深蓝学院
shenlanxueyuan.com

视觉SLAM：从理论到实践
第七次课 后端

主讲人 高翔

清华大学 自动控制与工程 博士
慕尼黑工业大学计算机视觉组 博士后
Email: gao.xiang.thu@gmail.com

第七讲 后端

深蓝学院
shenlanxueyuan.com

1. 递归方法
2. 批量方法
3. Pose Graph

1. 递归方法

1. 递归方法

深蓝学院
shenlanxueyuan.com

- 后端（Backend）
  - 从带噪声的数据估计内在状态——状态估计问题
  - Estimated the inner state from noisy data
- 渐进式（Incremental/Recursive）
  - 保持当前状态的估计，在加入新信息时，更新已有的估计（滤波）
  - 线性系统+高斯噪声=卡尔曼滤波器
  - 非线性系统+高斯噪声+线性近似=扩展卡尔曼
  - 非线性系统+非高斯噪声+非参数化=粒子滤波器
  - Sliding window filter & multiple state Kalman (MSCKF)
- 批量式（Batch）
  - 给定一定规模的数据，计算该数据下的最优估计（优化）

1. 递归方法

- 简单的例子
  - 当我们蒙着眼走路时：
    - 一开始我们知道自己在哪
    - 能粗略估计每步的距离，但不确定
    - 不确定性随时间累积
    - 导致自身的位置不确定性越来越大
  - 某个时刻睁开眼睛时
    - 能够观测到周围
    - 尽管也有不确定性，但比估计步长小很多
    - 能够将不确定性保持在一定范围内

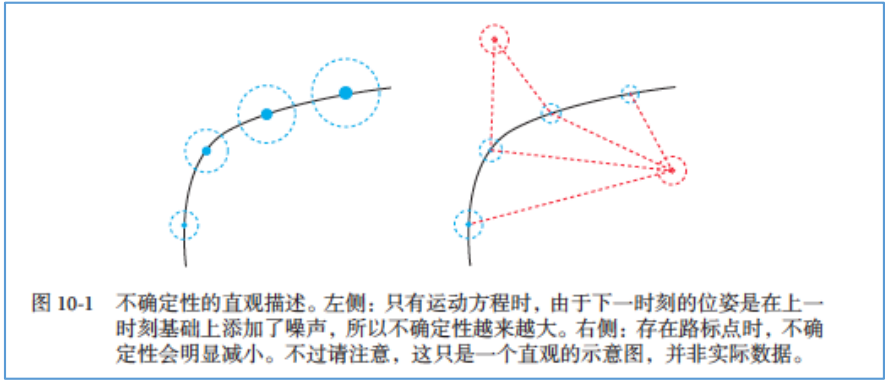


图 10-1　不确定性的直观描述。左侧：只有运动方程时，由于下一时刻的位姿是在上一时刻基础上添加了噪声，所以不确定性越来越大。右侧：存在路标点时，不确定性会明显减小。不过请注意，这只是一个直观的示意图，并非实际数据。

同理，可以用渐进式和批量式两种方式描述这个过程

1. 递归方法

- 数学描述
  - 符号定义：
  - t = 0 … N 时间内，机器人的位姿为：$\boldsymbol{x}_0$ 到 $\boldsymbol{x}_N$，同时有路标 $\boldsymbol{y}_1, \cdots, \boldsymbol{y}_M$
  - 运动和观测方程记为：

$$\begin{cases} \boldsymbol{x}_k = f\left(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k\right) + \boldsymbol{w}_k \\ \boldsymbol{z}_{k,j} = h\left(\boldsymbol{y}_j, \boldsymbol{x}_k\right) + \boldsymbol{v}_{k,j} \end{cases} \quad k = 1, \ldots, N,\ j = 1, \ldots, M.$$

  - 注意：运动和观测都受噪声影响；可以没有运动方程（纯视觉SLAM）
  - 我们从贝叶斯滤波器方法来推导卡尔曼滤波

1. 递归方法

深蓝学院
shenlanxueyuan.com

- 用随机变量表示状态，估计其概率分布
- k 时刻所有待估计的量组成k时刻状态：$\boldsymbol{x}_k \triangleq \{\boldsymbol{x}_k, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_m\}.$
- k 时刻观测统一记成：$\boldsymbol{z}_k$ 方程简化为：

符号稍微有点冲突

$$\begin{cases} \boldsymbol{x}_k = f\left(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k\right) + \boldsymbol{w}_k \\ \boldsymbol{z}_k = h\left(\boldsymbol{x}_k\right) + \boldsymbol{v}_k \end{cases} \quad k = 1, \ldots, N.$$

- 我们从批量方法开始，引出递归方法

1. 递归方法

深蓝学院 shenlanxueyuan.com

$$\begin{cases} x_k = f(x_{k-1}, u_k) + w_k \\ z_k = h(x_k) + v_k \end{cases} \quad k = 1, \ldots, N.$$

- 根据过去0时刻到k时刻的数据
- 估计当前的状态： $P(x_k | x_0, u_{1:k}, z_{1:k})$.
- Bayes展开： $P(x_k | x_0, u_{1:k}, z_{1:k}) \propto P(z_k | x_k) P(x_k | x_0, u_{1:k}, z_{1:k-1})$.

  似然　　先验
- 先验部分按条件概率展开：

$$P(x_k | x_0, u_{1:k}, z_{1:k-1}) = \int P(x_k | x_{k-1}, x_0, u_{1:k}, z_{1:k-1}) P(x_{k-1} | x_0, u_{1:k}, z_{1:k-1}) \mathrm{d}x_{k-1}. \quad (10.6)$$

运动模型的预测　　K-1时刻估计

1. 递归方法

深蓝学院 shenlanxueyuan.com

- 注意先验部分的组成：

k-1时刻的状态估计

$$P(x_k|x_0, u_{1:k}, z_{1:k-1}) = \int P(x_k|x_{k-1}, x_0, u_{1:k}, z_{1:k-1})\, P(x_{k-1}|x_0, u_{1:k}, z_{1:k-1})\, \mathrm{d}x_{k-1}. \tag{10.6}$$

k时刻受先前状态的影响

- 分歧：
  - 可以假设k时刻状态只和k-1时刻有关（假设了一阶马尔可夫性）
  - 或假设k时刻状态与先前所有时刻均有关（不假设马尔可夫性）

1. 递归方法

- 假设一阶马尔可夫性的情况： $P(x_k|x_{k-1}, x_0, u_{1:k}, z_{1:k-1}) = P(x_k|x_{k-1}, u_k)$.
  - 第二项： $P(x_{k-1}|x_0, u_{1:k}, z_{1:k-1}) = P(x_{k-1}|x_0, u_{1:k-1}, z_{1:k-1})$.
- 于是，该公式指出了如何从k-1时刻的状态分布递推至k时刻的分布
  - 只是现在我们还没有代入具体的分布形式
- 在线性模型、高斯状态分布下，我们将得到卡尔曼滤波器
  - 在非线性模型、高斯状态分布下，可以在工作点附近线性展开，得到扩展卡尔曼滤波器

1. 递归方法

深蓝学院 shenlanxueyuan.com

- 卡尔曼滤波器的推导
  - 线性模型和高斯噪声：

$$\begin{cases} \boldsymbol{x}_k = \boldsymbol{A}_k \boldsymbol{x}_{k-1} + \boldsymbol{u}_k + \boldsymbol{w}_k \\ \boldsymbol{z}_k = \boldsymbol{C}_k \boldsymbol{x}_k + \boldsymbol{v}_k \end{cases} \quad k = 1, \ldots, N.$$

$$\boldsymbol{w}_k \sim N(\boldsymbol{0}, \boldsymbol{R}). \quad \boldsymbol{v}_k \sim N(\boldsymbol{0}, \boldsymbol{Q}).$$

  - 状态的高斯分布（区别先后验） $P(x_{k-1}) = N\left(\hat{x}_{k-1}, \hat{P}_{k-1}\right)$ $\quad \overline{x}_k, \overline{P}_k$

后验　　先验

1. 递归方法

深蓝学院
shenlanxueyuan.com

- 预备知识：高斯分布的线性变换
  - 假定 $x \sim N(m, S), y = Ax + b$ 则 y 亦服从高斯分布
  - 有：

$$E[y] = E[Ax + b] = AE[x] + b = Am + b$$

$$Cov[y] = E[(y - E[y])(y - E[y])^T]$$

$$= E[A(x - m)(x - m)^T A^T] = ASA^T$$

1. 递归方法

深蓝学院 shenlanxueyuan.com

- 预测（k-1时刻后验通过运动方程推算k时刻先验）

$$P(\boldsymbol{x}_k|\boldsymbol{x}_0, \boldsymbol{u}_{1:k}, \boldsymbol{z}_{1:k-1}) = N\left(\boldsymbol{A}_k\hat{\boldsymbol{x}}_{k-1} + \boldsymbol{u}_k, \boldsymbol{A}_k\hat{\boldsymbol{P}}_k\boldsymbol{A}_k^{\mathrm{T}} + \boldsymbol{R}\right).$$

需要用到高斯分布的线性组合性质

- 并记成： $\bar{\boldsymbol{x}}_k = \boldsymbol{A}_k\hat{\boldsymbol{x}}_{k-1} + \boldsymbol{u}_k, \quad \bar{\boldsymbol{P}}_k = \boldsymbol{A}_k\hat{\boldsymbol{P}}_{k-1}\boldsymbol{A}_k^{\mathrm{T}} + \boldsymbol{R}.$

先验

- 根据观测方程，可知： $P(z_k \mid x_k) = N(C_k x_k, Q).$
- 根据前面的Bayes展开： $P(\boldsymbol{x}_k|\boldsymbol{x}_0, \boldsymbol{u}_{1:k}, \boldsymbol{z}_{1:k}) \propto P(\boldsymbol{z}_k|\boldsymbol{x}_k)\, P(\boldsymbol{x}_k|\boldsymbol{x}_0, \boldsymbol{u}_{1:k}, \boldsymbol{z}_{1:k-1}).$

- 需要计算两个分布的乘积

1. 递归方法

$$P(\boldsymbol{x}_k|\boldsymbol{x}_0, \boldsymbol{u}_{1:k}, \boldsymbol{z}_{1:k}) \propto P(\boldsymbol{z}_k|\boldsymbol{x}_k) P(\boldsymbol{x}_k|\boldsymbol{x}_0, \boldsymbol{u}_{1:k}, \boldsymbol{z}_{1:k-1}).$$

- 技巧：已经两侧均是高斯分布，所以：$N(\hat{x}_k, \hat{P}_k) = N\left(C_k x_k, Q\right) \times N(\bar{x}_k, \bar{P}_k).$
- 既然都是高斯分布，我们比较指数部分的二次项和一次项部分即可。
- 指数部分展开：

$$\left(x_k - \hat{x}_k\right)^{\mathrm{T}} \hat{P}_k^{-1} \left(x_k - \hat{x}_k\right) = \left(z_k - C_k x_k\right)^{\mathrm{T}} Q^{-1} \left(z_k - C_k x_k\right) + \left(x_k - \bar{x}_k\right)^{\mathrm{T}} \bar{P}_k^{-1} \left(x_k - \bar{x}_k\right).$$

- 比较 x_k 的二次和一次项系数，对于二次项，有：

$$\hat{P}_k^{-1} = C_k^{\mathrm{T}} Q^{-1} C_k + \bar{P}_k^{-1}.$$

1. 递归方法

- 再考虑一次项：$\left(x_k-\hat{x}_k\right)^{\mathrm{T}}\hat{P}_k^{-1}\left(x_k-\hat{x}_k\right)=\left(z_k-C_kx_k\right)^{\mathrm{T}}Q^{-1}\left(z_k-C_kx_k\right)+\left(x_k-\bar{x}_k\right)^{\mathrm{T}}\bar{P}_k^{-1}\left(x_k-\bar{x}_k\right).$

- 比较一次项系数：$-2\hat{x}_k^{\mathrm{T}}\hat{P}_k^{-1}x_k=-2z_k^{\mathrm{T}}Q^{-1}C_kx_k-2\bar{x}_k^{\mathrm{T}}\bar{P}_k^{-1}x_k$
- 整理之：$\hat{P}_k^{-1}\hat{x}_k=C_k^{\mathrm{T}}Q^{-1}z_k+\bar{P}_k^{-1}\bar{x}_k$
- 两侧同乘 $\hat{P}_k$ 并定义：$K=\hat{P}_kC_k^{\mathrm{T}}Q^{-1}$，得：

$$\begin{aligned}\hat{x}_k&=\hat{P}_kC_k^{\mathrm{T}}Q^{-1}z_k+\hat{P}_k\bar{P}_k^{-1}\bar{x}_k\\&=Kz_k+\left(I-KC_k\right)\bar{x}_k=\bar{x}_k+K\left(z_k-C_k\bar{x}_k\right).\end{aligned}$$

称为卡尔曼滤波的更新式

1. 递归方法

深蓝学院 shenlanxueyuan.com

- 卡尔曼增益： $K = \hat{P}_k C_k^{\mathrm{T}} Q^{-1}$ 需要计算 $\hat{P}_k$
- 另一种形式（不需要计算后验协方差）：

$$K = \left(C_k^T Q^{-1} C_k + \bar{P}_k^{-1}\right)^{-1} C_k^T Q^{-1}$$
$$= \bar{P}_k C_k^T (Q + C_k \bar{P}_k C^T)^{-1}$$

- 转换需要引入 Sherman-Morrison-Woodbury恒等式：

$$\left(\boldsymbol{A}^{-1} + \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C}\right)^{-1} \equiv \boldsymbol{A} - \boldsymbol{A}\boldsymbol{B}\left(\boldsymbol{D} + \boldsymbol{C}\boldsymbol{A}\boldsymbol{B}\right)^{-1}\boldsymbol{C}\boldsymbol{A} \tag{2.75a}$$

$$\left(\boldsymbol{D} + \boldsymbol{C}\boldsymbol{A}\boldsymbol{B}\right)^{-1} \equiv \boldsymbol{D}^{-1} - \boldsymbol{D}^{-1}\boldsymbol{C}\left(\boldsymbol{A}^{-1} + \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C}\right)^{-1}\boldsymbol{B}\boldsymbol{D}^{-1} \tag{2.75b}$$

$$\boldsymbol{A}\boldsymbol{B}\left(\boldsymbol{D} + \boldsymbol{C}\boldsymbol{A}\boldsymbol{B}\right)^{-1} \equiv \left(\boldsymbol{A}^{-1} + \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C}\right)^{-1}\boldsymbol{B}\boldsymbol{D}^{-1} \tag{2.75c}$$

$$\left(\boldsymbol{D} + \boldsymbol{C}\boldsymbol{A}\boldsymbol{B}\right)^{-1}\boldsymbol{C}\boldsymbol{A} \equiv \boldsymbol{D}^{-1}\boldsymbol{C}\left(\boldsymbol{A}^{-1} + \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C}\right)^{-1} \tag{2.75d}$$

1. 递归方法

- 小结：

1. 预测：

$$\bar{x}_k = A_k \hat{x}_{k-1} + u_k, \qquad \bar{P}_k = A_k \hat{P}_{k-1} A_k^T + R$$

2. 校正：
   - 计算卡尔曼增益 $K = \bar{P}_k C_k^T (Q + C_k \bar{P}_k C^T)^{-1}$
   - 更新状态估计

$$\hat{x}_k = \bar{x}_k + K(z_k - C_k \bar{x}_k)$$
$$\hat{P}_k = (I - K C_k) \bar{P}_k$$

经典卡尔曼滤波器的五个公式
给出了线性高斯系统的最优无偏估计

1. 递归方法

深蓝学院 shenlanxueyuan.com

- Kalman Filter的非线性扩展：EKF
- 当f,h为非线性函数时：

$$\begin{cases} \boldsymbol{x}_k = f\left(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k\right) + \boldsymbol{w}_k \\ \boldsymbol{z}_k = h\left(\boldsymbol{x}_k\right) + \boldsymbol{v}_k \end{cases} \qquad k = 1, \ldots, N.$$

- 在工作点附近进行一阶Taylor展开：

$$\boldsymbol{x}_k \approx f\left(\hat{\boldsymbol{x}}_{k-1}, \boldsymbol{u}_k\right) + \underbrace{\left.\frac{\partial f}{\partial \boldsymbol{x}_{k-1}}\right|_{\hat{\boldsymbol{x}}_{k-1}}}_{\text{记F}} \left(\boldsymbol{x}_{k-1} - \hat{\boldsymbol{x}}_{k-1}\right) + \boldsymbol{w}_k.$$

$$\boldsymbol{z}_k \approx h\left(\bar{\boldsymbol{x}}_k\right) + \underbrace{\left.\frac{\partial h}{\partial \boldsymbol{x}_k}\right|_{\bar{\boldsymbol{x}}_k}}_{\text{记H}} \left(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k\right) + \boldsymbol{n}_k.$$

1. 递归方法

深蓝学院
shenlanxueyuan.com

- 预测部分： $\bar{x}_k = f(\hat{x}_{k-1}, u_k), \quad \bar{P}_k = F\hat{P}_k F^{\mathrm{T}} + R_k.$
- 卡尔曼增益： $K_k = \bar{P}_k H^{\mathrm{T}} \left(H\bar{P}_k H^{\mathrm{T}} + Q_k\right)^{-1}.$
- 更新部分： $\hat{x}_k = \bar{x}_k + K_k\left(z_k - h(\bar{x}_k)\right), \hat{P}_k = (I - K_k H)\bar{P}_k.$

1. 递归方法

- EKF优点
  - 推导简单清楚，适用各种传感器形式
  - 易于做多传感器融合
- EKF缺点
  - 一阶马尔可夫性过于简单
  - 可能会发散（要求数据不能有outlier）
  - 线性化误差
  - 需要存储所有状态量的均值和方差，平方增长

2. 批量方法

2. 批量方法

- Bundle Adjustment已经在之前介绍过了
  - 事实上BA属于批量式的优化方法
  - 给定很多个相机位姿与观测数据，计算最优的状态估计
  - 定义每个运动/观测方程的误差，并从初始估计开始寻找梯度下降

$$\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}\|\boldsymbol{e}_{ij}\|^2=\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}\|\boldsymbol{z}_{ij}-h(\boldsymbol{\xi}_i,\boldsymbol{p}_j)\|^2.$$

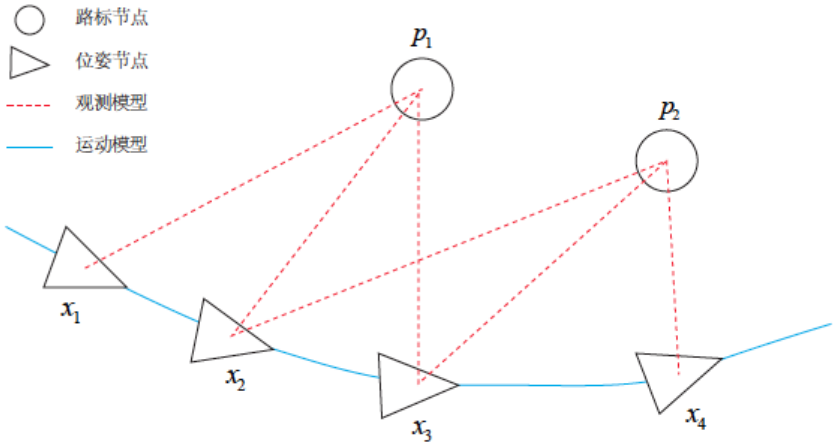


图 6-2　图优化的例子。

2. 批量方法

- BA问题与图结构的关系
  - BA虽然是个纯优化问题，但亦可以用图模型清晰地表述出来
  - 顶点为优化变量，边为运动/观测约束
  - 本身还有一些特殊的结构

- 考虑在位姿 i 处对路标 j 的一次观测 zij：

$$e_{ij} = z_{ij} - h\left(x_i, y_j\right)$$

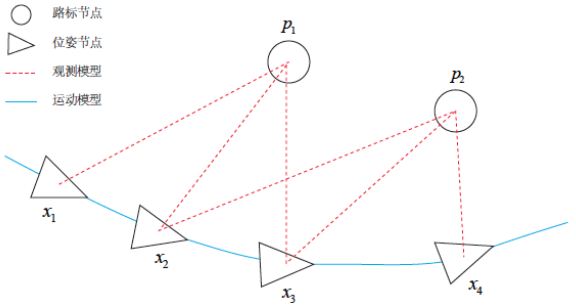


图 6-2　图优化的例子。

特点：

- 每个观测只关系两个变量，其中一个是相机，一个是路标
- 纯视觉Ba中，不存在相机与相机/路标与路标之间的关联
- 整个误差函数由许多个这样小的项组成

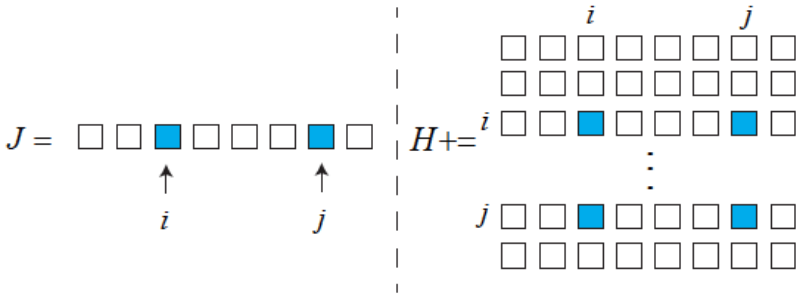
2. BA与图优化

深蓝学院 shenlanxueyuan.com

- 根据Gauss-Newton或Levernburg-Marquardt，最终我们会求解 Hx=-b 形式的方程
- 每一个误差项会对整体线程方程产生贡献：
- 而误差仅和 i, j 相关，导致它的雅可比为稀疏矩阵：$H = \sum_{i,j} J_{ij}^T J_{ij}$

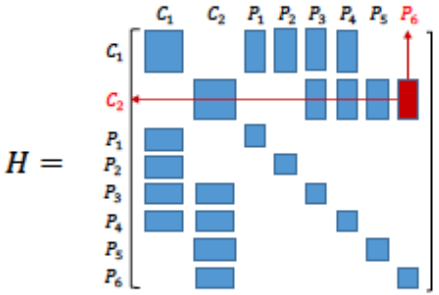
$$\boldsymbol{J}_{ij}(\boldsymbol{x}) = \left(\mathbf{0}_{2\times 6}, ... \mathbf{0}_{2\times 6}, \frac{\partial \boldsymbol{e}_{ij}}{\partial \boldsymbol{\xi}_i}, \mathbf{0}_{2\times 6}, ... \mathbf{0}_{2\times 3}, ... \mathbf{0}_{2\times 3}, \frac{\partial \boldsymbol{e}_{ij}}{\partial \boldsymbol{p}_j}, \mathbf{0}_{2\times 3}, ... \mathbf{0}_{2\times 3} \right).$$

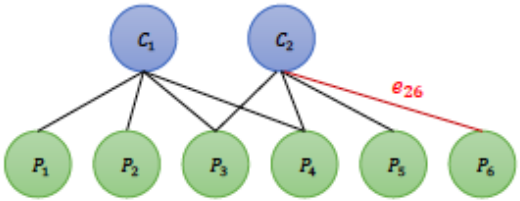
- 示意图：



2. BA与图优化

- 如果对变量做好排序，例如所有相机位姿在前，路标在后，那么 H 有一定的特殊结构：



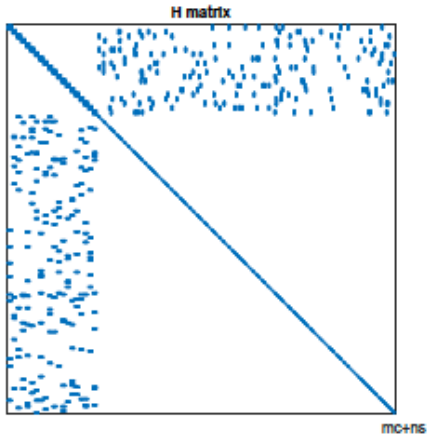


图模型与H矩阵存在对应关系：

- 图模型中存在边=>H相应地方出现非零块

2. BA与图优化

- 实际当中的H，路标数量远大于位姿数量（箭头形矩阵或镐子形矩阵）



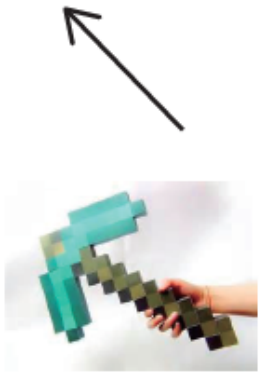
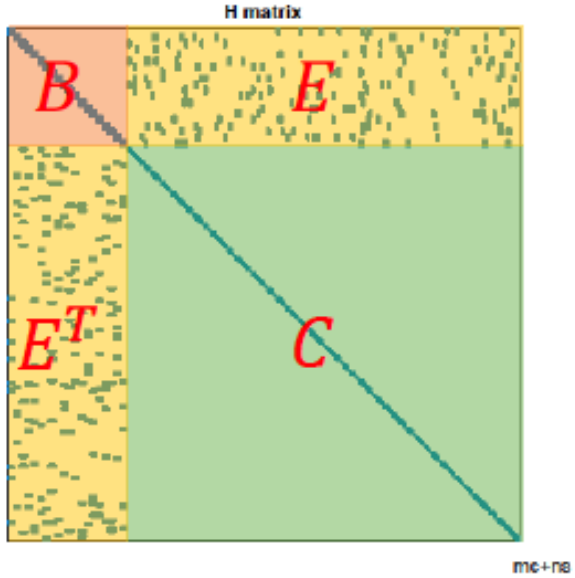


图 10-8　一般情况下的 H 矩阵。



2. BA与图优化

深蓝学院 shenlanxueyuan.com

- 利用H的特殊结构，可以大幅加速Hx=-b线性方程的求解
- 加速手段又称边缘化（Marginalization）[意义有很多种]
- Hx=-b的结构：

$$\begin{bmatrix} \boldsymbol{B} & \boldsymbol{E} \\ \boldsymbol{E}^{\mathrm{T}} & \boldsymbol{C} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{x}_c \\ \Delta \boldsymbol{x}_p \end{bmatrix} = \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{bmatrix}.$$

B：小，对角块；C，大，对角块
E和E转置：与图模型对应，可稠密

- 因为 C 是对角块，所以可以用C对E进行消元（高斯消元），得到：

$$\begin{bmatrix} \boldsymbol{I} & -\boldsymbol{E}\boldsymbol{C}^{-1} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{B} & \boldsymbol{E} \\ \boldsymbol{E}^{\mathrm{T}} & \boldsymbol{C} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{x}_c \\ \Delta \boldsymbol{x}_p \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{E}\boldsymbol{C}^{-1} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{bmatrix}.$$

从而

$$\begin{bmatrix} \boldsymbol{B} - \boldsymbol{E}\boldsymbol{C}^{-1}\boldsymbol{E}^{\mathrm{T}} & \boldsymbol{0} \\ \boldsymbol{E}^{\mathrm{T}} & \boldsymbol{C} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{x}_c \\ \Delta \boldsymbol{x}_p \end{bmatrix} = \begin{bmatrix} \boldsymbol{v} - \boldsymbol{E}\boldsymbol{C}^{-1}\boldsymbol{w} \\ \boldsymbol{w} \end{bmatrix}.$$

2. BA与图优化

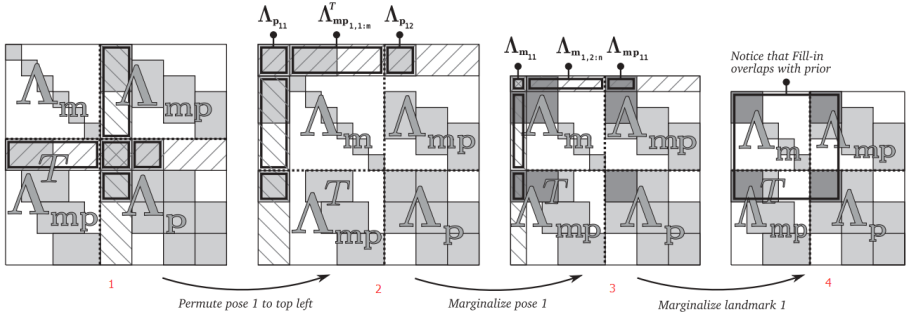
$$\begin{bmatrix} \boldsymbol{B}-\boldsymbol{E}\boldsymbol{C}^{-1}\boldsymbol{E}^{\mathrm{T}} & \boldsymbol{0} \\ \boldsymbol{E}^{\mathrm{T}} & \boldsymbol{C} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{x}_c \\ \Delta \boldsymbol{x}_p \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}-\boldsymbol{E}\boldsymbol{C}^{-1}\boldsymbol{w} \\ \boldsymbol{w} \end{bmatrix}.$$

- 该方程组分为两步来求：
  1. 求解上半部分，规模较小，得到 Dx_c
  2. 将结果代入下半部分，得到 Dx_p
- 这个做法称为Marginalization或Schur消元
  - 从消元角度来讲，亦可使用Cholesky等其他消元方式解此稀疏方程
  - 从Marginalization角度来讲，是我们把所有的路标信息边缘化到了相机的信息中

2. BA与图优化

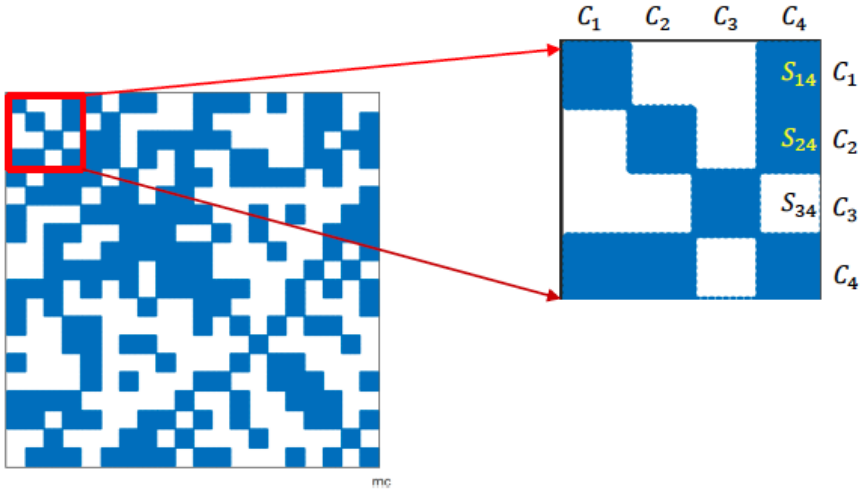
- Marginalization(marg)
  - 联合分布=边缘分布乘条件分布：$P(\boldsymbol{x}_c, \boldsymbol{x}_p) = P(\boldsymbol{x}_c) \cdot P(\boldsymbol{x}_p|\boldsymbol{x}_c).$
  - 上述的做法marg掉了所有的路标点，让概率全跑到相机位姿中，这是为了加速求解 Hx=-b
  - 然而，在sliding window等其他做法中，也可以选择边缘化一部分相机/路标，做法同上

- 但是，通常情况下的Marg操作会给H矩阵带来填充（fill-in），使其不再具有稀疏结构；
- 所以，要么刻意选择特定的marg策略，要么仅使用稠密的H求解线性方程，但这样效率会降低



2. BA与图优化

- Marginalization之后，左上角矩阵不再具有稀疏结构：
- 但它反映了相机之间的共视关系（Co-visibility）



S14/S24处的非零块表示1,4相机，2,4相机看到了同一个路标点

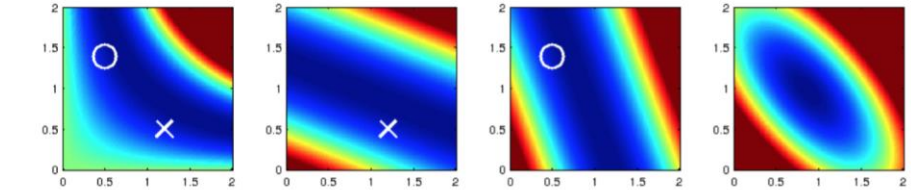
2. BA与图优化

深蓝学院 shenlanxueyuan.com

- Marg操作与FEJ
- 在处理Marg时，被Marg的部分要使用First-estimate-Jacobian
- 否则可能导致整个问题零空间被降维

Windowed, real-time optimization: Consistency.

(for now, let's assume we have initializations, and know which points to use and where they are visible.)



$E = E_1 + E_2$
$= (xy-1)^2 + (xy-1)^2$

$E_1' = E_1$
Lin. around (0.5, 1.4)

$E_2' = E_2$
Lin. around (1.2, 0.5)

$E' = E_1' + E_2'$

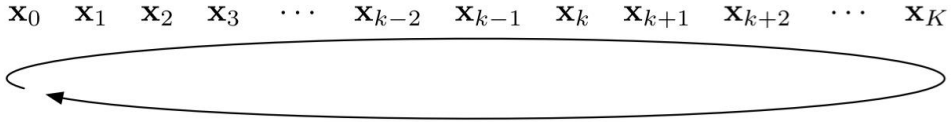
nullspace disappears!

never combine linearizations around different linearization points,
especially in the presence of non-linear nullspaces!
It will render unobservable dimensions observable, and corrupt the system.

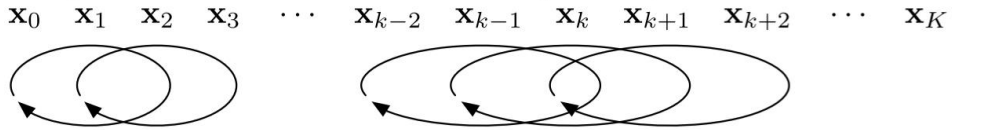
2. 批量方法

- 递归方法和批量方法的对比

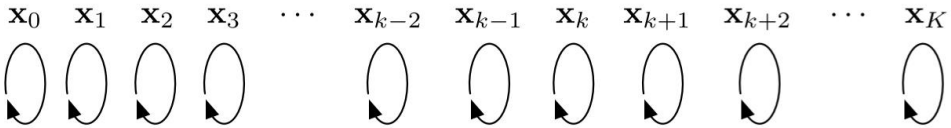
Gauss-Newton iterates over the entire trajectory, but runs offline and not in constant time



Sliding-window filters iterate over several timesteps at once, run online and in constant time



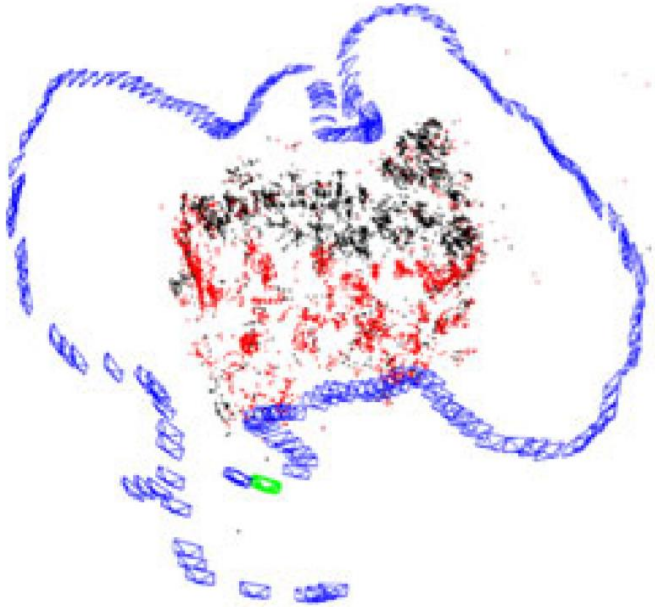
IEKF iterates at only one timestep at a time, but runs online and in constant time



2. 批量方法

深蓝学院
shenlanxueyuan.com

- 在SLAM中使用BA
  - 关键帧和地图的管理
- 批量方法
  - 用BA优化一部分图
  - 其余的固定
- 递归方法（滑动窗口法）
  - 保留一定数量的关键帧
  - 使用BA来优化窗口内的关键帧
  - 新的关键帧到来时，边缘化老的关键帧



3. Pose Graph

3. Pose Graph

深蓝学院 shenlanxueyuan.com

- 实际当中BA的计算量很大
  - 通常放在单独的后台线程中计算而无法实时
  - 主要计算来自于大量的特征点
- Pose Graph即是省略了特征点的Bundle Adjustment

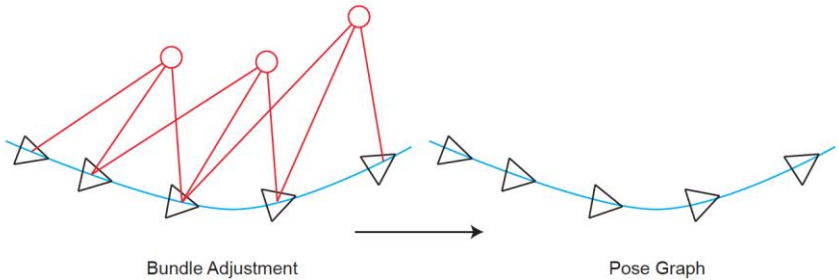


图 11-1　Pose Graph 示意图。当我们不再优化 Bundle Adjustment 中的路标点，仅把它们看成对姿态节点的约束时，就得到了一个计算规模减小很多的 Pose Graph。

3. Pose Graph

深蓝学院 shenlanxueyuan.com

- Pose Graph
  - 顶点仅由相机位姿组成
  - 边为位姿与位姿间的约束

$$\Delta\boldsymbol{\xi}_{ij} = \boldsymbol{\xi}_i^{-1} \circ \boldsymbol{\xi}_j = \ln\left(\exp\left((-\boldsymbol{\xi}_i)^\wedge\right)\exp\left(\boldsymbol{\xi}_j^\wedge\right)\right)^\vee,$$

$$\boldsymbol{\Delta T}_{ij} = \boldsymbol{T}_i^{-1}\boldsymbol{T}_j.$$

  - 不同的观测组成了不同的边
  - 误差项：

$$\begin{aligned} \boldsymbol{e}_{ij} &= \ln\left(\Delta\boldsymbol{T}_{ij}^{-1}\boldsymbol{T}_i^{-1}\boldsymbol{T}_j\right)^\vee \\ &= \ln\left(\exp\left((-\boldsymbol{\xi}_{ij})^\wedge\right)\exp\left((-\boldsymbol{\xi}_i)^\wedge\right)\exp\left(\boldsymbol{\xi}_j^\wedge\right)\right)^\vee. \end{aligned}$$

3. Pose Graph

深蓝学院 shenlanxueyuan.com

伴随 $\exp\left((\mathrm{Ad}(\boldsymbol{T})\boldsymbol{\xi})^\wedge\right) = \boldsymbol{T}\exp(\boldsymbol{\xi}^\wedge)\boldsymbol{T}^{-1}.$

- 误差项关于两个顶点的雅可比：

$$
\begin{aligned}
\hat{\boldsymbol{e}}_{ij} &= \ln\left(\boldsymbol{T}_{ij}^{-1}\boldsymbol{T}_i^{-1}\exp((-\boldsymbol{\delta\xi}_i)^\wedge)\exp(\boldsymbol{\delta\xi}_j^\wedge)\boldsymbol{T}_j\right)^\vee \\
&= \ln\left(\boldsymbol{T}_{ij}^{-1}\boldsymbol{T}_i^{-1}\boldsymbol{T}_j\exp\left(\left(-\mathrm{Ad}(\boldsymbol{T}_j^{-1})\boldsymbol{\delta\xi}_i\right)^\wedge\right)\exp\left(\left(\mathrm{Ad}(\boldsymbol{T}_j^{-1})\boldsymbol{\delta\xi}_j\right)^\wedge\right)\right)^\vee \\
&\approx \ln\left(\boldsymbol{T}_{ij}^{-1}\boldsymbol{T}_i^{-1}\boldsymbol{T}_j\left[\boldsymbol{I} - (\mathrm{Ad}(\boldsymbol{T}_j^{-1})\boldsymbol{\delta\xi}_i)^\wedge + (\mathrm{Ad}(\boldsymbol{T}_j^{-1})\boldsymbol{\delta\xi}_j)^\wedge\right]\right)^\vee \\
&\approx \boldsymbol{e}_{ij} + \frac{\partial \boldsymbol{e}_{ij}}{\partial \boldsymbol{\delta\xi}_i}\boldsymbol{\delta\xi}_i + \frac{\partial \boldsymbol{e}_{ij}}{\partial \boldsymbol{\delta\xi}_j}\boldsymbol{\delta\xi}_j
\end{aligned}
$$

$$
\frac{\partial \boldsymbol{e}_{ij}}{\partial \boldsymbol{\delta\xi}_i} = -\boldsymbol{\mathcal{J}}_r^{-1}(\boldsymbol{e}_{ij})\mathrm{Ad}(\boldsymbol{T}_j^{-1}).
$$

$$
\frac{\partial \boldsymbol{e}_{ij}}{\partial \boldsymbol{\delta\xi}_j} = \boldsymbol{\mathcal{J}}_r^{-1}(\boldsymbol{e}_{ij})\mathrm{Ad}(\boldsymbol{T}_j^{-1}).
$$

$$
\boldsymbol{\mathcal{J}}_r^{-1}(\boldsymbol{e}_{ij}) \approx \boldsymbol{I} + \frac{1}{2}\begin{bmatrix} \boldsymbol{\phi}_e^\wedge & \boldsymbol{\rho}_e^\wedge \\ \boldsymbol{0} & \boldsymbol{\phi}_e^\wedge \end{bmatrix}.
$$

3. Pose Graph

深蓝学院
shenlanxueyuan.com

- 实践：Pose球的例子

仿真的球形轨迹
Odom边和Loop边