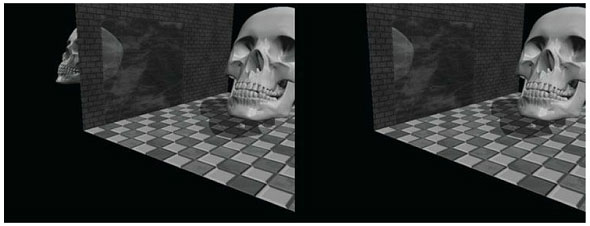
# 10.1 深度/模板格式及清空操作

模板缓冲区（stencil buffer）是一种用来实现特殊效果的离屏（off-screen）缓冲区。模板缓冲的大小与后台缓冲及深度缓冲的大小相同，也就是说，模板缓冲的第*ij*个像素对应于后台缓冲和深度缓冲第*ij*个像素。我们在4.1.5节的“注意”中提到，当指定一个模板缓冲时，它总是与深度缓冲共享相同的内存空间。尤如名字所指出的，模板缓冲区的用法就像是模板一样，它可以挡住某些像素片段，不让它们存入后台缓冲。（译者注：比如喷油漆时使用的图案模板，先把模板贴在汽车上或者其他什么地方，然后开始喷油漆。在模板镂空的地方会有油漆喷到汽车上，而没有镂空的地方会挡住油漆。在喷完之后，揭下模板，图案就喷涂在汽车上了。）

例如，当实现一个镜像效果时，我们需要反射镜子对面的物体；不过，我们希望镜像只显示在镜子里面。我们可以使用模板缓冲区来控制镜像范围，阻止镜像绘制到镜子之外的区域（参见图10.1）。

****

**图10.1 （左）头骨的镜像在镜中的显示是正确的，由于没有通过深度测试，镜像也正确地被砖墙阻挡。但是，我们能看到墙后的镜像，这样就不对了（镜像应只出现在镜子中）。（右）通过使用模板缓冲，我们可以阻止头骨的镜像绘制到镜子之外的区域。**

我们可以通过**ID3D11DepthStencilState**接口控制模板缓冲（和深度缓冲）。与混合一样，该接口也提供了一套灵活而强大的功能集合。要学习如何高效地使用模板缓冲区，最有效的方法是仔细研究现有的示例应用程序。当你弄懂了几个使用模板缓冲区的应用程序之后，就会对它有一个更清晰的认识，知道该如何用它来解决实际工作问题。

**学习目标：**

1．了解如何使用ID3D11DepthStencilState接口控制深度缓冲和模板缓冲。

2．学习如何通过模板缓冲来实现镜像效果，阻止镜像绘制到镜子之外的区域。

3．会辨认双重混合以及理解模板缓冲如何防止出现这种情况。

4．解释什么是深度复杂性，介绍两种方法测量深度复杂性。

前面讲过，深度/模板缓冲区是一个纹理，在创建它时必须为它指定某种数据格式。可用于深度/模板缓冲区的格式有：

**1．DXGI\_FORMAT\_D32\_FLOAT\_S8X24\_UINT**：32位浮点深度缓冲区。其中8位用于模板缓冲区，每个模板元素的取值范围是[0,255]。其余24位闲置。

**2．DXGI\_FORMAT\_D24\_UNORM\_S8\_UINT**：24位用于深度缓冲区，每个深度元素的取值范围是[0,1]；8位用于模板缓冲区，每个模板元素的取值范围是[0,255]。

在D3DApp框架中，我们使用如下格式创建深度缓冲区：

depthStencilDesc.Format = DXGI\_FORMAT\_D24\_UNORM\_S8\_UINT;

另外，在每帧开始时，模板缓冲区应该被重置为某些初始值。该操作由如下方法完成（它可以同清空深度/模板缓冲区）：

void ID3D11Device::ClearDepthStencilView(

ID3D11DepthStencilView \*pDepthStencilView,

UINT ClearFlags, FLOAT Depth, UINT8 Stencil);

**1．pDepthStencilView**：所要清空的深度/模板缓冲区的视图的指针。

**2．ClearFlags**：当设为**D3D11\_CLEAR\_DEPTH**时，表示只清空深度缓冲区；当设为**D3D11\_CLEAR\_STENCIL**时，表示只清空模板缓冲区；当设为**D3D11\_CLEAR\_DEPTH| D3D11\_CLEAR\_STENCIL**时，表示同时清空深度/模板缓冲区。

**3．Depth**：深度缓冲区元素的初始值，它必须是一个在0≤*x*≤1之间的浮点值。

**4．Stencil**：模板缓冲区元素的初始值，它必须是一个在0≤*n*≤255之间的整数值。

我们在示例的每帧中都会调用这个方法，例如：

void MirrorApp::DrawScene()

{

md3dImmediateContext->ClearRenderTargetView(mRenderTargetView,

reinterpret\_cast<const float\*>(&Colors::Black));

md3dImmediateContext->ClearDepthStencilView(mDepthStencilView,

D3D11\_CLEAR\_DEPTH|D3D11\_CLEAR\_STENCIL,1.0f,0);

}