# 8.2 纹理坐标

Direct3D的纹理坐标系由表示图像水平方向的*u*轴和表示图像垂直方向的*v*轴组成。坐标(*u*,*v*)指定了纹理上的一个元素，我们将该元素称为纹理元素（texel，译者注：texel是texture element的缩写），其中0≤*u*,*v*≤1。注意，*v*轴的正方向是“垂直向下”的（参见图8.2）。另外，将规范化坐标区间设为[0,1]，是因为这样可以使 Direct3D拥有一个独立于纹理尺寸的坐标空间；例如，无论纹理的实际尺寸是256×256、512×1024还是2048×2048，(0.5, 0.5)永远表示中间的纹理元素。同样，(0.25, 0.75)表示在水平方向上位于总宽度的1/4 处、在垂直方向上位于总高度的3/4处的纹理元素。现在，我们只讨论[0,1]区间内的纹理坐标，稍后会讲解当纹理坐标超出一范围时的处理方法。

****

**图8.2 纹理坐标系（texture coordinate system），有时也称为纹理空间（texture space）。**

对于每个3D三角形，我们都要在纹理上定义一个相应的2D三角形，以使纹理映射到 3D三角形上（参见图8.3）。

****

**图8.3 左图是3D空间中的一个三角形，右图是我们在纹理上定义的2D三角形，它会被映射到3D三角形上。**

若三角形的顶点坐标为**p**0，**p**1和**p**2，对应的纹理坐标为**q**0、**q**1和**q**2。对于3D三角形上的任意一个点(*x*,*y*,*z*)，它的纹理坐标(*u*,*v*)都可以通过在3D三角形表面对顶点纹理坐标进行线性插值得到，而线性插值中的参数*s*、*t*是相同的；也就是说：



若*s*≥0，*t*≥0，*s*+*t*≤1则，



通过一方式，三角形上的每个点都可以得到一个相应的纹理坐标。

为了使用纹理，我们需要再次修改顶点结构体，添加一对纹理坐标，指定纹理上的点， 使每个3D点都有一个相应的2D纹理点。这样，由3个顶点构成的每个3D三角形在纹理空间中都会有一个相应的2D纹理三角形（即，在每个2D纹理三角形和3D三角形之间建立对应关系）。

// Basic 32-byte vertex structure.

struct Basic32

{

 XMFLOAT3 Pos;

 XMFLOAT3 Normal;

 XMFLOAT2 Tex;

};

const D3D11\_INPUT\_ELEMENT\_DESC InputLayoutDesc::Basic32[3] =

{

 {"POSITION", 0, DXGI\_FORMAT\_R32G32B32\_FLOAT, 0, 0,

D3D11\_INPUT\_PER\_VERTEX\_DATA, 0},

 {"NORMAL", 0, DXGI\_FORMAT\_R32G32B32\_FLOAT, 0, 12,

D3D11\_INPUT\_PER\_VERTEX\_DATA, 0},

 {"TEXCOORD", 0, DXGI\_FORMAT\_R32G32\_FLOAT, 0, 24,

D3D11\_INPUT\_PER\_VERTEX\_DATA, 0}

};

**注意**：你可以创建一个2D纹理三角形和3D三角形差别很大的“奇怪”纹理映射。结果是，当这个2D纹理映射到3D三角形上时，会出现拉伸和扭曲，看起来不好。例如，将一个尖三角形映射到一个正三角形上就会发生拉伸现象。通常，纹理扭曲应尽量避免，除非艺术家期望这种扭曲的效果。

看一下图8.3，我们将整张纹理映射到立方体的每个面上，但这并不是必须的。我们可以只将纹理的一部分映射到几何体上。我们可以将几张毫不相关的图像合并在一张大纹理贴图上（叫做纹理贴图集texture atlas），然后将它用于不同的对象（图8.4）。纹理坐标决定了将那一部分纹理映射到三角形上。

****

**图8.4. 一张存储了4张纹理的纹理贴图集。设置每个顶点的纹理坐标就可以将期望的部分纹理映射到几何体上。**