目录

[左手系与右手系 1](#_Toc24318)

[向量 2](#_Toc24665)

[向量点积(dot product) 2](#_Toc31293)

[向量叉积(cross product) 2](#_Toc27621)

[a x b = |a|b|sinθ 3](#_Toc19815)

[矩阵 4](#_Toc26661)

[矩阵\*标量 4](#_Toc29194)

[矩阵\*矩阵 4](#_Toc15915)

[单位矩阵(identiti matrix) 5](#_Toc29642)

[转置矩阵(transposed matrix) 5](#_Toc16022)

[逆矩阵(inverse matrix) 5](#_Toc11817)

[正交矩阵(orthogonal matrix) 6](#_Toc30394)

[行矩阵与列矩阵 6](#_Toc414)

[线性变换(linear transform) 7](#_Toc29657)

[仿射变换(affine transform) 7](#_Toc25186)

[常用变换矩阵性质 8](#_Toc3396)

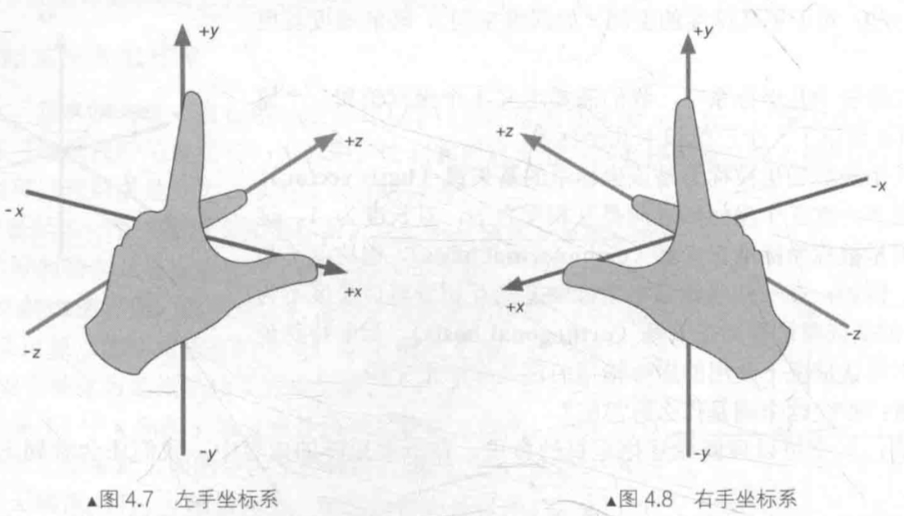
[后面的内容用到时再补充，因为之前多次复习并多次遗忘过，本次不做记录 8](#_Toc5784)

[4.9.3关于屏幕映射的计算 8](#_Toc21597)

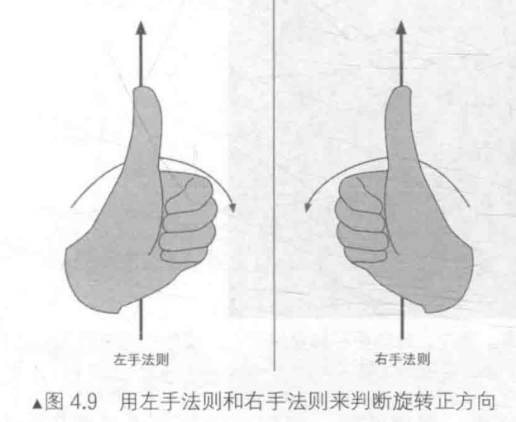
3d数学

只用来记录一些特殊3d数学

# 左手系与右手系



左手系，右手系的旋转正方向



gl是右手系，dx是左

unity使用的是左手系，但是观察空间(相机空间)使用的是右手系，因为相机前方是z轴负方向。

# 向量

## 向量点积(dot product)

也叫内积(inner product)

a · b = (ax,ay,az) · (bx,by,bz) = (axbx+ayby+azbz)

a · b = |a|b|cosθ 如果ab都是单位向量，结果就是cosθ

几何意义a · b如果a是单位矢量，那么就代表b在a上的投影（projection），结果是标量

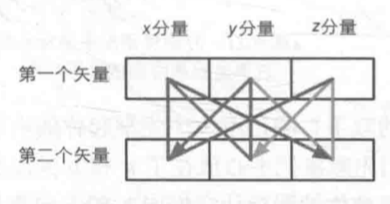
交换律 a· b = b· a

结合律 a· (b+c)=a· b+a· c

## 向量叉积(cross product)

也叫外积(outer product)

182C30EB-694A-4799-A012-4F4825819E5C



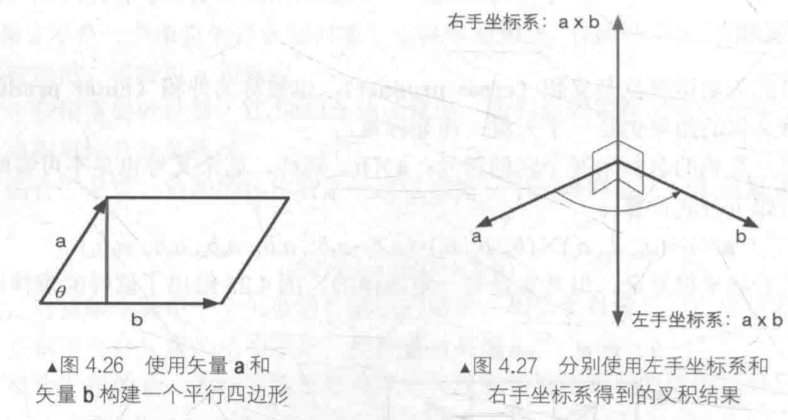
a x b = |a|b|sinθ

几何意义，，A是平行四边形面积

014AAE70-B5D8-4469-A382-8B86CCCA3907

当a，b平行a x b = 0这里可以认为平行四边形面积0，结果里的0是0向量，不是标量0

得到的结果是垂直与a和b的向量，具体方向与坐标系有关



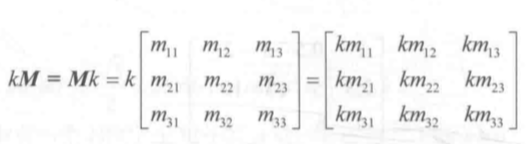
计算的值是平行四边形面积，方向如上图有图所示

交换律a x b ≠ b x a

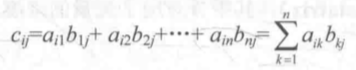
结合律 (a x b) x c ≠ a x (b x c)

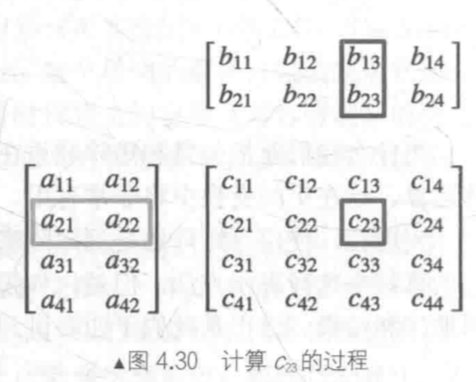
# 矩阵

## 矩阵\*标量



## 矩阵\*矩阵





0958E499-951D-4E51-81AE-BBFEE059CB3B

交换律AB ≠ BA

结合律 (AB)C = A(BC)

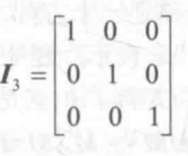
方块矩阵(square matrix),简称方阵，行列数相同的矩阵

对角元素(diagonal elements),行列号相同的元素

对角矩阵,除对角元素外所有元素为0

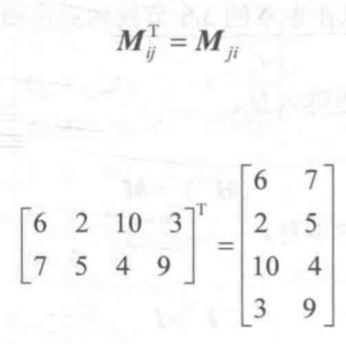
## 单位矩阵(identiti matrix)

对角元素都为1的对角矩阵



MI = IM =M 矩阵与单位矩阵相乘结果不变

## 转置矩阵(transposed matrix)



7D17A9C3-2883-4983-B2F7-697043D2215F

## 逆矩阵(inverse matrix)

不是所有矩阵都有逆矩阵

第一个前提是矩阵必须是方阵



3F5EC53B-1D8E-41AD-980B-610D1BD669B1

D2ADA9BB-7CF7-4ACF-B485-B2C6B3B1570C

EE6D646D-9CAE-46E0-8428-11347DFABF9D

F4552373-6222-4AE6-980F-EC6098DC7DF3

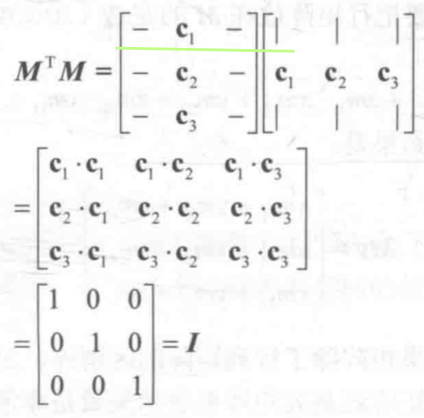
D48D9446-EAC6-48D2-9C5A-31C7AEAA5742

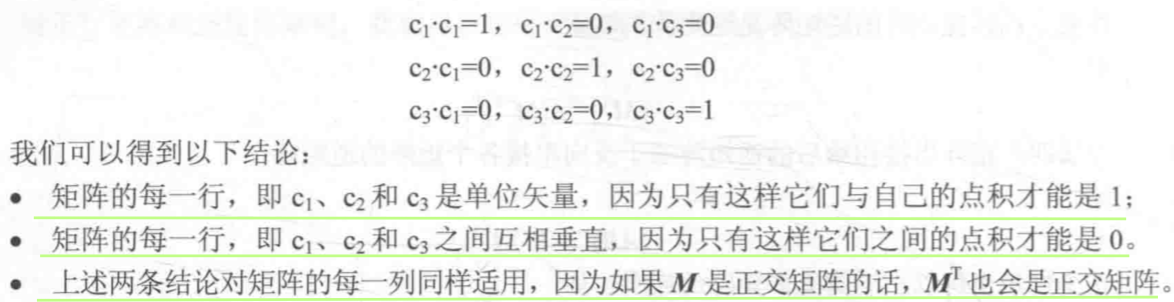
## 正交矩阵(orthogonal matrix)



B323AA69-4F61-4480-98F4-7BE24FC99D9B

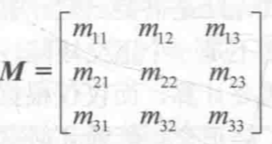
正交矩阵性质



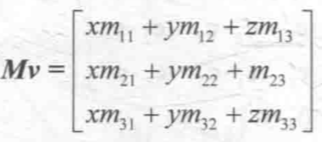


## 行矩阵与列矩阵

4F54F6F8-D2E1-4838-9F60-8C091B3A210B47281084-3EC1-43AD-9A05-A3F31EF8FFC5



5D900B36-2EB6-4591-8634-3CC49EFDE709



unity中使用的是列矩阵

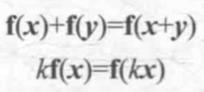


对应的行矩阵为

CBEBE06C-9259-4EAD-9835-E6C81B9189A1

## 线性变换(linear transform)

可以保留矢量加和标量乘的变化，具体就是满足如下公式



缩放，旋转，错切(shear),镜像(mirroring,也叫reflection)，正交投影(orthographic projection)等都是线性变换

考虑如下变换

F(x)=2x 是线性变换

F(x)=x+(1,2,3)不是线性变换

Eg:

x=(1,1,1)

F(x)+F(x) = (4,6,8)

F(x+x) =(3,4,5)

## 仿射变换(affine transform)

合并线性变换和平移变换的变换

## 常用变换矩阵性质



# 行矩阵与列矩阵

Unity shader用的是cg语法所以是行矩阵

Unity 脚本中的Matrix4x4 是列矩阵的形式，注意两者不同

# 后面的内容用到时再补充，因为之前多次复习并多次遗忘过，本次不做记录

旋转矩阵

缩放矩阵

平移矩阵

齐次矩阵

投影矩阵

模型空间

世界空间

观察空间

投影空间

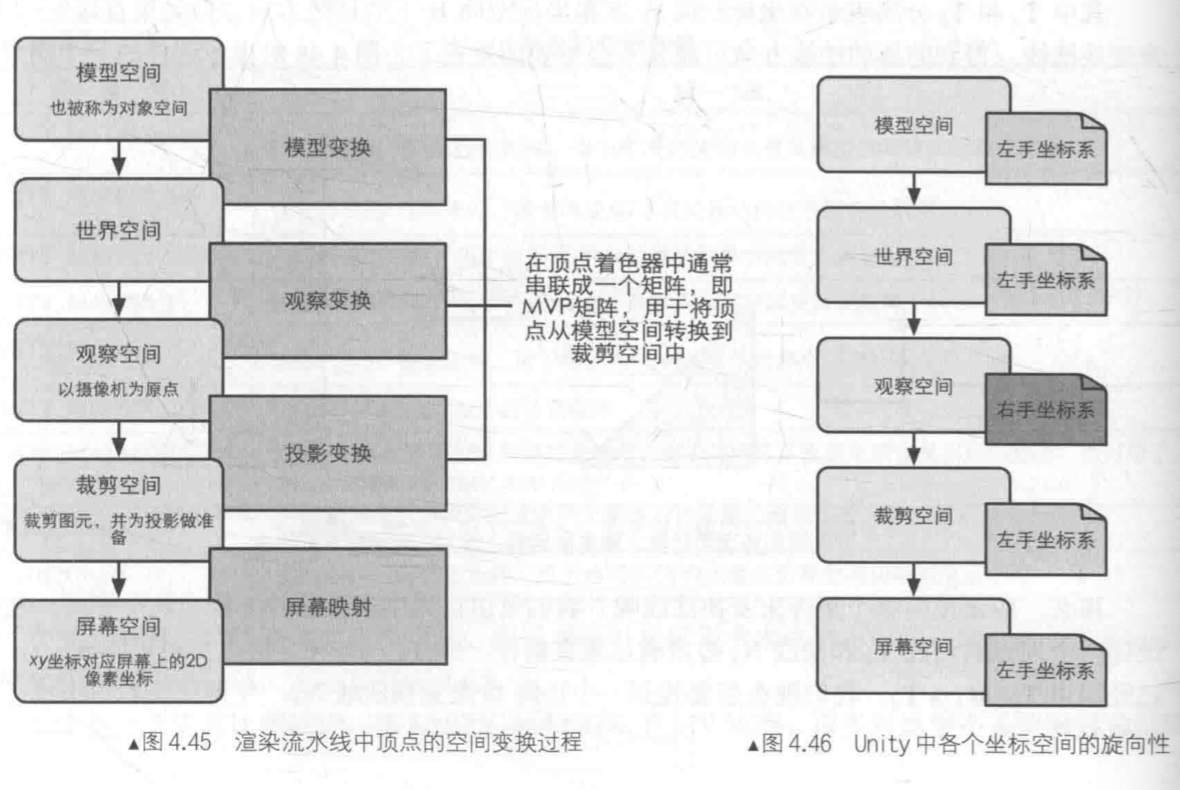
NDC

屏幕空间

切空间

法线变换

4.9.3关于屏幕映射的计算



# GL与dx不同的地方总结

Gl右手系

Dx左手系

Unity左手系,注意相机是右手系

屏幕

GL左上是0,0, 右下是1,1

DX左下是0,0, 右上是1,1

unity左下是0,0，右上是1,1

gl的ndc xyz分量范围都是-1,1

Dx的ndc z分量范围是0,1

Unity Shader中构建矩阵

float3x3 worldToTangent = float3x3(worldTangent, worldBinormal, worldNormal);

构建的是行矩阵

Unity 脚本Matrix4\*4进行填充时用的是列矩阵的形式