

SE(3)变换参数和流形优化教程

Jose-Luis Blanco

2014年10月21日

摘 要

SE(3)下的任意一个变换都可以分解为平移和旋转两部分,本技术报告 [1]将以统一的视角回顾一下旋转的三类表述方式: 欧拉角集合, 正交旋转矩阵(即SO(3))和四元数。将会描述的内容:

(i)三类描述方式的等价变换和变换公式(在所有的变换中, 将旋转和平移作为一个整体考虑);

(ii)在每种描述中, 怎样计算位姿-位姿、点-位姿和运算;

(iii)当以高斯分布建模时, 位姿的不确定度如何受变换和其他部分的影响。

简要的给出在流空间上实现最小二乘时, 雅克比矩阵的求解, 在最近工程文献中这是很有前途的方法。

本文采用MRPT的C++库函数实现所描述的算法。所有的实现过程都已经通过单元测试和雅克比矩阵数值估计检验。

注: 这是一篇SE(3)教程的中文翻译, 英文原文:

A tutorial on SE(3) transformation parameterizations and on-manifold optimization

如发现有错误的地方, 烦请告知译者: 1347893477@qq.com. 谢谢合作。

东北大学 孙志明

2016.06.20

目 录

1	3D空间中的刚体变换	5
1.1	基本定义	5
1.2	常用参数	6
1.2.1	3D平移+ yaw-pitch-roll(3D+YPR)	6
1.2.2	3D平移+ 四元数(3D+Quat)	7
1.2.3	4x4变换矩阵	9
2	各种描述之间的等价变换	10
2.1	3D+YPR到3D+Quat	10
2.1.1	变换	10
2.1.2	不确定度	10
2.2	3D+Quat到3D+YPR	11
2.2.1	变换	11
2.2.2	不确定度	12
2.3	3D+YPR到旋转矩阵	13
2.3.1	变换	13
2.4	3D+Quat到变换矩阵	14
2.4.1	变换	14
2.5	变换矩阵到3D+YPR	14
2.5.1	变换	14
2.6	变换矩阵到3D+Quat	15
2.6.1	变换	15
3	位姿与点的和运算	16
3.1	3D+YPR格式描述位姿	16
3.1.1	和运算	16
3.1.2	不确定度	16
3.2	3D+Quat格式描述位姿	17
3.2.1	和运算	17
3.2.2	不确定度	17
3.3	变换矩阵描述位姿	18
4	点与位姿的差运算	19
4.1	3D+YPR格式描述位姿	19
4.1.1	差运算	19

4.1.2	不确定度	19
4.2	3D+Quat格式描述位姿	19
4.2.1	差运算	19
4.2.2	不确定度	20
4.3	变换矩阵描述位姿	21
4.4	位姿-点和/差运算的关系	21
5	两个位姿之间的和运算	22
5.1	3D+YPR格式描述位姿	22
5.1.1	位姿和运算	22
5.1.2	不确定度	22
5.2	3D+Quat格式描述位姿	23
5.2.1	位姿和运算	23
5.2.2	不确定度	23
5.3	变换矩阵格式描述位姿	24
5.3.1	位姿和运算	24
6	位姿的逆	25
6.1	3D+YPR格式描述位姿	25
6.2	3D+Quat格式描述位姿	25
6.2.1	位姿的逆	25
6.2.2	不确定度	25
6.3	变换矩阵格式描述位姿	26
7	位姿变换矩阵的求导	27
7.1	操作	27
7.2	符号	27
7.3	有用的表达式	28
7.3.1	位姿一位姿和运算	28
7.3.2	位姿一点和运算	28
7.3.3	位姿的逆	29
7.3.4	位姿与点的差运算	29
8	李群的概念	30
8.1	定义	30
8.1.1	群	30
8.1.2	流形	30
8.1.3	N维空间的平滑流形	31
8.1.4	流形的切空间	31
8.1.5	李群	31

8.1.6	线性李群（矩阵群）	31
8.1.7	李代数	32
8.1.8	李群的指数和对数映射	32
9	李群SE(3)	33
9.1	特性	33
9.2	SO(3)的李代数	33
9.3	SE(3)的李代数	34
9.4	指数和对数映射	34
9.4.1	SO(3)	34
9.4.2	SE(3)	35
9.5	MRPT中实现	35
10	SE(3)的优化问题	36
10.1	平面欧氏空间的优化问题求解	36
10.2	一种更为恰当的方式：在流形上进行优化	37
10.3	有用的流形求导	37
10.3.1	SE(3) e^ε 的雅可比	37
10.3.2	SO(3)对数的雅可比	38
10.3.3	$e^\varepsilon \oplus D$ 的雅可比	38
10.3.4	$D \oplus e^\varepsilon$ 的雅可比	39
10.3.5	$e^\varepsilon \oplus D \oplus p$ 的雅可比	39
10.3.6	$p \ominus (e^\varepsilon \oplus D)$ 的雅可比	40
10.3.7	$A \oplus e^\varepsilon \oplus D$ 的雅可比	40
10.3.8	$A \oplus e^\varepsilon \oplus D \oplus p$ 的雅可比	41
10.3.9	$p \ominus (A \oplus e^\varepsilon \oplus D)$ 的雅可比	41
11	计算机视觉当中的应用	42
11.1	理想针孔相机的映射模型— $h(p)$	42
11.2	$e^\varepsilon \oplus A \oplus p$ 点的投影	43
11.3	$p \ominus (e^\varepsilon \oplus A)$ 点的投影	43

1 3D空间中的刚体变换

1.1 基本定义

本文关注欧式空间中最有趣的几何结构：三维空间 \mathbb{R}^3 。

在该空间中，通过函数或映射定义一个抽象的变换： $f: \mathbb{R}^3 \rightarrow \mathbb{R}^3$

假设 f 可以是任意的 3×3 矩阵 \mathbf{R} ,映射一个点 $\mathbf{x}_1 = [x_1 \ y_1 \ z_1]^T$ 到点 $\mathbf{x}_2 = [x_2 \ y_2 \ z_2]^T$:

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} = \mathbf{x}_2 = \mathbf{R}\mathbf{x}_1 = \mathbf{R} \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} \quad (1.2)$$

所有可逆 3×3 矩阵的集合形成了一般线性群 $\mathbf{GL}(3, \mathbb{R})$,其中行列式的值等于 ± 1 (i.e. $\mathbf{R}\mathbf{R}^T = \mathbf{R}^T\mathbf{R} = \mathbf{I}_3$)的矩阵集合形成正交群 $\mathbf{O}(3) \subset \mathbf{GL}(3, \mathbb{R})$.群的操作是标准的矩阵乘法, $\mathbf{O}(3)$ 中的两个矩阵相乘得到另外一个 $\mathbf{O}(3)$ 矩阵, 所有这些矩阵称为等距变换矩阵, 即该类变换保持任意两点间距离不变。在这里我们只对行列式值为+1的正向等距感兴趣, 它们构成了正交变换群 $\mathbf{SO}(3) \subset \mathbf{O}(3)$ 。

$\mathbf{SO}(3)$ 只表示纯旋转, 为了处理平移部分, 我们引入 4×4 变换矩阵 \mathbf{T} 并拓展为4维坐标, 第4个坐标为1:

$$\begin{pmatrix} \mathbf{x}_2 \\ 1 \end{pmatrix} = \mathbf{T} \begin{pmatrix} \mathbf{x}_1 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{pmatrix} = \left(\begin{array}{ccc|c} & & & t_x \\ & \mathbf{R} & & t_y \\ & & & t_z \\ 0 & 0 & 0 & 1 \end{array} \right) \begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{pmatrix}$$

$$\mathbf{x}_2 = \mathbf{R}\mathbf{x}_1 + (t_x \ t_y \ t_z)^T \quad (1.3)$$

通常, 任意的 4×4 矩阵属于一般线性群 $\mathbf{GL}(4, \mathbb{R})$,但是我们定义的 \mathbf{T} 形成了仿射刚体变换群即 $\mathbf{SE}(3)$ (旋转为: 正向旋转, 即 $|\mathbf{R}| = +1$). $\mathbf{SE}(3)$ 被证明是一个李群和一个 $\mathbf{SO}(3) \times \mathbb{R}^3$ 结构的流形。本文中将会把 $\mathbf{SE}(3)$ 变换称为位姿, 位姿可以由一个3D平移加上一个正交向量基(即: \mathbf{R} 的列)或一个相对于其他任意坐标参考系的坐标系组成。因为位姿有6个自由度, 所以也称为6D位姿。图1解释了位姿 \mathbf{p} 由相对于参考坐标系 $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ 的坐标系 $(\mathbf{X}', \mathbf{Y}', \mathbf{Z}')$ 来表示。

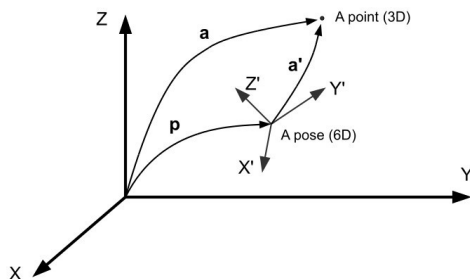


图 1: 6D位姿 \mathbf{p} 的示意图和它在3D点 \mathbf{a} 的相对坐标定义中的角色

给定一个6D位姿 \mathbf{p} 和一个3D点 \mathbf{a} ,两者都是相对于某个全局参考坐标系而言的, 令 \mathbf{a}' 为相对于位姿 \mathbf{p} 的坐标, 我们定义和运算 \oplus 和差运算 \ominus 如下:

$$\mathbf{a} \equiv \mathbf{p} \oplus \mathbf{a}' \quad \text{位姿和运算}$$

$$\mathbf{a} \equiv \mathbf{a} \ominus \mathbf{p} \quad \text{位姿差运算}$$

这些运算集中应用于许多机器人和计算机视觉问题, 例如: 计算一个3D视觉地标相对于相机的相对位姿, 从而计算视觉地标在像平面上的透镜映射。

上面描述的是这种和运算操作在6D位姿和3D点之间的使用, 其也可以用于6D位姿对之间。2个6D位姿和运算获得第3个位姿: $\mathbf{p} = \mathbf{p}_1 + \mathbf{p}_2$ 的物理含义为: 第一个位姿变换以全局坐标系作为参考坐标系, 第二个位姿变换以第一个位姿变换形成的坐标系作为新的参考坐标系, 把两者串联起来, 图片2说明了这点:

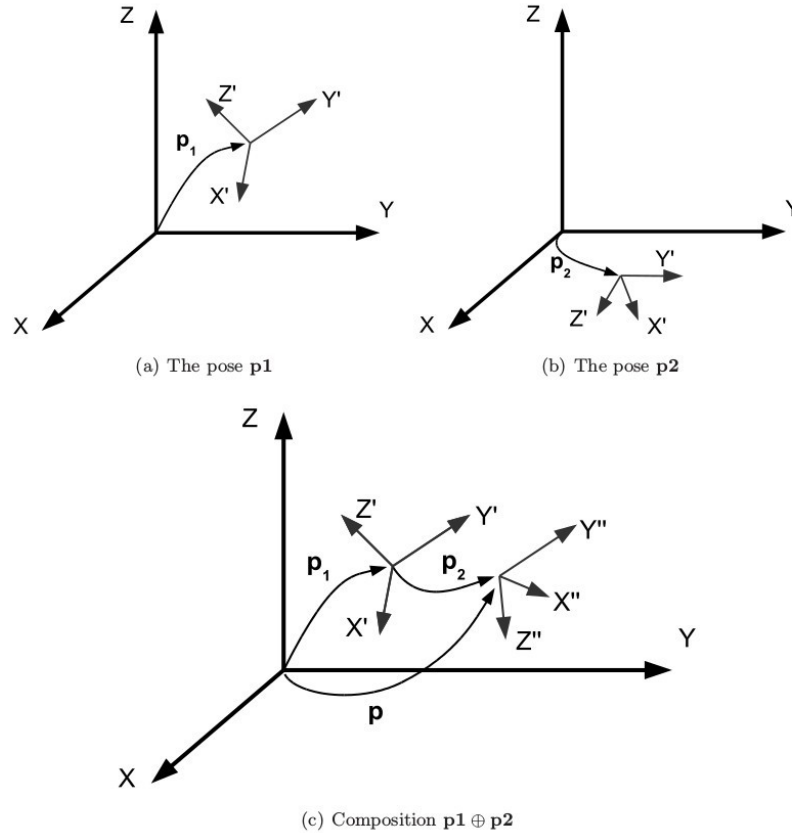


图 2: 6D位姿 $\mathbf{p}_1, \mathbf{p}_2$ 的和运算

6D位姿的差运算 $\mathbf{p}_2 = \mathbf{p} \ominus \mathbf{p}_1$ 表示: 位姿 \mathbf{p} (全局坐标系下)在以 \mathbf{p}_1 为参考坐标系时, 变为位姿 \mathbf{p}_2 .

到目前为止, 位姿、位姿/点、位姿/位姿的运算都是从点的角度出发的。下面部分将介绍几种常见的参数。

1.2 常用参数

1.2.1 3D平移+yaw-pitch-roll(3D+YPR)

一个6D位姿可以用一个6D平移加上一组指定顺序的欧拉角 $\text{yaw}(\phi), \text{pitch}(\chi), \text{roll}(\psi)$ 来表示:

$$\mathbf{p}_6 = [x \ y \ z \ \phi \ \chi \ \psi]^T$$

这些角度的几何含义在图3中被描述。欧拉角也有其他的表示方式，这里提到的是机器人领域最常见到的。

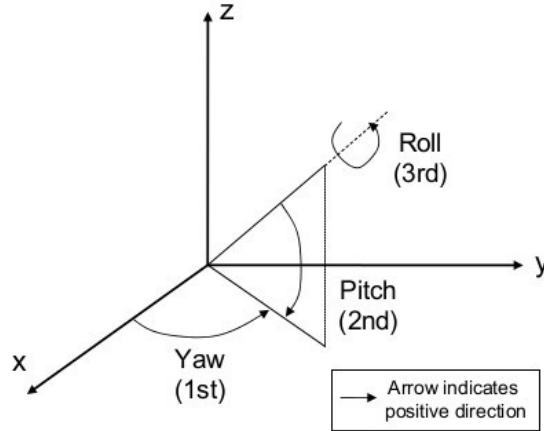


图 3: yaw,pitch,roll角度的约定

总的旋转用三个独立的旋转序列来描述，每一个旋转序列都是绕一个不同的轴。这里的旋转序列是：yaw绕Z轴，然后pitch绕新得到的Y轴，然后roll绕最新得到的X轴。在学术上常用的是固定角roll-pitch-yaw(RPY),绕固定坐标系的角值与之相同，顺序与之相反，两种表示形式完全等价。

这种位姿描述方式是最紧凑的，只有6个参数。在实际应用中其他的描述方式也许会有优势，即使需要的参数更多。

退化情况：万向锁

yaw-pitch-roll旋转描述方式的一个重要缺陷是存在两种退化情况：当pitch=±90°。这种情况下很容易意识到roll的变换变成了yaw的变换。

这意味着对于pitch=±90°，不存在一个唯一的旋转与yaw-pitch-roll一一对应关系。这种情况是需要检测和处理的，以后的描述中有叙述。

在MRPT中实现

基于C++类mrpt::poses::CPose3D实现yaw-pitch-roll位姿：

```
#include <mrpt/base.h>
using namespace mrpt::poses;
CPose3D p(x,y,z,yaw,pitch,roll);
CPose3D p(1.0,2.0,3.0,DEG2RAD(30.0),DEG2RAD(20.0),DEG2RAD(90.0)) ;
```

1.2.2 3D平移+四元数(3D+Quat)

位姿还可以用3D平移加上旋转四元数来表示：

$$\mathbf{P7} = [x \ y \ z \ q_r \ q_x \ q_y \ q_z]^T$$

这里四元数是 $[q_r, (q_x, q_y, q_z)]$.四元数可以解释为绕着轴 $\vec{v} = (v_x, v_y, v_z) \propto (q_x, q_y, q_z)$ 旋转 θ 角(弧度制)。

\vec{v} 、 θ 与四元数的关系为:

$$\begin{aligned} q_r &= \cos \frac{\theta}{2} \\ q_x &= \sin \frac{\theta}{2} v_x \\ q_y &= \sin \frac{\theta}{2} v_y \\ q_z &= \sin \frac{\theta}{2} v_z \end{aligned}$$

如图4所示，其中 q_r 是非负的。

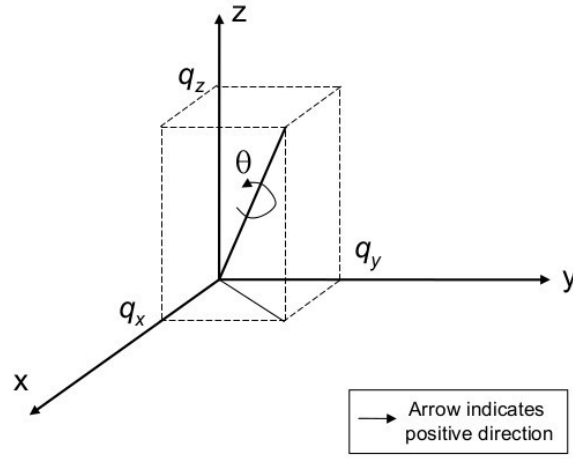


图 4: 一个四元数可以看做是绕3D轴的旋转

在MRPT中实现

基于类mrpt::poses::CPose3DQuat实现这种描述位姿的方式。

其中四元数部分是单位化了的($q_r^2 + q_x^2 + q_y^2 + q_z^2 = 1$).

```
#include <mrpt/base.h>
using namespace mrpt::poses;
using namespace mrpt::math;
CPose3DQuat p(x , y , z , q_r , q_x , q_y , q_z );
CPose3DQuat p(1.0 , 2.0 , 3.0 , CQuaternionDouble(1.0, 0, 0, 0));
```


一个四元数的单位化

在许多情况下，3D+Quat中的四元数部分会漂移不满足单位和等于1。如果四元数中的每一部分都是单独估计的(例如：卡尔曼滤波器或其他高斯牛顿迭代优化方法)，这种情况就会发生。

单位化函数比较简单：

$$\mathbf{q}'(\mathbf{q}) = \begin{pmatrix} q'_r \\ q'_x \\ q'_y \\ q'_z \end{pmatrix} = \frac{q}{|q|} = \frac{1}{(q_r^2 + q_x^2 + q_y^2 + q_z^2)^{1/2}} \begin{pmatrix} q_r \\ q_x \\ q_y \\ q_z \end{pmatrix} \quad (1.6)$$

单位化的雅克比矩阵为：

$$\frac{\partial q'(q_r, q_x, q_y, q_z)}{\partial q_r, q_x, q_y, q_z} = \frac{1}{(q_r^2 + q_x^2 + q_y^2 + q_z^2)^{3/2}} \begin{pmatrix} q_x^2 + q_y^2 + q_z^2 & -q_r q_x & -q_r q_y & -q_r q_z \\ -q_x q_r & q_r^2 + q_y^2 + q_z^2 & -q_x q_y & -q_x q_z \\ -q_y q_r & -q_y q_x & q_r^2 + q_x^2 + q_z^2 & -q_y q_z \\ -q_z q_r & -q_z q_x & -q_z q_y & q_r^2 + q_x^2 + q_y^2 \end{pmatrix} \quad (1.7)$$

1.2.3 4x4变换矩阵

3D空间的任意刚体变换都可以用 4×4 矩阵 \mathbf{p} 来表示：

$$\mathbf{p} = \left(\begin{array}{ccc|c} & & & x \\ & \mathbf{R} & & y \\ & & & z \\ \hline 0 & 0 & 0 & 1 \end{array} \right) \quad (1.8)$$

其中 3×3 正交矩阵 $\mathbf{R} \in \mathbf{SO}(3)$ 是旋转矩阵(\mathbf{p} 中表示旋转的部分)，向量 (x, y, z) 表示6D位姿的平移部分。当应用于3D点的时候，应首先把点转换为4维的齐次坐标，在我们使用的过程中第四维度的值始终是1。

在MRPT中实现

变换矩阵可以像正常矩阵一样来管理。

```
# include < mrpt/base.h >
```

```
using namespace mrpt::math;
```

```
CMatrixDouble44 P;
```

注意在3D+YPR类CPose3D中也存储着一个变换矩阵，可以通过函数CPose3D::getHomogeneousMatrix()。

2 各种描述之间的等价变换

这一章主要关注6D位姿各种描述中旋转部分的等价变换，因为平移部分在各种描述中都一样。

另一个关注点是不同参数变换是怎样影响位姿概率密度分布的不确定度的。假设一个多变量高斯模型，其变换函数的一阶线性化作为真值简单有效的近似。通常变量 $\mathbf{x} \sim N(\bar{x}, \Sigma_x)$ 是一个多变量高斯分布(\bar{x} 是平均值, Σ_x 是协方差), 变换函数 $\mathbf{y} = \mathbf{f}(\mathbf{x})$ 的分布可以用另一个高斯分布来近似:

$$\bar{\mathbf{y}} = \mathbf{f}(\bar{\mathbf{x}}) \quad (2.1)$$

$$\Sigma_y = \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\bar{\mathbf{x}}} \right) \Sigma_x \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\bar{\mathbf{x}}} \right)^T \quad (2.2)$$

注意该方法的另一个选择是使用比例UT变换，对于大幅度的不确定度会给出更好的结果，但需要更多的计算时间并且可能导致半正定协方差矩阵。

2.1 3D+YPR到3D+Quat

2.1.1 变换

任意的YPR旋转 $yaw(\phi)$ 、 $pitch(\chi)$ 、 $roll(\psi)$ 可以表示为一个四元数 (q_r, q_x, q_y, q_z) :

$$q_r(\phi, \chi, \psi) = \cos \frac{\psi}{2} \cos \frac{\chi}{2} \cos \frac{\phi}{2} + \sin \frac{\psi}{2} \sin \frac{\chi}{2} \sin \frac{\phi}{2} \quad (2.3)$$

$$q_x(\phi, \chi, \psi) = \sin \frac{\psi}{2} \cos \frac{\chi}{2} \cos \frac{\phi}{2} - \cos \frac{\psi}{2} \sin \frac{\chi}{2} \sin \frac{\phi}{2} \quad (2.4)$$

$$q_y(\phi, \chi, \psi) = \cos \frac{\psi}{2} \sin \frac{\chi}{2} \cos \frac{\phi}{2} + \sin \frac{\psi}{2} \cos \frac{\chi}{2} \sin \frac{\phi}{2} \quad (2.5)$$

$$q_z(\phi, \chi, \psi) = \cos \frac{\psi}{2} \cos \frac{\chi}{2} \sin \frac{\phi}{2} - \sin \frac{\psi}{2} \sin \frac{\chi}{2} \cos \frac{\phi}{2} \quad (2.6)$$

在MRPT中实现

基于YPR的CPose3D位姿变换为基于四元数的CPose3DQuat的位姿:

```
# include < mrpt/base.h >
using namespace mrpt::poses;
...
CPose3D p6;
CPose3DQuat p7=p6;
```

2.1.2 不确定度

给定一个基于YPR格式6D位姿高斯分布的平均值 $\bar{\mathbf{p}}_6$ 和协方差 $cov(\mathbf{p}_6)$ (6×6 的协方差矩阵), 则等价的基于四元数格式的 7×7 协方差矩阵近似为:

$$cov(\mathbf{p}_7) = \left(\frac{\partial \mathbf{p}_7(\mathbf{p}_6)}{\partial \mathbf{p}_6} \right) cov(\mathbf{p}_6) \left(\frac{\partial \mathbf{p}_7(\mathbf{p}_6)}{\partial \mathbf{p}_6} \right)^T \quad (2.7)$$

其中雅克比矩阵为:

$$\frac{\partial \mathbf{p}_7(\mathbf{p}_6)}{\partial \mathbf{p}_6} = \begin{pmatrix} \mathbf{I}_3 & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{4 \times 3} & \begin{pmatrix} (ssc - ccs)/2 & (scs - csc)/2 & (css - scc)/2 \\ -(csc + scs)/2 & -(ssc + ccs)/2 & (ccc + sss)/2 \\ (scc - css)/2 & (ccc - sss)/2 & (ccs - ssc)/2 \\ (ccc + sss)/2 & -(css + scc)/2 & -(csc + scs)/2 \end{pmatrix} \end{pmatrix}_{7 \times 6} \quad (2.8)$$

其中:

$$\begin{aligned} ccc &= \cos \frac{\psi}{2} \cos \frac{\chi}{2} \cos \frac{\phi}{2}; ccs = \cos \frac{\psi}{2} \cos \frac{\chi}{2} \sin \frac{\phi}{2}; css = \cos \frac{\psi}{2} \sin \frac{\chi}{2} \sin \frac{\phi}{2}; csc = \cos \frac{\psi}{2} \sin \frac{\chi}{2} \cos \frac{\phi}{2}; \\ sss &= \sin \frac{\psi}{2} \sin \frac{\chi}{2} \sin \frac{\phi}{2}; ssc = \sin \frac{\psi}{2} \sin \frac{\chi}{2} \cos \frac{\phi}{2}; scc = \sin \frac{\psi}{2} \cos \frac{\chi}{2} \cos \frac{\phi}{2}; scs = \sin \frac{\psi}{2} \cos \frac{\chi}{2} \sin \frac{\phi}{2}. \end{aligned}$$

在MRPT中实现

关于6D位姿的高斯分布用YPR和四元数表示在程序中分别用类CPose3DPDFGaussian和CPose3DQuatPDFGaussian来实现。通过显式构造函数可以实现两者之间的转换:

```
#include <mrpt/base.h>
using namespace mrpt::poses;
CPose3DPDFGaussian p6(p6_mean, p6_cov);
CPose3DQuatPDFGaussian p7 = CPose3DQuatPDFGaussian(p6);
```

2.2 3D+Quat到3D+YPR

2.2.1 变换

由于1.2.1中退化现象的存在迫使我们很多公式中考虑退化的特殊情况, 当发生退化时四元数必须转化为对应的角度。

首先, 假设一个已经单位化了的四元数, 我们定义判别式 Δ 为:

$$\Delta = q_r q_y - q_x q_z \quad (2.9)$$

然后, 大多数情况下我们会得到 $|\Delta| < 1/2$, 通过三角恒等式和四元数与YPR的变换关系(公式2.3, 2.4)可以得到以下公式来解算 $yaw(\phi)$, $pitch(\chi)$, $roll(\psi)$:

$$\begin{aligned} \phi &= \tan^{-1} \left(2 \frac{q_r q_z + q_x q_y}{1 - 2(q_y^2 + q_z^2)} \right) \\ \chi &= \sin^{-1}(2\Delta) \\ \psi &= \tan^{-1} \left(2 \frac{q_r q_x + q_y q_z}{1 - 2(q_x^2 + q_y^2)} \right) \end{aligned}$$

另一方面，当 $|\Delta| \approx 1/2$ 可以按照如下方式求解：

$$\begin{array}{c|c} \Delta = -\frac{1}{2} & \Delta = \frac{1}{2} \\ \hline \phi = 2\tan^{-1}\frac{q_x}{q_r} & \phi = -2\tan^{-1}\frac{q_x}{q_r} \\ \chi = -\pi/2 & \chi = \pi/2 \\ \psi = 0 & \psi = 0 \end{array} \quad (2.10)$$

在MRPT中实现

```
#include < mrpt/base.h >
using namespace mrpt::poses;
CPose3DQuat p7;
CPose3D p6 = p7;
```

2.2.2 不确定度

给定一个用四元数表示的位姿的高斯分布，其平均值为 $\bar{\mathbf{p}}_7$ 、协方差为 $cov(\mathbf{p}_7)$ 是一个 7×7 的协方差矩阵。我们可以估计出等价YPR表示的 6×6 协方差矩阵：

$$cov(\mathbf{p}_6) = \left(\frac{\partial \mathbf{p}_6(\mathbf{p}_7)}{\partial \mathbf{p}_7} \right) cov(\mathbf{p}_7) \left(\frac{\partial \mathbf{p}_6(\mathbf{p}_7)}{\partial \mathbf{p}_7} \right)^T \quad (2.11)$$

其中的雅可比矩阵为：

$$\frac{\partial \mathbf{p}_7(\mathbf{p}_6)}{\mathbf{p}_6} = \left(\begin{array}{c|c} \mathbf{I}_3 & \mathbf{0}_{3 \times 4} \\ \hline \mathbf{0}_{3 \times 3} & \frac{\partial(\phi, \chi, \psi)(\mathbf{q}_r, \mathbf{q}_x, \mathbf{q}_y, \mathbf{q}_z)}{\partial \mathbf{q}_r, \mathbf{q}_x, \mathbf{q}_y, \mathbf{q}_z} \end{array} \right)_{6 \times 7} \quad (2.12)$$

右下角的子雅可比矩阵由两个连续的变换组成：先雅可比矩阵的单位化，后变换为YPR格式。如下：

$$\frac{\partial(\phi, \chi, \psi)(q_r, q_x, q_y, q_z)}{\partial q_r, q_x, q_y, q_z} = \frac{\partial(\phi, \chi, \psi)(q'_r, q'_x, q'_y, q'_z)}{\partial q'_r, q'_x, q'_y, q'_z} \frac{\partial(q'_r, q'_x, q'_y, q'_z)(q_r, q_x, q_y, q_z)}{\partial q_r, q_x, q_y, q_z} \quad (2.13)$$

等式右侧相乘两项中的第二项是四元数单位化的雅可比矩阵(见1.2.2).在剩余报告当中，如果四元数是单位化了的，则该四元数单位化的雅可比矩阵为 \mathbf{I}_4 .相乘两项中的第一项是公式2.10的雅可比矩阵，对于 $\chi = 90^\circ, \chi = -90^\circ, \chi \neq 90^\circ$ 三种不同的情况有不同的形式。

在MRPT中实现

```
#include < mrpt/base.h >
using namespace mrpt::poses;
CPose3DQuat p7_mean;
CMatrixDouble77 p7_cov;
CPose3DQuatPDFGaussian p7(p7_mean, p7_cov);
CPose3DPDFGaussian p6 = CPose3DPDFGaussian(p7);
```

2.3 3D+YPR到旋转矩阵

2.3.1 变换

YPR表示的6 D位姿的矩阵是:

$$\mathbf{P}(x, y, z, \phi, \chi, \psi) = \left(\begin{array}{ccc|c} & & & x \\ & \mathbf{R}(\phi, \chi, \psi) & & y \\ & & & z \\ \hline 0 & 0 & 0 & 1 \end{array} \right) \quad (2.14)$$

3×3 旋转矩阵可以通过连续绕3个独立的坐标轴旋转得到, 因为是动坐标系所以可以使用矩阵右乘法则得到 \mathbf{R} :

$$\mathbf{R}_z(\phi) = \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix}; \mathbf{R}_y(\chi) = \begin{pmatrix} \cos\chi & \sin\chi & 0 \\ 0 & 0 & 1 \\ -\sin\chi & \cos\chi & 0 \end{pmatrix}; \mathbf{R}_x(\psi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{pmatrix} \quad (2.17)$$

$$\mathbf{R}(\phi, \chi, \psi) = \mathbf{R}_z(\phi)\mathbf{R}_y(\chi)\mathbf{R}_x(\psi) = \begin{pmatrix} \cos\phi\cos\chi & \cos\phi\sin\chi\sin\psi - \sin\phi\cos\psi & \cos\phi\sin\chi\cos\psi + \sin\phi\sin\psi \\ \sin\phi\cos\chi & \sin\phi\sin\chi\sin\psi + \cos\phi\cos\psi & \sin\phi\sin\chi\cos\psi - \cos\phi\sin\psi \\ -\sin\chi & \cos\chi\sin\psi & \cos\chi\cos\psi \end{pmatrix} \quad (2.18)$$

矩阵 \mathbf{P} 容易定义并且没有退化情况, 但是较大的存储要求($4 \times 4 = 16$ 个参数)使得其他的表示方式例如: 3D+YPR($3 + 3 = 6$ 个参数)或3D+Quat($3 + 4 = 7$ 个参数),在许多情况下是更理想的。一种重要的例外情况是计算时间要求严格, 最主要的运算是位姿与点的和/差运算, 这时采用 4×4 矩阵表示方式是其他形式用时的一半。与之相反, 位姿与位姿的和运算采用更为3D+Quat表示方式更为高效。对于不确定度来说, 在任何情况下, 4×4 变换矩阵都不是一个理想的选择, 因为保持协方差矩阵的代价是参数个数平方。在学术上, 带有不确定度的6D位姿最常用的描述方式是3D+Quat格式, 所以这里我们不再介绍如何获取 4×4 变换矩阵的协方差矩阵了。但是变换矩阵的雅可比矩阵有时会作为中间变量使用。

在MRPT中实现

```
#include <mrpt/base.h>
using namespace mrpt::math;
using namespace mrpt::poses;
CPose3D p;
CMatrixDouble44 M=p.getHomogeneousMatrixVal();
```

2.4 3D+Quat到变换矩阵

2.4.1 变换

用3D平移加四元数描述的6D位姿的变换矩阵是:

$$\mathbf{P}(x, y, z, q_r, q_x, q_y, q_z) = \left(\begin{array}{ccc|c} q_r^2 + q_x^2 - q_y^2 + q_z^2 & 2(q_x q_y - q_r q_z) & 2(q_z q_x + q_r q_y) & x \\ 2(q_x q_y + q_r q_z) & q_r^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_r q_x) & y \\ 2(q_z q_x - q_r q_y) & 2(q_y q_z + q_r q_x) & q_r^2 - q_x^2 - q_y^2 + q_z^2 & z \\ \hline 0 & 0 & 0 & 1 \end{array} \right) \quad (2.19)$$

在MRPT中实现

```
#include <mrpt/base.h>
using namespace mrpt::math;
using namespace mrpt::poses;
CPose3DQuat p;
CMatrixDouble44 M=p.getHomogeneousMatrixVal();
```

2.5 变换矩阵到3D+YPR

2.5.1 变换

3D+YPR形式的6D位姿的变换矩阵为:

$$\mathbf{P}(x, y, z, \phi, \chi, \psi) = \left(\begin{array}{ccc|c} \cos\phi\cos\chi & \cos\phi\sin\chi\sin\psi - \sin\phi\cos\psi & \cos\phi\sin\chi\cos\psi + \sin\phi\sin\psi & x \\ \sin\phi\cos\chi & \sin\phi\sin\chi\sin\psi + \cos\phi\cos\psi & \sin\phi\sin\chi\cos\psi - \cos\phi\sin\psi & y \\ -\sin\chi & \cos\chi\sin\psi & \cos\chi\cos\psi & z \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$$

$$= \left(\begin{array}{ccc|c} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ \hline 0 & 0 & 0 & 1 \end{array} \right)$$

很显然3D平移部分为:
$$\begin{cases} x = p_{14} \\ y = p_{24} \\ z = p_{34} \end{cases}$$

关于三个角度 $yaw(\phi)$, $pitch(\chi)$, $roll(\psi)$, 由于要处理退化现象, 它们必须分两步来求解。

首先, 获取pitch:

$$\chi = atan2(-p_{31}, \sqrt{p_{11}^2 + p_{21}^2}) \quad (2.20)$$

接下来根据退化条件来求解:

$$\chi = -90^\circ \rightarrow \begin{cases} \phi = \text{atan2}(-p_{23}, -p_{13}) \\ \psi = 0 \end{cases} \quad (2.21)$$

$$|\chi| \neq +90^\circ \rightarrow \begin{cases} \phi = \text{atan2}(p_{21}, p_{11}) \\ \psi = \text{atan2}(p_{32}, p_{33}) \end{cases} \quad (2.22)$$

$$\chi = +90^\circ \rightarrow \begin{cases} \phi = \text{atan2}(p_{23}, p_{13}) \\ \psi = 0 \end{cases} \quad (2.23)$$

在MRPT中实现

```
#include <mrpt/base.h>
using namespace mrpt::math;
using namespace mrpt::poses;
CMatrixDouble44 M;
...
CPose3D p= CPose3D(M);
```

2.6 变换矩阵到3D+Quat

2.6.1 变换

3×3 变换矩阵为四元数的数值稳定求解方法在引用中给出, 包括创建临时 4×4 矩阵、计算最大特征值对应的特征向量。然而如果我们确定矩阵是正交的, 一种更为高效的选择是先变换为YPR方式, 再转换为四元数表示方式。

在MRPT中实现

```
#include <mrpt/base.h>
using namespace mrpt::math;
using namespace mrpt::poses;
CMatrixDouble44 M;
...
CPose3DQuat p= CPose3DQuat(M);
```

3 位姿与点的和运算

本章回顾怎样计算给定位姿 \mathbf{p} 和局部点坐标 \mathbf{a}' 去求解全局坐标 \mathbf{a} 。如图1所示， $\mathbf{a} = \mathbf{p} \oplus \mathbf{a}'$ 。

3.1 3D+YPR格式描述位姿

3.1.1 和运算

这种情况下，首先用公式2.18计算 4×4 位姿变换矩阵，然后按照3.3节来处理。

在MRPT中实现

```

    位姿一点和运算实现:
    #include <mrpt/base.h>
    using namespace mrpt::math;
    using namespace mrpt::poses;

    CPose3D q;
    TPoint3D p_i, p_o;
    ...
    q.composePoint(p_i, p_o);

```

3.1.2 不确定度

给定一个3D+YPR格式6D位姿的高斯分布，它的平均值为 $\bar{\mathbf{p}}_6 = (\bar{x} \ \bar{y} \ \bar{z} \ \bar{\phi} \ \bar{\chi} \ \bar{\psi})^T$ 、协方差 $cov(\mathbf{p}_6)$ 为 6×6 的矩阵，一个局部点坐标 \mathbf{a}' 的平均值为 $\bar{\mathbf{a}}' = (\bar{a}'_x, \bar{a}'_y, \bar{a}'_z)^T$ 、协方差 $cov(\mathbf{a}')$ 。假定两个分布是独立的，那么全局点坐标 $\mathbf{a} = \mathbf{f}_{\mathbf{pr}}(\mathbf{p}_6, \mathbf{a}') = \mathbf{p}_6 \oplus \mathbf{a}'$ 的协方差为：

$$cov(\mathbf{a}) = \frac{\partial \mathbf{f}_{\mathbf{pr}}(\mathbf{p}_6, \mathbf{a}')}{\partial \mathbf{p}_6} cov(\mathbf{p}_6) \left(\frac{\partial \mathbf{f}_{\mathbf{pr}}(\mathbf{p}_6, \mathbf{a}')}{\partial \mathbf{p}_6} \right)^T + \frac{\partial \mathbf{f}_{\mathbf{pr}}(\mathbf{p}_6, \mathbf{a}')}{\partial \mathbf{a}'} cov(\mathbf{a}') \left(\frac{\partial \mathbf{f}_{\mathbf{pr}}(\mathbf{p}_6, \mathbf{a}')}{\partial \mathbf{a}'} \right)^T \quad (3.1)$$

$$\text{其中: } \frac{\partial \mathbf{f}_{\mathbf{pr}}(\mathbf{p}_6, \mathbf{a}')}{\partial \mathbf{p}_6} \Big|_{3 \times 6} = \left(\begin{array}{c|ccc} & j_{14} & j_{15} & j_{16} \\ \mathbf{I}_3 & j_{24} & j_{25} & j_{26} \\ & j_{34} & j_{35} & j_{36} \end{array} \right) \quad (3.2)$$

$$\frac{\partial \mathbf{f}_{\mathbf{pr}}(\mathbf{p}_6, \mathbf{a}')}{\partial \mathbf{a}'} \Big|_{3 \times 3} = \mathbf{R}(\bar{\phi}, \bar{\chi}, \bar{\psi}) \quad \text{See } Eq(2.18) \quad (3.3)$$

$$j_{14} = -\bar{a}'_x \sin \bar{\phi} \cos \bar{\chi} + \bar{a}'_y (-\sin \bar{\phi} \sin \bar{\chi} \sin \bar{\psi} - \cos \bar{\phi} \cos \bar{\psi}) + \bar{a}'_z (-\sin \bar{\phi} \sin \bar{\chi} \cos \bar{\psi} + \cos \bar{\phi} \sin \bar{\psi})$$

$$j_{15} = -\bar{a}'_x \cos \bar{\phi} \sin \bar{\chi} + \bar{a}'_y (\cos \bar{\phi} \cos \bar{\chi} \sin \bar{\psi}) + \bar{a}'_z (\cos \bar{\phi} \cos \bar{\chi} \cos \bar{\psi})$$

$$j_{16} = \bar{a}'_y (\cos \bar{\phi} \sin \bar{\chi} \cos \bar{\psi} + \sin \bar{\phi} \sin \bar{\psi}) + \bar{a}'_z (-\cos \bar{\phi} \sin \bar{\chi} \sin \bar{\psi} + \sin \bar{\phi} \cos \bar{\psi})$$

$$j_{24} = \bar{a}'_x \cos \bar{\phi} \cos \bar{\chi} + \bar{a}'_y (\cos \bar{\phi} \sin \bar{\chi} \sin \bar{\psi} - \sin \bar{\phi} \cos \bar{\psi}) + \bar{a}'_z (\cos \bar{\phi} \sin \bar{\chi} \cos \bar{\psi} + \sin \bar{\phi} \sin \bar{\psi})$$

$$j_{25} = -\bar{a}'_x \sin \bar{\phi} \sin \bar{\chi} + \bar{a}'_y (\sin \bar{\phi} \cos \bar{\chi} \sin \bar{\psi}) + \bar{a}'_z (\sin \bar{\phi} \cos \bar{\chi} \cos \bar{\psi})$$

$$j_{26} = \bar{a}'_y (\sin \bar{\phi} \sin \bar{\chi} \cos \bar{\psi} - \cos \bar{\phi} \sin \bar{\psi}) + \bar{a}'_z (-\sin \bar{\phi} \sin \bar{\chi} \sin \bar{\psi} - \cos \bar{\phi} \cos \bar{\psi})$$

$$j_{34} = 0$$

$$j_{35} = -\bar{a}'_x \cos \bar{\chi} - \bar{a}'_y \sin \bar{\chi} \sin \bar{\psi} - \bar{a}'_z \sin \bar{\chi} \cos \bar{\psi}$$

$$j_{36} = \bar{a}'_y \cos \bar{\chi} \cos \bar{\psi} - \bar{a}'_z \cos \bar{\chi} \sin \bar{\psi}$$

对于非常小的旋转来说，这个雅克比矩阵的近似在 [3]中给出。

方法是令 $\frac{\partial \mathbf{f}_{\mathbf{pr}}(\mathbf{p}_6, \mathbf{a}')}{\partial \mathbf{p}_6}$ 中的 $\sin \alpha \approx 0, \cos \alpha \approx 1$ 得到:

$$\frac{\partial \mathbf{f}_{\mathbf{pr}}(\mathbf{p}_6, \mathbf{a}')}{\partial \mathbf{p}_6} \Big|_{3 \times 6} \approx \begin{pmatrix} \mathbf{I}_3 & \begin{vmatrix} -\bar{a}'_y & \bar{a}'_z & 0 \\ \bar{a}'_x & 0 & -\bar{a}'_z \\ 0 & -\bar{a}'_x & \bar{a}'_y \end{vmatrix} \end{pmatrix} \quad (\text{small rotation only!!}) \quad (3.4)$$

在MRPT中实现

没有直接实现带有不确定度的位姿一点和运算方法，但需要的两个雅克比矩阵可以通过函数 `composePoint()`:

```
#include < mrpt/base.h >
using namespace mrpt::math;
using namespace mrpt::poses;
CPose3D q;
CMatrixFixedNumeric< double, 3, 3 > df__dpoint;
CMatrixFixedNumeric< double, 3, 6 > df__dpose;
q.composePoint(lx, ly, lz, gx, gy, gz, &df__dpoint, &df__dpose);
```

3.2 3D+Quat格式描述位姿

3.2.1 和运算

给定一个用 $\mathbf{p}_7 = [x, y, z, q_r, q_x, q_y, q_z]^T$ 描述的位姿，和已知坐标点 $\mathbf{a}' = [a'_x, a'_y, a'_z]^T$ ，通过 $\mathbf{a} = \mathbf{p}_7 \oplus \mathbf{a}'$ 获得点坐标 $\mathbf{a} = [a_x, a_y, a_z]^T$:

$$\mathbf{a} = \mathbf{f}_{\mathbf{qr}}(\mathbf{p}, \mathbf{a}') \quad (3.5)$$

其中函数 $\mathbf{f}_{\mathbf{qr}}(\cdot)$ 为:

$$\mathbf{f}_{\mathbf{qr}}(\mathbf{p}, \mathbf{a}') = \begin{pmatrix} x + a'_x + 2[-(q_y^2 + q_z^2)a'_x + (q_x q_y - q_r q_z)a'_y + (q_r q_y + q_x q_z)a'_z] \\ y + a'_y + 2[(q_r q_z + q_x q_y)a'_x - (q_x^2 + q_z^2)a'_y + (q_y q_z - q_r q_x)a'_z] \\ z + a'_z + 2[(q_x q_z - q_r q_y)a'_x + (a_r q_x + q_y q_z)a'_y - (q_x^2 + q_y^2)a'_z] \end{pmatrix} \quad (3.6)$$

3.2.2 不确定度

四元数描述位姿的高斯分布的平均值为 $\bar{\mathbf{p}}_7$ ， 7×7 协方差矩阵为 $\text{cov}(\mathbf{p}_7)$ ，3D点 \mathbf{a}' 的平均值和协方差分别为 $\bar{\mathbf{a}}', \text{cov}(\mathbf{a}')$ ，那么点 $\mathbf{a} = \mathbf{p}_7 \oplus \mathbf{a}'$ 的近似协方差矩阵为:

$$\text{cov}(\mathbf{a}) = \frac{\partial \mathbf{f}_{\mathbf{qr}}(\mathbf{p}_7, \mathbf{a}')}{\partial \mathbf{p}_7} \text{cov}(\mathbf{p}_7) \left(\frac{\partial \mathbf{f}_{\mathbf{qr}}(\mathbf{p}_7, \mathbf{a}')}{\partial \mathbf{p}_7} \right)^T + \frac{\partial \mathbf{f}_{\mathbf{qr}}(\mathbf{p}_7, \mathbf{a}')}{\partial \mathbf{a}'} \text{cov}(\mathbf{a}') \left(\frac{\partial \mathbf{f}_{\mathbf{qr}}(\mathbf{p}_7, \mathbf{a}')}{\partial \mathbf{a}'} \right)^T \quad (3.7)$$

其中雅可比矩阵为:

$$\frac{\partial \mathbf{f}_{\mathbf{qr}}(\mathbf{p}_7, \mathbf{a}')}{\partial \mathbf{p}_7} \Big|_{3 \times 7} = \left(\begin{array}{ccc|c} 1 & 0 & 0 & \frac{\partial \mathbf{f}_{\mathbf{qr}}(\mathbf{p}, \mathbf{a}')}{\partial [\mathbf{q}_r, \mathbf{q}_x, \mathbf{q}_y, \mathbf{q}_z]} \\ 0 & 1 & 0 & \\ 0 & 0 & 1 & \end{array} \right) \quad (3.8)$$

$$\frac{\partial \mathbf{f}_{\mathbf{qr}}(\mathbf{p}, \mathbf{a}')}{\partial [\mathbf{q}_r, \mathbf{q}_x, \mathbf{q}_y, \mathbf{q}_z]} = 2 \begin{pmatrix} -q_z a'_y + q_y a'_z & q_y a'_y + q_z a'_z & -2q_y a'_x + q_x a'_y + q_r a'_z & -2q_z a'_x - q_r a'_y + q_x a'_z \\ q_z a'_x - q_x a'_z & q_y a'_x - 2q_x a'_y - q_r a'_z & q_x a'_x + q_z a'_z & q_r a'_x - 2q_z a'_y + q_y a'_z \\ -q_y a'_x + q_x a'_y & q_z a'_x + q_r a'_y - 2q_x a'_z & -q_r a'_x + q_z a'_y - 2q_y a'_z & q_x a'_x + q_y a'_y \end{pmatrix} \times \frac{\partial (q'_r, q'_x, q'_y, q'_z)(q_r, q_x, q_y, q_z)}{\partial q_r, q_x, q_y, q_z} \quad (3.9)$$

另一个雅可比矩阵为:

$$\frac{\partial \mathbf{f}_{\mathbf{qr}}(\mathbf{p}_7, \mathbf{a}')}{\partial \mathbf{a}'} = 2 \begin{pmatrix} \frac{1}{2} - q_y^2 - q_z^2 & q_x q_y - q_r q_z & q_r q_y + q_x q_z \\ q_r q_z + q_x q_y & \frac{1}{2} - q_x^2 - q_z^2 & q_y q_z - q_r q_x \\ q_x q_z - q_r q_y & q_r q_x + q_y q_z & \frac{1}{2} - q_x^2 - q_y^2 \end{pmatrix} \quad (3.10)$$

在MRPT中实现

位姿一点和运算实现:

```
#include < mrpt/base.h >
using namespace mrpt::math;
using namespace mrpt::poses;
CPose3DQuat q;
TPoint3D p_i, p_o;
...
q.composePoint(p_i, p_o);
```

没有直接实现带有不确定度的位姿一点和运算方法, 但需要的两个雅可比矩阵可以通过函数composePoint()获得:

```
#include < mrpt/base.h >
using namespace mrpt::math;
using namespace mrpt::poses;
CPose3DQuat q;
CMatrixFixedNumeric< double, 3, 3 > df_dpoint;
CMatrixFixedNumeric< double, 3, 6 > df_dpose;
q.composePoint(lx, ly, lz, gx, gy, gz, &df_dpoint, &df_dpose);
```

3.3 变换矩阵描述位姿

用给定的 4×4 矩阵 \mathbf{M} 来求解6D位姿 \mathbf{p} 和局部坐标点 $\mathbf{a}' = [a'_x, a'_y, a'_z]$ 的和运算得到全局点坐标

$\mathbf{a} = [a_x, a_y, a_z]$: $\mathbf{a} = \mathbf{p} \oplus \mathbf{a}'$

$$\begin{pmatrix} a_x \\ a_y \\ a_z \\ 1 \end{pmatrix} = \mathbf{M} \begin{pmatrix} a'_x \\ a'_y \\ a'_z \\ 1 \end{pmatrix} \quad (3.10)$$

4 点与位姿的差运算

本章回顾下给定位姿 \mathbf{p} 和全局坐标 \mathbf{a} ,怎样来求解局部点坐标 \mathbf{a}' , 即 $\mathbf{a}' = \mathbf{a} \ominus \mathbf{p}$,如图1.1中所示。

4.1 3D+YPR格式描述位姿

4.1.1 差运算

先根据2.3获取位姿的变换矩阵, 在按照4.3求相对于位姿 \mathbf{p} 的局部点坐标。

在MRPT中实现

```
#include <mrpt/base.h>
using namespace mrpt::math;
using namespace mrpt::poses;
CPose3D q;
TPoint3D p_i, p_o;
q.inverseComposePoint(p_i, p_o);
```

4.1.2 不确定度

首先变换为3D + Quat格式, 然后再进行3D + Quat方式求不确定度。

4.2 3D+Quat格式描述位姿

4.2.1 差运算

给定一个位姿 $\mathbf{p}_7 = [x, y, z, q_r, q_x, q_y, q_z]^T$ 和全局点坐标 $\mathbf{a} = [a_x, a_y, a_z]^T$,则相对于位姿 \mathbf{p}_7 的局部点坐标 $\mathbf{a}' = \mathbf{a} \ominus \mathbf{p}_7$:

$$\mathbf{a}' = \mathbf{f}_{\text{qri}}(\mathbf{a}, \mathbf{p}_7) = \begin{pmatrix} (a_x - x) + 2[-(q_y^2 + q_z^2)(a_x - x) + (q_x q_y - q_r q_z)(a_y - y) + (q_r q_y + q_x q_z)(a_z - z)] \\ (a_y - y) + 2[(q_r q_z + q_x q_y)(a_x - x) - (q_x^2 + q_z^2)(a_y - y) + (q_y q_z - q_r q_x)(a_z - z)] \\ (a_z - z) + 2[(q_x q_z - q_r q_y)(a_x - x) + (q_r q_x + q_y q_z)(a_y - y) - (q_x^2 + q_y^2)(a_z - z)] \end{pmatrix} \quad (4.1)$$

在MRPT中实现

```
#include <mrpt/base.h>
using namespace mrpt::math;
using namespace mrpt::poses;
CPose3DQuat q;
TPoint3D p_i, p_o;
q.inverseComposePoint(p_i, p_o);
```

4.2.2 不确定度

给定一个3D+Quat格式的位姿，其平均值和 7×7 协方差矩阵分别为： $\bar{\mathbf{p}}_7$ 、 $\text{cov}(\mathbf{p}_7)$ ；给定一个点坐标的高斯分布，其平均值和 3×3 矩阵协方差分别为： $\bar{\mathbf{a}}$ 、 $\text{cov}(\mathbf{a})$ 可以估计变换得到的局部点坐标的协方差为：

$$\begin{aligned} \text{cov}(\mathbf{a}') &= \frac{\partial \mathbf{f}_{\text{qri}}(\mathbf{a}, \mathbf{p}_7)}{\partial \mathbf{p}_7} \text{cov}(\mathbf{p}_7) \left(\frac{\partial \mathbf{f}_{\text{qri}}(\mathbf{a}, \mathbf{p}_7)}{\partial \mathbf{p}_7} \right)^T \\ &+ \frac{\partial \mathbf{f}_{\text{qri}}(\mathbf{a}, \mathbf{p}_7)}{\partial \mathbf{a}} \text{cov}(\mathbf{a}) \left(\frac{\partial \mathbf{f}_{\text{qri}}(\mathbf{a}, \mathbf{p}_7)}{\partial \mathbf{a}} \right)^T \end{aligned} \quad (4.2)$$

其中雅可比矩阵：

$$\frac{\partial \mathbf{f}_{\text{qri}}(\mathbf{a}, \mathbf{p}_7)}{\partial \mathbf{a}} = \begin{pmatrix} 1 - 2(q_y^2 + q_z^2) & 2(q_x q_y + q_r q_z) & 2(-q_r q_y + q_x q_z) \\ 2(-q_r q_z + q_x q_y) & 1 - 2(q_x^2 + q_z^2) & 2(q_y q_z + q_r q_x) \\ 2(q_x q_z + q_r q_y) & 2(-q_r q_x + q_y q_z) & 1 - 2(q_x^2 + q_y^2) \end{pmatrix} \quad (4.3)$$

$$\frac{\partial \mathbf{f}_{\text{qri}}(\mathbf{a}, \mathbf{p}_7)}{\partial \mathbf{p}_7} = \begin{pmatrix} 2(q_y^2 + q_z^2) - 1 & -2(q_x q_y + q_r q_z) & 2(q_r q_y - q_x q_z) \\ 2(q_r q_z - q_x q_y) & 2(q_x^2 + q_z^2) - 1 & -2(q_y q_z + q_r q_x) \\ -2(q_x q_z + q_r q_y) & 2(q_r q_x - q_y q_z) & 2(q_x^2 + q_y^2) - 1 \end{pmatrix} \left| \frac{\partial \mathbf{f}_{\text{qri}}(\mathbf{a}, \mathbf{p})}{\partial \mathbf{p}} \right| \quad (4.4)$$

令 $\Delta x = (a_x - x)$ ， $\Delta y = (a_y - y)$ ， $\Delta z = (a_z - z)$ ，则：

$$\begin{aligned} \frac{\partial \mathbf{f}_{\text{qri}}(\mathbf{a}, \mathbf{p}_7)}{\partial \mathbf{p}_7} &= 2 * \begin{pmatrix} q_y \Delta z - q_z \Delta y & q_y \Delta y + q_z \Delta z & q_x \Delta y - 2q_y \Delta x + q_r \Delta z & q_x \Delta z - q_r \Delta y - 2q_z \Delta x \\ q_z \Delta x - q_x \Delta z & q_y \Delta x - 2q_x \Delta y - q_r \Delta z & q_x \Delta x + q_z \Delta z & q_r \Delta x - 2q_z \Delta y + q_y \Delta z \\ q_x \Delta y - q_y \Delta x & q_z \Delta x + q_r \Delta y - 2q_x \Delta z & q_z \Delta y - q_r \Delta x - 2q_y \Delta z & q_x \Delta x + q_y \Delta y \end{pmatrix} \\ &* \frac{\partial(q'_r, q'_x, q'_y, q'_z)(q_r, q_x, q_y, q_z)}{\partial q_r, q_x, q_y, q_z} \end{aligned} \quad (4.5)$$

这里第二项为四元数单位化的雅可比矩阵的(见1.2.2)。

在MRPT中实现

```
#include <mrpt/base.h>
using namespace mrpt::math;
using namespace mrpt::poses;
CPose3DQuat q;
TPoint3D g, l; //全局坐标和局部坐标
CMatrixFixedNumeric< double, 3, 3 > dfi_dpoint; //3 × 3雅可比矩阵
CMatrixFixedNumeric< double, 3, 7 > dfi_dpose; //3 × 7雅可比矩阵
...
q.inverseComposePoint(g.x, g.y, g.z, l.x, l.y, l.z, &dfi_dpoint, &dfi_dpose);
```

4.3 变换矩阵描述位姿

给定一个 4×4 的变换矩阵 \mathbf{M} 描述的6D位姿 \mathbf{p} 和一个全局坐标点 $\mathbf{a} = [a_x, a_y, a_z]$,则局部坐标 $\mathbf{a}' = [a'_x, a'_y, a'_z]$:

$$\mathbf{a}' = \mathbf{a} \ominus \mathbf{p}$$

$$\begin{pmatrix} a'_x \\ a'_y \\ a'_z \\ 1 \end{pmatrix} = \mathbf{M}^{-1} \begin{pmatrix} a_x \\ a_y \\ a_z \\ 1 \end{pmatrix} \quad (4.6)$$

这里使用了齐次坐标，在6.3中会有更为高效的方法。

4.4 位姿-点和/差运算的关系

由前面的矩阵描述我们可以得到一个有趣的结论：

$$\mathbf{a} = \mathbf{p} \oplus \mathbf{a}' \leftrightarrow \mathbf{a}' = \mathbf{a} \ominus \mathbf{p} \quad (4.7)$$

从 $\mathbf{a} = \mathbf{p} \oplus \mathbf{a}'$ 出发，使用矩阵格式，我们可以得到如下结论：

$$\begin{aligned} \mathbf{a} &= \mathbf{p} \oplus \mathbf{a}' \\ \mathbf{A} &= \mathbf{P}\mathbf{A}' \\ \mathbf{P}^{-1}\mathbf{A} &= \mathbf{A}' \\ (\ominus \mathbf{p}) \oplus \mathbf{a} &= \mathbf{a}' \\ (\ominus \mathbf{p}) \oplus \mathbf{a} &= \mathbf{a} \ominus \mathbf{p} \end{aligned}$$

这里 $(\ominus p)$ 是位姿的逆。所以任意的差运算都可以转换为和运算。

5 两个位姿之间的和运算

本章主要介绍位姿一位姿和运算，即 $\mathbf{p} = \mathbf{p}_1 \oplus \mathbf{p}_2$ ，如图1.2所示。

5.1 3D+YPR格式描述位姿

5.1.1 位姿和运算

3D+YPR格式描述的位姿没有简单实现两个位姿和运算的公式，推荐先转换为3D+Quat或变换矩阵格式，然后进行和运算，最后再把结果转换为3D+YPR格式。

5.1.2 不确定度

令 $N(\bar{\mathbf{p}}_6^1, cov(\mathbf{p}_6^1))$ 和 $N(\bar{\mathbf{p}}_6^2, cov(\mathbf{p}_6^2))$ 表示两个3D+YPR格式的独立的高斯分布对。(上角标表示位姿索引，而非指数。)两个位姿和运算的高斯分布 $\mathbf{p}_6^R = \mathbf{p}_6^1 \oplus \mathbf{p}_6^2$ 通过线性误差传播函数来近似，平均值:

$$\begin{aligned}\bar{\mathbf{p}}_6^R &= \mathbf{f}_{pc}(\bar{\mathbf{p}}_6^1, \bar{\mathbf{p}}_6^2) \\ &= \bar{\mathbf{p}}_6^1 \oplus \bar{\mathbf{p}}_6^2\end{aligned}\quad (5.1)$$

协方差:

$$\begin{aligned}cov(\mathbf{p}_6^R) &= \left(\frac{\partial \mathbf{f}_{pc}(\mathbf{p}_6^1, \mathbf{p}_6^2)}{\partial \mathbf{p}_6^1} \right) cov(\mathbf{p}_6^1) \left(\frac{\partial \mathbf{f}_{pc}(\mathbf{p}_6^1, \mathbf{p}_6^2)}{\partial \mathbf{p}_6^1} \right)^T \\ &\quad + \left(\frac{\partial \mathbf{f}_{pc}(\mathbf{p}_6^1, \mathbf{p}_6^2)}{\partial \mathbf{p}_6^2} \right) cov(\mathbf{p}_6^2) \left(\frac{\partial \mathbf{f}_{pc}(\mathbf{p}_6^1, \mathbf{p}_6^2)}{\partial \mathbf{p}_6^2} \right)^T\end{aligned}\quad (5.2)$$

因为 $\mathbf{f}_{pc}(\cdot, \cdot)$ 没有简单的表达式可以将yaw-pitch-roll角转换为和运算相应的三个角。所以雅可比矩阵 $\frac{\partial \mathbf{f}_{pc}(\mathbf{p}_6^1, \mathbf{p}_6^2)}{\partial \mathbf{p}_6^1}$ 和 $\frac{\partial \mathbf{f}_{pc}(\mathbf{p}_6^1, \mathbf{p}_6^2)}{\partial \mathbf{p}_6^2}$ 采用封闭表达式来求解。

解的求解过程如下:

首先，3D+YPR格式的 \mathbf{p}_6^1 转化为3D+Quat格式的 \mathbf{p}_7^1 ,

然后，进行和运算 $\mathbf{p}_7^R = \mathbf{p}_7^1 \oplus \mathbf{p}_7^2$,

最后位姿和再转化为3D+YPR格式的 \mathbf{p}_6^R .

由链式法则可得:

$$\frac{\partial \mathbf{f}_{pc}(\mathbf{p}_6^1, \mathbf{p}_6^2)}{\partial \mathbf{p}_6^1} = \frac{\partial \mathbf{p}_6^R(\mathbf{p}_7^R)}{\partial \mathbf{p}_7^R} \frac{\partial \mathbf{f}_{pc}(\mathbf{p}_7^1, \mathbf{p}_7^2)}{\partial \mathbf{p}_7^1} \frac{\partial \mathbf{p}_7^1(\mathbf{p}_6^1)}{\partial \mathbf{p}_6^1}\quad (5.3)$$

$$\frac{\partial \mathbf{f}_{pc}(\mathbf{p}_6^1, \mathbf{p}_6^2)}{\partial \mathbf{p}_6^2} = \frac{\partial \mathbf{p}_6^R(\mathbf{p}_7^R)}{\partial \mathbf{p}_7^R} \frac{\partial \mathbf{f}_{pc}(\mathbf{p}_7^1, \mathbf{p}_7^2)}{\partial \mathbf{p}_7^2} \frac{\partial \mathbf{p}_7^2(\mathbf{p}_6^2)}{\partial \mathbf{p}_6^2}\quad (5.4)$$

三个雅可比矩阵具体分别为公式2.12,公式5.8,公式2.8。

在MRPT中实现

通过重载+操作符来实现位姿和运算:

```
#include < mrpt/base.h >
using namespace mrpt::math;
using namespace mrpt::poses;
CPose3D p1, p2;
...
CPose3D p = p1 + p2;
```

通过重载+操作符来实现位姿和运算的不确定度:

```
#include < mrpt/base.h >
using namespace mrpt::poses;
CPose3DPDFGaussian p1(p1_bar, p1_cov);
CPose3DPDFGaussian p2(p2_bar, p2_cov);
...
CPose3DPDFGaussian pR = p1 + p2;
```

5.2 3D+Quat格式描述位姿

5.2.1 位姿和运算

给定两个位姿 $\mathbf{p}_1 = [x_1, y_1, z_1, q_{r1}, q_{x1}, q_{y1}, q_{z1}]^T$, $\mathbf{p}_2 = [x_2, y_2, z_2, q_{r2}, q_{x2}, q_{y2}, q_{z2}]^T$ 位姿的和运算 $\mathbf{p} = \mathbf{p}_1 \oplus \mathbf{p}_2$:

$$\mathbf{p} = \begin{pmatrix} x \\ y \\ z \\ q_r \\ q_x \\ q_y \\ q_z \end{pmatrix} = \mathbf{f}_{\mathbf{qn}} \cdot \mathbf{f}_{\mathbf{qc}}(\mathbf{p}_1, \mathbf{p}_2) = \mathbf{f}_{\mathbf{qn}} \cdot \begin{pmatrix} \frac{\mathbf{f}_{\mathbf{qr}}(\mathbf{p}_1, [\mathbf{x}_2, \mathbf{y}_2, \mathbf{z}_2]^T)}{q_{r1}q_{r2} - q_{x1}q_{x2} - q_{y1}q_{y2} - q_{z1}q_{z2}} \\ q_{r1}q_{x2} + q_{r2}q_{x1} + q_{y1}q_{z2} - q_{y2}q_{z1} \\ q_{r1}q_{y2} + q_{r2}q_{y1} + q_{z1}q_{x2} - q_{z2}q_{x1} \\ q_{r1}q_{z2} + q_{r2}q_{z1} + q_{x1}q_{y2} - q_{x2}q_{y1} \end{pmatrix} \quad (5.5)$$

$\mathbf{f}_{\mathbf{qr}}(\cdot)$ 在公式3.6中定义, $\mathbf{f}_{\mathbf{qn}}$ 是四元数单位化函数, 在1.2.2中讨论过。

5.2.2 不确定度

令 $N(\bar{\mathbf{p}}_1, \text{cov}(\mathbf{p}_1))$ 和 $N(\bar{\mathbf{p}}_2, \text{cov}(\mathbf{p}_2))$ 表示两个3D+Quat格式的独立的高斯分布对。两个位姿和运算的高斯分布 $\mathbf{p} = \mathbf{p}_1 \oplus \mathbf{p}_2$ 通过线性误差传播函数来近似, 平均值:

$$\bar{\mathbf{p}} = \bar{\mathbf{p}}_1 \oplus \bar{\mathbf{p}}_2 \quad (5.6)$$

协方差:

$$\begin{aligned} \text{cov}(\mathbf{p}) &= \frac{\partial \mathbf{f}_{\mathbf{qn}}}{\partial \mathbf{p}_1} \frac{\partial \mathbf{f}_{\mathbf{qc}}(\mathbf{p}_1, \mathbf{p}_2)}{\partial \mathbf{p}_1} \text{cov}(\mathbf{p}_1) \left(\frac{\partial \mathbf{f}_{\mathbf{qc}}(\mathbf{p}_1, \mathbf{p}_2)}{\partial \mathbf{p}_1} \right)^T \left(\frac{\partial \mathbf{f}_{\mathbf{qn}}}{\partial \mathbf{p}_1} \right)^T \\ &+ \frac{\partial \mathbf{f}_{\mathbf{qn}}}{\partial \mathbf{p}_2} \frac{\partial \mathbf{f}_{\mathbf{qc}}(\mathbf{p}_1, \mathbf{p}_2)}{\partial \mathbf{p}_2} \text{cov}(\mathbf{p}_2) \left(\frac{\partial \mathbf{f}_{\mathbf{qc}}(\mathbf{p}_1, \mathbf{p}_2)}{\partial \mathbf{p}_2} \right)^T \left(\frac{\partial \mathbf{f}_{\mathbf{qn}}}{\partial \mathbf{p}_2} \right)^T \end{aligned} \quad (5.7)$$

其中的雅可比矩阵:

$$\left. \frac{\partial \mathbf{f}_{\mathbf{q}\mathbf{c}}(\mathbf{p}_1, \mathbf{p}_2)}{\partial \mathbf{p}_1} \right|_{7 \times 7} = \begin{pmatrix} \left. \frac{\partial \mathbf{f}_{\mathbf{q}\mathbf{r}}(\mathbf{p}_1, [\mathbf{x}_2, \mathbf{y}_2, \mathbf{z}_2]^T)}{\partial \mathbf{p}_1} \right|_{3 \times 7} & \cdot \\ \mathbf{0}_{4 \times 3} & \begin{matrix} q_{r2} & -q_{x2} & -q_{y2} & -q_{z2} \\ q_{x2} & q_{r2} & q_{z2} & -q_{y2} \\ q_{y2} & -q_{z2} & q_{r2} & q_{x2} \\ q_{z2} & q_{y2} & -q_{x2} & q_{r2} \end{matrix} \end{pmatrix} \quad (5.8)$$

$$\left. \frac{\partial \mathbf{f}_{\mathbf{q}\mathbf{c}}(\mathbf{p}_1, \mathbf{p}_2)}{\partial \mathbf{p}_2} \right|_{7 \times 7} = \begin{pmatrix} \left. \frac{\partial \mathbf{f}_{\mathbf{q}\mathbf{r}}(\mathbf{p}_1, [\mathbf{x}_2, \mathbf{y}_2, \mathbf{z}_2]^T)}{\partial [\mathbf{x}_2, \mathbf{y}_2, \mathbf{z}_2]^T} \right|_{3 \times 3} & \cdot & \mathbf{0}_{3 \times 4} \\ \mathbf{0}_{4 \times 3} & \begin{matrix} q_{r1} & -q_{x1} & -q_{y1} & -q_{z1} \\ q_{x1} & q_{r1} & -q_{z1} & q_{y1} \\ q_{y1} & q_{z1} & q_{r1} & -q_{x1} \\ q_{z1} & -q_{y1} & q_{x1} & q_{r1} \end{matrix} \end{pmatrix} \quad (5.9)$$

公式中的偏导雅可比矩阵在公式3.8-3.10中给出了, $\mathbf{f}_{\mathbf{q}\mathbf{n}}$ 是单位化的雅可比矩阵, 在**1.2.2**中给出。

在MRPT中实现

通过重载+操作符来实现位姿和运算:

```
#include <mrpt/base.h>
using namespace mrpt::math;
using namespace mrpt::poses;
CPose3DQuat p1, p2;
...
CPose3DQuat p = p1 + p2;
```

通过重载+操作符来实现位姿和运算的不确定度:

```
#include <mrpt/base.h>
using namespace mrpt::poses;
CPose3DPQuatDFGaussian p1(p1_bar, p1_cov);
CPose3DQuatPDFGaussian p2(p2_bar, p2_cov);
...
CPose3DQuatPDFGaussian pR = p1 + p2;
```

5.3 变换矩阵格式描述位姿

5.3.1 位姿和运算

给定两个6D位姿 \mathbf{p}_1 和 \mathbf{p}_2 对应的 4×4 变换矩阵 \mathbf{M}_1 和 \mathbf{M}_2 ,则位姿和运算 $\mathbf{p} = \mathbf{p}_1 + \mathbf{p}_2$ 为:

$$\mathbf{M} = \mathbf{M}_1 \mathbf{M}_2 \quad (5.10)$$

在MRPT中实现

```
#include <mrpt/base.h>
using namespace mrpt::poses;
CMatrixDouble44 M1, M2;
...
CMatrixDouble44 M=M1*M2;
```


6 位姿的逆

给定一个位姿 \mathbf{p} ，我们定义它的逆也是一个位姿，符号： $\ominus \mathbf{p}$ ，表示与空集做运算。在实际当中，位姿的逆是在位姿坐标系观察原始坐标系得到的位姿。

6.1 3D+YPR格式描述位姿

这种情况下先把位姿变换为3D+Quat或变换矩阵格式，再转化为逆，最后再变换回3D+YPR格式。

在MRPT中实现

```
#include <mrpt/base.h>
using namespace mrpt::poses;
CPose3D q;
CPose3D q-1 = -q;
```

6.2 3D+Quat格式描述位姿

6.2.1 位姿的逆

位姿 $\mathbf{p}_7 = [x, y, z, q_r, q_x, q_y, q_z]^T$ 的逆由两部分组成，可以分别计算。我们定义位姿的逆为 $\mathbf{p}_7^* = [x^*, y^*, z^*, q_r^*, q_x^*, q_y^*, q_z^*]^T$ ，它的旋转部分为原始位姿四元数的共轭，平移部分必须是从位姿 \mathbf{p}_7 看原点 $[0, 0, 0]^T$ 获得的相对位置：

$$\mathbf{p}_7^* = \mathbf{p} = \begin{pmatrix} x^* \\ y^* \\ z^* \\ q_r^* \\ q_x^* \\ q_y^* \\ q_z^* \end{pmatrix} = \mathbf{f}_{\mathbf{qi}}(\mathbf{p}_7) = \begin{pmatrix} \mathbf{f}_{\mathbf{qri}}([0, 0, 0]^T, \mathbf{p}_7) \\ q_r \\ -q_x \\ -q_y \\ -q_z \end{pmatrix} \quad (6.1)$$

其中 $\mathbf{f}_{\mathbf{qri}}([0, 0, 0]^T, \mathbf{p}_7)$ 在公式4.1中定义。

在MRPT中实现

```
#include <mrpt/base.h>
using namespace mrpt::poses;
CPose3DQuat q;
CPose3DQuat q-1 = -q;
```

6.2.2 不确定度

令 $N(\bar{\mathbf{q}}, \text{cov}(\mathbf{q}))$ 表示一个3D+Quat格式的位姿 \mathbf{q} 的高斯分布，则通过线性误差传播函数获得该位姿的逆 $\mathbf{q}_i = \ominus \mathbf{q}$ 的概率分布的平均值：

$$\bar{\mathbf{q}}_i = \ominus \bar{\mathbf{q}} \quad (6.2)$$

协方差:

$$cov(\mathbf{q}_i) = \left(\frac{\partial \mathbf{f}_{\mathbf{q}_i}}{\partial \mathbf{q}} \right) cov(q) \left(\frac{\partial \mathbf{f}_{\mathbf{q}_i}}{\partial \mathbf{q}} \right)^T \quad (6.3)$$

其中雅可比矩阵:

$$\frac{\partial \mathbf{f}_{\mathbf{q}_i}}{\partial \mathbf{q}} = \left(\begin{array}{c|cccc} \frac{\partial \mathbf{f}_{\mathbf{q}_i}([0, 0, 0]^T, \mathbf{q})}{\partial \mathbf{q}} & & & & \\ \hline & 1 & 0 & 0 & 0 \\ & 0 & -1 & 0 & 0 \\ \mathbf{0}_{4 \times 4} & 0 & 0 & -1 & 0 \\ & 0 & 0 & 0 & -1 \end{array} \right) \quad (6.4)$$

其中顶部的子雅可比矩阵在公式4.4中定义。

在MRPT中实现

```
# include < mrpt/base.h >
using namespace mrpt::poses;
CPose3DQuatPDFGaussian q=... ;
CPose3DQuatPDFGaussian q-1 = -q ;
```

6.3 变换矩阵格式描述位姿

本章开始描述位姿的逆提到的空集用矩阵的形式来表达就是 \mathbf{I}_4 ,位姿的逆以矩阵形式来定义是简单的且满足: $\mathbf{M}^{-1}\mathbf{M} = \mathbf{I}$.齐次矩阵的逆可以通过转置它的 3×3 旋转部分来高效的获得, 公式如下:

$$\mathbf{M}^{-1} = \left(\begin{array}{ccc|c} \mathbf{i} & \mathbf{j} & \mathbf{k} & \mathbf{t} \\ 0 & 0 & 0 & 1 \end{array} \right)^{-1} = \left(\begin{array}{ccc|c} i_1 & j_1 & k_1 & x \\ i_2 & j_2 & k_2 & y \\ i_3 & j_3 & k_3 & z \\ 0 & 0 & 0 & 1 \end{array} \right)^{-1} = \left(\begin{array}{ccc|c} i_1 & i_2 & i_3 & -\mathbf{i} \cdot \mathbf{t} \\ j_1 & j_2 & j_3 & -\mathbf{j} \cdot \mathbf{t} \\ k_1 & k_2 & k_3 & -\mathbf{k} \cdot \mathbf{t} \\ 0 & 0 & 0 & 1 \end{array} \right) \quad (6.5)$$

7 位姿变换矩阵的求导

7.1 操作

以下的操作在解决矩阵求导过程中都是非常有用的:

(1)向量操作.将 $M \times N$ 矩阵按 $MN \times 1$ 向量格式存储。例如:

$$\text{vec} \left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \right) = \begin{pmatrix} 1 \\ 4 \\ 2 \\ 5 \\ 3 \\ 6 \end{pmatrix} \quad (7.1)$$

(2)Kronecker积(矩阵直接相乘)。

$M_A \times N_A$ 维的矩阵 \mathbf{A} 和 $M_B \times N_B$ 维的矩阵 \mathbf{B} 的 $M_A M_B \times N_A N_B$ 维的矩阵张量积 $\mathbf{A} \otimes \mathbf{B}$ 为:

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}B & a_{12}B & a_{13}B & \dots \\ a_{21}B & a_{22}B & a_{23}B & \dots \\ \dots & \dots & \dots & \dots \end{pmatrix} \quad (7.2)$$

(3)转置排列矩阵。定义 $\mathbf{T}_{\mathbf{M},\mathbf{N}}$ 为 $MN \times MN$ 维简单排列矩阵，每一行或列只有一个元素为1，其余元素为0。对于 $M \times N$ 的矩阵 \mathbf{A} ,转置排列矩阵满足:

$$\mathbf{T}_{\mathbf{N},\mathbf{M}} \cdot \text{vec}(\mathbf{A}) = \text{vec}(\mathbf{A}^T) \quad (7.3)$$

7.2 符号

在前面的章节中，我们已经讨论了三种描述位姿的格式： $3D + YPR$, $3D + Quat$, 4×4 变换矩阵。接下来，我们主要来讨论下矩阵格式。三维欧氏空间中的任意位姿都可以通过一个 4×4 矩阵来描述:

$$\mathbf{T} = \left(\begin{array}{c|c} \mathbf{R} & \mathbf{t} \\ \hline 0_{1 \times 3} & 1 \end{array} \right) \quad (7.4)$$

其中 $\mathbf{R} \in \mathbf{SO}(3)$ 是一个正向旋转矩阵， $\mathbf{t} = [t_x, t_y, t_z]^T \in \mathbb{R}^3$ 是平移向量。通常任意的 4×4 矩阵都属于一般线性矩阵 $\mathbf{GL}(4, \mathbb{R})$,但是上述格式的矩阵 \mathbf{T} 属于 $\mathbf{SE}(3)$,也就是 $\mathbf{GL}(4, \mathbb{R})$ 中的流形 $\mathbf{SO}(3) \times \mathbb{R}^3$ 。注意这类流形的维度为12: 3×3 矩阵的9个坐标+平移向量的3个坐标。

我们对位姿函数的求导感兴趣， $\mathbf{f}_{\oplus} : \mathbf{SE}(3) \times \mathbf{SE}(3) \rightarrow \mathbf{SE}(3)$ 表示位姿 \mathbf{p}_1 和 \mathbf{p}_2 的和运算，定义位姿矩阵求导符号表达式:

$$\frac{\partial \mathbf{f}_{\oplus}(\mathbf{p}_1, \mathbf{p}_2)}{\partial \mathbf{p}_1} \quad (7.5)$$

什么含义？如果 \mathbf{p}_i 为标量，表达式就是标准的一维求导。如果 \mathbf{p}_i 是向量，则表达式是雅可比矩阵。但是当 \mathbf{p}_i 是位姿时，位姿的参数形式必须是明确约定的，以方便理解该表达式。

位姿用矩阵表示。当求解矩阵的求导问题时，约定矩阵的表示方式为向量操作方式(见7.1)，这样矩阵的求导就变成了标准的雅可比。然后对于描述刚体运动的矩阵来说，我们只需要扩展顶部的 3×4 子矩阵；第4行可以忽略，因为始终不变。

总结：位姿求导问题采用 4×4 矩阵来解决，但是当扩展为向量时，忽略第4行。这样，位姿变成了一个12维的向量。尽管对于6自由度位姿来说是一种过参数表达方式，但在进行线性化时被证明是一种重要且有效的方式，使我们能获取精确的求导结果。

公式7.5拓展为：

$$\frac{\partial \mathbf{f}_{\oplus}(\mathbf{p}_1, \mathbf{p}_2)}{\partial \mathbf{p}_1} = \frac{\partial \mathbf{f}_{\oplus}(\mathbf{T}_1, \mathbf{T}_2)}{\partial \mathbf{T}_1} \Big|_{12 \times 12}. \quad (7.6)$$

定义 $\mathbf{f}_{\oplus}(\mathbf{p}_1, \mathbf{p}_2)$ 结果的矩阵形式为 \mathbf{F} ：

$$\frac{\partial \mathbf{f}_{\oplus}(\mathbf{p}_1, \mathbf{p}_2)}{\partial \mathbf{p}_1} = \frac{\partial \mathbf{F}(\mathbf{P}, \mathbf{Q})}{\partial \mathbf{P}} = \frac{\partial \text{vec}(\mathbf{F}(\mathbf{P}, \mathbf{Q}))}{\partial \text{vec}(\mathbf{P})} \quad (7.8)$$

$$= \frac{\partial [\mathbf{f}_{11}, \mathbf{f}_{21}, \mathbf{f}_{31}, \mathbf{f}_{21}, \mathbf{f}_{22}, \dots, \mathbf{f}_{14}, \mathbf{f}_{24}, \mathbf{f}_{34}]}{\partial [\mathbf{p}_{11}, \mathbf{p}_{21}, \mathbf{p}_{31}, \mathbf{p}_{21}, \mathbf{p}_{22}, \dots, \mathbf{p}_{14}, \mathbf{p}_{24}, \mathbf{p}_{34}]} = \begin{pmatrix} \frac{\partial f_{11}}{\partial p_{11}} & \frac{\partial f_{11}}{\partial p_{21}} & \dots & \frac{\partial f_{11}}{\partial p_{34}} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_{34}}{\partial p_{11}} & \frac{\partial f_{34}}{\partial p_{21}} & \dots & \frac{\partial f_{34}}{\partial p_{34}} \end{pmatrix}_{12 \times 12} \quad (7.9)$$

7.3 有用的表达式

一旦定义好符号，我们将给出以下一系列表达式，当解决优化问题对矩阵的求导时可能会用到。

7.3.1 位姿一位姿和运算

$\mathbf{f}_{\oplus} : \mathbf{SE}(3) \times \mathbf{SE}(3) \rightarrow \mathbf{SE}(3)$ 表示的和运算 $\mathbf{f}_{\oplus}(\mathbf{A}, \mathbf{B}) = \mathbf{A} \oplus \mathbf{B}$ 的求导与位姿 \mathbf{A} 和 \mathbf{B} 相关。如果我们令位姿 \mathbf{X} 定义的 4×4 变换矩阵表示为：

$$\mathbf{T}_{\mathbf{X}} = \left(\begin{array}{c|c} \mathbf{R}_{\mathbf{X}} & \mathbf{t}_{\mathbf{X}} \\ \hline \mathbf{0}_{1 \times 3} & 1 \end{array} \right) \quad (7.10)$$

则矩阵乘积 $\mathbf{T}_{\mathbf{A}}\mathbf{T}_{\mathbf{B}}$ 的元素按向量重新排列，可得求导表达式：

$$\frac{\partial \mathbf{f}_{\oplus}(\mathbf{A}, \mathbf{B})}{\partial \mathbf{A}} = \frac{\partial \mathbf{T}_{\mathbf{A}}\mathbf{T}_{\mathbf{B}}}{\partial \mathbf{T}_{\mathbf{A}}} = \mathbf{T}_{\mathbf{B}}^T \otimes \mathbf{I}_3 \quad (12 \times 12 \quad \text{Jacobian}) \quad (7.11)$$

$$\frac{\partial \mathbf{f}_{\oplus}(\mathbf{A}, \mathbf{B})}{\partial \mathbf{B}} = \frac{\partial \mathbf{T}_{\mathbf{A}}\mathbf{T}_{\mathbf{B}}}{\partial \mathbf{T}_{\mathbf{B}}} = \mathbf{I}_4 \otimes \mathbf{R}_{\mathbf{A}} \quad (12 \times 12 \quad \text{Jacobian}) \quad (7.12)$$

7.3.2 位姿一点和运算

$\mathbf{g}_{\oplus} : \mathbf{SE}(3) \times \mathbf{R}^3 \rightarrow \mathbf{R}^3$ 表示的和运算 $\mathbf{g}_{\oplus}(\mathbf{A}, \mathbf{p}) = \mathbf{A} \oplus \mathbf{p}$ 的求导与位姿 \mathbf{A} 和 \mathbf{p} 相关。我们可以获得求导公式：

$$\frac{\partial \mathbf{f}_{\oplus}(\mathbf{A}, \mathbf{p})}{\partial \mathbf{p}} = \frac{\partial \mathbf{T}_{\mathbf{A}}\mathbf{p}}{\partial \mathbf{p}} = \frac{\partial (\mathbf{R}_{\mathbf{A}}\mathbf{p} + \mathbf{t}_{\mathbf{A}})}{\partial \mathbf{p}} = \mathbf{R}_{\mathbf{A}} \quad (3 \times 3 \quad \text{Jacobian}) \quad (7.13)$$

$$\frac{\partial \mathbf{f}_{\oplus}(\mathbf{A}, \mathbf{p})}{\partial \mathbf{A}} = \frac{\partial \mathbf{T}_{\mathbf{A}} \mathbf{p}}{\partial \mathbf{T}_{\mathbf{A}}} = \begin{pmatrix} \mathbf{p}^T & 1 \end{pmatrix} \otimes \mathbf{I}_3 \quad (3 \times 12 \quad \text{Jacobian}) \quad (7.14)$$

7.3.3 位姿的逆

位姿的逆(见6.3)等于该位姿矩阵的逆，相应的求导公式：

$$\frac{\partial (\mathbf{T}_{\mathbf{A}}^{-1})}{\partial \mathbf{T}_{\mathbf{A}}} = \begin{pmatrix} \mathbf{T}_{3,3} & \mathbf{0}_{3 \times 9} \\ \mathbf{I}_3 \otimes (-\mathbf{t}_{\mathbf{A}}^T) & -\mathbf{R}_{\mathbf{A}}^T \end{pmatrix} \quad (12 \times 12 \quad \text{Jacobian}) \quad (7.15)$$

记住 $\mathbf{T}_{3,3}$ 代表 9×9 的转置排列矩阵(7.1节中定义)。

7.3.4 位姿与点的差运算

采用以上定义的求导雅可比矩阵和标准的链式法则，可以获得复杂抽象的雅可比矩阵。给定一个位姿 \mathbf{A} 和一个点 \mathbf{p} ，位姿-点差运算 $\mathbf{p} \ominus \mathbf{A} = \mathbf{A}^{-1} \mathbf{p}$ 求导：

$$\frac{\partial (\mathbf{T}_{\mathbf{A}}^{-1} \mathbf{p})}{\partial \mathbf{p}} \stackrel{\text{Eq 7.13}}{=} \mathbf{R}_{\mathbf{A}}^{-1} = \mathbf{R}_{\mathbf{A}}^T \quad (3 \times 3 \quad \text{Jacobian}) \quad (7.16)$$

$$\frac{\partial (\mathbf{T}_{\mathbf{A}}^{-1} \mathbf{p})}{\partial \mathbf{T}_{\mathbf{A}}} = \frac{\partial (\mathbf{T}_{\mathbf{A}}^{-1} \mathbf{p})}{\partial \mathbf{T}_{\mathbf{A}}^{-1}} \frac{\partial (\mathbf{T}_{\mathbf{A}}^{-1})}{\partial \mathbf{T}_{\mathbf{A}}} \stackrel{\text{Eq 7.14, 7.15}}{=} \left[\begin{pmatrix} \mathbf{p}^T & 1 \end{pmatrix} \otimes \mathbf{I}_3 \right] \begin{pmatrix} \mathbf{T}_{3,3} & \mathbf{0}_{3 \times 9} \\ \mathbf{I}_3 \otimes (-\mathbf{t}_{\mathbf{A}}^T) & -\mathbf{R}_{\mathbf{A}}^T \end{pmatrix} \quad (7.17)$$

$$= \left(\mathbf{I}_3 \otimes ((\mathbf{p} - \mathbf{t}_{\mathbf{A}})^T) \mid -\mathbf{R}_{\mathbf{A}}^T \right) \quad (3 \times 12 \quad \text{Jacobian}) \quad (7.18)$$

8 李群的概念

8.1 定义

在讨论把刚体运动作为李群的实际应用之前，我们需要给出几个数学定义，它们是后续讨论的基础。

8.1.1 群

群 G 是一种包含有限或无限个元素和一些二元操作(群操作)的结构，两个群元素 $A, B \in G$ 的积记为 AB 。群的定义满足如下条件：

1. 封闭性. 群操作是一个函数 $G \times G \rightarrow G$. 即 $A, B \in G$, 则 $AB \in G$.
2. 结合率. $A, B, C \in G$, $(AB)C = A(BC)$.
3. 单位元素. 必须存在一个单位元素 $I \in G$, 对任意的 $A \in G$ 满足 $IA = AI = A$.
4. 可逆. 对于任意的 $A \in G$ 必须存在一个逆元素 A^{-1} 满足 $AA^{-1} = A^{-1}A = I$.

简单群的例子：

- (1) 整数群 \mathbb{Z} ，满足加法条件(二元操作)。
- (2) 可逆的 $N \times N$ 矩阵集合 $\mathbf{GL}(N, \mathbb{R})$ 或3D特殊正交群 $\mathbf{SO}(3)$ 是满足标准矩阵乘法操作的群。

8.1.2 流形

一个 N -维流形 M 是一个拓扑空间，其中每个点 $p \in M$ 天生具备欧氏空间的局部结构。换句话说：每个点的邻域都与 \mathbb{R}^N 同构. 直观的角度来解释就是：一个点的无穷小邻域空间是平滑的。地球的表面就是一个2维的流形，我们在垂直的两个方向(南北、东西)上移动。尽管地球的表面是弯曲的，但在给定的一点可以看做是平滑的或一个2维欧氏空间(平面). 如图5所示。

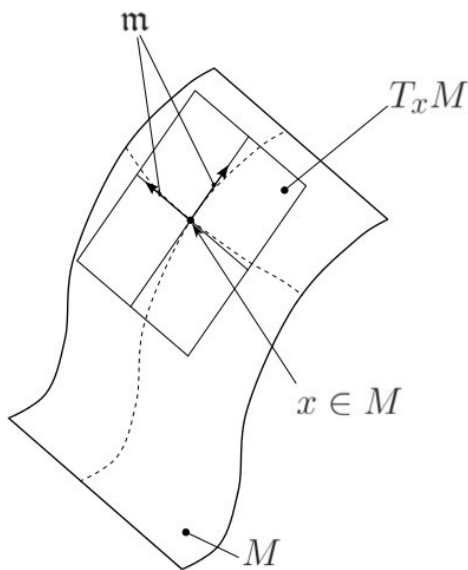


图 5: 2D简单流形 M (嵌入在3D空间中), 流形中一点 x , 在点 x 处的切空间 $T_x M$, 切空间的向量基: 李代数 \mathfrak{m} .

8.1.3 N维空间的平滑流形

一个D-维流形是N维空间的平滑流形($N \geq D$),如果每个点 $\mathbf{p} \in \mathbf{M}$ 被 $\mathbf{U} \subseteq \mathbf{M}$ 包含, 定义如下函数:

$$\varphi : \Omega \rightarrow \mathbf{U} \quad (8.1)$$

$$\mathbb{R}^N \rightarrow \mathbf{M} \quad (8.2)$$

其中 Ω 是 \mathbb{R}^N 的开子集, 包括了切空间的原点($\mathbf{0}_N$)。

函数 φ 必须满足:

- 1.是同构的.(例如: $\varphi(\cdot)$ 和 $\varphi(\cdot)^{-1}$ 是连续的.)
- 2.是平滑的(C^∞).
- 3.在原点的导数 $\varphi'(\mathbf{0}_N)$ 必须是内射的.

函数 $\varphi(\cdot)$ 是 \mathbf{M} 以点 \mathbf{p} 为中心的参数化。

$$\varphi(\mathbf{0}_N) = \mathbf{p} \quad , \quad \mathbf{p} \in \mathbf{M} \quad (8.3)$$

逆函数:

$$\varphi^{-1} : \mathbf{U} \rightarrow \Omega \quad (8.4)$$

$$\mathbf{M} \rightarrow \mathbb{R}^N \quad (8.5)$$

被称为 \mathbf{M} 的局部图, 因为提供了流形所在区域的扁平的描述。

8.1.4 流形的切空间

N维空间的一个D-维流形在每个点 $p \in M$ 有一个相应的N-维切空间, 这个空间用 $T_x M$ 表示, 空间每个非奇异点有D个维度, 如图8.1所示.

通俗的讲, 切空间 $T_x M$ 可以被看做是点 p 处受流形约束的所有曲面求导的向量空间,

例如: $T_x M$ 包括 p 点处粒子所有可能的速度矢量。

8.1.5 李群

李群是N维空间群 G 的一个子集, 满足:

- 1.是一个群。
- 2.是N维空间的一个流形。
- 3.群乘积和取逆操作都是平滑函数。

8.1.6 线性李群 (矩阵群)

$N \times N$ 矩阵记为 $\mathbf{M}(N, \mathbb{R})$.对于 $A, B \in \mathbf{M}(N, \mathbb{R})$ 定义操作符 $[\cdot, \cdot]$ 满足: $[A, B] = AB - BA$.

定理 1: $\mathbf{GL}(N, \mathbb{R})$ 的一个封闭的子群 G 是一个线性李群, 因此在 \mathbb{R}^{N^2} 空间上有一个平滑流形。集合 \mathbf{g} :

$$\mathbf{g} = \{\mathbf{X} \in \mathbf{M}(N, \mathbb{R}) | e^{t\mathbf{X}} \in G, \forall t \in \mathbb{R}\} \quad (8.6)$$

是一个向量空间等价于 $T_I G$ (流形 G 在点 I 的切空间).集合 \mathfrak{g} 对操作 $[\cdot, \cdot]$ 是封闭的。

对于任意的方阵 M ，指数映射 e^M 的定义与矩阵指数完全一致，通常有如下幂级数形式：

$$e^M = \sum_{k=0}^{\infty} \frac{1}{k!} M^k \quad (8.7)$$

根据上述理论， $\mathbf{SO}(3)$ 群可以被看做是李群，因为它是 $\mathbf{GL}(3, \mathbb{R})$ 的一个子群.刚体变换群 $\mathbf{SE}(3)$ 也是一个李群，因为它与 $\mathbf{GL}(4, \mathbb{R})$ 的一个子群是同构的。

8.1.7 李代数

李代数是满足 $[\cdot, \cdot]$ 操作的一种代数： $\mathfrak{m} \times \mathfrak{m} \rightarrow \mathfrak{m}$,对于任意的 $a, b, c \in \mathfrak{m}$ 满足：

$$[a, b] = -[b, a] \quad (8.8)$$

$$[c, [a, b]] = [[c, a], b] + [a, [c, b]] \quad (8.9)$$

对于任意的 $a \in \mathfrak{m}$ 满足 $[a, a] = 0$.

一个重要的事实是李代数是李群 M 在单位元素 I 处的切空间：

$$\mathfrak{m} = T_I M \quad (8.10)$$

8.1.8 李群的指数和对数映射

李群和其李代数有两个重要的映射：

指数映射：从李代数向流形映射，确定流形的局部结构：

$$\exp : \mathfrak{m} \rightarrow M \quad (8.11)$$

对数映射：从流形映射向李代数映射：

$$\ln : M \rightarrow \mathfrak{m} \quad (8.12)$$

9 李群SE(3)

9.1 特性

我们回顾下7.2中的刚体变换群SE(3)，它的成员是 4×4 的矩阵，结构如下：

$$\mathbf{T} = \left(\begin{array}{c|c} \mathbf{R} & \mathbf{t} \\ \hline 0_{1 \times 3} & 1 \end{array} \right) \quad (9.1)$$

其中 $\mathbf{R} \in \mathbf{SO}(3)$, $\mathbf{t} = [\mathbf{t}_x, \mathbf{t}_y, \mathbf{t}_z]^T \in \mathbb{R}^3$, 群的乘积操作是标准矩阵乘积。

SE(3)的一些事实：

- (1) SE(3)是一个6D流形。包含3D平移和3D旋转。
 - (2) SE(3)与GL(4, \mathbb{R})的子集同构。
 - (3) 作为流形，SE(3)与SO(3) \times \mathbb{R}^3 异构，因为SO(3) \times \mathbb{R}^3 用12个坐标描述。
 - (4) 作为群，SE(3)与SO(3) \times \mathbb{R}^3 不是同构的，因为两者的群乘积是不同的。
- SE(3)群是SO(3) \times \mathbb{R}^3 群的一个半直积。

9.2 SO(3)的李代数

因为SE(3)具有SO(3) \times \mathbb{R}^3 乘积的流形结构，所以SO(3)是一个李群。李群SO(3)有一个相关的李代数so(3), 由3个基本的反对称矩阵组成，每一个矩阵表示绕轴做无穷小的旋转：

$$\begin{aligned} \mathbf{so}(3) = \{\mathbf{G}_i^{\mathbf{so}(3)}\}_{i=1,2,3} \quad : \quad \mathbf{G}_1^{\mathbf{so}(3)} &= \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}_{\times} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \\ \mathbf{G}_2^{\mathbf{so}(3)} &= \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}_{\times} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} \quad \mathbf{G}_3^{\mathbf{so}(3)} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}_{\times} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{aligned} \quad (9.5)$$

所以代数so(3)有三个坐标(每个坐标乘以相应的反对称矩阵)，可以表示为 \mathbb{R}^3 中的一个向量。

我们引入反对称矩阵操作符 $[\cdot]_{\times}$, 定义如下：

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\times} = \begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix} \quad (9.7)$$

其反向操作被定义为 $[\cdot]_{\nabla}$:

$$\left[\begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix} \right]_{\nabla} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (9.8)$$

9.3 SE(3)的李代数

SE(3)对应的李代数是se(3),基是6个 4×4 矩阵, 每一个表示绕轴无穷小旋转或沿轴无穷小平移:

$$\begin{aligned} \mathbf{se}(\mathbf{3}) = \{\mathbf{G}_i^{\mathbf{se}(\mathbf{3})}\}_{i=1,2,3,4,5,6} \quad ; \quad \mathbf{G}_{\{1,2,3\}}^{\mathbf{se}(\mathbf{3})} &= \left(\begin{array}{c|c} \mathbf{G}_{\{1,2,3\}}^{\mathbf{so}(\mathbf{3})} & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \\ \hline 0 & 0 \end{array} \right) \quad \mathbf{G}_4^{\mathbf{se}(\mathbf{3})} = \left(\begin{array}{c|c} \mathbf{0}_{3 \times 3} & \begin{matrix} 1 \\ 0 \\ 0 \end{matrix} \\ \hline 0 & 0 \end{array} \right) \\ \mathbf{G}_5^{\mathbf{se}(\mathbf{3})} &= \left(\begin{array}{c|c} \mathbf{0}_{3 \times 3} & \begin{matrix} 0 \\ 1 \\ 0 \end{matrix} \\ \hline 0 & 0 \end{array} \right) \quad \mathbf{G}_6^{\mathbf{se}(\mathbf{3})} = \left(\begin{array}{c|c} \mathbf{0}_{3 \times 3} & \begin{matrix} 0 \\ 0 \\ 1 \end{matrix} \\ \hline 0 & 0 \end{array} \right) \end{aligned} \quad (9.9)$$

se(3)有6个坐标(每个坐标乘以相应的矩阵基), 可表示为 \mathbb{R}^6 中的一个向量, 也被称作流形se(3)的线性化。

9.4 指数和对数映射

与8.1.8定义相同, 指数映射和对数映射在李群和李代数之间进行参数变换。本章中, 我们有时用exp和ln来表示对向量进行操作和变换为向量的操作。这些向量是相应的李代数表示的矩阵向量空间的坐标。

9.4.1 SO(3)

$$\text{指数映射: } \exp : \mathbf{so}(\mathbf{3}) \rightarrow \mathbf{SO}(\mathbf{3}) \quad (9.14)$$

$$\boldsymbol{\omega} \rightarrow \mathbf{R}_{3 \times 3} \quad (9.15)$$

是满射切对应于矩阵指数, 根据罗德里格斯公式得到封闭的解:

$$e^{\boldsymbol{\omega}} \equiv e^{[\boldsymbol{\omega}]_{\times}} = \mathbf{I}_3 + \frac{\sin \theta}{\theta} [\boldsymbol{\omega}]_{\times} + \frac{1 - \cos \theta}{\theta^2} [\boldsymbol{\omega}]_{\times}^2 \quad (9.16)$$

这里角度 $\theta = |\theta|$, $[\boldsymbol{\omega}]_{\times}$ 是3维向量 $\boldsymbol{\omega}$ 生成的反对称矩阵。

$$\text{对数映射: } \ln : \mathbf{SO}(\mathbf{3}) \rightarrow \mathbf{so}(\mathbf{3}) \quad (9.17)$$

$$\mathbf{R}_{3 \times 3} \rightarrow \boldsymbol{\omega} \quad (9.18)$$

是满射的, 也是上述exp函数的逆。它对应于 3×3 旋转矩阵的对数, 由罗德里格斯公式得:

$$\begin{aligned} \ln(\mathbf{R}) &= \frac{\theta}{2 \sin \theta} (\mathbf{R} - \mathbf{R}^T) \\ \cos \theta &= \frac{\text{tr}(\mathbf{R}) - 1}{2} \\ \boldsymbol{\omega} &= [\ln(\mathbf{R})]_{\nabla} \end{aligned} \quad (9.19)$$

9.4.2 SE(3)

指数映射 令

$$\mathbf{v} = \begin{pmatrix} \mathbf{t} \\ \boldsymbol{\omega} \end{pmatrix} \quad (9.20)$$

表示李代数 $\mathfrak{se}(3)$ 的6个坐标，分为两部分： \mathbf{t} 定义平移， $\boldsymbol{\omega}$ 定义旋转，我们定义 4×4 矩阵：

$$\mathbf{A}(\mathbf{v}) = \begin{pmatrix} [\boldsymbol{\omega}]_{\times} & \mathbf{t} \\ 0 & 0 \end{pmatrix} \quad (9.21)$$

所以映射：

$$\exp : \mathfrak{se}(3) \rightarrow \mathbf{SE}(3) \quad (9.22)$$

是满射的，封闭格式：

$$e^{\mathbf{v}} \equiv \mathbf{e}^{\mathbf{A}(\mathbf{v})} = \begin{pmatrix} e^{[\boldsymbol{\omega}]_{\times}} & \mathbf{V}\mathbf{t} \\ 0 & 1 \end{pmatrix} \quad (9.23)$$

$$\mathbf{V} = \mathbf{I}_3 + \frac{1 - \cos \theta}{\theta^2} [\boldsymbol{\omega}]_{\times} + \frac{\theta - \sin \theta}{\theta^3} [\boldsymbol{\omega}]_{\times}^2 \quad (9.24)$$

对数映射

$$\ln : \mathbf{SE}(3) \rightarrow \mathfrak{se}(3) \quad (9.25)$$

$$\mathbf{A}(\mathbf{v}) \rightarrow \mathbf{v}$$

所以计算求解： $\mathbf{v} = \begin{pmatrix} \mathbf{t}' \\ \boldsymbol{\omega} \end{pmatrix}$

$$\boldsymbol{\omega} = [\ln \mathbf{R}]_{\nabla} \quad (\text{see } Eq9.19) \quad (9.26)$$

$$\mathbf{t}' = \mathbf{V}^{-1}\mathbf{t} \quad (\text{see } Eq9.24) \quad (9.27)$$

其中 \mathbf{R} 和 \mathbf{t} 是 $\mathbf{SE}(3)$ 的旋转和平移部分。

9.5 MRPT中实现

类`mrpt::poses::CPose3D`实现 $\mathbf{SO}(3)$ 、 $\mathbf{SE}(3)$ 的指数和对数映射。

指数

```
#include < mrpt/base.h >
using namespace mrpt::poses;
using namespace mrpt::math;
CArrayDouble< 3 > so3_vec;
CArrayDouble< 6 > se3_vec;
CPose3D rot = CPose3D::exp(so3_vec);
CPose3D pose = CPose3D::exp(se3_vec);
```

对数

```
#include < mrpt/base.h >
using namespace mrpt::poses;
using namespace mrpt::math;
CPose3D p = ... ;
CArrayDouble< 3 > so3_vec = p.ln_rotation();
CArrayDouble< 6 > se3_vec = p.ln();
```

10 SE(3)的优化问题

处理SE(3)需要的流形概念在前面的章节中已经给出。本章我们来关注下理论研究的最终目的：解决估计SE(3)位姿的实际数值问题。

10.1 平面欧氏空间的优化问题求解

梯度下降、高斯-牛顿、列文伯格-马夸特和卡尔曼类滤波器都是无价的方法，它们的核心执行相同的操作：迭代求取状态变量 \mathbf{x} ，以便估计值与观测值 \mathbf{z} 的误差平方和最小。

对于我们的目标来说使用哪种方法解决问题无所谓。在优化的某个阶段用到的预测函数 $\mathbf{f}(\mathbf{x})$ 是所有要使用的信息。目标是 minimized 从预测到观测的误差平方和：

$$\mathbf{S}(\mathbf{x}) = (\mathbf{f}(\mathbf{x}) - \mathbf{z})^T (\mathbf{f}(\mathbf{x}) - \mathbf{z}) = \|\mathbf{f}(\mathbf{x}) - \mathbf{z}\|^2 \quad (10.1)$$

状态量 \mathbf{x} 采用增量方式获得：

$$\mathbf{x} \leftarrow \mathbf{x} + \boldsymbol{\delta} \quad (10.2)$$

增量 $\boldsymbol{\delta}$ 通过求解以下方程获得：

$$\left. \frac{\partial \mathbf{S}(\mathbf{x} + \boldsymbol{\delta})}{\partial \boldsymbol{\delta}} \right|_{\boldsymbol{\delta}=\mathbf{0}} = \mathbf{0}. \quad (10.3)$$

函数 $\mathbf{S}(\cdot)$ 的导数为 0 意味着最小的误差平方和。注意雅可比矩阵是在当前估计值 \mathbf{x} 的附近 $\boldsymbol{\delta} = \mathbf{0}$ 处估计的。通常公式10.3和10.2迭代直到收敛或最大迭代次数为止。

当SE(3)位姿作为状态变量 \mathbf{x} 的一部分被估计时，无论使用那种方法总会遇到的一个问题是：所有的方法都是被设计用于平面欧氏空间的。如果我们想优化一个包含位姿的状态变量，我们必须把参数按照向量来存储，也就是说：

1. 3D+YPR: \mathbf{x} 中每个位姿包含6个元素。
2. 3D+Quat: \mathbf{x} 中每个位姿包含7个元素。
3. 4×4 矩阵: \mathbf{x} 中每个位姿包含16个元素。
4. 3×4 子矩阵: \mathbf{x} 中每个位姿包含12个元素。

没有任何一种表示方式能够得到理想解，并且某些方式还是比较糟糕的：

- 1.第一种方式存储要求最低，但对于所有位姿-位姿链式操作可能不是都有封闭的雅可比；更新规则 $\mathbf{x} \leftarrow \mathbf{x} + \boldsymbol{\delta}$ 意味着可能超出三个角度的有效值(需要重新对状态变量进行单位化)。而且当一个自由度丢失后，还可能存在万向节死锁问题。当有3个自由度时，优化方法可能沿着解退化的方向移动并卡死。
- 2.第二种方式雅可比总是存在，但多出的一个自由度会导致上面提到的问题。
- 3.第三、四种方式，雅可比始终存在，但多出的几个额外自由度使问题更加糟糕。

总结：存储状态变量中的位姿并试图优化它们不是一个好主意。尽管3D+Quat参数格式糟糕程度较低，仍然可以使用(在计算机视觉中获得了比较好的结果)，但下一节将介绍一种更为鲁棒和通用的方法。

10.2 一种更为恰当的方式：在流形上进行优化

在状态变量中存储3D+YPR或3D+Quat格式参数，在流形上进行优化不是一种新思想，但最近几年在机器人和计算机视觉社区当中变得日益普及。

$$\delta^* \leftarrow \left. \frac{\partial \mathbf{S}(\mathbf{x} + \delta)}{\partial \delta} \right|_{\delta=\mathbf{0}} = \mathbf{0} \implies \epsilon^* \leftarrow \left. \frac{\partial \mathbf{S}(\mathbf{x} \boxplus \epsilon)}{\partial \epsilon} \right|_{\epsilon=\mathbf{0}} = \mathbf{0}. \quad (10.4)$$

$$\mathbf{x} \leftarrow \mathbf{x} + \delta \implies \mathbf{x} \leftarrow \mathbf{x} \boxplus \epsilon^* \quad (10.5)$$

其中 $\mathbf{m} \in \mathbf{M}$ 是问题的状态变量，处于N-维流形M上， $\epsilon \in \mathbb{R}^N$ 是在 \mathbf{x} (用M的李代数作为向量基)周围流形线性化的增量，操作符 $\boxplus: M \times \mathbb{R}^N \rightarrow M$ 是标准+操作符的欧氏空间的泛化。

实现“ \boxplus ”有两种有效方式：令 $\mathbf{x}, \mathbf{x}' \in \mathbf{M}$ 是问题M的流形的元素， $\epsilon \in \mathbb{R}^N$ 是线性近似的增量，则：

$$\mathbf{x}' = \mathbf{x} \boxplus \epsilon \implies \mathbf{x}' = e^\epsilon \mathbf{x} \quad (10.6)$$

$$\mathbf{x}' = \mathbf{x} \boxplus \epsilon \implies \mathbf{x}' = \mathbf{x} e^\epsilon \quad (10.7)$$

$\mathbf{x}e^\epsilon$ 和 $e^\epsilon\mathbf{x}$ 是流形群操作定义的“积”， e^ϵ 是李群M的指数映射。需要强调的是 \mathbf{x} 的拓扑结构可能是许多子拓扑结构的积(例如：存储2个3D点和3个 $\mathbf{SE}(3)$ 位姿给出一个 $\mathbb{R}^3 \times \mathbb{R}^3 \times \mathbf{SE}(3) \times \mathbf{SE}(3) \times \mathbf{SE}(3)$ 的结构)。因此，如果估计的向量包含欧氏空间的一部分，群积将倒退为一般性加法(因为它称为了最初的优化方法)。

以上两种方法唯一的差别是优化过程中的中间变量 ϵ^* 值不同。在优化过程中带 $*$ ，在更新过程 $\epsilon \leftarrow \epsilon \boxplus \epsilon^*$ 中去掉 $*$ 。两种方法是等价的，下一部分将采用左乘($e^\epsilon\mathbf{x}$)作为惯例约定。

10.3 有用的流形求导

当采用流形优化方法时，以下的几个雅可比可能会在优化问题中出现。公式+雅可比链式法则足以获得机器人和计算机视觉领域大多数优化问题的有效表达式。

阅读本章节之前，确保已经看过了第7章的矩阵求导的符号约定。

10.3.1 $\mathbf{SE}(3)$ e^ϵ 的雅可比

应在几乎所有流形优化过程中出现，所以 e^ϵ 是最基本的雅可比。如前文解释的那样求导的条件是 $\epsilon = \mathbf{0}$ 。

在李代数坐标 $\epsilon = \mathbf{0}$ 处，推导指数映射得：

$$\left. \frac{\partial e^\epsilon}{\partial \epsilon} \right|_{\epsilon=\mathbf{0}} = \begin{pmatrix} \mathbf{0}_{3 \times 3} & -[\mathbf{e}_1]_\times \\ \mathbf{0}_{3 \times 3} & -[\mathbf{e}_2]_\times \\ \mathbf{0}_{3 \times 3} & -[\mathbf{e}_3]_\times \\ \mathbf{I}_3 & \mathbf{0}_{3 \times 3} \end{pmatrix} \quad (12 \times 6 \quad \text{Jacobian}) \quad (10.10)$$

其中 $\mathbf{e}_1 = [1, 0, 0]^T$ ， $\mathbf{e}_2 = [0, 1, 0]^T$ ， $\mathbf{e}_3 = [0, 0, 1]^T$ 。注意得到的雅可比矩阵顺序是按照Eq.9.20来排列的。这是本章真正的新的雅可比矩阵，下面的几个可以通过第7章和链式法则来获得。

10.3.2 SO(3)对数的雅可比

在某个点讨论 3×3 旋转矩阵并计算它的对数映射时，如果需要对一个函数求导，那么对数雅可比将会出现。例如在**Graph – SLAM**中优化位姿图时。

给定一个输入旋转矩阵 \mathbf{R} ,所以:

$$\left. \frac{d\ln(\mathbf{R})}{d\mathbf{R}} \right|_{3 \times 9} = \begin{cases} \left(\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & -\frac{1}{2} & 0 & 0 \end{array} \middle| \begin{array}{ccc} 0 & -\frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 \end{array} \right) , if \quad \cos \theta > 0.9999... \\ \left(\begin{array}{ccc|ccc} a_1 & 0 & 0 & 0 & a_1 & 0 \\ a_2 & 0 & -b & 0 & a_2 & 0 \\ a_3 & b & 0 & -b & a_3 & 0 \end{array} \middle| \begin{array}{ccc} 0 & -b & a_1 \\ b & 0 & a_2 \\ 0 & 0 & a_3 \end{array} \right) , otherwise \end{cases} \quad (10.11)$$

其中 \mathbf{R} 的9部分是按照列优先来排列的，并且:

$$\begin{aligned} \cos \theta &= \frac{\text{tr}(\mathbf{R}) - 1}{2} \\ \sin \theta &= \sqrt{1 - \cos^2 \theta} \\ \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} &= [\mathbf{R} - \mathbf{R}^T]_{\nabla} \frac{\theta \cos \theta - \sin \theta}{4 \sin^3 \theta} = \begin{bmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \end{bmatrix} \frac{\theta \cos \theta - \sin \theta}{4 \sin^3 \theta} \\ \mathbf{b} &= \frac{\theta}{2 \sin \theta} \end{aligned}$$

10.3.3 $\mathbf{e}^\varepsilon \oplus \mathbf{D}$ 的雅可比

令 $\mathbf{D} \in \mathbf{SE}(3)$ 是一个位姿，它的变换矩阵为:

$$\mathbf{T}(\mathbf{D}) = \left(\begin{array}{ccc|c} \mathbf{d}_{c1} & \mathbf{d}_{c2} & \mathbf{d}_{c3} & \mathbf{d}_t \\ 0 & 0 & 0 & 1 \end{array} \right) \quad (10.12)$$

根据左-和运算约定，对于无穷小位姿，可得雅可比:

$$\left. \frac{\partial e^\varepsilon \mathbf{D}}{\partial \varepsilon} \right|_{\varepsilon=0} = \left. \frac{\partial \mathbf{A} \mathbf{D}}{\partial \mathbf{A}} \right|_{\mathbf{A}=\mathbf{I}_4=e^\varepsilon} \cdot \left. \frac{\partial e^\varepsilon}{\partial \varepsilon} \right|_{\varepsilon=0} \quad (10.13)$$

$$= [\mathbf{T}(\mathbf{D})^T \oplus \mathbf{I}_3] \left. \frac{\partial e^\varepsilon}{\partial \varepsilon} \right|_{\varepsilon=0} \quad (10.14)$$

$$= \left(\begin{array}{ccc|c} \mathbf{0}_{3 \times 3} & -[\mathbf{d}_{c1}]_{\times} \\ \mathbf{0}_{3 \times 3} & -[\mathbf{d}_{c2}]_{\times} \\ \mathbf{0}_{3 \times 3} & -[\mathbf{d}_{c3}]_{\times} \\ \mathbf{I}_3 & -[\mathbf{d}_t]_{\times} \end{array} \right) \quad (12 \times 6 \quad \text{Jacobian}) \quad (10.15)$$

10.3.4 $\mathbf{D} \oplus \mathbf{e}^\varepsilon$ 的雅可比

令 $\mathbf{D} \in \mathbf{SE}(3)$ 是一个位姿，它的变换矩阵为：

$$\mathbf{T}(\mathbf{D}) = \left(\begin{array}{ccc|c} \mathbf{d}_{c1} & \mathbf{d}_{c2} & \mathbf{d}_{c3} & \mathbf{d}_t \\ 0 & 0 & 0 & 1 \end{array} \right) = \left(\begin{array}{c|c} \mathbf{R}(\mathbf{D}) & \mathbf{d}_t \\ \hline 0 & 1 \end{array} \right) \quad (10.16)$$

可以得到雅可比：

$$\left. \frac{\partial \mathbf{D} \mathbf{e}^\varepsilon}{\partial \varepsilon} \right|_{\varepsilon=0} = \left. \frac{\partial \mathbf{D} \mathbf{I}_4}{\partial \mathbf{I}_4} \frac{\partial \mathbf{e}^\varepsilon}{\partial \varepsilon} \right|_{\varepsilon=0} \quad (10.17)$$

$$= [\mathbf{T}(\mathbf{D})^T \oplus \mathbf{I}_3] \left(\begin{array}{c|c} \mathbf{0}_{3 \times 3} & -[\mathbf{e}_1]_\times \\ \mathbf{0}_{3 \times 3} & -[\mathbf{e}_2]_\times \\ \mathbf{0}_{3 \times 3} & -[\mathbf{e}_3]_\times \\ \hline \mathbf{I}_3 & \mathbf{0}_{3 \times 3} \end{array} \right) \quad (10.18)$$

$$= \left(\begin{array}{c|ccc} & 0_{3 \times 1} & -\mathbf{d}_{c3} & -\mathbf{d}_{c2} \\ \mathbf{0}_{9 \times 3} & \mathbf{d}_{c3} & 0_{3 \times 1} & -\mathbf{d}_{c1} \\ & -\mathbf{d}_{c2} & \mathbf{d}_{c1} & 0_{3 \times 1} \\ \hline \mathbf{R}(\mathbf{D}) & & 0_{3 \times 3} & \end{array} \right) \quad (12 \times 6 \quad \text{Jacobian}) \quad (10.19)$$

10.3.5 $\mathbf{e}^\varepsilon \oplus \mathbf{D} \oplus \mathbf{p}$ 的雅可比

在Bundle Adjustment [2]实现中需要这种位姿 \mathbf{D} 与点 \mathbf{p} 的和运算操作(点相对于相机是 $\mathbf{D} \oplus \mathbf{p}$, \mathbf{D} 是相机实际位姿的逆)。

令 $\mathbf{p} \in \mathbb{R}^3$ 是一个3D点， $\mathbf{D} \in \mathbf{SE}(3)$ 是一个位姿且变换矩阵是：

$$\mathbf{T}(\mathbf{D}) = \left(\begin{array}{ccc|c} \mathbf{d}_{c1} & \mathbf{d}_{c2} & \mathbf{d}_{c3} & \mathbf{d}_t \\ 0 & 0 & 0 & 1 \end{array} \right) = \left(\begin{array}{c|c} \mathbf{R}(\mathbf{D}) & \mathbf{d}_t \\ \hline 0 & 1 \end{array} \right) \quad (10.20)$$

可求得雅可比矩阵：

$$\left. \frac{\partial (\mathbf{e}^\varepsilon \mathbf{D}) \oplus \mathbf{p}}{\partial \varepsilon} \right|_{\varepsilon=0} = \left. \frac{\partial \mathbf{A} \oplus \mathbf{p}}{\partial \mathbf{A}} \right|_{\mathbf{A}=\mathbf{e}^\varepsilon \mathbf{D}=\mathbf{D}} \cdot \left. \frac{\partial \mathbf{e}^\varepsilon \mathbf{D}}{\partial \varepsilon} \right|_{\varepsilon=0} \quad (10.21)$$

$$= ((\mathbf{p}^T \quad \mathbf{1}) \otimes \mathbf{I}_3) \left(\begin{array}{c|c} 0_{3 \times 3} & -[\mathbf{d}_{c1}]_\times \\ 0_{3 \times 3} & -[\mathbf{d}_{c2}]_\times \\ 0_{3 \times 3} & -[\mathbf{d}_{c3}]_\times \\ \hline \mathbf{I}_3 & -[\mathbf{d}_t]_\times \end{array} \right) \quad (10.22)$$

$$= (\mathbf{I}_3 \quad -[\mathbf{D} \oplus \mathbf{p}]_\times) \quad (3 \times 6 \quad \text{Jacobian}) \quad (10.23)$$

10.3.6 $\mathbf{p} \ominus (\mathbf{e}^\varepsilon \oplus \mathbf{D})$ 的雅可比

在Bundle Adjustment [2]实现中需要这种点 \mathbf{p} 相对于位姿 \mathbf{D} 相对位置的差运算操作(点相对于相机是 $\mathbf{p} \ominus \mathbf{D}$, \mathbf{D} 是相机实际位姿).

令 $\mathbf{p} = [p_x, p_y, p_z]^T \in R^3$ 是一个3D点, $\mathbf{D} \in \mathbf{SE}(3)$ 是一个位姿且变换矩阵是:

$$\mathbf{T}(\mathbf{D}) = \left(\begin{array}{ccc|c} \mathbf{d}_{c1} & \mathbf{d}_{c2} & \mathbf{d}_{c3} & \mathbf{d}_t \\ 0 & 0 & 0 & 1 \end{array} \right) = \left(\begin{array}{c|c} \mathbf{R}_D & \mathbf{d}_t \\ \hline 0 & 1 \end{array} \right) \quad (10.24)$$

可以得到雅可比矩阵:

$$\begin{aligned} \left. \frac{\partial \mathbf{p} \ominus (\mathbf{e}^\varepsilon \mathbf{D})}{\partial \varepsilon} \right|_{\varepsilon=0} &= \left. \frac{\partial \mathbf{p} \ominus \mathbf{A}}{\partial \mathbf{A}} \right|_{\mathbf{A}=\mathbf{e}^\varepsilon \mathbf{D}=\mathbf{D}} \cdot \left. \frac{\partial \mathbf{e}^\varepsilon \mathbf{D}}{\partial \varepsilon} \right|_{\varepsilon=0} \\ &= (\mathbf{I}_3 \otimes ((\mathbf{p} - \mathbf{d}_t)^T) - \mathbf{R}_D^T) \left(\begin{array}{c|c} 0_{3 \times 3} & -[\mathbf{d}_{c1}]_\times \\ 0_{3 \times 3} & -[\mathbf{d}_{c2}]_\times \\ 0_{3 \times 3} & -[\mathbf{d}_{c3}]_\times \\ \hline \mathbf{I}_3 & -[\mathbf{d}_t]_\times \end{array} \right) \\ &= \begin{pmatrix} d_{21}p_z - d_{31}p_y & -d_{11}p_z + d_{31}p_x & d_{11}p_y - d_{21}p_x \\ -\mathbf{R}_D^T & d_{22}p_z - d_{32}p_y & -d_{12}p_z + d_{32}p_x & d_{12}p_y - d_{22}p_x \\ d_{23}p_z - d_{33}p_y & -d_{13}p_z + d_{33}p_x & d_{13}p_y - d_{23}p_x \end{pmatrix} (3 \times 6 \quad \text{Jacobian}) \end{aligned} \quad (10.25)$$

10.3.7 $\mathbf{A} \oplus \mathbf{e}^\varepsilon \oplus \mathbf{D}$ 的雅可比

令 $\mathbf{A}, \mathbf{D} \in \mathbf{SE}(3)$ 是两个位姿, \mathbf{D} 在上文定义过, $\mathbf{R}(\mathbf{A})$ 是 \mathbf{A} 的 3×3 矩阵。

当优化一个链式位姿序列 $(\mathbf{A} \oplus \mathbf{D})$ 时, 我们需要估计:

$$\begin{aligned} \left. \frac{\partial \mathbf{A}(\mathbf{e}^\varepsilon \mathbf{D})}{\partial \varepsilon} \right|_{\varepsilon=0} &= \left. \frac{\partial \mathbf{A} \oplus \mathbf{B}}{\partial \mathbf{B}} \right|_{\mathbf{B}=\mathbf{e}^\varepsilon \mathbf{D}=\mathbf{D}} \cdot \left. \frac{\partial \mathbf{e}^\varepsilon \mathbf{D}}{\partial \varepsilon} \right|_{\varepsilon=0} \\ &= [\mathbf{I}_4 \otimes \mathbf{R}(\mathbf{A})] \left. \frac{\partial \mathbf{e}^\varepsilon \mathbf{D}}{\partial \varepsilon} \right|_{\varepsilon=0} \\ &= \begin{pmatrix} 0_{3 \times 3} & -\mathbf{R}(\mathbf{A})[\mathbf{d}_{c1}]_\times \\ 0_{3 \times 3} & -\mathbf{R}(\mathbf{A})[\mathbf{d}_{c2}]_\times \\ 0_{3 \times 3} & -\mathbf{R}(\mathbf{A})[\mathbf{d}_{c3}]_\times \\ \hline \mathbf{R}(\mathbf{A}) & -\mathbf{R}(\mathbf{A})[\mathbf{d}_t]_\times \end{pmatrix} (3 \times 6 \quad \text{Jacobian}) \end{aligned} \quad (10.28)$$

10.3.8 $\mathbf{A} \oplus \mathbf{e}^\varepsilon \oplus \mathbf{D} \oplus \mathbf{p}$ 的雅可比

这个雅可比矩阵在计算机视觉问题中可能出现，例如bundle-adjustment [3]。令 \mathbf{p} 是一个3D点， $\mathbf{A}, \mathbf{D} \in \mathbf{SE}(3)$ 是两个位姿， $\mathbf{R}(\mathbf{A})$ 是 \mathbf{A} 的 3×3 矩阵， \mathbf{D} 有如下定义：

$$\mathbf{T}(\mathbf{D}) = \left(\begin{array}{ccc|c} \mathbf{d}_{c1} & \mathbf{d}_{c2} & \mathbf{d}_{c3} & \mathbf{d}_t \\ 0 & 0 & 0 & 1 \end{array} \right) = \left(\begin{array}{c|c} \mathbf{d}_{r1}^T & \mathbf{d}_{tx} \\ \mathbf{d}_{r2}^T & \mathbf{d}_{ty} \\ \mathbf{d}_{r3}^T & \mathbf{d}_{tz} \\ \hline \mathbf{0} & 1 \end{array} \right) \quad (10.29)$$

所以链式位姿一点和运算在位姿 \mathbf{D} 处增量的雅可比为：

$$\left. \frac{\partial \mathbf{A}(e^\varepsilon \mathbf{D}) \mathbf{p}}{\partial \varepsilon} \right|_{\varepsilon=0} = \mathbf{R}(\mathbf{A}) \left(\begin{array}{c|ccc} & 0 & \mathbf{p} \cdot \mathbf{d}_{r3} + d_{tz} & -(\mathbf{p} \cdot \mathbf{d}_{r2} + d_{ty}) \\ \mathbf{I}_3 & -(\mathbf{p} \cdot \mathbf{d}_{r3} + d_{tz}) & 0 & \mathbf{p} \cdot \mathbf{d}_{r1} + d_{tx} \\ & \mathbf{p} \cdot \mathbf{d}_{r2} + d_{ty} & -(\mathbf{p} \cdot \mathbf{d}_{r1} + d_{tx}) & 0 \end{array} \right) \quad (3 \times 6 \quad \text{Jacobian}) \quad (10.30)$$

当 \mathbf{A}, \mathbf{D} 非常接近于单位阵时，可以用以下公式近似：

$$\left. \frac{\partial \mathbf{A}(e^\varepsilon \mathbf{D}) \mathbf{p}}{\partial \varepsilon} \right|_{\varepsilon=0} \approx (\mathbf{I}_3 \quad -[\mathbf{p} + \mathbf{d}_t]_\times) \quad (3 \times 6 \quad \text{Jacobian}) \quad (10.31)$$

10.3.9 $\mathbf{p} \ominus (\mathbf{A} \oplus \mathbf{e}^\varepsilon \oplus \mathbf{D})$ 的雅可比

这个雅可比矩阵在计算机视觉问题中可能出现，例如bundle-adjustment [3]。令 \mathbf{p} 是一个3D点， $\mathbf{A}, \mathbf{D} \in \mathbf{SE}(3)$ 是两个位姿， $\mathbf{R}(\mathbf{A})$ 是 \mathbf{A} 的 3×3 矩阵， \mathbf{D} 参考以前的章节，并且：

$$\mathbf{T}(\mathbf{A})\mathbf{T}(\mathbf{D}) = \left(\begin{array}{c|c} \mathbf{R}(\mathbf{AD}) & \mathbf{t}_{AD} \\ \hline \mathbf{0} & 1 \end{array} \right) \quad (10.31)$$

所以雅可比矩阵：

$$\left. \frac{\partial (\mathbf{A} e^\varepsilon \mathbf{D})^{-1} \mathbf{p}}{\partial \varepsilon} \right|_{\varepsilon=0} = [\mathbf{I}_3 \otimes (\mathbf{p} - \mathbf{t}_{AB})^T \quad -\mathbf{R}(\mathbf{AD})^T] \left(\begin{array}{c|c} 0_{3 \times 3} & -\mathbf{R}(\mathbf{A})[\mathbf{d}_{c1}]_\times \\ 0_{3 \times 3} & -\mathbf{R}(\mathbf{A})[\mathbf{d}_{c2}]_\times \\ 0_{3 \times 3} & -\mathbf{R}(\mathbf{A})[\mathbf{d}_{c3}]_\times \\ \hline \mathbf{R}(\mathbf{AD}) & -\mathbf{R}(\mathbf{A})[\mathbf{d}_t]_\times \end{array} \right) \quad (3 \times 6 \quad \text{Jacobian}) \quad (10.32)$$

11 计算机视觉当中的应用

本章介绍章节7-10雅可比矩阵求导在计算机视觉中的应用。

11.1 理想针孔相机的映射模型— $h(p)$

给定相对于一个相机的点 $\mathbf{p} \in \mathbb{R}^3$,相机的轴采用如下图11.1的约定:

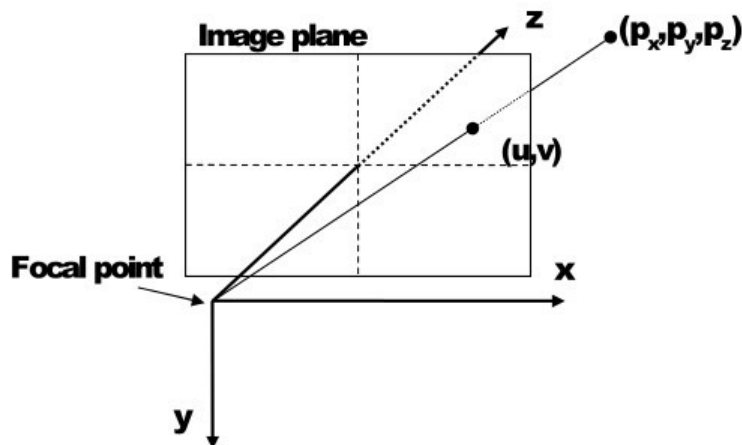


图 6: 针孔相机投影轴的约定

并且给出 3×3 的相机内参矩阵:

$$\mathbf{M} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \rightarrow \begin{cases} f_x : x \text{方向上的焦距} \\ f_y : y \text{方向上的焦距} \\ c_x : \text{图像中心点的} x \text{坐标} \\ c_y : \text{图像中心点的} y \text{坐标} \end{cases} \quad (\text{A.1})$$

不考虑扭曲的情况下, 一个3D点 $\mathbf{p} = [p_x, p_y, p_z]^T$ 的投影坐标 (u, v) 由函数 $h: \mathbb{R}^3 \rightarrow \mathbb{R}^2$ 给出:

$$h(\mathbf{p}) = h \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} = \begin{pmatrix} c_x + f_x \frac{p_x}{p_z} \\ c_y + f_y \frac{p_y}{p_z} \end{pmatrix} \quad (\text{A.2})$$

在计算机视觉中的诸多问题都需要这个投影函数的雅可比矩阵:

$$\frac{\partial \mathbf{h}(\mathbf{p})}{\partial \mathbf{p}} = \begin{pmatrix} f_x/p_z & 0 & -f_x p_x/p_z^2 \\ 0 & f_y/p_z & -f_y p_y/p_z^2 \end{pmatrix} \quad (\text{A.3})$$

11.2 $\mathbf{e}^\varepsilon \oplus \mathbf{A} \oplus \mathbf{p}$ 点的投影

给定一个位姿 $\mathbf{A} \in \mathbf{SE}(3)$ (它的旋转矩阵为 $\mathbf{R}_\mathbf{A}$),该位姿下的一个点 $\mathbf{p} \in \mathbb{R}^3$,我们想要推导出 $\mathbf{e}^\varepsilon \oplus \mathbf{A} \oplus \mathbf{p}$ 在针孔相机上的投影 $h(\mathbf{e}^\varepsilon \oplus \mathbf{A} \oplus \mathbf{p})$ 的雅可比矩阵. \mathbf{e}^ε 是李群辅助参数 ε 的指数,表示流形上 \mathbf{A} 处的小增量。

令 $\mathbf{g} = [\mathbf{g}_x, \mathbf{g}_y, \mathbf{g}_z]^\mathbf{T}$ 表示 $\mathbf{A} \oplus \mathbf{p}$.应用链式法则和公式7.13、10.23、A.3我们可以得到:

$$\begin{aligned} \frac{\partial \mathbf{h}(\mathbf{e}^\varepsilon \oplus \mathbf{A} \oplus \mathbf{p})}{\partial \mathbf{p}} &= \left. \frac{\partial \mathbf{h}(\mathbf{p}')}{\partial \mathbf{p}'} \right|_{\mathbf{p}'=\mathbf{A} \oplus \mathbf{p}=\mathbf{g}} \cdot \frac{\partial \mathbf{e}^\varepsilon \oplus \mathbf{A} \oplus \mathbf{p}}{\partial \mathbf{p}} \\ &= \left. \frac{\partial \mathbf{h}(\mathbf{p}')}{\partial \mathbf{p}'} \right|_{\mathbf{p}'=\mathbf{A} \oplus \mathbf{p}=\mathbf{g}} \cdot \frac{\partial \mathbf{A} \oplus \mathbf{p}}{\partial \mathbf{p}} \\ &= \begin{pmatrix} f_x/g_z & 0 & -f_x g_x/g_z^2 \\ 0 & f_y/g_z & -f_y g_y/g_z^2 \end{pmatrix} \mathbf{R}_\mathbf{A} \quad (3 \times 3 \quad \text{Jacobian}) \end{aligned} \quad (\text{A.6})$$

$$\begin{aligned} \frac{\partial \mathbf{h}(\mathbf{e}^\varepsilon \oplus \mathbf{A} \oplus \mathbf{p})}{\partial \varepsilon} &= \left. \frac{\partial \mathbf{h}(\mathbf{p}')}{\partial \mathbf{p}'} \right|_{\mathbf{p}'=\mathbf{A} \oplus \mathbf{p}=\mathbf{g}} \cdot \frac{\partial \mathbf{e}^\varepsilon \oplus \mathbf{A} \oplus \mathbf{p}}{\partial \varepsilon} \\ &= \begin{pmatrix} f_x/g_z & 0 & -f_x g_x/g_z^2 \\ 0 & f_y/g_z & -f_y g_y/g_z^2 \end{pmatrix} (\mathbf{I}_3 - [\mathbf{g}]_\times) \\ &= \begin{pmatrix} \frac{f_x}{g_z} & 0 & -f_x \frac{g_x}{g_z^2} & -f_x \frac{g_x g_y}{g_z^2} & f_x (1 + \frac{g_x^2}{g_z^2}) & -f_x \frac{g_y}{g_z} \\ 0 & \frac{f_y}{g_z} & -f_y \frac{g_y}{g_z^2} & -f_y (1 + \frac{g_y^2}{g_z^2}) & f_y \frac{g_x g_y}{g_z^2} & f_y \frac{g_x}{g_z} \end{pmatrix} \quad (3 \times 6 \quad \text{Jacobian}) \end{aligned} \quad (\text{A.8})$$

11.3 $\mathbf{p} \ominus (\mathbf{e}^\varepsilon \oplus \mathbf{A})$ 点的投影

以前的章节雅可比矩阵被应用于估计相机位姿的逆,本章节介绍的雅可比用于实际相机位姿。

为了获取函数 $\mathbf{h}(\mathbf{p} \ominus (\mathbf{e}^\varepsilon \oplus \mathbf{A}))$ 的雅可比矩阵,使用公式7.16、A.3和 $\mathbf{l} = [l_x, l_y, l_z]^\mathbf{T} = \mathbf{p} \ominus \mathbf{A}$:

$$\begin{aligned} \frac{\partial \mathbf{h}(\mathbf{p} \ominus (\mathbf{e}^\varepsilon \oplus \mathbf{A}))}{\partial \mathbf{p}} &= \left. \frac{\partial \mathbf{h}(\mathbf{p}')}{\partial \mathbf{p}'} \right|_{\mathbf{p}'=\mathbf{A} \oplus \mathbf{p}=\mathbf{l}} \cdot \frac{\partial \mathbf{p} \ominus (\mathbf{e}^\varepsilon \oplus \mathbf{A})}{\partial \mathbf{p}} \\ &= \left. \frac{\partial \mathbf{h}(\mathbf{p}')}{\partial \mathbf{p}'} \right|_{\mathbf{p}'=\mathbf{A} \oplus \mathbf{p}=\mathbf{l}} \cdot \frac{\partial \mathbf{p} \ominus \mathbf{A}}{\partial \mathbf{p}} \\ &= \begin{pmatrix} f_x/l_z & 0 & -f_x l_x/l_z^2 \\ 0 & f_y/l_z & -f_y l_y/l_z^2 \end{pmatrix} \mathbf{R}_\mathbf{A}^\mathbf{T} \quad (3 \times 3 \quad \text{Jacobian}) \end{aligned} \quad (\text{A.11})$$

$$\begin{aligned} \frac{\partial \mathbf{h}(\mathbf{p} \ominus (\mathbf{e}^\varepsilon \oplus \mathbf{A}))}{\partial \varepsilon} &= \left. \frac{\partial \mathbf{h}(\mathbf{p}')}{\partial \mathbf{p}'} \right|_{\mathbf{p}'=\mathbf{p} \ominus \mathbf{A}} \cdot \frac{\partial \mathbf{p} \ominus (\mathbf{e}^\varepsilon \oplus \mathbf{A})}{\partial \varepsilon} \\ &= \begin{pmatrix} f_x/l_z & 0 & -f_x l_x/l_z^2 \\ 0 & f_y/l_z & -f_y l_y/l_z^2 \end{pmatrix} \frac{\partial \mathbf{p} \ominus (\mathbf{e}^\varepsilon \oplus \mathbf{A})}{\partial \varepsilon} \end{aligned} \quad (\text{A.13})$$

最后一项见公式10.25.

References

- [1] Juan-Antonio Fernández Madrigal and José Luis Blanco. Simultaneous Localization and Mapping for Mobile Robots: Introduction and Methods. IGI Global, 2012.
- [2] B.Triggs. Bundle adjustment-a modern synthesis. Vision algorithms:theory and practice, 153(177), 2000.
- [3] G.Sibley. Relative bundle adjustment. Technical report, Department of Engineering Science, Oxford University, 2009.