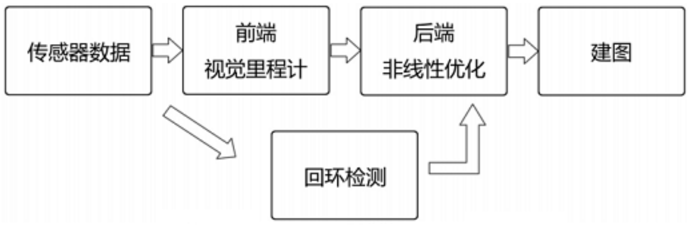
《视觉SLAM十四讲》笔记摘抄

原创 ncepu_Chen

《视觉 SLAM 十四讲》笔记摘抄

ch02 初识SLAM

经典视觉SLAM框架



整体视觉SLAM流程图

视觉SLAM流程包括以下步骤:

1. **传感器信息读取**: 在视觉SLAM中主要为相机图像信息的读取和预处理.如果是在机器人中,还可能有码盘、惯性传感器等信息的读取和同步.
2. **视觉里程计**(Visual Odometry,VO): 视觉里程计的任务是估算相邻图像间相机的运动,以及局部地图的样子.VO又称为前端(Front End).

   视觉里程计不可避免地会出现**累积漂移**(Accumulating Drift)问题.
3. **后端优化** (Optimization): 后端接受不同时刻视觉里程计测量的相机位姿,以及回环检测的信息,对它们进行优化,得到全局一致的轨迹和地图.由于接在VO之后,又称为后端(Back End).

   在视觉 SLAM中,前端和计算机视觉 研究领域更为相关,比如图像的特征提取与匹配等,后端则主要是滤波与非线性优化算法.
4. **回环检测** (Loop Closing): 回环检测判断机器人是否到达过先前的位置.如果检测到回环,它会把信息提供给后端进行处理.
5. **建图** (Mapping): 它根据估计的轨迹,建立与任务要求对应的地图.

   地图的形式包括**度量地图**(精确表示地图物体的位置关系)与**拓扑地图**(更强调地图元素之间的关系)两种.

SLAM问题的数学表述

“小萝卜携带着传感器在环境中运动”,由如下两件事情描述:

1. 什么是**运动** ?我们要考虑从$k-1$时刻到k时刻,小萝卜的位置x是如何变化的.

   运动方程:

   $$x_k = f(x_{k-1}, u_k, w_k)$$

   - x_k , x_{k-1} 表示小萝卜在k和$k-1$时刻的位置
   - u_k 表示运动传感器的读数(有时也叫**输入**)
   - w_k 表示噪声
2. 什么是**观测** ?假设小萝卜在k时刻于x_k 处探测到了某一个路标y_j ,我们要考虑这件事情是如何用数学语言来描述的.

观测方程:

$$z_{k,j} = h(y_j, x_k, v_{k,j})$$

- $z_{k,j}$ 表示小萝卜在x_k位置上看到路标点y_j,产生的观测数据
- y_j 表示第j个路标点
- $v_{k,j}$ 表示噪声

这两个方程描述了最基本的SLAM问题:当知道运动测量的读数u,以及传感器的读数z时,如何求解定位问题(估计x)和建图问题(估计y)?这时,我们就把SLAM问题建模成了一个**状态估计问题**:如何通过带有噪声的测量数据,估计内部的、隐藏着的状态变量?

ch03 三维空间刚体运动

旋转矩阵

点和向量,坐标系

1. 向量a在线性空间的基$[e_1, e_2, e_3]$下的坐标为$[a_1, a_2, a_3]^T$.

$$a = [e_1, e_2, e_3]\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = a_1 e_1 + a_2 e_2 + a_3 e_3$$

2. 向量的内积与外积

   - 向量的内积: 描述向量间的投影关系

$$a \cdot b = a^T b = \sum_{i=1}^{3} a_i b_i = |a|\,|b| \cos\langle a, b\rangle$$

   - 向量的外积: 描述向量的旋转

$$a \times b = \begin{bmatrix} i & j & k \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{bmatrix} = \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} b \triangleq a^\wedge b$$

   其中$a^\wedge$表示a的反对称矩阵

$$a^\wedge = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

坐标系间的欧氏变换

1. 欧式变换:

   在欧式变换前后的两个坐标系下,同一个向量的模长和方向不发生改变,是为欧式变换.

   一个欧式变换由一个旋转和一个平移组成.

2. 旋转矩阵R:

   - 旋转矩阵R的推导:

     设单位正交基$[e_1, e_2, e_3]$经过一次旋转变成了$[e_1', e_2', e_3']$,对于同一个向量a,在两个坐标系下的坐标分别为$[a_1, a_2, a_3]^T$和$[a_1', a_2', a_3']^T$.根据坐标的定义:

$$[e_1, e_2, e_3]\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = [e'_1, e'_2, e'_3]\begin{bmatrix} a'_1 \\ a'_2 \\ a'_3 \end{bmatrix}$$

等式左右两边同时左乘$[e_1^T, e_2^T, e_3^T]^T$,得到

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} e_1^T e'_1 & e_1^T e'_2 & e_1^T e'_3 \\ e_2^T e'_1 & e_2^T e'_2 & e_2^T e'_3 \\ e_3^T e'_1 & e_3^T e'_2 & e_3^T e'_3 \end{bmatrix}\begin{bmatrix} a'_1 \\ a'_2 \\ a'_3 \end{bmatrix} \triangleq Ra'$$

矩阵R描述了旋转,称为**旋转矩阵**.

- 旋转矩阵R的性质

  1. 旋转矩阵是**行列式为1的正交矩阵**,任何行列式为1的正交矩阵也是一个旋转矩阵.所有旋转矩阵构成特殊正交群SO:

  $$SO(n) = \{R \in \mathbb{R}^{n\times n} | RR^T = I, \det(R) = 1\}$$

  2. 旋转矩阵是正交矩阵(其转置等于其逆),旋转矩阵的逆R^{-1}(即转置R^T)描述了一个相反的旋转.

3. 欧式变换的向量表示:

   世界坐标系中的向量a,经过一次旋转(用旋转矩阵R描述)和一次平移(用平移向量t描述)后,得到了a':

$$a' = Ra + t$$

变换矩阵与齐次坐标

1. 变换矩阵T:

   在三维向量的末尾添加1,构成的四维向量称为**齐次坐标**.将旋转和平移写入**变换矩阵**T中,得到:

$$\begin{bmatrix} a' \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}\begin{bmatrix} a \\ 1 \end{bmatrix} \triangleq T\begin{bmatrix} a \\ 1 \end{bmatrix}$$

   齐次坐标的意义在于**将欧式变换表示为线性关系**.

2. 变换矩阵T的性质:

   1. 变换矩阵T构成特殊欧式群SE

   $$SE(3) = \left\{T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} | R \in SO(3), t \in \mathbb{R}^3\right\}$$

   2. 变换矩阵的逆表示一个反向的欧式变换

   $$T^{-1} = \begin{bmatrix} R^T & -R^T t \\ 0 & 1 \end{bmatrix}$$

齐次坐标(Homogeneous Coordinate)的优势

优势1:方便判断是否在直线或平面上

若点$p = (x, y)$在直线$l = (a, b, c)$上,则有:

$$ax + by + c = [a, b, c]^T \cdot [x, y, 1] = l^T \cdot p' = 0$$

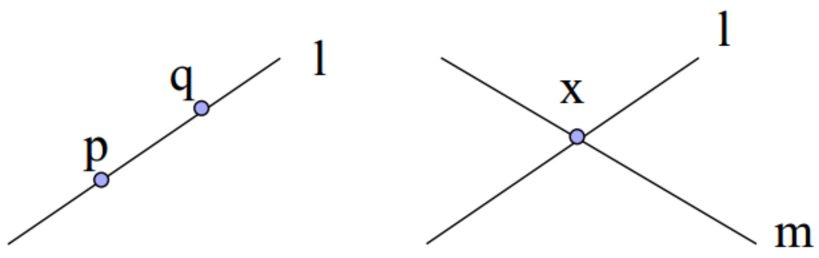
若点$p = (x, y, z)$在平面$A = (a, b, c, d)$上,则有:

$$ax + by + cz + d = [a, b, c, d]^T \cdot [x, y, z, 1] = A^T \cdot p' = 0$$

优势2:方便表示线线交点和点点共线

在齐次坐标下,

1. 可以用两个点p,q的齐次坐标叉乘结果表示它们的共线l.
2. 可以用两条直线l,m的齐次坐标叉乘结果表示它们的交点x.



这里利用叉乘的性质: 叉乘结果与两个运算向量都垂直:

- 性质1的证明:

$$
\begin{gathered}
\mathrm{l}^{\mathrm{T}} \cdot \mathrm{p} = (\mathrm{p} \times \mathrm{q}) \cdot \mathrm{p} = 0 \\
\mathrm{l}^{\mathrm{T}} \cdot \mathrm{q} = (\mathrm{p} \times \mathrm{q}) \cdot \mathrm{q} = 0
\end{gathered}
$$

- 性质2的证明:

$$
\begin{gathered}
\mathrm{l}^{\mathrm{T}} \cdot \mathrm{p} = \mathrm{l}^{\mathrm{T}} \cdot (\mathrm{l} \times \mathrm{m}) = 0 \\
\mathrm{m}^{\mathrm{T}} \cdot \mathrm{p} = \mathrm{m}^{\mathrm{T}} \cdot (\mathrm{l} \times \mathrm{m}) = 0
\end{gathered}
$$

优势3:能够区分向量和点

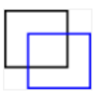
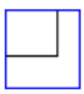
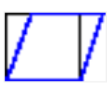
- 点(x, y, z)的齐次坐标为$(x, y, z, 1)$
- 向量(x, y, z)的齐次坐标为$(x, y, z, 0)$

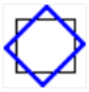
优势4:能够表达无穷远点

对于平行直线$l = (a, b, c)$和$m = (a, b, d)$,求取其交点的齐次坐标$x = l \times m = (kb, -ka, 0)$,将其转为非齐次坐标,得到$x = (kb/0, -ka/0) = (\inf, -\inf)$,这表示无穷远点.

优势5:能够简洁的表示变换

使用齐次坐标,可以将加法运算转化为乘法运算.

变换形式	图形示意	数学变换	MATLAB函数
位移(Translation)		$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$	`imtranslate()`
缩放(Scale)		$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$	`imresize()`
错切(Shear)		$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & h_x & 0 \\ h_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$	

变换形式	图形示意	数学变换	MATLAB函数
旋转(Rotate)		$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$	`imrotate()`

旋转向量和欧拉角

旋转向量

- 旋转矩阵的缺点:
  1. 旋转矩阵有9个量,但一次旋转只有3个自由度,这种表达方式是冗余的.
  2. 旋转矩阵自带约束(必须是行列式为1的正交矩阵),这些约束会给估计和优化带来困难.
- 旋转向量: 任意旋转都可以用一个**旋转轴**和一个**旋转角** 来刻画.于是,我们可以使用一个向量,其**方向表示旋转轴**而**长度表示旋转角**.这种向量称为**旋转向量**(或**轴角**,Axis-Angle).

  假设有一个旋转轴为n,角度为θ的旋转,其对应的旋转向量为θn.
- 旋转向量和旋转矩阵之间的转换:

  设旋转向量R表示一个绕单位向量n,角度为θ的旋转.
  - 旋转向量到旋转矩阵:

  $$R = \cos\theta I + (1-\cos\theta) n n^T + \sin\theta\, n^\wedge$$
  - 旋转矩阵到旋转向量:
    - 旋转角$\theta = \arccos\left(\frac{tr(R)-1}{2}\right)$
    - 旋转轴n是矩阵R特征值1对应的特征向量

欧拉角

- 欧拉角将一次旋转分解成**3个分离的转角**.常用的一种ZYX转角将任意旋转分解成以下3个轴上的转角:
  1. 绕物体的Z轴旋转,得到偏航角yaw
  2. 绕**旋转之后**的Y 轴旋转,得到俯仰角pitch
  3. 绕**旋转之后**的X轴旋转,得到滚转角roll
- 欧拉角的一个重大缺点是**万向锁问题**(**奇异性问题**): 在俯仰角为$\pm$90° 时,第一次旋转与第三次旋转将使用同一个轴,使得系统丢失了一个自由度(由3次旋转变成了2次旋转).

四元数

为什么需要四元数: 对于三维旋转,找不到**不带奇异性的三维向量描述方式**.因此引入四元数.
四元数是一种**扩展的复数**,**既是紧凑的**,**也没有奇异性**.

四元数的定义

1. 四元数的定义

   一个四元数q拥有一个实部和三个虚部

   $$q = q_0 + q_1 i + q_2 j + q_3 k$$

   其中i,j,k,为四元数的3个虚部,它们满足以下关系式(**自己和自己的运算像复数,自己和别人的运算像叉乘**):

$$\begin{cases} i^2 = j^2 = k^2 = -1 \\ ij = k, ji = -k \\ jk = i, kj = -i \\ ki = j, ik = -j \end{cases}$$

也可以用一个标量和一个向量来表达四元数:

$$q = [s, v], \quad s = q_0 \in \mathbb{R} \quad v = [q_1, q_2, q_3]^T \in \mathbb{R}^3$$

s为四元数的实部,v为四元数的虚部.有**实四元数**和**虚四元数**的概念.

2. 四元数与旋转角度的关系:
   - 在二维情况下,任意一个旋转都可以用**单位**复数来描述,乘i就是绕i轴旋转90°.
   - 在三维情况下,任意一个旋转都可以用**单位**四元数来描述,乘i就是绕i轴旋转180°.
3. 单位四元数和旋转向量之间的转换:

   设单位四元数q表示一个绕单位向量$n = [n_x, n_y, n_z]^T$,角度为θ的旋转.
   - 从旋转向量到单位四元数:

$$q = \left[\cos(\frac{\theta}{2}), n \sin(\frac{\theta}{2})\right]^T = \left[\cos(\frac{\theta}{2}), n_x \sin(\frac{\theta}{2}), n_y \sin(\frac{\theta}{2}), n_z \sin(\frac{\theta}{2})\right]^T$$

   - 从单位四元数到旋转向量:

$$\begin{cases} \theta = 2 \arccos q_0 \\ [n_x, n_y, n_z] = [q_1, q_2, q_3]^T / \sin\dfrac{\theta}{2} \end{cases}$$

用单位四元数表示旋转

给定一个空间三维点$p = [x, y, z] \in \mathbb{R}^3$,以及一个由轴角n,$\theta$指定的旋转,三维点p经过旋转后变为p′.如何使用单位四元数q表达旋转?

1. 把三维空间点用一个虚四元数p表示:

$$p = [0, x, y, z] = [0, v]$$

2. 把旋转用单位四元数q表示:

$$q = [\cos\frac{\theta}{2}, n \sin\frac{\theta}{2}]$$

3. 旋转后的点p'可表示为:

$$p' = qpq^{-1}$$

这样得到的点p'仍为一个纯虚四元数,其虚部的3个分量表示旋转后3D点的坐标.

> 只有**单位**四元数才能表示旋转,因此在程序中创建四元数后,要记得调用 `normalize()` 以将其单位化

ch04 李群与李代数

李群与李代数基础

旋转矩阵构成特殊正交群$\mathrm{SO}(3)$,变换矩阵构成了特殊欧氏群$\mathrm{SE}(3)$.

$$
\begin{aligned}
\mathrm{SO}(3) &= \left\{\mathrm{R} \in \mathbb{R}^{3\times 3} | \mathrm{R}\mathrm{R}^{\mathrm{T}} = \mathrm{I}, \det(\mathrm{R}) = 1\right\} \\
\mathrm{SE}(3) &= \left\{\mathrm{T} = \begin{bmatrix} \mathrm{R} & \mathrm{t} \\ 0^{\mathrm{T}} & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} | \mathrm{R} \in \mathrm{SO}(3), \mathrm{t} \in \mathbb{R}^3\right\}
\end{aligned}
$$

群的定义

- 群(Group)是**一种集合**加上**一种运算**的代数结构.把集合记作A,运算记作$\cdot$,那么群可以记作$G = (A, \cdot)$.群要求这个运算满足如下条件(**封结幺逆**):

  1. 封闭性: $\forall a_1, a_2 \in A, \quad a_1 \cdot a_2 \in A.$
  2. 结合律: $\forall a_1, a_2, a_3 \in A, \quad (a_1 \cdot a_2) \cdot a_3 = a_1 \cdot (a_2 \cdot a_3)$
  3. 幺元: $\exists a_0 \in A, \quad s.t. \quad \forall a \in A, \quad a_0 \cdot a = a \cdot a_0 = a$
  4. 逆: $\forall a \in A, \quad \exists a^{-1} \in A, \quad s.t. a \cdot a^{-1} = a_0$
- **李群**是指具有连续(光滑)性质的群.$\mathrm{SO}(3)$和$\mathrm{SE}(3)$都是李群

李代数的定义

每个李群都有与之对应的李代数,李代数描述了李群的局部性质.

通用的李代数的定义如下:
李代数由一个集合V ,一个数域F 和一个二元运算$[,]$组成.如果它们满足以下几条性质,则称$(V, F, [,])$为一个李代数,记作$\mathfrak{g}$.

1. 封闭性: $\forall X, Y \in V, [X, Y] \in V$.
2. 双线性: $\forall X,Y,Z \in V, a,b \in F $有:

$$
[aX + bY, Z] = a[X, Z] + b[Y, Z], \quad [Z, aX + bY] = a[Z, X] + b[Z, Y]
$$

3. 自反性: $\forall X, \in V, [X, X] = 0.$
4. 雅可比等价$\forall X, Y, Z \in V, \quad [X, [Y, Z]] + [Z, [X, Y]] + [Y, [Z, X]] = 0.$

其中的二元运算$[,]$被称为**李括号**.例如三维向量空间$\mathbb{R}^3$上定义的叉积$\times$是一种李括号.

李代数$\mathfrak{so}(3)$

- 李群$\mathrm{SO}(3)$对应的李代数$\mathfrak{so}(3)$是定义在$\mathbb{R}^3$上的向量,记作ϕ.

$$
\mathfrak{so}(3) = \left\{\phi \in \mathbb{R}^3, \Phi = \phi^\wedge = \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix} \in \mathbb{R}^{3\times 3}\right\}
$$

- 李代数$\mathfrak{so}(3)$的李括号为

$$
[\phi_1, \phi_2] = (\Phi_1 \Phi_2 - \Phi_2 \Phi_1)^\vee
$$

  其中$^\vee$是$^\wedge$的逆运算,表示将反对称矩阵还原为向量

- $\mathfrak{so}(3)$和$\mathrm{SO}(3)$间的映射关系为

$$
\begin{aligned}
\text{李群}R &= \exp(\phi^\wedge) = \exp(\Phi) \\
\text{李代数}\phi &= \ln(R)^\vee
\end{aligned}
$$

李代数$\mathfrak{se}(3)$

- 类似地,李群$\mathrm{SE}(3)$的李代数$\mathfrak{se}(3)$是定义在$\mathbb{R}^6$上上的向量.记作ξ:

$$
\mathfrak{se}(3) = \left\{\xi = \begin{bmatrix} \rho \\ \phi \end{bmatrix} \in \mathbb{R}^6, \rho \in \mathbb{R}^3, \phi \in \mathfrak{so}(3), \xi^\wedge = \begin{bmatrix} \phi^\wedge & \rho \\ 0^{\mathrm{T}} & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4}\right\}
$$

$\mathfrak{se}(3)$中的每个元素ξ,是一个六维向量.前三维ρ表示平移;后三维ϕ表示旋转,本质上是$\mathfrak{so}(3)$元素.

- 在这里同样使用$^\wedge$符号将六维向量扩展成为四维矩阵,但不再表示反对称

$$\xi^\wedge = \begin{bmatrix} \phi^\wedge & \rho \\ 0^T & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4}$$

- 李代数$\mathfrak{se}(3)$的李括号和$\mathfrak{so}(3)$类似:

$$[\xi_1, \xi_2] = (\xi_1^\wedge \xi_2^\wedge - \xi_2^\wedge \xi_1^\wedge)^\vee$$

- $\mathfrak{se}(3)$和$\mathrm{SE}(3)$间映射关系为

$$\begin{aligned} \text{李群}T &= \exp(\xi^\wedge) \\ \text{李代数}\xi &= \ln(T)^\vee \end{aligned}$$

李群与李代数的转换关系:指数映射和对数映射

$\mathrm{SO}(3)$**和**$\mathfrak{so}(3)$**间的转换关系**

- 将三维向量ϕ分解为其模长θ和方向向量α,即$\phi = \theta\alpha$.则从$\mathfrak{so}(3)$到$\mathrm{SO}(3)$的**指数映射**可表示为:

$$R = \exp(\phi) = \exp(\theta\alpha^\wedge) = \cos\theta I + (1-\cos\theta)\alpha\alpha^T + \sin\theta\alpha^\wedge$$

上式即为旋转向量到旋转矩阵的罗德里格斯公式,可见**$\mathfrak{so}(3)$本质上是旋转向量组成的空间**.

- 从$\mathrm{SO}(3)$到$\mathfrak{so}(3)$的**对数映射**可表示为:

$$\phi = \ln(R)^\vee$$

实际计算时可以通过迹的性质分别求出转角θ和转轴α

$$\theta = \arccos\frac{\mathrm{tr}(R)-1}{2}, \qquad R\alpha = \alpha$$

$\mathrm{SE}(3)$**和**$\mathfrak{se}(3)$**间的转换关系**

- 从$\mathfrak{se}(3)$到$\mathrm{SE}(3)$的**指数映射**可表示为:

$$T = \exp(\xi^\wedge) = \begin{bmatrix} R & J\rho \\ 0^T & 1 \end{bmatrix}$$

其中

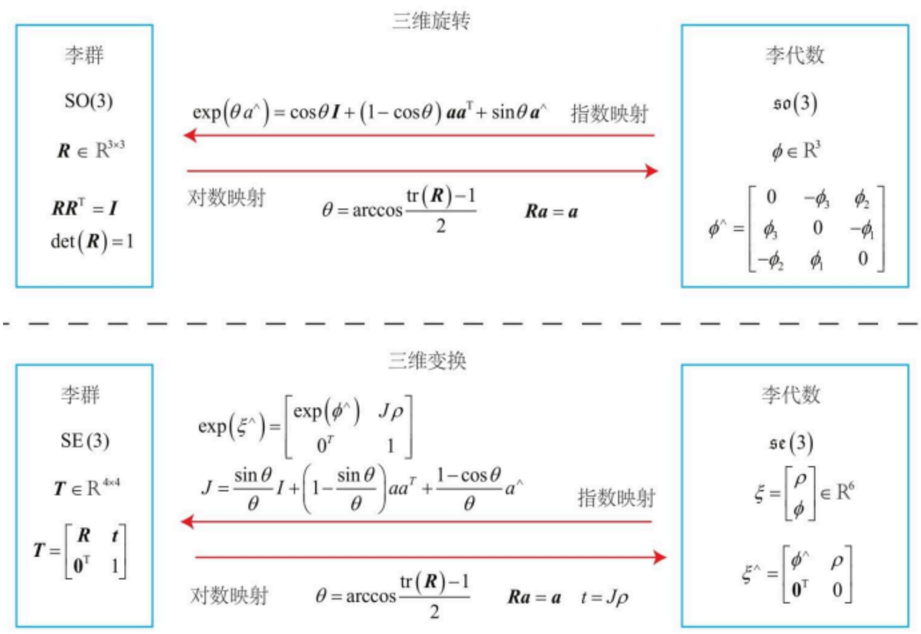
$$J = \frac{\sin\theta}{\theta} I + (1 - \frac{\sin\theta}{\theta})\alpha\alpha^T + \frac{1-\cos\theta}{\theta}\alpha^\wedge$$

可以看到,平移部分经过指数映射之后,发生了一次以J为系数矩阵的线性变换.

- 从$\mathrm{SE}(3)$到$\mathfrak{se}(3)$的**对数映射**可表示为:

$$\xi = \ln(T)^\vee$$

实际计算时ϕ可以由$\mathrm{SO}(3)$到$\mathfrak{so}(3)$的映射得到,ρ可以由$t = J\rho$计算得到.



李代数求导: 引入李代数的一大动机就是方便求导优化

李群乘法与李代数加法的关系

1. BCH公式及其近似形式

   - 很遗憾地,李群乘积和李代数加法并不等价,即:

     $$R_1 R_2 = \exp(\phi_1^\wedge) \exp(\phi_1^\wedge) \neq \exp((\phi_1 + \phi_2)^\wedge)$$

     李群乘积与李代数运算的对应关系由BCH公式给出:

     $$\ln(\exp(A)\exp(B)) = A + B + \frac{1}{2}[A, B] + \frac{1}{12}[A, [A, B]] - \frac{1}{12}[B, [A, B]] + \ldots$$

     上式中$[,]$表示李括号运算.

   - 当ϕ_1或ϕ_2为小量时,可以对BCH公式进行线性近似,得到**李群乘积对应的李代数**的表达式:

     $$R_1 \cdot R_2 \text{对应的李代数} = \ln(\exp(\phi_1^\wedge)\exp(\phi_1^\wedge))^\vee \approx \begin{cases} J_l(\phi_2)^{-1}\phi_1 + \phi_2 & \text{当}\phi_1\text{为小量时} \\ J_r(\phi_1)^{-1}\phi_2 + \phi_1 & \text{当}\phi_2\text{为小量时} \end{cases}$$

     其中左乘雅可比矩阵J_l即为从$SE(3)$到$\mathfrak{se}(3)$对数映射中的雅可比矩阵

     $$J_l = \frac{\sin\theta}{\theta} I + (1 - \frac{\sin\theta}{\theta}) \alpha\alpha^T + \frac{1-\cos\theta}{\theta}\alpha^\wedge$$

     其逆为

     $$J_l^{-1} = \frac{\theta}{2}\cot\frac{\theta}{2} I + (1 - \frac{\theta}{2}\cot\frac{\theta}{2})\alpha\alpha^T + \frac{\theta}{2}\alpha^\wedge$$

     右乘雅可比矩阵只需对自变量取负号即可

     $$J_r(\phi) = J_l(-\phi)$$

2. 李群$SO(3)$乘法与李代数$\mathfrak{so}(3)$加法的关系:

   - 对旋转R(李代数为ϕ)左乘一个微小旋转ΔR(李代数为$\Delta\phi$),得到的旋转李群$\Delta R \cdot R$对应的李代数为:

     $$\Delta R \cdot R\text{对应的李代数} = \ln(\exp(\Delta\phi^\wedge)\exp(\phi^\wedge)) = \phi + J_l^{-1}(\phi)\Delta\phi$$

- 反之,李代数加法$(\phi + \Delta\phi)$对应的李群元素可表示为:

$$(\phi + \Delta\phi)\text{对应的李群} = \exp((\phi + \Delta\phi)^\wedge) = \exp((J_l \Delta\phi)^\wedge)\exp(\phi^\wedge) = \exp(\phi^\wedge)\exp((J_r \Delta\phi)^\wedge)$$

3. 同理,李群$SE(3)$乘法与李代数$\mathfrak{se}(3)$加法的关系:

$$\begin{aligned}\exp(\Delta\xi^\wedge)\exp(\xi^\wedge) \approx \exp\left((\mathcal{J}_l^{-1}\Delta\xi + \xi)^\wedge\right) \\ \exp(\xi^\wedge)\exp(\Delta\xi^\wedge) \approx \exp\left((\mathcal{J}_r^{-1}\Delta\xi + \xi)^\wedge\right)\end{aligned}$$

$SO(3)$上的李代数求导

对空间点p进行旋转,得到Rp,旋转之后点的坐标对旋转的导数可表示为:

$$\frac{\partial(Rp)}{\partial R}$$

对于上式的求导,有两种方式:

1. 用李代数ϕ表示**姿态**R,然后根据李代数加法对ϕ求导.
2. 用李代数φ表示**微小扰动**∂R,然后根据李群左乘对φ求导.

其中扰动模型表达式简单,更为实用.

李代数求导

用李代数ϕ表示**姿态**R,求导得到

$$\frac{\partial(Rp)}{\partial R} = \frac{\partial(\exp(\phi^\wedge)p)}{\partial\phi} = -(Rp)^\wedge J_l$$

扰动模型(左乘)

另一种求导方式是对R进行一次左乘扰动∂R,设左乘扰动∂R对应的李代数为φ,对φ求导,得到

$$\frac{\partial(Rp)}{\partial R} = \frac{\exp((\phi + \varphi)^\wedge)p - \exp(\phi^\wedge)p}{\varphi} = -(Rp)^\wedge$$

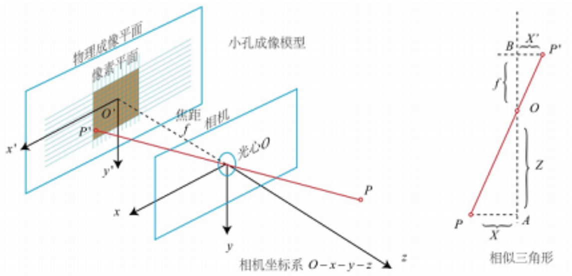
$SE(3)$上的李代数求导

类似地,空间点p经过变换T得到Tp,给T左乘一个扰动$\Delta T = \exp(\delta\xi^\wedge)$,则有

$$\frac{\partial(Rp)}{\delta\xi} = \begin{bmatrix} I & -(Rp + t)^\wedge \\ 0^T & 0^T \end{bmatrix} = (Tp)^\odot$$

ch05 相机与图像

针孔相机模型



$O - x - y - z$为相机坐标系,现实空间点P的**相机坐标**为$[X, Y, Z]^T$,投影到$O' - x' - y'$平面上的点P',坐标为$[X', Y', Z']^T$.

- 将成像平面对称到相机前方,根据几何相似关系$\frac{Z}{f} = \frac{X}{X'} = \frac{Y}{Y'}$,整理得到投影点$P'$在投影平面上的坐标$P' = [X', Y']$:

$$\begin{cases} X' = f\dfrac{X}{Z} \\ Y' = f\dfrac{Y}{Z} \end{cases}$$

- 转换得到投影点P'在像素平面上的**像素坐标**$P_{u,v} = [u, v]^T$

$$\begin{cases} u = \alpha X' + c_x = f_x\dfrac{X}{Z} + c_x \\ v = \beta Y' + c_y = f_x\dfrac{X}{Z} + c_x \end{cases}$$

  上式中u,v,c_x,c_y,f_x,f_y的单位为像素,α,β的单位为像素/米.

- 将上式写成矩阵形式,得到**现实空间点相机坐标P **和**投影点像素坐标P_{uv}**之间的关系:

$$ZP_{uv} = Z\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \triangleq KP$$

  其中矩阵K称为相机的**内参数矩阵**.

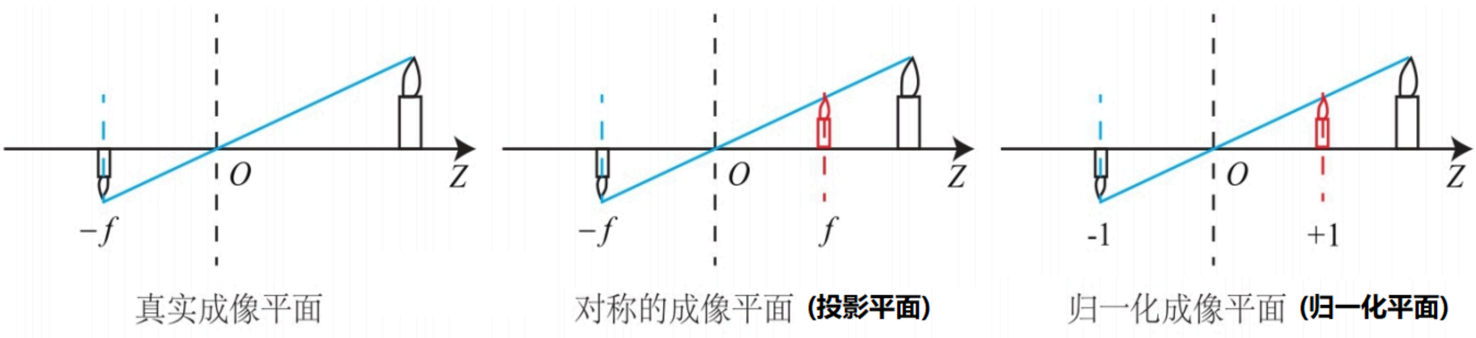
- 上式中的P为现实空间点在相机坐标系下的**相机坐标**,将其转为**世界坐标**P_W,有

$$ZP_{uv} = K(RP_W + t) = KTP_W$$

  因此R,t(或T)又称为相机的**外参数**.

- 将最后一维进行**归一化处理**,得到点P在归一化平面的**归一化坐标**$P_c = [X/Z, Y/Z, 1]^T$

$$P_c = \frac{P}{Z} = K^{-1}P_{uv}$$



参数矩阵有内参数K和外参数R,t,其中:

1. 内参数矩阵K体现了**归一化相机坐标**到**像素坐标**的变换.

   之所以是**归一化**坐标,这体现了投影性质:在某一条直线上的**空间点**,最终会投影到同一**像素点**上.

2. 外参数矩阵R,t(或T)体现了**世界坐标**到**相机坐标**的变换.

畸变模型

畸变包含两种: **径向畸变**和**切向畸变**.

- **径向畸变**: 由透镜形状引起,主要包括**桶形畸变**和**枕形畸变**.

  可以看成坐标点沿着长度方向发生了变化,也就是其距离原点的长度发生了变化.

$$x_{distorted} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r6)$$
$$y_{distorted} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r6)$$

- **切向畸变**: 由透镜和成像平面不严格平行引起.

  可以看成坐标点沿着切线方向发生了变化，也就是水平夹角发生了变化.

$$\begin{aligned} x_{\text{distorted}} &= x + 2p_1 xy + p_2(r^2 + 2x^2) \\ y_{\text{distorted}} &= y + p_1(r^2 + 2y^2) + 2p_2 xy \end{aligned}$$

单目相机的成像过程

单目相机的成像过程:

1. 世界坐标系下有一个固定的原点P ,其**世界坐标**P_W
2. 由于相机在运动,它的运动由R,t或变换矩阵$T \in SE(3)$描述.原点P的**相机坐标**$\tilde{P_c} = RP_W + t$
3. 这时$\tilde{P_c}$的分量为X,Y,Z,把它们投影到归一化平面$Z=1$上,得到P的**归一化相机坐标**$P_c = \frac{\tilde{P_c}}{Z} = [\frac{X}{Z}, \frac{Y}{Z}, 1]^T$
4. 有畸变时,根据畸变参数计算P_c发生畸变后的归一化相机坐标
5. P的**归一化相机坐标**P_c经过内参K后,对应到它的**像素坐标**$P_{uv} = KP_c$

在讨论相机成像模型时,我们一共谈到了四种坐标: **世界坐标、相机坐标、归一化相机坐标**和**像素坐标**.请读者厘清它们的关系,它反映了整个成像的过程.

ch06 非线性优化

状态估计问题

最大后验与最大似然

SLAM模型由状态方程和运动方程构成:

$$\begin{cases} x_k &= f(x_{k-1}, u_k, w_k) \\ z_{k,j} &= h(y_j, x_k, v_{k,j}) \end{cases}$$

通常假设两个噪声项w_k ,$v_{k,j}$满足零均值的高斯分布:

$$w_k \sim \mathcal{N}(0, R_k),\ v_{k,j} \sim \mathcal{N}(0, Q_{k,j})$$

对机器人的估计,本质上就是已知**输入数据**u和**观测数据**z的条件下,求机器人位姿x和路标点y的条件概率分布:

$$P(x, y|z, u)$$

利用贝叶斯法则,有:

$$P(x, y|z, u) = \frac{P(z, u|x, y)P(x, y)}{P(z, u)} \propto P(z, u|x, y)P(x, y)$$

其中$P(x, y|z, u)$为**后验概率**,$P(z, u|x, y)$为**似然**,$P(x, y)$为**先验**,上式可表述为$后验概率 \propto 似然 \cdot 先验$.**直接求后验分布是困难的,但是求一个状态最优估计,使得在该状态下后验概率最大化则是可行的**:

$$(x, y)^*_{MAP} = \arg\max P(x, y|z, u) = \arg\max P(z, u|x, y)P(x, y)$$

求解**最大后验概率相当于最大化似然和先验的乘积**.因为x,y未知,即不知道先验,则可以求最大似然估计:

$$(x, y)^*_{MLE} = \arg\max P(z, u|x, y)$$

最大似然估计的直观意义为:**在什么样的状态下，最可能产生现在观测到的数据**.

最小二乘

基于观测数据z的最小二乘

对于某一次观测

$$z_{k,j} = h(y_j, x_k) + v_{k,j}$$

由于假设噪声$v_{k,j} \sim \mathcal{N}(0, Q_{k,j})$,则观测数据$z_{j,k}$的似然为

$$P(z_{j,k} | x_k, y_j) = \mathcal{N}(h(y_j, x_k), Q_{k,j})$$

将上式代入高斯分布表达式中,并取负对数,得到

$$\begin{aligned}(x_k, y_j)^* &= \arg\max \mathcal{N}(h(y_j, x_k), Q_{k,j}) \\ &= \arg\min \left((z_{k,j} - h(x_k, y_j))^T Q_{k,j}^{-1} (z_{k,j} - h(x_k, y_j))\right)\end{aligned}$$

上式等价于最小化噪声项(即误差)的一个二次型,其中$Q_{k,j}^{-1}$称为**信息矩阵**,即高斯分布协方差矩阵的逆.

基于观测数据z和输入数据u的最小二乘

因为观测z和输入u是独立的,因此可对z和u的联合似然进行因式分解:

$$P(x, y | z, u) = \prod_k P(u_k | x_{k-1}, x_k) \prod_{k,j} P(z_{j,k} | x_k, y_j)$$

定义输入和观测数据与模型之间的误差:

$$\begin{aligned}e_{u,k} &= x_k - f(x_{k-1}, u_k) \\ e_{z,j,k} &= z_{k,j} - h(x_k, y_j)\end{aligned}$$

定义

$$J(x, y) = \sum_k e_{u,k}^T R_k^{-1} e_{u,k} + \sum_k \sum_j e_{z,k,j}^T Q_{k,j}^{-1} e_{z,k,j}$$

则有

$$(x_k, y_j)^* = \arg\min J(x, y)$$

非线性最小二乘

对于非线性最小二乘问题:

$$\min_x F(x) = \frac{1}{2} ||f(x)||_2^2$$

求解该问题的具体步骤如下:

1. 给定某个初始值x_0
2. 对于第k次迭代,寻找一个增量Δx_k,使得$||F(x_k + \Delta x_k)||_2^2$达到极小值
3. 若Δx_k足够小,则停止
4. 否则,令$x_{k+1} = x_k + \Delta x_k$,返回第2步

这样,最小二乘问题被转化为一个不断寻找下降增量Δx_k的问题.,具体有以下方法

一阶和二阶梯度法

将目标函数 $F(x)$在x_k附近进行泰勒展开

$$F(x_k + \Delta x_k) \approx F(x_k) + J(x_k)^T \Delta x_k + \frac{1}{2} \Delta x_k^T H(x_k) x_k$$

其中$J(x)$是$F(x)$关于x的一阶导数矩阵,$H(x)$是$F(x)$关于x的二阶导数矩阵.

- 若Δx_k取一阶导数,则

$$\Delta x_k^* = -J(x_k)$$

- 若Δx_k取二阶导数,则

$$\Delta x_k^* = \arg\min\left(F(x_k) + J(x_k)^T \Delta x_k + \frac{1}{2}\Delta x_k^T H(x_k) x_k\right)$$

  令上式对Δx_k导数等于0,则Δx_k^*可以取$H\Delta x_k = -J$的解.

高斯牛顿法

将$f(x_k)$而非$F(x_k)$在x_k附近进行泰勒展开

$$f(x_k + \Delta x_k) \approx f(x_k) + J(x_k)^T \Delta x_k$$

则

$$\Delta x_k^* = \arg\min_{\Delta x_k} \frac{1}{2} ||f(x_k) + J(x_k)^T \Delta x_k||^2$$

令上式对Δx的导数为0,得到**高斯牛顿方程**

$$J(x_k) f(x_k) + J(x_k) J^T(x_k) \Delta x_k = 0$$

令$H(x) = J(x) J^T(x), g(x) = -J(x) f(x)$,则$\Delta x_k^*$可以取$H\Delta x_k = g$的解.

列文伯格-马夸尔特方法

泰勒展开只能在展开点附近才有较好的近似效果,因此应给Δx添加一个范围,称为**信赖区域**.

定义一个指标ρ刻画这个近似的好坏程度,其分子为实际函数下降的值,分母是近似模型下降的值:

$$\rho = \frac{f(x + \Delta x) - f(x)}{J(x)^T \Delta x}$$

通过调整ρ来确定信赖区域:

- 若ρ接近1,则近似是最好的.
- 若ρ太小,说明实际下降的值远小于近似下降的值,则认为近似比较差,需要缩小近似范围.
- 若ρ太大,说明实际下降的比预计的更大,我们可以放大近似范围.

改良版的非线性优化框架如下:

1. 给定初始值x_0,以及初始优化半径μ
2. 对于第k次迭代,求解:

$$\min_{\Delta x_k} \frac{1}{2} ||f(x_k) + J(x_k)^T \Delta x_k||^2 \quad s.t. ||D\Delta x_k||^2 \leq \mu$$

   其中,μ是信赖区域的半径,D为系数矩阵

3. 计算ρ
4. 若$\rho > \frac{3}{4}$则$\mu = 2\mu$
5. 若$\rho < \frac{1}{4}$则$\mu = 0.5\mu$
6. 若ρ大于某阈值,则认为近似可行.令$x_{k+1} = x_k + \Delta x_k$
7. 判断算法是否收敛.如不收敛则返回第2步,否则结束.

第2步中Δx_k的求解要使用拉格朗日乘数法:

$$\mathcal{L}(\Delta x_k, \lambda) = \frac{1}{2}||f(x_k) + J(x_k)^T \Delta x_k||^2 + \frac{\lambda}{2}(||D\Delta x_k||^2 - \mu)$$

令上式对Δx_k导数为0,得到

$$(H + \lambda D^T D)\Delta x_k = g$$

考虑简化形式,即$D = I$,则相当于求解

$$(H + \lambda I)\Delta x_k = g$$

- 当λ较小时,H占主要地位,这说明二次近似模型在该范围内是比较好的,列文伯格-马夸尔特方法更接近于高斯牛顿法.
- 当λ比较大时,λI占据主要地位,这说明二次近似模型在该范围内不够好,列文伯格-马夸尔特方法更接近于一阶梯度下降法.

文章目录

ch07 视觉里程计01

特征点匹配

特征点

根据特征点匹配计算相机运动

根据特征点匹配计算相机运动.根据相机的成像原理不同,分为以下3种情况:

1. 当相机为单目时,我们只知道匹配点的像素坐标,是为2D-2D匹配,使用对极几何求解.
2. 当相机为双目或RGB-D时,我们就知道匹配点的像素坐标和深度坐标,是为3D-3D匹配,使用ICP求解.
3. 如果有3D点及其在相机的投影位置,也能估计相机的运动,是为3D-2D匹配,使用PnP求解.

2D-2D匹配: 对极几何

对极约束

[外链图片转存失败,源站可能有防盗链机制,建议将图片保存下来直接上传(img-QVwt5blH-1587570602884)(1587436458419.png)]{:height="50%" width="50%"}

假设我们要求取两帧图像I_1, I_2之间的运动,设第一帧到第二帧的运动为R, t,两个相机中心分别为O_1, O_2.考虑I_1中有一个特征点p_1,它在I_2中对应着特征点p_2.连线$\overrightarrow{O_1 p_1}$和$\overrightarrow{O_2 p_2}$在三维空间中交于点P,这时点O_1, O_2, P三个点可以确定一个平面,称为**极平面**.$O_1 O_2$连线与像平面I_1, I_2的交点分别为e_1, e_2.e_1, e_2称为极点,$O_1 O_2$称为基线,极平面与两个像平面I_1, I_2之间的相交线l_1, l_2称为极线.

P在I_1下的相机坐标为$P = [X, Y, Z]^T$,两个投影像素点p_1, p_2的像素位置为$s_1 p_1 = KP$,$s_2 p_2 = K(RP + t)$.

取p_1, p_2的归一化坐标$x_1 = K^{-1} p_1$,$x_1 = K^{-1} p_2$,则可以推得$x_2 \simeq R x_1 + t$.上式中$\simeq$表示尺度意义上相等,即在齐次坐标下是相等的,物理上表示对原点成投影关系.

经过推导,得到:

$$x_2^T t^\wedge R x_1 = 0 \tag{1}$$

代入p_1, p_2,得到:

$$p_2^T K^{-T} t^\wedge R K^{-1} p_1 \tag{2}$$

式(1)和式(2)都称为对极约束,定义基础矩阵F和本质矩阵E,可以进一步简化对极约束:

$$\mathrm{E} = \mathrm{t}^{\wedge}\mathrm{R} \qquad \mathrm{F} = \mathrm{K}^{-\mathrm{T}}\mathrm{E}\mathrm{K}^{-1} \qquad \mathrm{x}_2^{\mathrm{T}}\mathrm{E}\mathrm{x}_1 = \mathrm{p}_2^{\mathrm{T}}\mathrm{F}\mathrm{p}_1 = 0 \tag{3}$$

由于E与F之间只差了相机内参,相机内参是已知的,因此实践中往往使用形式更简单的E.

本质矩阵E的求解

考虑到E的尺度等价性,可以用8对点来估计E,是为八点法.

对于一对匹配点,其归一化坐标$\mathrm{x}_1 = [\mathrm{u}_1, \mathrm{v}_1, 1]^{\mathrm{T}}$,$\mathrm{x}_2 = [\mathrm{u}_2, \mathrm{v}_2, 1]^{\mathrm{T}}$.根据对极约束,有

$$(\mathrm{u}_1, \mathrm{v}_1, 1)\begin{pmatrix} \mathrm{e}_1 & \mathrm{e}_2 & \mathrm{e}_3 \\ \mathrm{e}_4 & \mathrm{e}_5 & \mathrm{e}_6 \\ \mathrm{e}_7 & \mathrm{e}_8 & \mathrm{e}_9 \end{pmatrix}\begin{pmatrix} \mathrm{u}_2 \\ \mathrm{v}_2 \\ 1 \end{pmatrix} = 0$$

把矩阵E展开为向量$\mathrm{e} = [\mathrm{e}_1, \mathrm{e}_2, \mathrm{e}_3, \mathrm{e}_4, \mathrm{e}_5, \mathrm{e}_6, \mathrm{e}_7, \mathrm{e}_8, \mathrm{e}_9]^{\mathrm{T}}$,对极约束可以写成与e有关的线性形式:

$$[\mathrm{u}_1\mathrm{u}_2, \mathrm{u}_1\mathrm{v}_2, \mathrm{u}_1, \mathrm{v}_1\mathrm{u}_2, \mathrm{v}_1\mathrm{v}_2, \mathrm{v}_2, \mathrm{u}_2, \mathrm{v}_2, 1] \cdot \mathrm{e} = 0$$

把八对点对应的x_1,x_2分别代入方程中,得到线性方程组:

$$\begin{pmatrix} \mathrm{u}_1^1\mathrm{u}_2^1 & \mathrm{u}_1^1\mathrm{v}_2^1 & \mathrm{u}_1^1 & \mathrm{v}_1^1\mathrm{u}_2^1 & \mathrm{v}_1^1\mathrm{v}_2^1 & \mathrm{v}_2^1 & \mathrm{u}_2^1 & \mathrm{v}_2^1 & 1 \\ \mathrm{u}_1^1\mathrm{u}_2^2 & \mathrm{u}_1^2\mathrm{v}_2^2 & \mathrm{u}_1^2 & \mathrm{v}_1^2\mathrm{u}_2^2 & \mathrm{v}_1^2\mathrm{v}_2^2 & \mathrm{v}_2^2 & \mathrm{u}_2^2 & \mathrm{v}_2^2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathrm{u}_1^1\mathrm{u}_2^8 & \mathrm{u}_1^8\mathrm{v}_2^8 & \mathrm{u}_1^8 & \mathrm{v}_1^8\mathrm{u}_2^8 & \mathrm{v}_1^8\mathrm{v}_2^8 & \mathrm{v}_2^8 & \mathrm{u}_2^8 & \mathrm{v}_2^8 & 1 \end{pmatrix}\begin{pmatrix} \mathrm{e}_1 \\ \mathrm{e}_2 \\ \mathrm{e}_3 \\ \mathrm{e}_4 \\ \mathrm{e}_5 \\ \mathrm{e}_6 \\ \mathrm{e}_7 \\ \mathrm{e}_8 \\ \mathrm{e}_9 \end{pmatrix} = 0$$

求得E后,对E进行SVD分解以求取R,t:设E的SVD分解为$\mathrm{E} = \mathrm{U}\Sigma\mathrm{V}^{\mathrm{T}}$,则对应的$\mathrm{R}$,$\mathrm{t}$分别为:

$$\mathrm{t}^{\wedge} = \mathrm{U}\mathrm{R}_{\mathrm{Z}}(\frac{\pi}{2})\Sigma\mathrm{U}^{\mathrm{T}} \qquad \mathrm{R} = \mathrm{U}\mathrm{R}_{\mathrm{Z}}^{\mathrm{T}}(\frac{\pi}{2})\Sigma\mathrm{U}^{\mathrm{T}}$$

其中$\mathrm{R}_{\mathrm{Z}}(\frac{\pi}{2})$表示沿$\mathrm{Z}$轴旋转90°得到的旋转矩阵.

对极几何的讨论

1. 尺度不确定性: 2D图像不具有深度信息,这导致了**单目视觉的尺度不确定性**.

   实践中设t为单位1,计算相机运动和和特征点的3D位置,这被称为单目SLAM的**初始化**.

2. 初始化的纯旋转问题: 若相机发生纯旋转,导致t为零,得到的E也将为零,会导致我们无从求解R.因此**单目初始化不能只有纯旋转,必须要有一定程度的平移**.

3. 多于8对点的情况:

   对于八点法,有$\mathrm{A}\mathrm{e} = 0$,其中A为一个8×9的矩阵.

   若匹配点的个数多于8个,A的尺寸变化,上述方程不成立.因此转而求取最小化二次型

   $$\min_{\mathrm{e}} ||\mathrm{A}\mathrm{e}||_2^2 = \min_{\mathrm{e}} \mathrm{e}^{\mathrm{T}}\mathrm{A}^{\mathrm{T}}\mathrm{A}\mathrm{e}$$

   是为最小二乘意义下的E矩阵.

3D-2D匹配: PnP(Perspective-n-Point)

2D-2D的对极几何方法需要8个或8个以上的点对（以八点法为例），且存在着初始化、纯旋转和尺度的问题。然而，如果两张图像中其中一张特征点的3D位置已知，那么最少只需3个点对（需要至少一个额外点验证结果）就可以估计相机运动。

在双目或RGB-D的视觉里程计中，我们可以直接使用PnP估计相机运动。而在单目视觉里程计中，必须先进行初始化，然后才能使用PnP。

PnP问题有多种解决方法:

1. 直接线性表变换(DLT): 先求解相机位姿,再求解空间点位置
2. P3P: 先求解空间点位置,再求解相机位姿
3. Bundle Adjustment: 最小化重投影误差,同时求解空间点位置和相机位姿

直接线性变换(DLT): 先求解相机位姿,再求解空间点位置

考虑某个空间点P的**齐次世界坐标**为$P = (X, Y, Z, 1)^T$.在图像I_1中投影到特征点的**归一化像素坐标**$x_1 = (u_1, v_1, 1)^T$.此时相机的位姿R,t是未知的,定义增广矩阵$[R|t]$(不同于变换矩阵T)为一个3×4的矩阵,包含了旋转与平移信息,展开形式如下:

$$s\begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = \begin{pmatrix} t_1 & t_2 & t_3 & t_4 \\ t_5 & t_6 & t_7 & t_8 \\ t_9 & t_{10} & t_{11} & t_{12} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

用最后一行把s消去,得到两个约束:

$$\begin{cases} \boldsymbol{t}_1^T P - \boldsymbol{t}_3^T P u_1 = 0 \\ \boldsymbol{t}_2^T P - \boldsymbol{t}_3^T P v_1 = 0 \end{cases}$$

其中$\boldsymbol{t}_1 = (t_1, t_2, t_3, t_4)^T$,$\boldsymbol{t}_2 = (t_5, t_6, t_7, t_8)^T$,$\boldsymbol{t}_3 = (t_9, t_{10}, t_{11}, t_{12})^T$.$\boldsymbol{t}_1$,$\boldsymbol{t}_2$,$\boldsymbol{t}_3$为待求量.

将N对匹配的特征点代入方程中,得到线性方程组:

$$\begin{pmatrix} P_1^T & 0 & -u_1 P_1^T \\ 0 & P_1^T & -v_1 P_1^T \\ \vdots & \vdots & \vdots \\ P_N^T & 0 & -u_N P_N^T \\ 0 & P_N^T & -v_N P_N^T \end{pmatrix} \begin{pmatrix} \boldsymbol{t}_1 \\ \boldsymbol{t}_2 \\ \boldsymbol{t}_3 \end{pmatrix} = 0$$

只需6对匹配点即可求解增广矩阵$[R|t]$,若匹配点数多于6对时,可以求最小二乘解.对于求解出的旋转矩阵R,可以通过QR分解等手段将其投影到$SE(3)$上.

P3P: 先求解空间点位置,再求解相机位姿

[外链图片转存失败,源站可能有防盗链机制,建议将图片保存下来直接上传(img-9IuduXXH-1587570602886)(1587451327097.png)]

已知3对匹配点的**世界坐标**A,B,C和**投影坐标**a,b,c,根据三角形的余弦定理,有

$$\begin{cases} OA^2 + OB^2 - 2OA \cdot OB \cdot \cos\langle a, b\rangle = AB^2 \\ OB^2 + OC^2 - 2OB \cdot OC \cdot \cos\langle b, c\rangle = BC^2 \\ OA^2 + OC^2 - 2OA \cdot OC \cdot \cos\langle a, c\rangle = AC^2 \end{cases}$$

记$x = OA/OC$,$y = OB/OC$,$u = BC^2/AB^2$,$v = AC^2/AB^2$

$$\begin{cases} (1-u)y^2 - ux^2 - \cos\langle b, c\rangle y + 2uxy\cos\langle a, b\rangle + 1 = 0 \\ (1-w)x^2 - wy^2 - \cos\langle a, c\rangle y + 2wxy\cos\langle a, b\rangle + 1 = 0 \end{cases}$$

上式中,三个余弦角$\cos\langle a, b\rangle$,$\cos\langle b, c\rangle$,$\cos\langle a, c\rangle$以及u,v是已知的,可以求解出x,y,进而求解出A,B,C三点的相机坐标.然后根据3D-3D的点对,计算相机的运动R,t.

Bundle Adjustment: 最小化重投影误差,同时求解空间点位置和相机位姿

设相机位姿变换矩阵T,某空间点的世界坐标$P_i = [X_i, Y_i, Z_i]^T$,其投影的像素坐标为$\boldsymbol{u}_i = [u_i, v_i]^T$,像素位置与空间点位置的关系如下:

$$s_i \boldsymbol{u}_i = KTP_i$$

由于相机位姿未知及观测点的噪声,上式存在一个误差,称为**重投影误差**$e = u_i - \frac{1}{s_i}KTP_i$.因此我们对重投影误差求和,寻找最好的相机位姿和特征点的空间位置,最小化重投影误差:

$$T^* = \arg\min_T \frac{1}{2}\sum_{i=1}^{n} ||u_i - \frac{1}{s_i}KTP_i||^2$$

$$P_i^* = \arg\min_{P_i} \frac{1}{2} \sum_{i=1}^{n} ||u_i - \frac{1}{s_i} K T P_i||^2$$

使用最小二乘优化,要分别求e对T和P的导数:

$$e(x + \Delta x) \approx e(x) + J \Delta x$$

- 求e对T的导数:

  当e为像素坐标误差(2维),x为相机位姿(6维)时,J将是一个2×6的矩阵.我们来推导J的形式:

  取中间变量$P' = (TP)_{1:3} = [X', Y', Z']^T$

  使用李代数求导的扰动模型,对T左乘微小扰动δξ,求导得到:

  $$\frac{\partial e}{\partial \delta \xi} = \lim_{\delta \xi = 0} \frac{e(\delta \xi \oplus \xi) - e(\xi)}{\delta \xi} = \frac{\partial e}{\partial P'} \frac{\partial P'}{\partial \delta \xi}$$

  其中的⊕表示李代数的左乘扰动

  其中第一项$\frac{\partial e}{\partial P'}$:

  $$\frac{\partial e}{\partial P'} = -\begin{bmatrix} \frac{\partial u}{\partial X'} & \frac{\partial u}{\partial Y'} & \frac{\partial u}{\partial Z'} \\ \frac{\partial v}{\partial X'} & \frac{\partial v}{\partial Y'} & \frac{\partial v}{\partial Z'} \end{bmatrix} = -\begin{bmatrix} \frac{f_x}{Z'} & 0 & -\frac{f_x X'}{Z'^2} \\ 0 & \frac{f_y}{Z'} & -\frac{f_y Y'}{Z'^2} \end{bmatrix}$$

  第二项$\frac{\partial P'}{\partial \delta \xi}$为变换后的点关于李代数的导数:

  $$\frac{\partial P'}{\partial \delta \xi} = \frac{(TP)}{\partial \delta \xi} = (TP)^{\odot} = \begin{bmatrix} I & -P'^{\wedge} \\ 0^T & 0^T \end{bmatrix}$$

  在P'定义中,取出前三维,得到

  $$\frac{\partial P'}{\partial \delta \xi} = [I, -P'^{\wedge}]$$

  将两项相乘,得到了2×6的雅可比矩阵J^T

  $$J^T = \frac{\partial e}{\partial \delta \xi} = -\begin{bmatrix} \frac{f_x}{Z'} & 0 & -\frac{f_x X'}{Z'^2} & -\frac{f_x X' Y'}{Z'^2} & f_x + \frac{f_x X'^2}{Z'^2} & -\frac{f_x Y'}{Z'} \\ 0 & \frac{f_y}{Z'} & -\frac{f_y Y'}{Z'^2} & -f_y - \frac{f_y Y'^2}{Z'^2} & \frac{f_y X' Y'}{Z'^2} & \frac{f_y X'}{Z'} \end{bmatrix}$$

- 求e对P的导数

3D-3D匹配: ICP

对于一组已配对好的3D点:

$$P = \{p_1, \cdots, p_n\}, \quad P' = \{p_1', \cdots, p_n'\}$$

现在,想要找一个欧氏变换R, t,使得:

$$\forall i, \quad p_i = R p_i' + t$$

ICP问题的求解包含两种方式:

1. 利用线性代数的求解(主要是SVD)
2. 利用非线性优化方式的求解(类似于Bundle Adjustment)

SVD方法

定义第i对点的误差项为$e_i = p_i - (Rp_i' + t)$,定义两组点的质心$p = \frac{1}{n} \sum_{i=1}^{n} (p_i)$,$p' = \frac{1}{n} \sum_{i=1}^{n} (p_i')$

构建最小二乘问题,求取最合适的R, t.

$$
\begin{aligned}
\min_{R,t} J &= \frac{1}{2}\sum_{i=1}^{n}||(p_i-(Rp_i'+t))||_2^2 \\
&= \frac{1}{2}\sum_{i=1}^{n}||p_i-p-R(p_i'-p')||^2+||p-Rp'-t||^2
\end{aligned}
$$

左边只和旋转矩阵R相关,而右边既有R也有t,但只和质心相关.因此令左边取最小值解出R,代入到右边令式子等于0求出t.

定义去质心坐标$q_i = p_i - p, q_i' = p_i' - p'$,则优化目标可写成:

$$
\begin{aligned}
R^* &= \min_R \sum_{i=1}^{n}||p_i-p-R(p_i'-p')||^2 \\
&= \min_R \sum_{i=1}^{n} -q_i^T R q_i' \\
&= -tr\left(R\sum_{i=1}^{n} q_i' q_i^T\right)
\end{aligned}
$$

省略数学证明,定义矩阵:

$$
W = \sum_{i=1}^{n} q_i q_i'^T
$$

对矩阵W进行SVD分解得到:

$$
W = U\Sigma V^T
$$

可求解

$$
R = UV^T
$$

非线性优化方法

使用李代数表达表达位姿,目标函数可以写成

$$
\min_{\xi} = \frac{1}{2}\sum_{i=1}^{n}||(p_i-\exp(\xi^\wedge)p_i')||_2^2
$$

误差项关于位姿的导数可以用李代数求导的扰动模型,计算导数得到:

$$
\frac{\partial e}{\partial \delta\xi} = -(\exp(\xi^\wedge)p_i')^\odot
$$

可以直接使用最小二乘优化方法求解位姿.

>