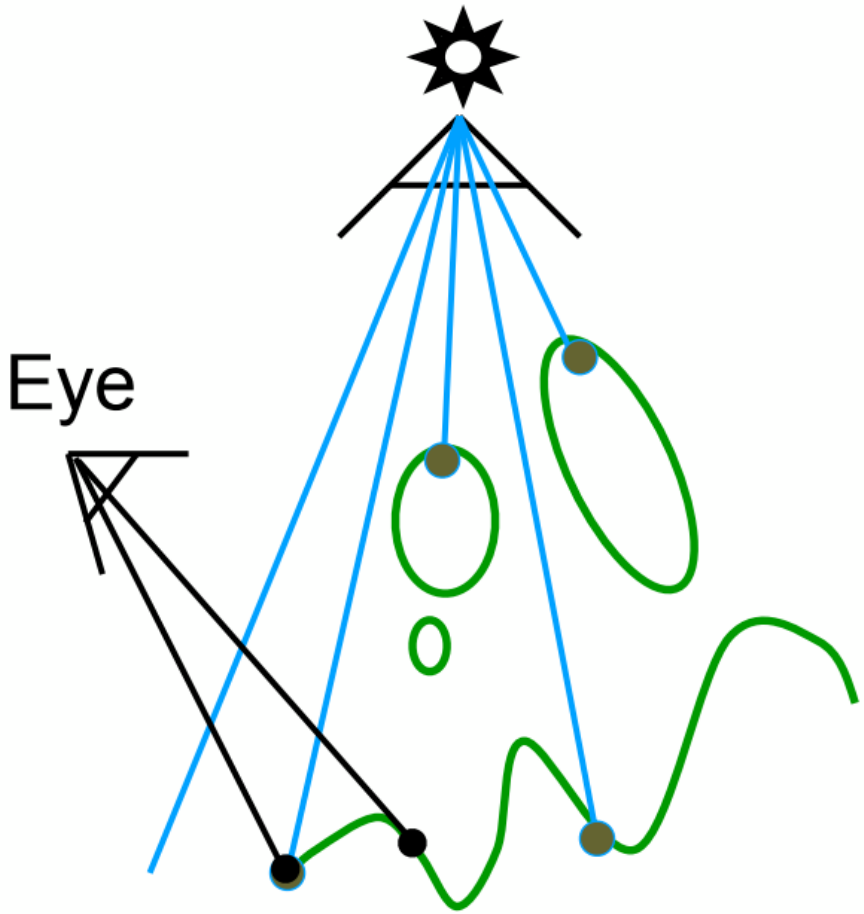
阴影贴图

- 从光源渲染，计算深度图



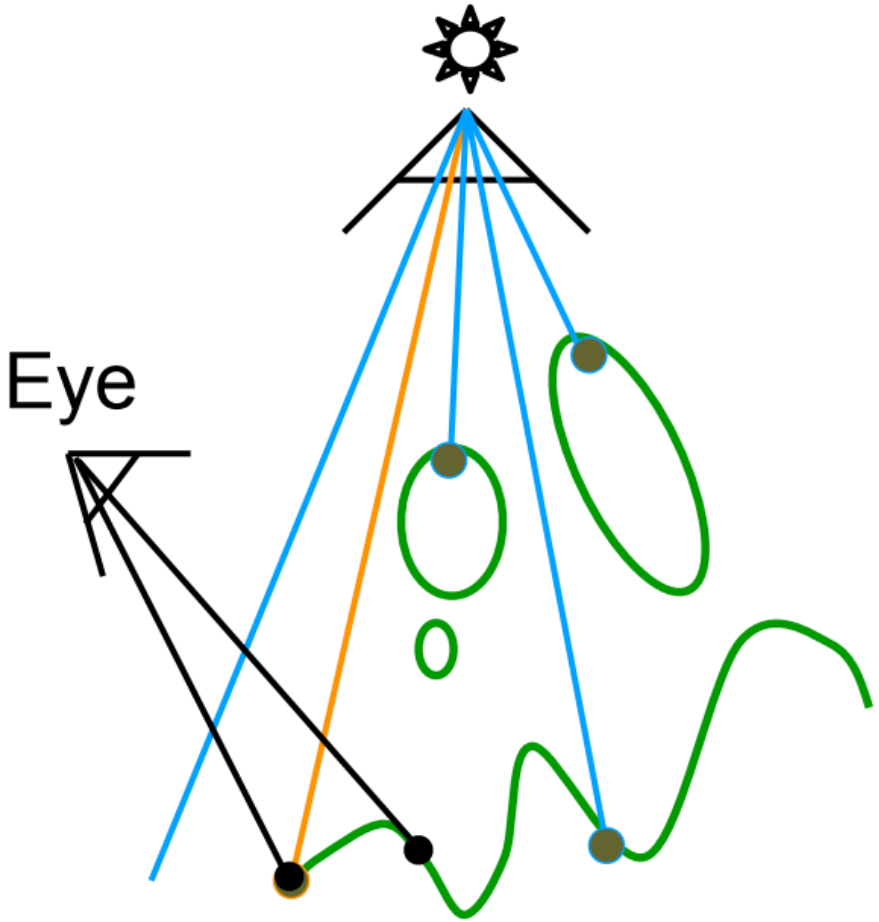
阴影贴图

- 从人眼渲染，得到图像（带深度）



阴影贴图

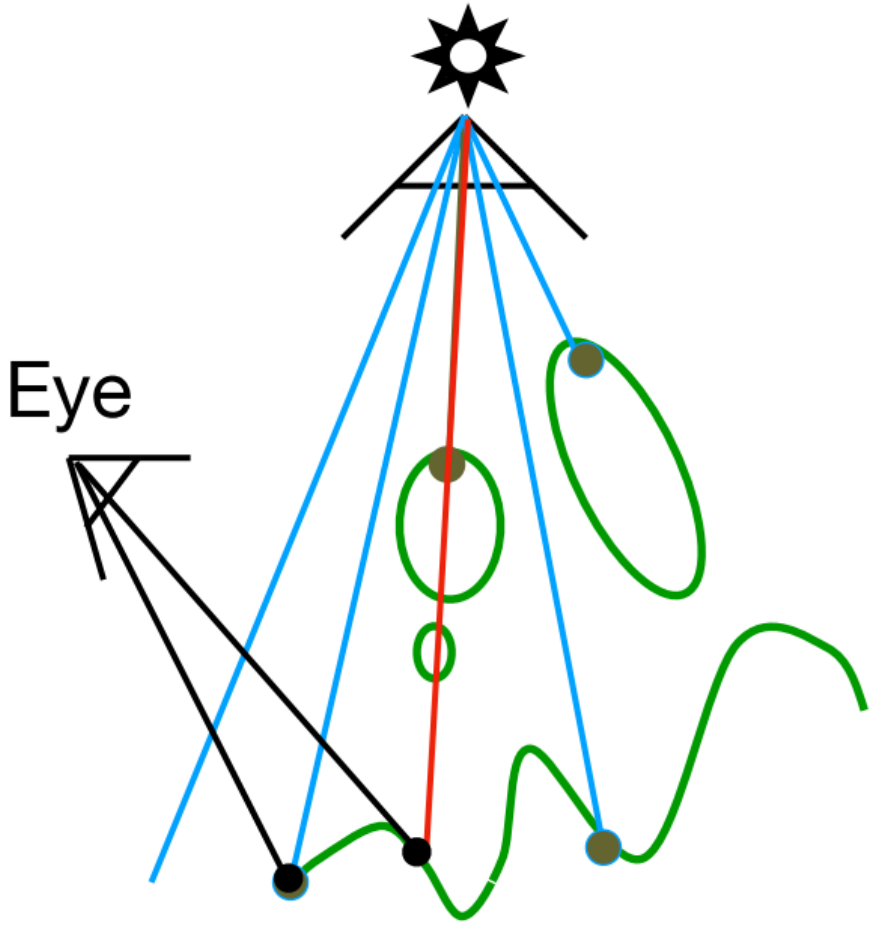
- 将人眼图像中的可见点投射回光源



(Reprojected) depths match for light and eye. VISIBLE

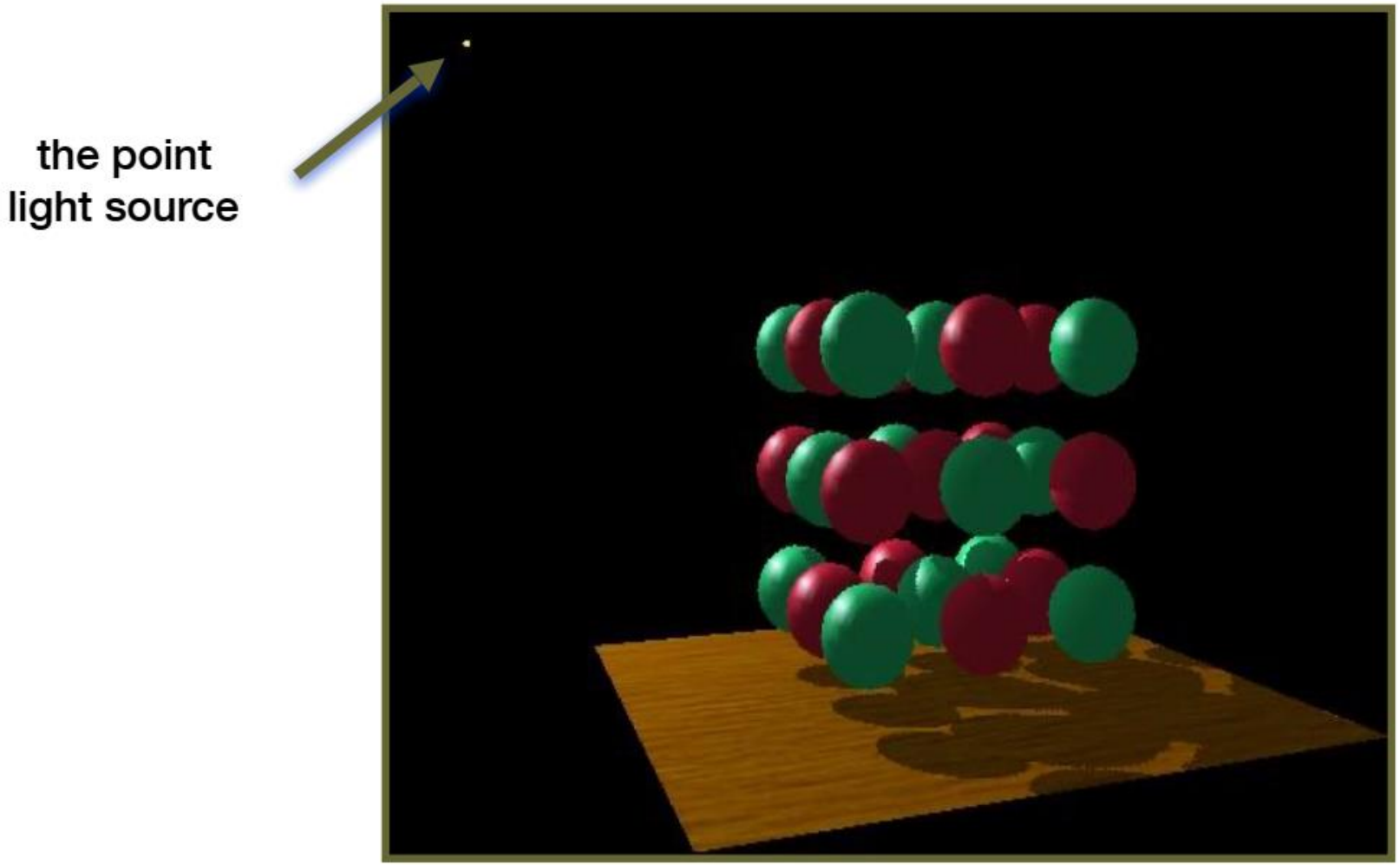
阴影贴图

- 将人眼图像中的可见点投射回光源

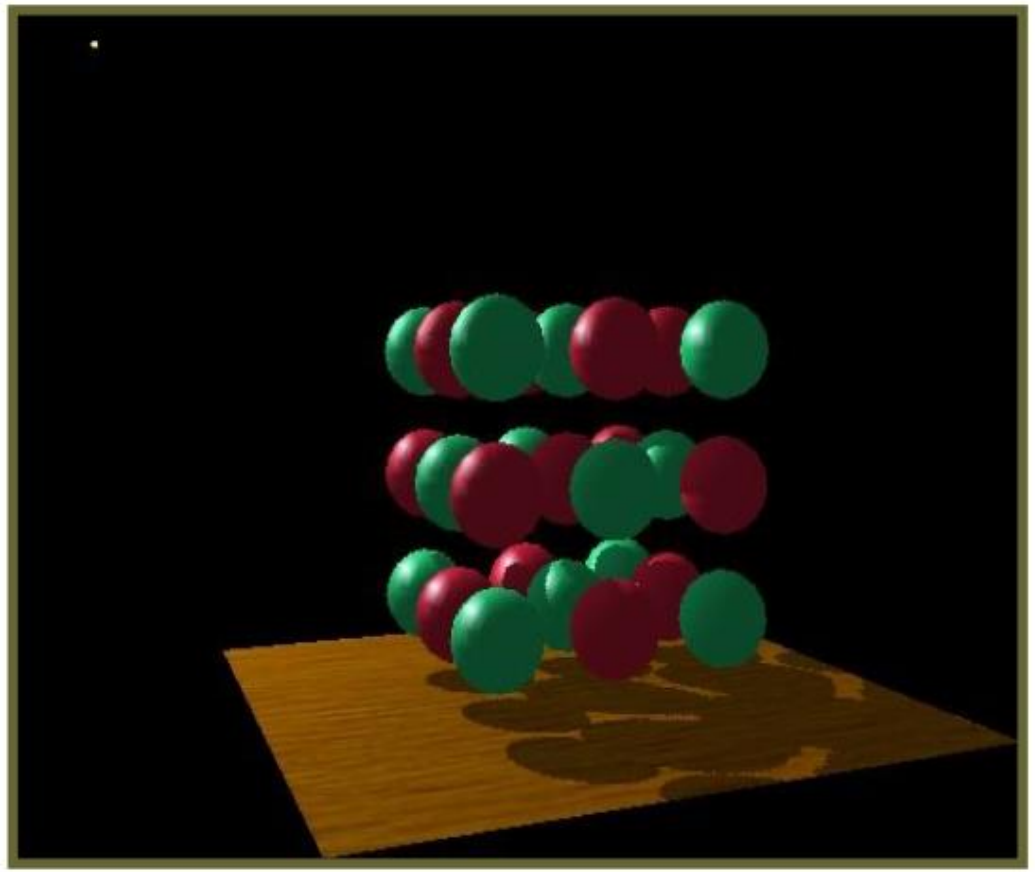


(Reprojected) depths from light and eye are not the same. BLOCKED!!

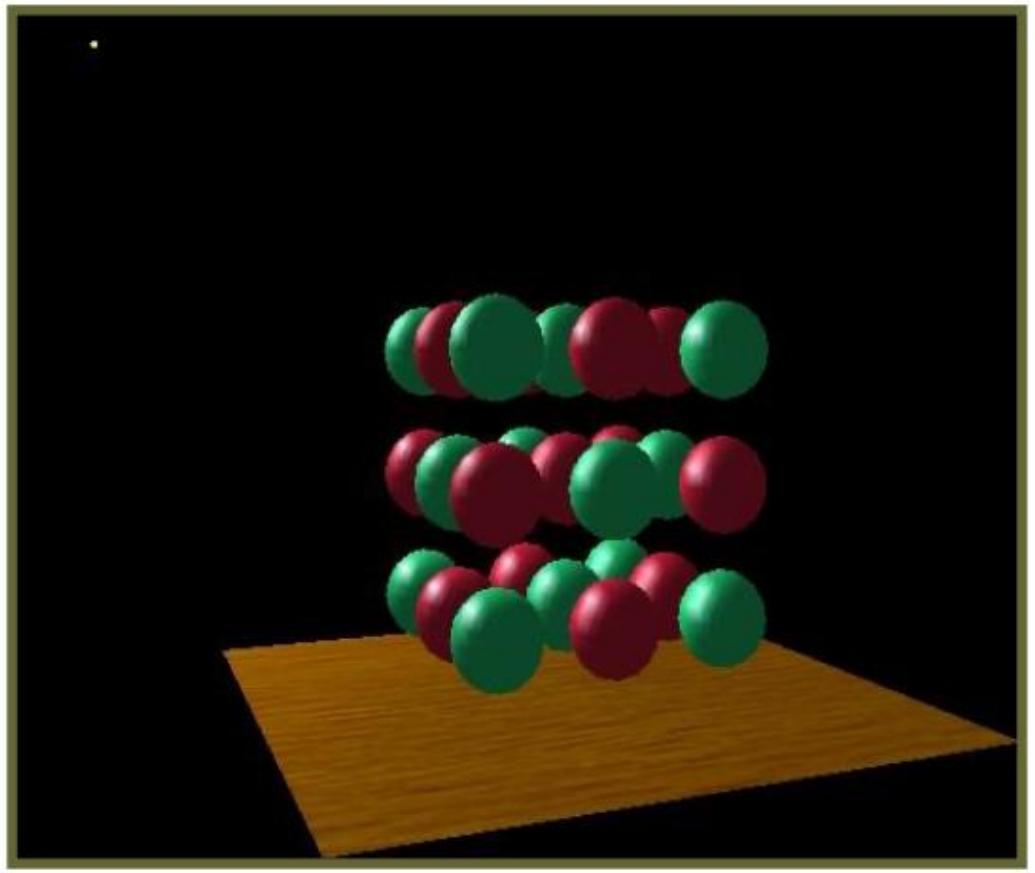
阴影贴图可视化



阴影贴图可视化

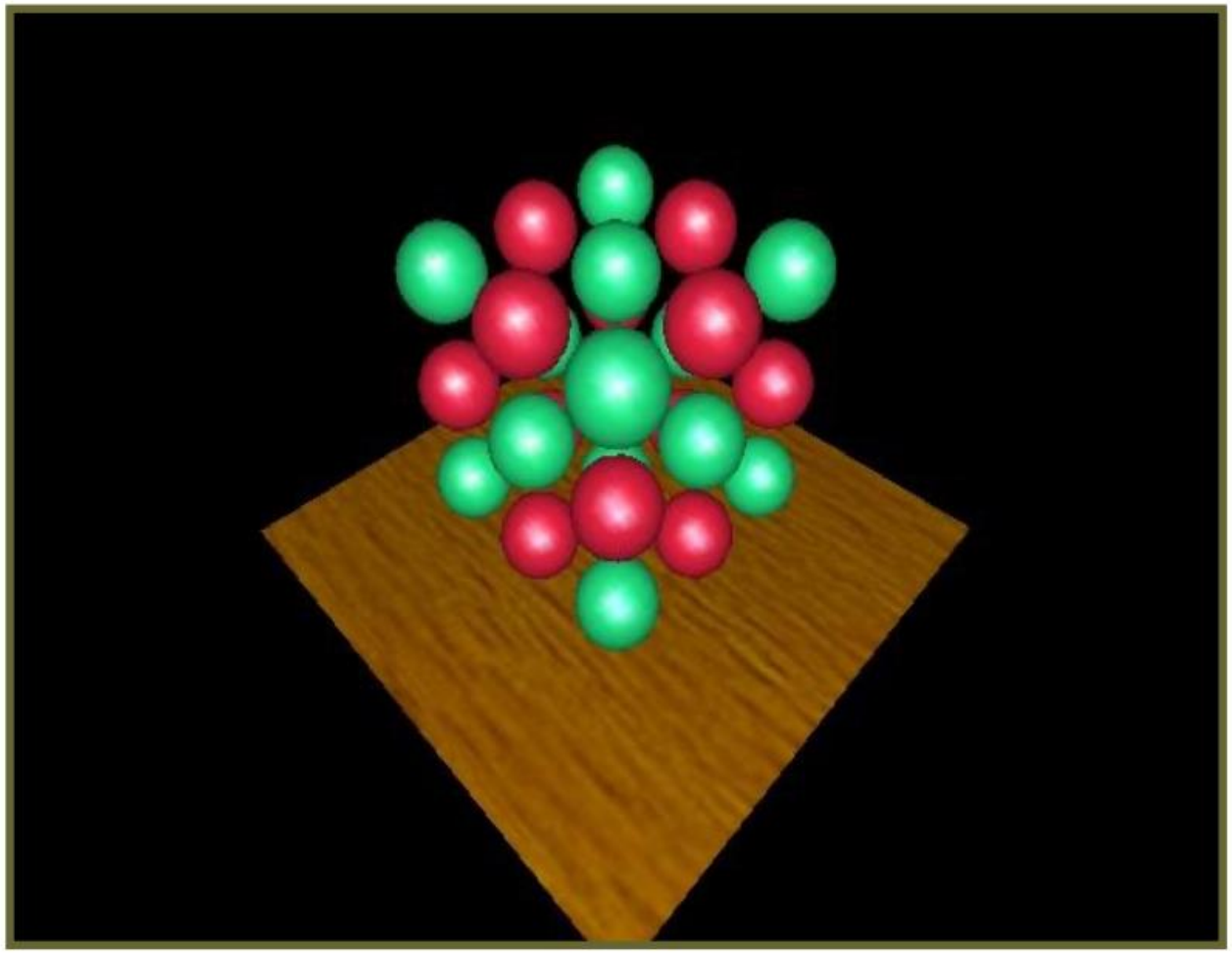


with shadows

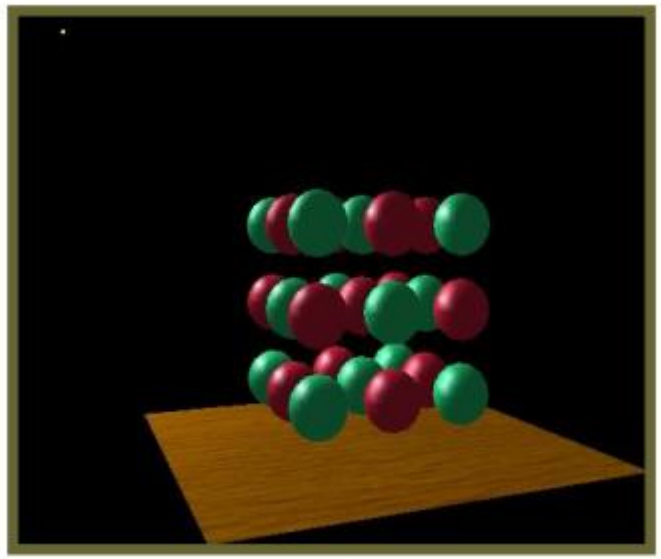


without shadows

阴影贴图可视化

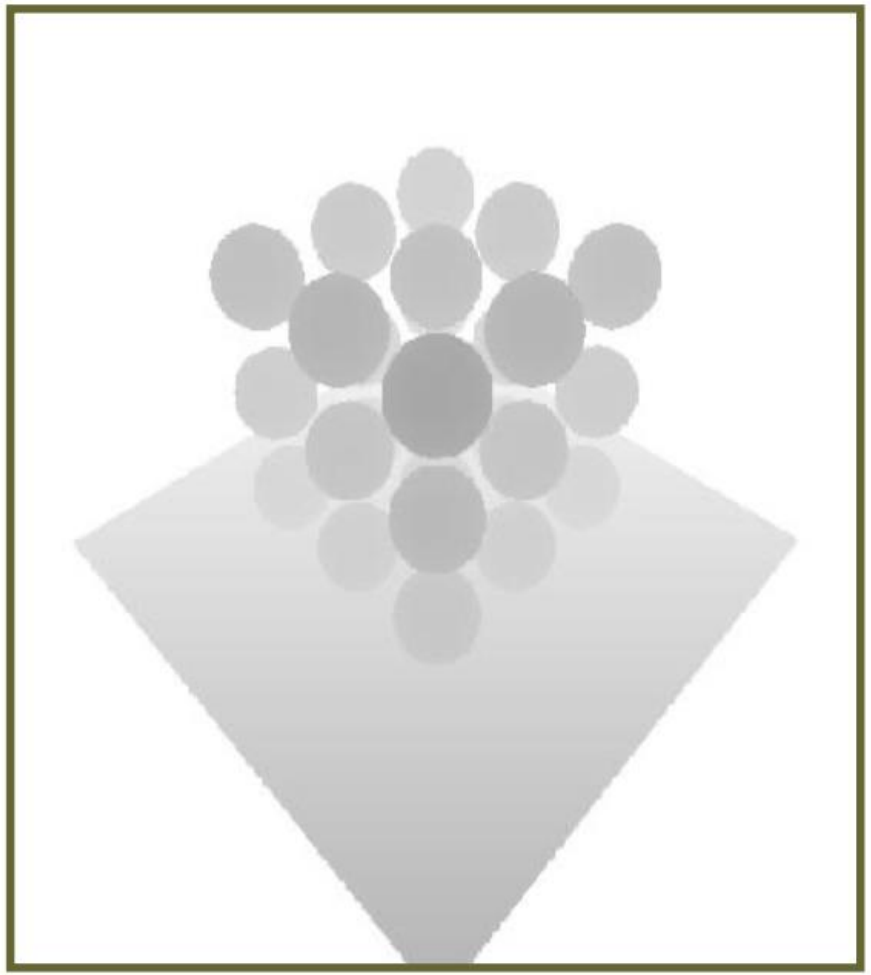


从光源看到的场景

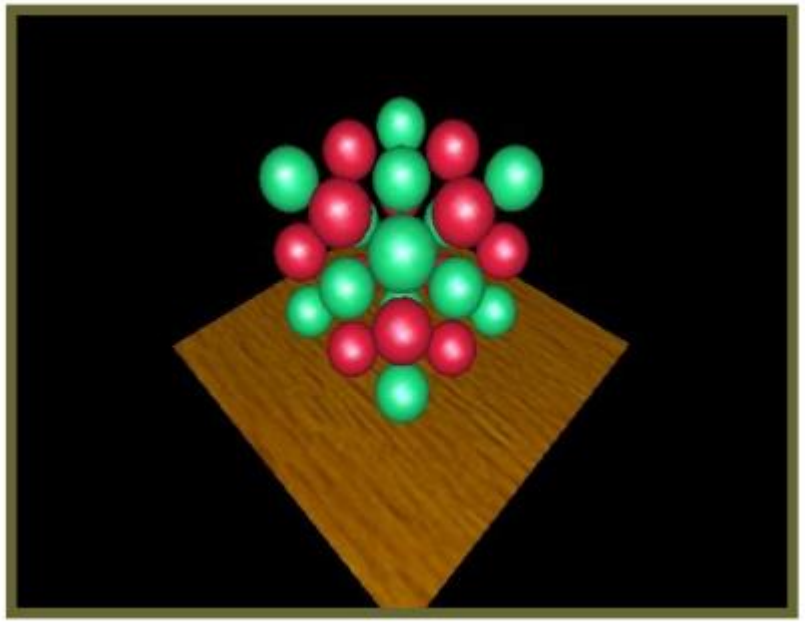


从人眼看到的场景

阴影贴图可视化



从光源看到的深度图

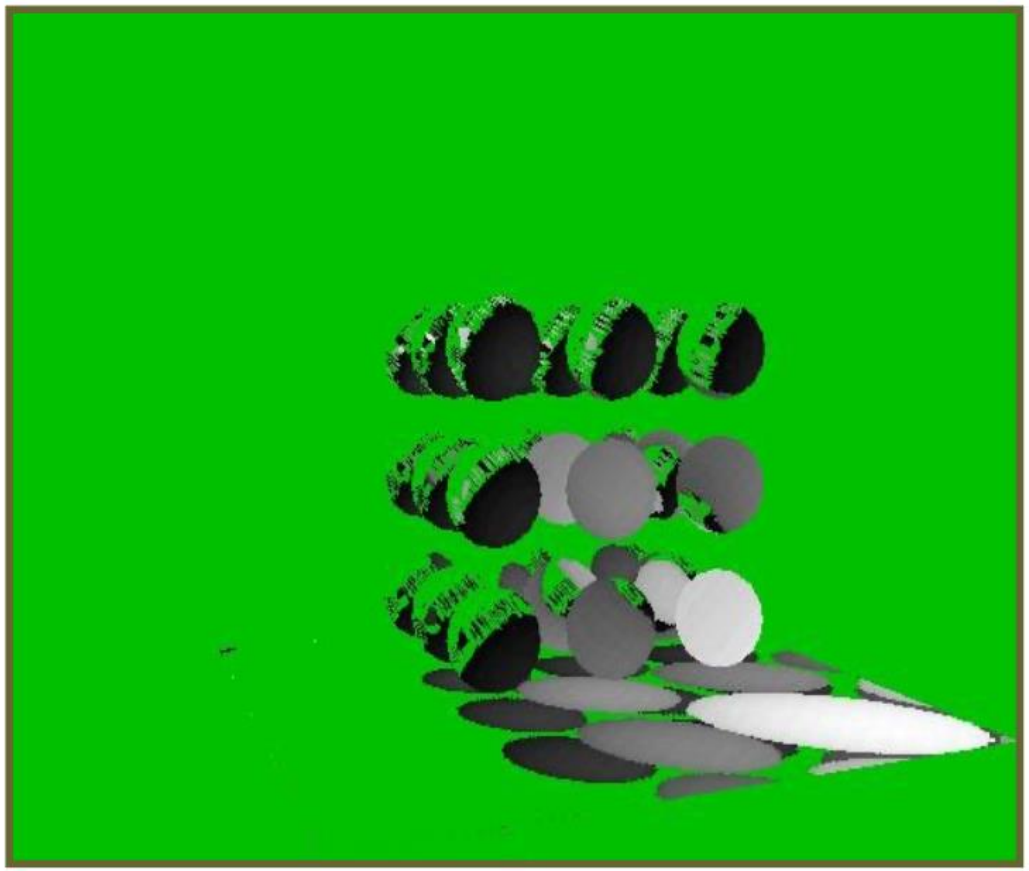


从光源看到的场景

阴影贴图可视化

- 与阴影贴图比较距离

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



Non-green is where shadows should be

阴影贴图存在的问题

- 生成的是硬阴影（仅限点光源）
- 质量取决于阴影贴图的分辨率（基于图像技术的一般问题）
- 涉及浮点深度的相等比较