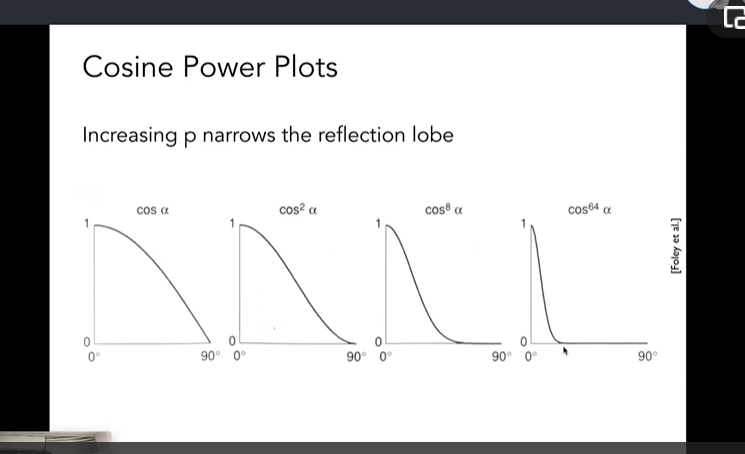
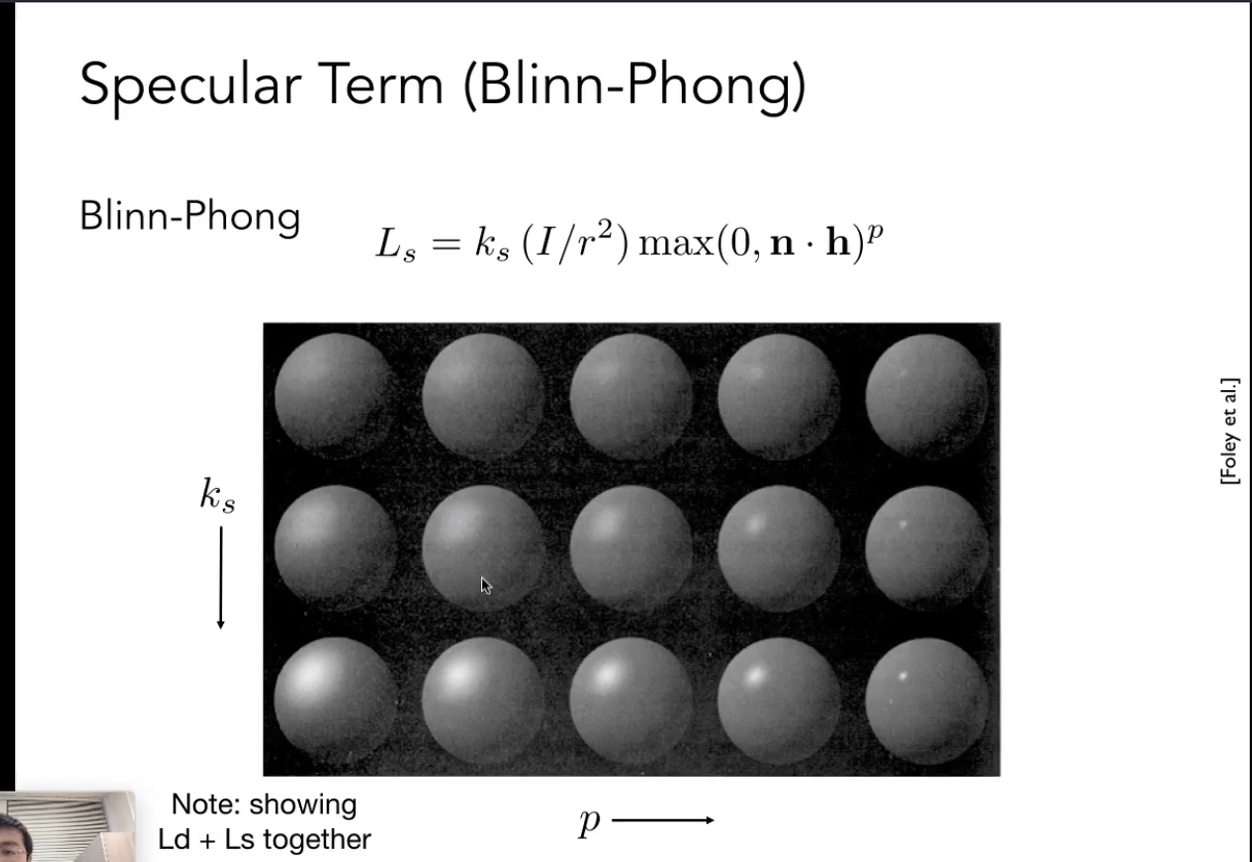
1. Blinn-Phong

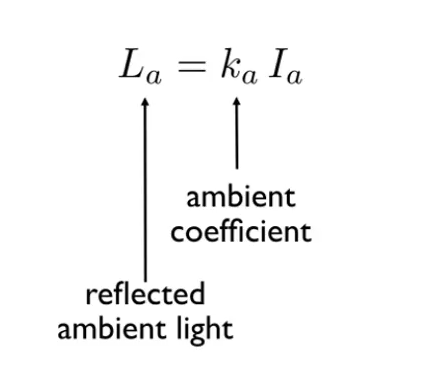
1.1 为什么要采用指数P？

单纯采用夹角余弦，“容忍度太高”，也就是高光的范围回比较大

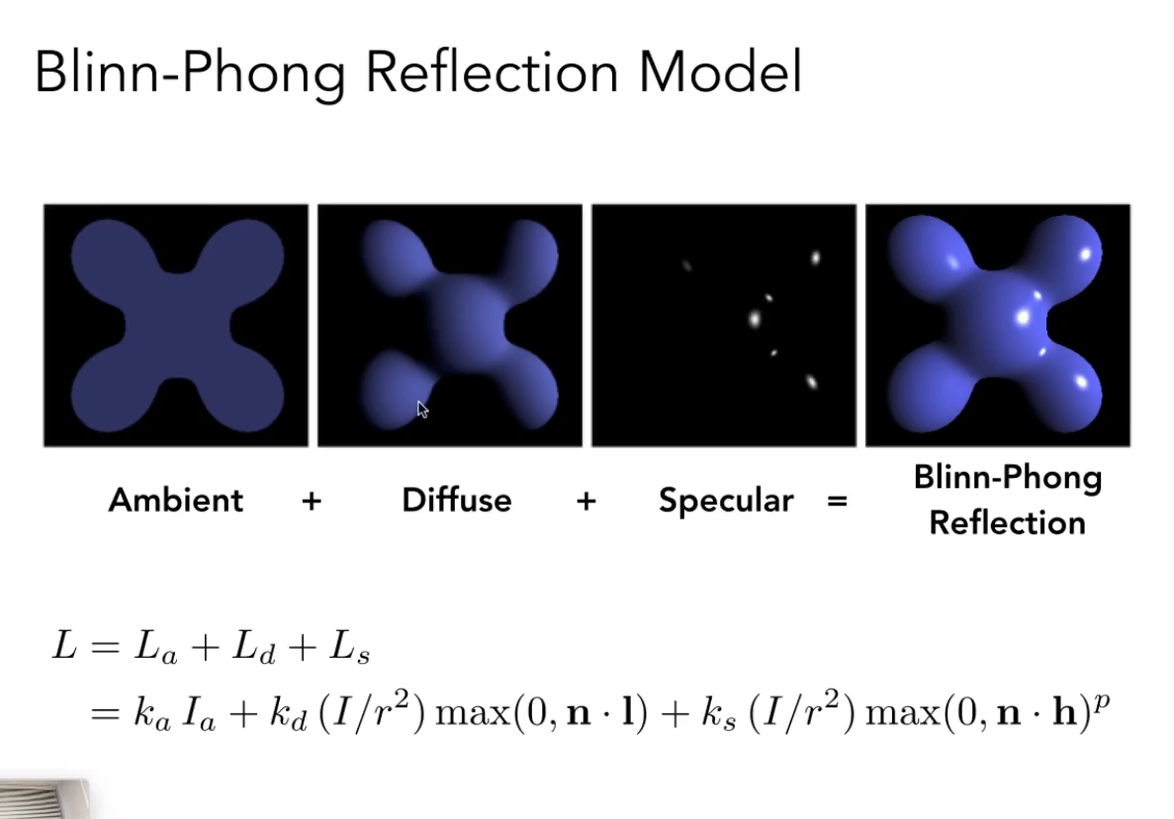


因此增加一个指数，使得高光能更集中与一点（如下，p控制高光大小，ks控制亮度）：

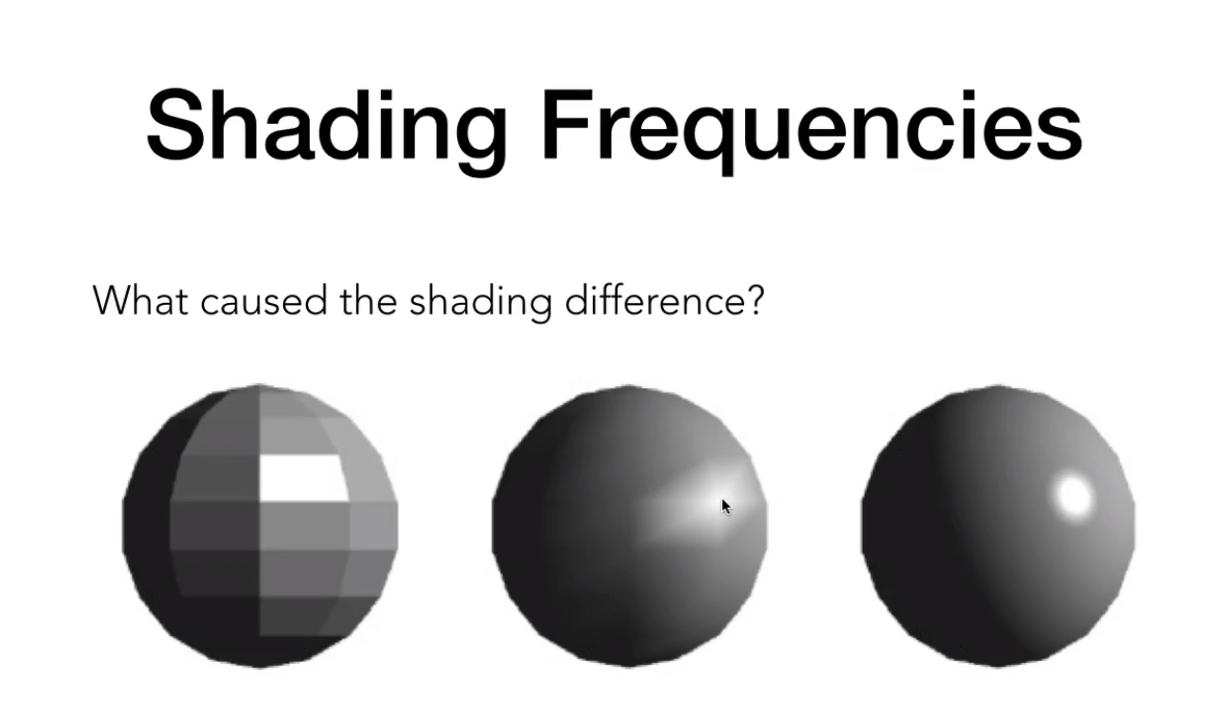
、

1.2 环境光：假设任何一个点收到的环境光都是相同的，作用是保证没有地方是完全黑色的，这只是一个简化的假设。

环境光+漫反射+镜面反射=Blinn-Phong 反射



1. 着色频率

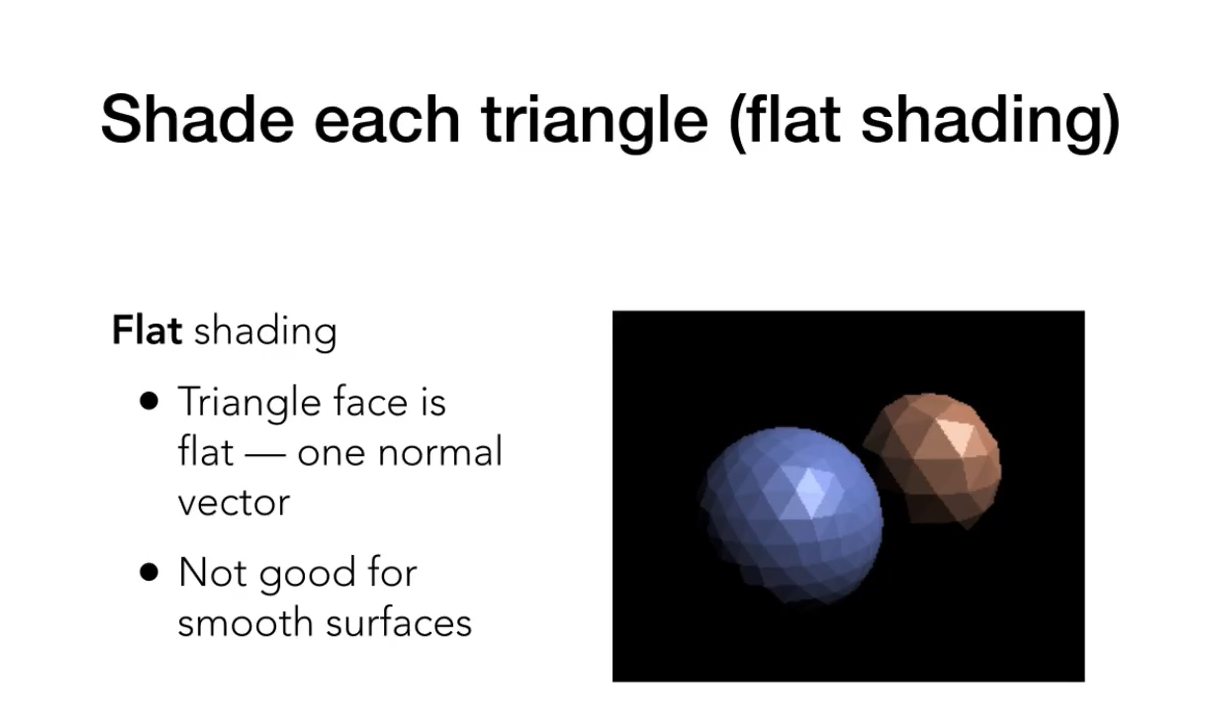


一个面只作一次shading；

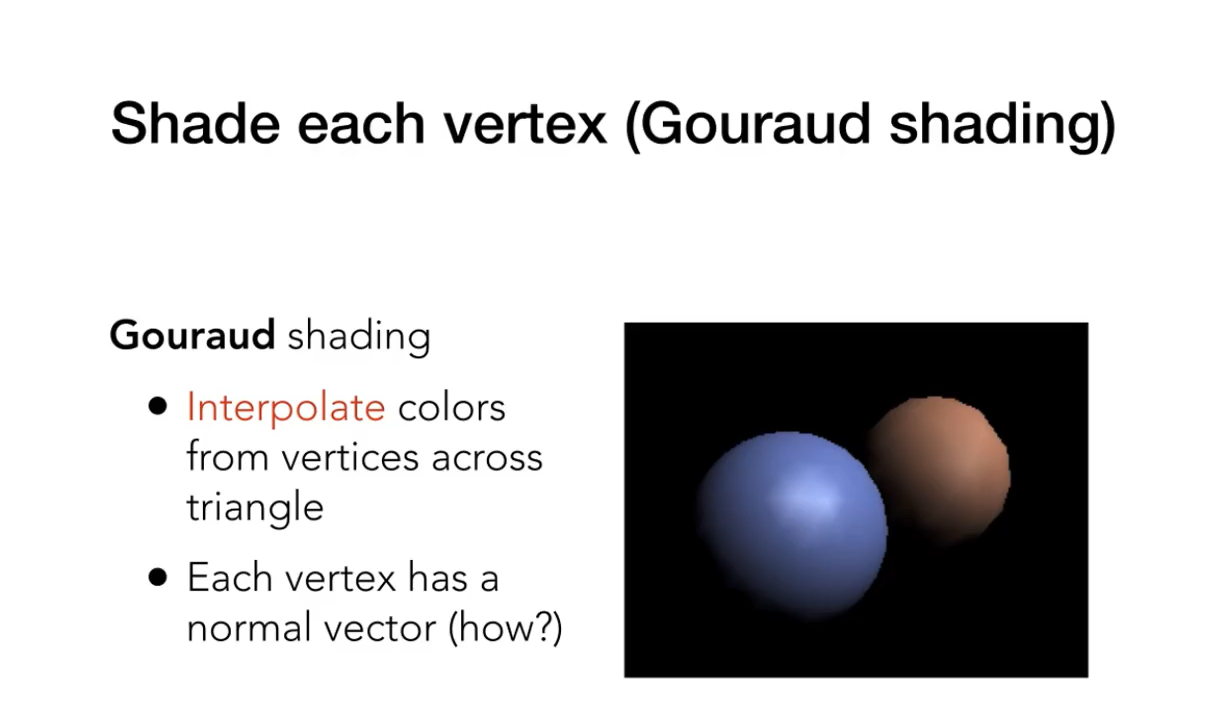
三角形每个顶点做一次shading，内部通过插值计算；

shading应用在每个像素（具体如何实现的？）

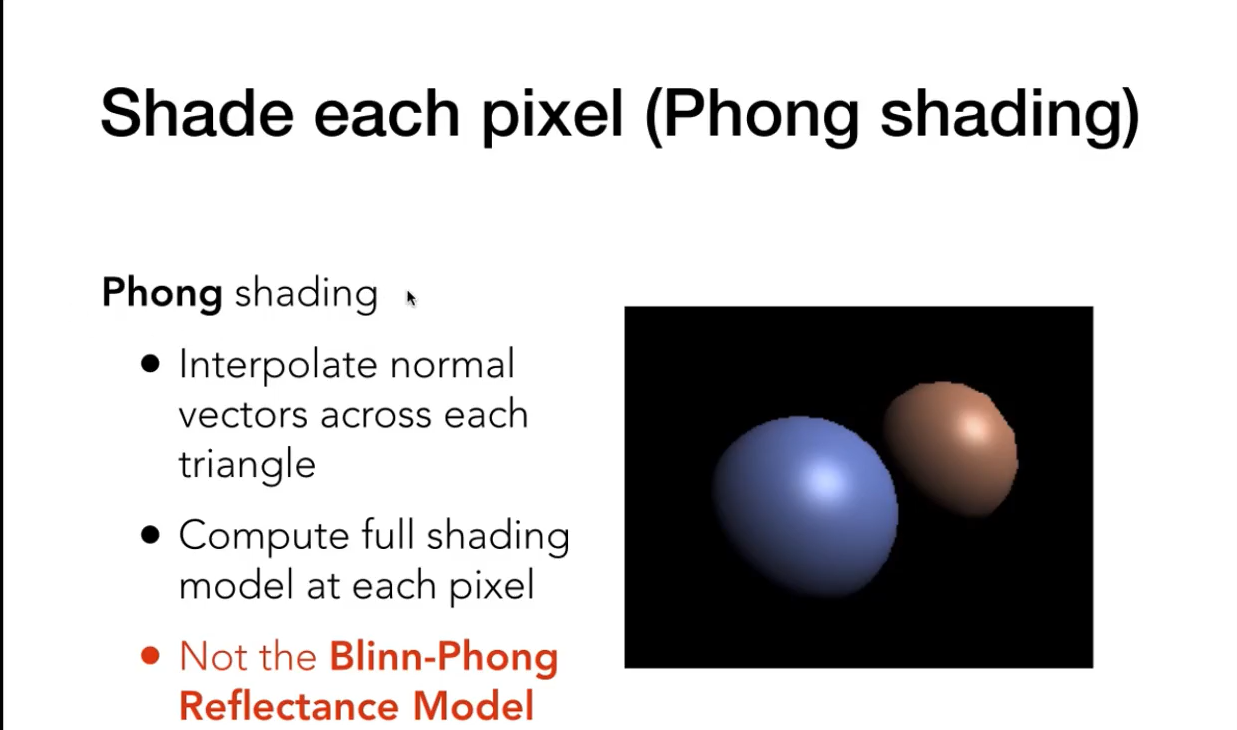
Flat shading逐三角形



Gouraud shading逐顶点



Phong Shading 逐像素

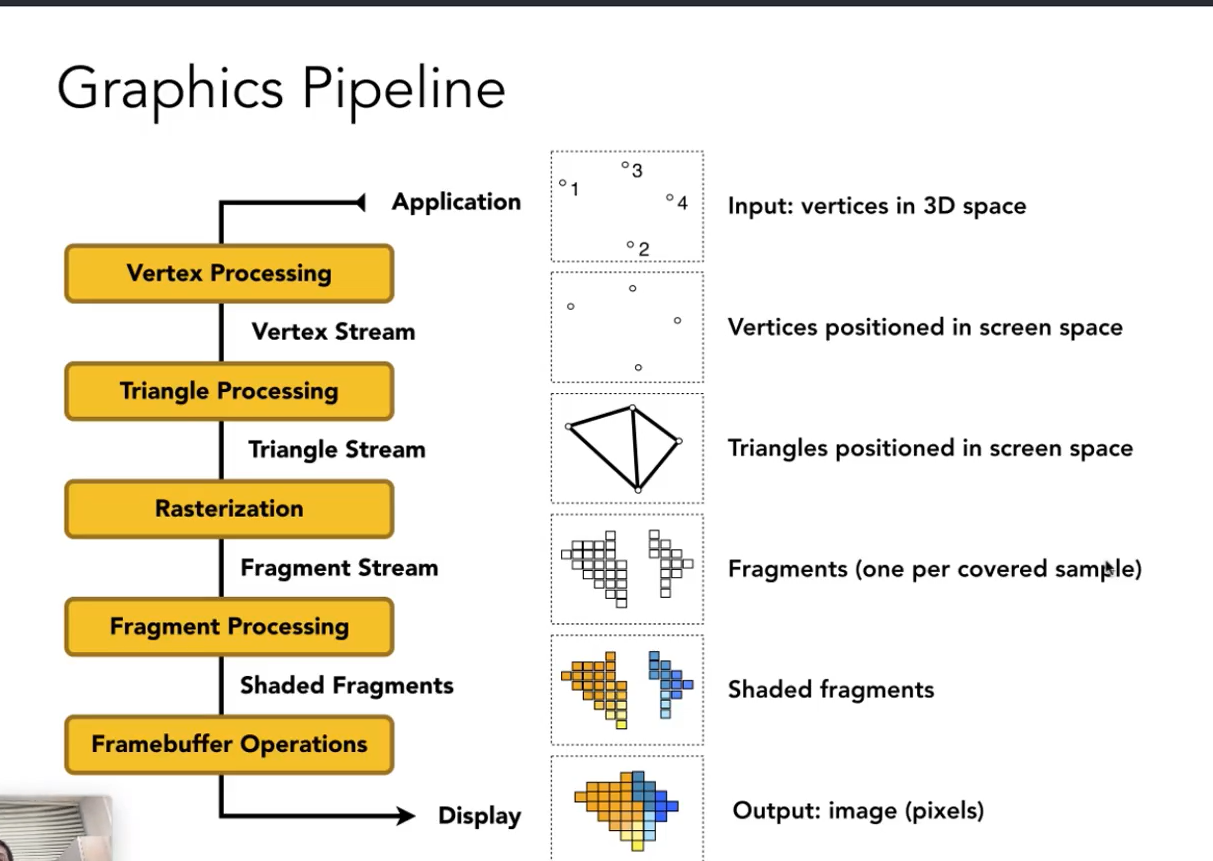


逐顶点法线：

逐像素的法线：



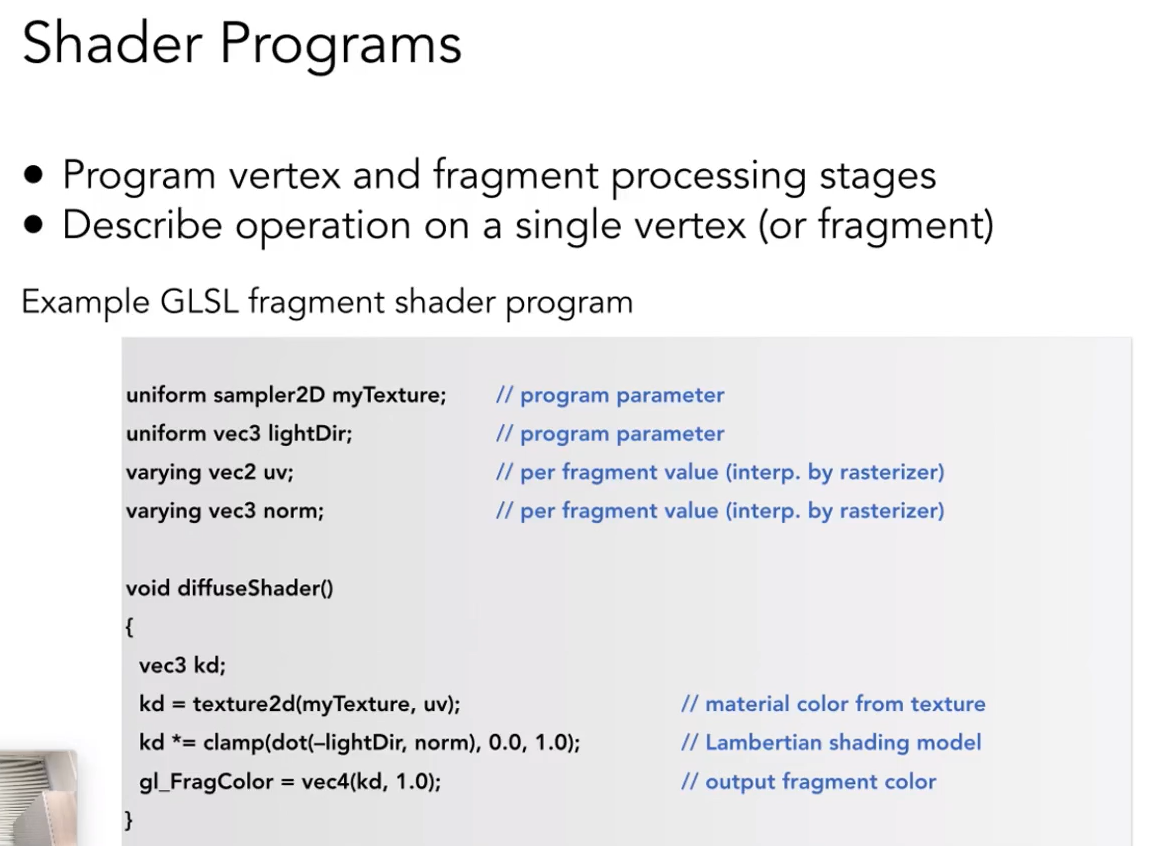
1. 实时渲染管线 Real-time rendering pipeline



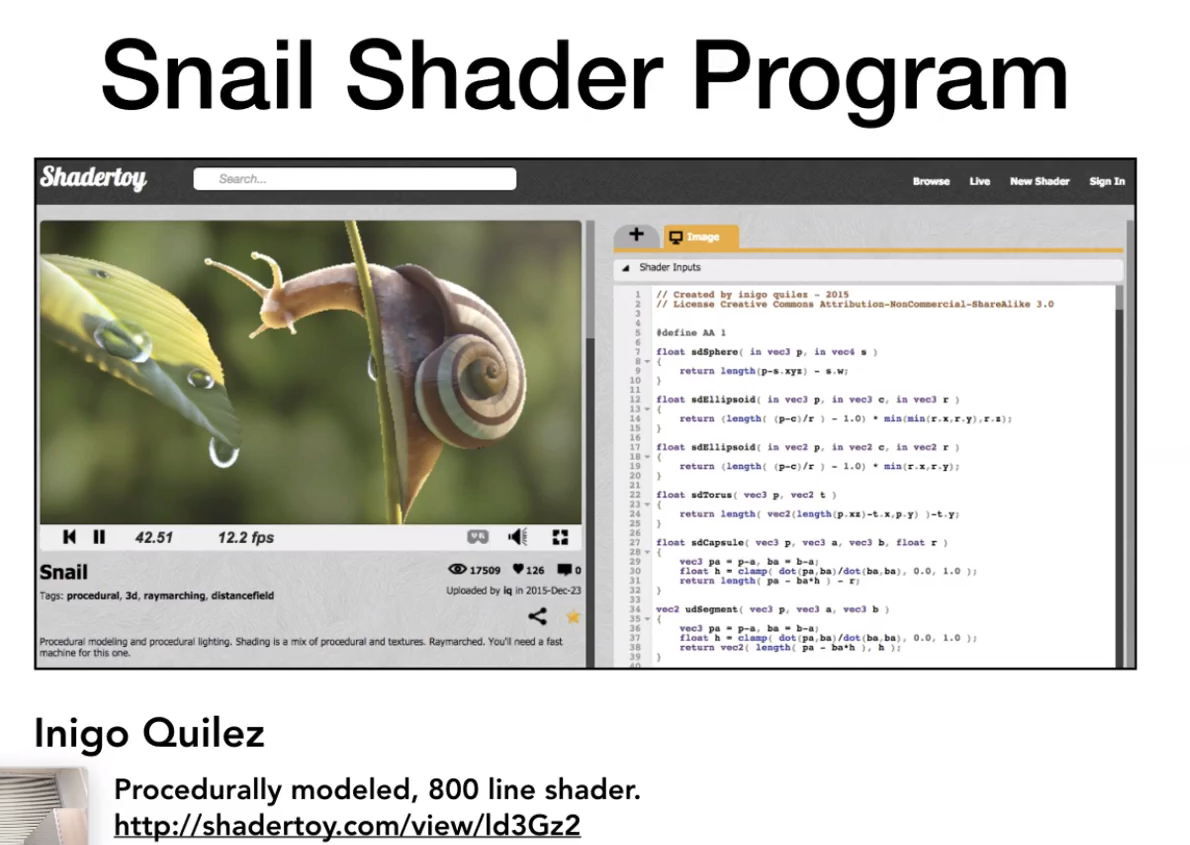
shader 程序

本质上是一个能在硬件上执行的语言：

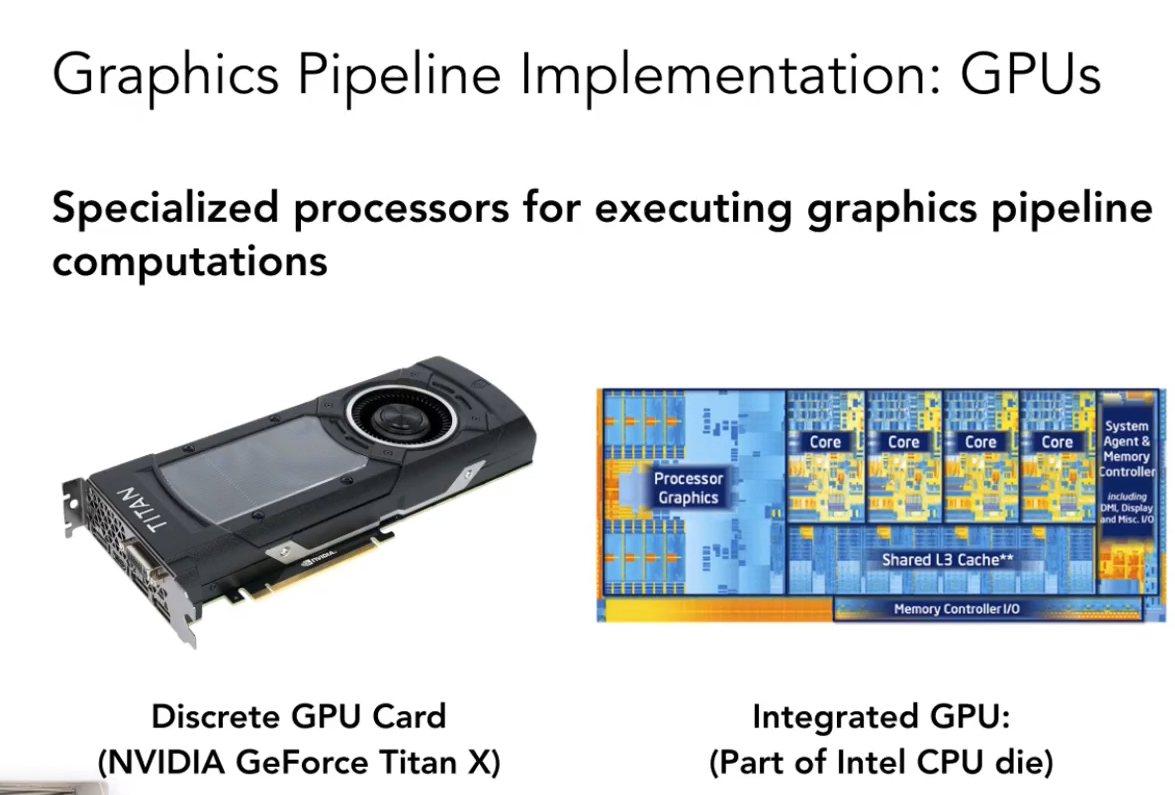
1. 在每个顶点或者每个像素上执行一次。
2. 如果是顶点的操作，是vertex shader；对像素/片段的操作，是fragment shader，告诉像素最后的颜色是什么



shadertoy 网站，在网页上直接执行

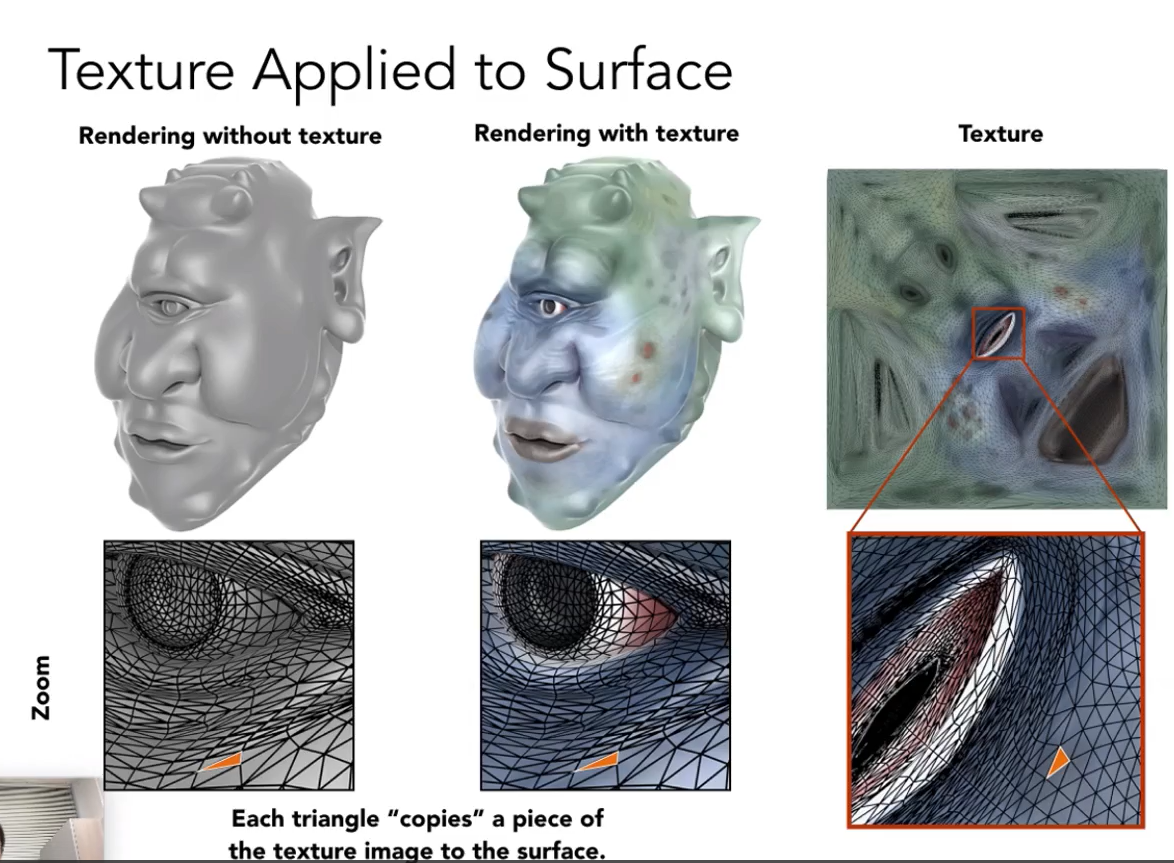


GPU



1. Texture mapping 纹理映射

物体表面任何一个点和纹理图片是一一对应的



纹理坐标系 u,v 不论纹理图片是正方形还是长方形，u v范围都是(0,1)

纹理可以被多次使用（无缝衔接）



如果知道了三角形顶点的(u, v)，如何知道三角形内部的(u, v)？答：使用插值 Interpolation