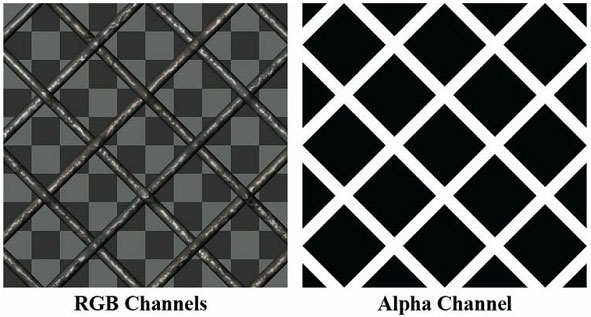
# 9.7 裁剪像素

有时，我们希望完全丢弃某个源像素，使它不再接受后续处理。这一工作可以由HLSL的内置函数**clip(x)**来实现。该函数只能在像素着色器中使用，当x<0时丢弃当前像素，使之不再接受后续处理。该函数在渲染如图9.9所示的铁丝网纹理时非常有用。也就是说，它非常适合于渲染那些完全不透明或者完全透明的像素。

****

**图9.9 带有alpha通道的铁丝网纹理。clip函数将丢弃那些带有黑色alpha值的像素，不对这些像素进行绘制；只有铁丝网部分会保留下来。从本质上讲，alpha通道剔除了纹理中的“非铁丝网”像素。**

在像素着色器中，我们攫取了漫反射纹理的alpha分量。当它的值接近于0时，我们将该像素视为完全透明，丢弃该像素，不再对它进行后续处理。

float4 PS(VertexOut pin, uniform int gLightCount, uniform bool gUseTexure, uniform bool gAlphaClip, uniform bool gFogEnabled) : SV\_Target

{

// 插值后的法线需要重新规范化

pin.NormalW = normalize (pin.NormalW);

// toEye矢量用于光照计算

float3 toEye = gEyePosW - pin.PosW;

// 保存表面顶点离开相机的距离信息

float distToEye = length(toEye);

// 规范化

toEye /= distToEye;

// 初始化纹理颜色

float4 texColor = float4(1, 1, 1, 1);

if (gUseTexure)

{

// 采样纹理

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

if (gAlphaClip)

{

// 如果纹理的alpha<0.1，则丢弃像素。

// 注意，我们应该尽可能早地进行这个测试，这样我们就可以及早退出

// shader，忽略其余shader代码。

clip(texColor.a - 0.1f);

}

}

…

因为我们可能只在某些几何体上进行裁剪操作，所以只有在参数**gAlphaClip**设置为true的情况下我们才进行裁剪，这样我们就可以根据特定的shader切换裁剪。

注意，使用混合也可以得到同样的效果，只是这种（裁剪）方式的运行效率更高一些。这种方式即不需要进行任何混合计算，也不需要考虑物体的绘制顺序。而且，它可以在像素着色器中尽早丢弃像素，避免执行不必要的像素着色器指令（被丢弃的像素不会参与任何计算）。

**注意**：由于过滤器的作用，alpha通道可能会变得有些模糊，所以当裁剪像素时，你应该保留一些容差值。例如，裁剪alpha值接近于0的像素，而不必让alpha值精确为0。

图9.10是“Blend”演示程序的屏幕截图。它使用透明混合绘制了半透明的水体，使用了新的铁丝网纹理，并且在像素着色器中加入了裁剪测试功能。另一个值得注意的地方是，由于我们现在要透过立方体看到背面的铁丝网纹理，所以我们希望禁用背面消隐功能：

D3D11\_RASTERIZER\_DESC noCullDesc;

ZeroMemory(&noCullDesc, sizeof(D3D11\_RASTERIZER\_DESC));

noCullDesc.FillMode = D3D11\_FILL\_SOLID;

noCullDesc.CullMode = D3D11\_CULL\_NONE;

noCullDesc.FrontCounterClockwise = false;

noCullDesc.DepthClipEnable = true;

ID3D11RasterizerState \* NoCullRS;

HR(device->CreateRasterizerState(&noCullDesc, &NoCullRS));

…

// 因为铁丝网纹理包含透明区域，我们可以透过立方体看到背面的三角形，所以我们希望禁用背面消隐功能

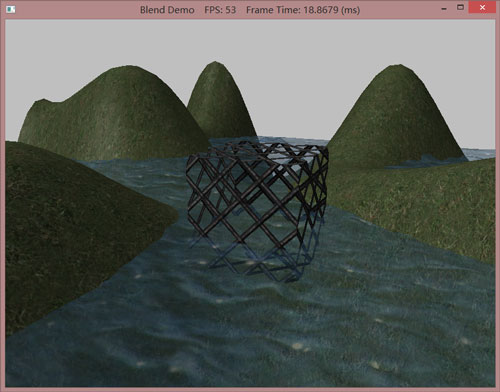
md3dImmediateContext->RSSetState(NoCullRS);

boxTech->GetPassByInde x(p)->Apply(0, md3dImmediateContext);

md3dImmediateContext->DrawIndexed(36, 0, 0);

// 恢复为默认的渲染状态

md3dImmediateContext->RSSetState(0);

****

**图9.10“Blend”演示程序的屏幕截图。**