1. 各种坐标系及其存在的原因：

要谈坐标系变换，那么坐标系有哪些呢？依次有:物体坐标系，世界坐标系，相机坐标系，投影坐标系以及屏幕坐标系.我要讨论的就是这些坐标系间的转换。

这些坐标系不是凭空而来，他们都是为了完成计算机3D图形学最最最基本的目标而出现.计算机3D图形学最最最基本的目标就是:**将构建好的3D物体显示在2D屏幕坐标上.**初看好像就是将最初的物体坐标系转换到屏幕坐标系就可以了呀，为什么多出了世界坐标系,相机坐标系，投影坐标系。这是因为:在一个大世界里有多个物体，而每个物体都有自己的坐标系，如何表述这些物体间相对的关系，这就多出了世界坐标系；如果只需要看到这个世界其中一部分，这就多出了相机坐标系；至于投影坐标系那是因为直接将3D坐标转换为屏幕坐标是非常复杂的(因为它们不仅维度不同,度量不同（屏幕坐标一般都是像素为单位,3D空间中我们可以现实世界的米，厘米为单位），XY的方向也不同，在2D空间时还要进行坐标系变换),所以先将3D坐标降维到2D坐标,然后2D坐标转换到屏幕坐标。

对于整个“如何将3D物体投射并显示在2D屏幕上”这一过程而言，坐标系转换的顺序为：

物体坐标系—>世界坐标系—>相机坐标系—>投影坐标系—>图像(像素)坐标系编号设为a. b. c. d. e

当然，本篇的目的是说明如何将3D物体投射并显示在2D屏幕上，且难点也在于此(即b,c,d,e过程)。a到b的过程不在本文讨论之内，不再赘述。如果很在意知识结构的完整性，请参考

<http://www.cnblogs.com/shanhaobo/articles/1065380.html>

该篇a.b过程十分详细。

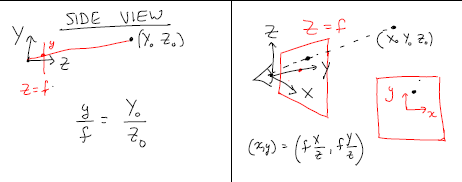
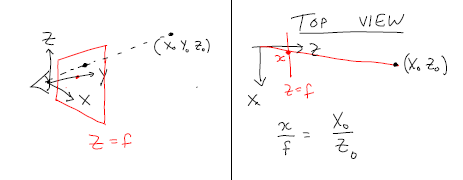
1. 3D-2D投影基础：图像的投影几何：

如何将空间中的点投射到2维图像中？这是一个问题。

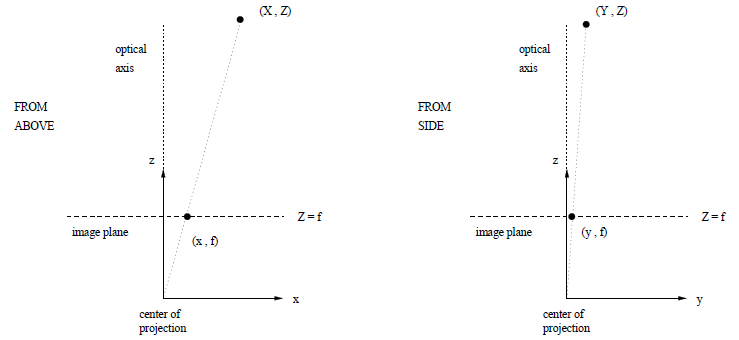
假设空间中的一个点的坐标是(X0, Y0, Z0) (这个坐标称作“相机坐标系”(camera coordinate system))，相对应的2D投影点坐标（此处是“投影坐标系”(projection coordinate) 。如果要进行成像等操作，还要将投影变换到图像坐标(image coordinate)上, 这些都在后面给出）

 （1）

不用捉急，请看下图 （第一，第四幅图的Z和Y标反了）：



其中，轴Z为光轴（optical axis）；这个横截面Z=f被叫做图像平面(image plane)或投影平面(projection plane)，与Z轴交点为图像位置(image position)，该点被称为主点(principal point)(这两个个概念很重要，会影响到投影坐标系到图像坐标系的转换)；原点称为投射中心(center of the projection)。如下图：



二．世界坐标系与相机坐标系（含转化问题）：

1、世界坐标系向相机坐标系映射流程：

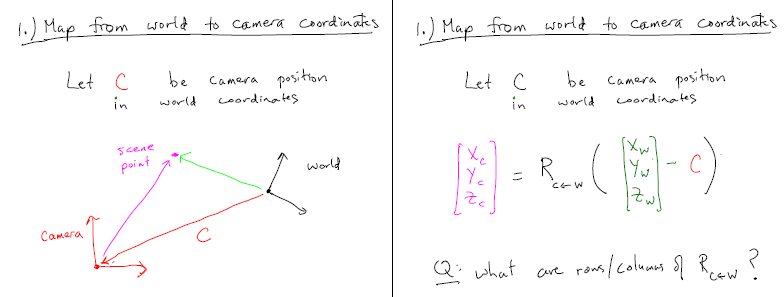
A. 世界坐标向相机坐标映射；

B. 投影到投影平面；

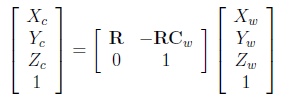
C. 从投影平面向pixel coordinate(就是image cooridnate)映射。

2、具体操作

A．相机外部参数（Camera extrinsic (or external) parameters）



然后，有：

 （2）

其中Cw就是图中的C，是相机中心在世界坐标中的3D点的坐标。R就是Rc<—w所以，从世界坐标到相机坐标的转换就是4X4的平移矩阵与4X4的旋转矩阵的乘积，如下式：

 （3）

旋转矩阵R和平移向量Cw便被称为相机的外部参数，也即相机在世界坐标中的朝向与位置(orientation and position)。

B．投影矩阵: 三维到二维（Projection Matrix: 3D to 2D）

接下来，我们需要将相机坐标投影到图像平面(Z=f, image plane)。在第一部分中，本文给出了如何将相机坐标中的3D点投影到2D的过程，即(1)式：



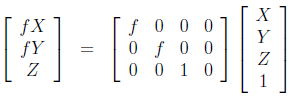
将(x,y)写成齐次坐标的形式，有：



知，如果Z不为0，有：



因而，我们能够将从3D点(X,Y,Z)(这个点是在相机坐标系中的)到2D投影点(x,y)的投影过程用3X4的矩阵描述成 (4)式：



（以上是Z轴为光轴，还可以讨论一下Y轴为光轴和X轴为光轴）

(4)

而此3X4的矩阵便为投影矩阵。关于此矩阵仍有一些讨论，比如Z是0和不0，无穷有穷，是不是原点什么的，因为过于理论，对于搞工程的亲们，洒家就不再赘述了。

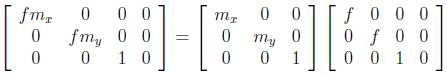
C. 相机内部参数（Camera intrinsic (or internal) parameters）

这一部分不太容易理解。不急，慢慢来。

通常，通过之前的步骤，我们已经能够将3D点投影到2D图像平面 (image plane，也叫投影平面, projection plane) 上了。但是，这并不够。因为，这样一个在投影平面上的图像是在投影坐标系下的，它的单位还是物理单位，位置也是物理位置。比如其单位可能是米，厘米，毫米等。因而，下一步，我们需要将投影坐标系转换至图像（也叫像素，pixel）坐标系。需要从其真实的物理位置(physical positions)转换至像素位置(pixel positions)。这就需要有**放缩(scaling)**与**平移(translation)**操作。

假设，图像物理单位是mm（毫米）。放缩的目的是要确定一个mm中究竟有几个像素（因为通常情况下，一个mm并不精确对应1个像素）；平移的目的是将图像原点移动到左下角（即通常情况下，图像左下角为(0,0)点，而投影坐标系中，图像是不满足这个要求的）。

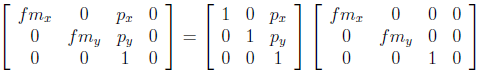
**对于放缩**操作，有下式：

 (5)

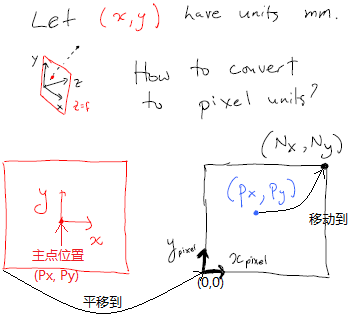
其中，mx表示x方向上，一单位(此处我们假设是1mm)中所含有像素个数；同理，my是y方向上的。需要注意的是，尽管mx与my数值上通常非常接近，但一般并不严格相等。

**对于平移**操作，我们需要知道主点（在第一部分中有介绍）的物理位置，这个位置通常是固定的。举个例子，显示的图像（在图像坐标系中）必然是在显示在显示屏幕上的，而屏幕则是由像素构成。则主点的位置（也即图像位置，在第一部分有说明）便是在这个物理屏幕的中心。如果屏幕包含1536行X2048列像素，则主点位置在(767.5, 1023.5)处。而我们的目的，就是将这个放缩好了的图像平移，使这个主点（也即目前的图像中心）移动至当前图像右上角处；而当前图像左下角移动至现在的主点位置，因而，当前的主点位置(767.5,1023.5)平移后便是(0,0)。这一段话够费解。我们再解释一二，后面跟张图，就清楚了。

在图像坐标系中，我们假定主点的位置是(px,py)，这个理论上就是上段中的(767.5, 1023.5),当然，他们之前可能会有细微的不用，比如像素不可能不是整数。因而要获得投影点(x,y)的像素坐标，我们需要在原来的投影矩阵中加入平移量：

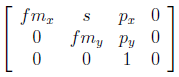
 (6)

请看下图：

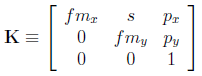


我觉得这张图已经相当清楚了。不在赘述。

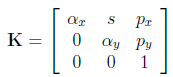
而对于(6)式左侧的放缩平移矩阵，我们应当明白，这是相当理想的情况。理想状况自然就是不靠谱的情况，所以，我们还需要加入其它的参数s，这个参数叫偏度参数（skew parameter），控制了一定的错切(shear)变换因而原来的矩阵变为：



通常，我们给予左侧3X3的部分一个特殊的名字，

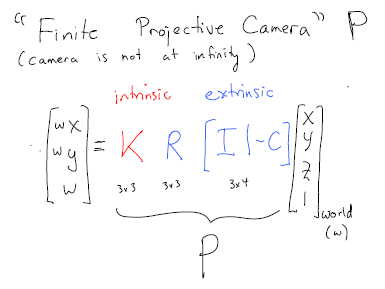


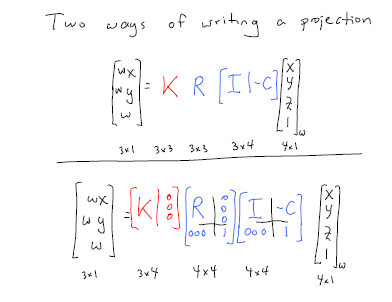
这个K，我们叫做相机标定矩阵(camera calibration matrix)，有时也写作：

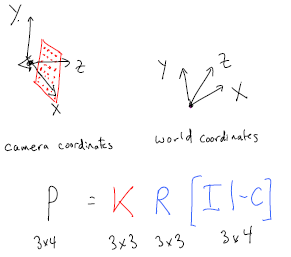


这个东西，便是千呼万唤始出来的相机内部参数了。

三．世界坐标系到图像坐标系的综合。







K一般是给好的。所以在实际编程中，不被过分在意。

至此。便讨论清楚了。

如果仍有疑问或其他需求，请自行查阅网站：

<http://www.cim.mcgill.ca/~langer>

里面都是瑰宝。是加拿大麦吉尔大学（加拿大最nb的大学）教授的页面，里面很多内容。讲解详细清楚。

很多网上的中文资料实在是令人遗憾。不是驴唇不对马嘴，就是简略抽象晦涩难懂。作为小白，为小白们服务义不容辞。

改变这个状况。