**Unity3D教程：法线贴图**

Posted on 2013年01月11日 by U3d / [Unity3D脚本/插件](http://www.unitymanual.com/category/script)/被围观 351 次

一：法线贴图的原理

光照效果很大程度上是由垂直于物体表面的法线决定的，因为法线影响反射光的方向。均匀垂直的法线是镜面贴图。但是有时候我们会给一个平面使用砖墙贴图，砖墙应该是凹凸不平的，而如果让砖墙使用该平面的法线的话，画面就会很假，神马？一面墙像镜子一样反光=。=

而如果按真实砖墙去做模型的话，即做高精度模型，一方面制作麻烦，另一方面运行时对性能损耗大。

法线贴图就是来解决这个问题的。法线贴图就是把法线信息储存在一张图里。使用法线贴图时，通常顶点数和三角形面数只有高精度模型的十分之一不到。

二：法线贴图的实现

将材质贴图对应的法线 绘制在一张贴图上。将贴图对应点的单位法线向量信息float3(x,y,z) 储存在图对应的颜色里color(r,g,b)里，其中x,y,z分别对应r,g,b。单位法线向量 float3(x,y,z),x，y,z的取值范围是 [-1,1]。在法线贴图中被压缩在颜色的范围[0,1]中，所以需要转换：

颜色 = 0.5 \* 法线 + 0.5;  
线 = 2 \* (颜色 - 0.5);

三：法线贴图的使用

主要步骤

（1）对法线贴图进行采样，取得压缩在颜色空间[0,1]里的法线

float4 packedNormal = tex2D(\_NormalMap, IN.uv\_MainTex);

（2）将压缩在[0,1]里的法线转换至3D空间[-1,1] (因为是单位向量)

float3 expand(float3 v) { return (v - 0.5) \* 2; }

之后使用该法线即可，方法与16讲里一样。

具体实现详见本文末的脚本。

四：法线贴图的格式

法线贴图主要分为2个类别：

（1）RGB法线贴图,即上面使用的。通常呈蓝色。（后缀可以是常见的.png .jpg等）

（2）压缩格式的法线贴图。例如DXT5nm（后缀名为.dds)

dds是DirectDraw Surface的缩写，实际上，它是DirectX纹理压缩（DirectX Texture Compression，简称DXTC）的产物。DXTC减少了纹理内存消耗的50%甚至更多，有3种DXTC的格式可供使用，它们分别是DXT1，DXT3和DXT5。

有关DDS的延伸阅读

压缩法线贴图的原理：

法线（x,y,z）是一条单位向量。故X2 + Y2 +Z2 =1。所以知道了x,y,z里的任意两个，剩下的那个就可以通过计算得出。所以我们就可以使用2个通道的图储存x,y,z里的两个值，将xyz里剩余的值省略，通过计算得出。

压缩法线贴图的好处：

压缩后的法线贴图，大小只有原来的1/4左右，故可以使用更大或者更多的贴图来提升画面品质。

Unity3d的法线贴图：

Unity3d使用的压缩法线贴图是DXT5nm格式的。有A和G两个通道。对于法线（x,y,z） A对应x,G对应y。

对压缩法线贴图的采样依然是如下函数：

float4 packedNormal = tex2D(\_NormalMap, IN.uv\_MainTex);

packedNormal.w对应A通道,即法线的x。

packedNormal.y对应G通道,即法线的y。

范围依然是[0,1], 依然需要转换至[-1,1]。

对DXT5nm法线贴图进行转换的函数如下，其中v传入packedNormal

float3 expand(float3 v)  
{  
fixed3 normal;  
normal.xy = v.wy \* 2 - 1;  
normal.z = sqrt(1 - normal.x\*normal.x - normal.y \* normal.y);  
return normal;  
}

Unity3d的标准法线解压函数是fixed3 UnpackNormal(fixed4 packednormal)。打开UnityCG.cginc找到对应函数：

inline fixed3 UnpackNormal(fixed4 packednormal)  
{  
#if defined(SHADER\_API\_GLES) && defined(SHADER\_API\_MOBILE)  
return packednormal.xyz \* 2 - 1;  
#else  
fixed3 normal;  
normal.xy = packednormal.wy \* 2 - 1;  
normal.z = sqrt(1 - normal.x\*normal.x - normal.y \* normal.y);  
return normal;  
#endif  
}

该函数定义的如果是移动平台或者OpenGL ES，那么断定使用的是RGB法线贴图，否则则为DXT5nm贴图。但实际上移动平台也可以用压缩格式的法线贴图，而Windows也能使用RGB法线贴图。故不建议使用UnpackNormal函数，建议根据法线贴图的具体格式来使用自己写的对应函数。

脚本：

// Shader: 带法线贴图的Surface Shader  
// Author: 风宇冲  
Shader "Custom/3\_NormalMap" {  
Properties  
{  
\_MainTex ("Texture", 2D) = "white" {}  
\_NormalMap ("NormalMap", 2D) = "white" {}  
}  
Subshader  
{  
CGPROGRAM  
#pragma surface surf BlinnPhong  
struct Input  
{  
float2 uv\_MainTex;  
};  
//法线范围转换：单位法线 float3(x,y,z),x，y,z的取值范围是 [-1,1]。在法线贴图中被压缩在颜色的范围[0,1]中，所以需要转换  
//(1)RGB法线贴图  
float3 expand(float3 v) { return (v - 0.5) \* 2; }  
//(2)DXT5nm法线贴图  
float3 expand2(float4 v)  
{  
fixed3 normal;  
normal.xy = v.wy \* 2 - 1;  
normal.z = sqrt(1 - normal.x\*normal.x - normal.y \* normal.y);  
return normal;  
}  
sampler2D \_MainTex;  
sampler2D \_NormalMap;  
void surf(Input IN,inout SurfaceOutput o)  
{  
half4 c = tex2D(\_MainTex, IN.uv\_MainTex);  
o.Albedo = c.rgb;  
o.Alpha = c.a;  
//对法线贴图进行采样，取得压缩在颜色空间里的法线（[0,1]）  
float4 packedNormal = tex2D(\_NormalMap, IN.uv\_MainTex);  
//要将颜色空间里的法线[0,1],转换至真正3D空间里的法线范围[-1,1]  
//注意：范围基本都是从[0,1]转换至[-1,1].主要是图的通道与法线xyz的对应关系要根据法线贴图格式而定  
//UnpackNormal, UnityCG.cginc里的函数  
//o.Normal = UnpackNormal(packedNormal);  
//expand,标准法线解压函数  
o.Normal = expand(packedNormal.xyz);  
}  
ENDCG  
}  
}