

青春都一饷，忍把浮名，换了代码轻狂。

关注DirectX

随笔 - 198, 文章 - 65, 评论 - 1663, 引用 - 0

## 导航

博客园  
首页  
新随笔  
联系  
订阅  
管理

< 2012年7月 >						
日	一	二	三	四	五	六
24	25	26	27	28	29	30
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

## 公告

昵称：zdd  
园龄：7年1个月  
荣誉：推荐博客  
粉丝：687  
关注：20  
+加关注

## 搜索

找找看

## 常用链接

我的随笔  
我的评论  
我的参与  
最新评论  
我的标签

## 随笔分类

Android(12)  
C/C++(19)  
D3D10  
D3D11(1)  
D3D9(4)  
Demo(3)  
Direct2D(14)  
DirectInput(1)  
DirectWrite(3)  
DirectX(58)  
DXGI  
HLSL  
IOS(2)  
Math(16)  
OpenGL(1)  
Perl(1)  
PS(1)  
Shader(2)  
Swift  
Unity3D  
Vim  
XAudio2  
编程语言(9)  
代码片段(19)  
数据结构与算法(26)  
图形学(10)  
移动开发(11)  
杂(24)

## 随笔档案

2016年6月 (1)  
2016年1月 (4)  
2015年12月 (1)  
2015年11月 (1)  
2015年10月 (2)  
2015年8月 (5)  
2015年7月 (1)  
2014年11月 (2)  
2014年6月 (2)  
2014年5月 (1)  
2013年9月 (1)  
2013年7月 (1)  
2013年6月 (1)

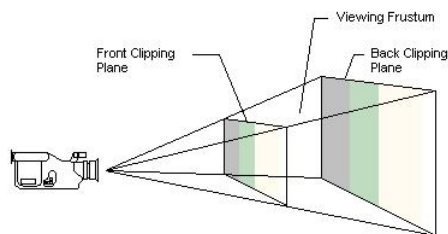
## 透视投影详解

### 概述

投影变换完成的是如何将三维模型显示到二维视口上，这是一个三维到二维的过程。你可以将投影变换看作是调整照相机的焦距，它模拟了为照相机选择镜头的过程。投影变换是所有变换中最复杂的一个。

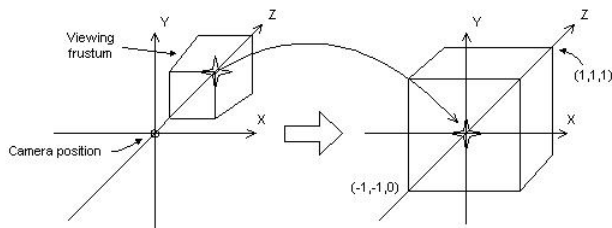
### 视锥体

视锥体是一个三维体，他的位置和摄像机相关，视锥体的形状决定了模型如何从camera space投影到屏幕上。最常见的投影类型-透视投影，使得离摄像机近的物体投影后较大，而离摄像机较远的物体投影后较小。透视投影使用棱锥作为视锥体，摄像机位于棱锥的椎顶。该棱锥被前后两个平面截断，形成一个棱台，叫做View Frustum，只有位于Frustum内部的模型才是可见的。



### 透视投影的目的

透视投影的目的就是将上面的棱台转换为一个立方体（cuboid），转换后，棱台的前剪裁平面的右上角点变为立方体的前平面的中心（下图中弧线所示）。由图可知，这个变换的过程是将棱台较小的部分放大，较大的部分缩小，以形成最终的立方体。这就是投影变换会产生近大远小的效果的原因。变换后的x坐标范围是[-1, 1]，y坐标范围是[-1, 1]，z坐标范围是[0, 1]（OpenGL略有不同，z值范围是[-1, 1]）。



### 透视投影矩阵推导

下面来推导一下透视投影矩阵，这样我们就可以自己设置投影矩阵了，就可以模拟神奇的D3DXMatrixPerspectiveLH函数的功能了。那么透视投影到底做了什么工作呢？这一部分算是个难点，无论是DX SDK的帮助文档，还是大多数图形学书籍，对此都是一带而过，很少有详细讨论的，早期的DX SDK文档还讨论的稍微多一些，而新近的文档则完全取消了投影矩阵的推导过程。

我们可以将整个投影过程分为两个部分，第一部分是Frustum内一点投影到近剪裁平面的过程，第二部分是由近剪裁平面缩放的过程。假设Frustum内一点P(x, y, z)在近剪裁平面上的投影是P'(x', y', z')，而P'经过缩放后的最终坐标设为P''(x'', y'', z'')。假设所求的投影矩阵为M，那么根据矩阵乘法可知，如下等式成立。

$PM = P''$ ，即

2013年4月 (2)  
 2013年3月 (3)  
 2013年2月 (3)  
 2012年11月 (2)  
 2012年10月 (2)  
 2012年9月 (5)  
 2012年8月 (6)  
 2012年7月 (5)  
 2012年6月 (2)  
 2012年5月 (2)  
 2012年4月 (1)  
 2011年9月 (2)  
 2011年7月 (3)  
 2011年6月 (2)  
 2011年5月 (4)  
 2011年4月 (3)  
 2011年3月 (3)  
 2011年2月 (4)  
 2010年12月 (3)  
 2010年11月 (3)  
 2010年9月 (1)  
 2010年8月 (11)  
 2010年7月 (17)  
 2010年6月 (9)  
 2010年5月 (12)  
 2010年4月 (2)  
 2010年3月 (16)  
 2010年2月 (3)  
 2010年1月 (2)  
 2009年12月 (2)  
 2009年11月 (9)  
 2009年10月 (11)  
 2009年9月 (3)  
 2009年8月 (1)  
 2009年7月 (4)  
 2009年6月 (10)  
 2009年5月 (2)

#### Game Engine

DX11 Tutorials  
 Irrlicht  
 MathWords  
 Ogre

#### OpenGL

#### 缤纷世界

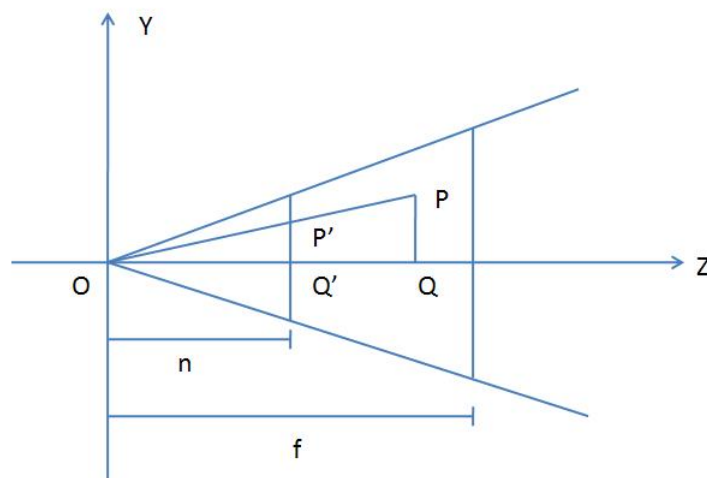
3D Controls  
 3D Fractal  
 chaos files  
 ChaosinChinese  
 Cloth Simulation  
 DirectX Developer Center  
 DirectX document online  
 DX tutorials  
 EuclideanSpace  
 Fractal Video  
 geometrictools  
 Google C++ Style  
 HardCode  
 HyperGraph  
 In Framez  
 LatexEditor  
 Mame  
 Mandelbrot Set  
 OpenGL official page  
 OpenGL Tutorials  
 pouet  
 Ray tracing  
 Rthdribl  
 TechInterview  
 toymaker  
 W3SCHOOL  
 XNA/DirectX Forum  
 X-Zone  
 云风的Blog

#### 其他

Aogo汇编小站  
 Channel 9  
 Code all in one  
 Emath  
 Math Circle  
 Math Game  
 Microsoft At Home  
 Microsoft At Work  
 Windows forum  
 中国DOS联盟

$$[x, y, z, 1] \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} = [x', y', z', w]$$

先看第一部分，为了简化问题，我们考虑YOZ平面上的投影情况，见下图。设P(x, y, z)是Frustum内一点，它在近剪裁平面上的投影是P'(x', y', z')。(注意：D3D以近剪裁平面作为投影平面)，设视锥体在Y方向的夹角为 $\theta$ 。



由上图可知，三角形OP'Q'与三角形OPQ相似，于是有如下等式成立。

$$\frac{P'Q'}{OQ'} = \frac{PQ}{OQ}, \text{ 即 } \frac{P_{y'}}{n} = \frac{P_y}{P_z}, \text{ 即 } y' = \frac{nP_y}{P_z}$$

$$\text{同理, } x' = \frac{nP_x}{P_z}, z' = n$$

$$\begin{cases} x' = \frac{nP_x}{P_z} \\ y' = \frac{nP_y}{P_z} \\ z' = n \end{cases}$$

在看第二部分，将P'缩放的过程，假设投影平面的高度为H，由于转换后cuboid的高度为2。所以有

$$\frac{y'}{y''} = \frac{H}{2}, \text{ 即 } y'' = \frac{y'}{\frac{H}{2}} = \frac{\frac{nP_y}{P_z}}{\frac{H}{2}}$$

$$\text{因为 } \frac{n}{\frac{H}{2}} = \cot \theta, \text{ 所以 } y'' = \frac{P_y}{P_z} \cot \theta$$

又因为投影平面的纵横比为Aspect，所以

#### 友情链接

LittleStart  
WW老弟

#### 积分与排名

积分 - 202944  
排名 - 781

#### 最新评论

1. Re:Direct2D教程 ( 十一 ) 几何变换  
你好, 请问怎么样可以把图片 ( 或者画刷 ) 弄成镜像对称的? 旋转180度就上下颠倒了。  
--kina1234
2. Re:字符串面试题 ( 一 ) 字符串逆序  
**shoucangshou**收藏  
--cnb\_yangwei
3. Re:GLFW初体验  
楼主, 我想咨询下—glfw 上面的 glDrawpixels 函数, 在mac上渲染出来只有1/4的内容呢我用glfw创建了一个200 \* 200的窗口, 然后声明了一个 uchar[200 \* 20.....  
--程序员小U
4. Re:DirectX Effects初探 初学龙书6.8 effects。  
--hplucky
5. Re:C++ 初始化列表  
31楼说的对。  
我就觉得这里有问题, 自己也试了, 确实不行。看评论找了半天, 看到有一样的了。  
楼主抓紧重新编辑下啊。  
--peiruibo

#### 阅读排行榜

1. 点到平面的距离公式 (43332)
2. 算法-求二进制数中1的个数 (43075)
3. C++ 初始化列表 (42090)
4. 字符串面试题 ( 一 ) 字符串逆序 (39842)
5. 关于数组的几道面试题 (36980)

#### 评论排行榜

1. 程序员, 请昂起你高贵的头! (364)
2. 关于数组的几道面试题 (96)
3. 几何变换详解 (76)
4. 判断点是否在三角形内 (62)
5. 使用DirectX截屏 (55)

#### 推荐排行榜

1. 程序员, 请昂起你高贵的头! (176)
2. 算法-求二进制数中1的个数 (28)
3. C++ 初始化列表 (25)
4. 字符串面试题 ( 一 ) 字符串逆序 (20)
5. 关于数组的几道面试题 (17)

$$x'' = \frac{P_x}{P_z} \cot \theta \frac{1}{Aspect}$$

最后看 $z''$ , 当Frustum内的点投影到近剪裁平面的时候, 实际上这个 $z'$ 值已经没有意义了, 因为所有位于近剪裁平面上的点, 其 $z'$ 值都是 $n$ , 看起来我们甚至可以抛弃这个 $z'$ 值, 可以么? 当然不行! 别忘了后面还有深度测试呢。由第一幅图可知, 所有位于线段 $p'p$ 上的点, 最终都会投影到 $p'$ 点, 那么如果这条线段上真的有多个点, 如何确定最终保留哪一个呢? 当然是离观察这最近的这个了, 也就是深度值 ( $z$ 值) 最小的。所以 $z'$ 坐标可以直接保存 $p$ 点的 $z$ 值。因为在光栅化之前, 我们需要对 $z$ 坐标的倒数进行插值 (原因请参见 Mathematics for 3D Game Programming and Computer Graphics 3rd section 5.4), 所以可以将 $z''$ 写成 $z$ 的一次表达式形式, 如下

$$z'' = a \frac{1}{P_z} + b$$

在映射前,  $z$ 的范围是 $[n, f]$ , 这里 $n$ 和 $f$ 分别是远近两个剪裁平面到原点的距离, 在映射后,  $z''$ 的范围是 $[0, 1]$ , 将数据代入上面的一次式, 可得下面的方程组

$$\begin{cases} 0 = a \frac{1}{n} + b \\ 1 = a \frac{1}{f} + b \end{cases}$$

解这个方程组得到

$$a = \frac{fn}{n-f}, b = \frac{f}{f-n}$$

所以

$$z'' = \frac{fn}{n-f} \frac{1}{P_z} + \frac{f}{f-n}$$

整理一下得

$$\begin{cases} x'' = \frac{P_x}{P_z} \cot \theta \frac{1}{Aspect} \\ y'' = \frac{P_y}{P_z} \cot \theta \\ z'' = \frac{fn}{n-f} \frac{1}{P_z} + \frac{f}{f-n} \end{cases}$$

将 $x'', y'', z''$ 代入最开始的矩阵乘法等式中得

$$\begin{bmatrix} x, y, z, 1 \end{bmatrix} \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} = \left[ \frac{P_x}{P_z} \cot \theta \frac{1}{Aspect}, \frac{P_y}{P_z} \cot \theta, \frac{fn}{n-f} \frac{1}{P_z} + \frac{f}{f-n}, w \right]$$

由上式可见,  $x'', y'', z''$ 都除以了 $P_z$ , 于是我们将他们再乘以 $P_z$  (这并不该变齐次坐标的大小), 得到如下等式。

$$\begin{bmatrix} x, y, z, 1 \end{bmatrix} \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} P_x \cot \theta \frac{1}{Aspect}, P_y \cot \theta, \frac{fn}{n-f} + \frac{P_z f}{f-n}, wP_z \end{bmatrix}$$

注意这里，x即Px，y即Py，z即Pz，解矩阵的每一列得到

$$\begin{cases} a_{00} = \frac{\cot \theta}{Aspect} \\ a_{11} = \cot \theta \\ a_{22} = \frac{f}{f-n} \\ a_{32} = \frac{fn}{n-f} \\ a_{23} = 1 (\text{令 } w = 1) \end{cases}$$

于是所求矩阵为

$$M = \begin{bmatrix} \frac{\cot \theta}{Aspect} & 0 & 0 & 0 \\ 0 & \cot \theta & 0 & 0 \\ 0 & 0 & \frac{f}{f-n} & 1 \\ 0 & 0 & \frac{fn}{n-f} & 0 \end{bmatrix}$$

代码

一般来说，在程序中我们通常给定四个参数来求透视投影矩阵，分别是y方向的视角，纵横比，近剪裁平面到原点的距离及远剪裁平面到原点的距离，通过这四个参数即可求出上面的矩阵，代码如下。

```
D3DXMATRIX BuildProjectionMatrix(float fov, float aspect, float zn, float zf)
{
    D3DXMATRIX proj;
    ZeroMemory(&proj, sizeof(proj));

    proj.m[0][0] = 1 / (tan(fov * 0.5f) * aspect);
    proj.m[1][1] = 1 / tan(fov * 0.5f);
    proj.m[2][2] = zf / (zf - zn);
    proj.m[2][3] = 1.0f;
    proj.m[3][2] = (zn * zf) / (zn - zf);

    return proj;
}
```

矩阵求解完毕，现在可以用如下代码试试效果，这和使用D3D函数D3DXMatrixPerspectiveFovLH所得效果是一致的。

```
D3DXMATRIX proj = BuildProjectionMatrix(D3DX_PI / 4, 1.0f, 1.0f, 1000);
g_pd3dDevice->SetTransform(D3DTS_PROJECTION, &proj);
```

Happy Coding!!!

作者：zdd

出处：<http://www.cnblogs.com/graphics/>

本文版权归作者和博客园共有，欢迎转载，但未经作者同意必须保留此段声明，且在文章页面明显位置给出原文连接，否则保留追究法律责任的权利。

分类: [DirectX](#)

好文要顶

关注我

收藏该文







zdd  
关注 - 20  
粉丝 - 687

8

0

荣誉: [推荐博客](#)  
[+加关注](#)

(请您对文章做出评价)

« 上一篇: [DirectX Triangle Strips](#)  
» 下一篇: [矩阵-DirectX与OpenGL的不同](#)

posted on 2012-07-25 08:57 zdd 阅读(16827) 评论(40) 编辑 收藏

## 评论

### #1楼

楼主把这个研究得好透彻啊.....我看了这个受益匪浅啊.....多谢楼主

支持(0) 反对(0)

2012-11-02 11:00 | adfan

### #2楼[楼主]

@ adfan

引用

楼主把这个研究得好透彻啊.....我看了这个受益匪浅啊.....多谢楼主

如果能给你帮助,就算没白写呀,有没有觉得哪里写得充分,或者哪里有需要改进的地方?  
说实话,遇到一个认真看完的人很不容易,所以,我非常希望你多多提宝贵意见。

支持(0) 反对(0)

2012-11-02 12:07 | zdd

### #3楼

"由上式可见,  $x'', y'', z''$  都除以了  $P_z$ , 于是我们将他们再乘以  $P_z$  (这并不该变齐次坐标的大小)"-----为什么可以只在等式一边成上  $P_z$

支持(0) 反对(0)

2012-12-10 13:32 | numlock

### #4楼[楼主]

@ numlock

引用

"由上式可见,  $x'', y'', z''$  都除以了  $P_z$ , 于是我们将他们再乘以  $P_z$  (这并不该变齐次坐标的大小)"-----为什么可以只在等式一边成上  $P_z$

感谢您提出的问题,我的理解是等式的右边是一个齐次坐标,而齐次坐标乘以一个数后得到的新坐标还表示原来的点,这里乘以  $P_z$  是为了方便计算。这篇文章也是我从网上许多资料搜集整理而成的,可能很多地方理解的还不够深刻,日后需要逐步完善。

支持(0) 反对(0)

2012-12-10 17:10 | zdd

### #5楼

这个很不错

支持(0) 反对(0)

2013-03-01 15:08 | 子夜东风

## #6楼[楼主]

@ 子夜东风

引用

这个很不错

谢谢支持！

支持(0) 反对(0)

2013-03-01 23:48 | zdd

## #7楼

大哥有点没看懂，你写的挺清晰的。可是我的数学底子差，大学根本就没怎么上过课，我想问问为什么 $z$ 要取它的倒数然后线性插值呢？？

支持(0) 反对(0)

2013-04-19 17:05 | 悠悠小法

## #8楼[楼主]

@ 悠悠小法

引用

大哥有点没看懂，你写的挺清晰的。可是我的数学底子差，大学根本就没怎么上过课，我想问问为什么 $z$ 要取它的倒数然后线性插值呢？？

问得好！看了半天我也没看懂，可能是我忘了，等我有空再去查查，谢谢你。

支持(0) 反对(0)

2013-04-19 19:02 | zdd

## #9楼

楼主，frustum中右上角坐标是 $(r, t, n, 0)$

变换后得到的齐次坐标是 $(1, 1, 0, 1)$ ，并没有在前平面的中心

话说D3D SDK中也说是在前平面中心，这是不是有问题

支持(0) 反对(0)

2013-05-27 14:50 | 子夜东风

## #10楼[楼主]

@ 子夜东风

引用

楼主，frustum中右上角坐标是 $(r, t, n, 0)$

变换后得到的齐次坐标是 $(1, 1, 0, 1)$ ，并没有在前平面的中心

话说D3D SDK中也说是在前平面中心，这是不是有问题

恩，我也发现这个问题了，很有可能是D3D SDK文档的错误，有待进一步研究。有进展会及时通知你。

支持(0) 反对(0)

2013-05-27 15:55 | zdd

## #11楼

楼主写的太好了，忍不住转载很多，要是当时我们做摄像机标定和校正的时候能搜索到您的博客估计会省下不少时间

支持(0) 反对(0)

2013-10-18 10:28 | magicrat

## #12楼[楼主]

@ magicrat

引用

楼主写的太好了，忍不住转载很多，要是当时我们做摄像机标定和校正的时候能搜索到您的博客估计会省下不少时间

感谢支持！欢迎转载。

支持(0) 反对(0)

2013-10-18 23:16 | zdd

#13楼

楼主写的真好 目前有点搞明白了=。=

支持(0) 反对(0)

2014-01-22 12:08 | 江河小虾

#14楼[楼主]

@ 江河小虾

引用

楼主写的真好 目前有点搞明白了=。=

多谢支持。

支持(0) 反对(0)

2014-01-22 17:01 | zdd

#15楼

看了好多关于透视投影变换的文章，感觉这个不错，还有图，

就是看到这一句的时候：“因为在光栅化之前，我们需要对z坐标的倒数进行插值（原因请参见Mathematics for 3D Game Programming and Computer Grahpics 3rd section 5.4）”

能不能把这个关于Z坐标的变换补充一下呀，我E文不好，怕是看不懂的，辛苦咯！

还有受你文章启发，透视投影变换可不可以按这样的步骤来：

1.首先直接投影-->这里的投影不是你文章里的棱锥体投影，而是所有点法线 式的投影，效果就是物体大小(x,y)没变距离(z)变了(压扁了)

2.再缩小，按最后的那个 (-1, 1)正立方体缩小

我不知道我说清楚了没！！

支持(0) 反对(0)

2014-01-28 17:41 | tcsj

#16楼

第一步也就是平行投影

支持(0) 反对(0)

2014-01-28 17:42 | tcsj

#17楼

@ tcsj

引用

看了好多关于透视投影变换的文章，感觉这个不错，还有图，

就是看到这一句的时候：“因为在光栅化之前，我们需要对z坐标的倒数进行插值（原因请参见Mathematics for 3D Game Programming and Computer Grahpics 3rd section 5.4）”

能不能把这个关于Z坐标的变换补充一下呀，我E文不好，怕是看不懂的，辛苦咯！

还有受你文章启发，透视投影变换可不可以按这样的步骤来：

1.首先直接投影-->这里的投影不是你文章里的棱锥体投影，而是所有点法线 式的投影，效果就是物体大小(x,y)没变距离...

在Mathematics for 3D Game Programming and Computer Graphics 3rd section 5.4中又提到过，但是我也没怎么看懂

支持(0) 反对(0)

2014-05-19 21:48 | NoahZuo

## #18楼

楼主，我看了后在转化到cuboid这个立方体有个疑问，计算x轴方向上的坐标，根据转换后的y来乘以纵横比，那么转换后的y最大值可以为1，如果纵横比是小于1的，那么计算出来的x轴方向上的值就会大于1，那么就没有在cuboid这个单位立方体里面，这个是什么原因呀？

支持(0) 反对(0)

2014-07-29 09:14 | 行2014

## #19楼

再次提出一个问题，在Mathematics for 3D Game Programming and Computer Graphics 3rd中提到视锥体中的z值和空间中的z值是相反的，换句话说应该是z值越大，离得越近.....在D3D中也是这样的设置

支持(0) 反对(0)

2014-07-29 13:40 | NoahZuo

## #20楼

楼主大哥，这个D3DXMATRIX结构体为什么还能这样用啊

```
D3DXMATRIX proj;
ZeroMemory(&proj, sizeof(proj));

proj.m[0][0] = 1 / (tan(fov * 0.5f) * aspect) ;
proj.m[1][1] = 1 / tan(fov * 0.5f) ;
怎么冒出一个m[][], 哪里定义了？
```

支持(0) 反对(0)

2014-08-01 18:46 | 大侠依旧笑红尘

## #21楼[楼主]

@ 大侠依旧笑红尘

引用

楼主大哥，这个D3DXMATRIX结构体为什么还能这样用啊

```
D3DXMATRIX proj;
ZeroMemory(&proj, sizeof(proj));

proj.m[0][0] = 1 / (tan(fov * 0.5f) * aspect) ;
proj.m[1][1] = 1 / tan(fov * 0.5f) ;
怎么冒出一个m[][], 哪里定义了？
```

是在D3DMATRIX里面定义的，而D3DXMATRIX继承自D3DMATRIX，详情看<http://msdn.microsoft.com/en-us/library/windows/desktop/bb172573%28v=vs.85%29.aspx>。

支持(0) 反对(0)

2014-08-04 10:22 | zdd

## #22楼

@ zdd

额 嘿嘿。。。发现了。。 A 4x4 matrix that contains methods and operator overloads. 后来去查了下。。以前没注意 哈哈。。新手。。Directx9.0以前学过 这个暑假一直在复习 Direct9.0 和 学习3D游戏编程。。。嘿嘿。。发现大神 以后不懂的请教您了~

支持(0) 反对(0)



2014-08-04 10:34 | 大侠依旧笑红尘

### #23楼[楼主]

@ 大侠依旧笑红尘

大神不敢当，多多交流。

支持(0) 反对(0)

2014-08-04 11:20 | zdd

### #24楼

好文章！正如博主所说的那样，许多资料都只是对其一笔带过啊。之前看《3D数学基础：图形与游戏开发》的时候，还纳闷它第15章裁剪空间那一节的矩阵是怎么来的。书中像DX SDK一样，直接给出了结果矩阵，这篇文章正好弥补了此书的不足！

支持(0) 反对(0)

2014-12-02 18:37 | 封剑隐居

### #25楼

还有一点比较疑惑。

文章中说到，设视锥体在Y方向的夹角为 $\Theta$ 。这样描述貌似不太对，我上面提到的那本书，它里面用视场角来描述其，也就是fov。

支持(0) 反对(0)

2014-12-02 19:07 | 封剑隐居

### #26楼

很不错，谢谢博主写的这么详细

支持(0) 反对(0)

2015-03-02 21:06 | chaoguo1234

### #27楼

大神的数学功力好强

支持(0) 反对(0)

2015-04-15 16:44 | 天天看世界

### #28楼

我真的看完了，大致看懂了这四个参数的由来。大致明白了透视的clip plane原来是这么回事。

支持(0) 反对(0)

2015-04-15 16:45 | 天天看世界

### #29楼

非常感谢大神的无私奉献

支持(0) 反对(0)

2015-04-15 16:45 | 天天看世界

### #30楼

非常感谢楼主，之前读过几篇别人关于透视投影变换的文章，有两个疑问：

- 1：关于z"的设法，以及这么设之后，岂不是也丢失了原始的z值吗？
- 2：经过最后这个矩阵的变换之后，将原来的p点变换到了正方体中，此时的z值和w值有什么用？

支持(0) 反对(0)

2015-04-17 11:18 | tanyongdahaoren

### #31楼

好博文，看了楼主的博文受益匪浅！！

支持(0) 反对(0)

2015-04-25 17:11 | WenMy

### #32楼

写的非常好,受益了

支持(0) 反对(0)

2015-07-29 21:53 | 睡梦无痕

### #33楼

写的不错

透视差值那个看以参考：<http://imgtec.eetrend.com/article/1940>

支持(0) 反对(0)

2015-08-25 19:01 | wubugui

### #34楼

我是特地注册帐号来感谢博主的

支持(0) 反对(0)

2015-10-09 17:03 | christoph

### #35楼[楼主]

@ wubugui

引用

写的不错

透视差值那个看以参考：<http://imgtec.eetrend.com/article/1940>

感谢！

支持(0) 反对(0)

2015-10-11 22:40 | zdd

### #36楼[楼主]

@ christoph

引用

我是特地注册帐号来感谢博主的

多谢！

支持(0) 反对(0)

2015-10-11 22:40 | zdd

### #37楼

为什么对平截头体转换的时候，近平面的右上角转到中心？？为什么不是近平面的中心转到中心？

支持(0) 反对(0)

2015-10-24 09:46 | prop\_jeff

### #38楼[楼主]

@ prop\_jeff

引用

为什么对平截头体转换的时候，近平面的右上角转到中心？？为什么不是近平面的中心转到中心？

这个地方好像是有误的，微软的DirectX文档里面是这么写的。正确性有待证实。

支持(0) 反对(0)

2015-10-26 10:07 | zdd

## #39楼

修改：谢谢，没事了.....我理解错了.....

-----

您好，我在调试DX的时候发现一个问题，对于近平面N为零的情况的处理。

调用这个函数：D3DXMatrixPerspectiveLH：

```
1 D3DXMatrixPerspectiveFovLH(&matProjection, D3DXToRadian(90), width / height, 0.0f, 100.0f);
2 D3DXMatrixPerspectiveFovLH(&matProjection, D3DXToRadian(45), width / height, 0.0f, 100.0f);
```

标准处理如文所述，或者是RLTB确定的矩形（SharpDX 的方法）。

SharpDX 用N和FOV确定视区域，然后用RLTB计算矩阵，在N=0时发生了W=0,H=0的情况，导致被零除得到含有NaN的矩阵。

FOV的方法计算结果不一致。

D3DX的结果：

```
[1.72751749 0 0 0
0 2.41421342 0 0
0 0 1 1
0 0 -0 1]
```

SharpDX的结果：

```
[NaN 0 0 0
0 NaN 0 0
NaN NaN 1 1
0 0 0 0]
```

本例的结果：

```
[0.7499999 0 0 0
0 1 0 0
0 0 1 1
0 0 0 0]
```

我很奇怪D3DX是怎么计算这个矩阵的，一直没想出来，望能为解惑。

支持(0) 反对(0)

2016-01-24 15:20 | Grid Science

## #40楼

博主对DX研究得很透啊

支持(0) 反对(0)

2016-04-12 12:46 | xiaoluo91

[刷新评论](#) [刷新页面](#) [返回顶部](#)

**注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。**

【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库  
【推荐】融云即时通讯云 - 豆果美食、Faceu等亿级APP都在用  
【推荐】报表开发别头大！类Excel 复杂报表开发实例，即学即用  
【推荐】福利Time，讯飞开放平台注册即送好礼！  
【推荐】阿里云万网域名：.xin .com将推出重磅优惠



# Meetup

野狗技术沙龙



## 最新IT新闻:

- Apple Watch运动表带展示方式获得设计专利
  - 3名宇航员在太空生活186天后安全返回地球
  - 吴文辉上演“王子复仇记”后，阅文集团为何还起内乱
  - 微软测试Windows 10新还原工具
  - 上线24天完成A轮融资 分答快速圈钱背后：盈利模式前景难料
- » 更多新闻...

JPush 极光推送

消息推送领导品牌全面升级



登录/注册

## 最新知识库文章:

- 学习如何学习
  - 一个32岁入门的70后程序员给我的启示
  - 技术发展瓶颈的突破
  - 高效编程之道：好好休息
  - 快速学习者的高效学习策略
- » 更多知识库文章...

Powered by:  
博客园  
Copyright © zdd