基于卡尔曼滤波的视觉SLAM方法

计算机图形学与混合现实研讨会（GAMES）线上讲座

《视觉SLAM教程》系列讲座之三

2018年04月26日

邹丹平副教授 导航与感知所

电子信息与电气工程学院

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

讲座概要

- 视觉SLAM方法分类
- 相关基础知识
- 基于扩展卡尔曼滤波SLAM方法
  - MonoSLAM
- 多状态约束卡尔曼滤波方法
  - MSCKF
- 结构和环境下的视觉SLAM方法
  - StructSLAM/VIO

视觉SLAM方法分类

- 视觉SLAM在近十年发展非常迅速，涌现了很多优秀的方法和代表系统。
  - 滤波法 MonoSLAM,2003
  - 关键帧优化方法 PTAM 2007, ORB-SLAM 2015
  - 稀疏法/密集法 DTAM 2012, LSD-SLAM 2014
  - 间接法/直接法 SVO 2014, DSO 2016
- 从所采用数学框架，视觉SLAM方法可分成两大类：
  - 基于关键帧优化的方法
  - 基于卡尔曼滤波的方法
- 今天的讲座着重介绍卡尔曼滤波的视觉SLAM方法

基于卡尔曼滤波的视觉SLAM方法

- 为什么要用卡尔曼滤波的视觉SLAM方法？

基于卡尔曼滤波的视觉SLAM方法

- 为什么要用卡尔曼滤波的视觉SLAM方法？
- 设计一个视觉SLAM系统需要考虑的关键因素：
  - 精度 - 累积误差尽量小
  - 实时性 - 计算速度足够快
  - 鲁棒性(Robust) - 在图像质量不佳、小视差运动方式下稳定运行
  - 可扩展性 - 可方便加入其它传感器，快速进行多源融合

基于卡尔曼滤波的视觉SLAM方法

- 相比基于关键帧优化方法，基于卡尔曼滤波的视觉SLAM方法在**鲁棒性**和**可扩展性**有优势。
- 使用一些更好的设计，相同精度下，基于卡尔曼滤波的视觉SLAM方法还具有**实时性**的优势
- 如果是重心放在在稳定性、扩展性，基于卡尔曼滤波的视觉SLAM是一个比较好的选择。
- 卡尔曼滤波方法还给出了状态估算的协方差矩阵，有助于评估系统运行的精度

什么是卡尔曼滤波？

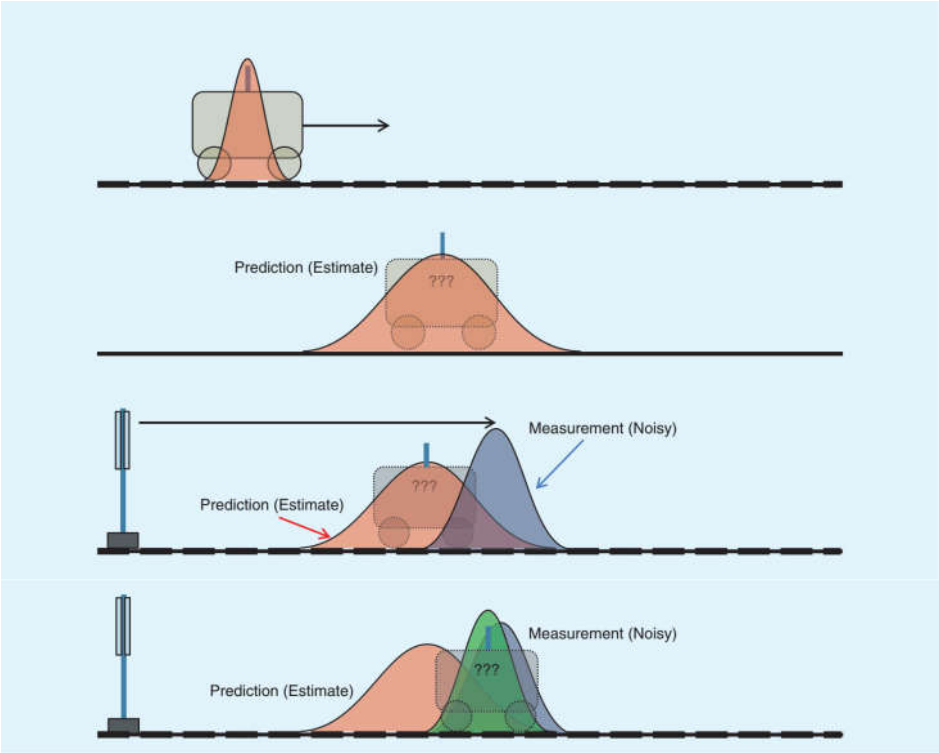
- Rudolf Emil Kalman （1930-2016）匈牙利数学家，电子工程师
  "A new approach to linear filtering and prediction problems." *Journal of basic Engineering* 82.1 (1960): 35-45.
- 卡尔曼滤波算法可应用的领域：
  - 视频/雷达跟踪
  - 定位导航
  - 通信
  - 语音
  - 地址勘探
  - 天气预报
  - 故障争端
  - 证券与股票行情预测

什么是卡尔曼滤波？

- 先来一个快速理解卡尔曼滤波的例子

预测

更新



Faragher, Ramsey. "Understanding the basis of the kalman filter via a simple and intuitive derivation" *IEEE Signal processing magazine* 29.5 (2012): 128-132.

什么是卡尔曼滤波？

- 卡尔曼滤波的要点：
  - 所有变量都是随机变量，由概率密度函数(高斯概率密度函数）来描述
  - 由预测、更新两部分组成：
    - 预测：根据自身的运动特性，预测下一时刻的状态分布
    - 更新：根据新的观测值，调整预测的状态分布

什么是卡尔曼滤波？

- 虽然用起来非常简单，但卡尔曼滤波理解和推导还是比较费劲
- 接下来的介绍思路：

预备知识 → **贝叶斯滤波** → **卡尔曼滤波**

预备知识
- 高斯分布
- 高斯变量的变换
- 高斯分布的分拆/组合
- 高斯分布的边缘化

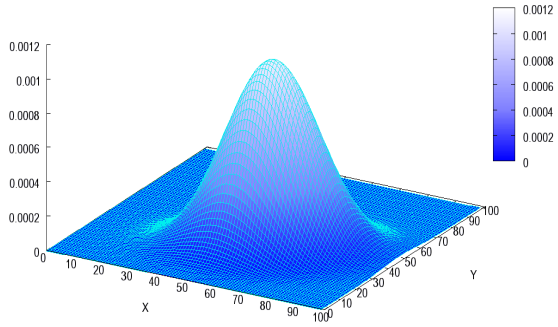
贝叶斯滤波
- 贝叶斯公式
- 先验、后验和似然
- 贝叶斯推断
- 贝叶斯滤波

卡尔曼滤波
- 预测
- 更新
- 非线性情况

基础知识

- 高斯概率密度函数(Gaussian Probability Density Function)

$$P(\boldsymbol{x}) = \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}|}} \exp(-\tfrac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}))$$



- 如果一个随机变量服从高斯概率密度分布，我们称之为**高斯随机变量**：

$$\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

高斯分布特性

- 协方差矩阵传递(Covariance propagation)
  - 一个高斯随机变量的仿射变换（线性变换+平移）还是高斯随机变量

$$\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \qquad \boldsymbol{x}' \sim \mathcal{N}(\boldsymbol{\mu}', \boldsymbol{\Sigma}')$$

$$\boldsymbol{x}' = \mathbf{A}\boldsymbol{x} + \boldsymbol{b}$$

$$\begin{aligned} E[\boldsymbol{x}'] &= E[\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}] = \boldsymbol{A}E[\boldsymbol{x}] + \boldsymbol{b} \\ cov[\boldsymbol{x}'] &= E[(\boldsymbol{x}' - E[\boldsymbol{x}'])(\boldsymbol{x}' - E[\boldsymbol{x}'])^T] \\ &= \boldsymbol{A}E[(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^T]\boldsymbol{A}^T \end{aligned}$$

$$\boldsymbol{\mu}' = \boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b}$$

$$\boldsymbol{\Sigma}' = \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^T$$

高斯分布特性

- 协方差矩阵传递(Covariance propagation)
  - 多个独立的高斯随机变量的线性组合还是高斯随机变量：

$$\boldsymbol{x}_1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \qquad \boldsymbol{x}_2 \sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$$

$$\boldsymbol{x}' = \mathbf{A}\boldsymbol{x}_1 + \mathbf{B}\boldsymbol{x}_2$$

$$\boldsymbol{x}' \sim \mathcal{N}(\boldsymbol{\mu}', \boldsymbol{\Sigma}')$$

$$\boldsymbol{\mu}' = E[\boldsymbol{x}'] = \mathbf{A}\boldsymbol{\mu}_1 + \mathbf{B}\boldsymbol{\mu}_2$$

$$\boldsymbol{\Sigma}' = cov[\boldsymbol{x}'] = \mathbf{A}\boldsymbol{\Sigma}_1\mathbf{A}^T + \mathbf{B}\boldsymbol{\Sigma}_2\mathbf{B}^T$$

高斯分布特性

- 一个多元的高斯联合分布分拆为一个先验分布与条件分布的乘积。

$$P(\boldsymbol{x}) = P(\boldsymbol{x}_1|\boldsymbol{x}_2)P(\boldsymbol{x}_2)$$

- 假设该分布为： $\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} \sim \mathcal{N}(\begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix})$

- 那么条件概率密度函数与先验（边缘）概率密度函数分别为：

$$\boxed{P(\boldsymbol{x}_1|\boldsymbol{x}_2) = \mathcal{N}(\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2), \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21})}$$

$$P(\boldsymbol{x}_2) = \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_{22})$$

分拆公式

高斯分布特性

- 反过来，如果我们事先知道条件概率和先验概率

$$P(\boldsymbol{x}_2) = \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_{22})$$

$$P(\boldsymbol{x}_1|\boldsymbol{x}_2) = \mathcal{N}(\mathbf{H}\mathbf{x}_2, \mathbf{R})$$

高斯分布特性

- 反过来，如果我们事先知道条件概率和先验概率

$$P(\boldsymbol{x}_2) = \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_{22})$$

$$P(\boldsymbol{x}_1|\boldsymbol{x}_2) = \mathcal{N}(\mathbf{H}\mathbf{x}_2, \mathbf{R})$$

- 那么他们的联合分布 $P(\boldsymbol{x})$ 可从调节概率和先验概率组合而成：

$$\boxed{\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{H}\boldsymbol{\mu}_2 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \begin{bmatrix} \mathbf{H}\boldsymbol{\Sigma}_{22}\mathbf{H}^T + \mathbf{R} & \mathbf{H}\boldsymbol{\Sigma}_{22} \\ \boldsymbol{\Sigma}_{22}\mathbf{H}^T & \boldsymbol{\Sigma}_{22} \end{bmatrix}\right)}$$

组合公式

高斯分布特性

- 高斯分布的边缘化（Marginalization）
- 什么是联合概率分布的边缘化？（为什么叫边缘化，因为在表格边缘）

Y \ X	x_1	x_2	x_3	x_4	$P_y(Y)$ ↓
y_1	$4/32$	$2/32$	$1/32$	$1/32$	$8/32$
y_2	$2/32$	$4/32$	$1/32$	$1/32$	$8/32$
y_3	$2/32$	$2/32$	$2/32$	$2/32$	$8/32$
y_4	$8/32$	0	0	0	$8/32$
$P_x(X)$ →	$16/32$	$8/32$	$4/32$	$4/32$	

- 对联合概率分布某个随机变量进行边缘化就是对该变量进行积分的过程

高斯分布特性

- 我们对高斯随机变量 $\boldsymbol{x}_2$ 进行边缘化，：

$$
\begin{aligned}
\int_{\boldsymbol{x}_2} P(\boldsymbol{x}_1, \boldsymbol{x}_2) d\boldsymbol{x}_2 &= \int_{\boldsymbol{x}_2} P(\boldsymbol{x}_2|\boldsymbol{x}_1) P(\boldsymbol{x}_1) d\boldsymbol{x}_2 \\
&= \int_{\boldsymbol{x}_2} P(\boldsymbol{x}_2|\boldsymbol{x}_1) d\boldsymbol{x}_2 P(\boldsymbol{x}_1) \\
&= P(\boldsymbol{x}_1) = \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11})
\end{aligned}
$$

- 得到边缘化概率密度函数 $P(\boldsymbol{x}_1)$
- 对高斯分布的边缘化，只需将对应变量的行和列去除

$$
\mathbf{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11}) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_1 \\ \cancel{\boldsymbol{\mu}_2} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \cancel{\boldsymbol{\Sigma}_{12}} \\ \cancel{\boldsymbol{\Sigma}_{21}} & \cancel{\boldsymbol{\Sigma}_{22}} \end{bmatrix}\right)
$$

小结

- 多维高斯概率密度函数
- 协方差的传播 (covariance propagation)
  - 仿射变换
  - 多个独立高斯随机变量的随机组合
- 高斯概率密度函数的分拆与组合：

$$P(\boldsymbol{x}) = P(\boldsymbol{x}_1|\boldsymbol{x}_2)P(\boldsymbol{x}_2)$$

- 高斯概率密度函数的边缘化

$$\mathbf{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11}) = \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_1 \\ \cancel{\boldsymbol{\mu}_2}\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_{11} & \cancel{\boldsymbol{\Sigma}_{12}} \\ \cancel{\boldsymbol{\Sigma}_{21}} & \cancel{\boldsymbol{\Sigma}_{22}}\end{bmatrix}\right)$$

贝叶斯公式

- 贝叶斯公式： $P(X,Y) = P(X|Y)P(Y)$

  联合概率=条件概率×边缘概率

- 后验概率： $P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)} = \frac{P(Y|X)P(X)}{\sum_{X'} P(Y|X')P(X')}$

- 若 Y 是观测量，X 是所要估算的参数，那么

$P(X)$ **：先验概率**

$P(X|Y)$ **：后验概率**

$P(Y|X)$ **：似然函数**

贝叶斯推断(Bayesian Inference)

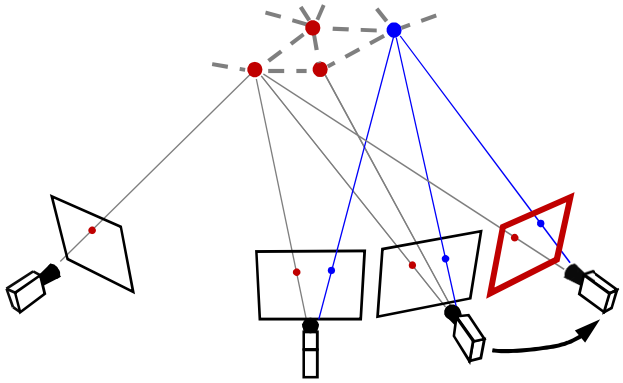
- 把所要估算的状态变量和观测量都看成是随机变量
- 如视觉SLAM的状态变量：

$$\boldsymbol{x}_{1:k} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_k\}$$

  - 其中 $\boldsymbol{x}_k$ 表示$\boldsymbol{K}$ 时刻下相机的姿态、位置速度和特征点的三维坐标（地图）
- 视觉SLAM的观测量：

$$\boldsymbol{z}_{1:k} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_k\}$$

  - 其中$\boldsymbol{z}_k$ 表示在K时刻下的特征点的像素坐标



贝叶斯推断

- 因此，我们将SLAM问题转化为求解如下后验概率问题：

$$P(\boldsymbol{x}_{1:k}|\boldsymbol{z}_{1:k},\boldsymbol{x}_0)$$

  - 其中 $\boldsymbol{x}_0$ 为初始状态。

- 实时上，我们更多关心的是**当前相机与地图**的状态，因此在线的SLAM问题又可写为：

$$P(\boldsymbol{x}_k|\boldsymbol{z}_{1:k},\boldsymbol{x}_0)=\int\int\ldots\int P(\boldsymbol{x}_{1:k}|\boldsymbol{z}_{1:k},\boldsymbol{x}_0)d\boldsymbol{x}_1d\boldsymbol{x}_2\ldots d\boldsymbol{x}_{k-1}$$

  - 其中边缘化(Marginalization，概率密度积分)在每一时刻迭代进行

- 在马尔可夫假设前提下，后验概率可以用迭代贝叶斯滤波方法求解

贝叶斯滤波

- 马尔可夫假设： $P(\boldsymbol{x}_k|\boldsymbol{x}_{1:k-1}) = P(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$
- 我们可以使用迭代贝滤波（也称为迭代贝叶斯估算）估算每一时刻的后验概率： $P(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}, \boldsymbol{x}_0)$
- 根据贝叶斯公式

$$P(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}, \boldsymbol{x}_0) = \frac{P(\boldsymbol{z}_k, \boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}, \boldsymbol{x}_0)}{\underbrace{\int_{\boldsymbol{x}_k} P(\boldsymbol{z}_k, \boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}, \boldsymbol{x}_0) d\boldsymbol{x}_k}_{\text{归一化常量}}}$$

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

$$= P(\boldsymbol{z}_k|\boldsymbol{x}_k)\boxed{P(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}, \boldsymbol{x}_0)} \rightarrow \int_{\boldsymbol{x}_{k-1}} P(\mathbf{x}_k, \mathbf{x}_{k-1}|\boldsymbol{z}_{1:k-1}, \boldsymbol{x}_0) d\boldsymbol{x}_{k-1}$$

$$= \underbrace{\boxed{P(\boldsymbol{z}_k|\boldsymbol{x}_k)}}_{\text{观测模型}} \int_{\boldsymbol{x}_{k-1}} \underbrace{\boxed{P(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})}}_{\text{预测模型}} \underbrace{\boxed{P(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1}, \boldsymbol{x}_0)}}_{\text{上一时刻状态分布}} d\boldsymbol{x}_{k-1}$$

贝叶斯滤波

- 第一步预测 - 从上一时刻状态分布预测当前时刻状态分布

$$P(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1},\boldsymbol{x}_0)=\int_{\boldsymbol{x}_{k-1}}P(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})P(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1},\boldsymbol{x}_0)d\boldsymbol{x}_{k-1}$$

- 第二步更新 - 使用当前时刻的观测量更新后验概率

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

- 反复迭代实现在线估算
- 卡尔曼滤波是贝叶斯滤波的特例：
  - 状态分布、预测模型、观测模型的概率密度分布函数均为高斯

小结

- 贝叶斯公式 $P(X,Y)=P(X|Y)P(Y)$
- 先验、后验、似然
- SLAM问题的贝叶斯推断建模 $P(\boldsymbol{x}_{1:k}|\boldsymbol{z}_{1:k},\boldsymbol{x}_0)$
- 贝叶斯滤波 - 基于马尔可夫假设的迭代求解方法

$$P(\boldsymbol{x}_k|\boldsymbol{z}_{1:k},\boldsymbol{x}_0)$$
$$=P(\boldsymbol{z}_k|\boldsymbol{x}_k)\int_{\boldsymbol{x}_{k-1}}P(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})P(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1},\boldsymbol{x}_0)d\boldsymbol{x}_{k-1}$$

- 卡尔曼滤波是贝叶斯滤波的特例

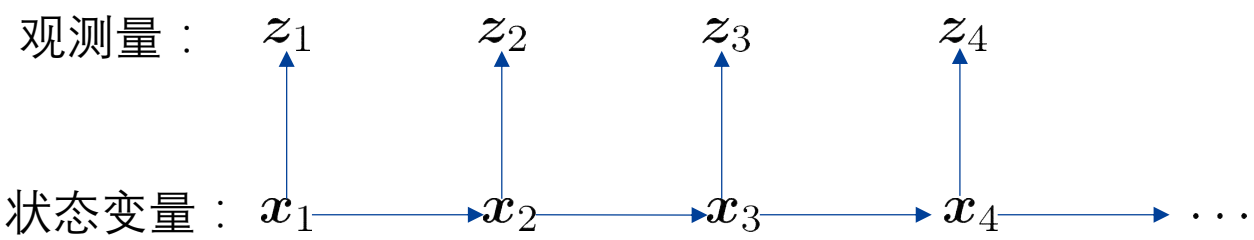
卡尔曼滤波

- 先介绍线性高斯马尔可夫模型（Linear Gauss-Markov Model)

预测模型：$\boldsymbol{x}_k = \mathbf{F}_k \boldsymbol{x}_{k-1} + \boldsymbol{n}_k \qquad \boldsymbol{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$

观测模型：$\boldsymbol{z}_k = \mathbf{H}_k \boldsymbol{x}_k + \boldsymbol{r}_k \qquad \boldsymbol{r}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$

- 其中，$\mathbf{F}_k$ 称为转移矩阵(transition matrix)，$\mathbf{H}_k$ 称为观测矩阵(Measurement matrix)



卡尔曼滤波

- 基于线性高斯马尔可夫模型，贝叶斯滤波框架中各概率函数都是高斯密度函数。
- 状态分布： $P(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1},\boldsymbol{x}_0)=\mathcal{N}(\boldsymbol{\mu}_{k-1},\boldsymbol{\Sigma}_{k-1})$
- 预测模型： $\boldsymbol{x}_k=\mathbf{F}_k\boldsymbol{x}_{k-1}+\boldsymbol{n}_k \qquad \boldsymbol{n}_k\sim\mathcal{N}(\mathbf{0},\mathbf{Q}_k)$

$$\Downarrow$$

$$P(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})=\mathcal{N}(\mathbf{F}_k\boldsymbol{x}_{k-1},\mathbf{Q}_k)$$

- 观测模型： $\boldsymbol{z}_k=\mathbf{H}_k\boldsymbol{x}_k+\boldsymbol{r}_k \qquad \boldsymbol{r}_k\sim\mathcal{N}(\mathbf{0},\mathbf{R}_k)$

$$\Downarrow$$

$$P(\boldsymbol{z}_k|\boldsymbol{x}_k)=\mathcal{N}(\mathbf{H}_k\boldsymbol{x}_k,\mathbf{R}_k)$$

卡尔曼滤波

- **预测**：$P(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1},\boldsymbol{x}_0)=\int_{\boldsymbol{x}_{k-1}}P(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})P(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1},\boldsymbol{x}_0)d\boldsymbol{x}_{k-1}$
- 第一步，从条件概率与先验概率组合联合概率

$$P(\boldsymbol{x}_k,\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1},\boldsymbol{x}_0)=\mathcal{N}(\mathbf{F}_k\boldsymbol{x}_{k-1},\mathbf{Q}_k)\mathcal{N}(\boldsymbol{\mu}_{k-1},\mathbf{R}_{k-1})$$

根据组合公式

$$\mathcal{N}\left(\begin{bmatrix}\mathbf{F}_k\boldsymbol{\mu}_{k-1}\\ \boldsymbol{\mu}_{k-1}\end{bmatrix},\begin{bmatrix}\mathbf{F}_k\boldsymbol{\Sigma}_{k-1}\mathbf{F}_k^T+\mathbf{Q}_k & \mathbf{F}_k\boldsymbol{\Sigma}_{k-1}\\ \boldsymbol{\Sigma}_{k-1}\mathbf{F}_k^T & \boldsymbol{\Sigma}_{k-1}\end{bmatrix}\right)$$

- 第二步，将 $\boldsymbol{x}_{k-1}$ 边缘化

$$P(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1},\boldsymbol{x}_0)=\mathcal{N}\left(\begin{bmatrix}\mathbf{F}_k\boldsymbol{\mu}_{k-1}\\ \boldsymbol{\mu}_{k-1}\end{bmatrix},\begin{bmatrix}\mathbf{F}_k\boldsymbol{\Sigma}_{k-1}\mathbf{F}_k^T+\mathbf{Q}_k & \mathbf{F}_k\boldsymbol{\Sigma}_{k-1}\\ \boldsymbol{\Sigma}_{k-1}\mathbf{F}_k^T & \boldsymbol{\Sigma}_{k-1}\end{bmatrix}\right)$$

$$=\mathcal{N}(\mathbf{F}_k\boldsymbol{\mu}_{k-1},\mathbf{F}_k\boldsymbol{\Sigma}_{k-1}\mathbf{F}_k^T+\mathbf{Q}_k)$$

卡尔曼滤波

- 另外一方面，我们可从高斯随机变量的线性组合理解预测过程：

$$\boldsymbol{x}_{k-1} \sim \mathcal{N}(\boldsymbol{\mu}_{k-1}, \boldsymbol{\Sigma}_{k-1})\,; \qquad \boldsymbol{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$$

$$\boldsymbol{x}_k = \mathbf{F}_k \boldsymbol{x}_{k-1} + \boldsymbol{n}_k$$

$$\hat{\boldsymbol{x}}_k \sim \mathcal{N}(\hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k)$$

$$\hat{\boldsymbol{\mu}}_k = E[\boldsymbol{x}_k] = \mathbf{F}_k \boldsymbol{\mu}_{\boldsymbol{k-1}}$$

$$\hat{\boldsymbol{\Sigma}}_k = cov[\boldsymbol{x}_k] = \mathbf{F}_k \boldsymbol{\Sigma}_{k-1} \mathbf{F}_k^T + \mathbf{Q}_k$$

卡尔曼滤波

- **更新**： $P(\boldsymbol{x}_k|\boldsymbol{z}_{1:k},\boldsymbol{x}_0) \propto P(\boldsymbol{z}_k|\boldsymbol{x}_k)P(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1},\boldsymbol{x}_0)$
- 第一步：从观测似然与预测先验组合联合概率分布

$$P(\boldsymbol{x}_k,\boldsymbol{z}_k|\boldsymbol{z}_{1:k-1},\boldsymbol{x}_0) = \mathcal{N}(\mathbf{H}_k\boldsymbol{x}_k,\mathbf{R}_k)\,\mathcal{N}(\underbrace{\mathbf{F}_k\boldsymbol{\mu}_{k-1}}_{\hat{\mu}_k},\underbrace{\mathbf{F}_k\boldsymbol{\Sigma}_{k-1}\mathbf{F}_k^T+\mathbf{Q}_k}_{\hat{\boldsymbol{\Sigma}}_k})$$

根据组合公式

$$\mathcal{N}\left(\begin{bmatrix}\hat{\boldsymbol{\mu}}_k\\ \mathbf{H}_k\hat{\boldsymbol{\mu}}_k\end{bmatrix},\begin{bmatrix}\hat{\boldsymbol{\Sigma}}_k & \boldsymbol{\Sigma}_k\mathbf{H}_k^T\\ \mathbf{H}_k\hat{\boldsymbol{\Sigma}}_k & \mathbf{H}_k\hat{\boldsymbol{\Sigma}}_k\mathbf{H}_k^T+\mathbf{R}_k\end{bmatrix}\right)$$

- 第二部：从联合分布中分拆后验概率

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

卡尔曼滤波

- 分拆后验概率 $P(\boldsymbol{x}_k, \boldsymbol{z}_k | \boldsymbol{z}_{1:k-1}, \boldsymbol{x}_0) = P(\boldsymbol{x}_k | \boldsymbol{z}_{1:k}, \boldsymbol{x}_0) P(\boldsymbol{z}_0 | \boldsymbol{z}_{1:k-1}, \boldsymbol{x}_0)$

$$\begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{z}_k \end{bmatrix} \sim P(\boldsymbol{x}_k, \boldsymbol{z}_k | \boldsymbol{z}_{1:k-1}, \boldsymbol{x}_0) = \mathcal{N}\left(\begin{bmatrix} \hat{\boldsymbol{\mu}}_k \\ \mathbf{H}_k \hat{\boldsymbol{\mu}}_k \end{bmatrix}, \begin{bmatrix} \hat{\boldsymbol{\Sigma}}_k & \boldsymbol{\Sigma}_k \mathbf{H}_k^T \\ \mathbf{H}_k \hat{\boldsymbol{\Sigma}}_k & \mathbf{H}_k \hat{\boldsymbol{\Sigma}}_k \mathbf{H}_k^T + \mathbf{R}_k \end{bmatrix} \right)$$

- 令 $\mathbf{S}_k = \mathbf{H}_k \hat{\boldsymbol{\Sigma}}_k \mathbf{H}_k^T + \mathbf{R}_k$， 后验概率如下：

根据分拆公式

$$P(\boldsymbol{x}_k | \boldsymbol{z}_{1:k}, \boldsymbol{x}_0) = \mathcal{N}(\hat{\mu}_k + \boxed{\hat{\boldsymbol{\Sigma}}_k \mathbf{H}_k^T \hat{\mathbf{S}}_k^{-1}} (\mathbf{z}_k - \mathbf{H}_k \hat{\boldsymbol{\mu}}_k), \ \hat{\boldsymbol{\Sigma}}_k - \boxed{\hat{\boldsymbol{\Sigma}}_k \mathbf{H}_k^T \mathbf{S}_k^{-1}} \mathbf{H}_k \hat{\boldsymbol{\Sigma}}_k)$$

$$\mathcal{N}(\hat{\mu}_k + \mathbf{G}_k \Delta \boldsymbol{z}_k, (\boldsymbol{I} - \mathbf{G}_k \mathbf{H}_k) \hat{\boldsymbol{\Sigma}}_k)$$

$\mathbf{G}_k$ - 卡尔曼增益

$\Delta \boldsymbol{z}_k$ - 观测残差

卡尔曼滤波整体流程

预测阶段

1. 状态预测：

$$\hat{\boldsymbol{\mu}}_k = \mathbf{F}_k \boldsymbol{\mu}_{\boldsymbol{k-1}}$$

2. 误差传递：

$$\hat{\boldsymbol{\Sigma}}_k = \mathbf{F}_k \boldsymbol{\Sigma}_{k-1} \mathbf{F}_k^T + \mathbf{Q}_k$$

更新阶段

1. 计算卡尔曼增益：

$$\mathbf{G}_k = \hat{\boldsymbol{\Sigma}}_k \mathbf{H}_k^T (\mathbf{H}_k \hat{\boldsymbol{\Sigma}}_k \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$$

2. 计算观测残差：

$$\Delta \boldsymbol{z}_k = (\mathbf{z}_k - \mathbf{H}_k \hat{\boldsymbol{\mu}}_k)$$

3. 状态更新:

$$\boldsymbol{\mu}_k = \hat{\boldsymbol{\mu}}_k + \mathbf{G}_k \Delta \boldsymbol{z}_k$$

4. 误差更新:

$$\boldsymbol{\Sigma}_k = (\boldsymbol{I} - \mathbf{G}_k \mathbf{H}_k) \hat{\boldsymbol{\Sigma}}_k$$

$\boldsymbol{x}_0, \boldsymbol{\Sigma}_0$

初始状态

卡尔曼滤波整体流程

- 应用卡尔曼滤波器只需关心：
  - 转移矩阵：$\mathbf{F}_k$
  - 观测矩阵：$\mathbf{H}_k$
- 所以说推导和理解起来挺麻烦，但用起来很简单

非线性情况-扩展卡尔曼滤波

- 非线性高斯马尔可夫模型

预测模型： $\boldsymbol{x}_k = \mathbf{f}(\boldsymbol{x}_{k-1}) + \boldsymbol{n}_k \qquad \boldsymbol{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$

观测模型： $\boldsymbol{z}_k = \mathbf{h}(\boldsymbol{x}_k) + \boldsymbol{r}_k \qquad \boldsymbol{r}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$

预测过程

$$\hat{\boldsymbol{x}}_k = \mathbf{f}(\boldsymbol{x}_{k-1})$$

$$\hat{\boldsymbol{\Sigma}}_k = \mathbf{F}_k \boldsymbol{\Sigma}_{k-1} \mathbf{F}_k^T + \mathbf{Q}_k$$

$$\mathbf{F}_k = \frac{\partial \mathbf{f}}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}_k}$$

观测过程

$$\Delta \boldsymbol{z}_k = \mathbf{z}_k - \mathbf{h}(\hat{\boldsymbol{\mu}}_k)$$

$$\mathbf{S}_k = \mathbf{H}_k \hat{\boldsymbol{\Sigma}}_k \mathbf{H}_k^T + \mathbf{R}_k$$

$$\mathbf{H}_k = \frac{\partial \mathbf{h}}{\partial \boldsymbol{x}}\Big|_{\hat{\mathbf{x}}_k}$$

小结

- 线性高斯马尔可夫模型
- 预测步骤的推导:

$$P(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1},\boldsymbol{x}_0) = \int_{\boldsymbol{x}_{k-1}} P(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})P(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1},\boldsymbol{x}_0)d\boldsymbol{x}_{k-1}$$

$$\boxed{\hat{\boldsymbol{\mu}}_k = \mathbf{F}_k\boldsymbol{\mu}_{\boldsymbol{k-1}},\ \hat{\boldsymbol{\Sigma}}_k = \mathbf{F}_k\boldsymbol{\Sigma}_{k-1}\mathbf{F}_k^T + \mathbf{Q}_k}$$

- 更新步骤的推导:

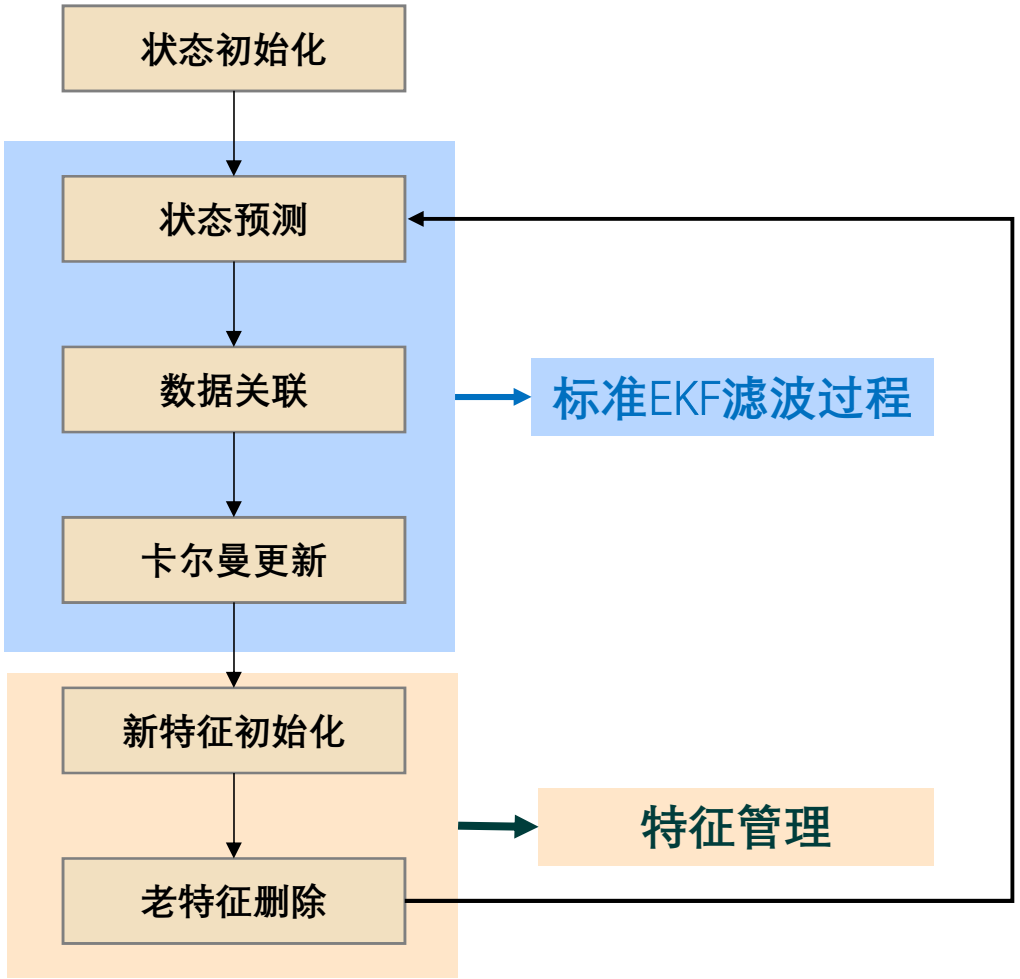
$$P(\boldsymbol{x}_k|\boldsymbol{z}_{1:k},\boldsymbol{x}_0) \propto P(\boldsymbol{z}_k|\boldsymbol{x}_k)P(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1},\boldsymbol{x}_0)$$

$$\boxed{\boldsymbol{\mu}_k = \hat{\boldsymbol{\mu}}_k + \mathbf{G}_k\Delta\boldsymbol{z}_k,\ \boldsymbol{\Sigma}_k = (\boldsymbol{I} - \mathbf{G}_k\mathbf{H}_k)\hat{\boldsymbol{\Sigma}}_k}$$

- 高斯分布的拆分、组合公式在推导过程中起到关键作用
- 非线性-扩展卡尔曼滤波

基于扩展卡尔曼滤波的视觉SLAM

- 状态变量设计
- 状态初始化
- 观测模型
- 数据关联
- 预测模型
- 特征管理-初始化,边缘化



EKF视觉SLAM

- MonoSLAM, Andrew Davison, 2003
  - 第一个能实时跑的视觉SLAM系统
- ROVIO - 使用反深度+直接法 的视觉惯性组合SLAM系统
- 以下介绍内容基于 Joan Sola 的EKF-SLAM教程

  Simultaneous localization and mapping with the extended Kalman filter， A very quick guide.

- 有兴趣同学可以仔细阅读该教程，并下载代码尝试运行

EKF视觉SLAM

- 状态变量定义 – 由两部分组成：
  - $\mathcal{R}$ 相机自身的运行信息，如位置、姿态、运动速度等
  - $\mathcal{M}$ 场景中的三维特征点信息

$$\mathbf{x} = \begin{bmatrix} \mathcal{R} \\ \mathcal{M} \end{bmatrix} = \begin{bmatrix} \mathcal{R} \\ \mathcal{L}_1 \\ \vdots \\ \mathcal{L}_n \end{bmatrix} \quad \mathbf{P} = \begin{bmatrix} \mathbf{P}_{\mathcal{R}\mathcal{R}} & \mathbf{P}_{\mathcal{R}\mathcal{M}} \\ \mathbf{P}_{\mathcal{M}\mathcal{R}} & \mathbf{P}_{\mathcal{M}\mathcal{M}} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{\mathcal{R}\mathcal{R}} & \mathbf{P}_{\mathcal{R}\mathcal{L}_1} & \cdots & \mathbf{P}_{\mathcal{R}\mathcal{L}_n} \\ \mathbf{P}_{\mathcal{L}_1\mathcal{R}} & \mathbf{P}_{\mathcal{L}_1\mathcal{L}_1} & \cdots & \mathbf{P}_{\mathcal{L}_1\mathcal{L}_n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{\mathcal{L}_n\mathcal{R}} & \mathbf{P}_{\mathcal{L}_n\mathcal{L}_1} & \cdots & \mathbf{P}_{\mathcal{L}_n\mathcal{L}_n} \end{bmatrix}$$

EKF视觉SLAM

- 观测量定义 – 图像上所检测的特征点
- 观测过程 - 透视投影 ： $\mathbf{y} = \boldsymbol{h}(\mathcal{R}, \mathcal{L}) + \mathbf{v}$
  - 线性化之后的观测模型

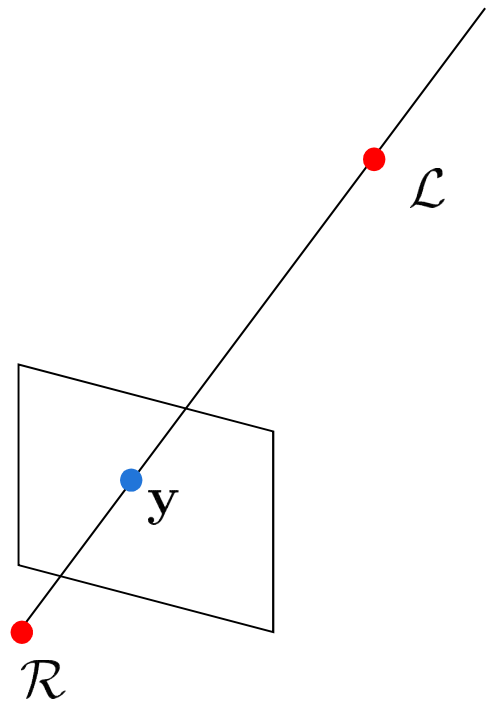
$$\Delta \boldsymbol{z} = \mathbf{H}_{\mathcal{R}} \delta \mathcal{R} + \mathbf{H}_{\mathcal{L}} \delta \mathcal{L}$$

- 对于第i个特征点，对应的观测矩阵为:

$$\mathbf{H}_i = [\mathbf{H}_{\mathcal{R}}^{(i)} \quad 0 \quad \cdots \quad 0 \quad \mathbf{H}_{\mathcal{L}}^{(i)} \quad 0 \quad \cdots \quad 0]$$

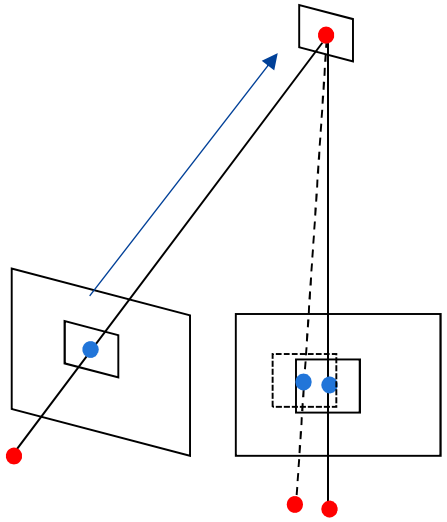
- 对所有特征点的整体观测矩阵为：

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \\ \vdots \\ \mathbf{H}_N \end{bmatrix}$$



EKF视觉SLAM

- 直接法观测量 (ROVIO,IROS 2015)， 直接对比像素灰度值



$\mathbf{P}$:初始图像块

$\mathbf{I}(\boldsymbol{h}(\mathcal{R},\mathcal{L}))$:预测投影点图像块

像素灰度差 (Photometric error):

$$\Delta\mathbf{I}(\boldsymbol{x}) = \mathbf{P}(\boldsymbol{x}) - \mathbf{I}(\boldsymbol{h}(\mathcal{R},\mathcal{L}) + \boldsymbol{x})$$

$$(\boldsymbol{x} \in \boldsymbol{\Omega})$$

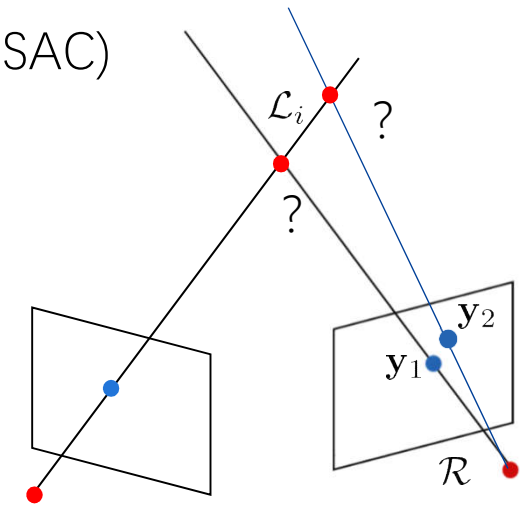
线性化之后的观测方程为：

$$\Delta\boldsymbol{z} = \nabla\mathbf{I}^{\mathrm{T}}\mathbf{H}_{\mathcal{R}}\delta\mathcal{R} + \nabla\mathbf{I}^{\mathrm{T}}\mathbf{H}_{\mathcal{L}}\delta\mathcal{L}$$

梯度： $\nabla\mathbf{I} = \begin{bmatrix}\mathbf{I}_x \\ \mathbf{I}_y\end{bmatrix}$

EKF视觉SLAM

- 数据关联 – 找出观测量与特征点变量的一一对应关系，类似特征匹配。
- 数据关联不可避免存在外点(outlier)，如何处理？
- 1）通过 χ^2 测试 – 使用预测状态与协方差将三维特征投影到图像平面，检查特征点是否在预测范围内。
- 2）通过随机抽样验证法(RANSAC)，使用单点更新整个状态，检查更新完毕所有观测量与状态的一致性。(1point-RANSAC)



EKF视觉SLAM

- 预测模型 – 状态量预测包括相机运动信息的预测，与特征点三维坐标的预测。

$$\bar{\mathbf{x}} \leftarrow f(\bar{\mathbf{x}}, 0)$$
$$\mathbf{P} \leftarrow \mathbf{F_x P F_x^\top} + \mathbf{F_n N F_n^\top}$$

- $\bar{\mathbf{x}}$ 为状态分布均值
- $\mathbf{P}$ 为协方差矩阵
- $\mathbf{N}$ 为预测噪声

- 三维坐标保持静止，状态预测函数定义为

$$\mathcal{R} \leftarrow f_{\mathcal{R}}(\mathcal{R}, \mathbf{n})$$
$$\mathcal{M} \leftarrow \mathcal{M}$$

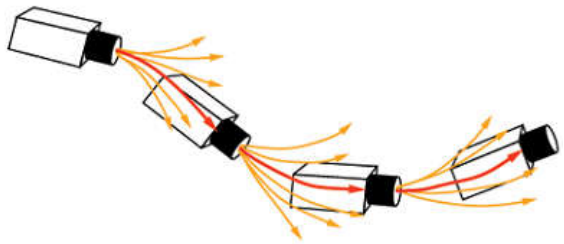
- 状态转移矩阵和噪声传递矩阵分别定义为

$$\mathbf{F_x} = \begin{bmatrix} \frac{\partial f_{\mathcal{R}}}{\partial \mathcal{R}} & 0 \\ 0 & \mathbf{I} \end{bmatrix} \qquad \mathbf{F_n} = \begin{bmatrix} \frac{\partial f_{\mathcal{R}}}{\partial \mathbf{n}} \\ 0 \end{bmatrix}$$

EKF视觉SLAM

- 预测模型实例
  - 匀速模型 (MonoSLAM)

$$\mathcal{R} = \begin{bmatrix} \mathbf{q} \\ \mathbf{p} \\ \mathbf{v} \\ \omega \end{bmatrix} \leftarrow \begin{bmatrix} \mathbf{q} \oplus \mathbf{q}\{\omega \Delta t\} \\ \mathbf{p} + \mathbf{v}\Delta t \\ \mathbf{v} \\ \omega \end{bmatrix}$$



  - 利用IMU积分进行预测

$$\mathcal{R} = \begin{bmatrix} \mathbf{q} \\ \mathbf{p} \\ \mathbf{v} \end{bmatrix} \leftarrow \begin{bmatrix} \mathbf{q} \oplus \mathbf{q}\{\omega \Delta t\} \\ \mathbf{p} + \mathbf{v}\Delta t \\ \mathbf{v} + (\mathbf{R}_q \mathbf{a} + \mathbf{g})\Delta t \end{bmatrix}$$

  - 融合里程计(Odometer)

EKF视觉SLAM

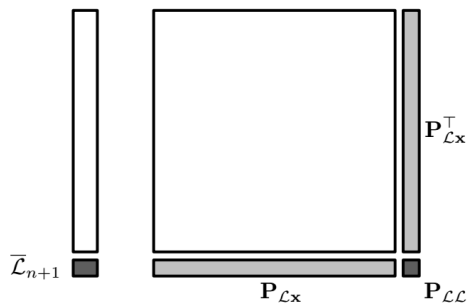
- 特征管理 - 新特征初始化：
  - 1. 将新特征点的初始状态加入滤波器状态变量中
  - 2. 将对应协方差与互相关方差插入协方差矩阵中
  - 新特征初始状态计算如下：

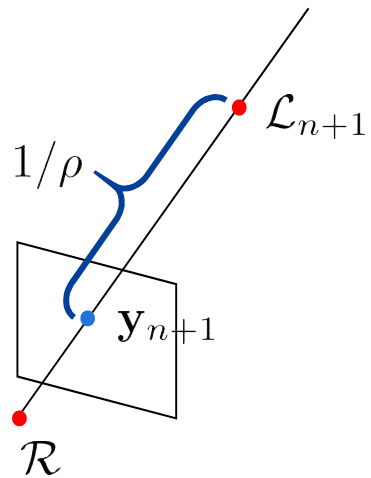
$$\mathcal{L}_{n+1} = g(\mathcal{R}, \mathbf{y}_{n+1}, \rho)$$

    - ρ 为初始反深度(inverse-depth)
    - $g(\cdot)$ 为初始化函数
  - 协方差(co-variance) 与互协方差(cross-variance)

$$\mathbf{P}_{\mathcal{L}_{n+1}\mathcal{L}_{n+1}} = (\frac{\partial g}{\partial \mathcal{R}})\mathbf{P}_{\mathcal{R}\mathcal{R}}(\frac{\partial g}{\partial \mathcal{R}})^T + (\frac{\partial g}{\partial \mathbf{y}_{n+1}})\mathbf{R}(\frac{\partial g}{\partial \mathbf{y}_{n+1}})^T$$

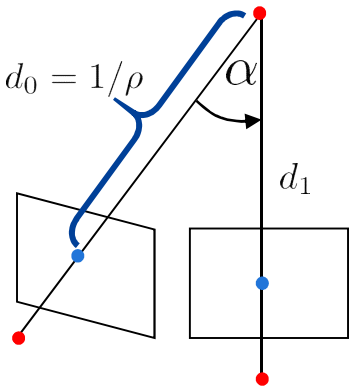
$$\mathbf{P}_{\mathcal{L}\mathbf{x}} = (\frac{\partial g}{\partial \mathcal{R}})\mathbf{P}_{\mathcal{R}\mathbf{x}} = (\frac{\partial g}{\partial \mathcal{R}})[\mathbf{P}_{\mathcal{R}\mathcal{R}}\ \mathbf{P}_{\mathcal{R}\mathcal{M}}]$$





EKF视觉SLAM

- 有关特征点的反深度表示方法：
  - 特征点用反深度表示可实现即时初始化（无需观察若干几帧通过三角化进行初始化）
- 反深度表示有着更好的线性特性（Linearity),尤其是在初始化阶段



笛卡尔坐标表示： $\boldsymbol{y}_t = \boldsymbol{h}(\mathcal{R}_t, \mathcal{L})$

反深度表示： $\boldsymbol{y}'_t = \boldsymbol{h}(\mathcal{R}_t, \boldsymbol{g}(\mathcal{R}_{t-1}, \boldsymbol{y}_t, \rho))$

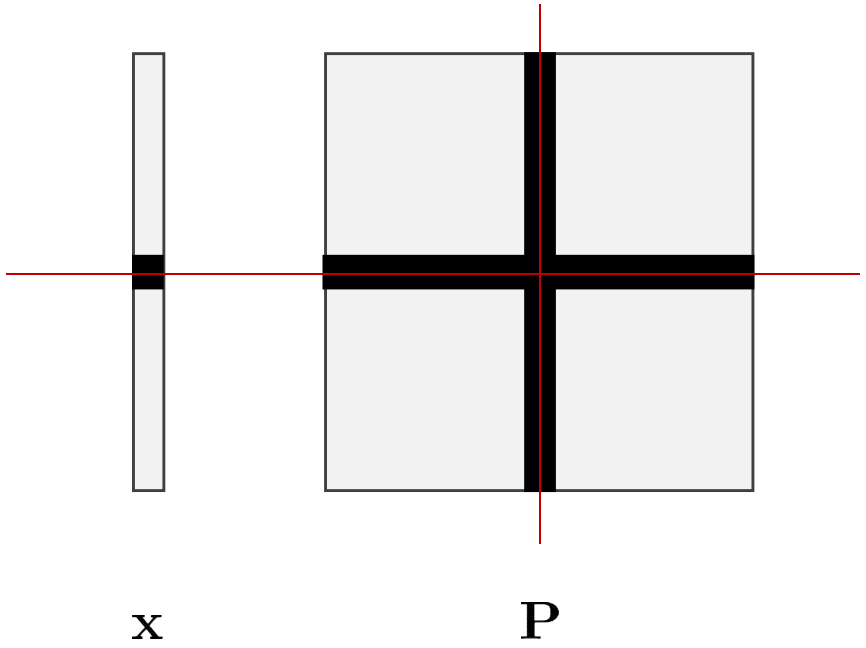
线性程度可用二阶导来衡量：

$$\alpha \to 0, d_0 \approx d_1 \Rightarrow \frac{\partial^2 y'}{\partial^2 \rho} \to 0$$

Civera, Javier, Andrew J. Davison, and JM Martinez Montiel. "Inverse depth parametrization for monocular SLAM." *IEEE transactions on robotics* 24.5 (2008): 932-945.

EKF视觉SLAM

- 特征管理 – 特征点不断加入，导致状态变量的维度越来越高。
- 解决方案：将老的特征点从状态变量中删除。其实质就是一个边缘化操作。



小结

- EKF-SLAM 状态变量定义 $\mathbf{x} = [\mathcal{R}\,\mathcal{M}]^T$
- 观测模型：透视投影 $\mathbf{y}_i = h_i(\mathcal{R}, \mathcal{L}_i) + \mathbf{v}$
- 数据关联与外点去除
  - χ^2 测试、单点RANSAC法
- 预测模型
  - 特征点静止
  - 相机运动模型 – 匀速模型、外界传感器辅助预测模型
- 特征管理
  - 新特征初始化 – 反深度表示
  - 老特征去除 - 边缘化

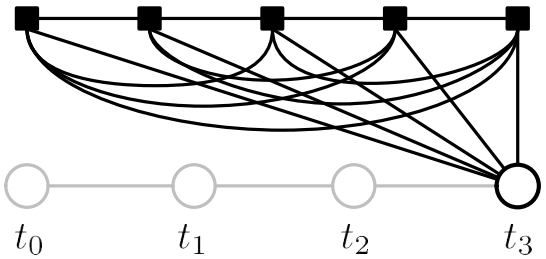
多状态约束滤波方法

- 经典EKF-SLAM方法策略
  - 将相机历史位姿信息立即进行边缘化，非线性因素将导致误差累积
  - 特征点较多时，计算量较大
- 多状态约束滤波方法 (MSCKF)
  - 将特征点估算与相机姿态估算解耦
  - 保留相机历史位姿信息，进行延时边缘化，降低非线性因素导致的误差
  - 滤波器状态变量中只包含相机位姿信息

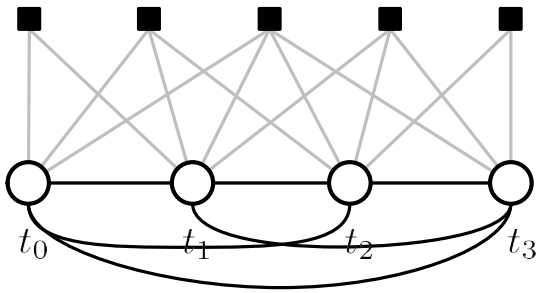
MSCKF论文： Mourikis A I, Roumeliotis S I. A multi-state constraint Kalman filter for vision-aided inertial navigation[C]//Robotics and automation, 2007 IEEE international conference on. IEEE, 2007: 3565-3572.

多状态约束滤波方法

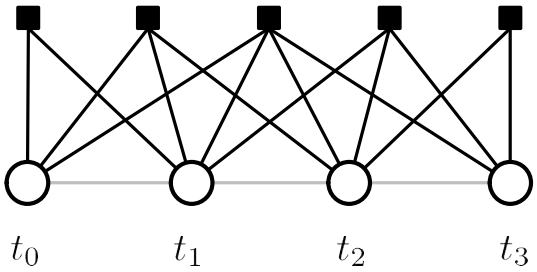
- EKF-SLAM方法、多状态约束方法和关键帧优化方法对比示意图，其中黑色方框表示特征点，圆圈表示在某时刻的相机状态



经典滤波方法



多状态滤波方法



关键帧优化方法

多状态约束

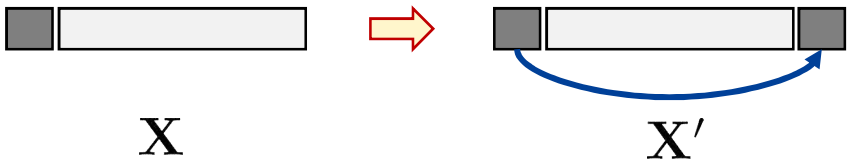
- 多状态约束的核心思想之一：状态克隆(State Clone)
- 多状态约束滤波状态定义如下：

$$\hat{\mathbf{X}}_k = \left[\hat{\mathbf{X}}_{\mathrm{IMU}_k}^T \quad {}^{C_1}_{G}\hat{\bar{q}}^T \quad {}^{G}\hat{\mathbf{p}}_{C_1}^T \quad \cdots \quad {}^{C_N}_{G}\hat{\bar{q}}^T \quad {}^{G}\hat{\mathbf{p}}_{C_N}^T \right]^T$$

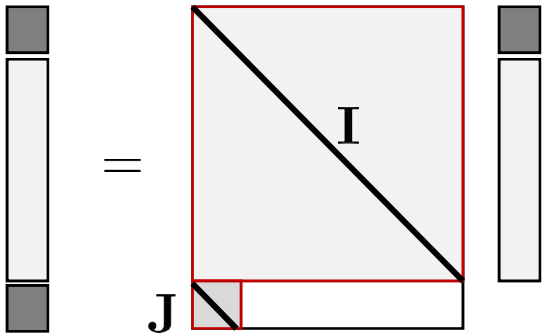
当前位姿信息　　历史位姿信息

多状态约束

- **状态克隆** (state clone) - 将当前的位姿信息加入到状态变量当中



- 该过程可用矩阵相乘表示



$$\mathbf{X}' \leftarrow \begin{bmatrix} \mathbf{I} & \\ [\mathbf{J} & 0] \end{bmatrix} \mathbf{X}$$

- 协方差矩阵插入公式： $\mathbf{P}' \leftarrow \begin{bmatrix} \mathbf{I} & \\ [\mathbf{J} & 0] \end{bmatrix} \mathbf{P} \begin{bmatrix} \mathbf{I} & \\ [\mathbf{J} & 0] \end{bmatrix}^T$

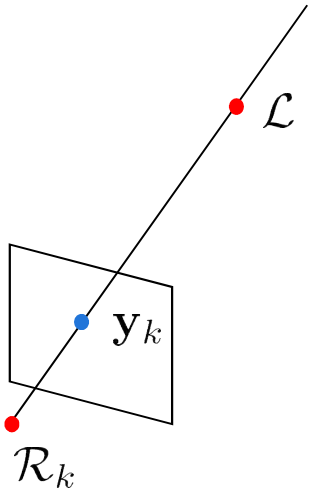
多状态约束

- 多状态约束的核心思想之二：**解耦** - 将观测模型从重投影误差变为位姿约束，消除滤波器对特征点三维坐标的依赖
- **重投影误差**观测模型：
  - k 时刻下特征 $\mathcal{L}$ 在图像中的投影：

$$\mathbf{y}_k = \boldsymbol{h}(\mathcal{R}_k, \mathcal{L})$$

  - 它的扩展卡尔曼观测模型为：

$$\Delta z_k = \underbrace{\mathbf{H}_{\mathcal{R}_k}}_{\frac{\partial \boldsymbol{h}}{\partial \mathcal{R}_k} \in \mathbb{R}^{2\times 6}} \delta\mathcal{R}_k + \underbrace{\mathbf{H}_{\mathcal{L}}}_{\frac{\partial \boldsymbol{h}}{\partial \mathcal{L}} \in \mathbb{R}^{2\times 3}} \delta\mathcal{L}$$



多状态约束

- 我们可以把多个时刻的观测方程写在一起

$$
\begin{aligned}
\Delta z_1 &= \mathbf{H}_{\mathcal{R}_1}\delta\mathcal{R}_1 + \mathbf{H}_{\mathcal{L}}^{(1)}\delta\mathcal{L} \\
\cdots \quad & \cdots \\
\Delta z_k &= \mathbf{H}_{\mathcal{R}_k}\delta\mathcal{R}_k + \mathbf{H}_{\mathcal{L}}^{(k)}\delta\mathcal{L} \\
\cdots \quad & \cdots \\
\Delta z_M &= \mathbf{H}_{\mathcal{R}_M}\delta\mathcal{R}_M + \mathbf{H}_{\mathcal{L}}^{(M)}\delta\mathcal{L}
\end{aligned}
$$

$$
\Rightarrow \quad \Delta \boldsymbol{z} = \underbrace{\mathbf{H}_{\mathcal{R}}}_{\mathbb{R}^{2M\times 6M}}\delta\mathcal{R} + \underbrace{\mathbf{H}_{\mathcal{L}}}_{\mathbb{R}^{2M\times 3}}\delta\mathcal{L}
$$

多状态约束

- 一条二维特征点轨迹的观测模型

$$\Delta \boldsymbol{z} = \underbrace{\mathbf{H}_{\mathcal{R}}}_{\mathbb{R}^{2M\times 6M}} \delta\mathcal{R} + \underbrace{\mathbf{H}_{\mathcal{L}}}_{\mathbb{R}^{2M\times 3}} \delta\mathcal{L}$$

- 为了消除特征点三位坐标的依赖，将观测模型投影到 $\mathbf{H}_{\mathcal{L}}$ 的左零空间 (2M-3维， 等价于$\mathbf{H}_{\mathcal{L}}^T$ 的零空间)。
- 所谓左零空间，就是如下齐次方程非零解的集合

$$\mathbf{H}_{\mathcal{L}}^T \mathbf{x} = \mathbf{0}$$

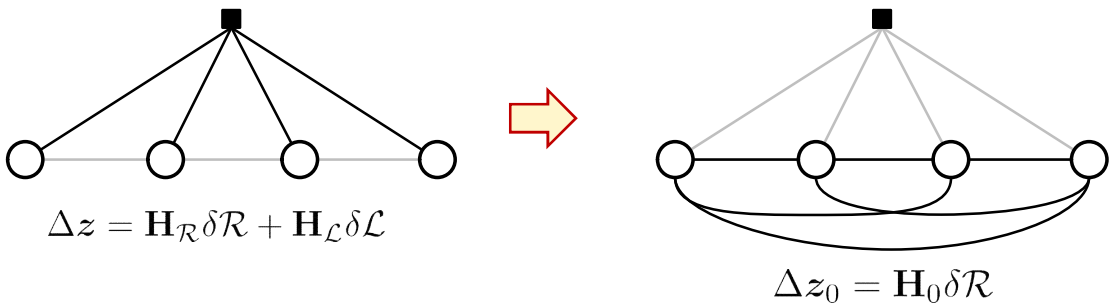
多状态约束

- 假设 $\mathbf{A}$ 的每一列由 $\mathbf{H}_{\mathcal{L}}$ 左零空间的基组成，将上面的式子两边同时乘的 $\mathbf{A}$ 转置，

$$\mathbf{A}^T \Delta \boldsymbol{z} = \mathbf{A}^T \mathbf{H}_{\mathcal{R}} \delta R \qquad (\mathbf{A} \in \mathbb{R}^{2M \times (2M-3)})$$

- 得到零空间下的一条特征点轨迹的观测误差模型：

$$\Delta \boldsymbol{z}_0 = \mathbf{H}_0 \delta \mathcal{R} \qquad (\mathbf{H}_0 \in \mathbb{R}^{(2M-3) \times (2M-3)})$$



多状态约束

- 使用N条特征点轨迹进行卡尔曼滤波更新

$$\left.\begin{aligned}\Delta \boldsymbol{z}_0^{(1)} &= \mathbf{H}_0^{(1)}\delta\mathcal{R}\\ \Delta \boldsymbol{z}_0^{(i)} &= \mathbf{H}_0^{(i)}\delta\mathcal{R}\\ &\cdots\ \cdots\\ \Delta \boldsymbol{z}_0^{(N)} &= \mathbf{H}_0^{(N)}\delta\mathcal{R}\end{aligned}\right\} \quad \Delta \boldsymbol{z}_0 = \mathbf{H}_0\delta\mathcal{R}$$

- 此时观测矩阵 $\mathbf{H}_0$ 列维度太高 N(2M-3)×(2M-3)，可通过QR分解，

$$\Delta \boldsymbol{z}_0 \rightarrow \begin{bmatrix}\Delta \boldsymbol{z}_0^a\\ \Delta \boldsymbol{z}_0^b\end{bmatrix} \qquad \mathbf{H}_0 \rightarrow \begin{bmatrix}\mathbf{Q}_a\\ \mathbf{Q}_b\end{bmatrix}\begin{bmatrix}\mathbf{H}_0^a\\ \mathbf{0}\end{bmatrix}$$

- 将观测方程进一步化简为 ：$\mathbf{Q}_a^T\Delta \boldsymbol{z}_0^a = \mathbf{H}_0^a\delta\mathcal{R} \rightarrow \Delta \boldsymbol{y}_0 = \mathbf{H}_0^a\delta\mathcal{R}$

多状态约束滤波算法整体流程

- **状态传递** – 使用状态预测模型预测下一时刻的状态和协方差
  - IMU积分、里程计积分、匀速模型
- **更新** – 在每帧图像中进行
  - 状态克隆 – 将当前相机姿态复制到历史状态中
  - 将消失的特征轨迹作为观测量，使用多状态约束方程进行卡尔曼更新
  - 当历史状态数目超过设定阈值，均匀选择1/3的历史状态进行更新（使用在所选历史状态可见的特征点轨迹作为观测量)，更新后删除
- **特征管理** – 在滤波器外
  - 特征点初始化
  - 特征点三维坐标更新 （三角化）

有关多状态约束滤波方法的性能

- **相比经典滤波方法 ：**
  - 特征点三维坐标解耦，将特征点状态估算排除在滤波器之外，降低了状态维度，提升了处理大量特征点的能力。
- **相比关键帧优化方法 ：**
  - 逐帧更新和边缘化，充分利用了持续时间较短的特征点轨迹（机会轨迹）
  - 可单条轨迹更新，增强了灵活性，方便控制计算峰值
  - 达到相近精度情况下，多状态约束方法计算成本较低。

小结

- 核心思想之一：**状态克隆**
- 核心思想之二：**解耦** - 重投影误差观测模型 -> 位姿约束观测模型
- 单特征轨迹观测模型推导
- 多轨迹观测模型推导 、QR分解降低观测方程维度
- 多状态滤波算法整体流程
  - 滤波器内 – 预测、更新
  - 滤波器外 - 特征点初始化、三角化
- 多状态约束滤波方法性能特点：精度高，速度快

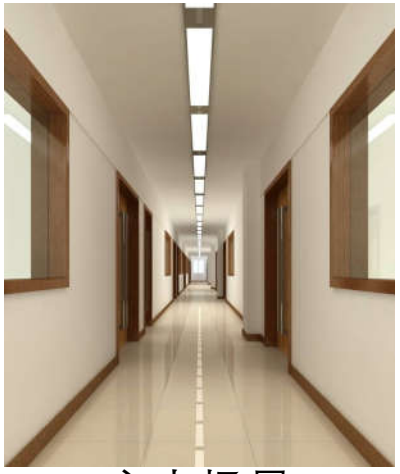
结构化环境下的视觉SLAM方法

- 不同于自然环境，人造环境中结构化特性明显

自然场景

街道场景

地下车库场景



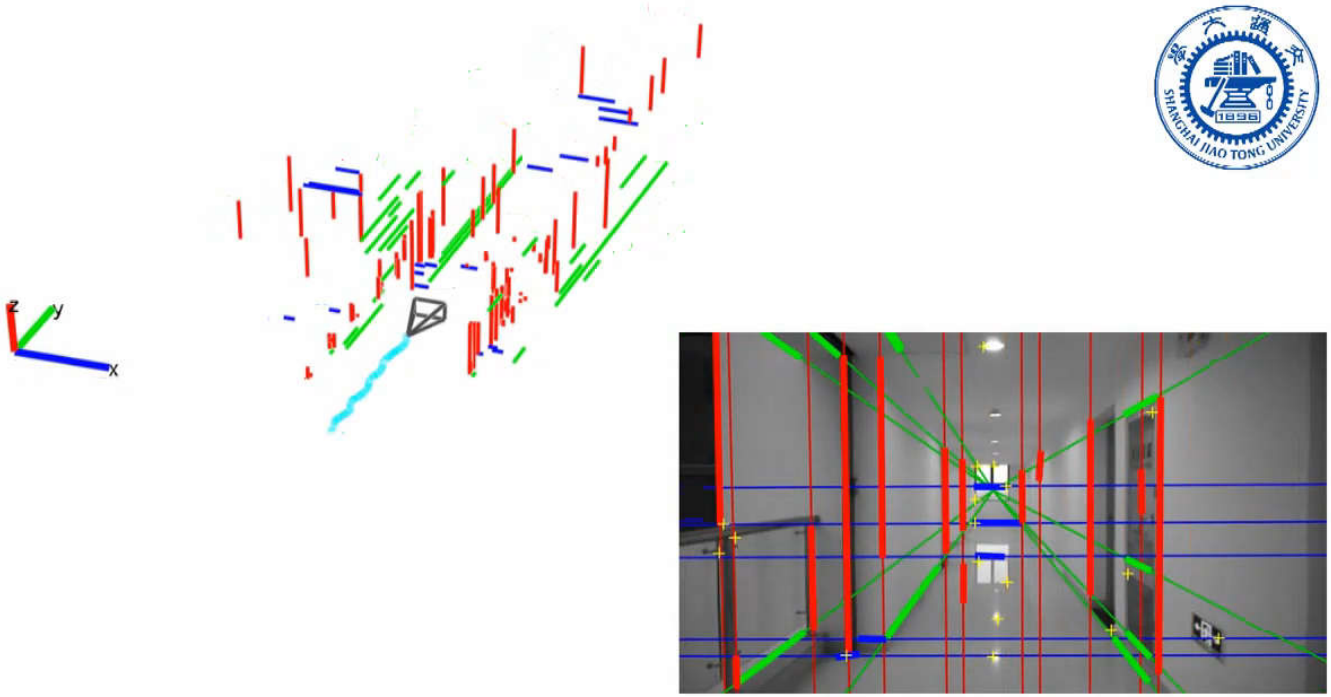
室内场景

结构化环境下的视觉SLAM方法

- 充分利用线条特征信息将有助于SLAM
  - 在一些纹理不够丰富的室内场景，线条特征比点特征更加可靠
  - 线条特征的方向性提升了位姿状态的可观测性(observability)

结构化环境下的视觉SLAM方法

- StructSLAM,2014



StructSLAM:Visual SLAM with Building Structure Lines

Zhou, Huizhong, Danping, Zou, et al. "StructSLAM: Visual SLAM with building structure lines." *Vehicular Technology, IEEE Transactions on* 64.4 (2015): 1364-1375. - Special session for indoor localization

结构化环境下的视觉SLAM方法

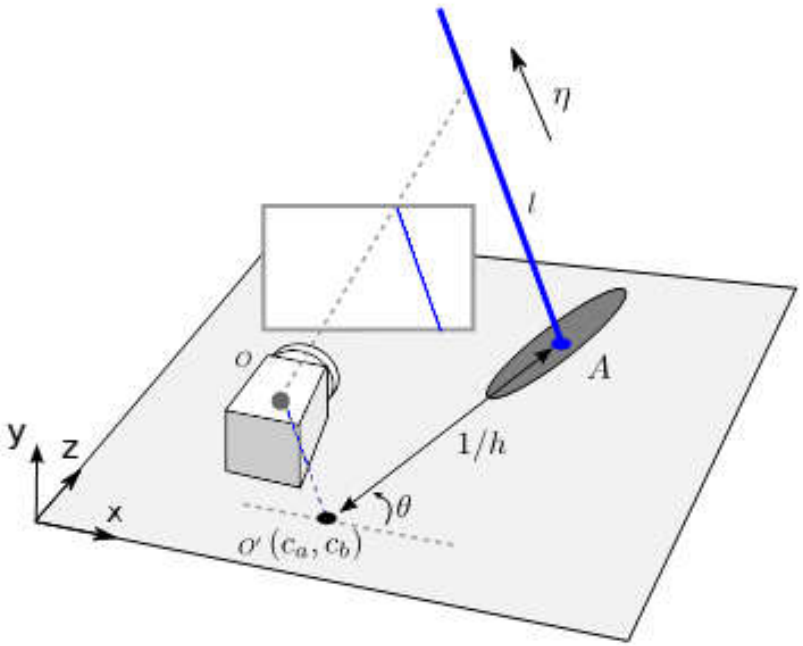
- StructSLAM 核心思想 - 结构线条特征
  - 与主方向一致的线条特征
  - 4维向量
  - 反深度

$$\mathbf{l} = \begin{pmatrix} c_a \\ c_b \\ \theta \\ h \end{pmatrix}$$

相机中心在主平面的投影

方向

反深度



结构化环境下的视觉SLAM方法

- StructSLAM采用的是经典卡尔曼滤波的方法，状态和协方差矩阵定义如下：

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_c \\ \mathbf{x}_p \\ \boxed{\mathbf{x}_l} \end{bmatrix} \qquad \mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma}_{cc} & \mathbf{\Sigma}_{cp} & \mathbf{\Sigma}_{cl} \\ \mathbf{\Sigma}_{pc} & \mathbf{\Sigma}_{pp} & \mathbf{\Sigma}_{pl} \\ \mathbf{\Sigma}_{lc} & \mathbf{\Sigma}_{lp} & \mathbf{\Sigma}_{ll} \end{bmatrix}$$

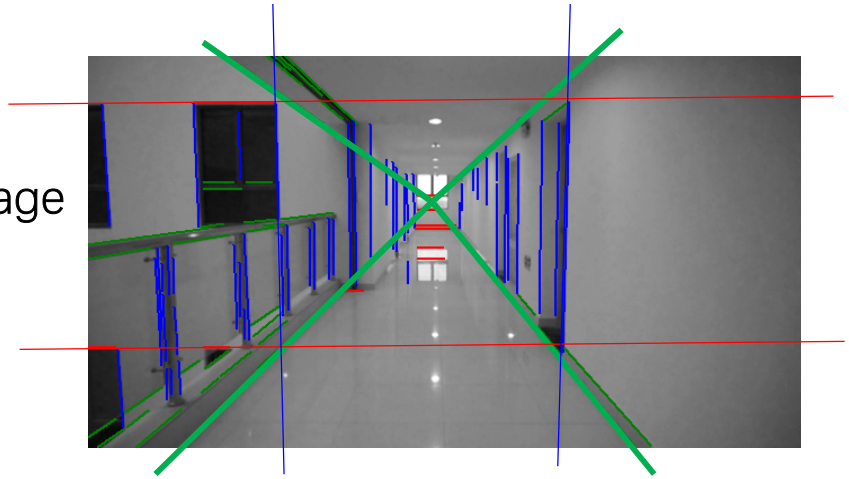
相机位姿+ 点特征 + 结构线条特征

$$\mathbf{x}_c = \begin{bmatrix} \mathbf{p}^w \\ \mathbf{q}^{wc} \\ \mathbf{v}^w \\ \omega^c \end{bmatrix} \qquad \mathbf{x}_p = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \vdots \end{bmatrix} \qquad \boxed{\mathbf{x}_p = \begin{bmatrix} \mathbf{l}_1 \\ \mathbf{l}_2 \\ \vdots \end{bmatrix}}$$

结构化环境下的视觉SLAM方法

- StructSLAM的初始化
  - 识别出至少两个投影灭点方向，采用 J-Linkage检测投影灭点

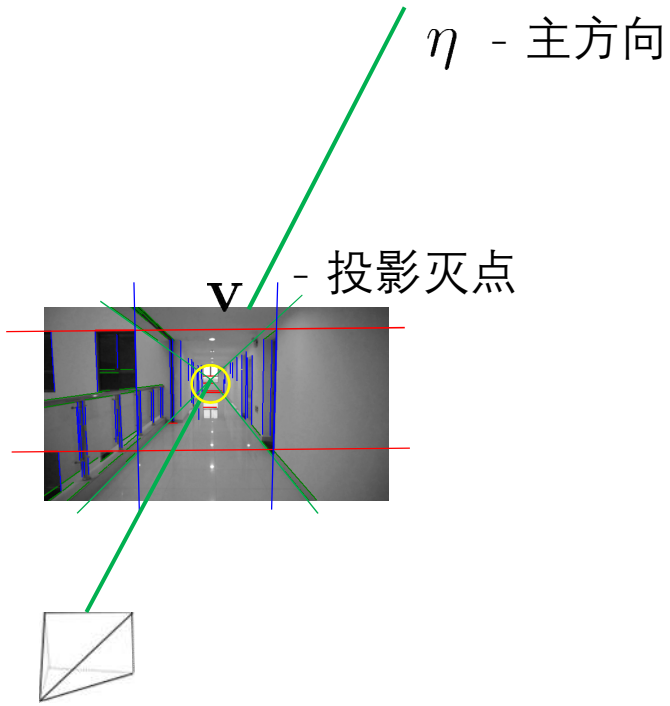
J-Linkage



*Toldo, Roberto, and Andrea Fusiello. "Robust multiple structures estimation with J-Linkage." *Computer Vision–ECCV 2008*. Springer Berlin Heidelberg, 2008. 537-547.

结构化环境下的视觉SLAM方法

- 从投影灭点中恢复主方向，在通过非线性最小二乘法进行优化



$$\mathbf{v} = \mathbf{K}\mathbf{R}^{cw}\eta$$

$$\eta \propto \mathbf{R}^{wc}\mathbf{K}^{-1}\mathbf{v}$$

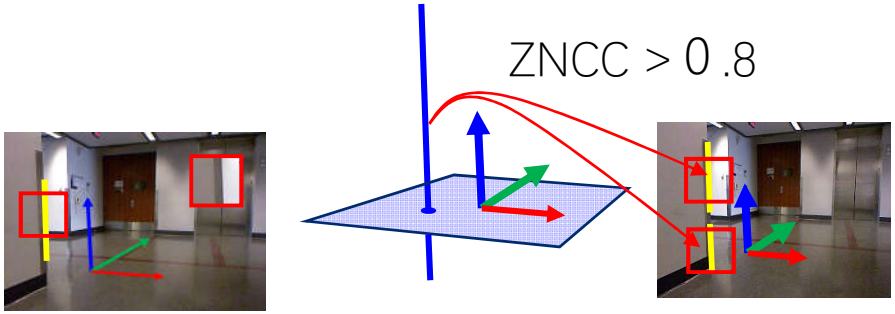
$\mathbf{K}$ ：相机内参

$\mathbf{R}^{cw}$ ：世界坐标系到相机坐标系的旋转变换

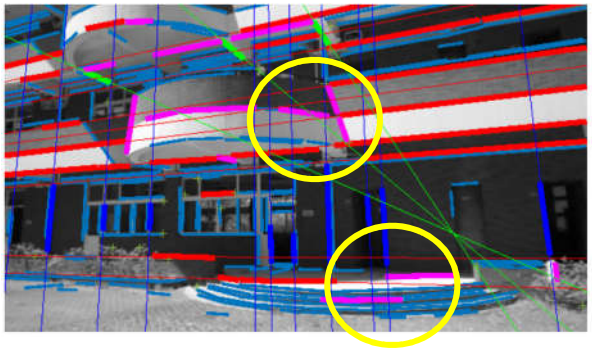
$$\mathbf{R}^{wc} = (\mathbf{R}^{cw})^T$$

结构化环境下的视觉SLAM方法

- **结构线条的跟踪**
  - 线条的跟踪比特征点跟踪要困难很多
    - 断裂成多条线段，一对多
    - 线条之间的外观相似
- 使用中心点图像像块做线条描述子
- 采取组合策略，卡方测试+单线条RANSAC



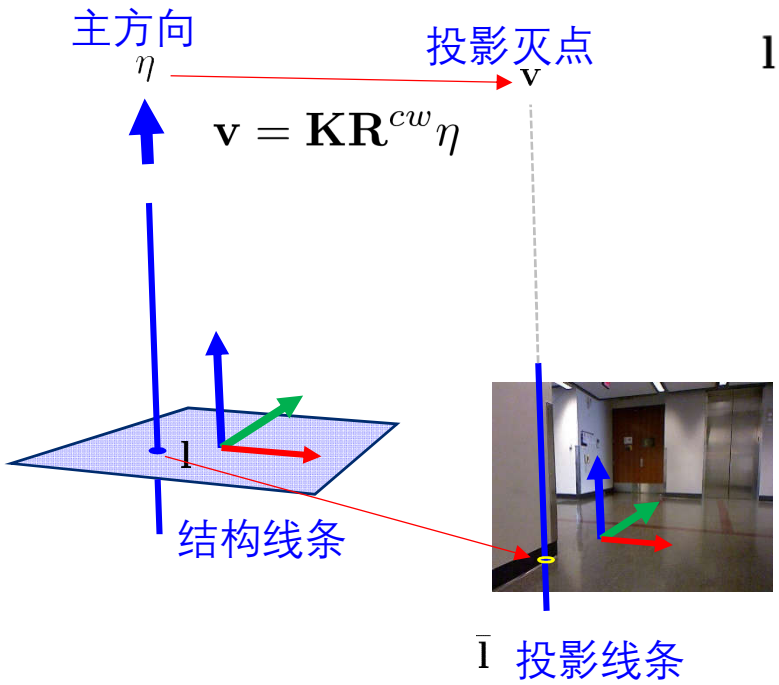
上一帧　　下一帧



单线条RANSAC可剔除噪声线段

结构化环境下的视觉SLAM方法

- 结构线条的投影过程



$$\mathbf{l} = [c_a, c_b, \theta, h]^T$$

世界坐标系

$$\mathbf{l}^w h = \mathbf{P}^T \left([\ c_a, c_b\ ]^T h + [\ \cos(\theta), \sin(\theta)\ ]^T \right)$$

相机坐标系

$$\mathbf{l}^c = \mathbf{R}^{cw}\mathbf{l}^w h - \mathbf{R}^{cw}\mathbf{p}^w h$$

像素坐标系

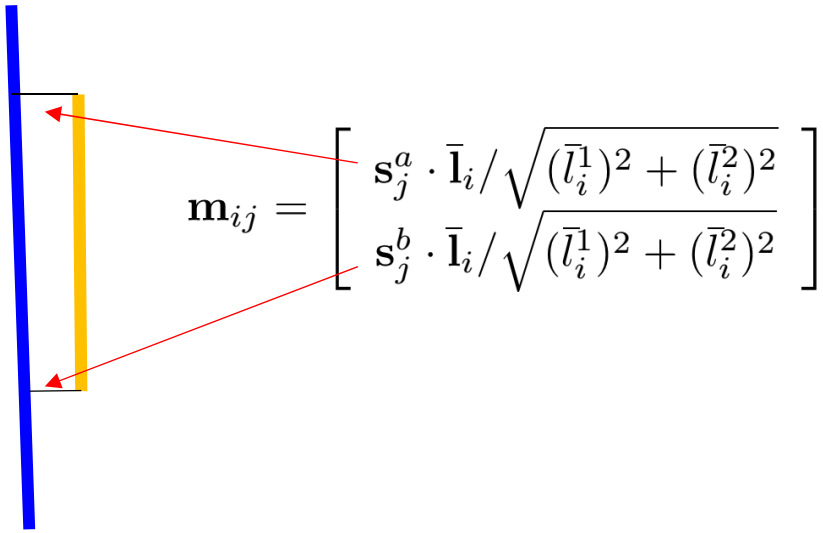
$$\mathbf{l}^i = \mathbf{K}\mathbf{l}^c$$

$$\bar{\mathbf{l}} = \mathbf{v} \times \mathbf{l}^i$$

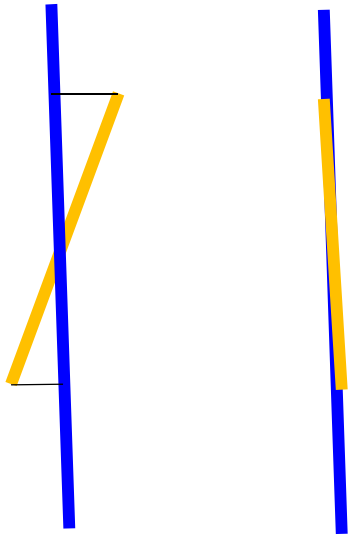
与投影灭点相连

结构化环境下的视觉SLAM方法

- 结构线条的观测模型：
  - 利用直线（模型）到线段（观测）的符号距离来描述直线与线段之间的方向一致性与接近程度



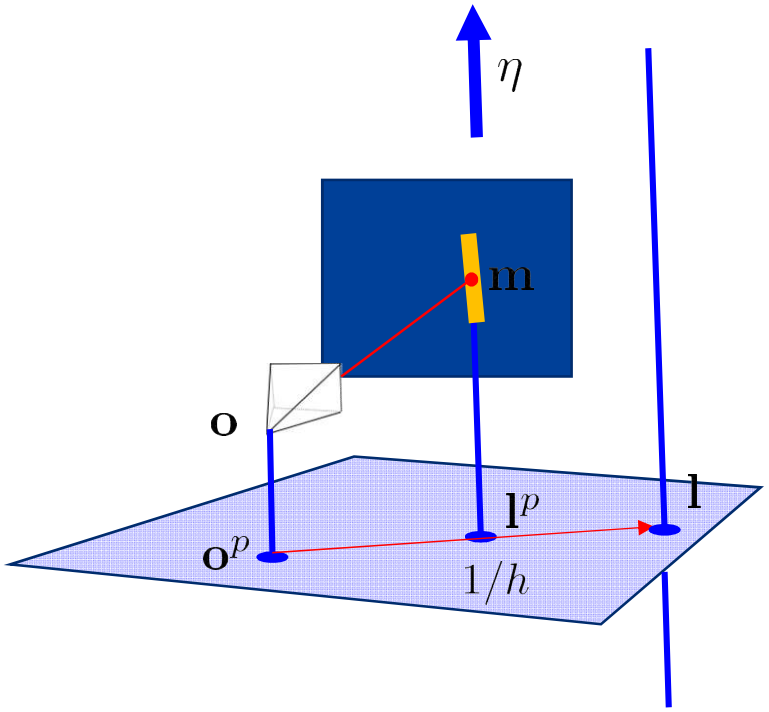
$$\mathbf{m}_{ij} = \begin{bmatrix} \mathbf{s}_j^a \cdot \bar{\mathbf{l}}_i / \sqrt{(\bar{l}_i^1)^2 + (\bar{l}_i^2)^2} \\ \mathbf{s}_j^b \cdot \bar{\mathbf{l}}_i / \sqrt{(\bar{l}_i^1)^2 + (\bar{l}_i^2)^2} \end{bmatrix}$$



残差较大　　残差较小

结构化环境下的视觉SLAM方法

- 结构线条的初始化 – 通过对图像上判别出来的的结构线段进行反投影



$$\mathbf{l} = [\ c_a, c_b, \theta, h\ ]^T$$
$$= \left[\ \mathbf{o}^p(1), \mathbf{o}^p(2), atan(\frac{\mathbf{l}^p(2)-\mathbf{o}^p(2)}{\mathbf{l}^p(1)-\mathbf{o}^p(1)}), h_0\ \right]^T$$

结构化环境下的视觉SLAM方法

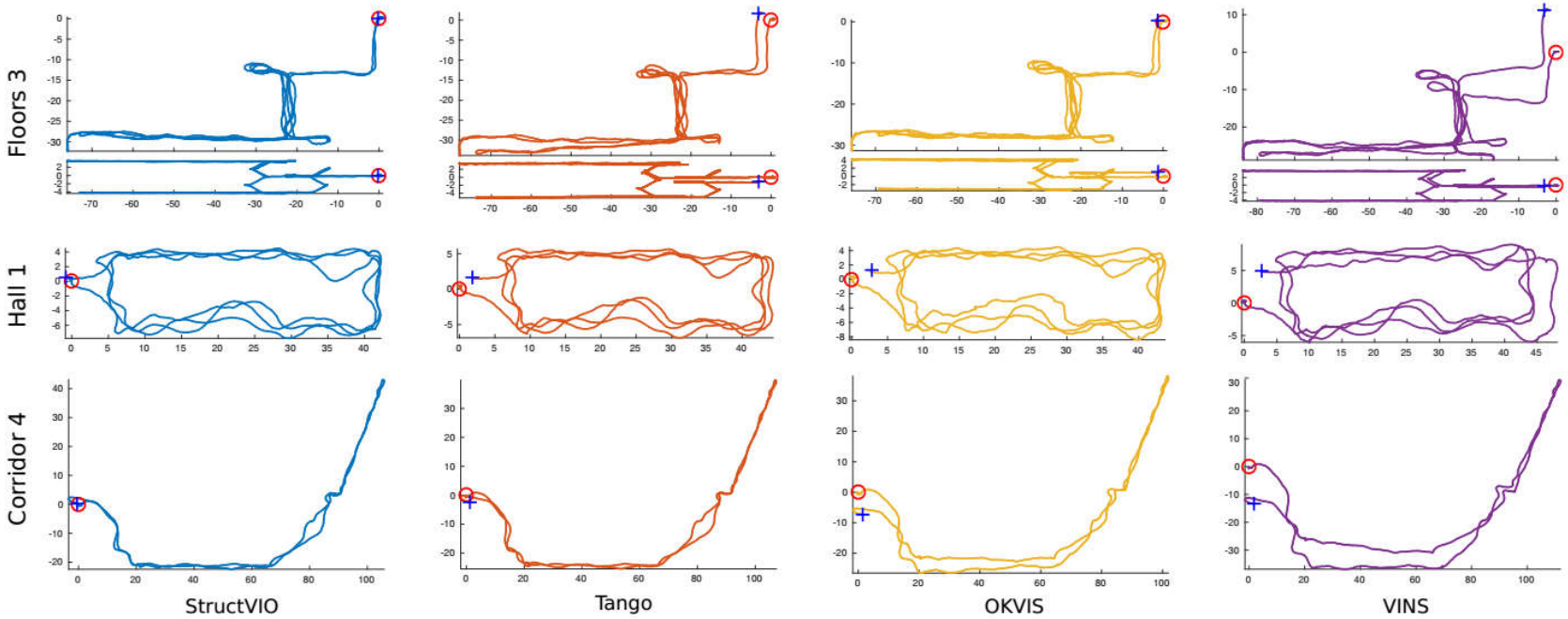
- 从StructSLAM的实验结果得出一些结论：
  - 在纹理缺失情况下，线条可增强系统鲁棒性
  - 线条所蕴含的全局方向性，可明显降低位置累积误差
  - 和里程计结合，可达到非常高的绝对定位精度 (Raseeds 0.78米平均定位误差)
  - 如果不把线条的方向性充分利用，传统的线条SLAM没有任何优势，和特征点组合甚至会降低精度

结构化环境下的视觉SLAM方法

- 改进版本： StructVIO
- 主要改进：
  - 1. 采用多个曼哈顿世界叠加的模型，处理更加不规则的场景
  - 2. 采用多状态滤波的数学框架，增加精度降低计算量
  - 3. 加入惯导进行状态预测 – 结构线条的方向先验，对快速初始化帮助很大

结构化环境下的视觉SLAM方法

- 利用结构线条信息，StructVIO比基于点的VIO方法精度都要高，
  - 评测实验一， 15组自采数据（室内大场景） –用Tango采集图像与IMU原始数据，与Tango,OKVIS,VINS等比较有代表性的VIO方法比较 （回环误差作为评测指标）



结构化环境下的视觉SLAM方法

- 评测实验二，Euroc benchmark数据 （场景较小，纹理较丰富），性能和单纯基于点特征的方法比较相近，但精度上面的改进还是能看出来

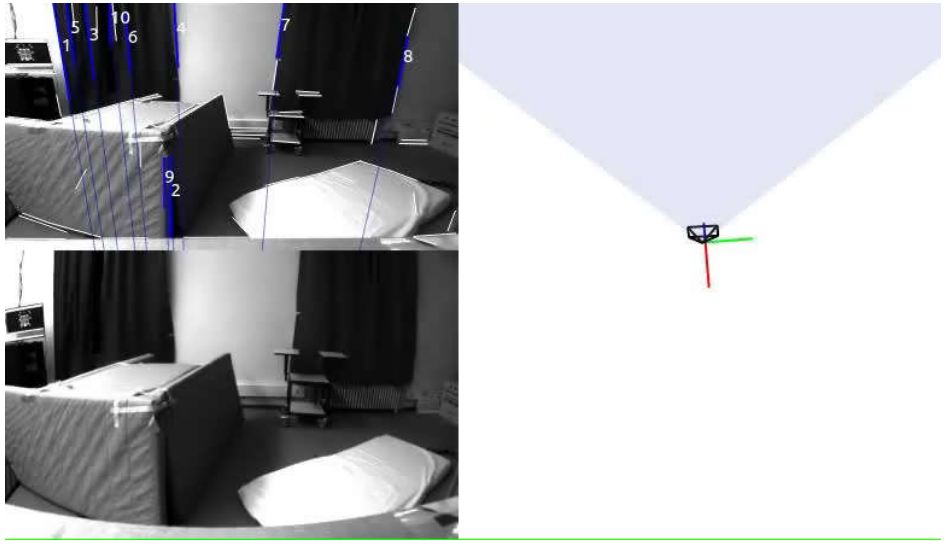
Dataset	OKVIS		VINS(w/o loop)		StructVIO	
	RMSE	Max.	RMSE	Max.	RMSE	Max.
MH_01_easy	0.349	0.737	**0.233**[2]	0.532	**0.170**[1]	0.413
MH_02_easy	4.634	7.827	**0.234**[2]	0.698	**0.135**[1]	0.299
MH_03_medium	0.251	0.538	**0.200**[2]	0.327	**0.148**[1]	0.349
MH_04_difficult	**0.238**[2]	0.476	0.368	0.651	**0.245**[1]	0.428
MH_05_difficult	0.424	0.725	**0.392**[2]	0.627	**0.275**[1]	0.501
V1_01_easy	**0.127**[2]	0.388	0.396	1.460	**0.079**[1]	0.213
V1_02_medium	**0.162**[2]	0.342	5.184	15.673	**0.146**[1]	0.343
V1_03_difficult	1.555	3.049	**0.199**[2]	0.441	**0.117**[1]	0.434
V2_01_easy	0.239	1.798	**0.081**[2]	0.140	**0.079**[1]	0.222
V2_02_medium	0.163	0.536	**0.153**[2]	0.373	**0.129**[1]	0.449
V2_03_difficult	1.919	3.872	**0.221**[1]	0.627	**0.280**[2]	0.528

- StructVIO比较适合室内大场景, 潜在应用-大商场，地库等。

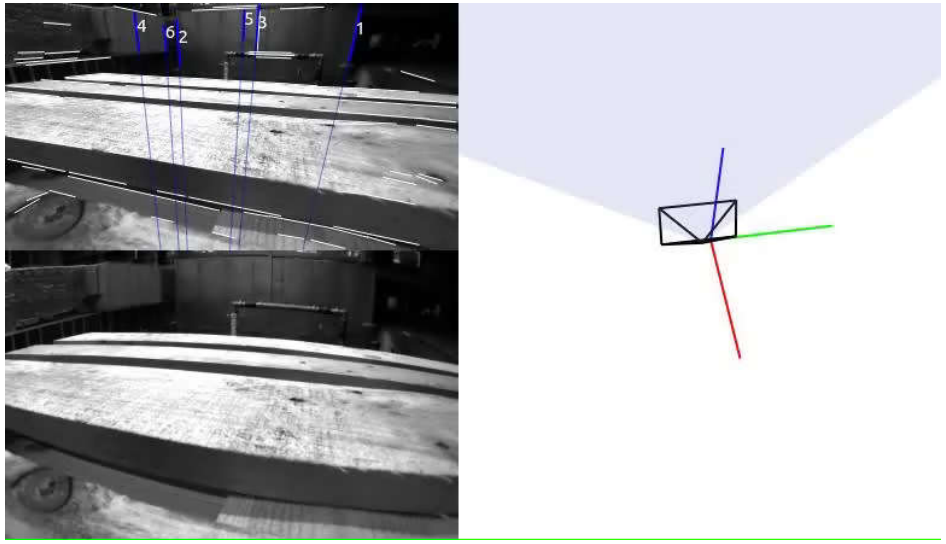
Tango 自采数据

Mulple floors - The camera travels in different floors.

Euroc



V2_03_difficult



MH_05_difficult

总结

- 高斯联合密度分布函数
  - 误差传递：随机变量的线性变换
  - 分拆公式：联合概率 -> 条件概率×边缘概率
  - 组合公式：条件概率×边缘概率 -> 联合概率
- 贝叶斯滤波->卡尔曼滤波->扩展卡尔曼滤波
- EKF-SLAM经典框架
  - 状态变量设计、观测模型、预测模型、特征管理
- 多状态约束卡尔曼滤波方法
  - 状态可露、观测方程解耦、单轨迹多状态约束观测方程、多轨迹多状态约束观测方程
- 结构化环境视觉SLAM
  - 结构线条、多曼哈顿世界模型