# 8.5 纹理采样

我们知道，**Texture2D**对象用于在effect文件中表示纹理。不过，还有一种与纹理相关的**SamplerState**（采样器）对象。它用于描述如何使用过滤器访问纹理资源。下面是它的一些例子：

// 在倍增、缩减、多级渐进纹理上使用线性过滤。

SamplerState mySampler0

{

Filter = MIN\_MAG\_MIP\_LINEAR;

};

// 在缩减上使用线性过滤，倍增和多级渐进纹理上使用点过滤。

SamplerState mySampler1

{

Filter = MIN\_LINEAR\_MAG\_MIP\_POINT;

};

// 在缩减上使用点过滤，倍增上使用线性过滤，多级渐进纹理上使用点过滤。

SamplerState mySampler2

{

Filter = MIN\_POINT\_MAG\_LINEAR\_MIP\_POINT;

};

// 在倍增、缩减、多级渐进纹理上使用各向异性过滤。

SamplerState mySampler3

{

Filter = ANISOTROPIC;

MaxAnisotropy = 4;

};

注意，对于各向异性过滤，我们必须指定各向异性的最大值，这个值介于1至16之间。大数值需要耗费更多的资源，但效果更好。你可以从这些例子中猜到一些其他的标志值，或者你可以在SDK文档中查找**D3D11\_FILTER**枚举类型。我们很快就会看到与采样器相关的其他属性，但是对于我们的第一个演示程序来说现在的这些内容已经足够了。

现在，传入像素着色器的每个像素都有一对纹理坐标，我们使用如下语法完成实际的纹理采样工作：

Texture2D gDiffuseMap;

struct VertexOut

{

float4 posH : SV\_POSITION;

float3 posW : POSITION;

float3 normalW : NORMAL;

float2 Tex : TEXCOORD;

};

SamplerState samAnisotropic

{

Filter = ANISOTROPIC;

MaxAnisotropy = 4;

};

float4 PS(VertexOut pin, uniform int gLightCount) : SV\_Target

{

// 从纹理中提取颜色

float4 texColor = gDiffuseMap.Sample(samAnisotropic, pin.Tex);

...

可以看到，我们使用**Texture2D::Sample**方法对一个纹理进行采样。通过第一个参数传递SamplerState对象，通过第二个参数传递像素的纹理坐标(*u*,*v*)。该方法使用由**SamplerState**对象指定的过滤方式从纹理贴图中返回位于点(*u*,*v*)处的插值颜色。

**注意**：HLSL类型**SamplerState**对应**ID3D11SamplerState**接口。采样状态也可以在应用程序中由**ID3DX11EffectSamplerVariable::SetSampler**方法进行设置。见**D3D11\_SAMPLER\_DESC**和**ID3D11Device::CreateSamplerState**。和渲染状态一样，必须在初始化阶段就创建采样状态。