# 8.11 压缩纹理格式

一个虚拟场景可能会载入数百幅纹理，而些纹理会占用大量的GPU内存（记住，我们必须让所有的纹理驻留在GPU内存中，只有这样着色器才能快速地访问纹理）。为了缓解GPU内存压力，Direct3D提供了以下压缩纹理格式：BC1、BC2、BC3、BC4、BC5，BC6和BC7。

1．BC1（**DXGI\_FORMAT\_BC1\_UNORM**）：该格式支持3个颜色通道，仅用1位（开/关）表示alpha分量。

2．BC2（**DXGI\_FORMAT\_BC2\_UNORM**）：该格式支持3个颜色通道，仅用4位表示alpha分量。

3．BC3**（DXGI\_FORMAT\_BC3\_UNORM）**：该格式支持3个颜色通道，以8位表示alpha分量。

4．BC4**（DXGI\_FORMAT\_BC4\_UNORM）**：该格式支持1个颜色通道（例如，灰阶图像）。

5．BC5**（DXGI\_FORMAT\_BC5\_UNORM）**：该格式支持两个颜色通道。

6．BC6**（DXGI\_FORMAT\_BC6\_UF16）**：该格式用于压缩的HDR（高动态范围，high dynamic range）图像数据。

7．BC7**（DXGI\_FORMAT\_BC7\_UNORM）**：该格式用于高质量的RGBA压缩。特别的有，这个格式可以显著地减少由于压缩法线贴图带来的错误。

关于这些格式的更多信息，请读者在SDK文档的索引中查找“Block Compression（块压缩）”。

**注意**：压缩纹理只能作为输入数据传递给渲染管线的着色器阶段。

**注意**：因为块压缩算法使用4×4像素块，所以纹理尺寸必须为4的倍数。

使用这些格式的好处是它们可以压缩存储在GPU内存中，当使用时由GPU实时解压缩。

我们可以在载入纹理时使用**D3DX11CreateShaderResourceViewFromFile**函数的**pLoadInfo**参数，让Direct3D把纹理转换为某种压缩格式。例如下面的代码，它载入了一个BMP文件：

D3DX11\_IMAGE\_LOAD\_INFO loadInfo;

loadInfo.Format = DXGI\_FORMAT\_BC3\_UNORM;

HR(D3DX11CreateShaderResourceViewFromFile(md3dDevice,

L"Textures/darkbrick.bmp", &loadInfo, 0, &mDiffuseMapSRV, 0 ));

// 从资源视图获取2D纹理

ID3D11Texture2D\* tex;

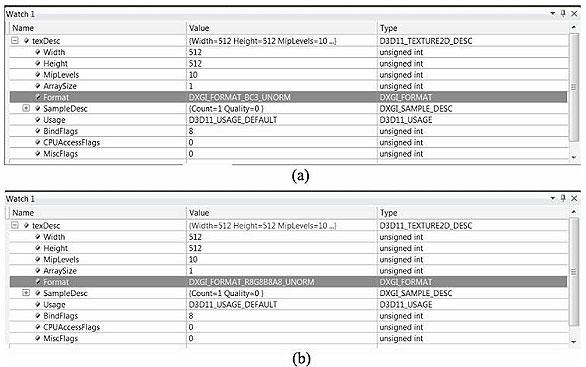
mDiffuseMapSRV->GetResource((ID3D11Resource\*\*)&tex);

// 从2D纹理获取纹理描述

D3D11\_TEXTURE2D\_DESC texDesc;

tex->GetDesc(&texDesc);

图8.17a是在调试器中看到的**texDesc**值；它包含了我们指定的压缩纹理格式。当参数**pLoadInfo**设为空值时，Direct3D将使用源图像格式（图8.17b），即非压缩格式**DXGI\_FORMAT\_R8G8B8A8\_UNORM**。

****

**图8.17 (a) 使用压缩格式DXGI\_FORMAT\_BC3\_UNORM创建纹理。(b)使用非压缩格式DXGI\_FORMAT\_R8G8B8A8\_UNORM创建纹理。**

另外，你也可以使用DDS（DirectDraw Surface）格式直接存储压缩纹理。操作步骤是运行SDK目录D:\Microsoft DirectX SDK(June 2010)\Utilities\Bin\x86中的DirectX纹理工具（DXTex.exe），打开你的图像文件。然后执行菜单命令**Menu>Format>Change Surface Format**，选择DXT1、DXT2、DXT3、DXT4或DXT5，并保存DDS文件。这些格式其实是Direct3D 9的压缩纹理格式，DXT1相当于BC1，DXT2和DXT3相当于BC2，DXT4和DXT5相当于BC3。例如，当我们用**D3DX11CreateShaderResourceViewFromFile** 函数载入一个DXT1格式的DDS文件时，它的实际纹理格式为DXGI\_FORMAT\_BC1\_UNORM：

HR(D3DX11CreateShaderResourceViewFromFile(md3dDevice,

L"Textures/darkbrick.dds", 0, 0, &mDiffuseMapSRV, 0 ));

// 从资源视图获取2D纹理

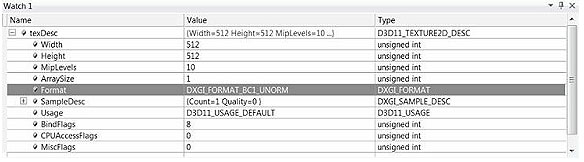
ID3D11Texture2D\* tex;

mDiffuseMapSRV->GetResource((ID3D11Resource \*\*)&tex);

// 从2D纹理获取纹理描述

D3D11\_TEXTURE2D\_DESC texDesc;

tex->GetDesc(&texDesc);

****

**图8.18：使用DXGI\_FORMAT\_BC1\_UNORM格式创建纹理。**

注意，如果DDS文件使用了某种压缩格式，那么我们可以将**pLoadInfo**参数设为空值，**D3DX11CreateShaderResourceViewFromFile**会自动使用由文件指定的压缩格式。

对于BC4和BC5格式，你可以使用NVIDIA Texture Tools（[http://code.google.com/p/nvi ia-texture-tools/](http://code.google.com/p/nvi%20ia-texture-tools/)）。对于BC6和BC7格式，DirectX SDK包含了一个叫做“BC6HBC7EncoderDecoder11”的示例。这个程序可以用来将纹理转换为BC6或BC7格式。这个示例包含了完整的源代码，所以你可以将它整合到你自己的素材管道中。而且，若你的显卡支持计算着色器，这个示例还会使用GPU进行转换工作，这比通过CPU进行转换快得多。

你还可以用DirectX纹理工具生成多级渐近纹理层（**Menu>Format>Generate Mip Maps**），并保存为DDS文件。通过一方式，多级渐近纹理层可以被提前计算出来并保存在文件中，节省载入时的计算时间（它们只需要被载入 。

将纹理存储为DDS压缩文件的另一个好处是可以减少磁盘空间的占用量。