

3.5 顶点着色器

顶点着色器是图3.2所示功能管道中的第一阶段。虽然这是直接由程序员控制的第一阶段，但值得注意的是，在此阶段之前会发生一些数据操作。在DirectX所称的输入汇编器[175, 530, 1208]中，可以将多个数据流编织在一起以形成沿管道发送的顶点和图元集。例如，一个对象可以由一组位置和一组颜色表示。输入汇编器将通过创建具有位置和颜色的顶点来创建此对象的三角形（或线或点）。第二个对象可以使用相同的位置数组（以及不同的模型变换矩阵）和不同的颜色数组来表示。数据表示在[第16.4.5节](#)中详细讨论。输入汇编器也支持执行实例化。这允许使用每个实例的一些不同数据多次绘制对象，所有这些都使用单个绘制调用。[第18.4.2节](#)介绍了实例化的使用。

三角形网格由一组顶点表示，每个顶点与模型表面上的特定位置相关联。除了位置之外，还有与每个顶点相关联的其他可选属性，例如颜色或纹理坐标。曲面法线也在网格顶点处定义，这似乎是一个奇怪的选择。从数学上讲，每个三角形都有一个明确定义的表面法线，直接使用三角形的法线进行着色似乎更有意义。但是，在渲染时，三角形网格通常用于表示底层曲面，顶点法线用于表示该曲面的方向，而不是三角形网格本身的方向。[16.3.4节](#)将讨论计算顶点法线的方法。图3.7显示了代表曲面的两个三角形网格的侧视图，一个是平滑的，一个带有锐利的折痕。

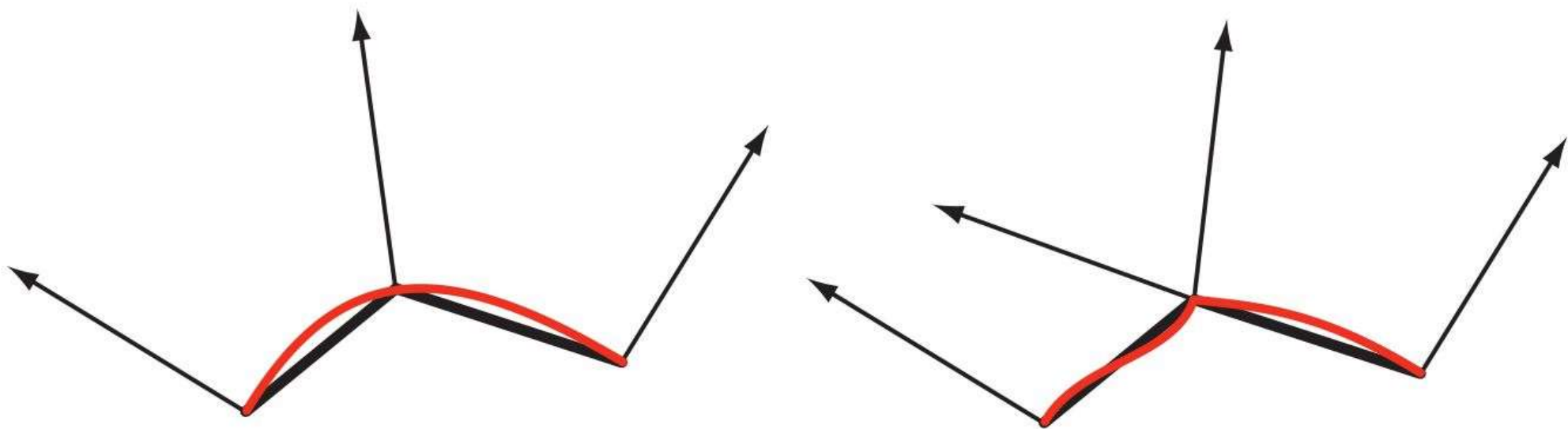


图3.7. 代表曲面（红色）的三角形网格（黑色，顶点法线）的侧视图。左侧平滑的顶点法线用于表示平滑的表面。在右侧，中间顶点已被复制并被赋予两个法线，代表一个折痕。

顶点着色器是处理三角形网格的第一阶段。顶点着色器无法使用描述形成哪些三角形的数据。顾名思义，它专门处理传入的顶点。顶点着色器提供了一种修改、创建或忽略与每个三角形顶点关联的的值的方法，例如其颜色、法线、纹理坐标和位置。通常，顶点着色器程序将顶点从模型空间转换为齐次裁剪空间（[第4.7节](#)）。一个最小化的顶点着色器必须始终输出这个齐次裁剪空间坐标。

顶点着色器与前面描述的统一着色器非常相似。传入的每个顶点都由顶点着色器程序处理，然后输出许多插入三角形或直线的值。顶点着色器既不能创建也不能破坏顶点，并且一个顶点生成的结果不能传递到另一个顶点。由于每个顶点都是独立处理的，因此可以将 GPU 上任意数量的着色器处理器并行应用于传入的顶点流。

输入组装通常表现为在执行顶点着色器之前发生的一个过程。这是物理模型通常与逻辑模型不同的示例。从物理上讲，获取数据以创建顶点可能发生在顶点着色器中，驱动程序将悄悄地每个着色器添加适当的指令，对程序员不可见。

随后的章节解释了几种顶点着色器效果，例如用于动画关节的顶点混合和轮廓渲染。顶点着色器的其他用途包括：

- 对象生成，通过只创建一次网格并使其由顶点着色器变形。
- 使用蒙皮和变形技术为角色的身体和面部制作动画。
- 程序变形，例如旗帜、布料或水的移动[802,943]。
- 粒子创建，通过沿管道发送退化（无区域）网格并根据需要为这些网格分配一个区域。
- 镜头失真、热雾、水波纹、页面卷曲和其他效果，通过使用整个帧缓冲区的内容作为经过程序变形的屏幕对齐网格上的纹理。
- 通过使用顶点纹理获取来应用地形高度场[40, 1227]。

使用顶点着色器完成的一些变形如图3.8所示。



图3.8 左边是一个普通的茶壶。由顶点着色器程序执行的简单剪切操作生成中间图像。在右侧，噪声函数创建了一个扭曲模型的场。（图片由FX Composer 2制作，由 NVIDIA Corporation提供。）

顶点着色器的输出可以通过几种不同的方式使用。通常的路径是每个实例的图元，例如三角形，然后被生成和光栅化，并且生成的单个像素片段被发送到像素着色器程序以继续处理。在某些GPU上，数据还可以发送到曲面细分阶段或几何着色器或存储在内存中。以下部分将讨论这些可选阶段。