法线贴图的计算方式

WELCOME
TO THE
UNITY
SPECIALTY COURSE
STUDY

知识回顾

唐老狮系列教程-法线贴图的计算方式

知识回顾

1. 转置矩阵
2. 正交矩阵
3. 基础变换矩阵的构成规则
4. 坐标空间的变换规则

WELCOME
TO THE
UNITY
SPECIALTY COURSE
STUDY

唐老狮系列教程-法线贴图的计算方式

转置矩阵

转置矩阵是**将原始矩阵的行和列互换**得到的新矩阵

假设矩阵为 M，那 M 的转置矩阵一般写为 M^T

$$\begin{bmatrix}5 & 6 & 9 & 7\\6 & 0 & 1 & 2\end{bmatrix}^T = \begin{bmatrix}5 & 6\\6 & 0\\9 & 1\\7 & 2\end{bmatrix}$$

$$\begin{bmatrix}1 & 2\\5 & 6\\9 & 8\end{bmatrix}^T = \begin{bmatrix}1 & 5 & 9\\2 & 6 & 8\end{bmatrix}$$

$$\begin{bmatrix}x & y & z\end{bmatrix}^T = \begin{bmatrix}x\\y\\z\end{bmatrix}$$

$$\begin{bmatrix}x\\y\\z\end{bmatrix}^T = \begin{bmatrix}x & y & z\end{bmatrix}$$

正交矩阵

正交矩阵是一种特殊的方阵，正交的意思是垂直

它的特点是：一个方阵和它的转置矩阵相乘为单位矩阵，那么它就是正交矩阵

$$MM^T = M^TM = I$$

正交矩阵的这一性质，再根据之前学习的逆矩阵的一个重要性质

$$MM^{-1} = M^{-1}M = I$$

如果一个矩阵是正交的，那么它的逆矩阵等于其转置矩阵

$$\boldsymbol{M^T = M^{-1}}$$

基础变换矩阵的构成规则

4x4矩阵的基本构成规则为：

$$\begin{bmatrix} M11 & M12 & M13 & t1 \\ M21 & M22 & M23 & t2 \\ M31 & M32 & M33 & t3 \\ 0 & 0 & 0 & 1 \end{bmatrix} => \begin{bmatrix} \boxed{\begin{matrix} M11 & M12 & M13 \\ M21 & M22 & M23 \\ M31 & M32 & M33 \end{matrix}} & \boxed{\begin{matrix} t1 \\ t2 \\ t3 \end{matrix}} \\ \boxed{\begin{matrix} 0 & 0 & 0 \end{matrix}} & \boxed{1} \end{bmatrix} => \begin{bmatrix} M^{3x3} & t^{3x1} \\ 0^{1x3} & 1 \end{bmatrix}$$

矩阵的 M^{3x3} 部分用于表示旋转和缩放变换

矩阵的 t^{3x1} 部分用于表示平移

矩阵的 0^{1x3} 部分始终为零矩阵

矩阵的 右下角元素 始终为 1

如果变换矩阵用来变换矢量直接用

M^{3x3} 部分即可，因为矢量不受平移影响

WELCOME
TO THE
UNITY
SPECIALTY COURSE
STUDY

唐老狮系列教程-法线贴图的计算方式

坐标空间的变换规则

子到父的变换矩阵： $M_{s-f} = \begin{bmatrix} | & | & | & | \\ X_s & Y_s & Z_s & O_s \\ | & | & | & | \\ 0 & 0 & 0 & 1 \end{bmatrix}$

父到子的变换矩阵： $M_{f\text{-}s} = M_{s-f}^{-1}$

如果子到父的变换矩阵是一个正交矩阵

那么父到子的变换矩阵就是其的转置矩阵

因为

如果一个矩阵是正交的，那么它的逆矩阵等于其转置矩阵

唐老狮系列教程-法线贴图的计算方式

主要讲解内容

唐老狮系列教程-法线贴图的计算方式

主要讲解内容

1.两种主流计算方式

2.在切线空间下计算

3.在世界空间下计算

4.关键知识点补充

WELCOME
TO THE
UNITY
SPECIALTY COURSE
STUDY

两种主流计算方式

唐老狮系列教程-法线贴图的计算方式

两种主流计算方式

通过上节课的学习我们知道，想要实现凹凸效果，主要就是使用基于切线空间的法线贴图中的法线信息参与到光照计算中。

而在计算光照模型时，通常会有两种选择：

1.在切线空间下进行光照计算，需要把光照方向、视角方向变换到切线空间下参与计算

2.在世界空间下进行光照计算，需要把法线方向变换到世界空间下参与计算

WELCOME
TO THE
UNITY
SPECIALTY COURSE
STUDY

唐老狮系列教程-法线贴图的计算方式

各自的优缺点——效率

在切线空间中计算，效率更高，因为可以在顶点着色器中就完成对光照、视角方向的矩阵变换，计算量相对较小。

(矩阵变换在顶点着色器中计算)

在世界空间中计算，效率较低，由于需要对法线贴图进行采样，所以变换过程必须在片元着色器中实现，我们需要在片元着色器中对法线进行矩阵变换。

(矩阵变换在片元着色器中计算)

各自的优缺点——全局效果

在切线空间中计算，对全局效果的表现可能会不够准确

在处理一些列如镜面反射、环境映射效果时表现效果可能不够准确

在世界空间中计算，对全局效果的表现更准确

可以更容易的应用于全局效果的计算

两种主流计算方式

因此我们在选择使用哪种计算方式时

主要考虑

若没有全局效果要求，我们优先使用在切线空间下进行光照计算，因为它效率较高

反之，我们选择在世界空间下计算

具体使用哪种还是以项目的实际情况决定

我们接下来就来了解下两种计算方式中的关键点，为我们之后编写Shader做准备

唐老狮系列教程-法线贴图的计算方式

在切线空间下计算

唐老狮系列教程-法线贴图的计算方式

在切线空间下计算

在切线空间下进行光照计算，需要把光照方向、视角方向变换到切线空间下参与计算

关键点：

计算模型空间到切线空间的变换矩阵

变换矩阵为子到父的 $\begin{bmatrix} | & | & | & | \\ X_s & Y_s & Z_s & O_s \\ | & | & | & | \\ 0 & 0 & 0 & 1 \end{bmatrix}$ 逆矩阵

由于我们主要用变换矩阵来进行矢量的变换而非点的变换，因此可以变为3x3矩阵 $\begin{bmatrix} | & | & | \\ X_s & Y_s & Z_s \\ | & | & | \end{bmatrix}$

而x、y、z轴分别为切线空间中顶点的切线、副切线、法线

已知切线、法线（从模型数据中可以获取），副切线为切线、法线的叉乘结果

而3个轴为相互垂直的单位向量，因此可以推出 $\begin{bmatrix} | & | & | \\ X_s & Y_s & Z_s \\ | & | & | \end{bmatrix}$ 是正交矩阵

因此该变换矩阵的逆矩阵为其转置矩阵 $\begin{bmatrix} - & X_s & - \\ - & Y_s & - \\ - & Z_s & - \end{bmatrix}$ 它就是模型空间到切线空间的变换矩阵

WELCOME
TO THE
UNITY
SPECIALTY COURSE
STUDY

唐老狮系列教程-法线贴图的计算方式

在切线空间下计算

其它用到的核心知识

内置函数：

得到模型空间光的方向：ObjSpaceLightDir(模型空间顶点坐标)

得到模型空间视角方向：ObjSpaceViewDir(模型空间顶点坐标)

得到光方向和视角方向相对于模型空间的数据表达后

再与模型空间到切线空间的变换矩阵进行运算

即可将他们转换到切线空间下参与后续计算

唐老狮系列教程-法线贴图的计算方式

在世界空间下计算

唐老狮系列教程-法线贴图的计算方式

在世界空间下计算

在世界空间下进行光照计算，需要把法线方向变换到世界空间下参与计算

关键点：

计算切线空间到世界空间的变换矩阵

变换矩阵为子到父的变换 $\begin{matrix} | & | & | & | \\ X_s & Y_s & Z_s & O_s \\ | & | & | & | \\ 0 & 0 & 0 & 1 \end{matrix}$

由于我们主要用变换矩阵来进行矢量的变换而非点的变换，因此可以变为3x3矩阵 $\begin{matrix} | & | & | \\ X_s & Y_s & Z_s \\ | & | & | \end{matrix}$

而x、y、z轴分别为切线空间中顶点的切线、副切线、法线

我们只需要得到3个轴相对于世界空间的向量表达，即可得到该变换矩阵

唐老狮系列教程-法线贴图的计算方式

在世界空间下计算

法线从模型空间到世界空间：UnityObjectToWorldNormal(模型空间法线数据)

切线从模型空间到世界空间： UnityObjectToWorldDir(模型空间切线数据)

世界空间的副切线：用上面计算的结果叉乘即可

由这三个向量组成最终的切线空间到世界的空间的变换矩阵即可

$$\begin{array}{ccc} | & | & | \\ X_s & Y_s & Z_s \\ | & | & | \end{array}$$

WELCOME
TO THE
UNITY
SPECIALTY COURSE
STUDY

唐老狮系列教程-法线贴图的计算方式

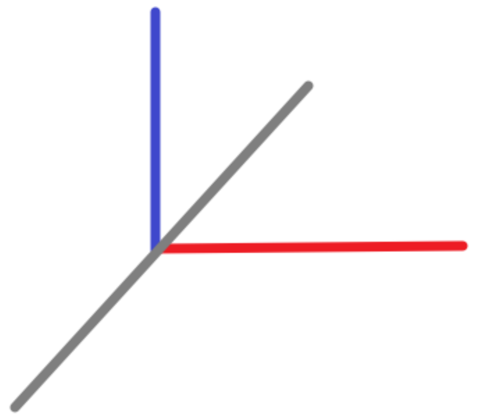
关键知识点补充

WELCOME
TO THE
UNITY
SPECIALTY COURSE
STUDY

唐老狮系列教程-法线贴图的计算方式

关键知识点补充

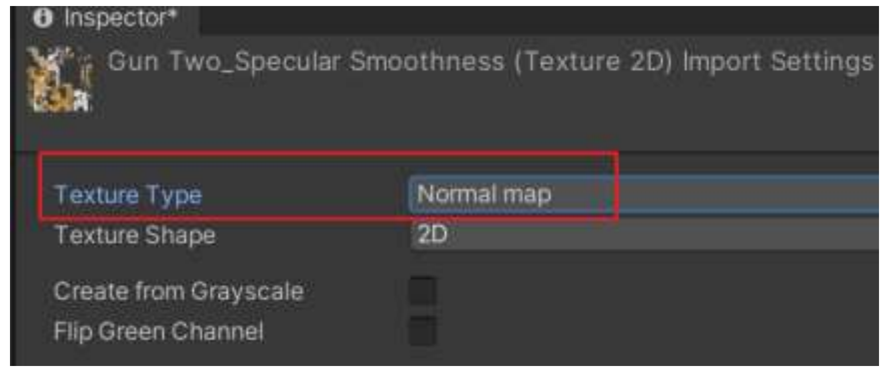
1.模型空间下的切线数据



模型数据中的切线数据为float4类型的，其中的w表示副切线的方向

用法线和切线叉乘得到的副切线方向可能有两个，用切线数据中的w与之相乘确定副切线方向

2.Unity当中的法线纹理类型



当我们把纹理类型设置为Normal map(法线贴图)时，我们可以使用Unity提供的内置函数UnpackNormal来得到正确的法线方向。该函数内部不仅可以进行**法线分量 = 像素分量 * 2 – 1**的逆运算，还会进行解压运算（Unity会根据不同平台对法线纹理进行压缩）

WELCOME
TO THE
UNITY
SPECIALTY COURSE
STUDY

唐老狮系列教程-法线贴图的计算方式

关键知识点补充

3. 法线纹理属性命名一般为_BumpMap (凸块贴图)

我们还会声明一个名为_BumpScale (凸块缩放) 的float属性

它主要用于控制凹凸程度

当它为0时，表示没有法线效果，法线的影响会被消除

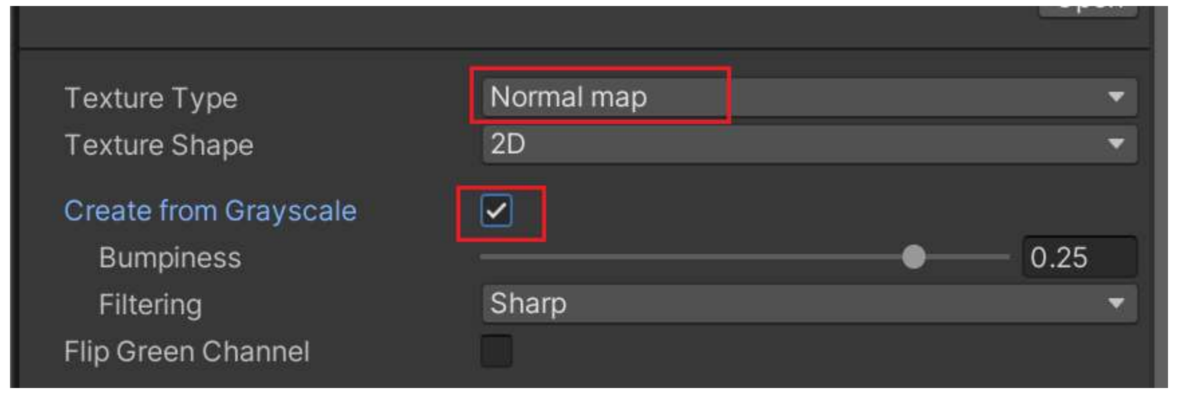
当它为1时，表示使用法线贴图中的原始法线信息，没有缩放

我们可以根据实际需求调整它的值，来达到视觉上令人满意的效果

唐老狮系列教程-法线贴图的计算方式

关键知识点补充

4. 如果使用的凹凸纹理不是法线纹理，而是高度纹理，我们需要进行如下设置



图片类型设置为Normal map（法线贴图）

勾选 Create from Grayscale（从灰度创建）

这样我们就可以把高度纹理当成切线空间下的法线纹理处理了

多出的Bumpiness（颠簸值）控制凹凸程度

Filtering（过滤模式）决定计算凹凸程度的算法

Sharp：滤波生成法线

Smooth：平滑的生成法线

总结

唐老狮系列教程-法线贴图的计算方式

主要讲解内容

1.两种主流计算方式

切线空间下计算(效率高，全局性差)和世界空间下计算（效率低，全局性好)

2.在切线空间下计算

主要是计算模型空间到切线空间的变换矩阵 $\begin{array}{ccc} - & Xs & - \\ - & Ys & - \\ - & Zs & - \end{array}$

3.在世界空间下计算

主要是计算切线空间到世界空间的变换矩阵 $\begin{array}{ccc} | & | & | \\ Xs & Ys & Zs \\ | & | & | \end{array}$

4.关键知识点补充

切线数据为float4、 UnpackNormal、 _BumpScale 、高度纹理也能用

WELCOME
TO THE
UNITY
SPECIALTY COURSE
STUDY

Thank

谢谢您的聆听

WELCOME
TO THE
UNITY
SPECIALTY COURSE
STUDY