胡说八道——齐次坐标

什么是齐次坐标

计算机图形学和计算机视觉中往往采用齐次坐标（Homogeneous Coordinate）。传统的2D欧几里德空间坐标表示为(x, y)，3D空间则表示为(x, y, z)，这种坐标称为非齐次坐标（Inhomogeneous Coordinate）。齐次坐标是在传统坐标系的基础上增加一维，即$(x, y) \mapsto (x, y, 1)$，$(x, y, z) \mapsto (x, y, z, 1)$，或者更通用的表示$(x, y) \mapsto (kx, ky, k)$，$(kx, ky, kz) \mapsto (kx, ky, kz, k)$。在齐次坐标系中，(kx,ky,kz,…,k)与(x,y,z,…,1)是等价的。下表给出了非齐次坐标和齐次坐标的对应关系：

非齐次坐标	齐次坐标
(x, y)	$(x, y, 1), (2x, 2y, 2), \ldots, (kx, ky, k)$
(x, y, z)	$(x, y, z, 1), (2x, 2y, 2z, 2), \ldots, (kx, ky, kz, k)$

齐次坐标看上去很古怪，为什么要用它呢？原因很简单，为了更好的数学性质。先来看一个简单的例子。l是2D空间中的一条直线，那么直线方程就是：

$$ax + by + c = 0 \tag{1}$$

我们把上述方程(1)改写为矩阵乘法的形式：

$$AX = \begin{bmatrix} a & b \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = -c \tag{2}$$

方程(2)的解X，其值恰好是$(x,y)^T$，即(x,y)如果满足方程(2)就是直线l上的点。但方程(1)也可以改成如下形式：

$$AX = \begin{bmatrix} a & b & c \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0 \tag{3}$$

方程(3)的解X，其值为$(x,y,1)^T$。并且进一步我们可以发现$kAX = 0$也是成立的，这意味着kX也是方程的解，其值恰好是$(kx,ky,k)^T$。
我们知道方程(2)被称为**非齐次线性方程**，方程(3)被称为**齐次线性方程**。所以**非齐次坐标**(x,y)就是**非齐次线性方程**的解，**齐次坐标**$(kx,ky,k)^T$就是**齐次线性方程**的解。所以**齐次坐标**和**非齐次坐标**不过是同一个问题，采用不同的方程写法，得出的不同解，二者本质上是完全等价的。
对于齐次坐标$(x,y,\ldots,w)$转换为非齐次坐标的过程，称为非齐次化，如下：

$$(x,y,\ldots,w) \longmapsto (x/w, y/w, \ldots, 1) \tag{4}$$

对于非齐次坐标$(x,y,\ldots,)$转换为齐次坐标的过程，称为齐次化，如下：

$$(x,y,\ldots,) \longmapsto (x,y,\ldots,1) \longmapsto (kx,ky,\ldots,k) \tag{5}$$

齐次坐标和非齐次坐标**本质上等价**，但对应的方程不同。

为什么要用齐次坐标

扯远一点，其实人类发明的坐标系很多，数学上比较常见的是直角坐标系（$x-y-z$）和极坐标系。不用说，我们都知道这两个坐标系是可以相互转换的。什么时候用直角坐标系，什么时候用极坐标系，完全取决于哪个更方便。再稍微扯远一点，GPS用的坐标系是经度纬度坐标系（一种球面坐标系）。这种坐标系也是可以和直接坐标系转换的。为什么用经纬度坐标系？因为地球是个球，用经纬度方便！当然也不是所有地方都方便。如果到了北极点和南极点，经纬度坐标系就抓瞎啦（通常涉及到极点的计算都需要进行坐标转换）。使用齐次坐标系的原因大体也是相同的——**为了方便**。

有什么好方便的？慢慢说吧。

2D直线和交点

按照方程(3)，在齐次坐标系下，直线L可以**表示**为：向量$L=(a,b,c)^T$。直线上的点表示为：向量$X=(kx,ky,k)^T$。故此方程(3)可以重写为：

$$X^TL = \begin{bmatrix} kx & ky & k \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = k(ax + by + c) = 0 \tag{6}$$

于是有如下的性质：

1. 点X_1和点X_2确定的直线$L = X_1 \times X_2$（其中$\times$为叉乘符号）。
2. 直线L_1和直线L_2的交点$X = L_1 \times L_2$。

举例来说，如果直线$L_1 = (-1, 2, 1)^T$（对应直线方程$-x + 2y + 1 = 0$），$L_2 = (3, 1, 2)$（对应直线方程$3x + y + 2 = 0$），则两条直线的交点计算如下：

$$X = L_1 \times L_2 = \begin{vmatrix} i & j & k \\ -1 & 2 & 1 \\ 3 & 1 & 2 \end{vmatrix} = \begin{bmatrix} 3 \\ 5 \\ -7 \end{bmatrix} \tag{7}$$

将X非齐次化，变为$(-3/7, -5/7, 1)$，代入直线方程验证，可知X的确是两条直线的交点（注意必须将X非齐次化，或者将方程改写为$-x + 2y + w = 0$和$3x + y + w = 0$）。

同理，如果有两个点$X_1 = (2, 3, 1)$、$X_2 = (7, 9, 1)$（对应非齐次坐标$(2, 3)$和$(7, 9)$），那么它们定义的直线可以按照如下计算：

$$L = X_1 \times X_2 = \begin{vmatrix} i & j & k \\ 2 & 3 & 1 \\ 7 & 9 & 1 \end{vmatrix} = \begin{bmatrix} -6 \\ 5 \\ -3 \end{bmatrix} \tag{8}$$

直线$-6x + 5y - 3 = 0$的确过$(2, 3)$和$(7, 9)$。

通过上述两个例子可知，用齐次坐标计算两点确定的直线、两条直线的交点，在数学表示上均比传统方式简单。**简单**的前提是知道**叉乘**如何计算。不用担心，现代的编程语言都有叉乘计算的函数库，何况叉乘还可以表示为矩阵运算哦！

设$X_1 = (a, b, c)$，$X_2 = (e, f, g)$，则：

$$X_1 \times X_2 = [X_1]_\times X_2 = \begin{bmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{bmatrix} \begin{bmatrix} e \\ f \\ g \end{bmatrix} \tag{9}$$

用公式(9)就可以将叉乘运算转变为矩阵运算，接着就可以利用矩阵运算的性质和函数库啦。

平行线也相交

如果在齐次坐标下，平行线也是相交的。例如直线$l_1=(1,1,1)^T$和$l_2=(1,1,2)^T$，有交点$l_1 \times l_2=(1,-1,0)^T$（符合方程$x+y+w=0$和$x+y+2w=0$）。$(1,-1,0)^T$显然是无法非齐次化的，因此在非齐次坐标中不存在这个点。换言之，在普通的欧式空间中（直角坐标系），这个点是不存在的，但这个点在现实中却是存在的。没错，**现实中存在相交的平行线**，例如远方的铁轨总是交于一点，这种现象叫做透视。齐次坐标系可以很好的描述这个特性。

图1 相交的平行线

齐次坐标$(x,y,0)^T$称为**无穷远点**，同理也有一条直线过所有的**无穷远点**，这条直线就是$(0,0,1)^T$，称为**无穷远线**。显然还有**无穷远面**的概念。

无论是**无穷远点**还是**无穷远线**都是很难想象的（其实肉眼是可以观察到的，但是中欧几里德的毒太深，反而让人难以理解），**这些概念都是从代数方程中推导出来的**（引力波花了100年才检测到，它也是从方程推导出来的）。这些概念当然不是无用的，**因为它们其实能够更好的刻画客观世界**。

下面举个例子。如下图所示：

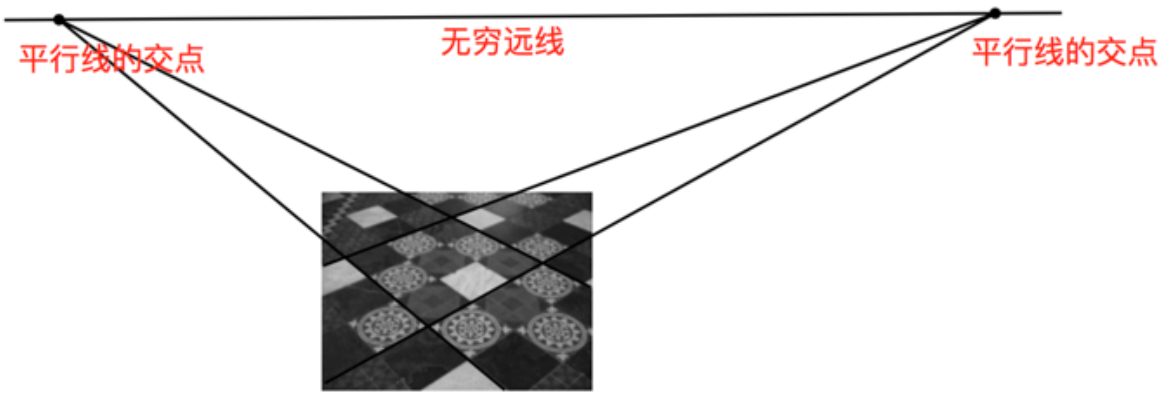

图2 用无穷远线进行图像校正

图像上的图案是正方形，本来应该是平行的，但由于透视，平行线相交于一点。通过找到两组平行线，就可以确定两个交点。这两个交点确定一条直线，该直线就是**无穷远线**。我们知道**无穷远线**等于$(0, 0, 1)^T$，这条线是无论如何在纸上画不出来的。没错，你看到有条线，那么显然这条线**不是无穷远线**，它可能是$(a, b, c)^T$，并且$(a, b, c)^T \neq (0, 0, 1)^T$。

现在反过来想这个问题，如果我们能够找到一个算法，使得$(a, b, c)^T \longmapsto (0, 0, 1)^T$，那么用同样的变换不就是可以将图上的相交的平行线全部校正为不相交吗？校正后的结果如下图：

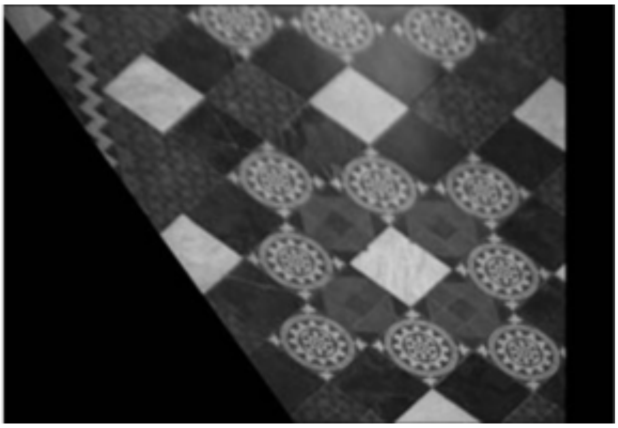
图3 平行线校正后的图案

齐次坐标表示的空间称为**射影空间（Projective Space）**，它对应的几何学称为**射影几何（Projective Geometry）**。射影几何是一门**描述真实世界的几何学**。没错，中学学习的欧几里德几何学是一种理想状态的几何学，很多高级问题用它还真解决不了。

虚圆点（Circular Points）

这是一个非常有趣的概念，很有用，也很难理解。在齐次坐标系下，圆方程可以表示为如下形式：

$$(x-aw)^2+(y-bw)^2=r^2w^2 \tag{10}$$

满足方程的点是$(x,y,w)^T$。可是有两个比较特殊的点也满足这个方程，那就是$(1,i,0)^T$和$(1,-i,0)^T$。你没有看错，这是两个**虚数**！当$w=0$时，方程变为$x^2+y^2=0$，显然$(1,i)^T$和$(1,-i)^T$是它的两个解。$(1,i,0)^T$和$(1,-i,0)^T$称为**虚圆点（Circular Points）**。

虚数自然是画不出来的。我们理解那些看不到的东西总是要稍微困难一点。一个合理的解释是：**你看不见上帝，但是上帝存在，而且上帝在起作用**。数学上有很多类似的概念，**虚圆点**就是这样的。

虚圆点是位于无穷远线上的点（位于$(0,0,1)$），并且它是复数，所以正常情况下，你不应该在图上看到这个点！如果有一张图，上面的某个点，通过严格的证明（其实只需要满足某个公式）被认为是虚圆点，那么这张图一定是有问题的。反过来讲，如果能够找到一个算法，将这个点重新变换到人看不到的地方，那么这张图就正常啦。计算机视觉的很多算法都依赖于**虚圆点**这个概念。

本文不想在这个问题上展开，本文只是想为学习相关算法的同学提供一些方向性的指导。既然是方向性的，那么我们就好好讲一下这类难懂的数学概念。看过《盗梦空间》吧？没看过就好好去看看。电影中男主人公有一个陀螺，在梦里陀螺会一直转，所以男主人公用陀螺来区分梦境和现实。**虚圆点**就是这样的**陀螺**。类似的还有很多数学概念，它们被发明出来，但是你既看不见也摸不着，除了用公式推导，无法证明它们的存在。但如果没有它们，现实和梦境就分不开。所以问题的重点**不是去纠结什么叫陀螺，而是看陀螺是否会停下来**。记住**脱离具体应用环境，脱离算法，这些数学概念毫无价值。**

几何变换

空间坐标的平移、旋转、缩放称之为几何变换。简言之，变换就是用一个公式将某坐标变成另外一个坐标。例如2D平面上坐标为$(x,y)^T$，变换后变为$(x^{'},y^{'})^T$，则记为:

$$(x,y)^T \longmapsto (x^{'},y^{'})^T$$

我们总是希望用简单的公式来表示这个变换。简单，简单，一切都是为了简单。怎么表示比较简单呢？假设我们希望对坐标进行等比例的缩放，即：

$$(x,y)^T \longmapsto \$(sx,sy)^T$$

怎么表示这个运算比较简单？显然可以是这样：

$$\begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} sx \\ sy \end{bmatrix} \tag{11}$$

如果是在2D平面上绕圆点逆时针旋转α角度呢？很显然，是这样：

$$(x,y)^T \longmapsto \$(xcos\alpha - ysin\alpha, xsin\alpha + ycos\alpha)^T$$

写成运算形式就是如下：

$$\begin{bmatrix} cos\alpha & -sin\alpha \\ sin\alpha & cos\alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} xcos\alpha - ysin\alpha \\ xsin\alpha + ycos\alpha \end{bmatrix} \tag{12}$$

那么如果是移动呢？例如：

$$(x,y)^T \longmapsto \$(x + t_x, y + t_y)^T$$

可以写成矩阵运算吗？可以，如下：

$$\begin{bmatrix} t_x \\ t_y \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \end{bmatrix} \tag{13}$$

我们有点不爽的地方，那就是公式(13)和公式(11)(12)不太一样。公式(11)和(12)是乘法运算，公式(13)是加法运算，有没有办法能够把所有运算统一起来？为什么要统一？为了简单啊，你希望编程的时候多一个if语句吗？

怎么统一？用齐次坐标系啊！如下：

$$\begin{gathered} \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} sx \\ sy \\ 1 \end{bmatrix} \\ \begin{bmatrix} cos\alpha & -sin\alpha & 0 \\ sin\alpha & cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} xcos\alpha - ysin\alpha \\ xsin\alpha + ycos\alpha \\ 1 \end{bmatrix} \\ \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix} \end{gathered} \tag{14}$$

计算机图形学里只要涉及坐标变换，无论是几维，都采用齐次坐标系，原因就在于此。

总结

本文主要解释了为什么要使用齐次坐标系。而齐次坐标，以及相关的射影几何，内容是非常丰富的，应用也非常广泛。由于有很多书籍和教材覆盖了这些内容，本文就不再对这些内容进行讲解。我希望我用到的数学表达不会吓到你，因为我的数学水平是非常低的（高考只考了99分，高数才60多分），所以我也不敢进行复杂的推导（怕出错啊）。另外你真需要找本

大牛的书。