

目录

1 什么是 slam	1
2 刚体的三维运动	2
3 李群与李代数	5
3.1 李群	5
3.2 李代数	6
3.3 李代数求导与扰动模型	7
4 畸变和相机模型	8
5 极几何	9
5.1 本质矩阵	10
5.2 基础矩阵	11
5.3 单应性矩阵	11
6 概率运动模型	12
6.1 速度运动模型	12
6.2 里程计运动模型	13
7 概率观测模型	14
7.1 波束模型	14
7.2 似然域模型	15
7.3 概率图模型	15
7.4 概率图模型在 slam 中的应用	16
8 梯度、雅可比矩阵、海森矩阵	18
8.1 高斯-牛顿算法	19
9 ORB 特征	19
10 三角测量	20
11 3D-2D:PnP	20
12 最小化重投影误差求解 PnP	21
13 高精度地图的全局规划路径算法有哪些	21

1 什么是 slam

运动方程

机器人或者一个物体在空间中的运动都可以抽象为一个数学模型

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k) \quad (1.1)$$

其中, \mathbf{u}_k 表示运动传感器的读书或者输入, \mathbf{w}_k 表示该过程中加入的噪声, \mathbf{x}_k 表示该时刻运动物体所在位置, 函数 f 描述这一个过程, 即使不清楚这个过程的具体数学表达。

观测方程

与运动方程相对应的还有一个观测方程, 它描述的是运动机器人在 $\text{textbf{x}}_k$ 位置上观测到某些物体 \mathbf{y}_k 产生的一个观测数据 $\text{textbf{z}}_{k,j}$, 同样可以用一个抽象的函数 h 来表示这样一个关系:

$$\mathbf{z}_{k,j} = h(\mathbf{y}_i, \mathbf{x}_k, \mathbf{v}_{k,j}) \quad (1.2)$$

这里 $\mathbf{v}_{k,j}$ 是这一次的观测噪声。

上述两个方程描述了最基本的 SLAM 问题，当知道运动测量读书 u ，以及传感器的读书 z 时，如何求解定位问题（估计 \mathbf{x} ）和建图问题（估计 \mathbf{y} ），这样就把 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。

2 刚体的三维运动

内积

$$\mathbf{a} \cdot \mathbf{b} = \mathbf{a}^T \mathbf{b} = \sum_{i=1}^3 a_i b_i = |\mathbf{a}| |\mathbf{b}| \cos \angle \mathbf{a}, \mathbf{b} > \quad (2.1)$$

外积

$$\mathbf{a} \times \mathbf{b} = \begin{vmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix} = \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \mathbf{b} \stackrel{def}{=} \mathbf{a} \wedge \mathbf{b} \quad (2.2)$$

外积的结果是一个向量，它的方向垂直于这两个向量，大小为 $|\mathbf{a}| |\mathbf{b}| \sin \angle \mathbf{a}, \mathbf{b} >$ ，是两个向量张成的四边形的有向面积。

旋转矩阵

旋转矩阵各分量从定义推导上来看是两个坐标系基的内积，由于基向量的长度为 1，所以实际上是各个基向量的夹角的余弦值，因此也称作**方向余弦矩阵**，它实际上是一个行列式为 1 的正交矩阵。

对于 n 维旋转矩阵集合定义如下：

$$SO(n) = \{\mathbf{R} \in \mathbb{R}^{n \times n} | \mathbf{R} \mathbf{R}^T = \mathbf{I}, \det(\mathbf{R}) = 1\} \quad (2.3)$$

特别的 $SO(3)$ 指三维空间的旋转，通过旋转矩阵就可以直接讨论两个坐标系之间的转换，而不用再从基开始谈起。

由于旋转矩阵为正交矩阵，它的逆（即转置）描述了一个相反的旋转，即 $\mathbf{R}^{-1}, \mathbf{R}^T$ 刻画来了一个与 \mathbf{R} 相反的旋转矩阵。

对于两个坐标系 1 和坐标系 2 来说，向量 \mathbf{a} 在两个坐标系下的坐标为 $\mathbf{a}_1, \mathbf{a}_2$ ，它们之间的关系应该为：

$$\mathbf{a}_1 = \mathbf{R}_{12} \mathbf{a}_2 + \mathbf{t}_{12} \quad (2.4)$$

这里的 \mathbf{R}_{12} 是指把坐标系 2 的向量变换到坐标系 1 中，关于平移 \mathbf{t}_{12} ，它实际意义上指的是坐标系 1 的原点指向坐标系 2 原点的向量。注意在两个坐标系的相互变换中，其中的 \mathbf{t}_{12} 和 \mathbf{t}_{21} 并不存在 $\mathbf{t}_{12} = -\mathbf{t}_{21}$ 这样的关系，因为这两个坐标系之间还有旋转变换关系。尽管从空间上看两个平移向量确实为相反方向，但在数值上并不是简单的取反操作

上述坐标系之间的变换关系在实现多次变换时会显得十分繁琐，如进行了两次变换的表达式为 $\mathbf{c} = \mathbf{R}_2(\mathbf{R}_1 \mathbf{a} + \mathbf{t}_1) + \mathbf{t}_2$ 。因此引入齐次坐标系（齐次坐标系的由来是个巧妙的）：

$$\begin{bmatrix} \mathbf{a}' \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ 1 \end{bmatrix} \stackrel{def}{=} \mathbf{T} \begin{bmatrix} \mathbf{a} \\ 1 \end{bmatrix} \quad (2.5)$$

矩阵 \mathbf{T} 称为**变换矩阵**，变换为齐次坐标系后可以将旋转和平移写在同一个矩阵中，使得整个变换关系变为线性关系，多次坐标系变换就可以直接右乘。

$$\tilde{\mathbf{c}} = \mathbf{T}_2 \mathbf{T}_1 \tilde{\mathbf{a}} \quad (2.6)$$

像上述这种变换矩阵的形式，又称之为特殊欧氏群：

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

同样的求解该矩阵的一个逆表示一个反向的变换：

$$\mathbf{T}^{-1} = \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (2.8)$$

旋转向量和欧拉角

对于旋转矩阵来说用了 9 个量来表示坐标系的旋转，变换矩阵更是用了 16 个量来表示刚体坐标系的变换，这种表达方式有些冗余但直观。算了直接给出旋转向量的定义 (为了解决旋转矩阵的冗余表达)，任何的旋转都可以用一个旋转轴和一个旋转角来表示，于是可以用一个三维向量来描述旋转，**三维向量方向与旋转轴一致，而长度等于旋转角**。这样的向量称为**旋转向量**，因此，对于变换矩阵可用一个旋转向量和一个平移向量表示，这时变量个数为 6 个。

假设旋转轴为一个单位长度的向量 \mathbf{n} ，旋转角为 θ ，那么向量 $\theta\mathbf{n}$ 也可以描述一个旋转过程，同旋转矩阵 \mathbf{R} 一样。两种表达方式可以由**罗德里格斯公式**表明，推导过程较复杂，直接给出结果：

$$\mathbf{R} = \cos \theta \mathbf{I} + (1 - \cos \theta) \mathbf{nn}^T + \sin \theta \mathbf{n}^\wedge \quad (2.9)$$

符号 \wedge 是向量到反对称矩阵的转化符。

计算一个旋转矩阵到旋转向量的转换，对于旋转角，可以计算上面旋转向量到旋转矩阵等式两边的迹的到：

$$\begin{aligned} \text{tr}(\mathbf{R}) &= \cos \theta \text{tr}(\mathbf{I}) + (1 - \cos \theta) \text{tr}(\mathbf{nn}^T) + \sin \theta \text{tr}(\mathbf{n}^\wedge) \\ &= 3 \cos \theta + (1 - \cos \theta) \\ &= 1 + 2 \cos \theta \end{aligned} \quad (2.10)$$

因此：

$$\theta = \arccos \frac{\text{tr}(\mathbf{R}) - 1}{2} \quad (2.11)$$

关于旋转轴 \mathbf{n} ，在旋转轴上向量在旋转后不发生改变，说明：

$$\mathbf{Rn} = \mathbf{n} \quad (2.12)$$

因此，转轴 \mathbf{n} 是矩阵 \mathbf{R} 特征值 1 对应的特征向量，求解此方程再归一化就得到了旋转轴。

欧拉角

欧拉角有一个万向锁的问题，所以不经常用。

四元数

在复平面，一个复向量的旋转 θ 可以乘 $e^{i\theta}$ 来表示，这个是极坐标的表示，如果需要用到普通形式，用欧拉公式转换一下就好。在三维旋转也可用一个单位四元数来表示，四元数 \mathbf{q} 拥有一个实部和三个虚部： $\mathbf{q} = q_0 + q_1i + q_2j + q_3k$ ，这三个虚部满足以下关系：

$$\begin{cases} i^2 = j^2 = k^2 = -1 \\ ij = k, ji = -k \\ jk = i, kj = -i \\ ki = j, ik = -j \end{cases} \quad (2.13)$$

如果将 i, j, k 看成三个坐标轴，那么它们与自己的乘法和复数一样，相互之间的乘法和外积一样。四元数的另外一种表示方式：

$$\mathbf{q} = [\mathbf{s}, \mathbf{v}]^T, \quad \mathbf{s} = q_0 \in \mathbb{R}, \quad \mathbf{v} = [q_1, q_2, q_3]^T \in \mathbb{R}^3 \quad (2.14)$$

这里 \mathbf{s} 为四元数的实部， \mathbf{v} 为虚部。

四元数的运算

1、加法和减法

$$\mathbf{q}_a \pm \mathbf{q}_b = [\mathbf{s}_a \pm \mathbf{s}_b, \mathbf{v}_a \pm \mathbf{v}_b]^T \quad (2.15)$$

2、乘法，就是将每一项相乘

$$\begin{aligned}\mathbf{q}_a \mathbf{q}_b = & \mathbf{s}_a \mathbf{s}_b - x_a x_b - y_a y_b - z_a z_b \\ & + (s_a x_b + x_a s_b + y_a z_b - z_a y_b)i \\ & + (s_a y_b - x_a z_b + y_a s_b + z_a x_b)j \\ & + (s_a z_b + x_a y_b - y_a x_b + z_a s_b)k\end{aligned}\quad (2.16)$$

上式复杂但有规律，如果改写成向量形式并利用内外积运算，会更加简洁：

$$\mathbf{q}_a \mathbf{q}_b = [\mathbf{s}_a \mathbf{s}_b - \mathbf{v}_a^T \mathbf{v}_b, \mathbf{s}_a \mathbf{v}_b + \mathbf{s}_b \mathbf{v}_a + \mathbf{v}_a \times \mathbf{v}_b]^T \quad (2.17)$$

四元数的乘法通常是不可交换的，除非 \mathbf{v}_a 和 \mathbf{v}_b 是共线的，这时最后一项外积为零。

3、模长

四元数模场定义为：

$$\|\mathbf{q}_a\| = \sqrt{s_a^2 + x_a^2 + y_a^2 + z_a^2} \quad (2.18)$$

可以验证的是： $\|\mathbf{q}_a \mathbf{q}_b\| = \|\mathbf{q}_a\| \|\mathbf{q}_b\|$

4、共轭

四元数的共轭就是把虚部取成相反数，四元数共轭与其本身相乘会得到一个实四元数，实部为其模长的平方。

5、逆

$$\mathbf{q}^{-1} = \mathbf{q}^* / \|\mathbf{q}\|^2 \quad (2.19)$$

按照定义四元数和自己的逆的乘积为实四元数 **1**

6、数乘

和向量矩阵的数乘一样。

用四元数表示旋转

有一个空间三维点 $\mathbf{p} = [x, y, z] \in \mathbb{R}^3$ ，以及一个单位四元数 \mathbf{q} 指定的旋转，那么旋转后的点 $\mathbf{p}' = \mathbf{q} \mathbf{p} \mathbf{q}^{-1}$ ，其中把三维空间点 \mathbf{p} 用一个虚四元数来表述 $\mathbf{p} = [0, x, y, z]^T = [0, \mathbf{v}]^T$ ，最后把 \mathbf{p}' 的虚部取出即得旋转之后得坐标。

四元数到其它旋转表示的转换

四元数到其他旋转的推导过程比较复杂，直接给出四元数到旋转向量的转换公式如下：

$$\begin{cases} \theta = 2 \arccos q_0 \\ [n_x, n_y, n_z]^T = [q_1, q_2, q_3]^T / \sin \frac{\theta}{2} \end{cases} \quad (2.20)$$

相似、仿射、射影变换

1、相似变换

相似变换比欧式变换多了一个自由度，它允许物体进行均匀缩放，其变换矩阵为

$$\mathbf{T}_s = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (2.21)$$

其中旋转部分多了一个缩放因子 s ，表示在对向量旋转过后，可以在 x, y, z 三个坐标上进行均匀缩放。

2、仿射变换

与欧式变换不同的是，仿射变换只要求 \mathbf{A} 是一个可逆矩阵，而不必是正交矩阵，仿射变换也叫正交投影。

$$\mathbf{T}_A = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (2.22)$$

3、射影变换

射影变换是最一般最常见的变换，矩阵形式为

$$\mathbf{T}_p = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{a}^T & v \end{bmatrix} \quad (2.23)$$

左上角为可逆矩阵 \mathbf{A} ，右上角为平移 \mathbf{t} ，左下角为缩放 \mathbf{a}^T 。由于采用了齐次坐标，所以可以对整个矩阵除以 v 得到右下角不是 1 就是 0 的矩阵，于是 2D 的射影变换有 8 个自由度，3D 的射影变换有 15 个自由度。

3 李群与李代数

3.1 李群

群，只有一种（良好的）运算的集合叫做群。比如对于任何一个旋转或者变换矩阵加法是不封闭的，乘法是封闭的。群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 ，群则可以记作 $G = (A, \cdot)$ ，满足封闭性、结合律、幺元和逆的条件。

李代数的引出，由于旋转矩阵满足 $\mathbf{R}\mathbf{R}^T = \mathbf{I}$ ，现在考虑旋转矩阵为时间相关函数 $\mathbf{R}(t)\mathbf{R}(t)^T = \mathbf{I}$ ，在等式两边对时间求导得到

$$\dot{\mathbf{R}}(t)\mathbf{R}(t)^T + \mathbf{R}(t)\dot{\mathbf{R}}(t)^T = 0 \quad (3.1)$$

整理可以知道

$$\dot{\mathbf{R}}(t)\mathbf{R}(t)^T = -\mathbf{R}(t)\dot{\mathbf{R}}(t)^T \quad (3.2)$$

可以看出 $\dot{\mathbf{R}}(t)\mathbf{R}(t)^T$ 是一个反对称矩阵，根据前面所学的对于一个反对称矩阵一定唯一有个向量与之对应，这里引出符号 $\hat{\cdot}$ 来表示这种关系，对于一个向量 \mathbf{a} 和一个矩阵 A 有这样的关系存在

$$\mathbf{a}^\wedge = \mathbf{A} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \quad (3.3)$$

相反的有 $\mathbf{A}^\vee = \mathbf{a}$

于是对于反对称矩阵 $\dot{\mathbf{R}}(t)\mathbf{R}(t)^T = \phi(t)^\wedge, \phi(t) \in \mathbb{R}^3$ 等式两边右乘 $\mathbf{R}(t)$ ，由于 \mathbf{R} 为正交矩阵，有

$$\dot{\mathbf{R}}(t) = \phi(t)^\wedge \mathbf{R}(t) = \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix} \mathbf{R}(t) \quad (3.4)$$

由此可知对于旋转矩阵求导只需左乘一个 $\phi(t)^\wedge$ 矩阵即可，考虑 $t_0 = 0$ 时候，设定此时旋转矩阵为 \mathbf{I} ，按照导数定义可以把 $\mathbf{R}(t)$ 在 $t = 0$ 附近进行一阶泰勒展开

$$\mathbf{R}(t) \approx \mathbf{R}(t_0) + \dot{\mathbf{R}}(t_0)(t - t_0) = \mathbf{I} + \phi(t_0)^\wedge t \quad (3.5)$$

可以看到 ϕ 反映了 \mathbf{R} 的导数性质，故称它在 $SO(3)$ 原点附近的正切空间，同时在 t_0 附近，设 ϕ 保持为常数 $\phi(t_0) = \phi_0$ ，则有

$$\dot{\mathbf{R}}(t) = \phi(t_0)^\wedge \mathbf{R}(t) = \phi_0^\wedge \mathbf{R}(t) \quad (3.6)$$

上面是一个关于 \mathbf{R} 的微分方程，而且具有初始值 $\mathbf{R}(0) = \mathbf{I}$ ，求解可得

$$\mathbf{R}(t) = \exp(\phi_0^\wedge t) \quad (3.7)$$

由此可知旋转矩阵 \mathbf{R} 和另一个反对称矩阵 ϕ_0^\wedge 通过指数关系发生了联系。

3.2 李代数

每个李群都有之对应的李代数，李代数描述了李群的局部性质，准确来说式单位元附近的正切空间。

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

- 1 封闭性, $\forall \mathbf{X}, \mathbf{Y} \in \mathbb{V}, [\mathbf{X}, \mathbf{Y}] \in \mathbb{V}$
- 2 双线性, $\forall \mathbf{X}, \mathbf{Y}, \mathbf{Z} \in \mathbb{V}, a, b \in \mathbb{F}$, 有
 $[a\mathbf{X} + b\mathbf{Y}, \mathbf{Z}] = a[\mathbf{X}, \mathbf{Z}] + b[\mathbf{Y}, \mathbf{Z}], [\mathbf{Z}, a\mathbf{X} + b\mathbf{Y}] = a[\mathbf{Z}, \mathbf{X}] + b[\mathbf{Z}, \mathbf{Y}]$
- 3 自反性, $\forall \mathbf{X} \in \mathbb{V}, [\mathbf{X}, \mathbf{X}] = \mathbf{0}$
- 4 雅可比等价, $\forall \mathbf{X}, \mathbf{Y}, \mathbf{Z} \in \mathbb{V}, [\mathbf{X}, [\mathbf{Y}, \mathbf{Z}]] + [\mathbf{Z}, [\mathbf{X}, \mathbf{Y}]] + [\mathbf{Y}, [\mathbf{Z}, \mathbf{X}]] = \mathbf{0}$

其中二元运算被称为**李括号**

之前提到的 ϕ 事实上是一种李代数， $SO(3)$ 对应的李代数是定义在 $\mathbb{R}^{3 \times 3}$ 上的向量，记作 ϕ ，根据前面的推导，每个 ϕ 都可以生成一个反对称矩阵：

$$\phi^\wedge = \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix} \in \mathbb{R}^{3 \times 3} \quad (3.8)$$

由此定义下，两个向量 ϕ_1, ϕ_2 的李括号为

$$[\phi_1, \phi_2] = (\phi_1 \times \phi_2)^\vee \quad (3.9)$$

由此可知 $\mathfrak{so}(3) = \{\phi \in \mathbb{R}^3, \phi^\wedge \in \mathbb{R}^{3 \times 3}\}$

对于 $SE(3)$ ，它也有对应的李代数 $\mathfrak{se}(3)$ ，与 $\mathfrak{so}(3)$ 相似， $\mathfrak{se}(3)$ 位于 \mathbb{R}^6 空间中：

$$\mathfrak{se}(3) = \left\{ \xi = \begin{bmatrix} \rho \\ \phi \end{bmatrix} \in \mathbb{R}^6, \rho \in \mathbb{R}^3, \phi \in \mathfrak{so}(3), \xi^\wedge = \begin{bmatrix} \phi^\wedge & \rho \\ \mathbf{0}^T & 0 \end{bmatrix} \in \mathbb{R}^{4 \times 4} \right\} \quad (3.10)$$

我们把每个 $\mathfrak{se}(3)$ 元素记作 ξ ，它是一个六维向量，前三维为平移（但含义和变换矩阵中的平移不同），记作 ρ ，后面三维为旋转记作 ϕ ，实质上是 $\mathfrak{se}(3)$ 元素，同时拓展了 \wedge 符号的含义，在 $\mathfrak{se}(3)$ 中同样使用 \wedge 符号，将一个六维向量转换成四维度矩阵，不再表述为反对称。

$$\xi^{s\phi at} = \begin{bmatrix} \phi^\wedge & \rho \\ \mathbf{0}^T & 0 \end{bmatrix} \in \mathbb{R}^{4 \times 4} \quad (3.11)$$

如何计算 $\exp(\phi^\wedge)$ ，这是一个矩阵的指数，在李群和李代数中称为指数映射。如何计算它，这里先将其泰勒展开

$$\exp \mathbf{A} = \sum_{n=0}^{\infty} \frac{1}{n!} \mathbf{A}^n \quad (3.12)$$

但是这个方法没有办法直接计算，并且也不想计算矩阵的无穷次幂，于是推导了一种简便的计算方法，由于 ϕ 是三维向量，我们可以定义它的模长和方向，分别记作 θ 和 α ，于是有 $\phi = \theta\alpha$ ，这里 α 为一个长度为 1 的方向向量，即 $\|\alpha\| = 1$ ，首先它有以下两条性质

$$\alpha^\wedge \alpha^\wedge = \begin{bmatrix} -\alpha_2^2 - \alpha_3^2 & \alpha_1 \alpha_2 & \alpha_1 \alpha_3 \\ \alpha_1 \alpha_2 & -\alpha_1^2 - \alpha_3^2 & \alpha_2 \alpha_3 \\ \alpha_1 \alpha_3 & \alpha_2 \alpha_3 & -\alpha_1^2 - \alpha_2^2 \end{bmatrix} = \mathbf{a}\mathbf{a}^T - \mathbf{I} \quad (3.13)$$

以及

$$\mathbf{a}^\wedge \mathbf{a}^\wedge \mathbf{a}^\wedge = \mathbf{a}^\wedge (\mathbf{a}\mathbf{a}^T - \mathbf{I}) = -\mathbf{a}^\wedge \quad (3.14)$$

这两个式子的提供使得可以处理 α^\wedge 高阶项的方法。

$$\exp(\phi^\wedge) = \exp(\theta \alpha^\wedge) = \sum_{n=0}^{\infty} \frac{1}{n!} (\theta \alpha^\wedge)^n = \cos \theta \mathbf{I} + (1 - \cos \theta) \alpha \alpha^T + \sin \theta \alpha^\wedge \quad (3.15)$$

上式和罗德里格斯公式一模一样，这表明 $\mathfrak{so}(3)$ 实际上就是所谓的旋转向量组成的空间，而指数映射及罗德里格斯公式，通过它们可以把 $\mathfrak{so}(3)$ 中任意一个向量对应到一个位于 $SO(3)$ 中的旋转矩阵，反之，如果定义对数映射，也能把 $SO(3)$ 中的元素对应到 $\mathfrak{so}(3)$ 中

$$\phi = \ln(\mathbf{R})^\vee = \left(\sum_{n=0}^{\infty} \frac{(-1)^n}{n+1} (\mathbf{R} - \mathbf{I})^{n+1} \right)^\vee \quad (3.16)$$

在实际运算中没必要直接用泰勒展开计算对数映射，可以利用迹的性质分别求解转角和转轴，采用这种方法更加省事。后面的 $SE(3)$ 上的对数映射和 $\mathfrak{se}(3)$ 上的指数映射也如此推导不在赘述，直接给出下图

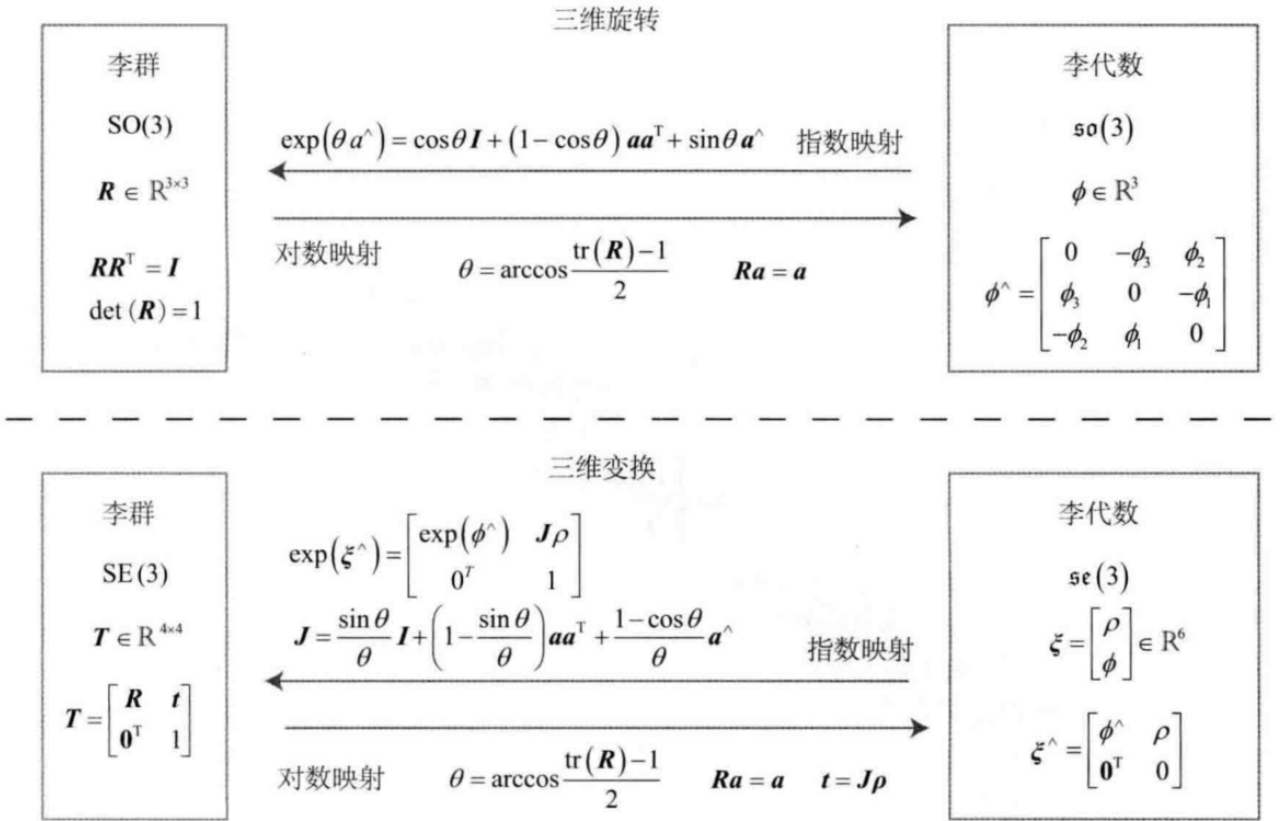


图 1: $SO(3), SE(3), \mathfrak{so}(3), \mathfrak{se}(3)$ 的对应关系

3.3 李代数求导与扰动模型

使用李代数是为了进行优化，在优化过程中不可避免的有求导，那么李代数的求导如何进行？

李代数的指数映射与李群相对应，那么李代数的指数映射是否符合指数运算的基本规则，显然这是不成立的，因为 $\exp \phi^\wedge$ 中是一个矩阵，不满足 $\exp(\phi_1^\wedge) \exp(\phi_2^\wedge) = \exp(\phi_1^\wedge + \phi_2^\wedge)$ ，它满足的是一个 BCH 公式 (Baker-Campbell-Hausdorff)，给出其完整公式的前面几项

$$\ln(\exp(\mathbf{A}) \exp(\mathbf{B})) = \mathbf{A} + \mathbf{B} + \frac{1}{2}[\mathbf{A}, \mathbf{B}] + \frac{1}{12}[\mathbf{A}, [\mathbf{A}, \mathbf{B}]] - \frac{1}{12}[\mathbf{B}, [\mathbf{A}, \mathbf{B}]] + \dots \quad (3.17)$$

其中 $[\cdot]$ 为李括号，当考虑 $SO(3)$ 上的李代数 $\ln(\exp(\phi_1^\wedge) \exp(\phi_2^\wedge))^\vee$ ，当 ϕ_1 或 ϕ_2 为小两时，小量二次以上的项都可以被忽略，此时 BCH 拥有线性近似表达

$$\ln(\exp(\phi_1^\wedge) \exp(\phi_2^\wedge))^\vee \approx \begin{cases} \mathbf{J}_l(\phi_2)^{-1} \phi_1 + \phi_2 & \text{当 } \phi_1 \text{ 为小量} \\ \mathbf{J}_r(\phi_1)^{-1} \phi_2 + \phi_1 & \text{当 } \phi_2 \text{ 为小量} \end{cases} \quad (3.18)$$

以上式的第一个近似为例，当对一个旋转矩阵 \mathbf{R}_2 (李代数为 ϕ_2) 左乘一个微小旋转矩阵 \mathbf{R}_1 (李代数为 ϕ_1) 时，可以近似的看作，在原有李代数 ϕ_2 上加了一项 $\mathbf{J}_l(\phi_2)\phi_1$ ，同理第二个近似描述了右乘一个微小位移的情况。于是在 BCH 近似下，分成了左乘近似和右乘近似两种，以左乘为例

$$\mathbf{J}_l = \mathbf{J} = \frac{\sin \theta}{\theta} \mathbf{I} + (1 - \frac{\sin \theta}{\theta}) \mathbf{a} \mathbf{a}^T + \frac{1 - \cos \theta}{\theta} \hat{\mathbf{a}} \quad (3.19)$$

它的逆为

$$\mathbf{J}_l^{-1} = \frac{\theta}{2} \cot \frac{\theta}{2} \mathbf{I} + (1 - \frac{\theta}{2} \cot \frac{\theta}{2}) \mathbf{a} \mathbf{a}^T - \frac{\theta}{2} \hat{\mathbf{a}} \quad (3.20)$$

而右乘雅可比仅需对自变量取符号即可：

$$\mathbf{J}_r(\phi) = \mathbf{J}_l(-\phi) \quad (3.21)$$

由此对某一旋转 \mathbf{R} 对应的李代数 ϕ 给它左乘一个微小旋转 $\Delta \mathbf{R}$ (对应的李代数为 $\Delta \phi$) 在李群上得到 $\Delta \mathbf{R} \cdot \mathbf{R}$ ，而在李代数上，根据 BCH 相似，为 $\mathbf{J}_l^{-1}(\phi) \Delta \phi + \phi$ ，合并起来可写为

$$\exp(\Delta \phi) \exp(\phi) = \exp((\phi + \mathbf{J}_l^{-1}(\phi) \Delta \phi)) \quad (3.22)$$

反之，在李代数上进行加法，让一个 ϕ 加上 $\Delta \phi$ ，那么可以近似为李群上带所有雅可比的乘法：

$$\exp((\phi + \Delta \phi)) = \exp((\mathbf{J}_l \Delta \phi)) \exp(\phi) = \exp(\phi) \exp((\mathbf{J}_r \Delta \phi)) \quad (3.23)$$

同样对于 $SE(3)$ 也有类似的 BCH 近似

4 畸变和相机模型

由透镜形状引起的畸变叫做径向畸变，会将现实中的一条直线变为一条曲线，它主要分为两大类**桶形畸变**和**枕形畸变**，桶形畸变图像放大率随着光轴之间的距离增加而减小，而枕形畸变恰好相反。在这两种畸变中穿过图像中心和光轴有交点的直线还能保持形状不变。在相机的组装过程中不能使透镜和成像平面严格平行会引入**切向畸变**

归一化平面上任意一点 \mathbf{p} 坐标为 $[x, y]^T$ ，也可用极坐标形式 $[r, \theta]^T$ ，其中 r 表示点 \mathbf{p} 与坐标系原点之间的距离， θ 表示与水平轴的夹角。径向畸变可以堪称坐标点沿着长度方向发生了变化，切向畸变可以看成坐标点沿着切线方向发生了变化，对于径向畸变

$$\begin{aligned} x_{distorted} &= x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ y_{distorted} &= y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \end{aligned} \quad (4.1)$$

对于切向畸变可以用另外两个参数来表示

$$\begin{aligned} x_{distorted} &= x + 2p_1 xy + p_2(r^2 + 2x^2) \\ y_{distorted} &= y + p_1(r^2 + 2y^2) + 2p_2 xy \end{aligned} \quad (4.2)$$

联合上面的公式就可以通过五个畸变系数找到这个点在像素平面中正确的位置：

$$\begin{cases} x_{distorted} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + x + 2p_1 xy + p_2(r^2 + 2x^2) \\ y_{distorted} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + y + p_1(r^2 + 2y^2) + 2p_2 xy \end{cases} \quad (4.3)$$

将畸变后的点通过内参数矩阵投影到像素平面，得到该点在图像上的正确位置：

$$\begin{cases} u = f_x x_{distorted} + c_x \\ v = f_y y_{distorted} + c_y \end{cases} \quad (4.4)$$

5 极几何

三角化，已知两幅图片中对应真实世界中点 P 的两个像素坐标 p 和 p' ，两个相机的内参 K 和 K' 以及两个相机之间的位姿变换矩阵 \mathbf{R} 和 \mathbf{T} ，求解 P 的三维坐标。即寻找 P 满足

$$\begin{cases} p = MP = K[I \ 0]P \\ p' = M'P = K'[R \ T]P \end{cases} \quad (5.1)$$

但现实是对于相机内参 K 和 K' 、相机位姿旋转矩阵 R 、平移向量 T 往往不知道，仅清楚两张图片之间的对应点 p 和 p' 。

于是在多视图几何中存在以下几个关键问题

1. 摄像几何：从一张或者多张图像中求解摄像机的内外参数
2. 场景几何：通过两张至多张图片寻找 3D 场景坐标
3. 对应关系：已知一个图像中的 p 点，如何在另一个图像中找到对应的 p' 点

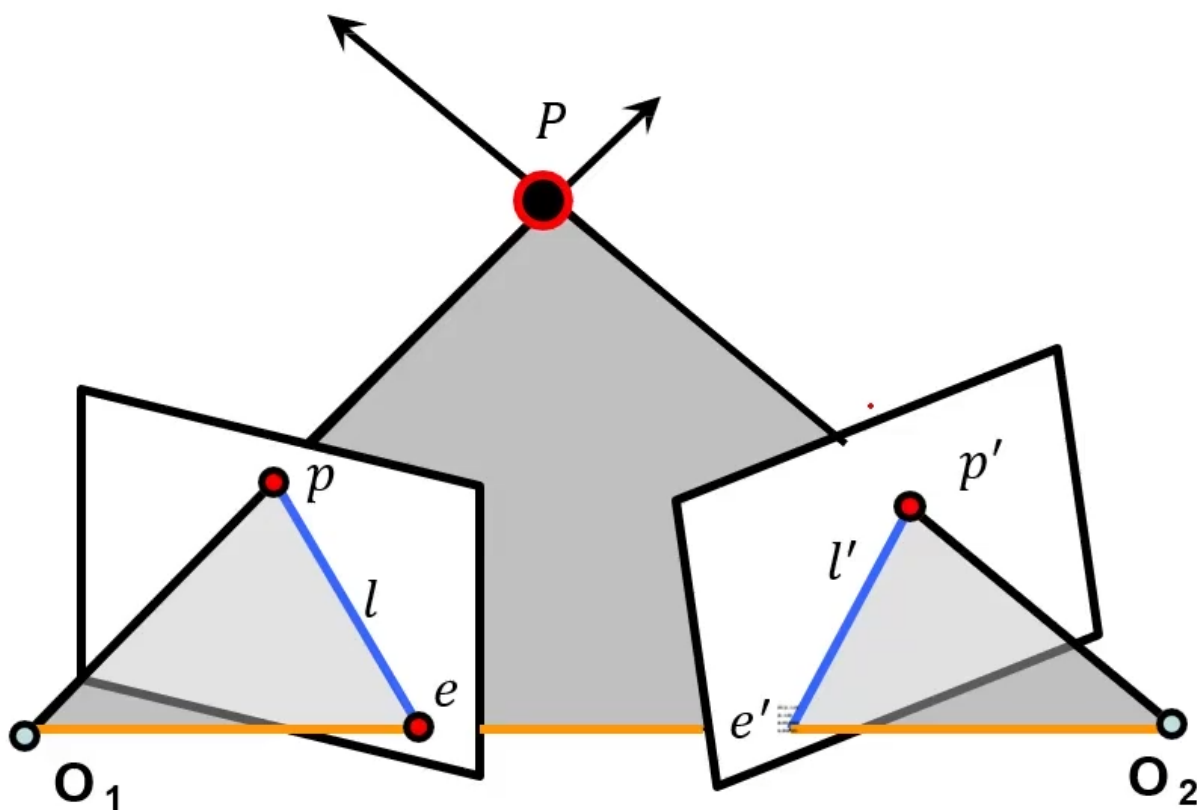


图 2: 极几何

如图：

1. 极平面：过点 P ， O_1 与 O_2 的平面
2. 基线： O_1 与 O_2 的连线
3. 极线：极平面与成像平面的交线
4. 极点：基线与成像平面的交点

极几何的特例：平行视图（常见双目立体、VR）：

1. 两个图像平面平行
2. 基线平行于图像平面，极点 e 和 e' 位于无穷远处
3. 极线平行于图像坐标系的 u 轴

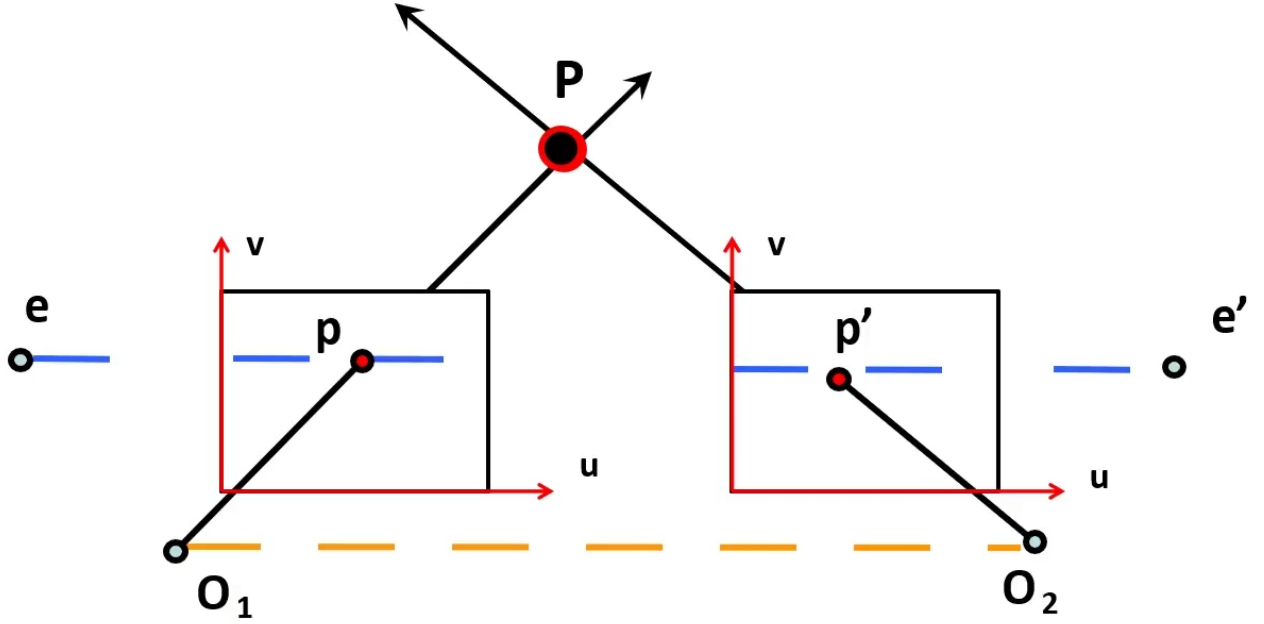


图 3: 极几何的特例: 平行视图

5.1 本质矩阵

对于规范化相机, 图像 I 上点 p 像素坐标为 (u, v) , 图像 I' 上的点 p' 坐标为 (u', v') , $K = K' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, 因为为规范化相机所以点 p 在 O_1 坐标系下的非齐次坐标为 $(u, v, 1)$, 同理空间点 p' 在 O_2 下的非齐次坐标为 $(u', v', 1)$ 知道 O_1 到 O_2 的旋转矩阵 R 和平移向量 T , 于是有

$$\begin{aligned} p' \text{ 在 } O_1 \text{ 的坐标} &= R^T p' - R^T T \\ O_2 \text{ 在 } O_1 \text{ 的坐标} &= -R^T T \end{aligned} \quad (5.2)$$

于是可以求得一个垂直于极平面的向量

$$R^T T \times (R^T p' - R^T T) = R^T T \times R^T p' \quad (5.3)$$

因为 p 点在极平面上, 所以有

$$[R^T T \times R^T p']^T \cdot p = 0 \quad (5.4)$$

将上面的式子推导可以得出

$$p'^T [T \times R] p = 0 \quad (5.5)$$

于是点 p 和点 p' 存在一个极几何约束——本质矩阵

$$E = T \times R \Rightarrow E = \text{本质矩阵} \quad (5.6)$$

$$p'^T E p = 0 \quad (5.7)$$

其中本质矩阵有以下性质:

1. p 对应的极线是 $l' (l' = E p)$
2. p' 对应的极线是 $l (l = E^T p')$
3. $E e = 0$ 与 $e'^T E = 0$
4. E 是奇异的 (秩 2)
5. E 是 5 个自由度 (三个旋转 + 三个平移, $\det(E)$ 去掉一个自由度)

5.2 基础矩阵

这个为正常透视相机下的点 p 与点 p' 的对应关系，此时推导类比规范化相机下的推导，先将两张图像上点的坐标转换为规范化相机下的坐标，有：

$$\begin{cases} p_c = K^{-1}p \\ p'_c = K'^{-1}p' \end{cases} \quad (5.8)$$

于是有： $p_c'^T E p_c = (K'^{-1}p')^T \cdot [T_\times] R K^{-1}p = p'^T K'^{-T} [T_\times] R K^{-1}p = 0$ 即可求得基础矩阵

$$F = K'^{-T} [T_\times] R K^{-1} \rightarrow p'^T F p = 0 \quad (5.9)$$

同样它也有上述的几个性质，不同的是它有 7 个自由度，因为尺度无法确定。

常见的基础矩阵估计方法八点法，归一化八点法

5.3 单应性矩阵

空间平面在两个摄像机下的投影几何 如图单应性矩阵推导已知第一个摄像机的内参矩阵 K ，第二个内参矩

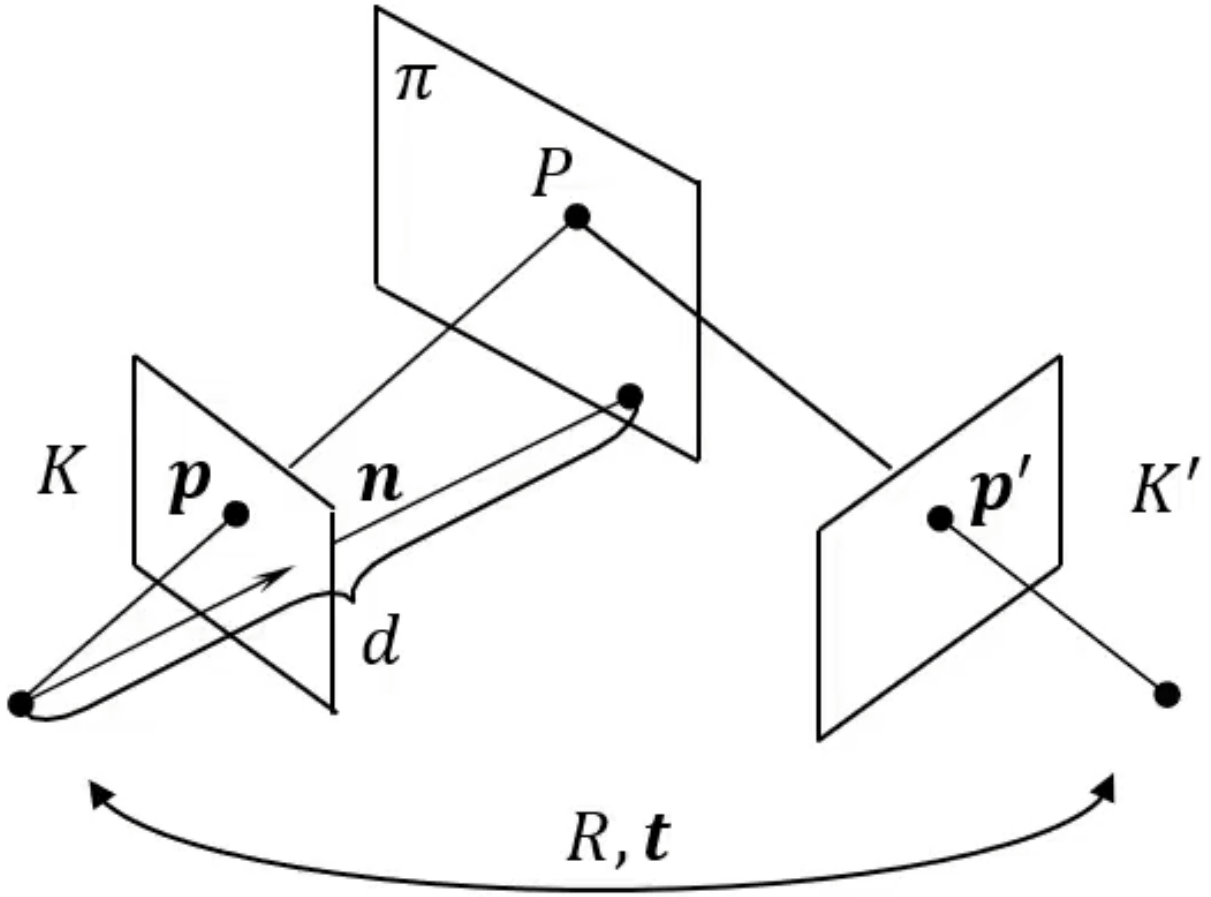


图 4: 单应性矩阵推导

阵 K' ，第二个摄像机相对于第一个摄像机的位置 (R, t) ， n 为平面 π 在第一个摄像机坐标系下的单位法向量， d 为坐标原点到平面 π 的距离

平面 π 的方程为 $n^T \tilde{P} = d$ ， \tilde{P} 表示点 P 的欧式坐标，由此直接给出平面 π 的单应性矩阵 $H = K'(R + tn_d^T)K^{-1}$ ，其中， $n_d = \frac{n}{d}$

约束关系

基础矩阵建立点和极线的对应关系

单应矩阵建立点和点的对应

6 概率运动模型

6.1 速度运动模型

理想模型，不考虑速度的误差机器人以速度 u_k 在很短时间 Δt 内做匀速运动，位姿从 x_{k-1} 转移到 x_k 。

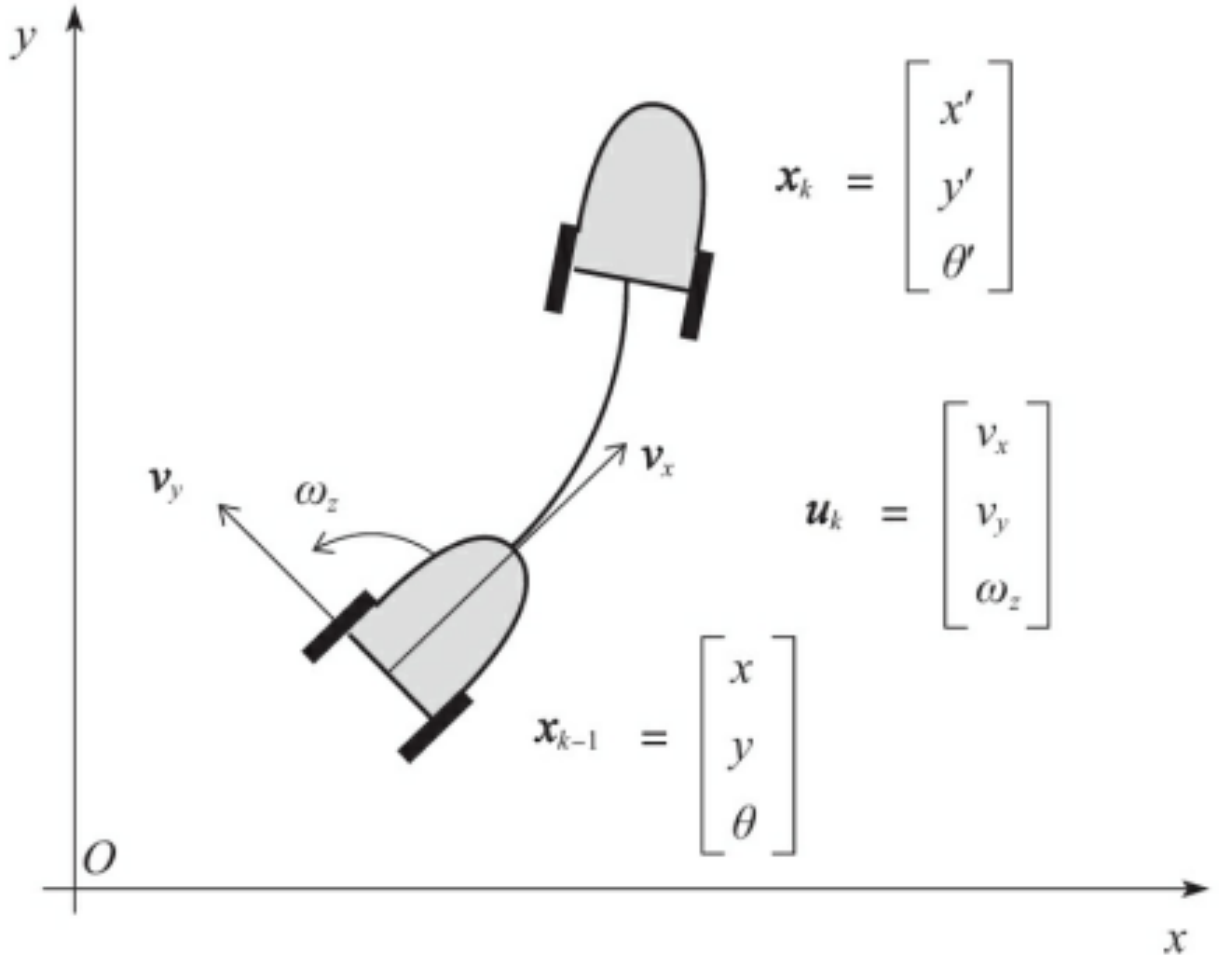


图 5: 速度运动模型

$$\mathbf{x}_k = \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = g(\mathbf{x}_{k-1}, \mathbf{u}_k) = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega_z \end{bmatrix} \Delta t \quad (6.1)$$

考虑速度的误差，在实际中，需要考虑 u_k x_{k-1} x_k 的误差，这里简单假设为服从高斯分布

$$\begin{aligned} \mathbf{u} &\sim N(\bar{\mathbf{u}}_k, \Sigma_{\mathbf{u}_k}) \\ \mathbf{x}_{k-1} &\sim N(\bar{\mathbf{x}}_{k-1}, \Sigma_{\mathbf{x}}) \\ \mathbf{x}_k &\sim N(\bar{\mathbf{x}}_k, \Sigma_{\mathbf{x}_k}) \end{aligned} \quad (6.2)$$

这样， u_k x_{k-1} x_k 都变成了多元随机变量，随机变量的均值就是其估计值，用协方差矩阵描述其不确定性大小。 u_k 的误差主要是由 3 个速度分量 v_x v_y ω_z 的误差决定的，所以 u_k 的协方差矩阵如下所示

$$\Sigma_{\mathbf{u}_k} = \begin{bmatrix} \sigma_{v_x}^2 & 0 & 0 \\ 0 & \sigma_{v_y}^2 & 0 \\ 0 & 0 & \sigma_{\omega_z}^2 \end{bmatrix} \quad (6.3)$$

已知 u_k 和 x_{k-1} 的分布特性，利用运动的状态转移公式就可以推导出 x_k 的分布特性，即状态转移概率 $\mathbf{P}(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k)$ 。其中一个最简单的方法是先将状态转移函数 g 进行一阶泰勒级数展开来实现线性化，然后直

接通过线性计算就能够得到 x_k 的协方差矩阵

$$\Sigma_{x_k} = \begin{bmatrix} \frac{\partial g}{\partial \mathbf{x}_{k-1}} & \frac{\partial g}{\partial \mathbf{u}_k} \end{bmatrix} \begin{bmatrix} \Sigma_{x_{k-1}} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \Sigma_{\mathbf{u}_k} \end{bmatrix} \begin{bmatrix} \frac{\partial g}{\partial \mathbf{x}_{k-1}} & \frac{\partial g}{\partial \mathbf{u}_k} \end{bmatrix}^T \quad (6.4)$$

其中, $\begin{bmatrix} \frac{\partial g}{\partial \mathbf{x}_{k-1}} & \frac{\partial g}{\partial \mathbf{u}_k} \end{bmatrix}$ 为函数 g 的雅可比矩阵。上面的推导都是基于两大假设, 一是运动时间 Δt 很小, 二是 u_k x_{k-1} x_k 的误差都服从高斯分布。

6.2 里程计运动模型

上面的速度运动模型中有太强的假设, 准确性很难保证, 实际中机器人一般配备有里程计等感知传感器来用于反馈运动。通过前后两个里程计的差值, 可以求出里程计的增量值 u_k , 即所谓的机器人状态转移量。

$$\mathbf{u}_k = \text{odom}_k - \text{odom}_{k-1} = \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{bmatrix} \quad (6.5)$$

与讨论速度运动模型一样, 先不讨论有误差的模型, 直接推导状态转移概率 $\mathbf{P}(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k)$ 的数学形式。此时

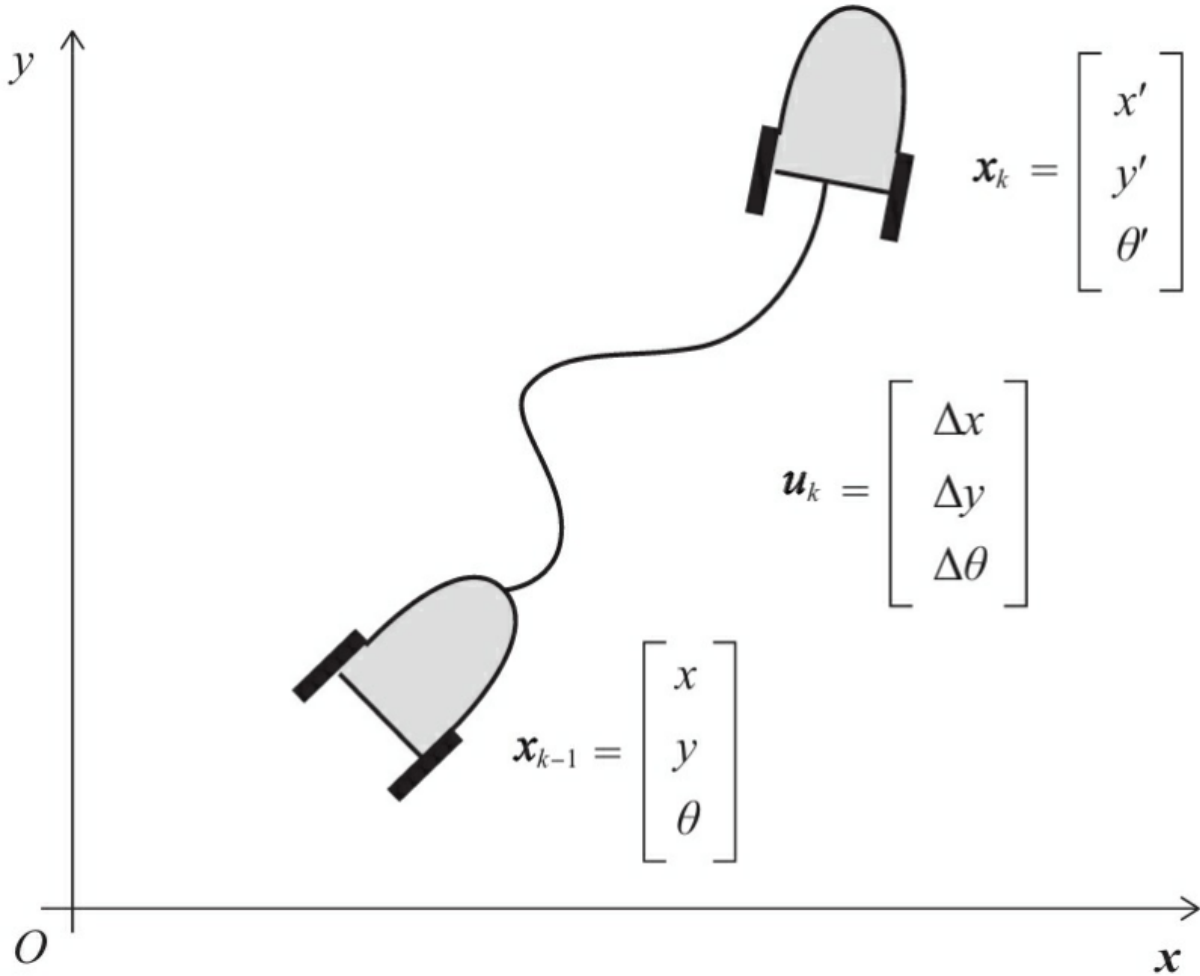


图 6: 里程计运动模型

可直接求出 x_k

$$\mathbf{x}_k = \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = g(\mathbf{x}_{k-1}, \mathbf{u}_k) = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{bmatrix} \quad (6.6)$$

考虑里程计误差的情况下，这里将 u_k 通用的表示在三维空间运动，但是机器人底盘被限制在二维平面运动，所以其中 z 轴的偏移量、 x 轴的旋转量和 y 轴的旋转量均为 0。

$$\Sigma_{u_k} = \begin{bmatrix} \sigma_{\Delta x}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{\Delta y}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{\Delta \theta}^2 \end{bmatrix} \quad (6.7)$$

轮式里程计是存在累计误差的，即机器人地盘运动得越远累积误差越大。可以用运动距离 $\sqrt{(\Delta x)^2 + (\Delta y)^2}$ 和旋转量 $|\Delta \theta|$ 来量化 u_k 的误差。

$$\begin{aligned} \sigma_{\Delta x}^2 &= \xi_{\Delta xy} + \alpha_1 \sqrt{(\Delta x)^2 + (\Delta y)^2} + \alpha_2 |\Delta \theta| \\ \sigma_{\Delta y}^2 &= \xi_{\Delta xy} + \alpha_1 \sqrt{(\Delta x)^2 + (\Delta y)^2} + \alpha_2 |\Delta \theta| \\ \sigma_{\Delta \theta}^2 &= \xi_{\Delta \theta} + \alpha_3 \sqrt{(\Delta x)^2 + (\Delta y)^2} + \alpha_4 |\Delta \theta| \end{aligned} \quad (6.8)$$

7 概率观测模型

这先介绍激光模型

7.1 波束模型

每个激光束可以得到一个距离值 z_k^i ，对于 n 个测量值 $\mathbf{z}_k = \{z_k^1, z_k^2, \dots, z_k^n\}$ ，那么观测 z_k 的不确定性由每个单独光束测量值的不确定性相乘得到

$$P(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m}) = \prod_{i=1}^n P(z_k^i | \mathbf{x}_k, \mathbf{m}) \quad (7.1)$$

根据实验和经验所得，单个激光束测距值误差主要分为四类：读数误差、动态干扰、测量失效和随机误差。

读数误差，传感器由于构造、计算精度、分辨率等因素给测量带来的误差，可直接用高斯分布来表示，其中 η_{hit} 是归一化因子， z_{max} 是激光束的最大测量值。

$$P(z_k^i | \mathbf{x}_k, \mathbf{m}) = \begin{cases} \eta_{hit} N(\bar{z}_k^i, \sigma_{hit}^2), 0 \leq z_k^i \leq z_{max} \\ 0, \text{其它} \end{cases} \quad (7.2)$$

动态干扰误差，环境中动态障碍物挡住激光束，导致原本对应在地图中的障碍物的测量距离变短，采用指数分布来表示，其中 η_{short} 是归一化因子， λ_{short} 是指数分布参数

$$P_{short}(z_k^i | \mathbf{x}_k, \mathbf{m}) = \begin{cases} \eta_{short} \lambda_{short} \exp^{-\lambda_{short} z_k^i}, 0 \leq z_k^i \leq \bar{z}_k^i \\ 0, \text{其它} \end{cases} \quad (7.3)$$

环境中的黑色吸光、透明、镜面反反射等障碍物会导致测量失效，使得激光测距值超量程或者无穷远，这用最大量程 z_{max} 附近的窄矩形分布来描述

$$P_{max}(z_k^i | \mathbf{x}_k, \mathbf{m}) = \begin{cases} \frac{1}{\varepsilon}, z_{max} - \varepsilon \leq z_k^i \leq z_{max} \\ 0, \text{其它} \end{cases} \quad (7.4)$$

最后哪些不能明确建模的误差，用一个均匀分布的随机误差来统一描述，均匀分布的范围是整个量程区间

$$P_{rand}(z_k^i | \mathbf{x}_k, \mathbf{m}) = \begin{cases} \frac{1}{z_{max}}, 0 \leq z_k^i \leq z_{max} \\ 0, \text{其它} \end{cases} \quad (7.5)$$

将上述四种误差加权平均后就是光束模型，其中 $z_{hit}, z_{short}, z_{max}, z_{rand}$ 为加权系数，它们的和为 1，加权后的概率分布为

$$P(z_k^i | \mathbf{x}_k, \mathbf{m}) = z_{hit} \cdot P_{hit} + z_{short} \cdot P_{short} + z_{max} \cdot P_{max} + z_{rand} \cdot P_{rand} \quad (7.6)$$

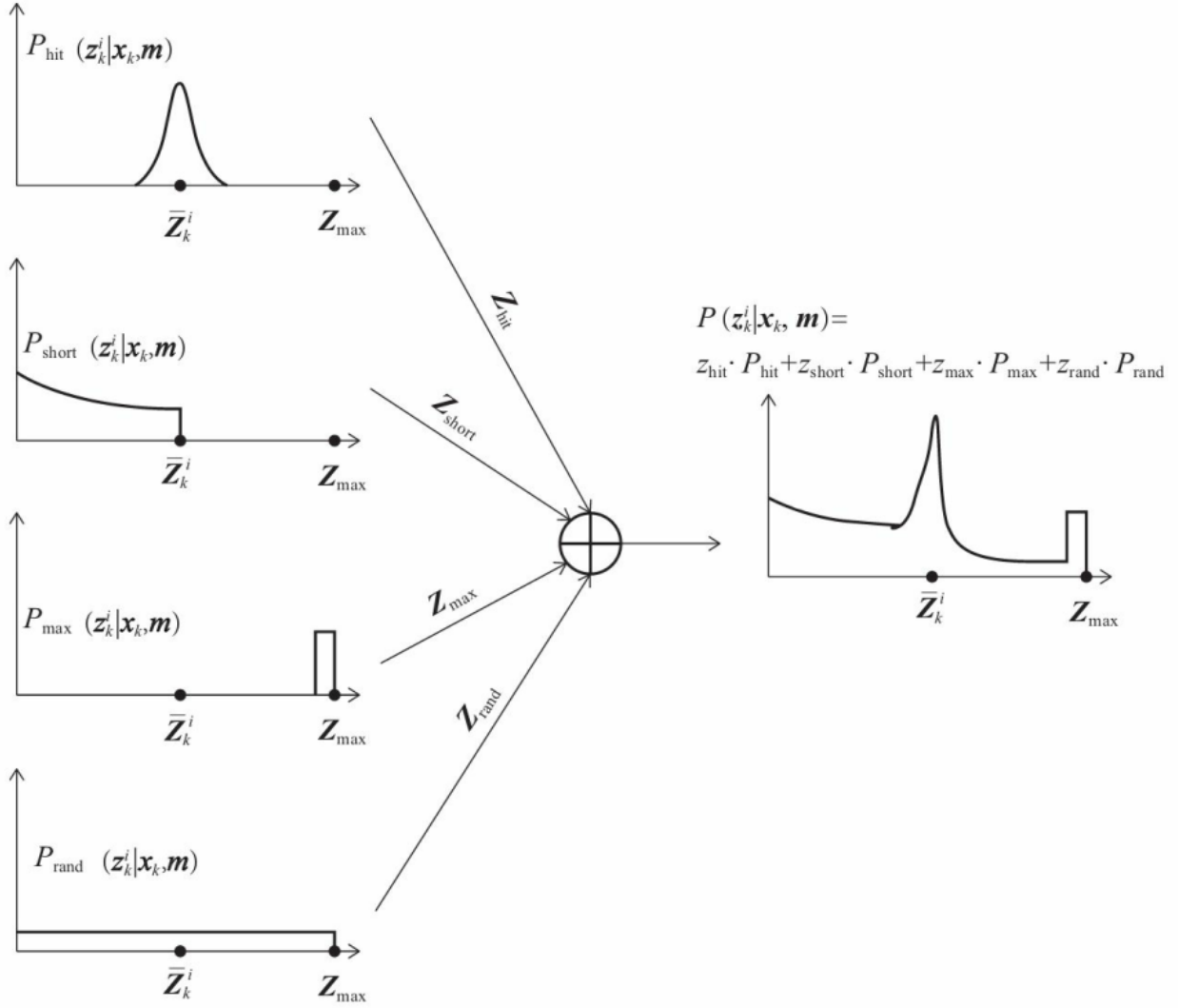


图 7: 波束模型的概率描述

7.2 似然域模型

它的思路是利用测量值对应的机器人位姿，将测量值 z_k^i 透射到已有的地图上，将超过量程的测量值直接丢弃，然后用似然域描述测量值分布。所谓似然域就是激光束上障碍物出现的概率大小由其附近的地图点远近决定。

7.3 概率图模型

对于实际问题中，比如 slam 系统中对一个随机变量的估计往往涉及多个变量之间的关系，比如对于随机变量 (A, B, C, D) ，有

$$P(A) = \sum_D \sum_C \sum B P(A, B, C, D) \quad (7.7)$$

$$\begin{aligned} P(A, B, C, D) &= P(A, B, C)P(D|A, B, C) \\ &= P(A, B)P(C|A, B)P(D|A, B, C) \\ &= P(A)P(B|A)P(C|A, B)P(D|A, B, C) \end{aligned} \quad (7.8)$$

上面的式子一个叫做变量 A 的边缘分布概率，一个叫做随机变量集合 $\{A, B, C, D\}$ 的联合分布。这种求解和表示往往比较复杂，借助图论的方式结构化数据可以让随机变量之间的依赖关系变得更为直观。

贝叶斯网络：有向图，各个节点都对应一个随机变量事件，网络结构描述各个事件之间的因果关系，就构成了静态贝叶斯网络。当随机变量由它的时序状态构成，每个状态产生对应的观测，两个状态节点之间的方向表示状态更新方向，状态节点和观测节点直接的箭头表示观测过程，这样的网络为动态贝叶斯网络。

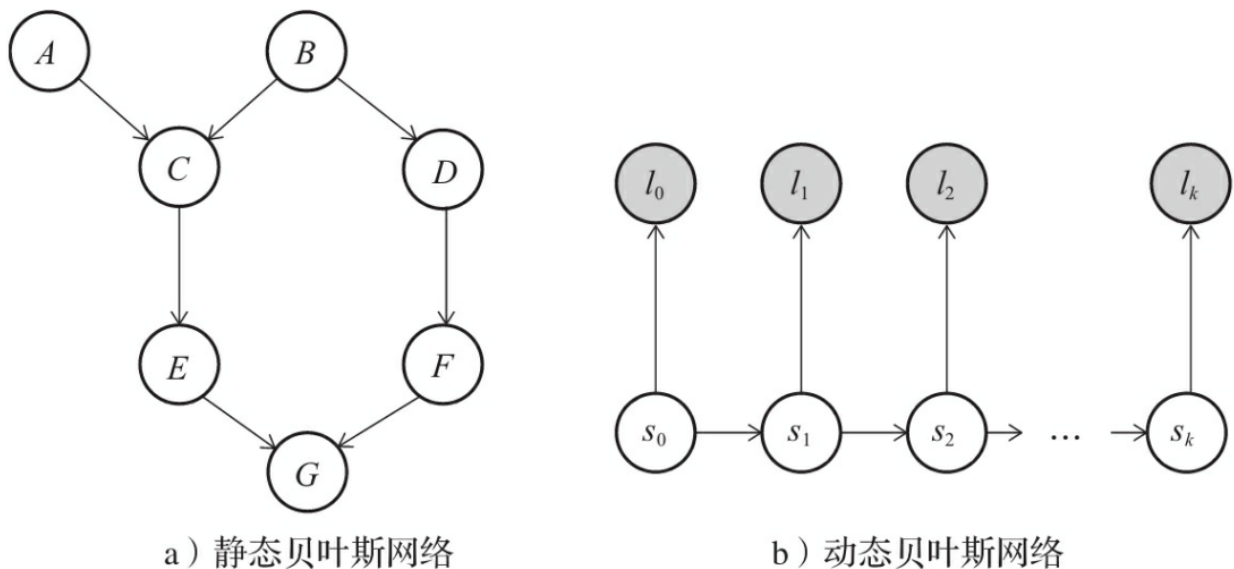


图 8: 贝叶斯网络

马尔可夫网络：每个节点表示一个随机变量，连接两个节点之间是没有方向的，边代表两个随机变量之间的相关性。马尔可夫网络参数化过程，其随机变量之间的联合概率可分解为团 Q_i 的势能函数 $\Psi_{Q_i}(Q_i)$ 的乘积，团是一个节点的集合，团中的节点都必须两两之间有边连接。以团中随机变量作为自变量构成团的势能函数，团的势能函数是一个抽象函数，在实际应用中被具体化为特定形式。**往团中加入其它任何节点，将不再成团，这样的团就是最大团**

将图结构中各个随机变量记为随机变量 $\{X_1, X_2, \dots, X_n\}$ ，图结构划分出来所有的最大团记为 $Q = \{Q_1, Q_2, \dots, Q_m\}$ ，那么它们的联合概率分布就可以记作

$$P(X_1, X_2, \dots, X_n) = \frac{1}{Z} \prod_{i=1}^m \Psi_{Q_i}(Q_i) \quad (7.9)$$

7.4 概率图模型在 slam 中的应用

首先用动态贝叶斯网络表示 slam 问题

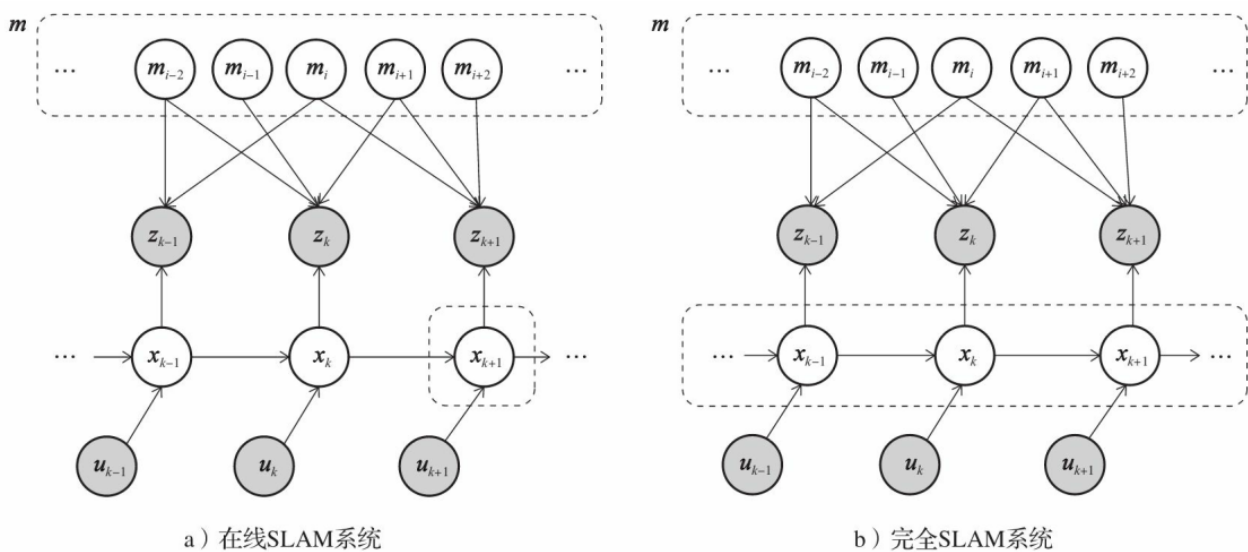


图 9: 用贝叶斯网络表示 slam

在线 slam 系统只对环境路标特征和机器人当前位姿状态进行估计 $P(x_k, m | Z_{0:k}, U_{0:k}, x_0)$ ，完全 slam 系统对环境路标特征和机器人当前及过去所有位姿状态 $x = \{x_1, x_2, \dots, x_k\}$ 进行估计 $P(X_{0:k}, m | Z_{0:k}, U_{0:k}, x_0)$

那么如何将一个 slam 问题表示为因子图的方式呢，以一个简单的 slam 系统为例

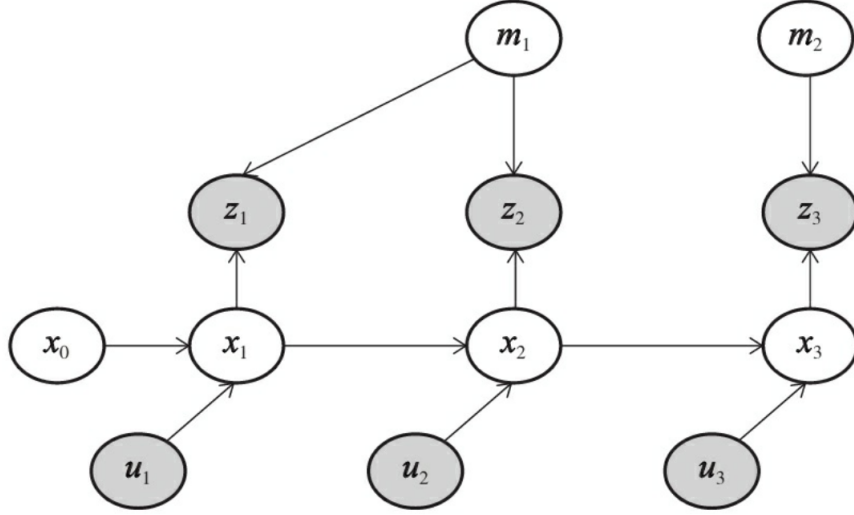


图 10: 小规模 slam 示例

根据前面的参数化方法，可以表示此个联合概率分布为

$$\begin{aligned}
 P(X, m, Z, U) = & P(m_1)P(m_2)P(u_0)P(u_1)(u_2) \quad (\text{先验, 可忽略}) \times \\
 & P(x_0)P(x_1|x_0, u_1)P(x_2|x_1, u_2)P(x_3|x_2, u_3) \quad \text{运动} \times \\
 & P(z_1|x_1, m_1)P(z_2|x_2, m_1)P(z_3|x_3, m_2) \quad \text{观测}
 \end{aligned} \tag{7.10}$$

看到联合概率分布由先验、运动和观测组成，其中先验一般为常量可以忽略，那么概率图模型就包含了 SLAM 问题所有的运动和观测信息。随机变量集 $V = \{Z, U\}$ 表示直接测量信息， $S = \{X, m\}$ 表示不可直接观测信息，也就是待估计量，然后将不关心的量隐藏到因子函数约束中，直接以上式的乘积项作为因子图的因子项，那么就可以得到如下的因子图表示：

$$P(\mathbf{S}|\mathbf{V}) \propto \psi(\mathbf{S}) = \prod_i \psi_i(\mathbf{X}, \mathbf{m}, \mathbf{Z}, \mathbf{U}) \tag{7.11}$$

对应的最大后验估计可以用因子图方式写为

$$\mathbf{S}_{MAP} = \arg \max_s \psi(\mathbf{S}) \tag{7.12}$$

给定运动方程和观测方程：

$$\mathbf{x}_k = g(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{R}_k \tag{7.13}$$

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{m}_j) + \mathbf{Q}_k \tag{7.14}$$

假设运动噪声 \mathbf{R}_k 和观测噪声 \mathbf{Q}_k 都是 0 均值高斯噪声，噪声协方差矩阵分别用 Σ_{R_k} 和 Σ_{Q_k} 表示，用运动误差和观测误差分别来构造这两种因子项，运动约束的因子项和观测约束的因子项如下

$$\Psi_{i1}(\mathbf{x}_k, \mathbf{x}_{k-1}) \propto e^{-\frac{1}{2}(\mathbf{x}_k - g(\mathbf{x}_{k-1}, \mathbf{u}_k))^T \Sigma_{R_k}^{-1} (\mathbf{x}_k - g(\mathbf{x}_{k-1}, \mathbf{u}_k))} \tag{7.15}$$

$$\Psi_{i2}(\mathbf{x}_k, \mathbf{m}_j) \propto e^{-\frac{1}{2}(\mathbf{z}_k - h(\mathbf{x}_k, \mathbf{m}_j))^T \Sigma_{Q_k}^{-1} (\mathbf{z}_k - h(\mathbf{x}_k, \mathbf{m}_j))} \tag{7.16}$$

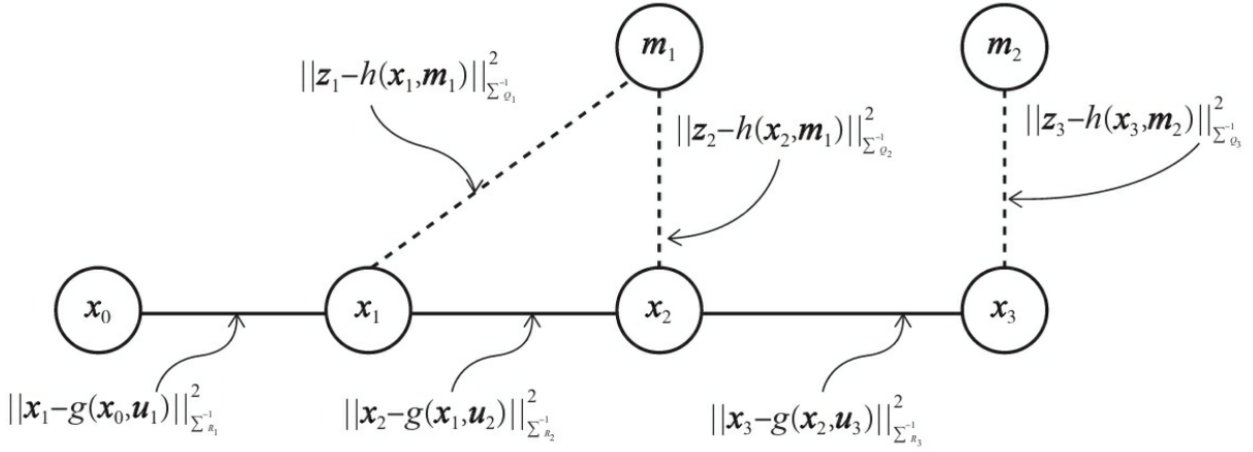


图 11: 最小二乘对应的因子图

8 梯度、雅可比矩阵、海森矩阵

对于两个自变量为多维，因变量为一维和多维的函数表达如下所示：

$$f(\mathbf{x}) = f(x_1, x_2, \dots, x_n) \quad (8.1)$$

$$f(\mathbf{x}) = \begin{bmatrix} f_1(x_1, x_2, \dots, x_n) \\ f_2(x_1, x_2, \dots, x_n) \\ \dots \\ f_m(x_1, x_2, \dots, x_n) \end{bmatrix} \quad (8.2)$$

对于多维自变量和一维因变量的函数式，其一阶导数就是梯度，是一个与自变量 \mathbf{x} 同维度的向量

$$\mathbf{grad} = \nabla f(\mathbf{x}) = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right] \quad (8.3)$$

函数的二阶导数就是海森矩阵，对一个 n 维自变量一维因变量的函数求导，一阶导数就是梯度（一个 $n \times 1$ 的向量），对梯度再求一阶导数就是海森矩阵（ $n \times n$ 的矩阵）

$$\begin{aligned} \mathbf{Hessian} &= \nabla^2 f(\mathbf{x}) = \nabla(\nabla f(\mathbf{x})) = \nabla(\mathbf{grad}) = \nabla \left(\begin{bmatrix} f_1(x_1, x_2, \dots, x_n) \\ f_2(x_1, x_2, \dots, x_n) \\ \dots \\ f_m(x_1, x_2, \dots, x_n) \end{bmatrix}^T \right) \\ &= \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix} \end{aligned} \quad (8.4)$$

那么多维自变量与因变量的函数式中的一阶导数就是雅可比矩阵，雅可比矩阵的每一行就是多维因变量中每一维梯度的转置，意思就是单独将每一维因变量看成一位因变量和多个自变量的梯度，然后拼接起来

$$\mathbf{Jacobian} = \nabla f = \begin{bmatrix} \mathbf{grad}_1^T \\ \mathbf{grad}_2^T \\ \dots \\ \mathbf{grad}_m^T \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \dots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}_{m \times n} \quad (8.5)$$

二阶导也可求，结果是由多个海森矩阵组成的一个大矩阵。

8.1 高斯-牛顿算法

高斯牛顿法是牛顿法在求解非线性最小二乘问题的特例，所以给出如下的非线性最小二乘代价函数

$$\Psi(\mathbf{x}) = \sum_i \| \mathbf{y}_i - f(\mathbf{x}_i) \|^2 = \sum_{i=1}^m \| e_i(\mathbf{x}) \|^2 = \| e(\mathbf{x}) \|^2 = e(\mathbf{x})^T \cdot e(\mathbf{x}) \quad (8.6)$$

其中，误差函数 $e(\mathbf{x})$ 是多维自变量和多维因变量的函数，因变量 $e(\mathbf{x}) = [e_1(\mathbf{x}), \dots, e_m(\mathbf{x})]^T$ 中的每个元素代表非线性最小二乘问题中的一个约束 $(\Sigma_i^{-1/2}) \cdot (y - f(x_i))$ ，自变量 \mathbf{x} 中每个元素代表一个待估计状态量。对上式中 $\Psi(\mathbf{x})$ 进行二阶泰勒展开需要求出其一阶导数和二阶导数，给出 $\Psi(\mathbf{x})$ 的一阶导数就是梯度，给出梯度向量中的第 j 行元素表达式

$$\frac{\partial \Psi}{\partial \mathbf{x}_j} = \sum_{i=1}^m 2 \cdot e_i(\mathbf{x}) \cdot \frac{\partial e_i(\mathbf{x})}{\partial \mathbf{x}_j} \quad (8.7)$$

可以观察得到，其中 $\frac{\partial e_i(\mathbf{x})}{\partial \mathbf{x}_j}$ 代表雅可比矩阵中的元素，那么 $\Psi(\mathbf{x})$ 就可以用 $e(\mathbf{x})$ 函数的雅可比矩阵表示

$$\underbrace{\nabla \Psi}_{grad_{\Psi}} = 2 \cdot \mathbf{Jacobian}_e^T \cdot e(\mathbf{x}) \quad (8.8)$$

而 $\Psi(\mathbf{x})$ 的二阶导数就是海森矩阵，给出海森矩阵中第 k 行第 j 列元素的表达式

$$\frac{\partial^2 \Psi}{\partial \mathbf{x}_k \partial \mathbf{x}_j} = \frac{\partial}{\partial \mathbf{x}_k} \left(\frac{\partial \Psi}{\partial \mathbf{x}_j} \right) = \frac{\partial}{\partial \mathbf{x}_k} \left(\sum_{i=1}^m 2 \cdot e_i(\mathbf{x}) \cdot \frac{\partial e_i(\mathbf{x})}{\partial \mathbf{x}_j} \right) = 2 \sum_{i=1}^m \left(\frac{\partial e_i(\mathbf{x})}{\partial \mathbf{x}_k} \cdot \frac{\partial e_i(\mathbf{x})}{\partial \mathbf{x}_j} + \underbrace{e_i(\mathbf{x}) \cdot \frac{\partial^2 e_i(\mathbf{x})}{\partial \mathbf{x}_k \partial \mathbf{x}_j}}_{\text{高阶项 } S_{k,j}} \right) \quad (8.9)$$

通过观察可知 $\frac{\partial e_i(\mathbf{x})}{\partial \mathbf{x}_k}$ 和 $\frac{\partial e_i(\mathbf{x})}{\partial \mathbf{x}_j}$ 都代表雅可比矩阵中的元素，另外一部分是高阶项 $S_{k,j}$ ，当迭代点离目标点比较近时，误差 $e_i(\mathbf{x})$ 和二阶导数项都比较小，因此可以忽略，那么 $\Psi(\mathbf{x})$ 的二阶导数也可以用 $e(\mathbf{x})$ 函数的雅可比矩阵表示

$$\nabla \Psi(\mathbf{x}) \approx 2 \cdot \mathbf{Jacobian}_e^T \cdot \mathbf{Jacobian}_e \quad (8.10)$$

那么对于 $\Psi(\mathbf{x})$ 的二阶泰勒展开求极值有

$$\nabla \Psi(\mathbf{x}) \approx \text{二阶泰勒展开求梯度} \Leftrightarrow 2 \cdot \mathbf{Jacobian}_e^T \cdot e(\mathbf{x}^{(k)}) + 2 \cdot \mathbf{Jacobian}_e^T \cdot \mathbf{Jacobian}_e \cdot (\mathbf{x} - \mathbf{x}^{(k)}) = 0 \quad (8.11)$$

令 $\Delta \mathbf{x} = \mathbf{x} - \mathbf{x}^{(k)}$ ，那么： $\mathbf{J}_e^T \mathbf{J}_e \cdot \Delta \mathbf{x} = -\mathbf{J}_e^T \cdot \mathbf{e}$ ，那么求解这个方程就可以解出迭代更新量 $\Delta \mathbf{x}$ ，下一步的迭代点就可以用下式计算

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - (\mathbf{J}_{e(\mathbf{x}^{(k)})}^T \mathbf{J}_{e(\mathbf{x}^{(k)})})^{-1} \cdot \mathbf{J}_{e(\mathbf{x}^{(k)})}^T \cdot e(\mathbf{x}^{(k)}) \quad (8.12)$$

9 ORB 特征

ORB 特征由关键点和特征描述子构成，关键点为“Oriented FAST”，是一种改进的 FAST 角点，描述子称为 BRIEF，提取 FAST 角点的步骤为

1. 在图像中选取像素 p ，假设它的亮度为 I_p
2. 设置一个阈值 T （比如 I_p 的百分之二十）
3. 以像素 p 为中心选取半径为 3 的圆上的 16 个像素点
4. 假如选取的圆上有连续 N 个点的亮度大于 $I_p + T$ 或者小于 $I_p - T$ ，那么像素 p 被认为是特征点，(N 的不同对应不同的算法)
5. 循环以上四个步骤，对每个像素执行相同操作

为了更高效的计算，常常有预测试项和非极大值抑制等方法来改良角点提取的算法，除此之外为了让角点具有尺度和方向特性，通过构建图像金字塔和灰度质心法来增加尺度和方向信息。

BRIEF 描述子是一种二进制描述子，其描述向量由 0 和 1 构成

10 三角测量

前面用对极几何的方式估计了相机运动，在得到运动之后，需要用相机的运动估计特征点的空间位置，在单目 SLAM 中，需要通过三角测量 (三角化) 的方法估计地图点的深度

设 $\mathbf{x}_1, \mathbf{x}_2$ 为两个特征点的归一化坐标，它们满足

$$s_2 \mathbf{x}_2 = s_1 \mathbf{R} \mathbf{x}_1 + \mathbf{t} \quad (10.1)$$

如果要求解两个特征点的深度 s_1, s_2 ，在两式左乘一个 \mathbf{x}_2^T

$$s_2 \mathbf{x}_2^T \mathbf{x}_2 = 0 = s_1 \mathbf{x}_2^T \mathbf{R} \mathbf{x}_1 + \mathbf{x}_2^T \mathbf{t} \quad (10.2)$$

注意：三角测量是由平移得到的，有平移才会有对极几何中的三角形，才有三角化，因此纯旋转是无法使用三角测量的。并且当平移很小的时候像素上的不确定性将导致较大的深度不确定性。因此要提高三角化的精度可以提高特征点的提取精度和使平移量增加，但这又会导致一些问题。

11 3D-2D:PnP

PnP 是求解 3D 到 2D 点对运动的方法，2D-2D 的对极几何的方法需要八个及其以上点对，且存在初始化，纯旋转和尺度问题。然而如果一直一张图片中特征点的 3D 位置，那么最少只需 3 个点对就可以估计相机运动，特征点的 3D 位置可以由三角化或者 RGB-D 相机的深度图确定。PnP 有多种求解方法，如用 3 对点估计位姿的 P3P、直接线性变换 (DLT)、EPnP、UPnP 等，还有非线性优化的方法。

这里略过直接线性变换，简单阐述以下 P3P

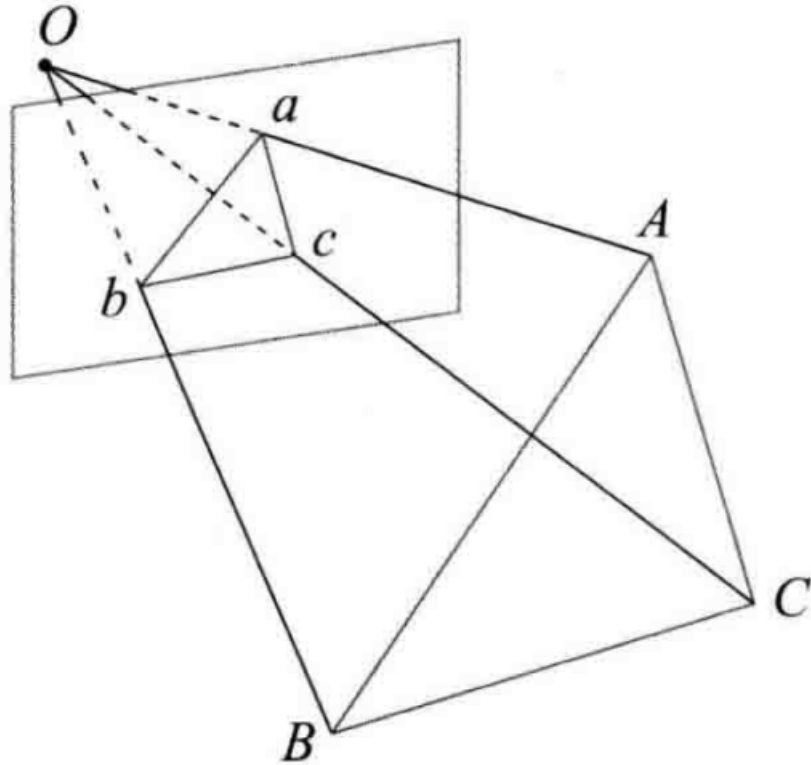


图 12: P3P 问题示意图

根据相似三角形的原理有以下方程

$$\begin{aligned} OA^2 + OB^2 - 2OA \cdot OB \cdot \cos \angle a, b &= AB^2 \\ OB^2 + OC^2 - 2OB \cdot OC \cdot \cos \angle b, c &= BC^2 \\ OA^2 + OC^2 - 2OA \cdot OC \cdot \cos \angle a, c &= AC^2 \end{aligned} \quad (11.1)$$

对上三式全体除以 OC^2 ，并且记 $x = OA/OC, y = OB/OC, v = AB^2/OC^2, w = BC^2/OC^2, uv = AC^2/OC^2$ ，有

$$\begin{aligned}x^2 + y^2 - 2xy \cos < a, b > - v &= 0 \\y^2 + 1 - 2y \cos < b, c > - uv &= 0 \\x^2 + 1 - 2x \cos < a, c > - wv &= 0\end{aligned}\tag{11.2}$$

将上面第一个式子的 v 放到一边，并带入后两式得

$$\begin{aligned}(1 - u)y^2 - ux^2 - \cos < b, c > y + 2uxy \cos < a, b > + 1 &= 0 \\(1 - w)x^2 - wy^2 - \cos < a, c > x + 2wxy \cos < a, b > + 1 &= 0\end{aligned}\tag{11.3}$$

注意方程中的已知量和未知量，该式子中的 x, y 是未知量，因此，该方程组是一个关于 x, y 的二元二次方程。这种方程的求解是一个复杂的过程，需要用吴消元法，该方程最多有四个解，可以用验证点来计算可能的解。但是这个方法存在一定的局限性

1. P3P 只用了 3 个点的信息，当给定多于三个点的时候难以利用更多的信息
2. 如果 3D 点或 2D 点受到噪声影响，或者存在误匹配，则算法失效

所以提出了一些后续改进方法，如 EPnP、UPnP 等。接下来讨论非线性优化的角度来对待 PnP 问题

12 最小化重投影误差求解 PnP

前面所说的线性方法，往往是先求相机位姿，再求空间点位置，而非线性优化则是把它们都看成优化变量，放在一起优化，这一类把相机和三维点放在一起进行最小化的方法广泛统称为 BA(Bundle Adjustment)。

根据前面所学的知识，像素位置与空间位置的关系如下

$$s_i \mathbf{u}_i = \mathbf{KTP}_i\tag{12.1}$$

13 高精度地图的全局规划路径算法有哪些

1. Dijkstra
2. BFS(Best-First-Search)
3. A*
4. hybrid A*
5. D*
6. RRT
7. RRT*
8. 蚁群算法
9. Rectangular Symmetry Reduction(RSR)
10. BUG
11. Beam Search
12. Iterative Deepeningc

目前主流的还是深度优先搜索、广度优先搜索、A* 和 Dijkstra

14 路径规划的主要内容有哪些

路径规划分为：寻路 (Routing)、决策 (Behavioral Decision)、运动规划 (Motion Planning)。寻路主要是全局路径规划，现阶段主要为传统导航地图 (SD 地图) 和高精度地图结合进行路径规划；决策主要为在遇到交通灯和行人时的避让、高速路分流合流的选择、以及路口时转向判断等；运动规划是指局部路径规划，判断前方几百米内的期望行驶路径，需结合汽车运动学、动力学、舒适性和无碰撞等要求。