三维视觉入门

3D-Vision Learning By Doing

作者：Steve Canvas

时间：December 28, 2021

版本：1.0

The important thing is not to stop questioning. —— Albert Einstein

特别声明

自 2019 年 ElegantLaTeX 系列模板上线 GitHub、CTAN 以来，受到很多用户的喜爱。

2020 年，我打算做 ElegantLaTeX 的最后一个版本，也就是原定计划 ElegantBook 4.x 版本为 ElegantLaTeX 系列模板的终止符。基于我想把 4.x 做成一个最终版本，我计划了很多事情，包括将代码转为 dtx，将三个模板的文档打包进 dtx 里面，然后重新设计封面，补充各种页面，增加元素等等。我想的很多，但是做起来并不是很顺利，中间也发生了很多事情，不想解释。直至今年 4 月，我决定，不论如何，先把 4.1 发布出来。

另外，在临近 ElegantLaTeX 模板告别之际，我想和各位用户说：多分享，多奉献。

<u>如果你无法认同我的想法，建议直接删除本模板。</u>

Ethan Deng

May 2, 2021

目录

第 1 章 线性代数的简单回顾

之后我们会用到一些线性代数的知识，其中有一些我认为是比较核心的一些概念，就在这里首先回顾一下，还有一些会用到的知识会放在附录里面。默认向量为列向量。

1.1 内积与外积

设两个列向量 $\mathbf{a} = (a_1, a_2, a_3)^T, \mathbf{b} = (b_1, b_2, b_3)^T$

- 内积

a,**b** 的内积是一个标量，可以表示为 $<\mathbf{a}, \mathbf{b}>$、$\mathbf{a} \cdot \mathbf{b}$、$\mathbf{a^T b}$、$\mathbf{b^T a}$

$$\begin{aligned} <\mathbf{a}, \mathbf{b}> &= a_1b_1 + a_2b_2 + a_3b_3 \\ &= \|\mathbf{a}\| \cdot \|\mathbf{b}\| cos\theta \end{aligned}$$

其中 $\|a\|$ 表示 a 的模长，为 $\|\mathbf{a}\| = \sqrt{\mathbf{a} \cdot \mathbf{a}}$，$\theta$ 是向量 **a**，**b** 的夹角。

- 外积

向量外积的结果是一个向量。方向根据右手定则确定，模长为 **a**，**b** 围成的面积大小，即

$$\|\mathbf{a} \times \mathbf{b}\| = \|\mathbf{a}\|\|\mathbf{b}\| \sin\theta$$

用坐标的形式来表达

$$\begin{aligned} \mathbf{a} \times \mathbf{b} &= \begin{pmatrix} a_2b_3 - a_3b_2 \\ -a_1b_3 + a_3b_1 \\ a_1b_2 - a_2b_1 \end{pmatrix} \\ &= \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \\ &= \mathbf{a}^\wedge \mathbf{b} \triangleq [\mathbf{a}]_\times \mathbf{b} \end{aligned}$$

外积可以变成一个反对称矩阵和一个向量的乘积。而且可以看到反对称矩阵里面只有三个元素，和向量元素个数是对应的。也就是说一个三维向量和对应的反对称矩阵是一一对应的。

我们定义运算符：

- $\mathbf{a}^\wedge = A$: hat operation，表示将向量变成对应的反对称矩阵
- $A^\vee = \mathbf{a}$: vee operation，表示将反对称矩阵变成对应的向量。

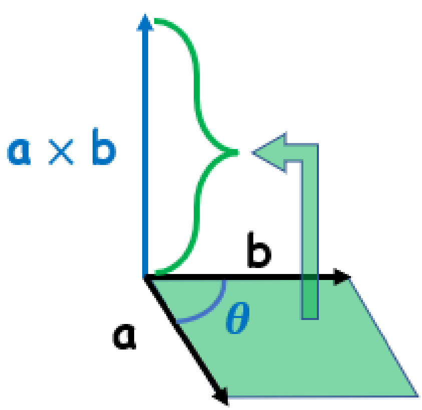


图 1.1: 向量叉积

1.2 向量与坐标

$\mathbf{x} \in \mathbb{R}^n$, x 既可以被看作向量, 也可以认为被认为是一个点。当我们对点对 $(x_1, x_2, x_3, \cdots, x_n)^T$ 来表示时如何区分，什么时候是点，什么时候是向量呢？我们以二维欧氏空间 $\mathbb{E}^2$ 来举例子，如下图

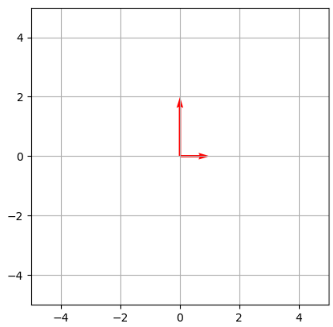


图 1.2: 二维欧式空间中的两个向量

如图表示了两个向量, 用点对来表示 $(2, -2) - (0, 0) = (2, -2)$, $(4, 0) - (2, 2) = (2, -2)$ 如果只给了一个点对 $(2, -2)$, 不说明是点还是向量的时候，我们可以认为 $(2, 2)$ 是向量，也可以认为是一个点。

那是否有一个数学上的表示，可以将点和向量不混淆在一起呢？

我们将点 $(2, -2)$ 表示成 $(2, -2, 1)$, 将向量 $(2, -2)$ 表示成 $(2, -2, 0)$。将二维坐标用三维坐标来表示，通过增加一个变量的代价，我们就可以只通过点对，来区分点和向量，将他们分别表示。可以看起来似乎有一些愚蠢，要将向量和点进性解耦。别急，之后你会体会到这样做的极大便利。

定义 1.1 (齐次坐标)

更一般的我们将 $\mathbf{x} \in \mathbb{R}^n$, 增加一个维度将其表示为 $(x_1, x_2, x_3, \cdots, x_n, \omega)^T$ 如果 $\omega = 0$，则 $\mathbf{x}$ 表示的是一个向量。如果 $\omega \neq 0$ 则 $\mathbf{x}$ 表示一个点，点的坐标为 $(\frac{x_1}{\omega}, \frac{x_2}{\omega}, \frac{x_3}{\omega}, \cdots, \frac{x_n}{\omega})^T$. 这种表示方法叫做齐次坐标。 ♣

齐次的意思是只我们用 n+1 个变量来表示 n 维空间。齐次坐标转换成非齐次坐标也就是刚才除以 ω 的操作。

此外，如果用一个点 (u, v) 用齐次坐标表示的话，如果我们给他前面乘以一个常数 z，那么他实际坐标是不会变得。

$$z\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} zu \\ zv \\ z \end{pmatrix}$$

因为转化成二维坐标需要除以 ω，所以 $(u, v) = (\frac{zu}{z}, \frac{zv}{z})$。

1.3 坐标变换

一个三维空间点的数学表示为

$$\mathbf{p} = (x_p, y_p, z_p)^T \equiv \mathbf{o} + x_p\mathbf{x} + y_p\mathbf{y} + z_p\mathbf{z} \tag{1.1}$$

其中 $\mathbf{x}, \mathbf{y}, \mathbf{z}$ 是 $\mathbb{R}^3$ 中的一组基。

通常我们习惯使用标准正交基 $\mathbf{o} = (0, 0, 0)^T$, $\mathbf{x} = (1, 0, 0)^T = \mathbf{i}$, $\mathbf{y} = (0, 1, 0)^T = \mathbf{j}$, $\mathbf{z} = (0, 0, 1)^T = \mathbf{k}$.

我们用另一组基，来表示同一个点

$$\mathbf{p} = (u_p, v_p, w_p)^T \equiv \mathbf{e} + u_p\mathbf{u} + v_p\mathbf{v} + w_p\mathbf{w} \tag{1.2}$$

我们式1.1与式1.2联立，就可以得到一个点在一组基下的坐标如何转换到另一组基下的坐标表示

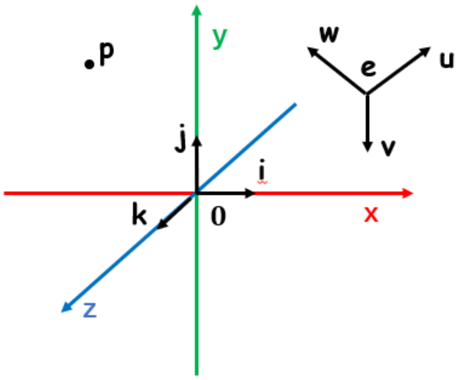


图 1.3: 两组基在坐标系下的表示

$$
\begin{pmatrix} x_p \\ y_p \\ z_p \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & x_e \\ 0 & 1 & 0 & y_e \\ 0 & 0 & 1 & z_e \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_u & x_v & x_w & 0 \\ y_u & y_v & y_w & 0 \\ z_u & z_v & z_w & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_p \\ v_p \\ w_p \\ 1 \end{pmatrix}
$$

$$
= \begin{pmatrix} x_u & x_v & x_w & x_e \\ y_u & y_v & y_w & y_e \\ z_u & z_v & z_w & z_e \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u_p \\ v_p \\ w_p \\ 1 \end{pmatrix}
$$

定义 1.2 (坐标变换公式)

点 $\mathbf{P}$ 在向量空间 $(\mathbf{i}, \mathbf{j}, \mathbf{k}, \mathbf{0})$ 下的坐标与在向量空间 $(\mathbf{u}, \mathbf{v}, \mathbf{w}, \mathbf{e})$ 下的转换关系。

$$
\mathbf{p_{ijk}} = \begin{pmatrix} \mathbf{u} & \mathbf{v} & \mathbf{w} & \mathbf{e} \\ 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{p_{uvw}}
$$

♣

可以注意到通过使用非齐次坐标，我们将形如 $Ax + b$(x 为齐次坐标) 的形式变成了 Tx(x 为非齐次坐标) 的形式。

注: 基只要线性无关即可，不需要标准正交。但我们讨论的基一般是默认是标准正交基。实际上基只需要施密特正交化，标准化之后就可以化为标准正交基。此外很显然当给定一个点对 (x, y)，不特指基的话，默认就是 $\mathbf{i}, \mathbf{j}, \mathbf{k}$ 下的坐标。

1.4 矩阵乘法

一个矩阵可以按照行或者列分为行向量组或者列向量组，所以两个矩阵的乘法 AB，按照行列分解的不同，会有四种不同的理解方式。

1.4.1 与向量的关系

$$
AB = \begin{pmatrix} 1 & 2 \\ 2 & 1 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 1 & 3 & 2 & 1 \\ 2 & 3 & 1 & 2 \end{pmatrix}
$$

- A-列向量组，B-行向量组

$$\left(\begin{array}{c|c} 1 & 2 \\ 2 & 1 \\ 3 & 4 \end{array}\right)\left(\begin{array}{cccc} 1 & 3 & 2 & 1 \\ \hline 2 & 3 & 1 & 2 \end{array}\right) = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}\begin{pmatrix} 1 & 3 & 2 & 1 \end{pmatrix} + \begin{pmatrix} 2 \\ 1 \\ 4 \end{pmatrix}\begin{pmatrix} 2 & 3 & 1 & 2 \end{pmatrix}$$

- A-列向量组，B-列向量组

$$\left(\begin{array}{c|c} 1 & 2 \\ 2 & 1 \\ 3 & 4 \end{array}\right)\left(\begin{array}{c|c|c|c} 1 & 3 & 2 & 1 \\ 2 & 3 & 1 & 2 \end{array}\right) = \begin{pmatrix} \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 \end{pmatrix}$$

$$\gamma_1 = 1 \cdot \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} + 2 \cdot \begin{pmatrix} 2 \\ 1 \\ 4 \end{pmatrix} \quad \gamma_2 = 3 \cdot \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} + 3 \cdot \begin{pmatrix} 2 \\ 1 \\ 4 \end{pmatrix}$$

$$\gamma_3 = 2 \cdot \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} + 1 \cdot \begin{pmatrix} 2 \\ 1 \\ 4 \end{pmatrix} \quad \gamma_4 = 1 \cdot \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} + 2 \cdot \begin{pmatrix} 2 \\ 1 \\ 4 \end{pmatrix}$$

- A-行向量组，B-列向量组

$$\left(\begin{array}{cc} 1 & 2 \\ \hline 2 & 1 \\ \hline 3 & 4 \end{array}\right)\left(\begin{array}{c|c|c|c} 1 & 3 & 2 & 1 \\ 2 & 3 & 1 & 2 \end{array}\right) = \dots$$

- A-行向量组，B-行向量组

$$\left(\begin{array}{cc} 1 & 2 \\ \hline 2 & 1 \\ \hline 3 & 4 \end{array}\right)\left(\begin{array}{cccc} 1 & 3 & 2 & 1 \\ \hline 2 & 3 & 1 & 2 \end{array}\right) = \begin{pmatrix} \eta_1{}^T \\ \eta_2{}^T \\ \eta_3{}^T \end{pmatrix}$$

$$\eta_1{}^T = 1 \cdot \begin{pmatrix} 1 & 3 & 2 & 1 \end{pmatrix} + 2 \cdot \begin{pmatrix} 2 & 3 & 1 & 2 \end{pmatrix}$$

$$\eta_2{}^T = 2 \cdot \begin{pmatrix} 1 & 3 & 2 & 1 \end{pmatrix} + 1 \cdot \begin{pmatrix} 2 & 3 & 1 & 2 \end{pmatrix}$$

$$\eta_3{}^T = 3 \cdot \begin{pmatrix} 1 & 3 & 2 & 1 \end{pmatrix} + 4 \cdot \begin{pmatrix} 2 & 3 & 1 & 2 \end{pmatrix}$$

1.4.2 左乘与右乘

通常矩阵不满足交换律，即 $AB \neq BA$. 我们从基变换的角度来形象的理解一下。

如果我们令章节 1.2 这里的 e为 0，如图 1.4 我们可以得到基之间的变换，即**过渡矩阵 C**。

定义 1.3 (基变换与坐标变换)

$\mathbf{x}, \mathbf{y} \in \mathbb{R}^3$，基 $(\mathbf{i}, \mathbf{j}, \mathbf{k})$ 下的坐标为 $\mathbf{x}$, 基 $(\mathbf{u}, \mathbf{v}, \mathbf{w})$ 下的坐标为 $\mathbf{y}$, 则基变换与坐标变换分别为

$$\begin{pmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \end{pmatrix} C = \begin{pmatrix} \mathbf{u} & \mathbf{v} & \mathbf{w} \end{pmatrix}$$

$$\mathbf{x} = C\mathbf{y}$$

♣

之前我们谈到一组点对既可以看作向量，也可以看作点。但是如果是基的话，无论齐次还是非齐次我们统一认为是向量，而不是点。我们给这种点对起一个名字叫做 **free vector**。

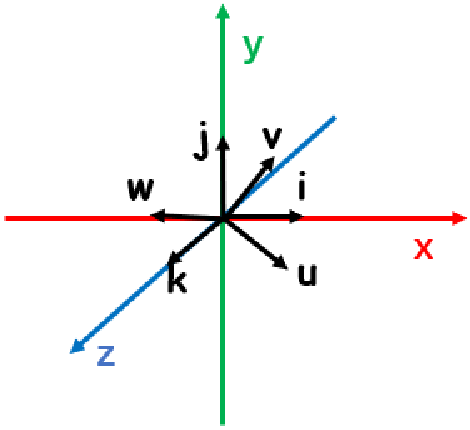


图 1.4: 无平移变换的两组基在坐标系下的表示

观察到基变换的是右乘 C，坐标变换是 y 左乘 C。也就是说**左乘是坐标变换，右乘是基变换**这是什么意思呢？为了理解我们不妨考虑

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 0 \\ 2 & 2 \end{pmatrix}$$

我们将 B 作为对象，考虑 AB 与 BA 对 B 造成的不同影响。我们从几何的角度来理解，这里

$$AB = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 2 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 4 & 4 \end{pmatrix}$$

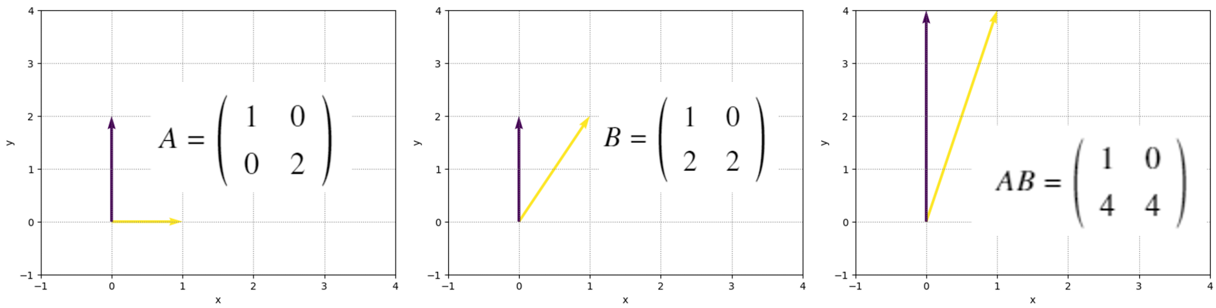


图 1.5: 矩阵 A 左乘 B

如图 1.5，我们将 A，B 按列分，然后画到坐标轴上。我们来看 A 的物理意义: $A\mathbf{i} = \mathbf{i}, A\mathbf{j} = 2\mathbf{j}$，也就是说 A 左乘以后 **i** 方向上长度不变，**j** 方向上长度变成原来的两倍。所以可以看到 AB 与 B 的第一行元素一样，第二行元素变成的原来的两倍。AB 就是将 B 看成坐标，对 B 做 A 的变换，最后得到在基 $(\mathbf{i}, \mathbf{j})$ 下的坐标。

接下来我们考虑 BA 如图 1.6,

$$BA = \begin{pmatrix} 1 & 0 \\ 2 & 2 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 2 & 4 \end{pmatrix}$$

A 的作用还是类似的，只不过作用对象变成了 B 的列向量。记 $B = (\beta_1, \beta_2)$，β_1, β_2 是列向量 A 作用在 β_1, β_2 上，使得 β_1 保持不变 β_2 长度变成原来两倍，方向不变。也就是说 BA 是将 B 看成基，对基做了变换，而不是坐标的变换。

这就是之前所讲的左乘是坐标变换，右乘是基变换的意义。所以矩阵一般是不满足交换律的。

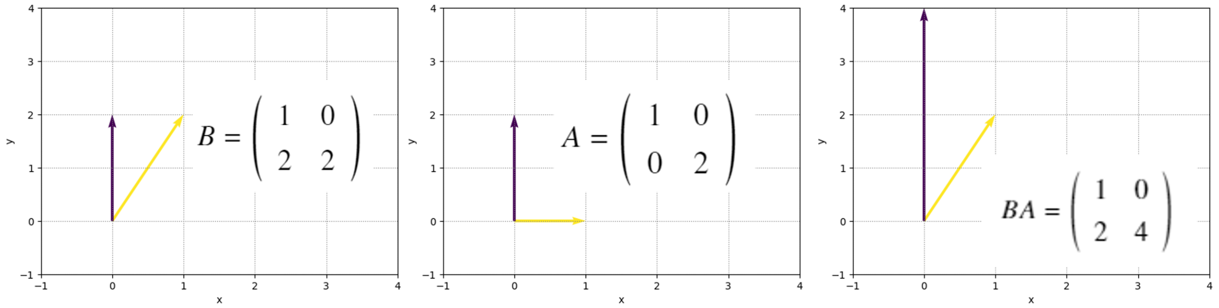


图 1.6: 矩阵 A 右乘 B

1.5 线性变换在不同基下的表示

我们来看这样一个变换 $A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$，如图1.7所示，它实际代表的物理意义得到坐标关于直线 $y = x$ 对称的点：$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} y \\ x \end{pmatrix}$

不做特别说明的话，默认点 $(x, y)^T$ 是在基 $\mathbf{i} = (1, 0)^T$，$\mathbf{j} = (0, 1)^T$ 下的表示，即 $\begin{pmatrix} x \\ y \end{pmatrix} = x\mathbf{i} + y\mathbf{j}$，所以变换 A 也是在基 $\mathbf{i}, \mathbf{j}$ 下的变换。

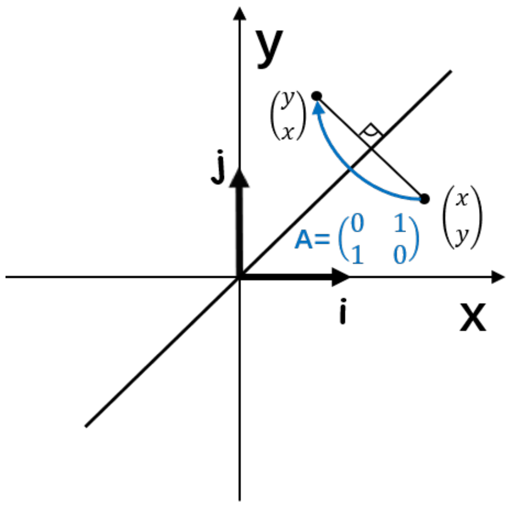


图 1.7: A 变换

我们现在换一组基 $\mathbf{m}, \mathbf{n}$ 来讨论关于关于直线 $y = x$ 对称的变换矩阵 $\mathbf{B}$ 是如何组成的。

$$\begin{cases} \mathbf{m} = \mathbf{i} + \mathbf{j} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ \mathbf{n} = -\mathbf{i} + \mathbf{j} = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \end{cases}$$

写成矩阵的形式:

$$\begin{pmatrix} \mathbf{m} & \mathbf{n} \end{pmatrix} = \begin{pmatrix} \mathbf{i} & \mathbf{j} \end{pmatrix}\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \triangleq \begin{pmatrix} \mathbf{i} & \mathbf{j} \end{pmatrix}\mathbf{P}$$

现在我们得到了的一组新的基 $\mathbf{m}, \mathbf{n}$，而且基 $\mathbf{m}$ 于直线 $y = x$ 共线，基 $\mathbf{n}$ 垂直于直线 $y = x$。关于直线 $y = x$

对称的变换 B，不会改变基 **m**，但会使得 **n** 反向。即：

$$\begin{cases} \mathbf{Bm} = \mathbf{m} \\ \mathbf{Bn} = -\mathbf{n} \end{cases}$$

所以如图1.8，就可以很简单的得到 $B = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$

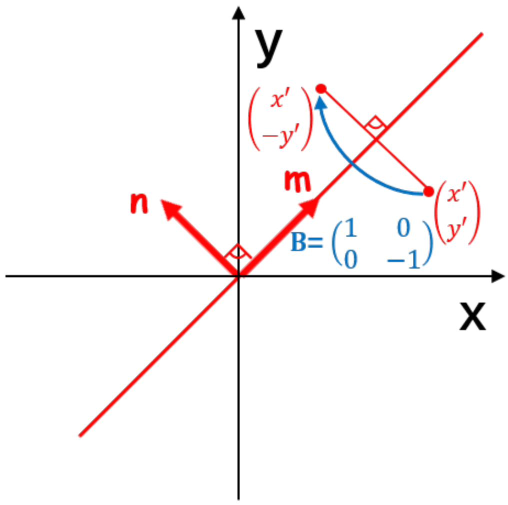


图 1.8: B 变换

A 可以通过怎样的变换得到 B 呢？其实也就是相似。即：

$$\begin{aligned} B = P^{-1}AP &= \frac{1}{2}\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \\ &= \frac{1}{2}\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \\ &= \frac{1}{2}\begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \end{aligned}$$

稍微做个总结，如图1.9。

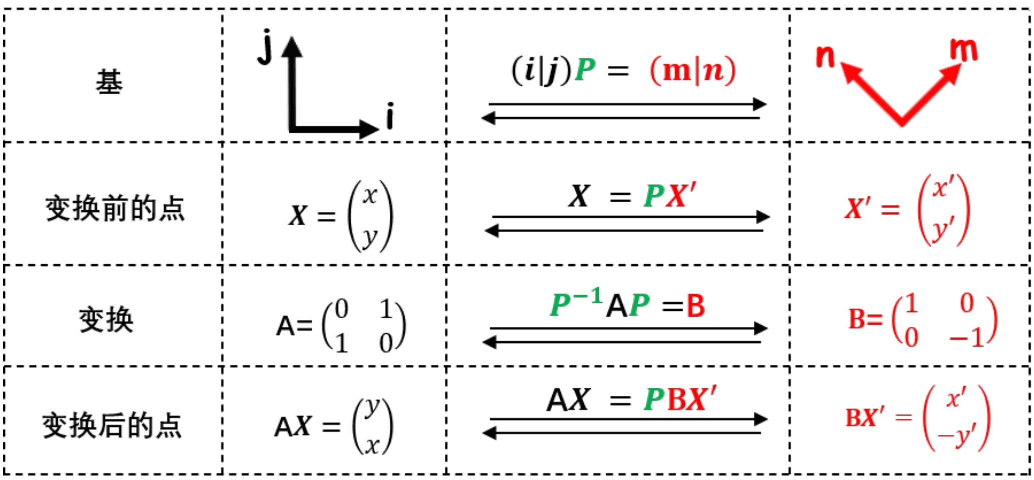


图 1.9: 不同基下表示的变换

第 2 章 三维变换

直接说变换会有一点点抽象，我们通过三维物体在变换前后的变化来理解三维变换。我们现在姑且将三维物体数学表示为点的集合, 即 $\{p_i|p_i \in \mathbb{R}^3\}$ 方便起见我们只将一个点 $p = (x, y, z)^T$ 作为一个对象来研究三维空间的变换。显然用一个可逆矩阵 A 来描述变换，即 Ap。这里我们着重研究平移和旋转变换。

笔记 这里不严谨将平移理解成位置信息，旋转理解成所谓的朝向信息。想想一下我们人眼观察到的物体首先取决于我们眼睛的位置在哪里，其次也取决于我们眼睛的凝视的方向。

2.1 平移变换

对一个点做平移非常简单，只需 $p' = (x, y, z)^T + (t_x, t_y, t_z)^T$, 但是有一个问题是如何写成矩阵 Ap 的形式呢？可以使用齐次坐标 $p_h = (x, y, z, 1)^T$。

$$p'_h = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

这里将平移写成一个矩阵的形式好处在于，能够将各种变换统一成矩阵乘积 Ap 的形式。

2.2 旋转变换

旋转矩阵可能稍微有一些复杂，我们先从二维的来看。之前我们提到过，点 $P(x, y)^T$ 可以表示为 $P = x\mathbf{i} + y\mathbf{j}$ 我们引入一个矩阵 R 来描述旋转变换，即 $P' = RP$，

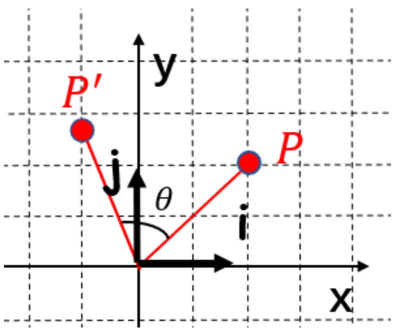


图 2.1: 二维旋转

$$\begin{aligned} RP &= R(x\mathbf{i} + y\mathbf{j}) \\ &= xR\mathbf{i} + yR\mathbf{j} \\ &= x(cos\theta\mathbf{i} + \sin\theta\mathbf{j}) + y(-\sin\theta\mathbf{i} + cos\theta\mathbf{j}) \\ &= x\begin{pmatrix} cos\theta \\ \sin\theta \end{pmatrix}(\mathbf{i}, \mathbf{j}) + y\begin{pmatrix} -\sin\theta \\ cos\theta \end{pmatrix}(\mathbf{i}, \mathbf{j}) \\ &= \begin{pmatrix} cos\theta & -\sin\theta \\ \sin\theta & cos\theta \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} \end{aligned}$$

这里可以看到，基 $\mathbf{i}, \mathbf{j}$ 变换之后是什么，对应的变换矩阵 A（这里是旋转矩阵 R）就是什么，换句话说

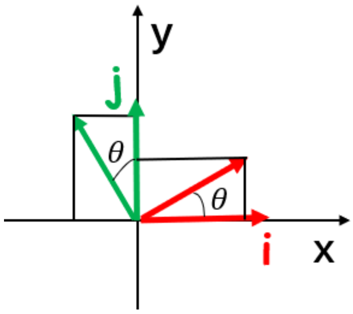


图 2.2: 基 ij 的旋转

定理 2.1 (线性变换的矩阵定理)

A: $\mathbb{R}^2 \to \mathbb{R}^2$ 的一个线性变换，A 的列向量为 $A(i), A(j)$，即：

$$A = \begin{pmatrix} A\mathbf{i} & A\mathbf{j} \end{pmatrix}$$

笔记 这里其实也包含左乘右乘的问题。如果想得到点 P 旋转以后，在基 $\mathbf{i},\mathbf{j}$ 下的点 P'，左乘 $P' = RP$。如果想得到旋转后的基 $\mathbf{i}',\mathbf{j}'$，右乘，$[\mathbf{i}',\mathbf{j}'] = [\mathbf{i},\mathbf{j}]R = R$。所以你要时刻清楚，自己想得到的是经过变换后的点在原来的基下的新坐标 (左乘)，还是想把当前的基经过变换得到一组新的基 (右乘)。变换是同一个变换 (R)，但是左乘和右乘导致了完全不同的意义。

直观来看，旋转是不会改变向量的长度的，所以旋转矩阵的行列式为 ±1。但是很容易验证我们推导出的 R 的行列式为 1，那么-1 是什么情况呢？我们来看这么一个矩阵

$$mirror = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

根据2.1线性变换矩阵定理，我们很容易理解这个变换。如图2.3，可以看到变换前后的手性 (右手变换到左手) 发生了变化。

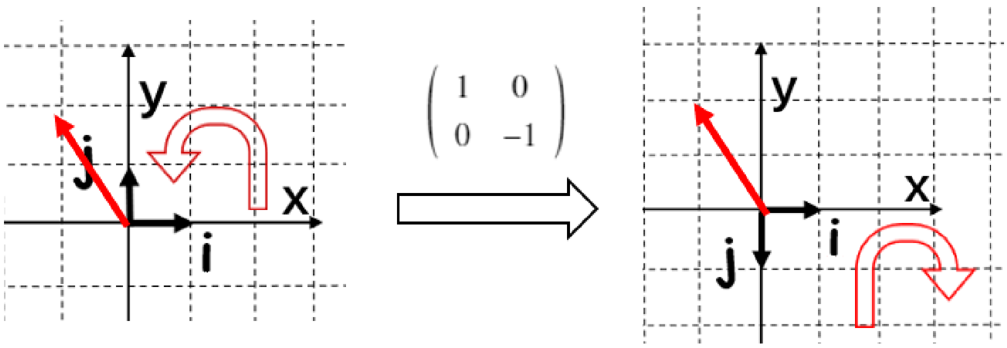


图 2.3: 手性变换

旋转变换不应该改变手性，所以行列式为 1。

实际上在所有的类型的可逆矩阵当中，**有且只有旋转矩阵能同时保证变换前后的长度和手性不变。**

此外还有一个小小的细节，就是旋转角度 θ 的正负问题。不管手性如何，θ 的旋转角度总是和手性的角度一致的。

如果绕着某一个点进行旋转，旋转矩阵又如何组成呢？可以先将点平移到原点，在旋转，最后在平移回到原来的位置。完成习题 2。

2.2.1 旋转矩阵

根据2.1线性变换矩阵定理，就可以得到绕 x，y，z 轴旋转的旋转矩阵。

$$\mathbf{R}_x(\alpha)=\begin{pmatrix}1&0&0&0\\0&\cos\alpha&-\sin\alpha&0\\0&\sin\alpha&\cos\alpha&0\\0&0&0&1\end{pmatrix}\quad \mathbf{R}_y(\beta)=\begin{pmatrix}\cos\beta&0&\sin\beta&0\\0&1&0&0\\-\sin\beta&0&\cos\beta&0\\0&0&0&1\end{pmatrix}\quad \mathbf{R}_z(\gamma)=\begin{pmatrix}\cos\gamma&-\sin\gamma&0&0\\\sin\gamma&\cos\gamma&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}$$

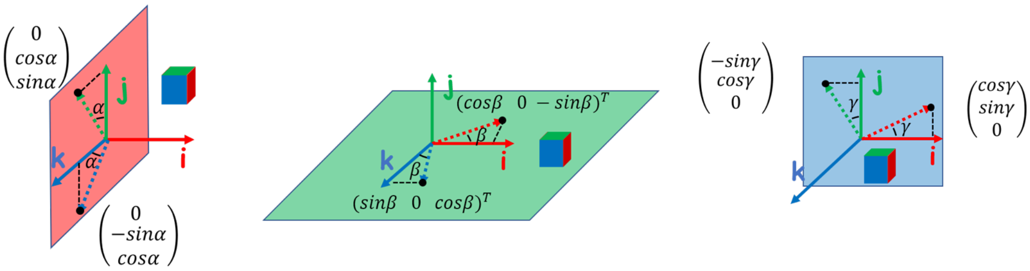


图 2.4: 绕坐标轴旋转的旋转矩阵

现在得到的旋转矩阵是绕，x，y，z 旋转的旋转矩阵，那么如何得到绕任意轴旋转的旋转矩阵呢？

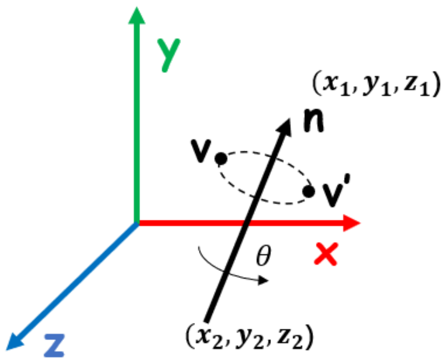


图 2.5: 点 v 绕任意的轴旋转

2.2.2 轴角

我们来解决上一小节的问题。为了问题简单一些，我们先分析旋转轴过原点的情况，旋转轴不过原点的情况之后讨论。

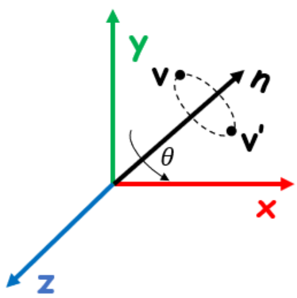


图 2.6: 点 v 绕任意的轴旋转

假设一个点 $\mathbf{v}$，绕旋转轴 $\mathbf{n}=(n_x,n_y,n_z)$（$\|\mathbf{n}\|=1$），旋转 θ 后得到点 $\mathbf{v}'$。

如何求得 $\mathbf{v}'$，由罗德里格斯公式给出，由于推导过程有些复杂，在附录B.1中给出。这里直接给出结果：

定理 2.2 (罗德里格斯公式)

假设一个点 $\mathbf{v}$，绕过原点旋转轴 $\mathbf{n}=(n_x,n_y,n_z)$（$\|\mathbf{n}\|=1$），旋转 θ 后得到点 $\mathbf{v}'$。

$$\begin{aligned}\mathbf{v}' &= \left[\mathbf{I}+(1-cos\theta)*N^2+\sin\theta*N\right]\mathbf{v}\\ &= R_{\mathbf{n}}(\theta)\mathbf{v}\end{aligned}$$

其中 $N=\mathbf{n}^\wedge$

这个公式非常重要，强烈建议要自己推导一下，并且要有深刻理解。

轴角有些时候会写成这样的形式：

$$\begin{aligned}\omega &= \theta\mathbf{n}\\ &= (\theta n_x,\theta n_y,\theta n_z)^T\\ &= (\omega_x,\omega_y,\omega_z)^n\end{aligned}$$

ω 一般被叫做**旋转向量**。这样的好处在于相比存储 n_x,n_y,n_z,θ 少存储了一个变量。需要角度 θ 和旋转轴 $\mathbf{n}$ 的时候只需通过简单的计算：

$$\begin{cases}\|\omega\|=\theta\\ \mathbf{n}=\frac{\omega}{\|\omega\|}\end{cases}$$

这样我们得到了绕过原点的轴旋转的变换，如果旋转轴不过原点，如图2.5。我们只需将旋转轴平移到原点，然后旋转以后，在平移回去即可。旋转轴 $\mathbf{u}=(\mathbf{x_1}-\mathbf{x_2},\mathbf{y_1}-\mathbf{y_2},\mathbf{z_1}-\mathbf{z_2})^{\mathbf{T}}$ 归一化以后，$\mathbf{n}=\frac{\mathbf{u}}{\|\mathbf{u}\|}$。

$$\begin{pmatrix}x_v'\\y_v'\\z_v'\\1\end{pmatrix}=\begin{pmatrix}1&0&0&x_2\\0&1&0&y_2\\0&0&1&z_2\\0&0&0&1\end{pmatrix}R_{\mathbf{n}}(\theta)\begin{pmatrix}1&0&0&-x_2\\0&1&0&-y_2\\0&0&1&-z_2\\0&0&0&1\end{pmatrix}\begin{pmatrix}x_v\\y_v\\z_v\\1\end{pmatrix}$$

2.2.3 欧拉角

如果我们通过罗德里格斯公式2.2（分别令 $\mathbf{v}=(1,0,0)^T,(0,1,0)^T,(0,0,1)^T$ 得到 $\mathbf{R}$ 的三列即可）将 $\mathbf{R_n}(\theta)$ 具体写出来：

$$\mathbf{R_n}(\theta)=\begin{pmatrix}n_x^2(1-\cos\theta)+\cos\theta & n_xn_y(1-\cos\theta)-n_z\sin\theta & n_xn_z(1-\cos\theta)+n_y\sin\theta\\ n_xn_y(1-\cos\theta)+n_z\sin\theta & n_y^2(1-\cos\theta)+\cos\theta & n_yn_z(1-\cos\theta)-n_x\sin\theta\\ n_xn_z(1-\cos\theta)-n_y\sin\theta & n_yn_z(1-\cos\theta)+n_x\sin\theta & n_z^2(1-\cos\theta)+\cos\theta\end{pmatrix}$$

如果描述我现在旋转的样子，这显然非常的不直观，所以通常在和别人交流的时候会使用 **欧拉角**。欧拉角将绕一个旋转轴 $\mathbf{n}$ 旋转 θ 角度分解成一次绕三个轴旋转旋转三个角度。

比如我们这里先绕 x 轴旋转，再绕 y 轴旋转，最后再绕 z 轴旋转。(XYZ 顺序)

$$\mathbf{R_n}(\theta)=\mathbf{R}_z(\gamma)\mathbf{R}_y(\beta)\mathbf{R}_x(\alpha)$$

根据绕旋转轴的先后顺序问题，有 XYZ，ZYX 等等的差异。

方向锁 (Gibmal Lock)

2.2.4 四元数

在我们推导罗德里格斯公式的时候（见章节B.1.2），我们提到 $R_{\mathbf{n}}(\theta)$ 用四个参数 n_x,n_y,n_z,θ，三个自由度（因为 $\|n\|=1$）表示了绕过原点的旋转轴旋转 θ 的变换。旋转变换有三个自由度，而 $R_{\mathbf{n}}(\theta)$ 是一个 3×3 的矩阵，有 9 个变量，这显然是冗余的。到这里我们似乎可以采取相似对角化的想法（如我们在章节 1.5中描述的那样），

对 $R_{\mathbf{n}}(\theta)$ 做相似对角化，但是旋转矩阵的特征值除了 1 以外（显然旋转轴绕自己旋转不变，即 $\mathbf{Rn}=\mathbf{n}$），通常会有复数，求出特征值有复数，更别说求变换矩阵 p 还需要求特征向量了，你可以想象这个计算量并不低。

举一个绕 z 轴旋转的矩阵 $\mathbf{R}_z(\gamma)=\begin{pmatrix}\cos\gamma & -\sin\gamma & 0\\ \sin\gamma & \cos\gamma & 0\\ 0 & 0 & 1\end{pmatrix}$ 的特征值。先求特征多项式：

$$|\lambda E-\mathbf{R}_z(\gamma)|=(1-\lambda)\left(\lambda^2-2\lambda\cos\gamma+1\right)=0$$

1. $\gamma=0\qquad \lambda_1=\lambda_2=\lambda_3=1$
2. $\gamma=\pi\qquad \lambda_1=1,\lambda_2=\lambda_3=-1$
3. $0<\gamma<\pi\qquad \lambda_1=1,\lambda_2=e^{i\gamma},\lambda_3=e^{-i\gamma}$

你可以直观的感觉到，如果绕任意轴旋转的化，旋转矩阵的特征值很大概率会有复数，所以相似对角化不太可行。看起来如果从存储变量的角度来描述旋转，矩阵并不是一个很好的方式，尽管我们可以利用2.1线性变换的矩阵定理来直观的理解旋转。

那么是否有一个好的数学形式，存储了尽可能少的变量来表示旋转呢？正是**四元数** (Quaternion)。现在我们来讨论旋转与四元数的关系，四元数和复数很类似，四元数有三个的虚部 i,j,k，一个实部用 q 来表示一个四元数，$\mathbb{H}$ 来表示四元数的集合。对于所有的 $q\in\mathbb{H}$，可以写成如下形式

$$q=w+xi+yj+zk\quad(w,x,y,z\in\mathbb{R})\tag{2.1}$$

其中

$$i^2=j^2=k^2=ijk=-1$$

ijk 之间的乘积不满足交换律

$$\begin{cases}ij=k,ji=-k\\ jk=i,kj=-i\\ ki=j,ik=-j\end{cases}$$

ijk 之间乘积的关系可以用图2.7结合叉积右手法则来记忆。以 ij 相乘举例：

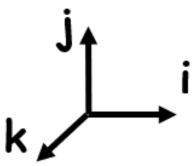


图 2.7: 四元数虚部乘积的关系

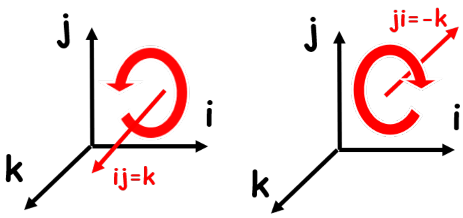


图 2.8: 四元数虚部 ij 的乘积

为了方便，也会把四元数 q 记成标量（实部）和向量（虚部）的形式：

$$q=[s,\mathbf{v}],\quad \mathbf{v}=\begin{pmatrix}x\\ y\\ z\end{pmatrix},s,x,y,z\in\mathbb{R}$$

有了 ijk 乘积之间的关系就可以定义四元数 q_1，q_2 的乘积 $q_1 = a + bi + cj + dk$, $\quad q_2 = e + fi + gj + hk$

$$\begin{aligned} q_1q_2 =& ae + afi + agj + ahk + \\ & bei - bf + bgk - bhj + \\ & cej - cfk - cg + chi + \\ & dek + dfj - dgi - dh \\ =& (ae - bf - cg - dh) + \\ & (be + af - dg + ch)i \\ & (ce + df + ag - bh)j \\ & (de - cf + bg + ah)k \end{aligned}$$

也可以整理成矩阵形式

$$q_1q_2 = \begin{pmatrix} a & -b & -c & -d \\ b & a & -d & c \\ c & d & a & -b \\ d & -c & b & a \end{pmatrix} \begin{pmatrix} e \\ f \\ g \\ h \end{pmatrix}$$

更普遍情况下我们会使用标量和向量的形式表示乘积：

定义 2.1 (四元数 Graßmann 积)

$q_1, q_2 \in \mathbb{H}, q_1 = [s, \mathbf{v}] \quad , q_2 = [t, \mathbf{u}]$

$$q_1q_2 = [st - \mathbf{v} \cdot \mathbf{u}, s\mathbf{u} + t\mathbf{v} + \mathbf{v} \times \mathbf{u}] \tag{2.2}$$

♣

可以很容易验证四元数乘法一般不满足交换律，即 $q_1q_2 \neq q_2q_1$

笔记 $q_1 = a + bi + cj + dk, q_2 = e + fi + gj + hk$ 乘积如果写成矩阵形式，注意一下左右乘积的问题：

$$q_1q_2 = \begin{pmatrix} a & -b & -c & -d \\ b & a & -d & c \\ c & d & a & -b \\ d & -c & b & a \end{pmatrix} \begin{pmatrix} e \\ f \\ g \\ h \end{pmatrix}$$

$$q_2q_1 = \begin{pmatrix} a & -b & -c & -d \\ b & a & d & -c \\ c & -d & a & b \\ d & c & -b & a \end{pmatrix} \begin{pmatrix} e \\ f \\ g \\ h \end{pmatrix}$$

类似复数有模长，逆，共轭，纯虚数等概念，四元数 $q = [s, \mathbf{v}]$ 也有

- 模长（范数）：$\|q\| = \sqrt{s^2 + \|\mathbf{v}\|^2} = \sqrt{s^2 + \mathbf{v} \cdot \mathbf{v}}$
- 逆：$qq^{-1} = q^{-1}q = 1 \quad (q \neq 0)$
- 共轭：$q^* = [s, -\mathbf{v}]$，其中 $q^*q = qq^* = \|q\|^2$，这个特殊的乘法满足交换律
- 纯四元数：$q = [0, \mathbf{v}]$

根据如上我们可以得到四元数的逆：

$$\begin{aligned} qq^{-1} &= 1 \\ q^*qq^{-1} &= q^* \\ (q^*q)\, q^{-1} &= q^* \\ \|q\|^2 \cdot q^{-1} &= q^* \end{aligned}$$

即：

$$q^{-1} = \frac{q^*}{\|q\|^2}$$

如果是**单位四元数** ($\|q\| = 1$) 的话，$q^{-1} = q^*$。

简单介绍四元数以后，我们来推导如何用四元数表示三维旋转，推导过程和罗德里格斯公式非常类似，我们在附录B.2中给出，这里直接给出结论：

定理 2.3 (四元数表示三维旋转)

三维点 $\mathbf{v}$ 绕旋转轴 $\mathbf{n}(\|n\| = 1)$ 旋转 θ 之后的 $\mathbf{v}'$，可以使用四元数乘法来获得。

令 $v = [0, \mathbf{v}], q = [cos(\frac{1}{2}\theta), \sin(\frac{1}{2}\theta)\mathbf{n}]$，则：

$$v' = qvq^* = qvq^{-1}$$

这里最后得到的 v' 也是一个纯四元数，如果想得到三维坐标，只需拿出虚部即可。

2.2.4.1 插值

我们来看一个这样一个问题：给定了相机的两个位置，我们如何平滑的从一个位置过渡到另一个位置？如图2.9。

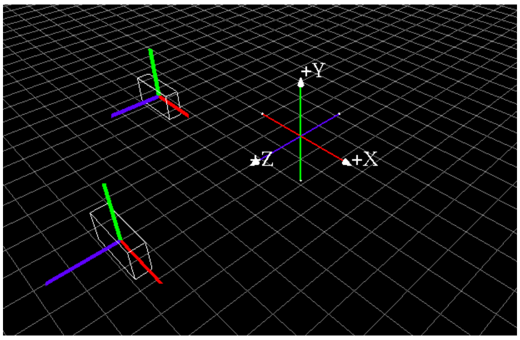


图 2.9: 相机的平移

举个例子，如图2.10，如果得到图中白点的三个位置，我们就可以得到过渡的位置了。如果我们需要过渡的更平滑，只需要增加插值的点就可以了。

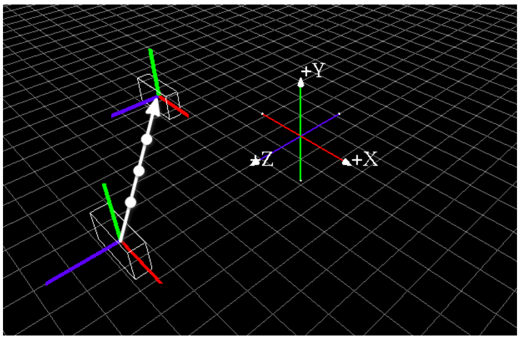


图 2.10: 相机平移的插值

由于是 $\mathbb{R}^3$ 到 $\mathbb{R}^3$ 的插值，只需要使用线性插值（**L**inear **Interp**olation），简称为 Lerp 就可以了。

我们把图2.10，简化成图2.11 根据向量加减法，可以得到：

$$\begin{aligned}\mathbf{v}_t &= = \mathbf{v}_0 + t\,(\mathbf{v}_1 - \mathbf{v}_0) \\ &= (1-t)\mathbf{v}_0 + t\mathbf{v}_1 \\ &\triangleq \mathrm{Lerp}\,(\mathbf{v}_0, \mathbf{v}_1, t)\end{aligned} \tag{2.3}$$

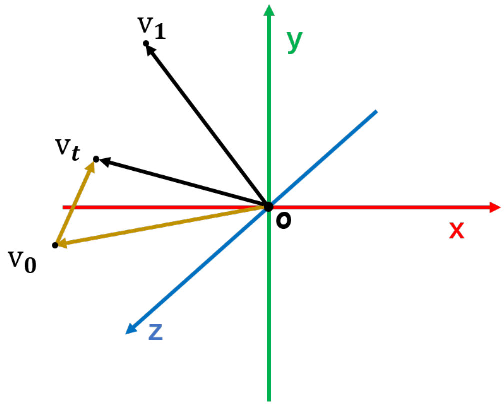
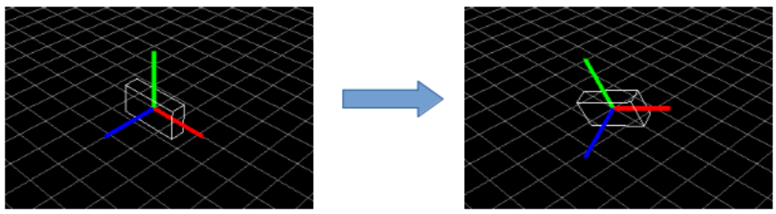


图 **2.11:** 线性插值

关于 t 的确定，只需规定 t 在 0-1 之间即可。

- $t=0$ 时，$\mathbf{v}_t=(1-0)\mathbf{v}_0+0\mathbf{v}_1=\mathbf{v}_0$
- $t=1$ 时，$\mathbf{v}_t=(1-1)\mathbf{v}_0+1\cdot\mathbf{v}_1=\mathbf{v}_1$
- $t=\frac{i}{4}, i=1,2,3$ 时，就可以得到图2.10的相机位置。

但是如果相机发生了旋转，我们又如何平滑的过渡呢？换句话说，旋转之间如何插值呢？



我们用单位四元数来表示旋转，如果我们直接应用 Lerp 到四元数上，会得到

$$q_t=\mathrm{Lerp}\left(q_0,q_1,t\right)=(1-t)q_0+tq_1 \tag{2.4}$$

你可以简单验证一下只有 $t=0$ 或者 $t=1$ 的时候，$\|q_t\|=1$。

这里我们用几何的方式来直观理解一下，我们知道单位四元数表示旋转，那么旋转变换应该是在 $\mathbb{E}^4$ 的单位球上。可以我们没有办法将一个四维的单位球画在二维的平面上，所以我们用三维球上的点代表旋转变换，当然这样数学上是不严谨的，但我们只是为了有一个直观上的理解，并借此定性的说明一些问题。

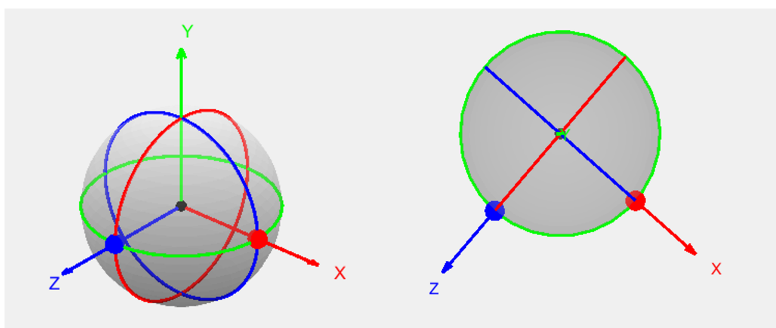


图 **2.12:** 左图：三维球面上的蓝色小球 q_0，红色小球 q_1，分别表示两个不同的旋转变换
右图：左图的俯视图

笔记 小球其实表示的是一个点，这里绘制成小球，只是为了看起来方便一些，下同。

直接将 Lerp 作用于旋转，会得到如图2.13所示的黑色小球。如图2.13所示，不言自明。显然将 Lerp 直接作用于旋转进行插值是不对的，那么我们正确的 q_t 应该是什么样子的呢？

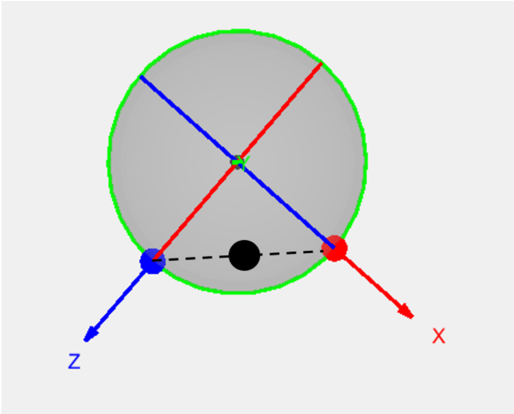


图 2.13: 黑色小球为 $q_t = \text{Lerp}(q_0, q_1, t)$

如图2.14，我们期望得到的 q_t 的样子应该位于球面上。所以我们只需要调整一下黑色小球的模长就可以了，我们把这种先用 Lerp 再调整模长的方式记作 Nlerp（Normalized Linear Interpolation）:

$$q_t = \text{Nlerp}(q_0, q_1, t) = \frac{(1-t)q_0 + tq_1}{\|(1-t)q_0 + tq_1\|} \tag{2.5}$$

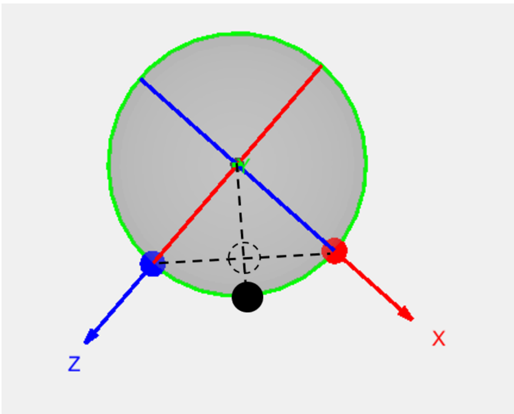


图 2.14: 因为四元数模长为 1，只有球面上的小球才能表示旋转。所以我们想得到的黑色小球应该位于球面上

至此我们就可以用 Nlerp 来做四元数插值了。但是是否还有不足呢？我们想象一下蓝色小球在绿色圆环上运动，也就是说蓝色小球运动到红色小球的位置的过程，始终是在圆周（球面）上进行的。而 Nlerp 小球实际是在“弦”上运动的，只不过我们出于模长的问题，最后调整了一下模长，但**本质是在还是在直线上运动的**。当红球与蓝球相差不大的时候，圆周与弦的差距就不大。但是当红球与篮球相差到一定程度的时候问题就会出现，我们会发现 Nlerp 并不会平滑的过渡。

多说无益，一图胜千言。如图2.15，我们分别取 $t = 0.25, 0.5, 0.75$ 得到三个插值点。黑色小球是一种平滑的过渡。可以看到从蓝色小球、三个黑色小球与红色小球之间间隔是等间距的，复合 $t = 0- > 0.25- > 0.5- > 0.75- > 1.0$ 的递增规律。我们把这种平滑的插值叫做 Slerp（Spherical Linear Interpolation）

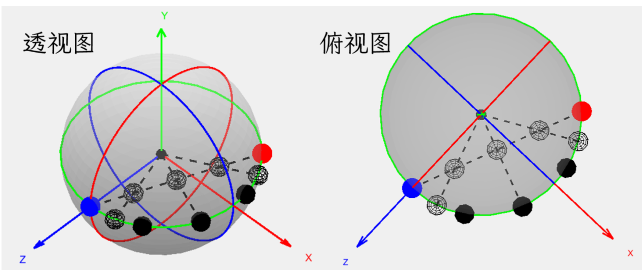


图 2.15: Lerp, Nlerp, Slerp 的比较

如图2.15，注意一下 $t = 0.5$ 时，Nlerp 与 Slerp 结果是一样的。所以绿色圆周上看起来只有两个黑色空心小球了，剩下的那个黑色空心小球与黑色实心小球重叠了。

- Lerp：红蓝小球连线上的黑色空心小球。
- Nlerp：绿色圆周上的黑色空心小球。
- Slerp：绿色圆周上的黑色实心小球。

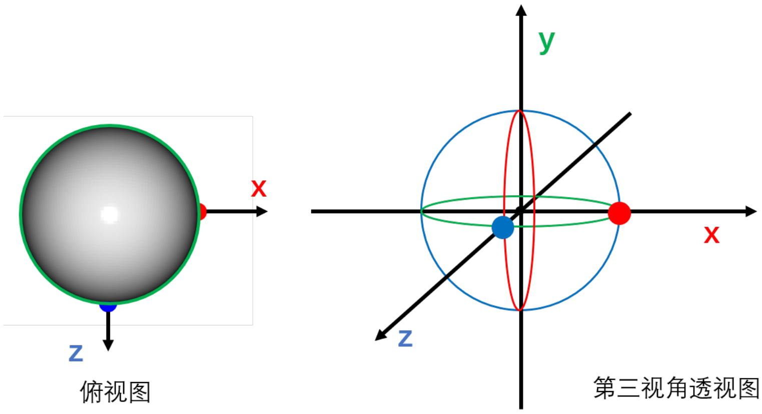
那么，Slerp(q_0, q_1, t) 怎么计算？Slerp 是在“圆弧上的插值”，如何对圆弧插值呢？我们转换将其成对角度的插值，这是合理的，因为角度和弧长有对应的关系。你可以尝试着自己去推导，这里直接给出结果：

$$q_t = \text{Serp}\,(q_0, q_1, t) = \frac{\sin[(1-t)\theta]}{\sin(\theta)} q_0 + \frac{\sin(t\theta)}{\sin(\theta)} q_1 \tag{2.6}$$

如果你推导有困难的话，可以参考一下附录的部分。

2.2.4.2 Double Cover

我们还是来看一下蓝色小球如何变化到红色小球，只不过我们这次更关注路径。



实际上我们在绿色圆环上，从蓝色小球移动到红色小球，有两条路径，如图2.16，

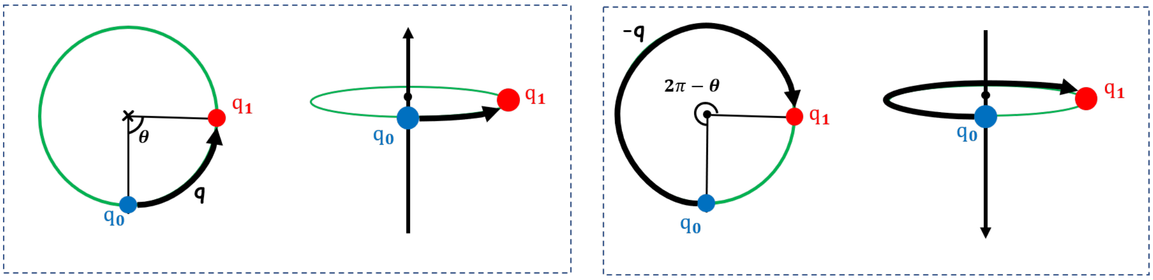


图 2.16: 注意 × 和 · 分别表示箭头穿出和射入。左右右图旋转轴方向是不同的，但都符合右手定则

我们从蓝色小球到红色小球，可以有两条路径，而且对应的旋转轴和角度不同。从代数形式上来看：

$$\begin{aligned} q &= \left[cos(\frac{1}{2}\theta), \sin(\frac{1}{2}\theta)\mathbf{n}\right] \\ -q &= \left[-\cos\left(\frac{1}{2}\theta\right), -\sin\left(\frac{1}{2}\theta\right)\mathbf{n}\right] \\ &= \left[\cos\left(\pi - \frac{1}{2}\theta\right), \sin\left(\pi - \frac{1}{2}\theta\right)(-\mathbf{n})\right] \end{aligned}$$

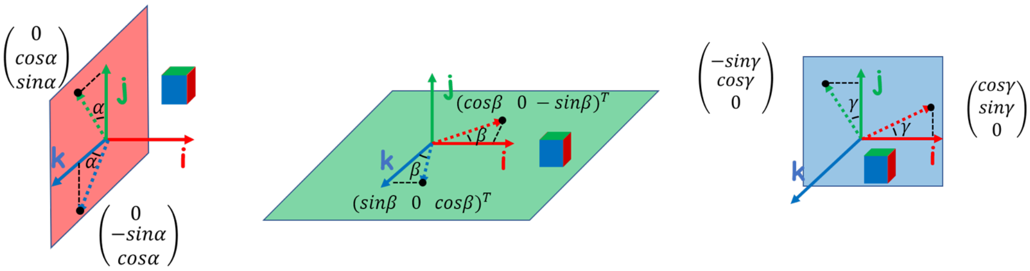
$$(-q)v(-q)^* = (-1)^2 qvq^* = qvq^*$$

换句换说，物理上一个旋转（从蓝色小球到红色小球），四元数有两种表达方式（q 和 $-q$）。

这种两个单位四元数，对应于同一个三维旋转现象被称为四元数对三维旋转的**双倍覆盖（Double Cover）**。

笔记 注意我们用一个 3 维的单位球对单位四元数做了很多的可视化和定性的分析，但是实际上单位四元数应该是位于 4 维球面上的，我们用 3 维球体可视化是一种迫不得已的行为。

我们研究单位球面上元素的好处在于，我们不需要作用对象就可以研究旋转本身。你一定还记得我们推导旋转矩阵的时候，都是作用于一个向量来得到旋转矩阵的。



如果我们只聚焦于几何上四维空间单位球面上的元素，我们就不需要作用对象（向量或者向量的四元数表示形式），能够更加专注于旋转本身，这更加接近于本质。

我们可以把这种几何上存在但我们无法想象的四维空间单位球，称为**流形（manifold）**。当然流形是一个很抽象的数学概念。五维球，或者六维一个很复杂的几何体，都可以称为流形。总之流形是一个几何概念，我们可以通过研究流形，得到位于这个流形上元素的种种性质。

2.2.5 总结

我们现在来稍微总结一下旋转变换的几种表示方式。绕过原点的轴 $\mathbf{n} = (n_x, n_y, n_z)^T$ 旋转 θ 的旋转变换可以表示为：

1. 行列式为 1 的正交矩阵
2. 单位四元数
3. 指数形式 $e^{\theta \mathbf{n}^\wedge}$（或者用旋转向量表示 $e^{\omega^\wedge}$）

笔记 欧拉角本质上还是矩阵表示，只不过把旋转矩阵分解成三个矩阵的乘积，便于我们理解和想象物体旋转变换后的样子，本质还是矩阵的表达形式。

矩阵相对比四元数直观一些，但是需要 9 个变量来存储。四元数只需要 4 个变量来表达，但是不够直观。

至于指数形式，看起来最简洁，它用旋转角度和旋转轴完全表征了旋转变换，只不过具体计算的时候还需要转换成矩阵才可以与向量运算。

2.2.5.1 数学形式之间的转换

他们之间如何转换呢？我们先来看从旋转矩阵提取轴角。

旋转轴我们只需求旋转矩阵特征值为 1 的特征向量即可。

$$\mathbf{Rn} = \mathbf{n}$$

旋转角度，可以用如下的方式提取：

$$\begin{aligned}
\mathbf{R} &= \cos\theta * \mathbf{I} + (1 - \cos\theta) * \mathbf{nn}^T + \sin\theta * \mathbf{n}^\wedge \\
tr(\mathbf{R}) &= \cos\theta * tr(\mathbf{I}) + (1 - \cos\theta) * tr(\mathbf{nn}^T) + \sin\theta * tr(\mathbf{n}^\wedge) \\
&= 3\cos\theta + (1 - \cos\theta) \\
&= 1 + 2\cos\theta
\end{aligned}$$

所以：

$$\theta = \arccos(\frac{tr(\mathbf{R}) - 1}{2})$$

至此通过轴角作为中间桥梁，可以将实现各种形式的转换。

为了之后讨论旋转方便，我们在这里给旋转矩阵和单位四元数一些记号：

$$SO(3) = \left\{\boldsymbol{R} \in \mathbb{R}^{3\times 3} \mid \boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{I}, \det(\boldsymbol{R}) = 1\right\}$$

$$S^3 = \{q \in \mathbb{H} \mid qq* = 1\}$$

这里 S^3 中的 3 是自由度的意思。

2.2.5.2 旋转的复合

假设对一个三维点 $\mathbf{v}$，先旋转 $e^{\theta_1 \mathbf{n_1}^\wedge}$，再旋转 $e^{\theta_2 \mathbf{n_2}^\wedge}$ 以后得到 $\mathbf{v}'$。

$SO(3)$ 表达：

$$\mathbf{v} = \mathbf{R}_{\mathbf{n_2}}(\theta_2)\mathbf{R}_{\mathbf{n_1}}(\theta_1)\mathbf{v}'$$

S^3 表达：

$$[0, \mathbf{v}'] = q_2 q_1 [0, \mathbf{v}] q_1^* q_2^*$$

其中 $q_1 = [cos(\frac{1}{2}\theta_1), \sin(\frac{1}{2}\theta_1)\mathbf{n_1}]$，$q_2 = [cos(\frac{1}{2}\theta_2), \sin(\frac{1}{2}\theta_2)\mathbf{n_2}]$。

记 $\mathbf{R_n}(\theta) = \mathbf{R}_{\mathbf{n_2}}(\theta_2)\mathbf{R}_{\mathbf{n_1}}(\theta_1) = e^{\theta_2 \mathbf{n_2}^\wedge} e^{\theta_1 \mathbf{n_1}^\wedge}$，我们看一种特殊情况：$\mathbf{n_1} = \mathbf{n_2}$，此时显然

$$\begin{aligned}\mathbf{R_n}(\theta = \boldsymbol{\theta_2} + \boldsymbol{\theta_1}) &= \mathbf{R_n}(\theta_2)\mathbf{R_n}(\theta_1) = e^{\theta_2 \mathbf{n}^\wedge} e^{\theta_1 \mathbf{n}^\wedge} \\ &= \mathbf{R_n}(\theta_1)\mathbf{R_n}(\theta_2) = e^{\theta_1 \mathbf{n}^\wedge} e^{\theta_2 \mathbf{n}^\wedge}\end{aligned}$$

因为绕同一个轴先旋转 θ_1，后旋转 θ_2 与先旋转 θ_2，后旋转 θ_1 没有区别。

我们再来看这样一个有一些愚蠢的问题，“为什么复合的时候是乘法，而不是加法除法平方等其他运算？”，这个问题似乎就像在 C++ 里面，“拼接字符串为什么不把两个字符串乘起来？”一样让人哭笑不得。

但实际上它说明在很多时候我们研究一个元素所属于的集合是不够的，**集合常常需要与某种运算结合在一起看才有实际意义。**

我们来回忆大一高数上不定积分：存在这样一些函数，他们的存在原函数，但是我们无法用初等函数表示其原函数，比如 $\int e^2 dx, \int \frac{dx}{lnx}, \int \frac{\sin(x)}{x}$。

这种现象本质的原因是**不定积分运算对于基本初等函数不封闭**。也就是说在一些情况下，不定积分作用于基本初等函数，会导致函数不在基本初等函数的集合里，所以我们也就无法用初等函数表示其原函数了。

再比如对 2 做开方运算，得到 $\sqrt{2}$，就变成了无理数。即：开方运算对有理数不封闭。

现在我们来看解释为什么“为什么旋转矩阵复合的时候是乘法，而不是加法除法平方等其他运算？”

$$\begin{pmatrix} cos\alpha & -\sin\alpha \\ \sin\alpha & cos\alpha \end{pmatrix} + \begin{pmatrix} cos\beta & -\sin\beta \\ \sin\beta & cos\beta \end{pmatrix}$$

$$\begin{pmatrix} cos\alpha & -\sin\alpha \\ \sin\alpha & cos\alpha \end{pmatrix} * \begin{pmatrix} cos\beta & -\sin\beta \\ \sin\beta & cos\beta \end{pmatrix}$$

不言自明，加法对旋转矩阵不封闭。所以我们复合的时候用乘法，也就是说乘法对于旋转矩阵才是有意义的，我们应该将旋转矩阵与乘法紧密的联系在一起。

通过这种思想，我们可以理解为什么旋转矩阵必须是行列式为 1 的正交矩阵，而不能行列式为-1。(因为若为-1，奇数个矩阵相乘与偶数个矩阵相乘的结果行列式不同，违反了封闭性。)

笔记 实际上，一个集合与一种二元运算被统一称作 **代数结构（algebraic structure）**。根据这个二元运算满不满足封闭性，结合律，交换律等等性质，具体又将代数结构分成了群、环、域等等。

你可以思考旋转矩阵与乘法这种代数结构，具备什么样的性质？留给读者思考，之后我们会深入来讲解，旋转矩阵和乘法这种代数结构的性质。

2.3 其他三维变换

到现在我们主要讲了两种三维变换，平移和旋转

$$\mathbb{R}^3 = \left\{\mathbf{t} \in \mathbb{R}^3 \mid \mathbf{t} - \mathbf{t} = 0\right\}$$

$$SO(3) = \left\{\boldsymbol{R} \in \mathbb{R}^{3\times 3} \mid \boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{I}, \det(\boldsymbol{R}) = 1\right\}$$

相比于其他变换，只有旋转变换除了矩阵以外，还有单位四元数 S^3 的表达，我们为了各种变换形式上的统一，统一用矩阵来表示变换。所以我们把平移变换 $\mathbf{v} + \mathbf{t}$ 也用矩阵齐次坐标的形式表达出来:

$$\begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} v_x \\ v_y \\ v_z \\ 1 \end{pmatrix}$$

如果把旋转和平移变换 $\mathbf{R}\mathbf{v} + \mathbf{t}$ 用矩阵齐次坐标表达：

$$\begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix}\begin{pmatrix} v_x \\ v_y \\ v_z \\ 1 \end{pmatrix}$$

笔记 $\begin{pmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix}$ 是一个 4×4 的矩阵，加粗的 $\mathbf{0}$ 不是标量，是一个 3×1 的列向量。

上式可以分块矩阵相乘写成一个矩阵：

$$\begin{aligned}\begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix}\begin{pmatrix} v_x \\ v_y \\ v_z \\ 1 \end{pmatrix} &= \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix}\begin{pmatrix} v_x \\ v_y \\ v_z \\ 1 \end{pmatrix} \\ &\triangleq \mathbf{T}\begin{pmatrix} v_x \\ v_y \\ v_z \\ 1 \end{pmatrix}\end{aligned}$$

可以发现使用齐次坐标可以把旋转变换和平移变换写到一个矩阵 $SE(3)$ 里面：

$$SE(3) = \left\{\boldsymbol{T} = \begin{pmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \in \mathbb{R}^{4\times 4} \mid \boldsymbol{R} \in SO(3), \boldsymbol{t} \in \mathbb{R}^3\right\} \tag{2.7}$$

不言自明，$SE(3)$ 是对向量先旋转再平移 $\mathbf{R}\mathbf{v} + \mathbf{t}$，当然你也可以先平移再旋转 $\mathbf{R}(\mathbf{v} + \mathbf{t})$)，结果会稍微不同，留给读者在习题完成。

我们来看一下缩放变换，如图2.17所示，很容易理解。

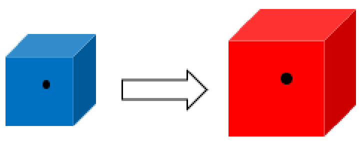


图 2.17: 缩放变换

我们只需要对立方体上每一个点 $\mathbf{v_i}$ 乘以一个标量 s 作为缩放因子，即 $s\mathbf{v_i}$ 就可以得到缩放后的点坐标。

写成矩阵的形式：

$$\begin{pmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{v_i} = \begin{pmatrix} s\mathbf{E} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \mathbf{v_i} \tag{2.8}$$

笔记 如果缩放矩阵中的对角线上的 s 不一样话，正方体经过缩放变换后就不是正方体了，可能会变成长方体。

我们如果把缩放和 $SE(3)$ 放在一起会如何呢？思考一下图2.18 中右侧的变换应该是怎样呢？

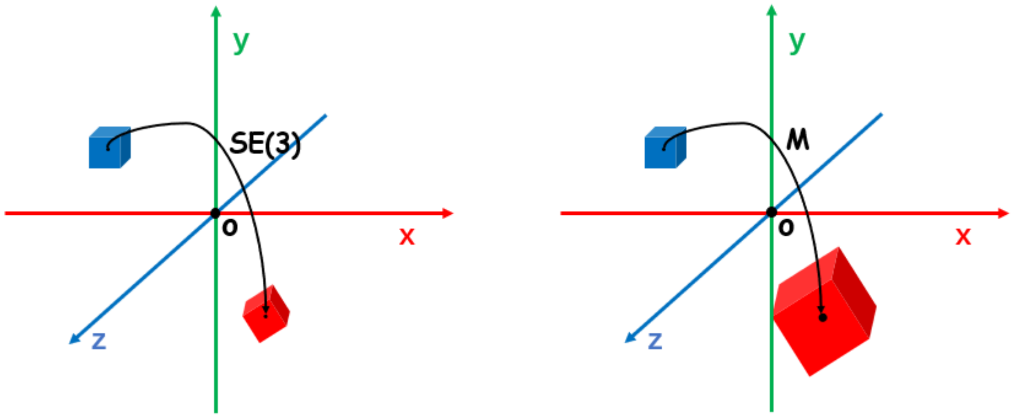


图 2.18: $SE(3)$ 与缩放变换结合

这里用颜色区分点 $\mathbf{v}$ 和边长 $\mathbf{L}$ 是描述蓝色方块还是红色方块。

我们来稍微看一下，我们取蓝色立方体的两个点 $\mathbf{v_1}, \mathbf{v_2}$，令 $\mathbf{L} = \mathbf{v_1} - \mathbf{v_2}$，我们取模长 $\|\mathbf{L}\|$ 即可得到蓝色小球的边长。

我们先来看图2.18左侧的变换，即 $SE(3)$，经过 $SE(3)$ 那么立方体的边长显然是不变的。用数学来表述的话就是：

$$\begin{aligned} \|\mathbf{L}\| &= \left\| \begin{matrix} \boldsymbol{R} & \boldsymbol{t} \\ \mathbf{0}^T & 1 \end{matrix} \right\| \|\mathbf{L}\| \\ &= \|\mathbf{L}\| \end{aligned}$$

我们来看图2.18中右侧的变换 $\mathbf{M}$，显然 $\mathbf{M}$ 包含着一个缩放变换，我们记这个缩放为 s，即 $\mathbf{v} = s\mathbf{v}$，于此同时显然有：

$$\|\mathbf{L}\| = s\|\mathbf{L}\| \tag{2.9}$$

$\mathbf{M}$ 首先应该包含 $SE(3)$ 同时还要满足式2.9。请读者思考一下 $\mathbf{M}$ 应该怎么组成。

$$\mathbf{M}\mathbf{v_1} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} s\mathbf{E} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \mathbf{v_1} = \begin{pmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \mathbf{v_1}$$

笔记 为什么这里不先 $SE(3)$，在缩放？而是先缩放后 $SE(3)$？留给读者习题中完成。

我们给 $\mathbf{M}$ 这一类矩阵做一个记号：

$$Sim(3) = \left\{ \begin{pmatrix} s\boldsymbol{R} & \boldsymbol{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \in \mathbb{R}^{4\times 4} \mid \boldsymbol{R} \in SO(3), \boldsymbol{t} \in \mathbb{R}^3, s \in \mathbb{R}^1 \right\} \tag{2.10}$$

2.4 习题

1. 证明 n 阶的旋转矩阵 $R_{n\times n}$ 是正交矩阵。

2. 我们已经知道在基 i, j 下旋转变换为 $\begin{pmatrix} cos\theta & -\sin\theta \\ \sin\theta & cos\theta \end{pmatrix}$，但是描述图像的坐标系有点不太一样如图2.19，我们用齐次坐标 $(u, v, 1)^T$ 表示旋转前图像的中一个像素的位置。

一个 3x3 的矩阵 M 表示绕图像中心 $(C_x, C_y)^T$ 逆时针旋转 θ 的变换，如图2.19，求 M 矩阵。

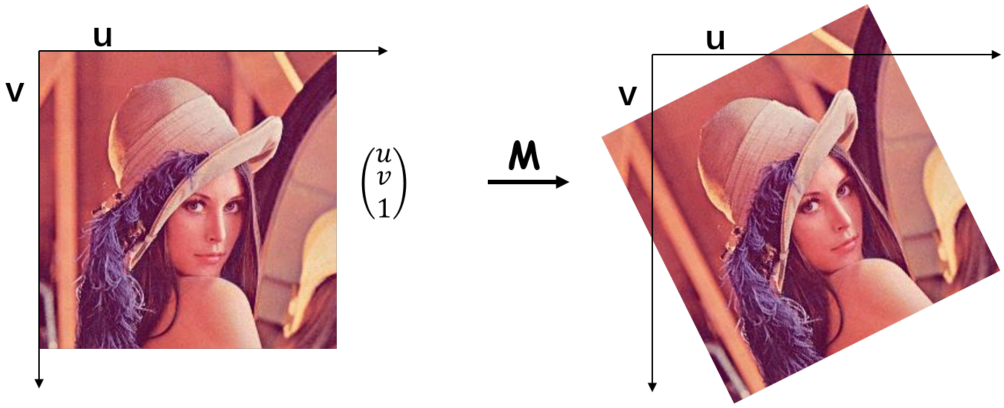


图 2.19: 像素坐标系下绕图像中心点旋转的矩阵 M

3. 三维点 $\mathbf{v}$ 绕过原点的旋转轴 $\mathbf{n}(\|n\| = 1)$ 旋转 θ 之后得到 $\mathbf{v}'$。令 $v = [0, \mathbf{v}], q = [cos\frac{\theta}{2}, (\sin\frac{\theta}{2})\mathbf{n}]$。证明：

$$qvq^* = [0, cos\theta\mathbf{v} + (1 - cos\theta)(\mathbf{n}\cdot\mathbf{v})\mathbf{n} + \sin\theta(\mathbf{n}\times\mathbf{v})]$$

可能用到的公式：$\mathbf{a}\times(\mathbf{b}\times\mathbf{c}) = (\mathbf{a}\cdot\mathbf{c})\mathbf{b} - (\mathbf{a}\cdot\mathbf{b})\mathbf{c}$

4. 完成插值 Lerp, NLerp, Slerp

（可以用一个三维模型代替一下，结果是鹦鹉螺那种东西）

5. 推导 $\mathbf{R}(\mathbf{v}+\mathbf{t})$ 的矩阵表达形式，并与 $SE(3)$ 先旋转再平移 $\mathbf{R}\mathbf{v}+\mathbf{t}$ 对比。

6. 推导 $s[\mathbf{R}\mathbf{v}+\mathbf{t}]$ 的矩阵表达形式，并与 $Sim(3)$ 先缩放再 $SE(3)$，$\mathbf{R}(s\mathbf{v})+\mathbf{t}$ 对比。

第 3 章 相机与成像

之前我们花了很大的篇幅来讲解一个三维点在空间中的变换，我们还没有讲如何获取三维数据。因为我们讲的是 3D 视觉，所以我们先来看一下如何相机如何成像，即一个三维点如何变成图片上的一个点。

3.1 针孔相机模型

假设我们有一个相机，只不过这个相机没有透镜，我们将透镜简化成一个极小的小孔 o，小孔 o 称作光心。在此基础上我们建立坐标系。如图3.1

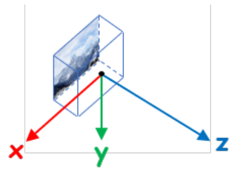


图 3.1: 简陋的相机。雪山的图片来源于Unsplash

笔记 你可能注意到这里 xyz 轴的建立和我们之前讲的有些许不同，但实际上手性还是一致的，因为 $\mathbf{x} \times \mathbf{y} = \mathbf{z}$。即使手性变了，我们也可以很方便的利用线性变换的矩阵定理2.1得到各种变换的关系。

之所以在光心所在平面这样建系只是一种约定俗称的规则而已。

我们来把这个简陋的相机放大看一下：

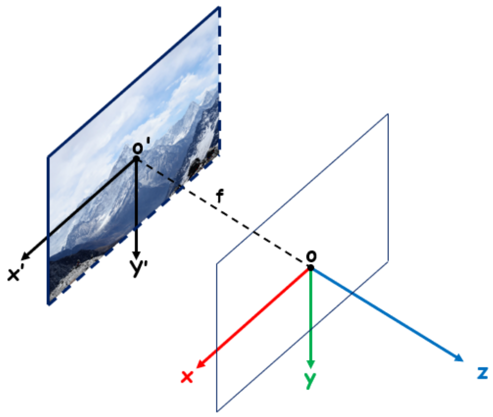


图 3.2: 针孔相机内部放大图

平面 $o-x-y$ 与平面 $o'-x'-y'$ 之间的距离就是焦距 f，是相机的固有参数。

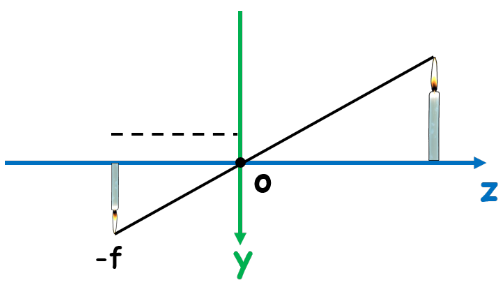


图 3.3: 真实成像平面。蜡烛的原始图片来源于Unsplash，下同

图3.2看起来并没有什么问题，但是稍微回忆一下中学时学的小孔成像原理，就会发现实际成像应该是倒相，如图3.3。

所以把图片3.2 上下左右翻转一下，如图3.4，才是真实的成像。当然实际照相的时候得到的图片肯定不是反的，这是由于相机在成像之后帮我们做了翻转。

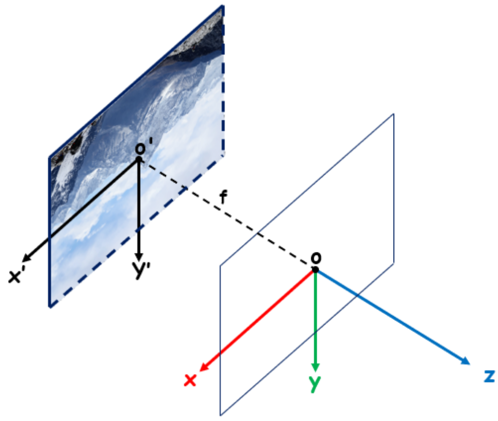


图 3.4: “大山”上下左右翻转---成像为倒像，才是符合小孔成像的相机

我们现在来看真实世界中的一个点 p 如何投影成二维上的一个点 p'，如图3.5。

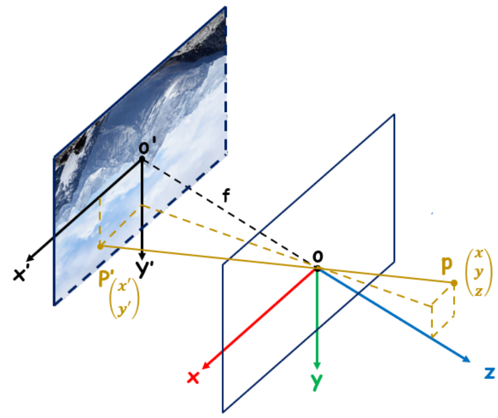


图 3.5: 3d 点投影到平面得到 2d 点

通过图3.6，图3.7的相似关系可得：

$$\frac{z}{f} = -\frac{x}{x'} = -\frac{y}{y'}$$

笔记 这里负号的实际物理意义也表明了成像是倒像。

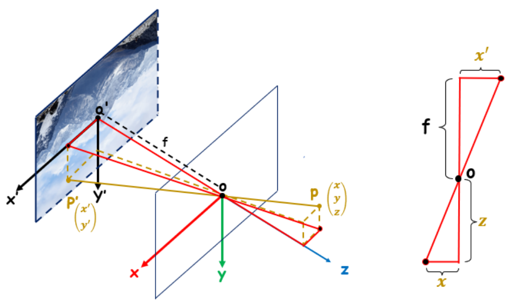


图 3.6: x 方向相似三角形

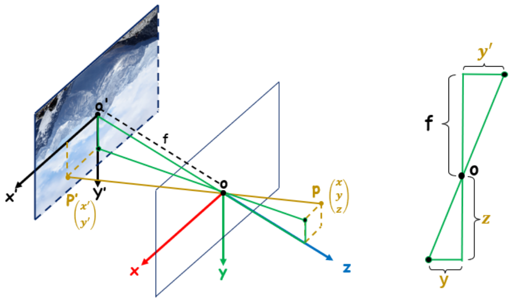


图 3.7: y 方向相似三角形

我们可以把成像平面放到相机前方，如图3.8，就可以把负号去掉：

$$\frac{z}{f} = \frac{x}{x'} = \frac{y}{y'} \tag{3.1}$$

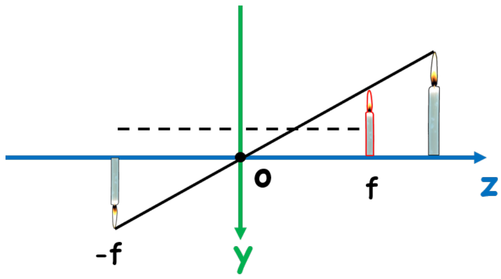


图 3.8: 对称的成像平面

通常图片遍历像素的坐标不同于 $o' - x' - y'$，而是采用 $u - v$ 坐标。所以我们需要一个转换，这个转换是一个缩放和平移，如图3.9。x' 方向缩放 α，y' 方向缩放 β。我们可以利用下式将 P' 转换成 $u - v$ 坐标系下的坐标：

$$\begin{cases} u = \alpha x' + c_x \\ v = \beta y' + c_y \end{cases} \tag{3.2}$$

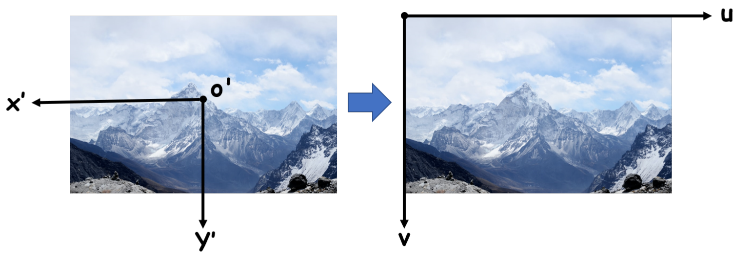


图 3.9: 成像平面下的坐标转换到像素坐标系下的坐标

带入公式3.1，可以得到

$$\begin{cases} u = \alpha f \frac{x}{z} + c_x \triangleq f_x \frac{x}{z} + c_x \\ v = \beta f \frac{y}{z} + c_y \triangleq f_y \frac{y}{z} + c_y \end{cases} \tag{3.3}$$

方便起见将 αf 记作 f_x，βf 记作 f_y，符号 $\triangleq$ 是定义的意思。

笔记 $o' - x' - y'$ 描述的是连续坐标，必须转化成遍历图片离散的 $u - v$ 坐标。所以：

- f 的单位是米。
- α 表示水品方向 1m 的长度包含的像素个数，β 表示竖直方向 1m 的长度包含的像素个数。

$f_x = f\alpha, f_y = f\beta$ 单位就是像素个数。之所以要给两个量 α, β，是因为有时候由于工艺的问题，像素并不是完完全全的矩形。通常情况下我们可以认为 $\alpha == \beta$。

写成矩阵的形式，会看起来简洁一些：

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{1}{z} \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \triangleq \frac{1}{z}\mathbf{KP}$$

按照习惯通常把 z 移到等式左侧：

$$z\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \triangleq \mathbf{KP} \tag{3.4}$$

K 被称作相机的**内参矩阵**（Intrinsics），你可以看到这个矩阵里面的参数都是相机内部的参数，一旦相机制作出来，内参矩阵就确定了。

我们来稍微总结一下相机内参矩阵的物理意义：**内参矩阵 K 描述了相机内部是如何采样的，是一个连续到离散的过程，决定了我们相机硬件层面得到的图片分辨率。**如图3.10。

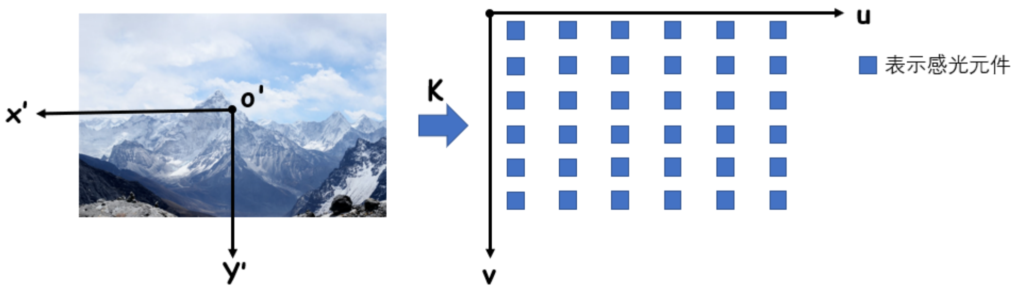


图 3.10: $o'-x'-y'$ 是连续坐标，因为小孔成像的原因，也就是说“光成像的位置”在这个平面，但是我们要采集这个平面上的“光”，必须要使用很多感光元件（比如 CCD 或者 CMOS），比如 640*480 的图片，就是有 640*480 个感光元件，这就会导致离散化的问题。

如果想得到感光元件的尺寸的话，即一个像素实际的物理有多大，可以用 $\frac{1}{\alpha}\times\frac{1}{\beta}$，单位是平方米。

有内参矩阵，对应的也有相机**外参矩阵**。看图3.11，假设我们相机发生了经过了旋转和平移的变换，不同相机视角下，我们对同一个空间点的描述发生了变化（因为基 x-y-z 发生了变化）。

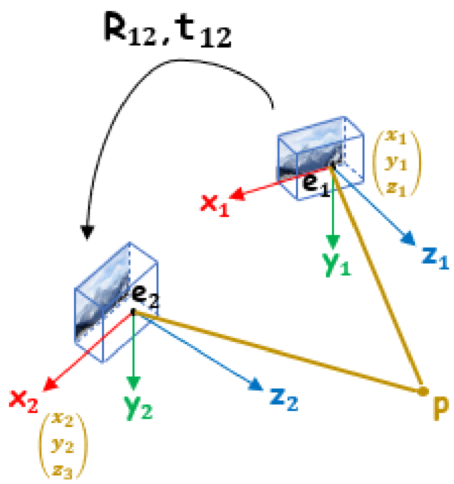


图 3.11: 同一个空间点在不同相机视角下的三维坐标不同，点 P 在坐标系 $\mathbf{e_1}-\mathbf{x_1}-\mathbf{y_1}-\mathbf{z_1}$ 下的坐标为 $(x_1, y_1, z_1)^T$

为了对三维点 **P** 有一个统一的描述，我们需要引入**世界坐标系**，如图3.12。在世界坐标系下的对点 p 的描述是 $(x_w, y_w, z_w)^T$

通过外参矩阵我们可以实现同一个三维点在某一个相机坐标系下的描述与在世界坐标系下的描述的一个转换。即式3.5。

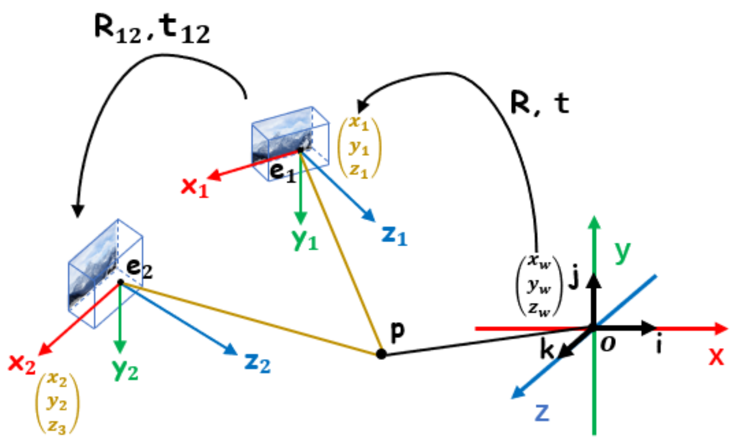


图 3.12: 相机外参 $\mathbf{R}, \mathbf{t}$

$$\mathbf{P} = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = \mathbf{R} \begin{pmatrix} x_w \\ y_w \\ z_w \end{pmatrix} + \mathbf{t} \triangleq \mathbf{R}\mathbf{P_w} + \mathbf{t} \tag{3.5}$$

如果 $\mathbf{P}$ 与 $\mathbf{P_w}$ 采用齐次坐标：

$$\begin{pmatrix} \mathbf{R}_{3\times3} & \mathbf{t}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{pmatrix}_{4\times4} \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix} \triangleq \mathbf{T}\mathbf{P_w}$$

同样的，我们来稍微总结一下外参的物理意义：**内参矩阵是二维的一种变换，与相机的位姿（相机的位置和朝向）没有关系。而外参矩阵则刻画了相机在世界坐标系中的位姿。**

即：

$$\mathbf{T}^{-1} = \begin{pmatrix} \mathbf{R}^{-1} & -\mathbf{R}^{-1}\mathbf{t} \\ \mathbf{0}_{1\times3} & 1 \end{pmatrix}_{4\times4} = \begin{pmatrix} \mathbf{x_1} & \mathbf{y_1} & \mathbf{z_1} & \mathbf{e}_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}_{4\times4} \tag{3.6}$$

笔记 式3.6中的 $\mathbf{e_1}, \mathbf{x_1}, \mathbf{y_1}, \mathbf{z_1}$ 都是三维列向量，表述了相机在世界坐标系中的位姿。此外如果你对式3.6有任何疑问，可以回顾一下公式1.3与公式 1.2，并参考附录分块矩阵求逆的部分。

至此我们可以将相机外参---式3.5与相机内参---式3.4结合，就可以得到：

$$z\mathbf{P_{uv}} = z\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{KP} = \mathbf{K}\left(\mathbf{R}\mathbf{P_w} + \mathbf{t}\right) \tag{3.7}$$

若采用齐次形式表示：

$$\begin{aligned} z\mathbf{P_{uv}} = z\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{KP} = \begin{pmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{R}_{3\times3} & \mathbf{t}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{pmatrix}_{4\times4} \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix} \\ \triangleq \mathbf{K}\left[\mathbf{R} \mid \mathbf{t}\right]\mathbf{P_w} \\ \triangleq \mathbf{KT}\mathbf{P_w} \end{aligned} \tag{3.8}$$

笔记 注意到虽然字母一样，但是 $\mathbf{K}, \mathbf{P_w}$ 在式3.7与式3.8中是不同的形式。这取决于齐次还是非齐次，从而内参矩阵的形式会变成 3x3 或者 3x4。之后我们不特别说明 $\mathbf{K}, \mathbf{P_w}$ 采用哪一种形式。在不同的公式中，哪种形式说的通

就采用哪一种。

请完成习题 1。

3.1.1 孔径大小对成像的影响

针孔太大的话，真实世界的多个物体的光线会投影到像素平面的同一个位置，从而导致模糊。如图3.13

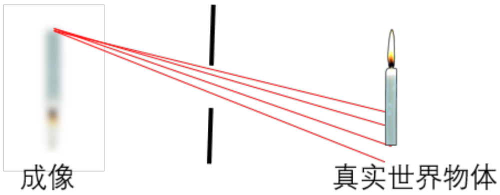


图 3.13: 针孔太大，会导致模糊

针孔太小的话，又会导致射入的光线太少，接受到的能量太少，成像的图片就会变得很暗。

孔径不能太大也不能太小，到底孔径多大合适似乎不太好定义，为了解决这个问题，通常会选择在小孔前增加透镜。

增加透镜以后，投影关系还是与小孔成像一致的，只不过，成像的同一个位置会来自更多的光线，而非理想小孔成像那样只有一根光线，这样就缓解了小孔较小时亮度不够的问题。

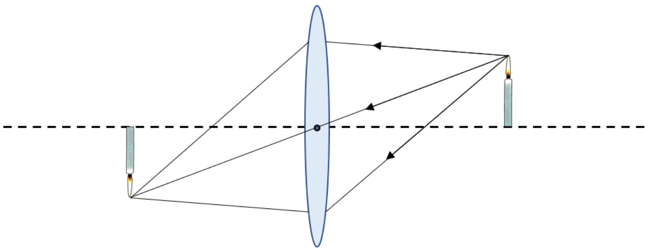


图 3.14: 凸透镜会汇聚光线

笔记 实际的镜头往往非常复杂，会有多组透镜组成，最终整体呈现的话也不一定是凸性的透镜。

3.1.2 畸变

3.2 RGB-D 相机

3.2.1 RGB alignment depth

3.3 习题

1. 之前做的 20 次迭代的投影变换，生成图像，并说明生成图像的物理意义。2. 相机标定，程序联系

第 4 章 三维物体的表示

之前我们提到了如何获取

4.1 点云

点云是用一些离散的三维点 $\mathbf{p_i} \in \mathbb{R}^3, i = 1, 2, ..., n$ 来表示物体。

4.2 体素

体素是像素在三维空间中的推广。其用一系列的立方体来表示三维物体。这种表达方式的好处在于比较“规整”，缺点在于分辨率（立方体的大小）与内存占用存在矛盾，体素分辨率越高，表达的物体会更加逼真细致，但是内存占用也就会越大。

4.3 Mesh

Mesh 是空间中的三角形通过一系列方式组合在一起。在图形学中得到了非常广泛的应用。非常适合于物体的拓扑变换。

附录 A 基本数学工具

A.1 范数

范数 $\|x\|$ 是一种映射：$\mathbb{R}^n \mapsto \mathbb{R}$ 这种映射需要有满足以下三种关系：

1. $v \in \mathbb{R}^n, \|v\| \geq 0$。$\|v\| \geq 0$ 当且仅当 $v = 0$
2. $\alpha \in \mathbb{R}, v \in \mathbb{R}^n, \|\alpha v\| = |\alpha|\|v\|$
3. $v \in \mathbb{R}^n, \omega \in \mathbb{R}^n, \|v + \omega\| \leqslant \|v\| + \|\omega\|$

A.1.1 向量范数

向量 $\mathbf{v} = (v_1, v_2, ..., v_n)^T$ 的 L_p 范数定义为:

$$\|\mathbf{v}\|_p = (|v1|^p + |v2|^p + ... + |v_n|^p)^{\frac{1}{p}} \quad (p \geq 1) \tag{A.1}$$

通常使用的是 1 范数 $\|\mathbf{v}\|_1$，2 范数 $\|\mathbf{v}\|_2$，一般不加下标，默认是 2 范数 $\|\mathbf{v}\| = \|\mathbf{v}\|_2$。有些会使用无穷范数

$$\|\mathbf{v}\|_\infty = max|v_i|$$

定理 A.1 (柯西不等式)

向量 $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$，$|\mathbf{a}^T\mathbf{b}| \leq \|\mathbf{a}\|_2\|\mathbf{b}\|_2$，取等号当且仅当 $\mathbf{a} = \lambda\mathbf{b}(\lambda \neq 0)$。

A.1.2 矩阵范数

A.1.2.1 Frobenius 范数

类比向量范数，进行推广，一个矩阵 A，$\mathbf{A} \in \mathbb{R}^{m\times n}$，

- l_1：$\|\mathbf{A}\|_1 = \sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|$
- l_2，即 F 范数：$\|\mathbf{A}\|_2 = \|\mathbf{A}\|_F = \sqrt{tr(\mathbf{A}\mathbf{A}^T)} = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}a_{ij}^2}$

推论 A.1 (F 范数对正交矩阵不变性)

正交矩阵 $\mathbf{U} \in \mathbb{R}^{m\times m}$ 和正交矩阵 $\mathbf{V} \in \mathbb{R}^{n\times n}$，$\mathbf{A} \in \mathbb{R}^{m\times n}$，则

$$\|\mathbf{UAV}\|_F = \|\mathbf{A}\|_F$$

证明也很简单，因为正交矩阵转置等于逆，以及 $tr(AB) = tr(BA)$ 所以

$$\begin{aligned}
\|\mathbf{UAV}\|_F^2 &= tr\left(\mathbf{UAV}(\mathbf{UAV})^\mathbf{T}\right) \\
&= tr\left(\mathbf{UAVV^TA^TU^T}\right) \\
&= tr\left(\mathbf{UAA^TU^T}\right) \\
&= tr\left(\mathbf{A^TU^TUA}\right) \\
&= tr\left(\mathbf{A^TA}\right) \\
&= \|\mathbf{A}\|_F^2
\end{aligned}$$

A.2 自由度 (Degree Of Freedom)

自由度 (dof) 描述了一个系统里所有变量“独特”的个数。这里独特的意思是说独立，不相关的意思。来看几个简单的例子。

1. $\begin{pmatrix} cos\theta & -\sin\theta \\ \sin\theta & cos\theta \end{pmatrix}$ 显然这个矩阵 dof 是 1，因为每个元素都与 θ 有关。

2. $\begin{pmatrix} a & c \\ d & b \end{pmatrix}$ （a,b,c,d 均不等于 0，且不相关）这个矩阵 dof 就是 4。如果说这个矩阵的行列式为常数 k 的话，那么 dof 就会变成 3。也就是说因为 $ab - cd = k$，a，b，c，d 中任意三个都可以表示剩余的一个。比如 $a = (k + cd)/b$。也就是说加入一个约束通常自由度会降低。但是不要不理智的乱加约束，比如 A 是三阶方阵，且 $rank(A) = 2$ 然后你加了两个约束说，一个是 $rank(A) = 2$，一个是行列式为 0。然后你说 A 的自由度少了两个。非常无语。。。，显然这两个约束是耦合的。

3. 再看这么一个例子：

$\bar{X}$ 已知，那么 $\sum_{i=1}^{n}(X_i - \bar{X})^2$ 的 dof 应该是多少呢？

应该是 n-1，因为 $\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n}$ 其实就相当与一个约束，X_j 可以由 $X_i, (i = 1, 2, ..., j-1, j+1, ..., n)$ 来表示。这也就是样本方差 $s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2$ 为什么前面除以的归一化系数为什么是 n-1，而不是 n。

4. 矩阵的自由度和秩有一定的关系，但是又不完全一致。看这么一个例子：

已知三阶方阵 B 的秩 $rank(B) = 2$ 且 B 可以表示为 $B = K'TK$，其中 K', K 均为三阶 4d0f 可逆矩阵，T 是 6dof 的三阶矩阵，且存在非 0 的三维列向量 V', V，使得 B 满足 $V'^{T}BV = 0$，那么 B 的自由度是多少呢？

应该是 $4+4+6 = 14$ 个自由度。恭喜你打错了。B 一共就有 9 个元素，自由度怎么会大于 9 呢？自由度描述的是独特的个数，K, K' 与 T 相乘会有可能使得 T 中的独特元素消失掉，使得 B 的自由度小于等于 T。这样看起来似乎 $dof(B) <= dof(T) = 6$。但是别忘了，B 还需要满足秩为 2，且 $V'BV$。而且秩为 2 与我们讲的 K, K' 对自由度的影响是有耦合的。现在这样似乎越来越晕了。

别急，我们来简单梳理一下。给定一个三阶方阵 B，那么 $dof(B) <= 9$。加上 $r(B) = 2$ 的约束以后呢？B 的自由度会减少多少呢？减少 1，而不是 2。原因如下

$$B = \begin{pmatrix} b_1 & b_2 & b_3 \\ b_4 & b_5 & b_6 \\ b_7 & b_8 & b_9 \end{pmatrix} \xrightarrow[\text{向量}]{\text{按列分为}} (\beta_1 \ \ \beta_2 \ \ \beta_3) \xrightarrow[\text{初等变换}]{r(B)=2} (\beta_1 \ \ \beta_2 \ \ 0)$$

可以看到秩为 2，使得 $\lambda_1\beta_1 + \lambda_2\beta_2 = \beta_3$。也就是 b_3 可以由 $k_1b_1 + k_2b_2$ 表示，$l_1b_4 + l_2b_5$ 看起来貌似 b_3, b_6, b_9 被其他 6 个元素表示了。但是这种表示本身就是有约束的 $k_1 = l_1, k_2 = l_2$。

所以自由度减少的本质是类似例子 2 中形式。一个变量有其他变量表示。

- 行列式为 0，就是九个变量加减乘除运算以后为 0，一定有一个变量可以有其余八个变量表示，自由度减少 1。
- $V'BV$ 是一个标量，这个标量九个变量加减乘除数乘运算得到的，为 0，一定有一个变量可以有其余八个变量表示，自由度减少 1。

约束不存在耦合，所以 B 自由度为 $9 - 1 - 1 = 7$

A.3 分块矩阵的逆

注意 A，B，C 都是矩阵，0 也是矩阵，而不是一个标量。下面这两个上 (下) 三角矩阵如果可逆，那么行列式必定不为 0，即 A 的行列式与 B 的行列式的乘积也一定不能为 0。也就是说矩阵如果可逆，A，B 也一定可逆。

$$\begin{pmatrix} A & C \\ 0 & B \end{pmatrix}^{-1} = \begin{pmatrix} A^{-1} & -A^{-1}CB^{-1} \\ 0 & B^{-1} \end{pmatrix}$$

$$\begin{pmatrix} A & 0 \\ C & B \end{pmatrix}^{-1} = \begin{pmatrix} A^{-1} & 0 \\ -B^{-1}CA^{-1} & B^{-1} \end{pmatrix}$$

这里逆矩阵是如何求得的呢？我们以第一个为例。通常我们求解 M 的逆矩阵是这么求的 $(M|E)->(E|M^{-1})$，把我们想求逆的矩阵写成 $(M|E)$ 然后对 M 经过一堆猛烈的行变换化成 E，之后 E 的位置自然而然就是 M^{-1}。

原理其实是 AM 相当于对 M 进行初等行变换，MA 相当于对 M 进行初等列变换。我们将 $(M|E)$ 进性初等行变换化成 $(E|?)$，其实就相当于 $A(M|E)->(E|A)$，显然 A 就是 M^{-1}。之所以写成将 M 写成 $(M|E)$ 其实也就是通过 E 来记录这个 A 矩阵罢了。

所以也就不难得出：

$$\left(\begin{array}{cc|cc} A & C & E & 0 \\ 0 & B & 0 & E \end{array}\right) \to \left(\begin{array}{cc|cc} E & A^{-1}C & A^{-1} & 0 \\ 0 & E & 0 & B^{-1} \end{array}\right) \to \left(\begin{array}{cc|cc} E & 0 & A^{-1} & -A^{-1}CB^{-1} \\ 0 & E & 0 & B^{-1} \end{array}\right)$$

附录 B 公式推导补充

B.1 罗德里格斯公式

B.1.1 推导

假设一个点 $\mathbf{v}$，绕过原点的旋转轴 $\mathbf{n} = (n_x, n_y, n_z)$（$\|\mathbf{n}\| = 1$），旋转 θ 后得到点 $\mathbf{v}'$。

笔记 之后的推导会用到向量的叉积，如果你不熟悉的话，请到本书的首章复习一下。

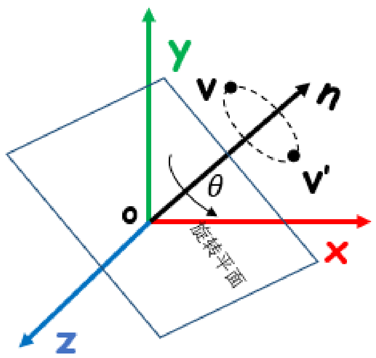


图 B.1: 旋转平面

我们可以将 $\mathbf{n}$ 当作一个平面的法向量，得到旋转平面的方程：$n_x x + n_y y + n_z z = 0$。

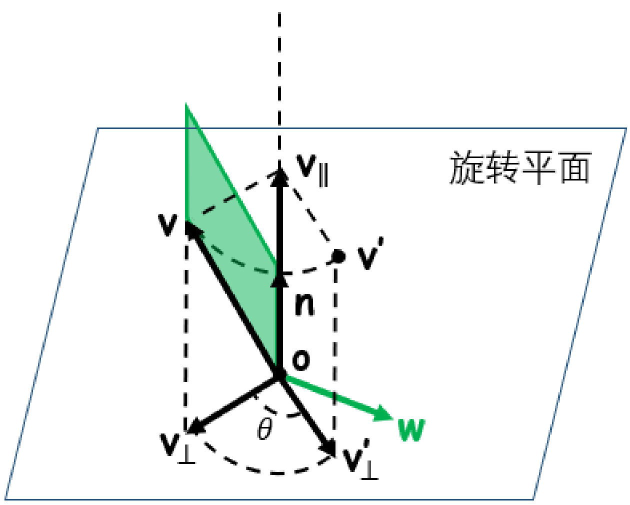


图 B.2: 罗德里格斯公式

为了直观一点，我们换个角度来看，如图B.2，将向量 $\mathbf{v} = \mathbf{v}_{\|} + \mathbf{v}_{\perp}$ 分解为：

- 平行于旋转轴的分量 $\mathbf{v}_{\perp}$
- 垂直于旋转轴（即 $\mathbf{v}$ 在旋转平面的投影）的分量 $\mathbf{v}_{\|}$

我们现在用 $\mathbf{v}$，$\mathbf{n}$ 来表示 $\mathbf{v}_{\perp}$，为此我们引入一个向量 $\mathbf{w} = \mathbf{n} \times \mathbf{v}$，$\mathbf{w}$ 的方向由右手法则确定，$\|w\|$ 就是图中

绿色平行四边形的面积 S_{green}。又因为平行四边形面积等于底乘以高，即：

$$
\begin{aligned}
\|w\| &= \|\mathbf{n} \times \mathbf{v}\| \\
&= S_{green} \\
&= \|n\| \cdot \|\mathbf{v}_\perp\| = \|\mathbf{v}_\perp\|
\end{aligned}
$$

所以也就不难得出：

- $\mathbf{v}_\perp = -\mathbf{n} \times \mathbf{w} = -\mathbf{n} \times (\mathbf{n} \times \mathbf{v})$
- $\mathbf{v}_\| = \mathbf{v} - \mathbf{v}_\perp = \mathbf{v} + \mathbf{n} \times (\mathbf{n} \times \mathbf{v})$

笔记 因为叉积不满足结合律，这里不要随便去掉括号。

$\mathbf{v}' = \mathbf{v}_\| + \mathbf{v}'_\perp$，我们已经得到了 $\mathbf{v}_\|$，再求得 $\mathbf{v}'_\perp$ 即可。

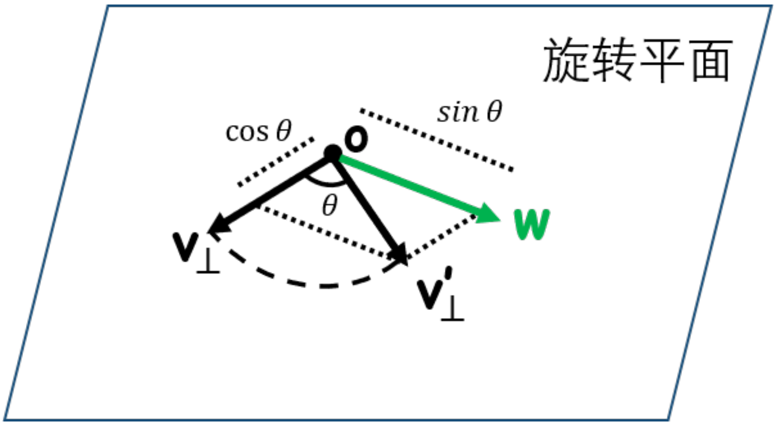


图 B.3: 求解 $\mathbf{v}'_\perp$

只需得到 $\mathbf{v}'_\perp$ 的方向上的单位向量，再乘以模长即可。在旋转平面上将 $\mathbf{w}$，$\mathbf{v}'$ 作为基，不难得到：

$$
\begin{aligned}
\mathbf{v}'_\perp &= \|\mathbf{v}'_\perp\| \cdot \left[cos\theta \frac{\mathbf{v}_\perp}{\|\mathbf{v}_\perp\|} + \sin\theta \frac{\mathbf{w}}{\|\mathbf{w}\|}\right] \\
&= cos\theta \mathbf{v}_\perp + \sin\theta \mathbf{w} \\
&= cos\theta(-\mathbf{n} \times (\mathbf{n} \times \mathbf{v})) + \sin\theta(\mathbf{n} \times \mathbf{v})
\end{aligned}
$$

于是 $\mathbf{v}'$ 终于得到了：

$$
\begin{aligned}
\mathbf{v}' &= \mathbf{v}_\| + \mathbf{v}'_\perp \\
&= \mathbf{v} + \mathbf{n} \times (\mathbf{n} \times \mathbf{v}) - cos\theta(\mathbf{n} \times (\mathbf{n} \times \mathbf{v})) + \sin\theta(\mathbf{n} \times \mathbf{v}) \\
&= \mathbf{v} + (1 - cos\theta)\mathbf{n} \times (\mathbf{n} \times \mathbf{v}) + \sin\theta(\mathbf{n} \times \mathbf{v}) \\
&= \mathbf{v} + (1 - cos\theta)\mathbf{n}^\wedge * \mathbf{n}^\wedge * \mathbf{v} + \sin\theta \mathbf{n}^\wedge * \mathbf{v} \\
&= \left[\mathbf{I} + (1 - cos\theta) * N^2 + \sin\theta * N\right] \mathbf{v}
\end{aligned}
$$

即：

$$
\mathbf{v}' = \left[\mathbf{I} + (1 - cos\theta) * N^2 + \sin\theta * N\right] \mathbf{v} \tag{B.1}
$$

其中 $\mathbf{I}$ 表示单位阵，方便起见，我们将 $\left[\mathbf{I} + (1 - cos\theta) * N^2 + \sin\theta * N\right]$ 这个矩阵记为 $R_{\mathbf{n}}(\theta)$，表示绕过原点的旋转轴 $\mathbf{n}$ 旋转 θ。

$$
\mathbf{v}' = R_{\mathbf{n}}(\theta)\mathbf{v}
$$

注意在B.1中，我们用 N 来表示，有些地方习惯用 n 来表示。

$$\mathbf{nn^T} = \begin{pmatrix} n_x^2 & n_x n_y & n_x n_z \\ n_x n_y & n_y^2 & n_y n_z \\ n_x n_z & n_y n_z & n_z^2 \end{pmatrix}$$

$$n^\wedge = N = \begin{pmatrix} 0 & -n_z & n_y \\ n_z & 0 & -n_x \\ -n_y & n_x & 0 \end{pmatrix}$$

$$N^2 = \begin{pmatrix} -n_y^2 - n_z^2 & n_x n_y & n_x n_z \\ n_x n_y & -n_x^2 - n_z^2 & n_y n_z \\ n_x n_z & n_y n_z & -n_x^2 - n_y^2 \end{pmatrix}$$

所以，

$$\mathbf{nn^T} = \mathbf{I} + N^2 \tag{B.2}$$

将B.2代入B.1可以得到用 $\mathbf{n}$ 表示的形式：

$$R_{\mathbf{n}}(\theta) = \cos\theta * \mathbf{I} + (1 - \cos\theta) * \mathbf{nn}^T + \sin\theta * \mathbf{n}^\wedge \tag{B.3}$$

B.1.2 理解

$R_{\mathbf{n}}(\theta)$ 用四个参数 n_x, n_y, n_z, θ，三个自由度（因为 $\|n\| = 1$）表示了绕过原点的旋转轴旋转 θ 的变换。有些地方也会用一个向量 omega 来表示这四个参数。

$$\begin{aligned} \omega &= \theta\mathbf{n} \\ &= (\theta n_x, \theta n_y, \theta n_z)^T \\ &= (\omega_x, \omega_y, \omega_z)^n \end{aligned}$$

这样的表达存在奇异性。主要的原因在于：

- $R_{\mathbf{n}}(\theta)$ 和 $R_{\mathbf{n}}(\theta + 2k\pi)$ 表示了相同的旋转，k = 1, 2, 3, ...
- $R_{\mathbf{n}}(\theta)$ 和 $R_{-\mathbf{n}}(-\theta)$ 也表示了相同的旋转

而使用罗德里格斯公式的一个好处在于当 θ 非常小的时候，我们可以利用罗德里格斯公式得到近似值：

$$\begin{aligned} R_{\mathbf{n}}(\theta) &= = \mathbf{I} + (1 - cos\theta) * N^2 + \sin\theta * N \\ &\approx \mathbf{I} + \sin\theta * N \\ &\approx \mathbf{I} + \theta * N \\ &= \begin{pmatrix} 1 & -\omega_z & \omega_y \\ \omega_z & 1 & \omega_x \\ -\omega_y & \omega_x & 1 \end{pmatrix} \end{aligned}$$

$$\begin{aligned} error &= \|(1 - cos\theta)N * N + \sin\theta * N - \theta * N\| \\ &= \|(1 - cos\theta)N * N + (\sin\theta - \theta) * N\| \\ &= \|\,[(1 - cos\theta)N + (\sin\theta - \theta) * \mathbf{I}] * N\| \end{aligned}$$

B.1.3 Exponential Twist

用指数的形式来表示 $R_{\mathbf{n}}(\theta)$：

$$
\begin{aligned}
R_{\mathbf{n}}(\theta) &= \lim_{k\to\infty}\left(\mathbf{I}+\frac{1}{k}(\theta*N)\right)^k \\
&= e^{\theta*N}
\end{aligned}
$$

注意 $\theta*N$ 是一个矩阵，对 $e^{\theta*N}$ 对做泰勒展开：

$$
e^{\theta*N} = \mathbf{I}+(\theta*N)+\frac{(\theta*N)^2}{2!}+\frac{(\theta*N)^3}{3!}+\ldots
$$

因为反对称矩阵 N 满足：

$$
N^{k+2} = -N^k, k\in\mathbb{Z}^+
$$

所以

$$
\begin{aligned}
e^{\theta*N} &= \mathbf{I}+(\theta-\frac{\theta^3}{3!}+\ldots)*N+(\frac{\theta^2}{2!}-\frac{\theta^4}{4!}+\ldots)*N^2 \\
&= \mathbf{I}+\sin\theta*N+(1-cos\theta)*N^2
\end{aligned}
$$

笔记 如果你对 $e^{矩阵}$ 表示成级数展开，有一些奇怪或者别扭的话，不用特别纠结。因为这只是一个 ，这样定义的好处在于可以和其他数学知识关联起来变得自洽。具体的说在这里你可以想象到如果 $e^{\theta*N}$ 不是定义成级数展开的样子，是不会将 $e^{\theta*N}$ 表示成 $\mathbf{I}+\sin\theta*N+(1-cos\theta)*N^2$ 的。这有一点类似于牛顿第二定律和万有引力定律 $G\frac{m_1m_2}{r^2}=ma$。如果你定理力 F 的时候，定义成了 $G'ma$，而不是如牛顿一样 $F=ma$。那么就会变成 $\frac{m_1m_2}{r^2}=G'ma$。只是你定义的力的单位和牛顿定义的力的单位会有一些不一样，但本质并没有什么太大的区别，都反映了力、质量和加速度之间的关系。

因为 $e^{矩阵}$ 定义不同于 $e^{标量}$，所以通常 $e^{\mathbf{X}}\cdot e^{\mathbf{Y}}\neq e^{\mathbf{X+Y}}$（$\mathbf{x},\mathbf{Y}$ 是矩阵。只有 $\mathbf{XY}=\mathbf{YX}$ 成立时，$e^{\mathbf{X}}\cdot e^{\mathbf{Y}}=e^{\mathbf{X+Y}}$，你可以通过级数展开的定义轻松的验证这件事情。

B.2 四元数表示三维旋转

B.2.1 推导

正如我们在推导罗德里格斯公式的时候，我们将旋转向量 $\mathbf{v}$ 分解为 $\mathbf{v}=\mathbf{v}_{\parallel}+\mathbf{v}_{\perp}$。这里也一样，只不过我们将他们定义为纯四元数：

$$
\begin{aligned}
v &= [0,\mathbf{v}] & v' &= [0,\mathbf{v}'] \\
v_{\perp} &= [0,\mathbf{v}_{\perp}] & v'_{\perp} &= \left[0,\mathbf{v}'_{\perp}\right] \\
v_{\parallel} &= \left[0,\mathbf{v}_{\parallel}\right] & v'_{\parallel} &= \left[0,\mathbf{v}'_{\parallel}\right] \\
n &= [0,\mathbf{n}]
\end{aligned}
$$

类似的，可以得到纯四元数的形式：

$$
v = v_{\parallel}+v_{\perp} \quad v = v'_{\parallel}+v'_{\perp}
$$

我们只需要分开讨论 $v_{\perp}$ 和 $v_{\parallel}$ 即可。因为 $v_{\parallel}=v'_{\parallel}$，所以只讨论 $v_{\perp}$：

$$
\begin{aligned}
\mathbf{v}'_{\perp} &= cos\theta\mathbf{v}_{\perp}+\sin\theta(\mathbf{n}\times\mathbf{v}) \\
&= cos\theta\mathbf{v}_{\perp}+\sin\theta[\mathbf{n}\times(\mathbf{v}_{\parallel}+\mathbf{v}_{\perp})] \\
&= cos\theta\mathbf{v}_{\perp}+\sin\theta(\mathbf{n}\times\mathbf{v}_{\perp})
\end{aligned}
$$

只需要把上式换成四元数的形式即可，利用式2.1，不难得到 $nv_\perp = [0, \mathbf{n} \times \mathbf{v}_\perp]$，则

$$v'_\perp = cos\theta v_\perp + \sin\theta(nv_\perp)$$

因为四元数乘法满足分配律，稍微整理一下：

$$v'_\perp = [cos\theta + (\sin\theta)n]v_\perp$$

仔细观察的话，你可以发现 $q = cos\theta + (\sin\theta)n$ 也是四元数，q 可以写成 $[cos\theta, (\sin\theta)\mathbf{n}]$ 的形式。

定理 B.1 (四元数表示三维旋转，正交情况)

$\mathbf{v}_\perp$ 绕旋转轴 $\mathbf{n}(\|n\| = 1)$ 旋转 θ 之后的 $\mathbf{v}'$，可以使用四元数乘法来获得。

令 $v_\perp = [0, \mathbf{v}_\perp], q = [cos\theta, (\sin\theta)\mathbf{n}]$：

$$v'_\perp = qv_\perp$$

♡

笔记 如果你尝试计算一下 q 的范数，会发现 $\|q\| = 1$，q 代表的变换前后不会改变模长，代表了一个纯旋转。此外若旋转 2θ，那么只需使变换的四元数为 $q' = [cos2\theta, (\sin 2\theta)\mathbf{n}]$，即 $v'_\perp = q'v_\perp$。当然也可以先旋转 θ 以后再旋转 θ，即 $v'_\perp = qqv_\perp$。换句话说，

$$q^2 = [cos2\theta, (\sin 2\theta)\mathbf{n}]$$

你可以利用代数和三角恒等式的知识，轻松的证明，但是这里从实际物理意义来讲会更加直观。

现在可以得到 v 了：

$$\begin{aligned} v' &= v_\parallel + qv_\perp \qquad (q = [cos\theta, (\sin\theta)\mathbf{n}]) \\ &= 1 \cdot v_\parallel + qv_\perp \\ &= pp^{-1}v_\parallel + ppv_\perp \qquad (p = [cos\tfrac{1}{2}\theta, (\sin\tfrac{1}{2}\theta)\mathbf{n}]) \\ &= pp^{*}v_\parallel + ppv_\perp \qquad (\|p\| = 1, p^{-1} = p^{*}) \end{aligned}$$

为了能够进一步化简，还需要以下两个引理：

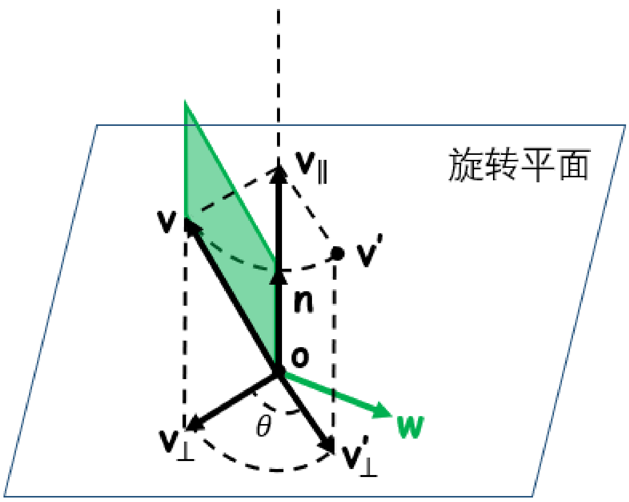
定理 B.2 (四元数的两个引理)

$v_\parallel = [0, \mathbf{v}_\parallel]$，$v_\perp = [0, \mathbf{v}_\perp]$ 是两个纯四元数，四元数 $q = [\alpha, \beta\mathbf{n}]$，$(\|n\| = 1, \alpha, \beta \in \mathbb{R})$，则

$$qv_\parallel = v_\parallel q$$

$$qv_\perp = v_\perp q^{*}$$

♡



利用式2.1，这两个引理不难证明。

利用这两个引理，我们可以继续化简：

$$\begin{aligned} v' &= pp^*v_{\parallel} + ppv_{\perp} \\ &= pv_{\parallel}p^* + pv_{\perp}p^* \\ &= p(v_{\parallel} + v_{\perp})p^* \\ &= pvp^* \end{aligned}$$

B.2.2 理解

通过推导过程可以看到，我们旋转 θ 角度，对应的四元数 p 的角度是 $\frac{1}{2}\theta$。

对 $v_{\perp}$ 施加的变换是 $pp = p^2$，实际上是 $\frac{\theta}{2} + \frac{\theta}{2} = \theta$。

对 $v_{\parallel}$ 施加的变换是 pp^{-1}，实际上是没有变换的。

可以注意到一个四元数 $q = [a, \mathbf{b}]$，如果想要表示纯旋转变换，必须是一个单位四元数。如果想要提取旋转的角度 θ 和旋转轴 $\mathbf{n}$（假设 $\|q\| = 1$），可以直接得到：

$$\begin{cases} \frac{\theta}{2} = \arccos(a) \\ \mathbf{n} = \frac{\mathbf{b}}{\sin(\arccos(a))} \end{cases}$$

B.3 Slerp

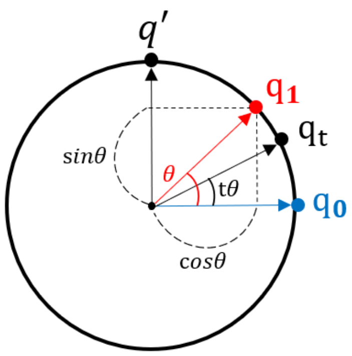


图 B.4: Slerp 推导，注意因为模长为 1 的原因，此圆半径为 1.

$$\begin{aligned} \mathbf{q_t} &= \mathbf{q_0}\cos(t\theta) + \mathbf{q}'\sin(t\theta) \\ &= \mathbf{q_0}\cos(t\theta) + (\mathbf{q_1} - \mathbf{q_0}\cos(\theta))\frac{\sin(t\theta)}{\sin(\theta)} \\ &= \mathbf{q_0}\frac{\cos(t\theta)\sin(\theta) - \sin(t\theta)\cos(\theta)}{\sin(\theta)} + \mathbf{q_1}\frac{\sin(t\theta)}{\sin(\theta)} \\ &= \mathbf{q_0}\frac{\sin(1-t)\theta}{\sin\theta} + \mathbf{q_1}\frac{\sin(t\theta)}{\sin\theta} \end{aligned}$$

附录 C 习题参考答案

C.1 第一章

1. 假设单位正交基 $\{\mathbf{e_1}, \mathbf{e_2}, ..., \mathbf{e_n}\}$ 经过旋转 R 可得到 $\{\mathbf{e'_1}, \mathbf{e'_2}, ..., \mathbf{e'_n}\}$，即 $[\mathbf{e_1}, \mathbf{e_2}, ..., \mathbf{e_n}]\, R = \left[\mathbf{e'_1}, \mathbf{e'_2}, ..., \mathbf{e'_n}\right]$。

$$
\begin{aligned}
R &= [\mathbf{e_1}, \mathbf{e_2}, ..., \mathbf{e_n}]^{-1}\left[\mathbf{e'_1}, \mathbf{e'_2}, ..., \mathbf{e'_n}\right] \\
&= \begin{bmatrix} \mathbf{e_1^T} \\ \mathbf{e_2^T} \\ \cdots \\ \mathbf{e_n^T} \end{bmatrix} \left[\mathbf{e'_1}, \mathbf{e'_2}, ..., \mathbf{e'_n}\right]
\end{aligned}
$$

$$
\begin{aligned}
R^{-1} &= \left[\mathbf{e'_1}, \mathbf{e'_2}, ..., \mathbf{e'_n}\right]^{-1} [\mathbf{e_1}, \mathbf{e_2}, ..., \mathbf{e_n}] \\
&= \begin{bmatrix} \mathbf{e_1'^T} \\ \mathbf{e_2'^T} \\ \cdots \\ \mathbf{e_n'^T} \end{bmatrix} [\mathbf{e_1}, \mathbf{e_2}, ..., \mathbf{e_n}] = R^T
\end{aligned}
$$

2. 图像坐标系下的旋转矩阵为

$$
R = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} cos(\theta) & -\sin(\theta) \\ \sin(\theta) & cos(\theta) \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} cos(\theta) & \sin(\theta) \\ -\sin(\theta) & cos(\theta) \end{pmatrix}
$$

R 的求解也可以直接利用定理2.1得到。

$$
\begin{aligned}
M &= \begin{pmatrix} 1 & 0 & C_x \\ 0 & 1 & C_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -C_x \\ 0 & 1 & -C_y \\ 0 & 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} cos(\theta) & \sin(\theta) & (1 - cos(\theta))C_x - \sin(\theta)C_y \\ -\sin(\theta) & cos(\theta) & \sin(\theta)C_x + (1 - cos(\theta))C_y \\ 0 & 0 & 1 \end{pmatrix}
\end{aligned}
$$

3. 三维点 $\mathbf{v}$ 绕过原点的旋转轴 $\mathbf{n}(\|n\| = 1)$ 旋转 θ 之后得到 $\mathbf{v}'$。

$$
\begin{aligned}
qvq^* &= \left[cos\frac{\theta}{2}, (\sin\frac{\theta}{2})\mathbf{n}\right] * [0, \mathbf{v}] * \left[cos\frac{\theta}{2}, -(\sin\frac{\theta}{2})\mathbf{n}\right] \\
&= \left[-(\sin\frac{\theta}{2})\mathbf{n} \cdot \mathbf{v}, cos\frac{\theta}{2}\mathbf{v} + (\sin\frac{\theta}{2})\mathbf{n} \times \mathbf{v}\right] * \left[cos\frac{\theta}{2}, -(\sin\frac{\theta}{2})\mathbf{n}\right]
\end{aligned}
$$

为了看着方便一些，分成实部 (Real) 和虚部 (Img)。

$$
\begin{aligned}
Real &= -cos\frac{\theta}{2}\sin\frac{\theta}{2}\mathbf{n} \cdot \mathbf{v} + \left[cos\frac{\theta}{2}\mathbf{v} + (\sin\frac{\theta}{2})\mathbf{n} \times \mathbf{v}\right] \sin\frac{\theta}{2}\mathbf{n} \\
&= -cos\frac{\theta}{2}\sin\frac{\theta}{2}\mathbf{n} \cdot \mathbf{v} + cos\frac{\theta}{2}\sin\frac{\theta}{2}\mathbf{v} \cdot \mathbf{n} + 0 = 0
\end{aligned}
$$

$$
\begin{aligned}
Img &= \sin^2\frac{\theta}{2}(\mathbf{n}\cdot\mathbf{v})\mathbf{n} + cos^2\frac{\theta}{2}\mathbf{v} + \sin\frac{\theta}{2}cos\frac{\theta}{2}\mathbf{n}\times\mathbf{v} - \sin\frac{\theta}{2}cos\frac{\theta}{2}\mathbf{v}\times\mathbf{n} - \sin^2\frac{\theta}{2}(\mathbf{n}\times\mathbf{v})\times\mathbf{n} \\
&= \sin^2\frac{\theta}{2}(\mathbf{n}\cdot\mathbf{v})\mathbf{n} + cos^2\frac{\theta}{2}\mathbf{v} + \sin\theta\mathbf{n}\times\mathbf{v} + \sin^2\frac{\theta}{2}\mathbf{n}\times(\mathbf{n}\times\mathbf{v}) \\
&= \sin^2\frac{\theta}{2}(\mathbf{n}\cdot\mathbf{v})\mathbf{n} + cos^2\frac{\theta}{2}\mathbf{v} + \sin\theta\mathbf{n}\times\mathbf{v} + \sin^2\frac{\theta}{2}\left[(\mathbf{n}\cdot\mathbf{v})\mathbf{n} - \mathbf{v}\right] \\
&= 2*\sin^2\frac{\theta}{2}(\mathbf{n}\cdot\mathbf{v})\mathbf{n} + \left[cos^2\frac{\theta}{2} - \sin^2\frac{\theta}{2}\right]\mathbf{v} + \sin\theta\mathbf{n}\times\mathbf{v} \\
&= (1-cos\theta)(\mathbf{n}\cdot\mathbf{v})\mathbf{n} + cos\theta\mathbf{v} + \sin\theta\mathbf{n}\times\mathbf{v}
\end{aligned}
$$

4.

5. 其实没有必要写成矩阵形式，直接写个等式就可以看出关系，如下

$$\mathbf{R}\mathbf{v} + \mathbf{t} = \mathbf{R}'\left(\mathbf{v} + \mathbf{t}'\right)$$

但你应该熟悉矩阵的表达，并且看到矩阵的表达形式，就能很清楚知道旋转和平移变换作用的前后顺序。这里的方式，有助于之后让我们将矩阵和物体上发生的变换，建立牢牢的对应关系。

先旋转再平移 $\mathbf{R}\mathbf{v} + \mathbf{t}$：

$$
\begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}
\begin{pmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix}
\begin{pmatrix} v_x \\ v_y \\ v_z \\ 1 \end{pmatrix}
=
\begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix}
\begin{pmatrix} v_x \\ v_y \\ v_z \\ 1 \end{pmatrix}
$$

先平移再旋转 $\mathbf{R}'\left(\mathbf{v} + \mathbf{t}'\right)$：

$$
\begin{pmatrix} \mathbf{R}' & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix}
\begin{pmatrix} 1 & 0 & 0 & t'_x \\ 0 & 1 & 0 & t'_y \\ 0 & 0 & 1 & t'_z \\ 0 & 0 & 0 & 1 \end{pmatrix}
\begin{pmatrix} v_x \\ v_y \\ v_z \\ 1 \end{pmatrix}
=
\begin{pmatrix} \mathbf{R}' & \mathbf{R}'\mathbf{t}' \\ \mathbf{0}^T & 1 \end{pmatrix}
\begin{pmatrix} v_x \\ v_y \\ v_z \\ 1 \end{pmatrix}
$$

$\begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix}$ 比 $\begin{pmatrix} \mathbf{R}' & \mathbf{R}'\mathbf{t}' \\ \mathbf{0}^T & 1 \end{pmatrix}$ 看起来要好看一些，因为前者的平移部分 $\mathbf{t}$ 没有像后者一样与旋转耦合在一起 $\mathbf{R}'\mathbf{t}'$，看起来很简洁，直观。

从数学意义上来说先旋转后平移，更好。但是在一些情况下，我们思考解决问题的时候更倾向于先平移后旋转。比如看下面的两个小兔子，他们之间的变换只有旋转和平移。如图C.1：

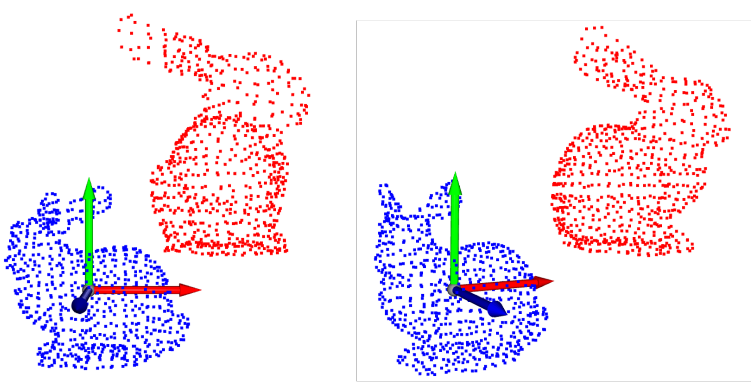


图 C.1: 左右两图是同一场景的不同观察视角（下同），蓝色兔子质心为坐标原点

如果我们想让他们完全重合到一起，通常情况下我们会先平移（红色兔子减去质心），让他们大致重合到一起，再考虑旋转变换。如图C.2。

6. 先缩放再 $SE(3)$，$\mathbf{R}\left(s\mathbf{v}\right) + \mathbf{t}$：

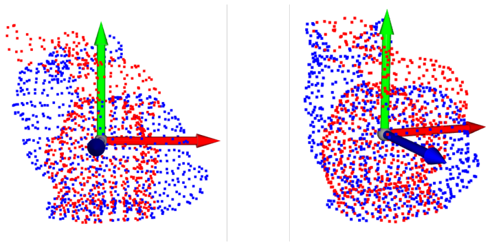


图 C.2: 先平移到一起，再考虑旋转，可能更符合我们的思考方式

$$\begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} s\mathbf{E} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \mathbf{v} = \begin{pmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \mathbf{v}$$

先 $SE(3)$ 再缩放，$s\left[\mathbf{R}\mathbf{v}+\mathbf{t}\right]$：

$$\begin{pmatrix} s\mathbf{E} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \mathbf{v} = \begin{pmatrix} s\mathbf{R} & s\mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \mathbf{v}$$

附录 D 参考资料

如有遗漏，不吝赐教。

1. Fundamentals Of Computer Graphics - Peter Shirley, Steve Marschner.
2. An Invitation to 3-D Vision: From Images to Models - Yi Ma, etc.
3. 3D Math Primer for Graphics and Game Development (2nd Ed) - Fletcher Dunn, Ian Parberry.
4. CS231A: Computer Vision, From 3D Reconstruction to Recognition.
5. Lecture: Multiple View Geometry By Prof. Daniel Cremers
6. Summer School 2020 ROBOTICS & AI - Lie theory for the roboticist
7. 四元数与三维旋转
8. Matrix exponential
9. Eigenvalues and eigenvectors of rotation matrices
10. First Principles of Computer Vision
11. Serrano.Academy
12. 四元数线性插值方法：Slerp（详解证明）
13. wikiwand Slerp
14. Gimbal-lock-visualization
15. 视觉 slam14 讲
16. Logo 部分字体来源于LXGW WenKai / 霞鹜文楷

TODO

1. 3x3 旋转矩阵特征值有复数的概率
2. 旋转矩阵近似误差分析