Bonus: Gimbal Lock

Krasjet*

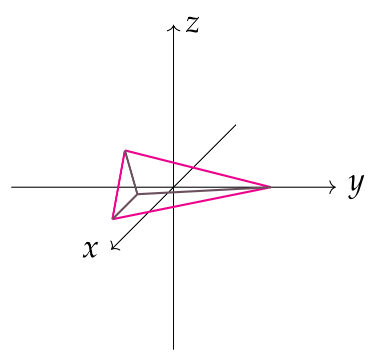
（这个 Bonus 章节是由我很久之前的笔记修改而成，所以有些概念没有从头开始介绍，但对理解应该不会造成太大的困难．如果你想了解的是四元数的相关知识，请点击这里）

在这里，我会简单介绍一下使用欧拉角表示 3D 朝向 (Orientation) 或者旋转时 Gimbal Lock（通常译为**万向锁**或者**万向节死锁**）的产生原因．虽然这方面的资料很多，但是某些解释可能会让人比较费解（比如 Wikipedia 上那个球型 Gimbal 的动画），有时候直接从数学上来理解反而会更直观一点．和四元数的教程一样，我们在这里使用的也是右手坐标系．

0.1 欧拉角

在介绍 Gimbal Lock 之前，我们需要先了解朝向和旋转的欧拉角表示方法．因为欧拉角的资料有很多，我不会从它的最基础讲起．如果你不了解什么是欧拉角，在继续阅读之前可以先读一下欧拉角的 Wikipedia.

首先，假设我们有一个物体，我们使用三个互相正交的坐标轴对它建立一个坐标系：

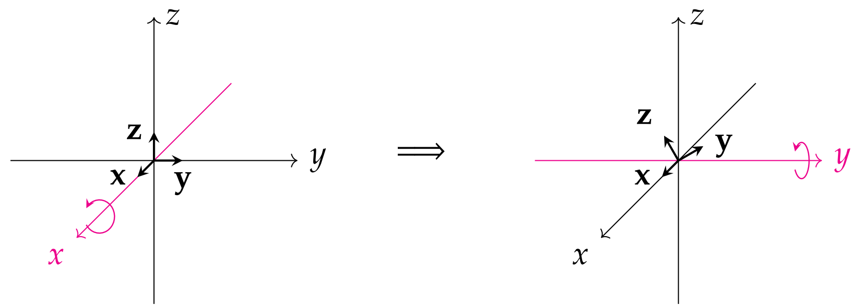


现在，我们的目标就是能改变这个物体的朝向．有一点需要注意的是，**朝向** (Orientation) 和**方向** (Direction) 虽然看起来很像，但是它们是完全不同的两个概念．朝向并不是使用一个向量定义的，我们通常会将朝向定义为将某一个「正朝向」**旋转**至当前朝向所进行的**变换**，所以当你思考朝向的时候，你需要想到的其实是一个旋转（想象一下，我们需要对物体的每一个顶点都施加一个旋转的变换，将它变换到当前的朝向）．

在很多年以前，欧拉就证明了，3D 空间中的任意一个旋转都可以拆分成为沿着物体自身三个正交坐标轴旋转的复合，而欧拉角就规定了这三次旋转的角度，我们分别称他们为 yaw、pitch 和 roll（具体含义可以查看之前提到的 Wikipedia 页面）.

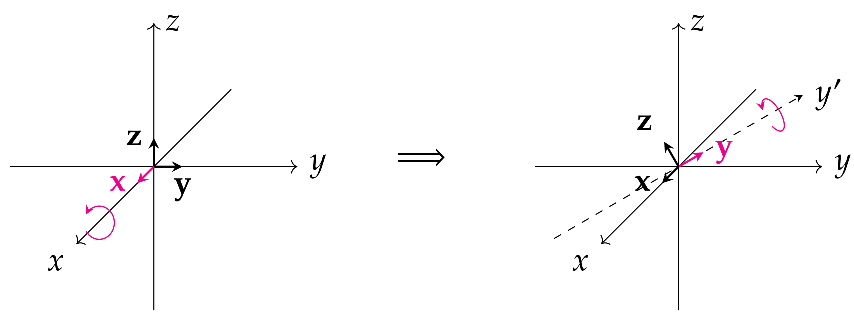
这里我们要特别说明一下**旋转的复合**，它可以分为外旋（Extrinsic Rotation）和内旋（Intrinsic Rotation）. 不同的文献中可能采取不同的定义，但在操作朝向时一般指的是内旋.

外旋指的是，在旋转时，我们采用的旋转轴永远是一个外部、固定的世界坐标系，它不会受到已经进行的旋转的影响:



图中，我们对一个由三个向量组成的局部坐标系首先沿着 x 轴旋转、再沿着 y 轴旋转. 可以看到，在经过第一次旋转之后，尽管局部坐标的 y 轴向量 $\mathbf{y}$ 已经偏离了外部坐标系的 y 轴，我们仍然沿着固定的外部坐标系 y 轴进行第二次旋转.

而内旋正相反. 在旋转时，我们采用的旋转轴依据的是物体本身的坐标系，而这个局部坐标系会被旋转而改变:



可以看到，在第二次旋转中，我们采用的是被旋转过的、局部坐标系的 y 轴，它指向的方向是局部坐标系 $\mathbf{y}$ 向量被旋转后所指的方向. 这个新的旋转轴我们用 y' 来表示. 它与外部坐标系的 y 轴是不同的.

欧拉的定理对外旋和内旋都是成立的. 当我们采用外旋的定义时，我们

可以得出，3D 空间中任意的旋转都可以由三个旋转矩阵相乘的方式得到：

$$E_{\text{extrinsic},xyz}(\text{yaw},\text{pitch},\text{roll}) = R_z(\text{yaw})R_y(\text{roll})R_x(\text{pitch}),$$

其中 R_x、R_y、R_z 是沿着**外部** x、y、z 轴旋转的旋转矩阵（它们的推导不是很困难，线性代数的课本上应该都有）：

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}$$

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}$$

$$R_z(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

这里我们采用的定义是：向量为列向量 $\mathbf{v} = [v_x, v_y, v_z]^T$，施加变换是矩阵左乘向量（有些文献会用行向量，对应的变换就是右乘）

$$\mathbf{v}' = E_{\text{extrinsic}}(\text{yaw},\text{pitch},\text{roll})\mathbf{v}.$$

当我们采用内旋的定义时，这个旋转仍然可以用三个旋转矩阵的相乘而得到，但相乘的顺序正好和外旋相反：

$$E_{\text{intrinsic},xyz}(\text{yaw},\text{pitch},\text{roll}) = R_x(\text{pitch})R_y(\text{roll})R_z(\text{yaw}),$$

注意这里的 R_x、R_y、R_z 仍然是沿着**外部** x、y、z 轴旋转的旋转矩阵．这个公式的由来可能不是很明显，但对 Gimbal Lock 理解关系不是那么大，所以我会在最后的附录中证明．

因为我们将一个旋转拆分成了三个 3D 旋转矩阵的复合，而且 3D 旋转矩阵的相乘一般是不可交换的（即一般 $R_x(\theta)R_z(\phi) \neq R_z(\phi)R_x(\theta)$），所以进行这三次旋转的**顺序**非常重要．在上面的公式中，我们是按照 x-y-z 轴的顺序进行旋转的．当然，这并不是唯一的选择，如果你想用 y-x-z 的顺序进行旋转也是完全可以的．不过你可以发现，按照 x-y-z 轴顺序进行的外旋和按照 z-y-x 轴顺序进行的内旋是等同的，正好相反．

然而，要注意的是，一般情况下，我们一般只会选择其中**固定的一个**顺序来编码旋转或者朝向，而这正是导致 Gimbal Lock 的原因.

0.2 Gimbal Lock（内旋）

还记得，我们将 3D 空间中任意一个旋转写成了三个矩阵相乘的形式. 在下面的讨论中，我们将采用内旋的定义，并固定使用 x-y-z 这一套旋转顺序 $E = E_{\text{intrinsic},xyz}$：

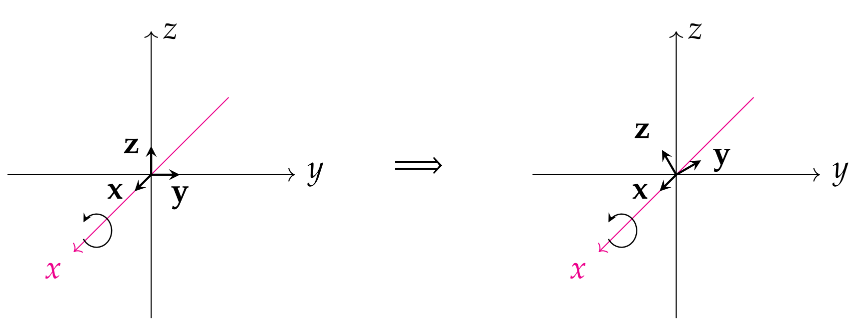
$$E(\text{yaw}, \text{pitch}, \text{roll}) = R_x(\text{pitch})R_y(\text{roll})R_z(\text{yaw}),$$

其中，每一次旋转变换就代表着有一个 Gimbal（通常译为**万向节**或者**平衡环架**）. 所以，在这里我们一共有**固定的**三个 Gimbal. 我们可以改变每次旋转的角度，但是不论怎么变化角度这个旋转的顺序都是不能改变的.

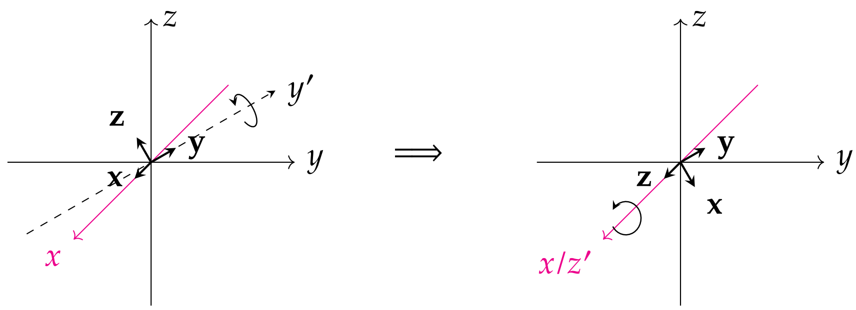
大部分情况下，这一套顺序都能够对**三个**不同的轴分别进行三次旋转. 然而，在某一些特殊情况下，它其中的某两个旋转变换可能变换的是同一个轴（这里指的不是物体自身局部坐标系的轴，而是外部坐标系的轴），这样就导致我们丧失了一个轴的自由度，从而导致 Gimbal Lock 的产生. 虽然看起来有点不可思议，但是下面我们会用实际的例子来说明.

假设我们使用 x-y-z 的顺序旋转任意一个点. 其中，我们沿 x 轴旋转任意度数，沿 y 轴旋转 $\pi/2$ 弧度，最后，我们再沿着 z 轴旋转任意的度数. 我们来仔细研究这个过程.

首先，我们沿着 x 轴旋转任意的度数（比如下图中用的 30°）：



接下来，我们沿局部坐标系的 y 轴旋转 $\pi/2$ 弧度:



注意，经过这一次的变换之后，我们将局部坐标系的 z 轴（z'）变换到**原来**局部坐标系的 x 轴方向．这个变换执行完毕后，我们仅仅只剩下一个 z 轴的旋转了．然而，**当前的**局部坐标系 z 轴与**原来的**局部坐标系 x 轴重合，也就是说，最后 z 轴的旋转与 x 轴的旋转其实操纵的是同一个外部坐标系轴（在图中用红色标出）．三次旋转变换仅仅覆盖了两个外部轴的旋转，一个自由度就这样丢失了，这也就导致了 Gimbal Lock 的现象．

这个变换用公式来理解的话，则是这样（你可以自己来代入验证一下）：

$$
\begin{aligned}
E\left(\alpha, \frac{\pi}{2}, \beta\right) &= R_x(\alpha) R_y\left(\frac{\pi}{2}\right) R_z(\beta) \\
&= \begin{bmatrix} 0 & 0 & 1 \\ \cos(\alpha)\cdot\sin(\beta)+\sin(\alpha)\cdot\cos(\beta) & \cos(\alpha)\cdot\cos(\beta)-\sin(\alpha)\cdot\sin(\beta) & 0 \\ \sin(\alpha)\cdot\sin(\beta)-\cos(\alpha)\cdot\cos(\beta) & \cos(\alpha)\cdot\sin(\beta)+\sin(\alpha)\cdot\cos(\beta) & 0 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 0 & 1 \\ \sin(\alpha+\beta) & \cos(\alpha+\beta) & 0 \\ -\cos(\alpha+\beta) & \sin(\alpha+\beta) & 0 \end{bmatrix} \\
&= R_x(\alpha+\beta) R_y\left(\frac{\pi}{2}\right).
\end{aligned}
$$

利用一些三角恒等式，我将原本由三个旋转矩阵所组成的变换化简成了两个变换矩阵．即便我们分别对 x-y-z 三轴进行了旋转，实际上这个矩阵只等价于一个只旋转 x-y 两轴的旋转，只有两个自由度．$R_x(\alpha)$ 与 $R_z(\beta)$ 这两个变换被合并为单独的一个 $R_x(\alpha+\beta)$ 变换（因为化简完之后变换顺序不一样了，严格来说并不是合并，只不过是能够使用一步来完成）．注意，我们在这里化简的并不是单独的一个变换，而是**一系列**变换．因为它对任意 α 和 β 角都是成立的．无论 x 轴与 z 轴的旋转角是多少，变换都会丧失一个自由度．

从图像来说，不论我们怎么旋转，**z** 这个向量永远只能停留在外部坐标系的 x 轴上（所以，**x** 和 **y** 向量不论怎么旋转都旋转不出外部坐标系的 yz 平面）．如果我们想要将 **z** 转到其它地方，换句话说将局部坐标系的 **xy** 平面旋转出外部坐标系的 yz 平面，则需要再添加一个沿着 **x** 方向或者 **y** 方向的旋转．

当然这也不是做不到，比如说只需要在最内侧再添加一个 x 轴 Gimbal，或者说在变换序列最后再添加一个局部 x 轴的变换，就能解决问题

$$\begin{aligned} E\left(\alpha, \frac{\pi}{2}, \beta, \gamma\right) &= R_x(\alpha) R_y\left(\frac{\pi}{2}\right) R_z(\beta) R_x(\gamma) \\ &= R_x(\alpha+\beta) R_y(\frac{\pi}{2}) R_x(\gamma), \end{aligned}$$

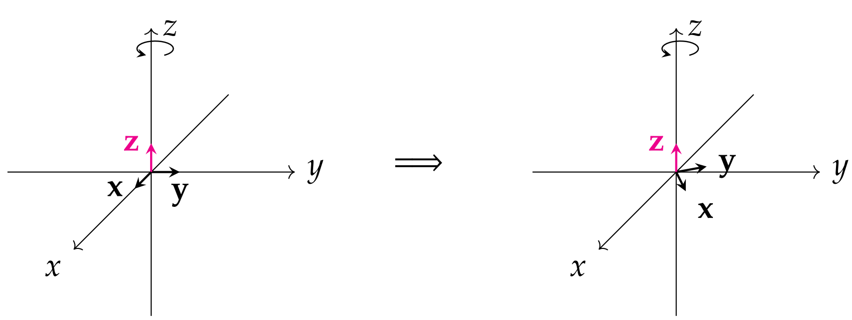
然而，我们之前说过，我们选择的是固定的三个旋转顺序，实际上并没有这个第四个 Gimbal，所以没办法解决这个问题．虽然你在当初选择顺序的时候也可以选择采用四个 Gimbal 的设计，但是这仍不能完全解决问题，因为除了 y 轴之外其它轴的旋转仍然可能将其中的两个轴或者三个轴对齐，只不过现在 Gimbal Lock 产生的几率可能会变小而已．Gimbal Lock 问题的核心还是在于我们采用了固定的旋转顺序．

0.3 Gimbal Lock（外旋）

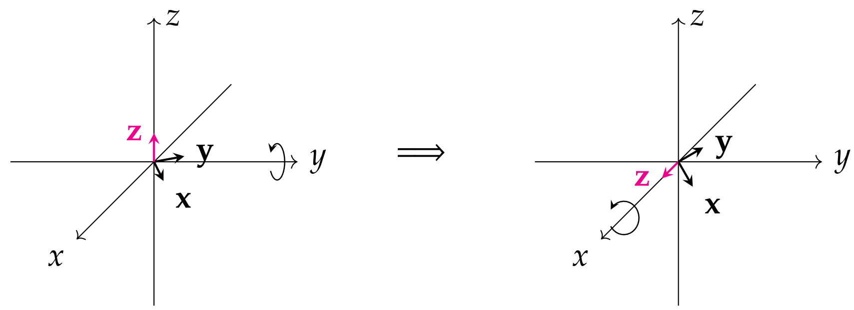
虽然我们知道内旋和外旋在公式上是等同的，都是矩阵相乘，只是顺序相反，即 x-y-z 顺序的内旋等同于 z-y-x 顺序的外旋

$$E(\text{yaw}, \text{pitch}, \text{roll}) = R_x(\text{pitch}) R_y(\text{roll}) R_z(\text{yaw}) = E_{\text{intrinsic},xyz} = E_{\text{extrinsic},zyx},$$

但是它们所代表的含义是不同的．我们知道内旋会造成 Gimbal Lock，因此外旋也不能避免同样的问题，不过通过研究外旋的 Gimbal Lock 可以给我们另一种理解 Gimbal Lock 的方式，所以在这里我们重新研究一下外旋时 Gimbal Lock 到底意味着什么．这次首先我们先旋转外部 z 轴：



然后沿外部坐标系的 y 轴旋转 $\pi/2$ 弧度:



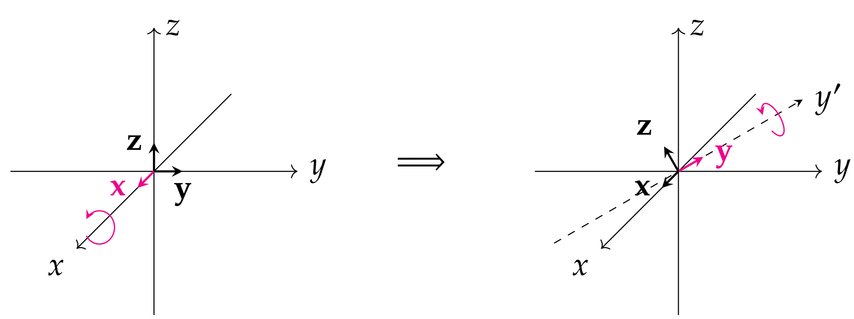
可以看到，现在如果我们仍然沿着最后的外部 x 轴进行旋转，它仍然旋转的是局部坐标系的 z 轴，这就等同于最初外部坐标、局部坐标重合时我们对 z 轴做出的旋转了. 回想之前用三角恒等式推导的公式

$$
\begin{aligned}
E\left(\alpha, \frac{\pi}{2}, \beta\right) &= R_x(\alpha) R_y\left(\frac{\pi}{2}\right) R_z(\beta) \\
&= R_x(\alpha + \beta) R_y\left(\frac{\pi}{2}\right),
\end{aligned}
$$

你可以更清晰看出，外旋时第一个对 z 轴的旋转和最后一个 x 轴旋转其实都可以合并为最后对 x 轴的旋转，我们只获得了两个自由度. 做出旋转之后局部 z 轴 $\mathbf{z}$ 永远只能停留在外部的 x 轴上. 要想将它指向其他朝向，必须要再对外部 x 轴或者 y 轴进行旋转，但这是不可能的，因为我们固定了旋转的顺序.

0.4 附录：内旋的矩阵公式

这里，我们解释一下为什么内旋的公式矩阵相乘的顺序是相反的. 我们只考虑两个旋转复合的情况，多个旋转的情况可以遵循同样的步骤推出. 假设我们先沿着 x 轴旋转，之后再沿着被旋转之后的局部坐标 y 轴的方向 y' 进行旋转:



我们在这里想要证明的是

$$E_{\text{intrinsic},xy} = R_{y'}(\beta)R_x(\alpha) = R_x(\alpha)R_y(\beta).$$

重点是 $R_{y'}$ 这个旋转矩阵，它代表的是沿着 y' 旋转的变换. 我们只要弄清楚它的表达式就可以证明这个等式了. 怎么沿着 y' 进行旋转呢？我们可以进行以下步骤：

1. 暂时撤销所有已有的变换（R_x），将局部 y 轴 y' 还原到外部 y 轴上.
2. 使用旋转矩阵 R_y 对外部 y 轴进行旋转.
3. 再还原之前被撤销的变换（R_x）

也就是说

$$R_{y'}(\beta) = R_x(\alpha)R_y(\beta)R_x(\alpha)^{-1}$$

所以

$$\begin{aligned} R_{y'}(\beta)R_x(\alpha) &= R_x(\alpha)R_y(\beta)R_x(\alpha)^{-1}R_x(\alpha) \\ &= R_x(\alpha)R_y(\beta). \end{aligned}$$

这也就解释了为什么矩阵的相乘是相反的.

要注意的是 $R_{y'}$ 的公式中，在两个旋转的特例中被撤销和被还原的变换是一个旋转，但最好将它考虑为先施加一个逆变换，再施加原本的变换，而不是仅仅是旋转. 在两个以上旋转复合的情况下这个变换是多个旋转的复合.

□ 最后修改：2025/07/16