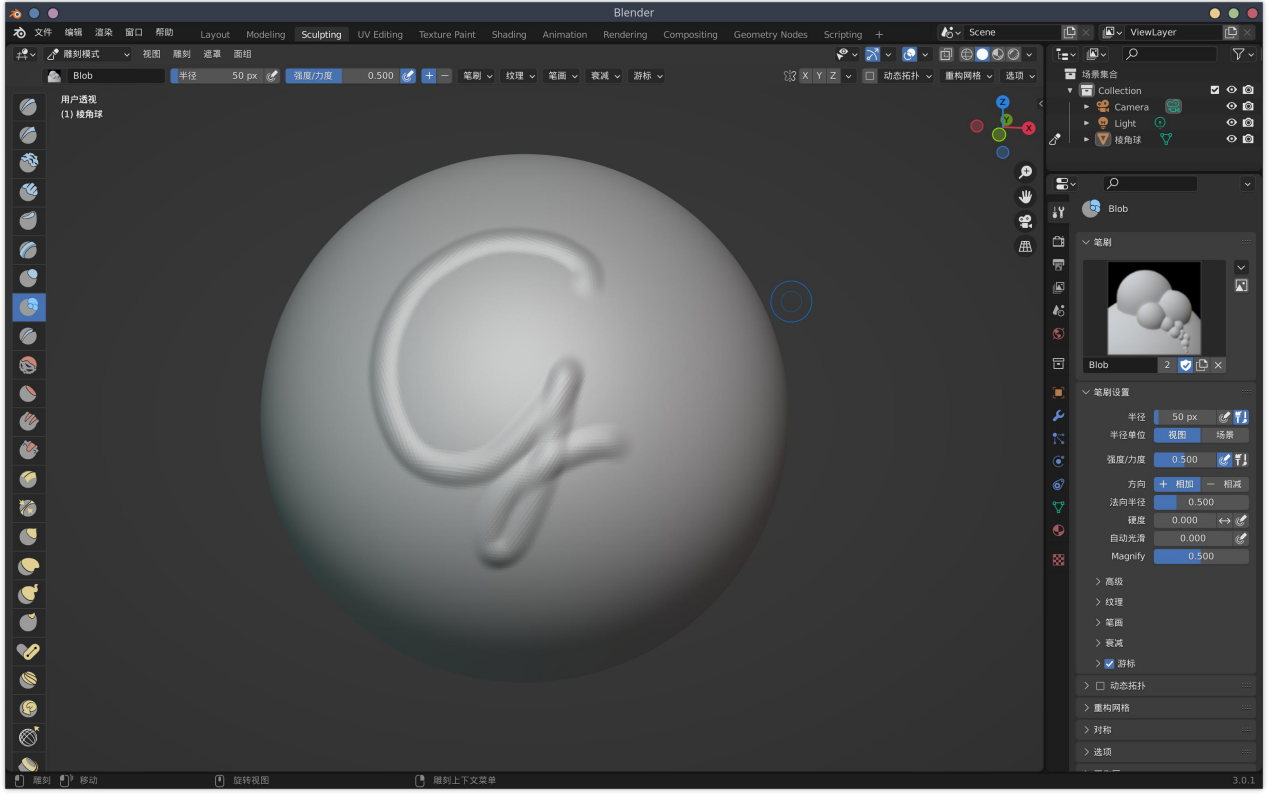
计算机图形学实验课

几何与建模部分

计算机图形学课题组 李昊东

真实的建模工具：实体式建模



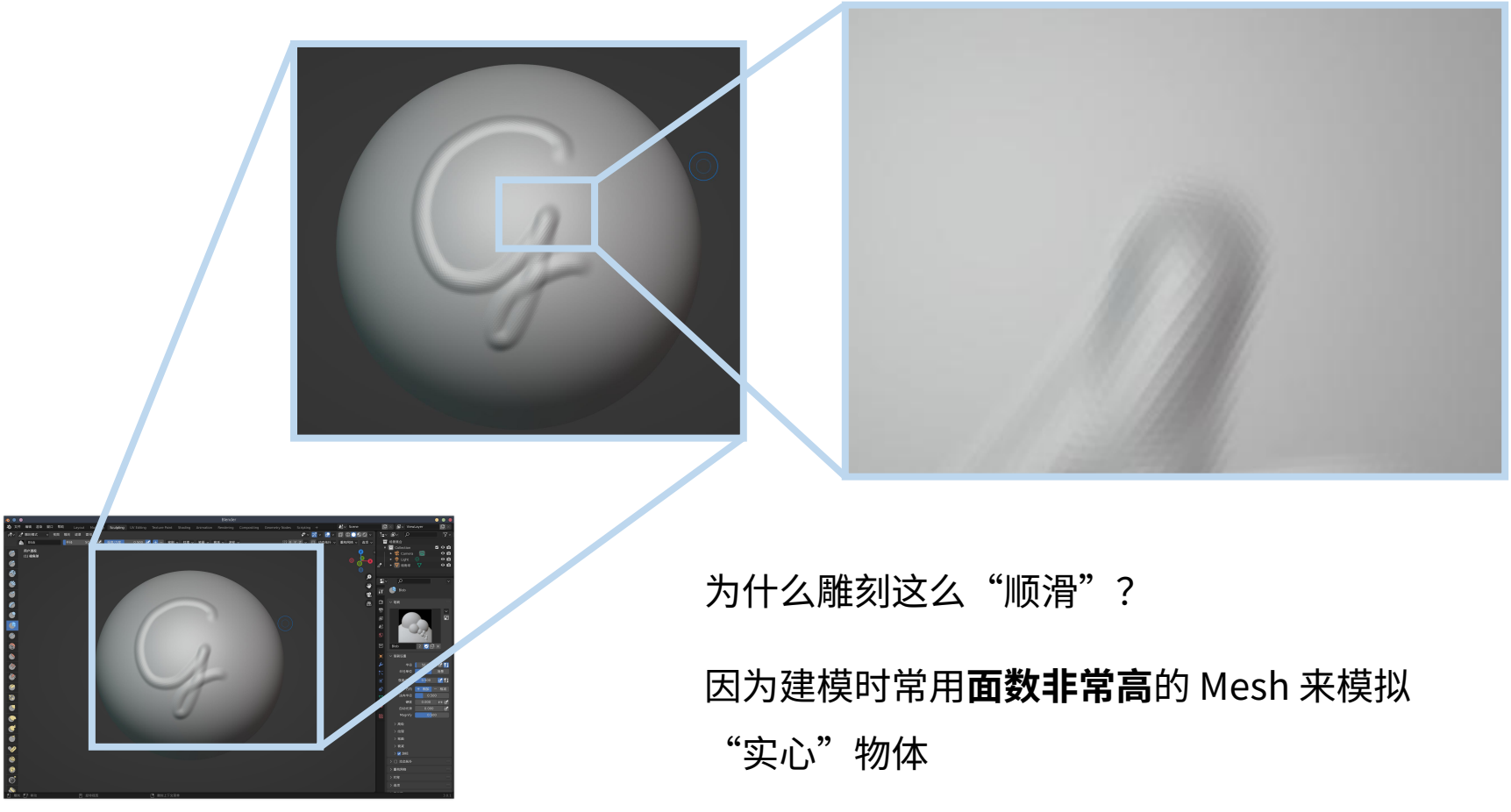
我们以 Blender 为例：

- 从“实心”的形状开始
- 不断雕刻或者塑造
- 最后得到想要的形状

我们可以看一个建模演示：

https://www.bilibili.com/video/BV1AZ4y1A7ck

美术人员的“建模工作流”



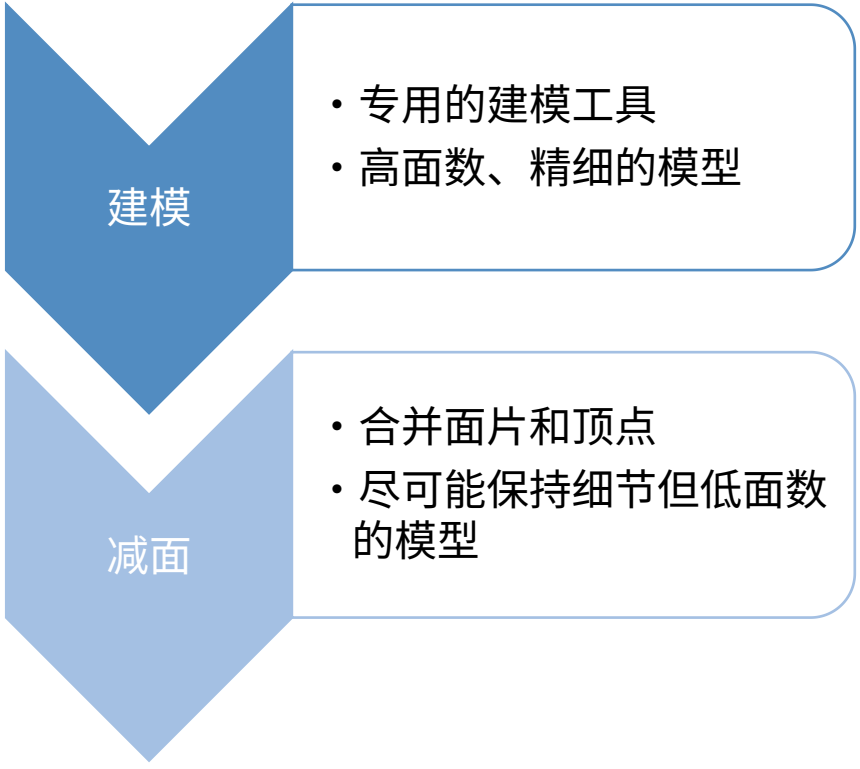
为什么雕刻这么“顺滑”？

因为建模时常用**面数非常高**的 Mesh 来模拟“实心”物体

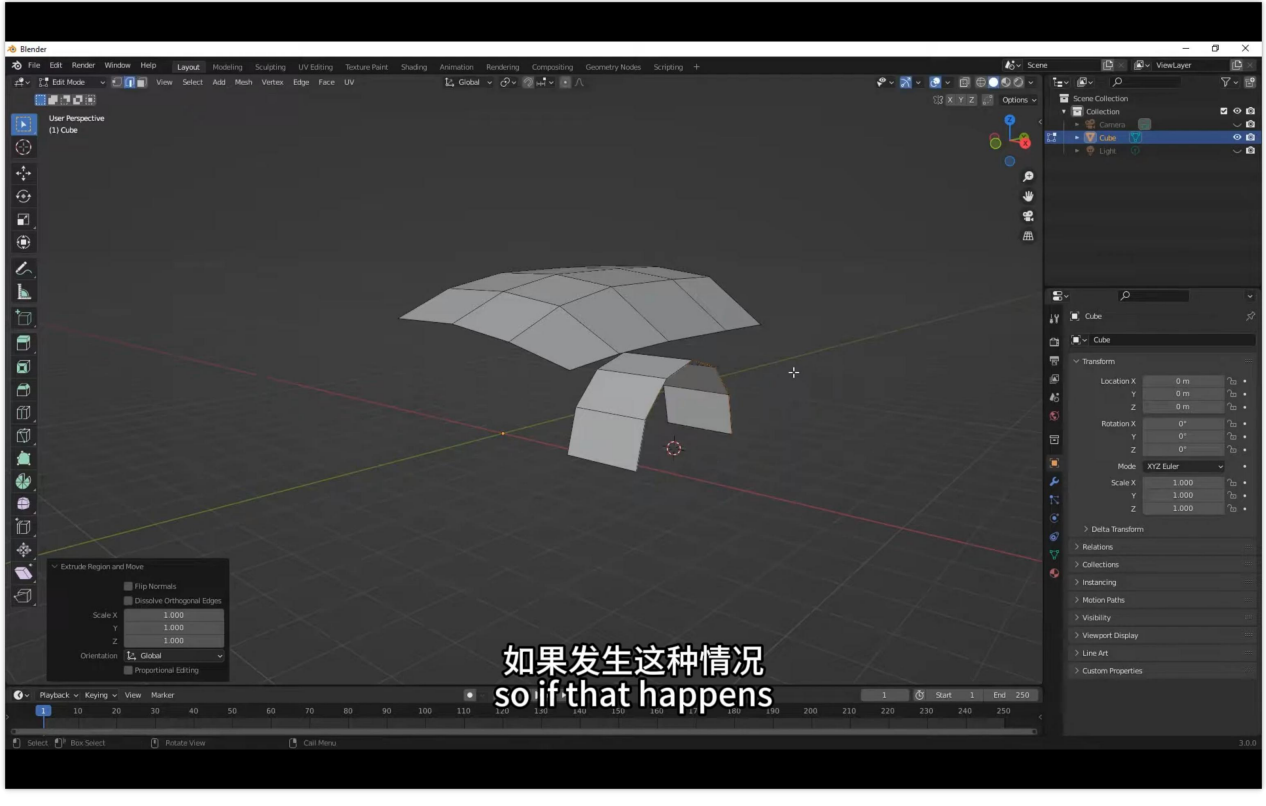
大场景，更大的场景

梦工厂《驯龙高手2》剧照

美术人员的“建模工作流”



真实的建模工具：表面式建模



还是以 Blender 为例：

- 从简单的多边形开始
- 首先设计出大致的形状
- 然后通过细分生成曲面

我们可以看一个建模演示：

https://www.bilibili.com/video/BV17w411s79W

美术人员的“建模工作流”


建模
• 专用的建模工具
• 高面数、精细的模型
减面
• 合并面片和顶点
• 尽可能保持细节但低面数的模型
建模
• 专用的建模工具
• 简单的多边形“格子”
细分
• 指定曲面和边界
• 用多个细分曲面拼合物体

“建模”与“几何处理”

建模

- 得到一个三维模型的过程
- 由人完成其中“设计”的部分
- 并不一定要用特定的某种工具

总体来说是一种任务

几何处理

- 自动或半自动处理几何体的过程
- 人只负责指定输入和参数
- 每种几何处理操作有特定的算法去实现

核心是算法设计和实现

几何处理可以成为**辅助建模**的工具，比如简化或者细分

几何处理算法的实现基础：程序化操纵 Mesh 的方法

渲染

- 需要每个面片的坐标——关心几何
- 不需要了解邻接关系——不关心拓扑

几何处理

- 需要制造形变——关心几何
- 形变的时候往往要形变一块区域——频繁的邻域访问和修改

使用连续数组存储

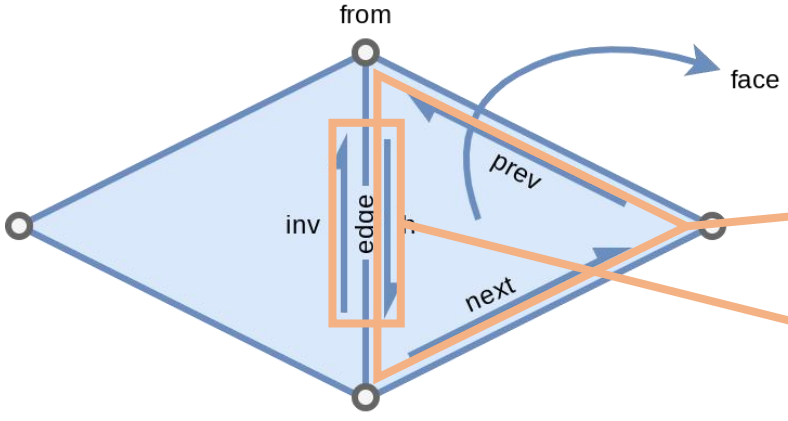
Vertex	
x, y, z	0
x, y, z	1
x, y, z	2
……	…

Face
0, 1, 2
3, 0, 5
6, 2, 8
……

用指针？邻接表？

Vertex	
x, y, z	0
x, y, z	1
x, y, z	2
……	…

几何处理算法的实现基础：程序化操纵 Mesh 的方法



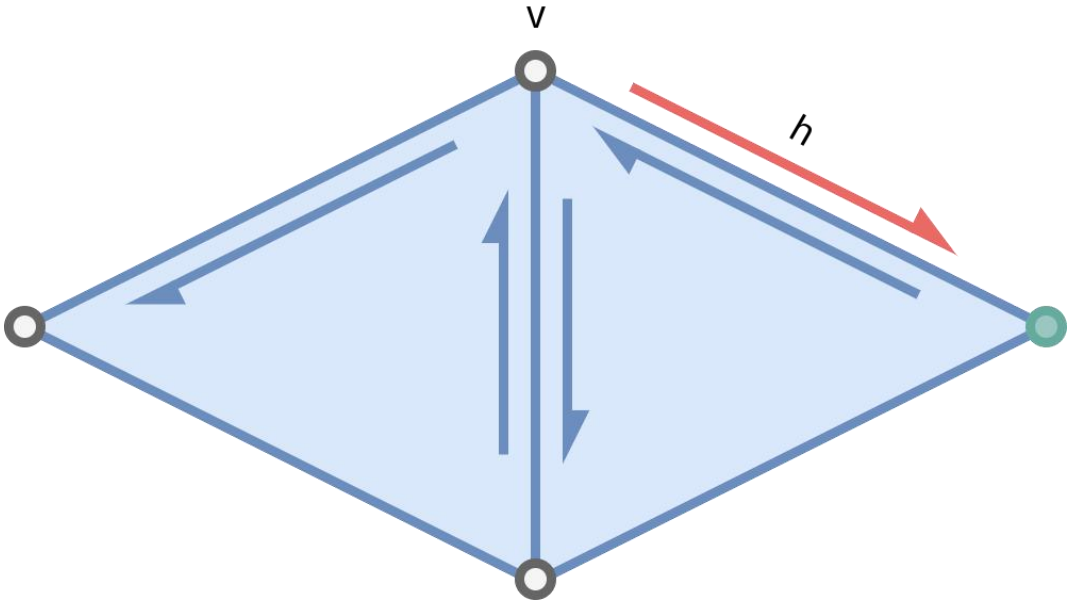
归纳 Mesh 中的两种关键邻接关系：

1. 同一个面上，边首尾相接
2. 每一条边被两个面片共享

Halfedge
+ prev: Halfedge*
+ next: Halfedge*
+ inv: Halfedge*
+ from: Vertex*
+ edge: Edge*
+ face: Face*

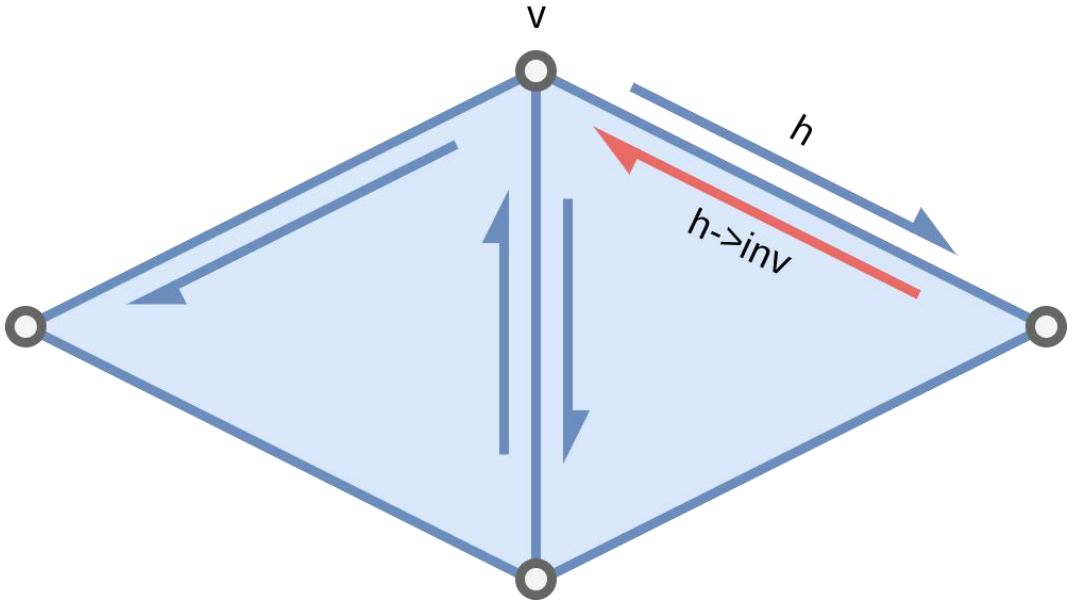
可以将一条边（无向）拆分成两条半边（有向）

示例：一次性访问并操作一个顶点及其邻接顶点



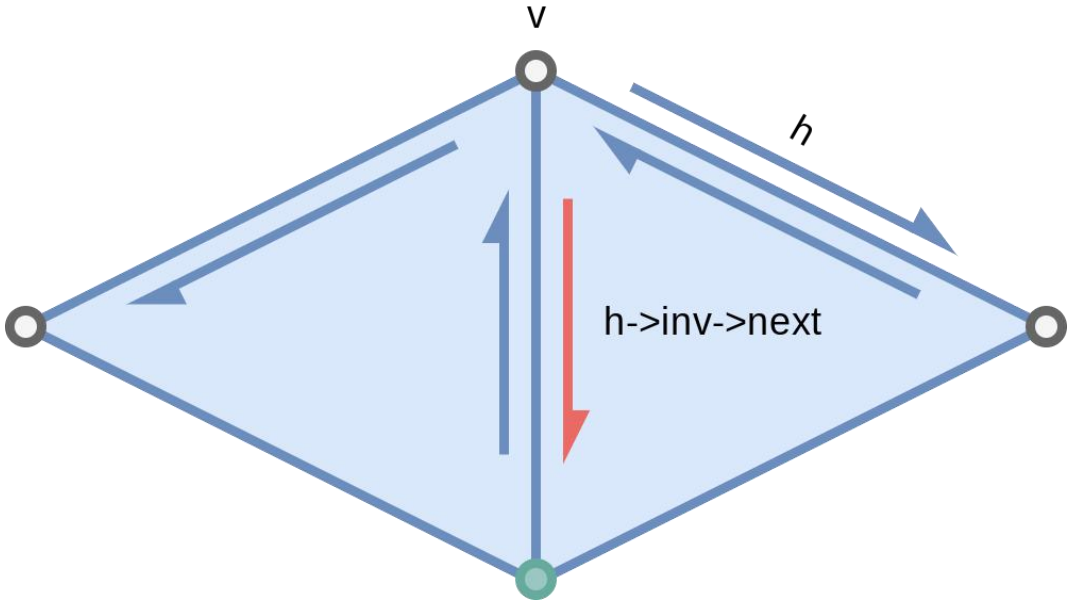
从顶点 v 出发，访问它的某条半边

示例：一次性访问并操作一个顶点及其邻接顶点



访问半边的反向，再次回到顶点 v

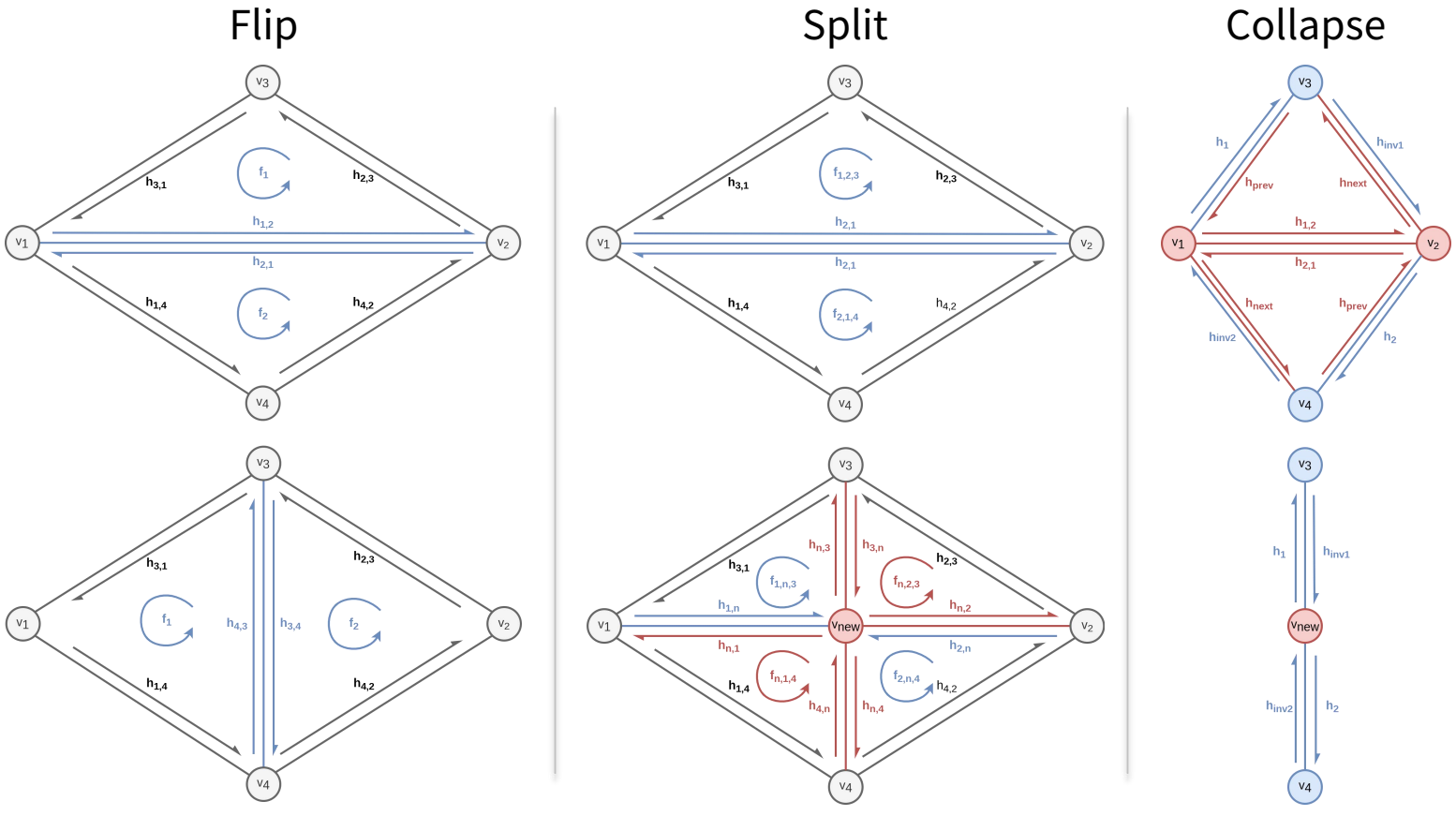
示例：一次性访问并操作一个顶点及其邻接顶点



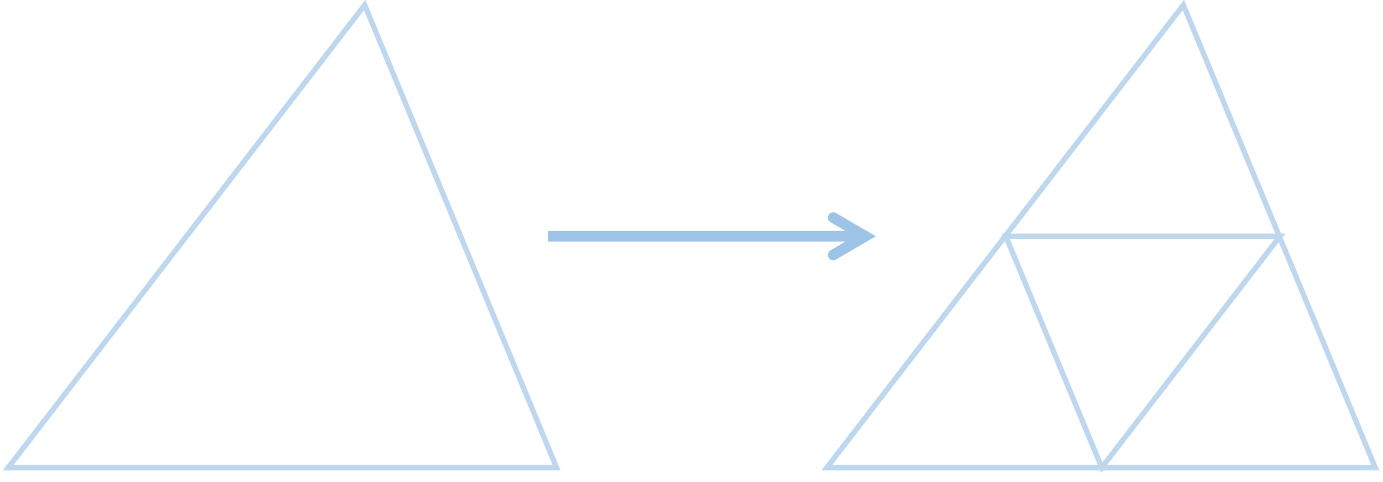
访问反向半边的下一条半边，到达下一个邻接顶点

如此循环往复，直至再次到达 h 为止

程序化操纵 Mesh 的最小单位：局部操作

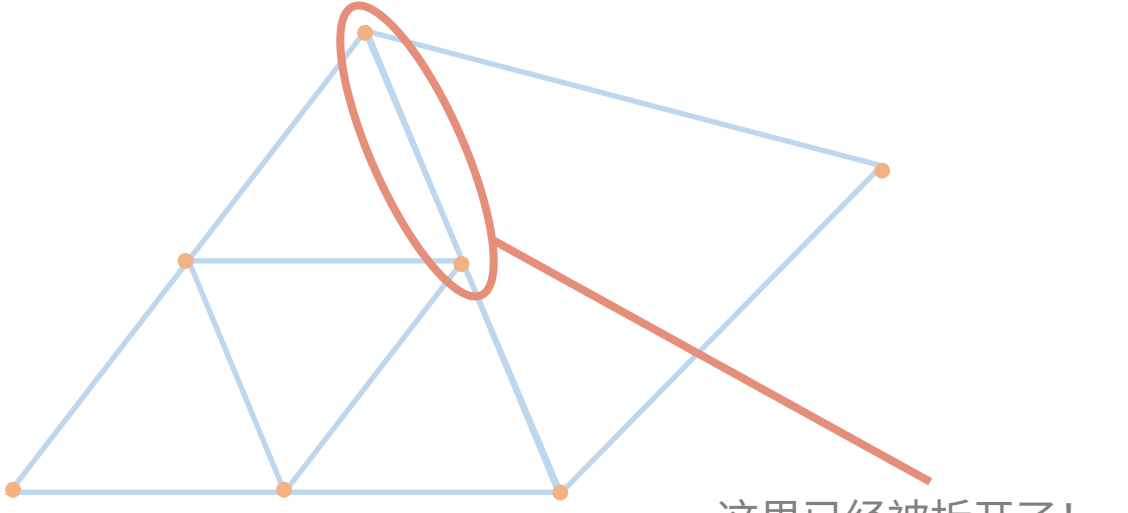


从局部操作到几何处理：Loop 曲面细分



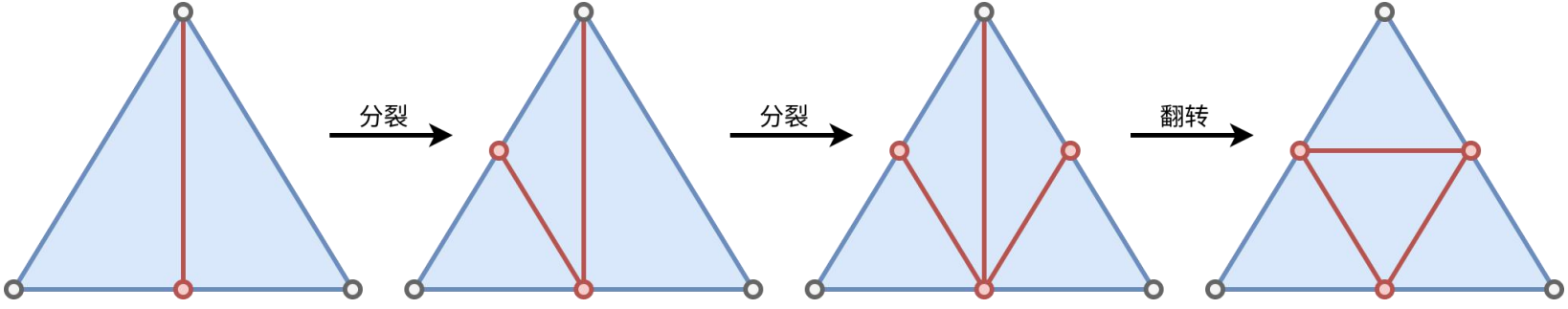
如果不考虑面片的邻接关系，这是个很简单的问题

从局部操作到几何处理：Loop 曲面细分



我们标出顶点再考虑一下情况……

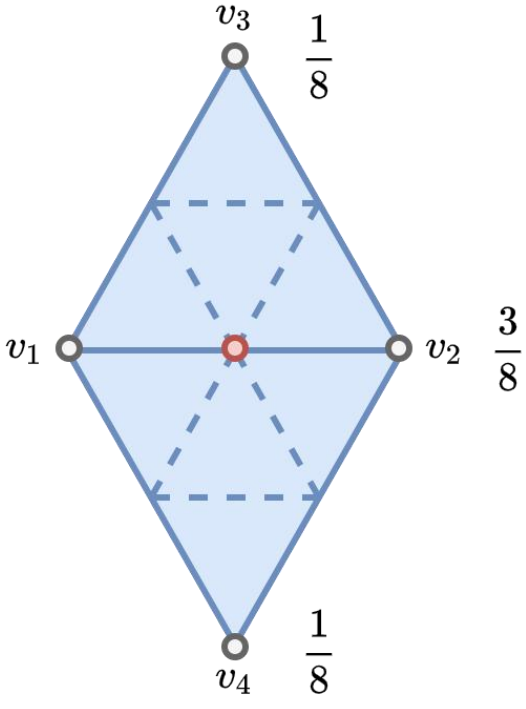
从局部操作到几何处理：Loop 曲面细分

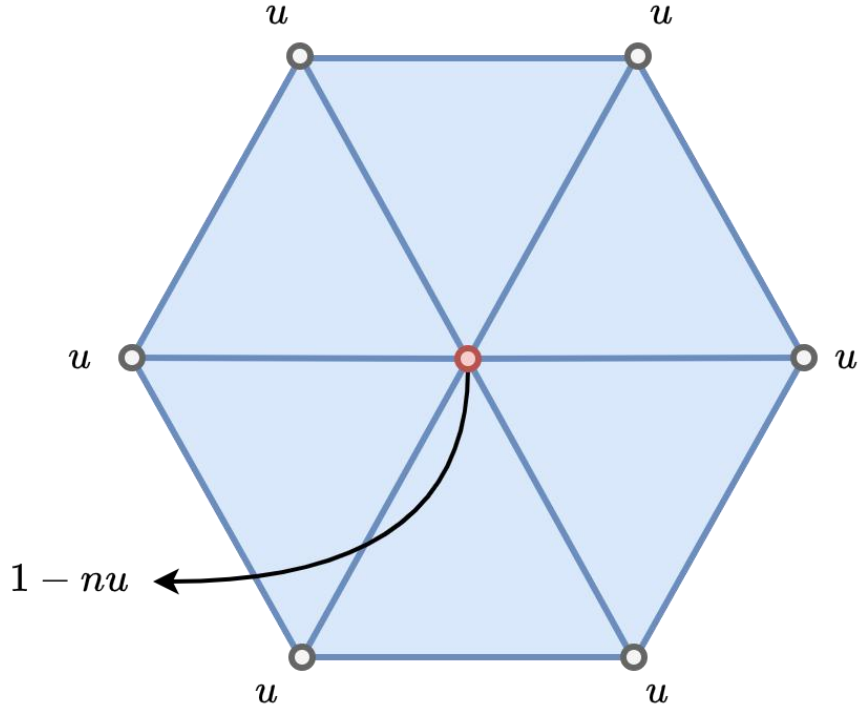


为什么这样做?

- 在任何一次局部操作结束后，都不会产生四边形
- 操作时完全不关心相邻面片

从局部操作到几何处理：Loop 曲面细分





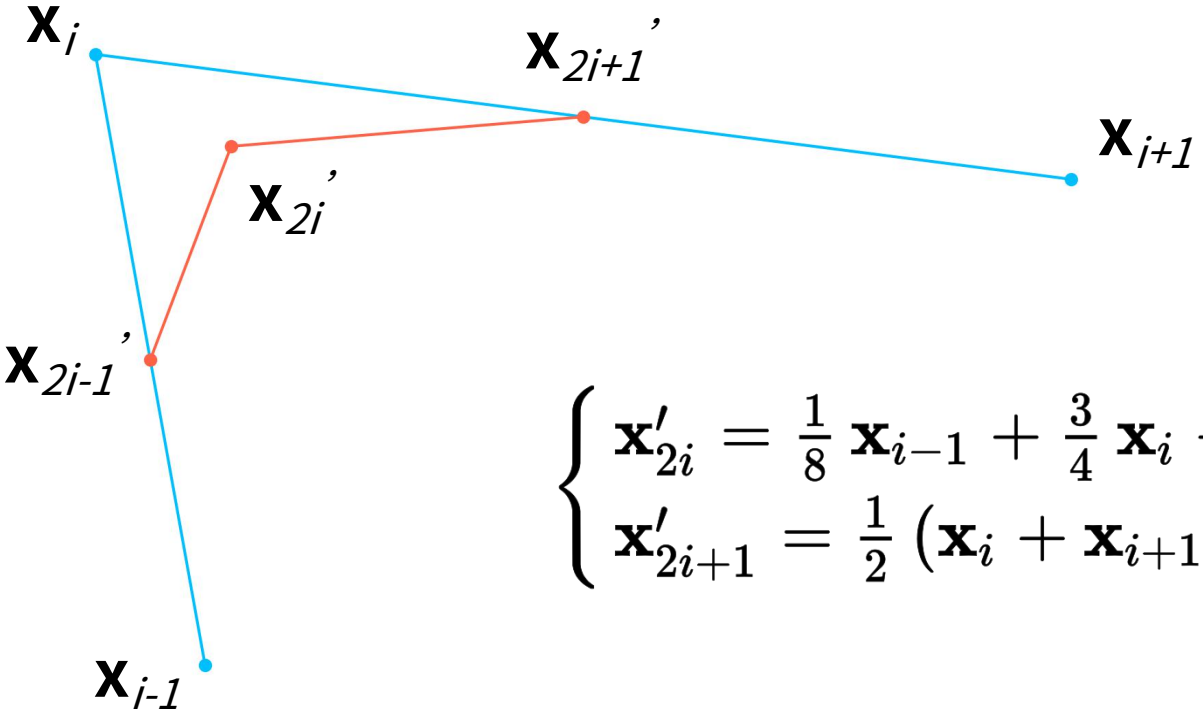
从局部操作到几何处理：Loop 曲面细分

为什么这样的权重可以保证收敛?

如果我换其他的行不行?

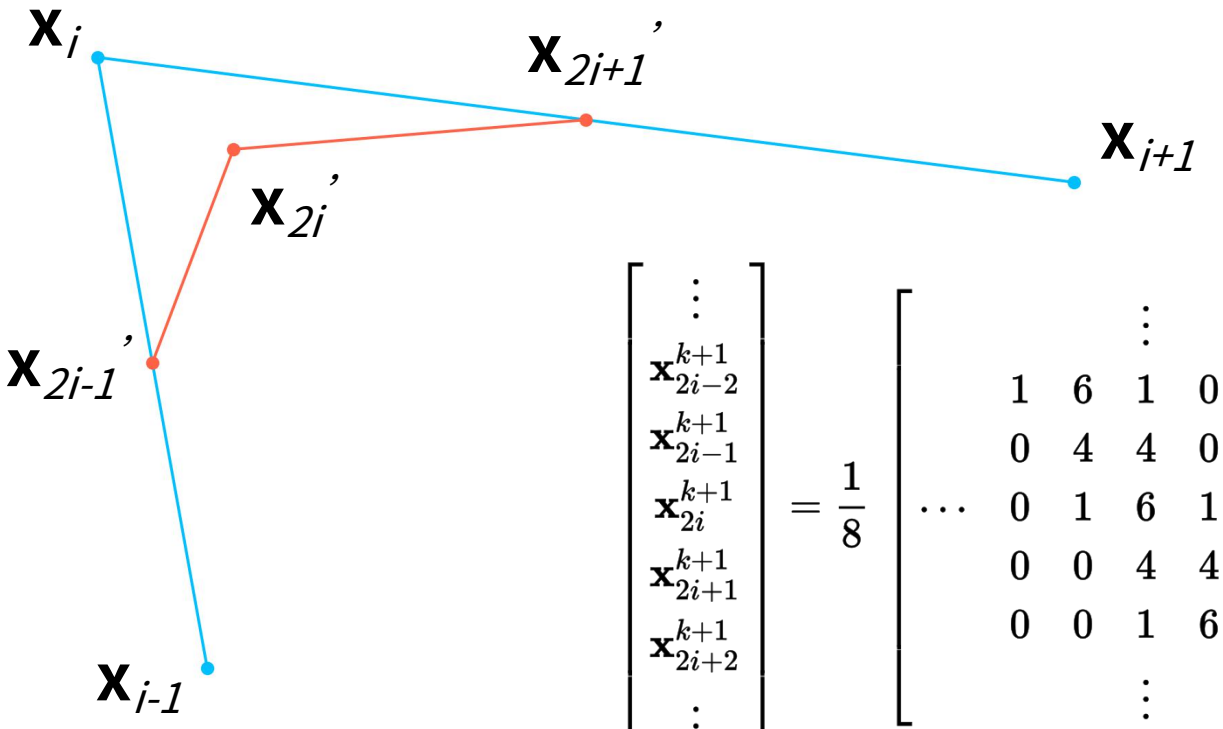
首先我们来做个简化版的实验。

以细分曲线为例



$$\begin{cases} \mathbf{x}'_{2i} = \frac{1}{8}\mathbf{x}_{i-1} + \frac{3}{4}\mathbf{x}_i + \frac{1}{8}\mathbf{x}_{i+1} \\ \mathbf{x}'_{2i+1} = \frac{1}{2}\left(\mathbf{x}_i + \mathbf{x}_{i+1}\right) \end{cases}$$

以细分曲线为例



$$\begin{bmatrix} \vdots \\ \mathbf{x}_{2i-2}^{k+1} \\ \mathbf{x}_{2i-1}^{k+1} \\ \mathbf{x}_{2i}^{k+1} \\ \mathbf{x}_{2i+1}^{k+1} \\ \mathbf{x}_{2i+2}^{k+1} \\ \vdots \end{bmatrix} = \frac{1}{8} \begin{bmatrix} & & \vdots & & & \\ & 1 & 6 & 1 & 0 & \\ & 0 & 4 & 4 & 0 & \\ \cdots & 0 & 1 & 6 & 1 & 0 & \cdots \\ & 0 & 0 & 4 & 4 & 0 & \\ & 0 & 0 & 1 & 6 & 1 & \\ & & & \vdots & & & \end{bmatrix} \begin{bmatrix} \vdots \\ \mathbf{x}_{i-2}^{k} \\ \mathbf{x}_{i-1}^{k} \\ \mathbf{x}_{i}^{k} \\ \mathbf{x}_{i+1}^{k} \\ \mathbf{x}_{i+2}^{k} \\ \vdots \end{bmatrix}$$

$2n \times 1$ $\qquad$ $2n \times n$ $\qquad$ $n \times 1$

以细分曲线为例

$$\begin{bmatrix} \vdots \\ \mathbf{x}_{2i-2}^{k+1} \\ \mathbf{x}_{2i-1}^{k+1} \\ \mathbf{x}_{2i}^{k+1} \\ \mathbf{x}_{2i+1}^{k+1} \\ \mathbf{x}_{2i+2}^{k+1} \\ \vdots \end{bmatrix} = \frac{1}{8} \begin{bmatrix} & & \vdots & & & \\ & 1 & 6 & 1 & 0 & 0 \\ & 0 & 4 & 4 & 0 & 0 \\ \cdots & 0 & 1 & 6 & 1 & 0 & \cdots \\ & 0 & 0 & 4 & 4 & 0 \\ & 0 & 0 & 1 & 6 & 1 \\ & & & \vdots & & \end{bmatrix} \begin{bmatrix} \vdots \\ \mathbf{x}_{i-2}^{k} \\ \mathbf{x}_{i-1}^{k} \\ \mathbf{x}_{i}^{k} \\ \mathbf{x}_{i+1}^{k} \\ \mathbf{x}_{i+2}^{k} \\ \vdots \end{bmatrix}$$

将它扩大成 2n×2n 维方阵

右边缺少的地方都填上 0

以细分曲线为例

$$\mathbf{x}^{\infty} = \lim_{k \to \infty} M^{k} \mathbf{x}^{0}$$

随着 k→∞ ，这个矩阵的维数和幂的次数都趋于无穷

如何讨论一个矩阵幂的收敛性呢？显然能算出来是最好的

事实上，这个矩阵可以**对角化**，它的幂最终转化为它**特征值的幂**

如果我们只想知道收敛性，那么只关注最大的特征值即可

以细分曲线为例

$$\begin{cases} \lambda_{\max} < 1, & \lambda_{\max}^{\infty} = 0 \\ \lambda_{\max} = 1, & \lambda_{\max}^{\infty} = 1 \\ \lambda_{\max} > 1, & \lambda_{\max}^{\infty} = \infty \end{cases}$$

所有顶点的坐标都“归零”，曲线消失

细分曲线收敛到某个形状

出现了无穷项，曲线“爆炸”

回到曲面细分算法

曲面细分可以分为两个步骤：

1. 产生新的图元但不改变形状 *splitting*
2. 重算坐标产生形变 *averaging*

相应的是两种规则：

- 拓扑规则指定如何分裂
- 几何规则指定如何形变

1. 分裂特别长的边
2. 坍缩特别短的边
3. 通过翻转一些边让顶点的度更加平均
4. 微调顶点的位置，使之更接近 $\mathcal{N}_1$ 邻域顶点的中心

为什么会有点像呢？

更一般的几何处理思路：离散曲面与离散微分

当我们将一个光滑的曲面**离散**为一个 Mesh

很多微分运算都会退化为**线性**形式

感兴趣的同学可以去查一下 Laplace-Baltrami 算子

一些利用日志进行调试的技巧

- 何时使用 logger，何时使用全局的 spdlog API?
- 应该用怎样的级别输出日志?
- 要输出多少日志？调试完之后要删掉多少?
- 如何过滤日志中的信息