MSCKF学习整理

张 涛

2019.6.12

- **起源与典型代表**

1. **2007** Anastasios I. Mourikis and Stergios I. Roumeliotis. A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation. **Anastasios I. Mourikis**在明尼苏达大学上学时，开创性提出MSCKF滤波框架，历史原因称作MSCKF 1.0。
2. **2013** Mingyang Li and Anastasios I. Mourikis. High-Precision Consistent EKF-based Visual-Inertial Odometry. **李名杨**（Anastasios I. Mourikis 的学生）在加州大学上学对MSCKF框架的一致性等进行大幅完善，至此MSCKF框架理论基本完善，通常称这个版本为MSCKF 2.0，传闻Google Tango采用该算法。
3. 2017. A Comparative Analysis of Tightly-coupled Monocular, Binocular, and Stereo VINS. **Anastasios I. Mourikis**实验室的，从理论角度分析说明双目版本(both stereo and binocular)比单目版本VINS具有更高的精度、一致性、鲁棒性。
4. 2017. Robust Stereo Visual Inertial Odometry for Fast Autonomous Flight. 宾夕法尼亚大学GRASP Lab基于MSCKF实现双目+IMU版本VINS，并开源该代码

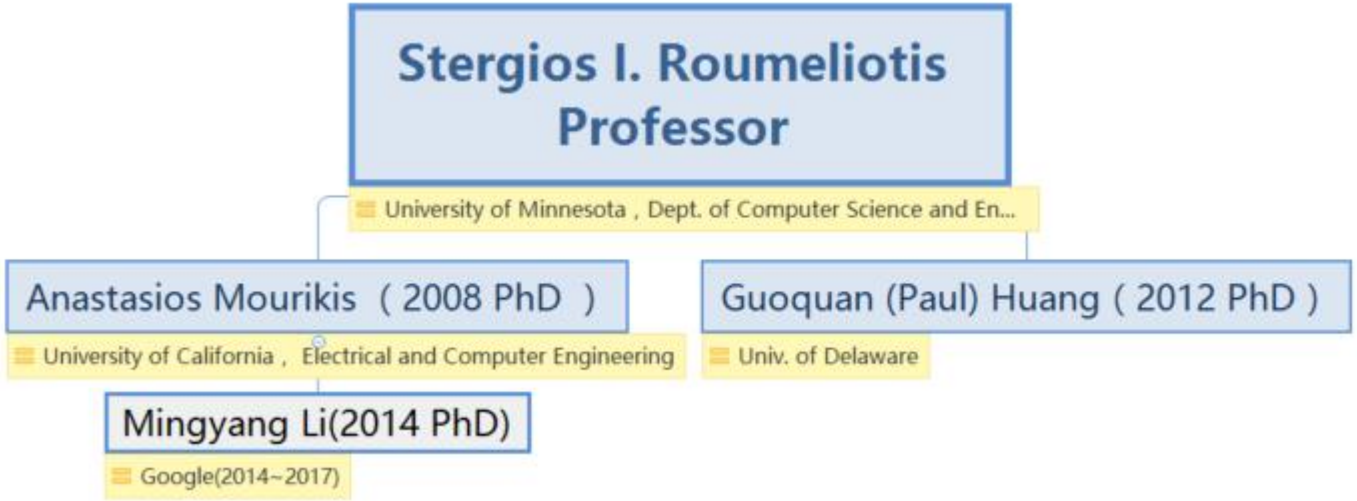
- **重要的开源代码**

PS： Anastasios I. Mourikis课题组对MSCKF源码并未开源

1. 2018. 宾夕法尼亚大学Daniilidis-group 开源其实现的Mono-MSCKF. https://github.com/daniilidis-group/msckf_mono
2. 2018.宾夕法尼亚大学GRASP Lab开源其实现的Stereo-MSCKF. https://github.com/KumarRobotics/msckf_vio
3. 2019.黄国权组 https://github.com/rpng/open_vins

- **MSCKF**

Stergios I. Roumeliotis 教授的主页： https://www-users.cs.umn.edu/~stergios/group.html



Stergios I. Roumeliotis
Professor

University of Minnesota，Dept. of Computer Science and En...

Anastasios Mourikis （2008 PhD ）

University of California， Electrical and Computer Engineering

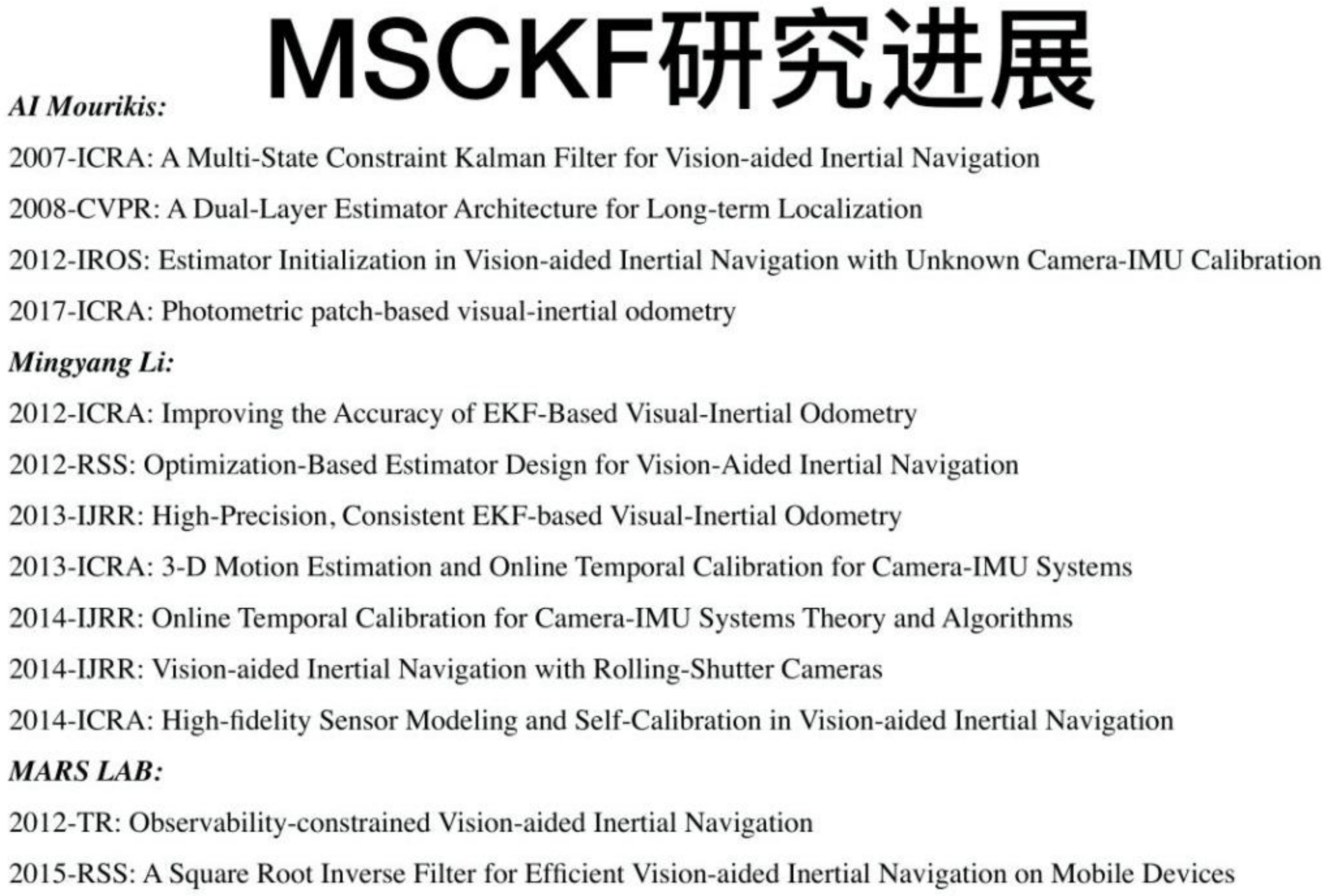
Guoquan (Paul) Huang (2012 PhD)

Univ. of Delaware

Mingyang Li(2014 PhD)

Google(2014~2017)

- **最新进展**



MSCKF研究进展

AI Mourikis:

2007-ICRA: A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation

2008-CVPR: A Dual-Layer Estimator Architecture for Long-term Localization

2012-IROS: Estimator Initialization in Vision-aided Inertial Navigation with Unknown Camera-IMU Calibration

2017-ICRA: Photometric patch-based visual-inertial odometry

Mingyang Li:

2012-ICRA: Improving the Accuracy of EKF-Based Visual-Inertial Odometry

2012-RSS: Optimization-Based Estimator Design for Vision-Aided Inertial Navigation

2013-IJRR: High-Precision, Consistent EKF-based Visual-Inertial Odometry

2013-ICRA: 3-D Motion Estimation and Online Temporal Calibration for Camera-IMU Systems

2014-IJRR: Online Temporal Calibration for Camera-IMU Systems Theory and Algorithms

2014-IJRR: Vision-aided Inertial Navigation with Rolling-Shutter Cameras

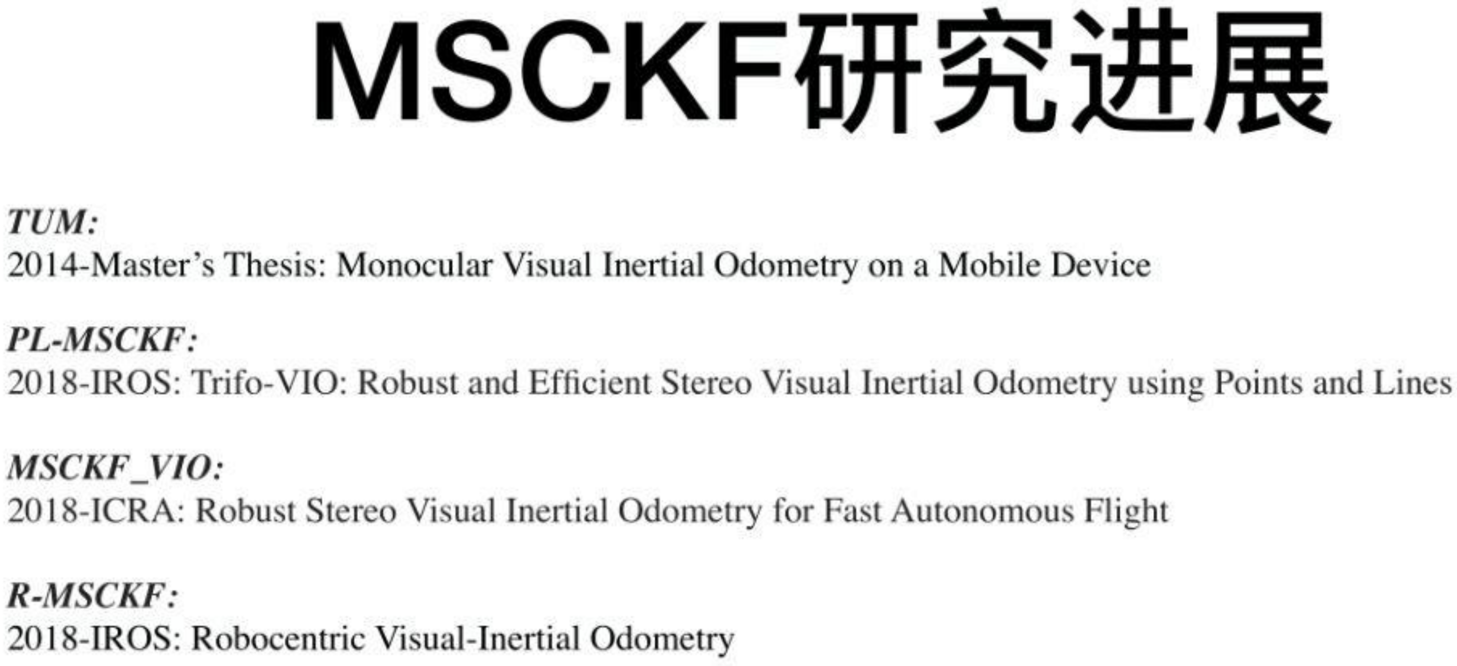
2014-ICRA: High-fidelity Sensor Modeling and Self-Calibration in Vision-aided Inertial Navigation

MARS LAB:

2012-TR: Observability-constrained Vision-aided Inertial Navigation

2015-RSS: A Square Root Inverse Filter for Efficient Vision-aided Inertial Navigation on Mobile Devices

- **最新进展**



MSCKF研究进展

TUM:
2014-Master's Thesis: Monocular Visual Inertial Odometry on a Mobile Device

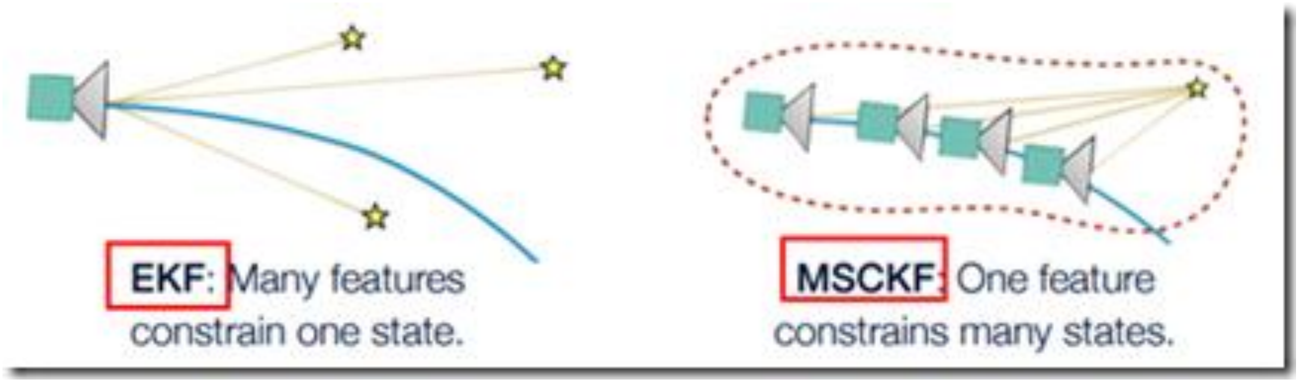
PL-MSCKF:
2018-IROS: Trifo-VIO: Robust and Efficient Stereo Visual Inertial Odometry using Points and Lines

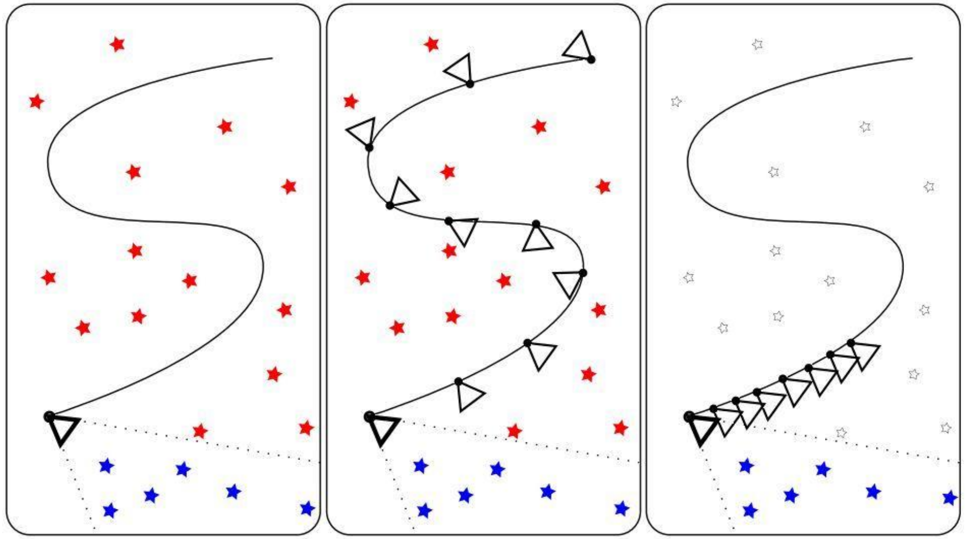
MSCKF_VIO:
2018-ICRA: Robust Stereo Visual Inertial Odometry for Fast Autonomous Flight

R-MSCKF:
2018-IROS: Robocentric Visual-Inertial Odometry

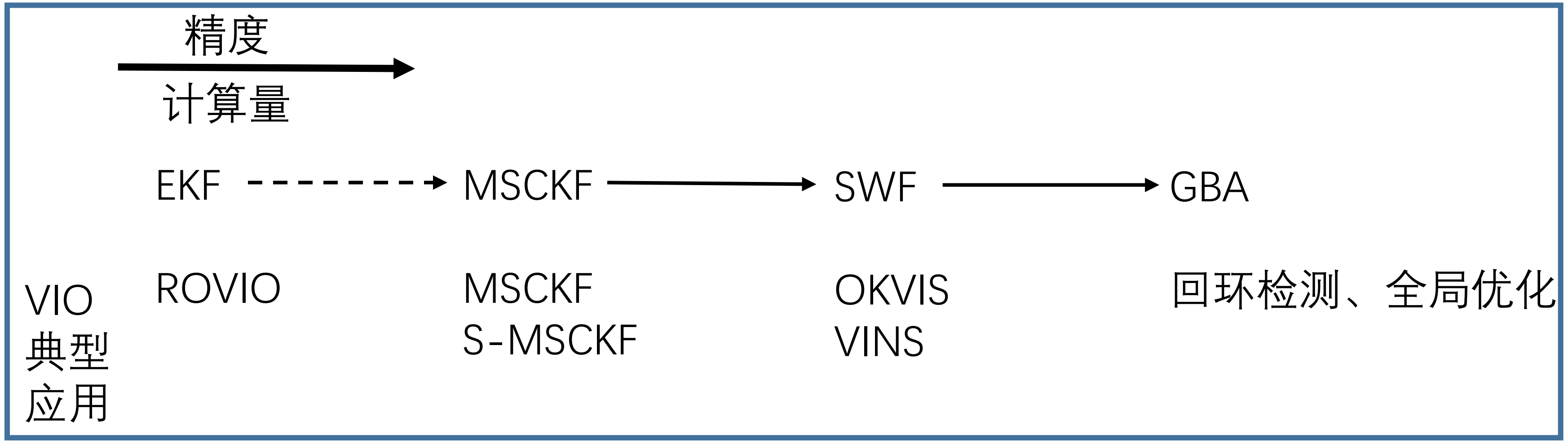
● 什么是MSCKF

MSCKF（ Multi-State Constraint Kalman Filter），中文直译“多状态约束的卡尔曼滤波”，基于EKF


EKF: Many features constrain one state.
MSCKF: One feature constrains many states.



EKF keyframe-based(BA) MSCKF



精度

计算量

EKF - - - → MSCKF → SWF → GBA

VIO典型应用

ROVIO

MSCKF
S-MSCKF

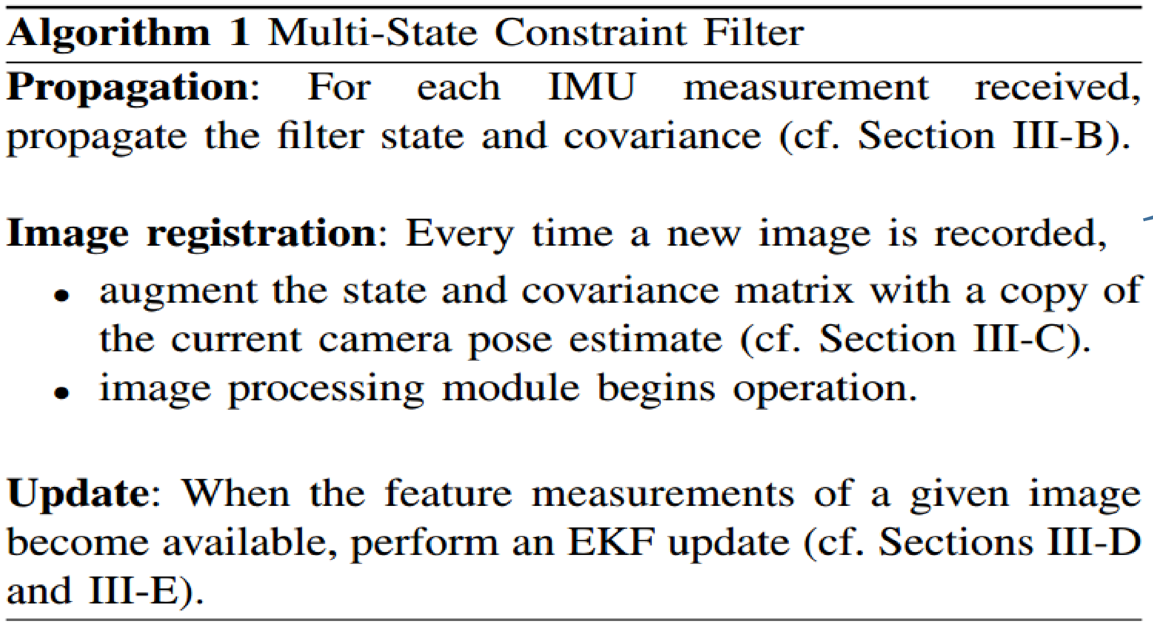
OKVIS
VINS

回环检测、全局优化

PS：VIORB是否采用滑窗不确定，到时看过论文后再补上，另外SR-ISWF 、VINet 、PIRVS还不咋了解

● 什么是MSCKF

MSCKF算法流程



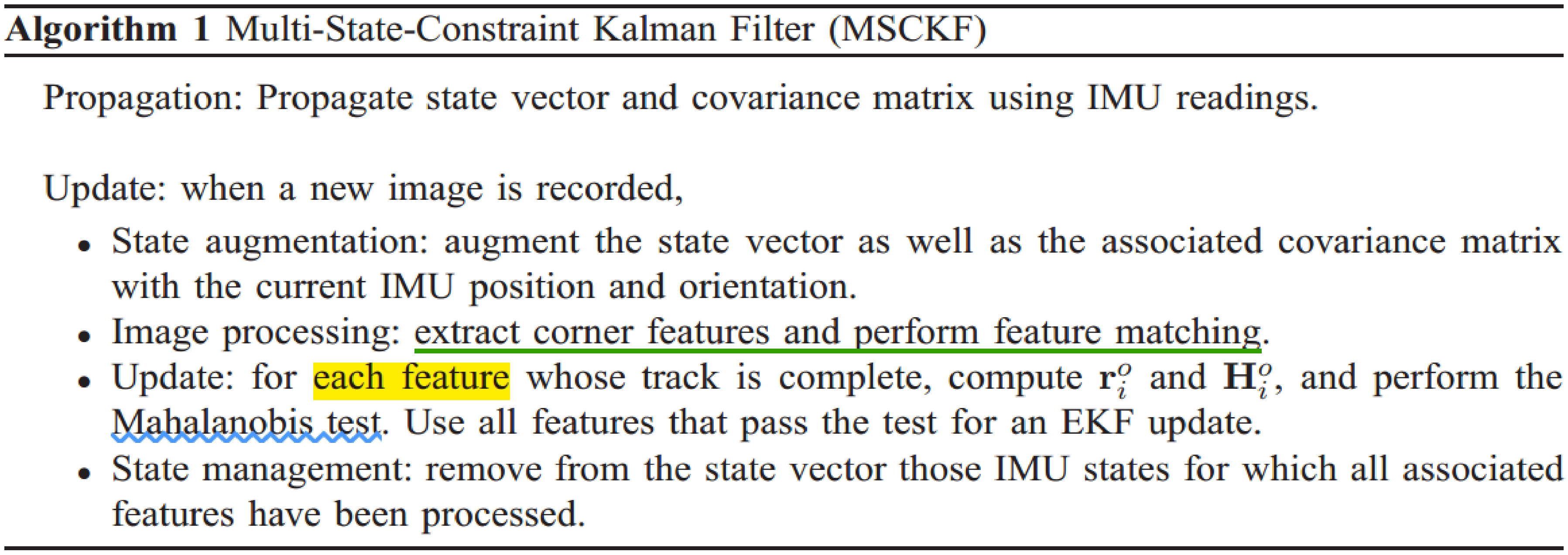
Algorithm 1 Multi-State Constraint Filter

Propagation: For each IMU measurement received, propagate the filter state and covariance (cf. Section III-B).

Image registration: Every time a new image is recorded,

- augment the state and covariance matrix with a copy of the current camera pose estimate (cf. Section III-C).
- image processing module begins operation.

Update: When the feature measurements of a given image become available, perform an EKF update (cf. Sections III-D and III-E).



Algorithm 1 Multi-State-Constraint Kalman Filter (MSCKF)

Propagation: Propagate state vector and covariance matrix using IMU readings.

Update: when a new image is recorded,

- State augmentation: augment the state vector as well as the associated covariance matrix with the current IMU position and orientation.
- Image processing: extract corner features and perform feature matching.
- Update: for each feature whose track is complete, compute $\mathbf{r}_i^o$ and $\mathbf{H}_i^o$, and perform the Mahalanobis test. Use all features that pass the test for an EKF update.
- State management: remove from the state vector those IMU states for which all associated features have been processed.

● MSCKF与EKF比较

EKF: **多个特征点同时约束一个相机位姿**，进行KF更新
MSCKF ：**一个特征点同时约束多个相机位姿**(多相机观测同时优化，窗口多帧优化)，进行KF更新

传统的 EKF-based SLAM 做 IMU 融合时：
1、**系统状态向量：**包含当前的 位姿pose、速度velocity、以及 3D map points 坐标、 IMU 的 bias（零飘和温飘）等；
2、**预测阶段**：用 IMU 做预测predict step，
3、**更新阶段**：用 image frame 中观测 3D map points 的观测误差做更新update step。

MSCKF 的 **motivation**是EKF的每次更新update step 是基于 3D map points 在单帧 frame 里观测的，如果能基于其在多帧中的观测效果应该会好（有点类似于 local bundle adjustment 的思想）； 另外，状态向量中不包含3D map points 坐标，计算复杂性仅与points线性相关。

MSCKF 的改进如下：
1、**系统状态向量**：不包含3D map points 坐标
2、**预测阶段**：跟 EKF 的predict step一样
3、**更新阶段**：更新update step 推迟到某一个 3D map point 在多个 frame 中观测之后进行计算；在 update 之前每接收到一个 image frame，只是将 state vector 扩充并加入当前 frame 的 pose estimate。

这个思想基本类似于 local bundle adjustment（或者 sliding window smoothing），在update step时，相当于基于多次观测同时优化 pose 和 3D map point。

- **EKF的常见表示**

$$\begin{aligned} \text{motion model:} \quad & \mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{v}_k + \mathbf{w}_k, \quad k = 1 \ldots K \\ \text{observation model:} \quad & \mathbf{y}_k = \mathbf{C}_k\mathbf{x}_k + \mathbf{n}_k, \quad k = 0 \ldots K \end{aligned}$$

系统线性化的运动模型与观测模型

$$\begin{aligned} \text{predictor:} \quad & \check{\mathbf{P}}_k = \mathbf{A}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{A}_{k-1}^T + \mathbf{Q}_k \\ & \check{\mathbf{x}}_k = \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1} + \mathbf{v}_k, \\ \text{corrector:} \quad & \hat{\mathbf{P}}_k^{-1} = \check{\mathbf{P}}_k^{-1} + \mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{C}_k, \\ & \hat{\mathbf{P}}_k^{-1}\hat{\mathbf{x}}_k = \check{\mathbf{P}}_k^{-1}\check{\mathbf{x}}_k + \mathbf{C}_k^T\mathbf{R}_k^{-1}\mathbf{y}_k, \end{aligned}$$

EKF表现形式1

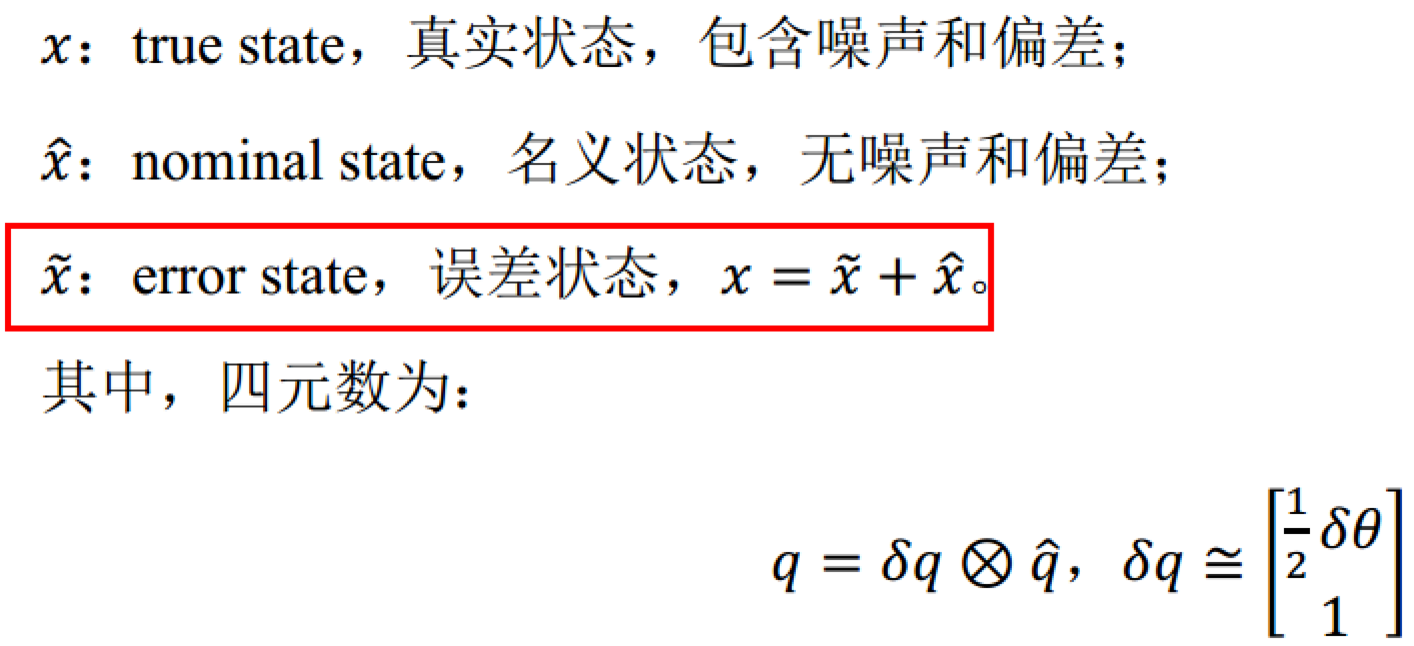
$$\begin{aligned} \text{predictor:} \quad & \check{\mathbf{P}}_k = \mathbf{A}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{A}_{k-1}^T + \mathbf{Q}_k, \\ & \check{\mathbf{x}}_k = \mathbf{A}_{k-1}\hat{\mathbf{x}}_{k-1} + \mathbf{v}_k, \\ \text{Kalman gain:} \quad & \mathbf{K}_k = \check{\mathbf{P}}_k\mathbf{C}_k^T\left(\mathbf{C}_k\check{\mathbf{P}}_k\mathbf{C}_k^T + \mathbf{R}_k\right)^{-1} \\ \text{corrector:} \quad & \hat{\mathbf{P}}_k = \left(\mathbf{1} - \mathbf{K}_k\mathbf{C}_k\right)\check{\mathbf{P}}_k, \\ & \hat{\mathbf{x}}_k = \check{\mathbf{x}}_k + \mathbf{K}_k\underbrace{\left(\mathbf{y}_k - \mathbf{C}_k\check{\mathbf{x}}_k\right)}_{\text{innovation}}, \end{aligned}$$

EKF表现形式2（带卡尔曼增益）

● MSCKF——状态向量

➢ 符号说明

符号情况说明



x：true state，真实状态，包含噪声和偏差；

$\hat{x}$：nominal state，名义状态，无噪声和偏差；

$\tilde{x}$：error state，误差状态，$x = \tilde{x} + \hat{x}$。

其中，四元数为：

$$q = \delta q \otimes \hat{q}，\delta q \cong \begin{bmatrix} \frac{1}{2}\delta\theta \\ 1 \end{bmatrix}$$

采用误差模型的几点原因：

1、参数数目与系统自由度一致，避免过参数化或冗余，也能避免协方差矩阵奇异

2、误差状态较小，二阶及高阶项可忽略，这样可减小雅克比等过程的计算量；

3、误差状态变化速度慢，滤波算法中更新阶段可采用更低的更新速率；

- **MSCKF——状态向量**

➢ 真实状态向量

$$X_k^{(16+7N)\times 1} = \begin{bmatrix} X_{IMU_k} \\ {}^{C_1}_{G}\bar{q} \\ {}^{G}p_{C_1} \\ \vdots \\ {}^{C_N}_{G}\bar{q} \\ {}^{G}p_{C_N} \end{bmatrix}$$

$$X_{IMU_k}^{16\times 1} = \begin{bmatrix} {}^{I}_{G}\bar{q} \\ b_g \\ {}^{G}v_I \\ b_a \\ {}^{G}p_I \end{bmatrix}$$

$$\pi_i^{7\times 1} = \begin{bmatrix} {}^{C_i}_{G}\bar{q} \\ {}^{G}p_{C_i} \end{bmatrix}$$

真实状态向量，包括第k时刻的IMU状态，N个相机位姿

➢ 误差状态向量

$$\tilde{X}_k^{(15+6N)\times 1} = \begin{bmatrix} \tilde{X}_{IMU_k} \\ \delta\theta_{C_1} \\ {}^{G}\tilde{p}_{C_1} \\ \vdots \\ \delta\theta_{C_N} \\ {}^{G}\tilde{p}_{C_N} \end{bmatrix}$$

$$\tilde{X}_{IMU_k}^{15\times 1} = \begin{bmatrix} \delta\theta_I \\ \tilde{b}_g \\ {}^{G}\tilde{v}_I \\ \tilde{b}_a \\ {}^{G}\tilde{p}_I \end{bmatrix}$$

误差状态下：四维四元数采用三维theta替代

MSCKF——基于IMU的运动方程

IMU系统模型

$$ {}^I_G\dot{\bar{q}}(t) = \frac{1}{2}\boldsymbol{\Omega}\big(\boldsymbol{\omega}(t)\big)\,{}^I_G\bar{q}(t), \quad \dot{\mathbf{b}}_g(t) = \mathbf{n}_{wg}(t) $$

$$ {}^G\dot{\mathbf{v}}_I(t) = {}^G\mathbf{a}(t), \quad \dot{\mathbf{b}}_a(t) = \mathbf{n}_{wa}(t), \quad {}^G\dot{\mathbf{p}}_I(t) = {}^G\mathbf{v}_I(t) $$

$$ \boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} -\lfloor\boldsymbol{\omega}\times\rfloor & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix}, \quad \lfloor\boldsymbol{\omega}\times\rfloor = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} $$

$$ \boldsymbol{\omega}_m = \boldsymbol{\omega} + \mathbf{C}({}^I_G\bar{q})\boldsymbol{\omega}_G + \mathbf{b}_g + \mathbf{n}_g \qquad (7) $$

$$ \begin{aligned} \mathbf{a}_m = {} & \mathbf{C}({}^I_G\bar{q})({}^G\mathbf{a} - {}^G\mathbf{g} + 2\lfloor\boldsymbol{\omega}_G\times\rfloor\,{}^G\mathbf{v}_I + \lfloor\boldsymbol{\omega}_G\times\rfloor^2\,{}^G\mathbf{p}_I) \\ & + \mathbf{b}_a + \mathbf{n}_a \qquad (8) \end{aligned} $$

IMU系统模型，此时建模过程中考虑了地球自转的影响

$$ {}^I_G\dot{\hat{\bar{q}}} = \frac{1}{2}\boldsymbol{\Omega}(\hat{\boldsymbol{\omega}})\,{}^I_G\hat{\bar{q}}, \quad \dot{\hat{\mathbf{b}}}_g = \mathbf{0}_{3\times1}, $$

$$ {}^G\dot{\hat{\mathbf{v}}}_I = \mathbf{C}_{\hat{q}}^T\hat{\mathbf{a}} - 2\lfloor\boldsymbol{\omega}_G\times\rfloor\,{}^G\hat{\mathbf{v}}_I - \lfloor\boldsymbol{\omega}_G\times\rfloor^2\,{}^G\hat{\mathbf{p}}_I + {}^G\mathbf{g} $$

$$ \dot{\hat{\mathbf{b}}}_a = \mathbf{0}_{3\times1}, \quad {}^G\dot{\hat{\mathbf{p}}}_I = {}^G\hat{\mathbf{v}}_I $$

- MSCKF——基于IMU的运动方程

➢ 误差运动方程

连续形式

$$\dot{\tilde{X}}_{IMU}{}^{15\times1} = F^{15\times15}\tilde{X}_{IMU}{}^{15\times1} + G^{15\times12}n_{IMU}{}^{12\times1}$$

离散形式

$$\tilde{X}_{IMU_{k+1}} = \Phi(t_k + \Delta T, t_k)\tilde{X}_{IMU_k} + (G\Delta T)n_{IMU}$$

$$\Phi(t_k + \Delta T, t_k) = exp(F) \cong \mathrm{I}_{15\times15} + \mathrm{F}\Delta T$$

SWF与MSCKF对比
文章 式（27）

$$\mathbf{F} = \begin{bmatrix} -\lfloor\hat{\boldsymbol{\omega}}\times\rfloor & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ -\mathbf{C}_{\hat{q}}^T\lfloor\hat{\mathbf{a}}\times\rfloor & \mathbf{0}_{3\times3} & -2\lfloor\boldsymbol{\omega}_G\times\rfloor & -\mathbf{C}_{\hat{q}}^T & -\lfloor\boldsymbol{\omega}_G\times\rfloor^2 \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}$$

$$\boldsymbol{\Phi}_k = \boldsymbol{\Phi}(t_{k+1}, t_k) = \exp\left(\int_{t_k}^{t_{k+1}} \mathbf{F}(\tau)d\tau\right)$$

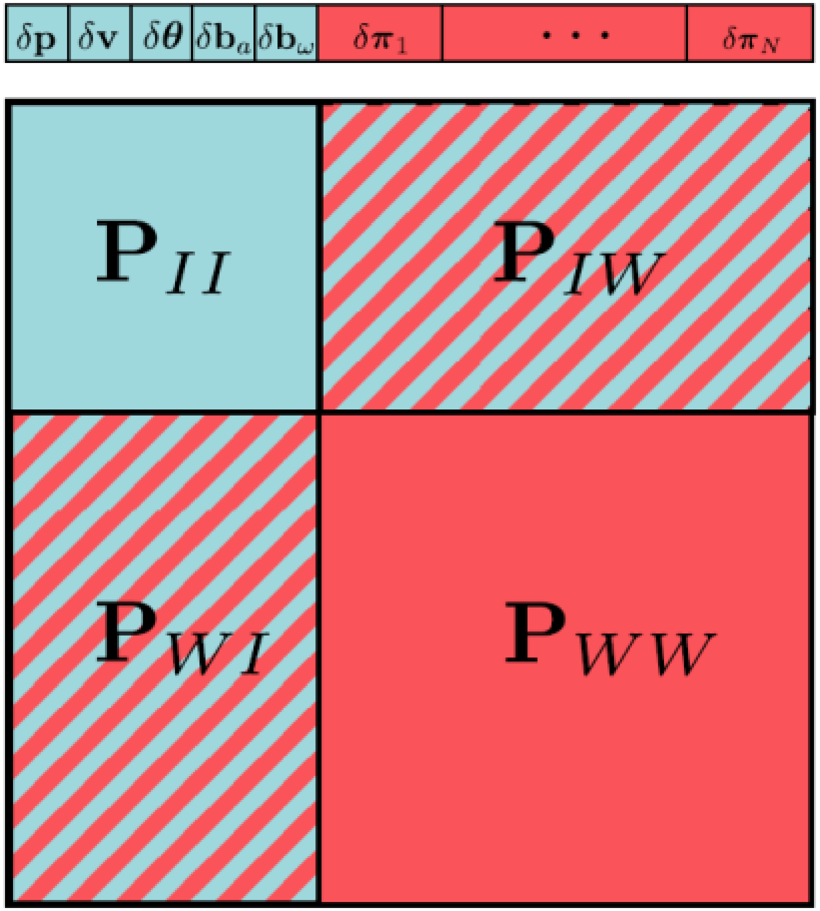
同样的表达式，
SMSCKF的计算方法

$$\mathbf{G} = \begin{bmatrix} -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{C}_{\hat{q}}^T & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix} \qquad n_{IMU} = \begin{bmatrix} n_g \\ n_{wg} \\ n_a \\ n_{wa} \end{bmatrix}$$

- MSCKF——基于IMU的运动方程

➢ 协方差矩阵传播

$$P_{k|k}{}^{(15+6N)\times(15+6N)}=\begin{bmatrix}P_{II_{k|k}} & P_{IC_{k|k}}\\ P_{IC_{k|k}}^{T} & P_{CC_{k|k}}\end{bmatrix}$$



第k时刻的协方差矩阵

第k + 1时刻预测的协方差矩阵
SWF与MSCKF对比文章 式（27）

$$P_{k+1|k}{}^{(15+6N)\times(15+6N)}=\begin{bmatrix}P_{II_{k+1|k}} & \Phi(t_k+\Delta T,t_k)P_{IC_{k|k}}\\ P_{IC_{k|k}}^{T}\Phi(t_k+\Delta T,t_k)^{T} & P_{CC_{k|k}}\end{bmatrix}$$

$$P_{II_{k+1|k}}=\Phi P_{II_{k|k}}\Phi^{T}+GQ_{IMU}G^{T}\Delta T$$

$$P_{IC_{k+1|k}}=\Phi(t_k+\Delta T,t_k)P_{IC_{k|k}}$$

$$\mathbf{P}_{II_{k+1|k}}=\boldsymbol{\Phi}_k\mathbf{P}_{II_{k|k}}\boldsymbol{\Phi}_k^{\top}+\mathbf{Q}_k$$

$$\boldsymbol{\Phi}_k=\boldsymbol{\Phi}(t_{k+1},t_k)=\exp\left(\int_{t_k}^{t_{k+1}}\mathbf{F}(\tau)d\tau\right)$$

$$\mathbf{Q}_k=\int_{t_k}^{t_{k+1}}\boldsymbol{\Phi}(t_{k+1},\tau)\mathbf{GQG}\boldsymbol{\Phi}(t_{k+1},\tau)^{\top}d\tau$$

同样的表达式，SMSCKF的计算方法

● MSCKF——相机位姿状态增广

➢ 状态向量增广

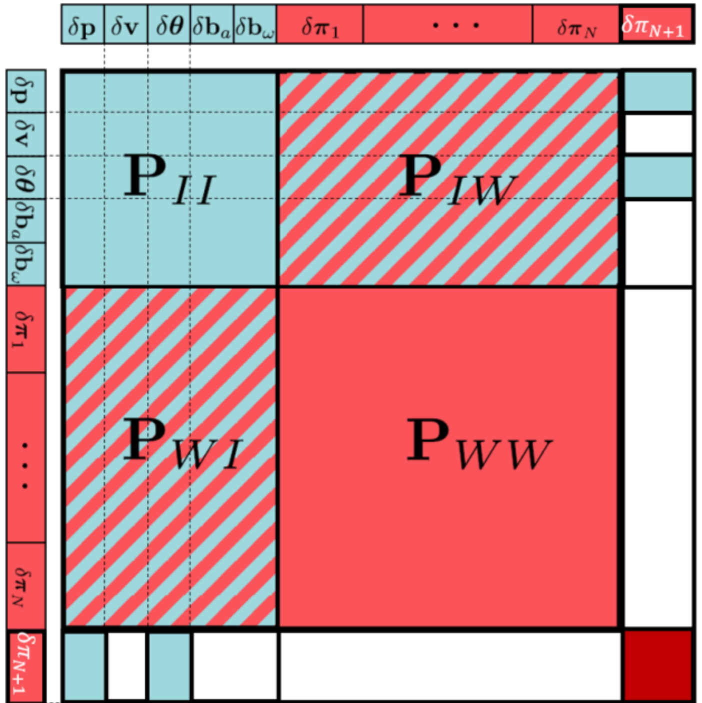
当获得新的图像帧时，需要将新图像帧的相机位姿添加到当前状态向量；并且新图像帧的相机位姿通过IMU估计

$${}^{C}_{G}\hat{\bar{q}} = {}^{C}_{I}\bar{q} \otimes {}^{I}_{G}\hat{\bar{q}}, \quad \text{and} \quad {}^{G}\hat{\mathbf{p}}_C = {}^{G}\hat{\mathbf{p}}_I + \mathbf{C}_{\hat{\bar{q}}}^{T}\,{}^{I}\mathbf{p}_C$$

通过IMU估计新图像帧的相机位姿

➢ 协方差矩阵增广

$$\mathbf{P}_{k|k} \leftarrow \begin{bmatrix}\mathbf{I}_{6N+15} \\ \mathbf{J}\end{bmatrix} \mathbf{P}_{k|k} \begin{bmatrix}\mathbf{I}_{6N+15} \\ \mathbf{J}\end{bmatrix}^{T} \qquad \mathbf{J} = \begin{bmatrix} \mathbf{C}\left({}^{C}_{I}\bar{q}\right) & \mathbf{0}_{3\times 9} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 6N} \\ \lfloor \mathbf{C}_{\hat{\bar{q}}}^{T}\,{}^{I}\mathbf{p}_C \times \rfloor & \mathbf{0}_{3\times 9} & \mathbf{I}_{3} & \mathbf{0}_{3\times 6N} \end{bmatrix}$$



● MSCKF——视觉测量模型

➢ 视觉测量残差

第j个路标点在第i个相机中的重投影误差为：

$$\mathbf{r}_i^{(j)} = \mathbf{z}_i^{(j)} - \hat{\mathbf{z}}_i^{(j)}$$

其中：

$$\hat{\mathbf{z}}_i^{(j)} = \frac{1}{^{C_i}\hat{Z}_j}\begin{bmatrix} ^{C_i}\hat{X}_j \\ ^{C_i}\hat{Y}_j \end{bmatrix}, \quad \begin{bmatrix} ^{C_i}\hat{X}_j \\ ^{C_i}\hat{Y}_j \\ ^{C_i}\hat{Z}_j \end{bmatrix} = \mathbf{C}(^{C_i}_{G}\hat{\bar{q}})(^{G}\hat{\mathbf{p}}_{f_j} - {}^{G}\hat{\mathbf{p}}_{C_i})$$

➢ 残差线性化

$$\mathbf{r}_i^{(j)} \simeq \boxed{\mathbf{H}_{\mathbf{X}_i}^{(j)}\widetilde{\mathbf{X}}} + \boxed{\mathbf{H}_{f_i}^{(j)}\,{}^{G}\widetilde{\mathbf{p}}_{f_j}} + \mathbf{n}_i^{(j)}$$

第j个路标点在第i个相机残差的线性化

系统状态向量相关的部分

路标点位置相关的部分

$$\mathbf{r}^{(j)} \simeq \mathbf{H}_{\mathbf{X}}^{(j)}\widetilde{\mathbf{X}} + \mathbf{H}_{f}^{(j)}\,{}^{G}\widetilde{\mathbf{p}}_{f_j} + \mathbf{n}^{(j)}$$

第j个路标点在Mj个相机残差的线性化（提升形式）

➢ 边缘化路标点的位置误差

对于 EKF，残差线性化需要满足残差与误差项成线性化的形式，而上述线性化形式中，包含不在系统状态误差项中的路标点位置，为使用EKF，需要将路标点位置部分边缘化

$$\begin{aligned}\mathbf{r}_o^{(j)} &= \mathbf{A}^T(\mathbf{z}^{(j)} - \hat{\mathbf{z}}^{(j)}) \simeq \mathbf{A}^T\mathbf{H}_{\mathbf{X}}^{(j)}\widetilde{\mathbf{X}} + \mathbf{A}^T\mathbf{n}^{(j)} \\ &= \mathbf{H}_o^{(j)}\widetilde{\mathbf{X}}^{(j)} + \mathbf{n}_o^{(j)}\end{aligned}$$

H_f左零空间构成的矩阵A边缘化路标点位置相关的部分

● MSCKF——EKF更新

- ➢ EKF更新被下述**任一条件**触发：

1. 路标点被跟丢（When a feature that has been tracked in a number of images is no longer detected）

   （PS：该情况更常见）；

2. 当状态向量中图像帧数量达到设定的最大值时，需移除某些老的图像帧（If the maximum allowable number of camera poses, Nmax, has been reached）

- 从second-oldest 开始等间隔移除状态向量中1/3的图像帧
- 参照了VINS-Mono的关键帧选择策略（SMSCKF做的改进）

- ➢ 具体更新过程

$$\mathbf{H}_{\mathbf{X}} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{bmatrix} \begin{bmatrix} \mathbf{T}_H \\ \mathbf{0} \end{bmatrix}$$

$$\mathbf{r}_o = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{bmatrix} \begin{bmatrix} \mathbf{T}_H \\ \mathbf{0} \end{bmatrix} \widetilde{\mathbf{X}} + \mathbf{n}_o \Rightarrow$$

$$\begin{bmatrix} \mathbf{Q}_1^T \mathbf{r}_o \\ \mathbf{Q}_2^T \mathbf{r}_o \end{bmatrix} = \begin{bmatrix} \mathbf{T}_H \\ \mathbf{0} \end{bmatrix} \widetilde{\mathbf{X}} + \begin{bmatrix} \mathbf{Q}_1^T \mathbf{n}_o \\ \mathbf{Q}_2^T \mathbf{n}_o \end{bmatrix}$$

$$\mathbf{r}_n = \mathbf{Q}_1^T \mathbf{r}_o = \mathbf{T}_H \widetilde{\mathbf{X}} + \mathbf{n}_n$$

QR分解

所有路标点在对应图像帧的残差（By stacking all residuals in a single vector）

$$\mathbf{r}_o = \mathbf{H}_{\mathbf{X}} \widetilde{\mathbf{X}} + \mathbf{n}_o$$

$$\mathbf{K} = \mathbf{P}\mathbf{T}_H^T \left(\mathbf{T}_H \mathbf{P} \mathbf{T}_H^T + \mathbf{R}_n \right)^{-1}$$

卡尔曼增益

$$\Delta \mathbf{X} = \mathbf{K} \mathbf{r}_n$$

状态更新部分

$$\mathbf{P}_{k+1|k+1} = \left(\mathbf{I}_\xi - \mathbf{K}\mathbf{T}_H \right) \mathbf{P}_{k+1|k} \left(\mathbf{I}_\xi - \mathbf{K}\mathbf{T}_H \right)^T + \mathbf{K}\mathbf{R}_n\mathbf{K}^T$$

协方差矩阵更新部分

● MSCKF——路标点位置计算

- ➢ 逆深度参数化

$$
{}^{C_i}\mathbf{p}_{f_j} = \mathbf{C}({}^{C_i}_{C_n}\bar{q})\,{}^{C_n}\mathbf{p}_{f_j} + {}^{C_i}\mathbf{p}_{C_n}, \qquad i \in \mathcal{S}_j
$$

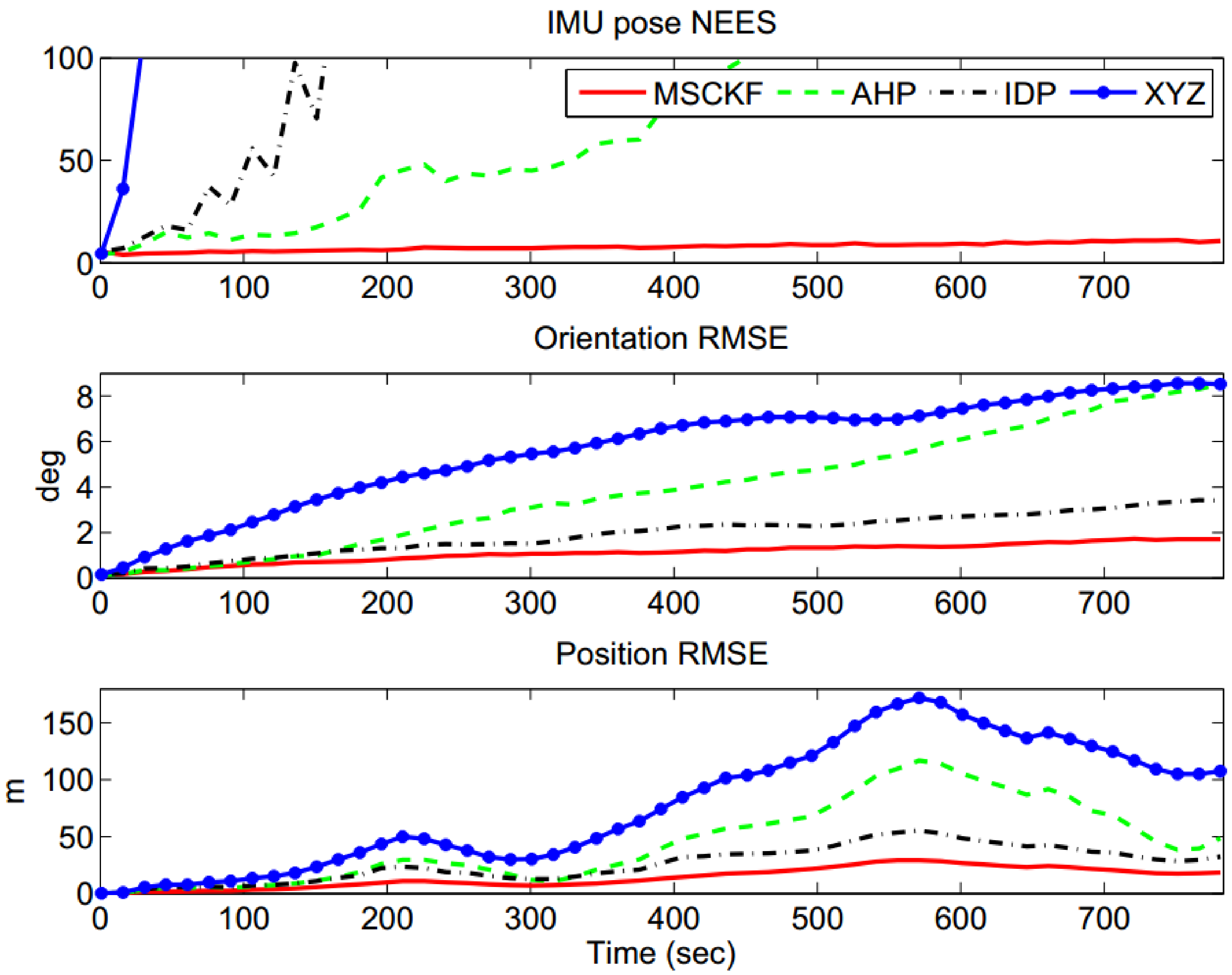
$$
\begin{aligned}
{}^{C_i}\mathbf{p}_{f_j} &= {}^{C_n}Z_j \left(\mathbf{C}({}^{C_i}_{C_n}\bar{q}) \begin{bmatrix} \frac{{}^{C_n}X_j}{{}^{C_n}Z_j} \\ \frac{{}^{C_n}Y_j}{{}^{C_n}Z_j} \\ 1 \end{bmatrix} + \frac{1}{{}^{C_n}Z_j}\,{}^{C_i}\mathbf{p}_{C_n} \right) \\
&= {}^{C_n}Z_j \left(\mathbf{C}({}^{C_i}_{C_n}\bar{q}) \begin{bmatrix} \alpha_j \\ \beta_j \\ 1 \end{bmatrix} + \rho_j\,{}^{C_i}\mathbf{p}_{C_n} \right) \\
&= {}^{C_n}Z_j \begin{bmatrix} h_{i1}(\alpha_j, \beta_j, \rho_j) \\ h_{i2}(\alpha_j, \beta_j, \rho_j) \\ h_{i3}(\alpha_j, \beta_j, \rho_j) \end{bmatrix}
\end{aligned}
$$

$$
\alpha_j = \frac{{}^{C_n}X_j}{{}^{C_n}Z_j}, \qquad \beta_j = \frac{{}^{C_1}Y_j}{{}^{C_n}Z_j}, \qquad \rho_j = \frac{1}{{}^{C_n}Z_j};
$$

- ➢ 最小化重投影误差，GN等
- ➢ 最小二乘优化中相机位姿固定，忽略相机位姿的协方差影响，即**struct-BA**

- **MSCKF 1.0与EKF比较**

主要参见论文《High-precision, consistent EKF-based visual-inertial odometry》



XYZ、AHP、IDP代表三种不同的对特征点的参数化方式，且采用EKF进行信息融合

AHP(anchored homogeneous feature parametrization)

IDP(inverse depth parametrization)

XYZ(“traditional” XYZ feature parametrization)

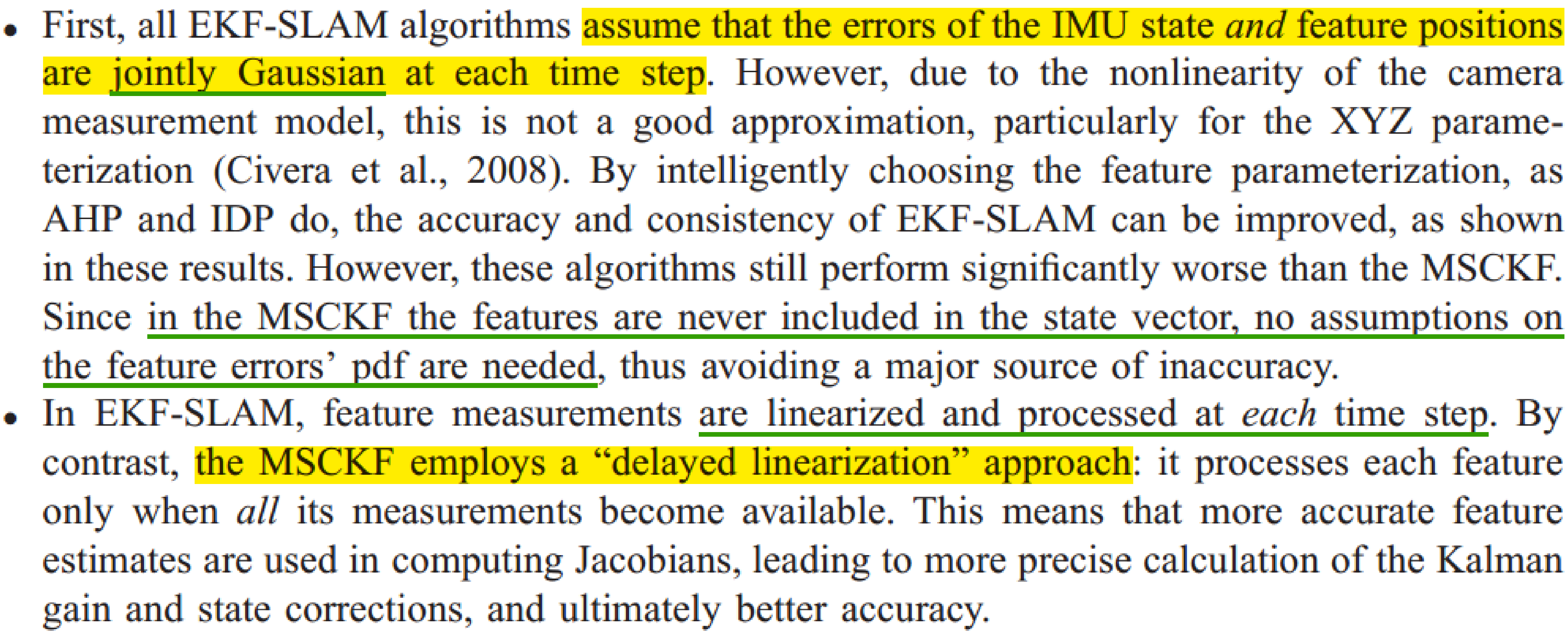
We show that an EKF formulation where the state vector comprises **a sliding window of poses** (the MSCKF algorithm) **attains better accuracy, consistency, and computational efficiency** than the SLAM formulation of the EKF, in which the state vector contains the **current pose and the features** seen by the camera.

- **MSCKF 1.0与EKF比较**

MSCKF 1.0比EKF性能优越原因分析：

1、MSCKF无需对特征点情况进行概率分布假设；

2、延迟线性化方法，则更精确的特征估计用来进行计算



- First, all EKF-SLAM algorithms assume that the errors of the IMU state *and* feature positions are jointly Gaussian at each time step. However, due to the nonlinearity of the camera measurement model, this is not a good approximation, particularly for the XYZ parameterization (Civera et al., 2008). By intelligently choosing the feature parameterization, as AHP and IDP do, the accuracy and consistency of EKF-SLAM can be improved, as shown in these results. However, these algorithms still perform significantly worse than the MSCKF. Since in the MSCKF the features are never included in the state vector, no assumptions on the feature errors' pdf are needed, thus avoiding a major source of inaccuracy.
- In EKF-SLAM, feature measurements are linearized and processed at *each* time step. By contrast, the MSCKF employs a "delayed linearization" approach: it processes each feature only when *all* its measurements become available. This means that more accurate feature estimates are used in computing Jacobians, leading to more precise calculation of the Kalman gain and state corrections, and ultimately better accuracy.

- **MSCKF 2.0 ——李名杨进行的改进**

  Li M, Mourikis A I. **High-precision, consistent EKF-based visual-inertial odometry**[J]. The International Journal of Robotics Research, 2013, 32(6): 690-711.

- 传统EKF和MSCKF 1.0存在观测不一致（*inconsistent*）
- 主要改进包括两点：

1. 采用**FEJ（First Estimates Jacobians）**技巧提升系统观测一致性
2. 将IMU与相机间外参加入状态向量，进行同步在线估计

$$\mathbf{x}_I^{\star} = \begin{bmatrix} {}^I_G\bar{\mathbf{q}}^T & {}^G\mathbf{p}^T & {}^G\mathbf{v}^T & \mathbf{b_g}^T & \mathbf{b_a}^T & {}^C_I\mathbf{p}^T & {}^C_I\bar{\mathbf{q}}^T \end{bmatrix}^T$$

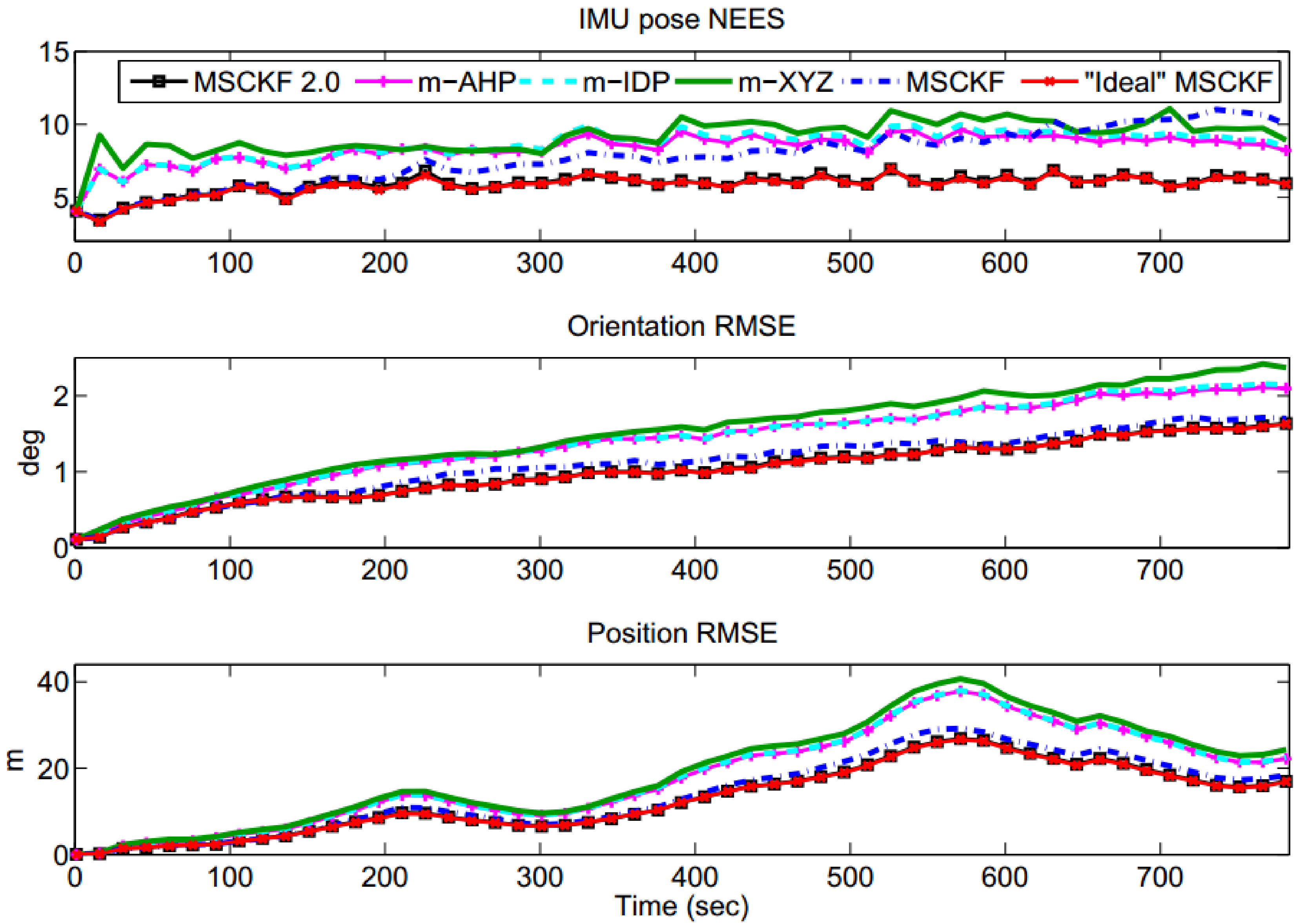
改进的状态向量

$$\begin{aligned} \mathbf{r}_{ij} &= \mathbf{z}_{ij} - \mathbf{h}(\hat{\boldsymbol{\pi}}_{j|\ell-1}, \hat{\boldsymbol{\pi}}_{CI_{\ell|\ell-1}}, \ddot{\mathbf{f}}_i) \\ &\simeq \mathbf{H}_{ij}\tilde{\boldsymbol{\pi}}_{j|\ell-1} + \mathbf{H}_{C_{ij}}\tilde{\boldsymbol{\pi}}_{CI_{\ell|\ell-1}} + \mathbf{H}_{\mathbf{f}_{ij}}\tilde{\mathbf{f}}_i + \mathbf{n}_{ij} \end{aligned}$$

观测方程线性化

This algorithm,which we term MSCKF 2.0, is shown to achieve **accuracy and consistency higher** than even an iterative, sliding-window fixed-lag smoother, in both Monte-Carlo simulations and real-world testing.

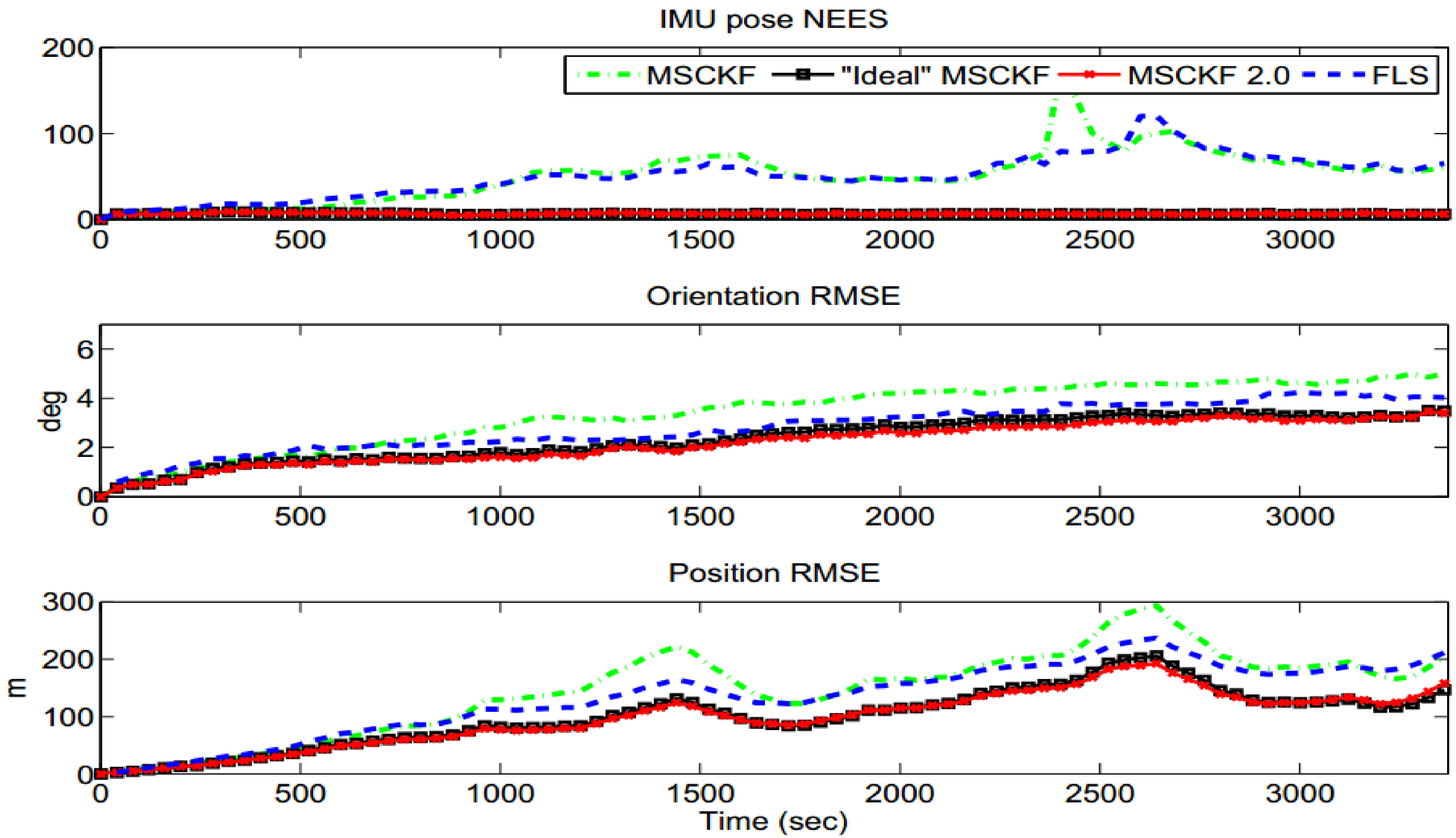
- **MSCKF 2.0 与EKF-based SLAM对比**

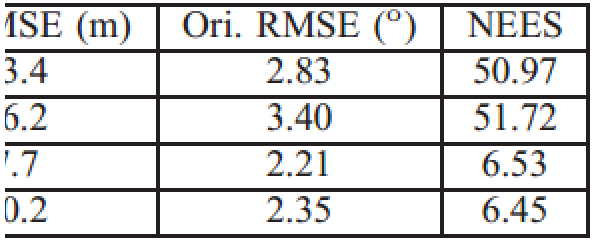


“ideal” MSCKF algorithm:true IMU states and feature positions are used for computing all the filter Jacobians .

- **MSCKF 2.0 与FLS对比**

FLS(information-form fixed-lag smoother)：This is **a sliding-window bundle adjustment method** that marginalizes older states to maintain a constant computational cost.





ISE (m)	Ori. RMSE (°)	NEES
3.4	2.83	50.97
6.2	3.40	51.72
.7	2.21	6.53
0.2	2.35	6.45

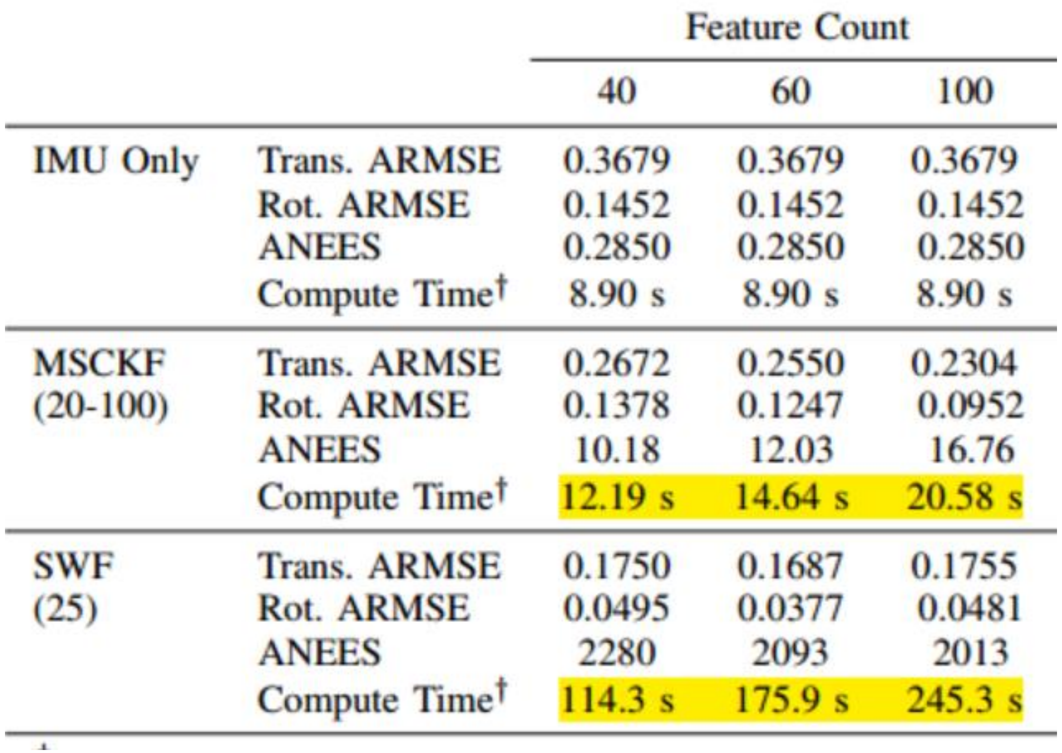
- **MSCKF与SWF比较**
  主要参见论文《The Battle for Filter Supremacy: A Comparative Study of the Multi-State Constraint Kalman Filter and the Sliding Window Filter》

◆ MSCKF优点及适用场景：

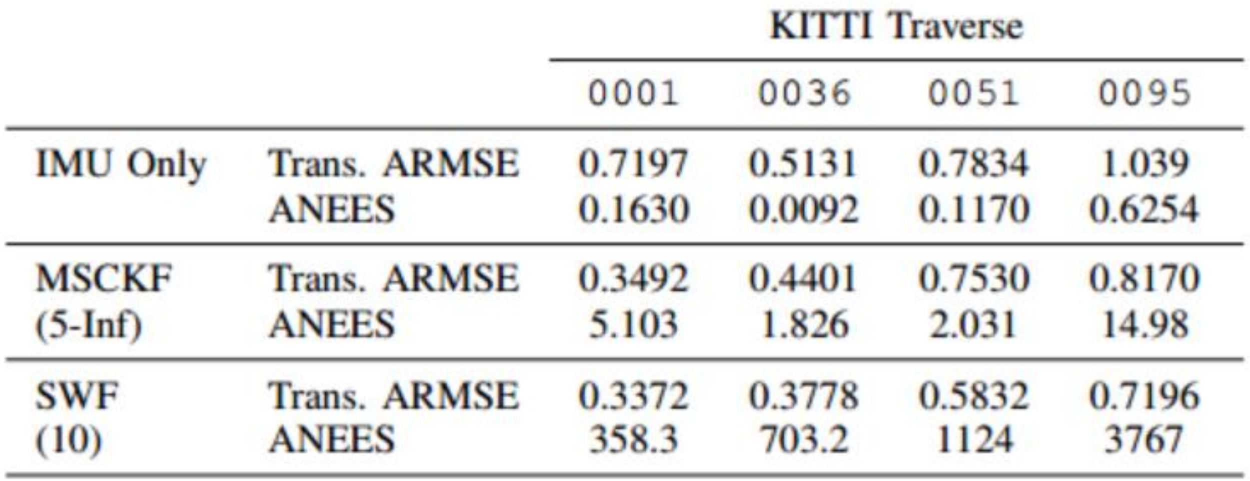
1、计算代价低；　2、更好的一致性特性；　3、适用于**计算资源有限、环境特征**丰富的场景

◆ SWF优点及适用场景：

1、精度更高；　2、对特征质量相对不敏感；　3、对参数敏感度低；　4、适用于鲁棒性要求高的场景



		Feature Count		
		40	60	100
IMU Only	Trans. ARMSE	0.3679	0.3679	0.3679
	Rot. ARMSE	0.1452	0.1452	0.1452
	ANEES	0.2850	0.2850	0.2850
	Compute Time†	8.90 s	8.90 s	8.90 s
MSCKF (20-100)	Trans. ARMSE	0.2672	0.2550	0.2304
	Rot. ARMSE	0.1378	0.1247	0.0952
	ANEES	10.18	12.03	16.76
	Compute Time†	12.19 s	14.64 s	20.58 s
SWF (25)	Trans. ARMSE	0.1750	0.1687	0.1755
	Rot. ARMSE	0.0495	0.0377	0.0481
	ANEES	2280	2093	2013
	Compute Time†	114.3 s	175.9 s	245.3 s

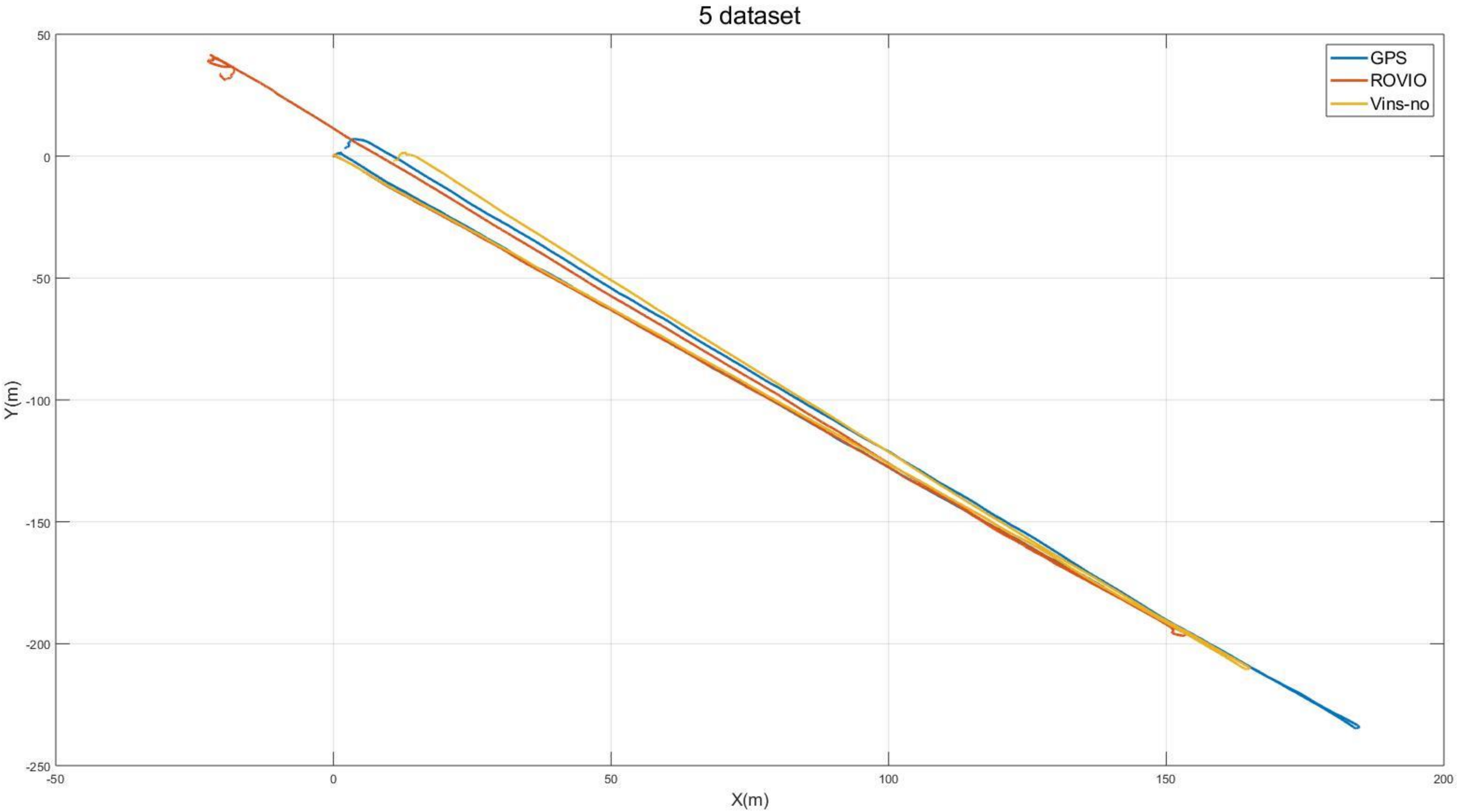


		KITTI Traverse			
		0001	0036	0051	0095
IMU Only	Trans. ARMSE	0.7197	0.5131	0.7834	1.039
	ANEES	0.1630	0.0092	0.1170	0.6254
MSCKF (5-Inf)	Trans. ARMSE	0.3492	0.4401	0.7530	0.8170
	ANEES	5.103	1.826	2.031	14.98
SWF (10)	Trans. ARMSE	0.3372	0.3778	0.5832	0.7196
	ANEES	358.3	703.2	1124	3767

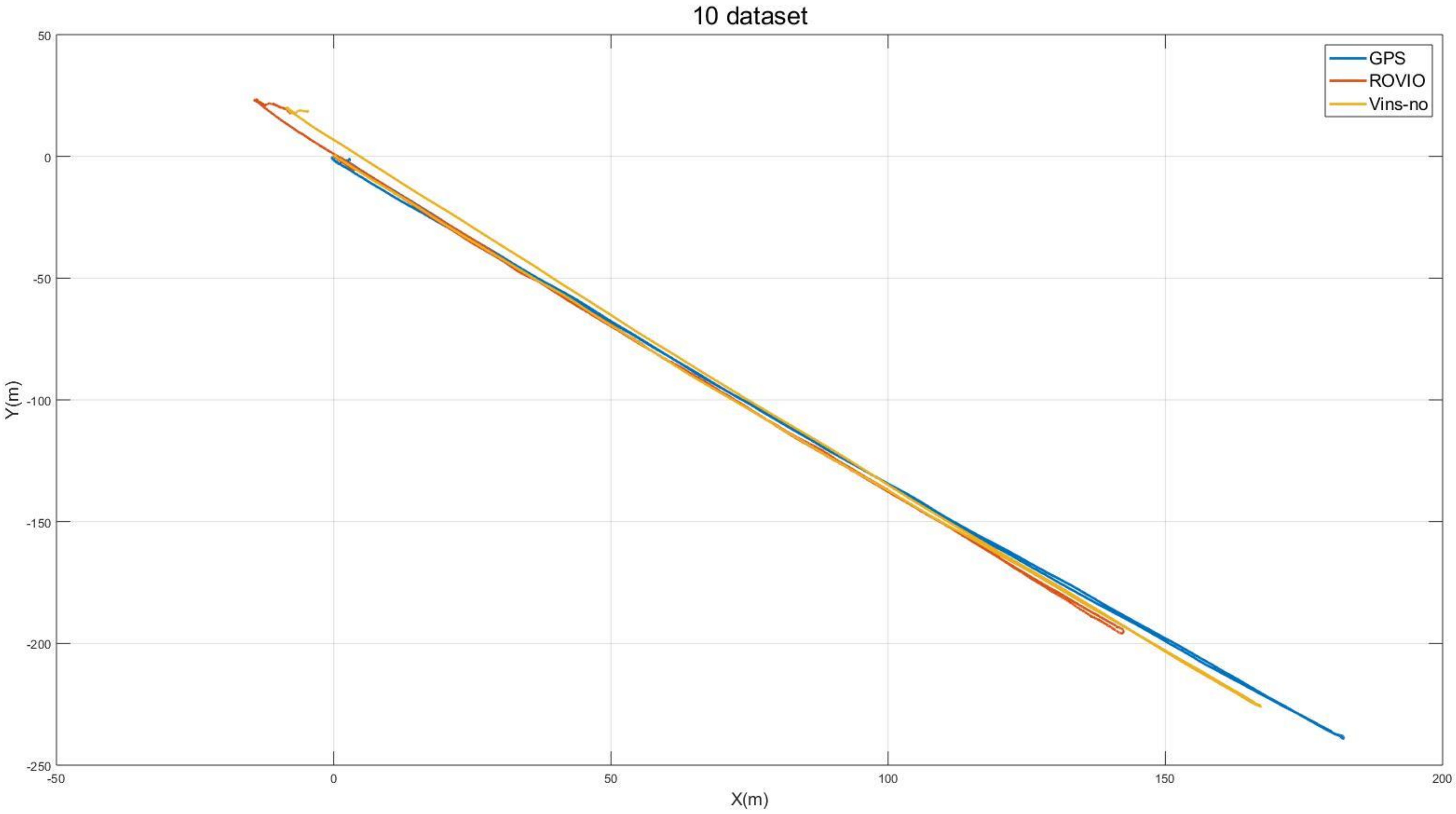
The MSCKF can be thought of as **a hybrid of EKF-SLAM and the SWF** in the sense that it maintains a variable window of poses and applies batch updates using all observations of each landmark.

- ROVIO与VINS-Mono运行SMSCKF室外飞机飞行数据集比较

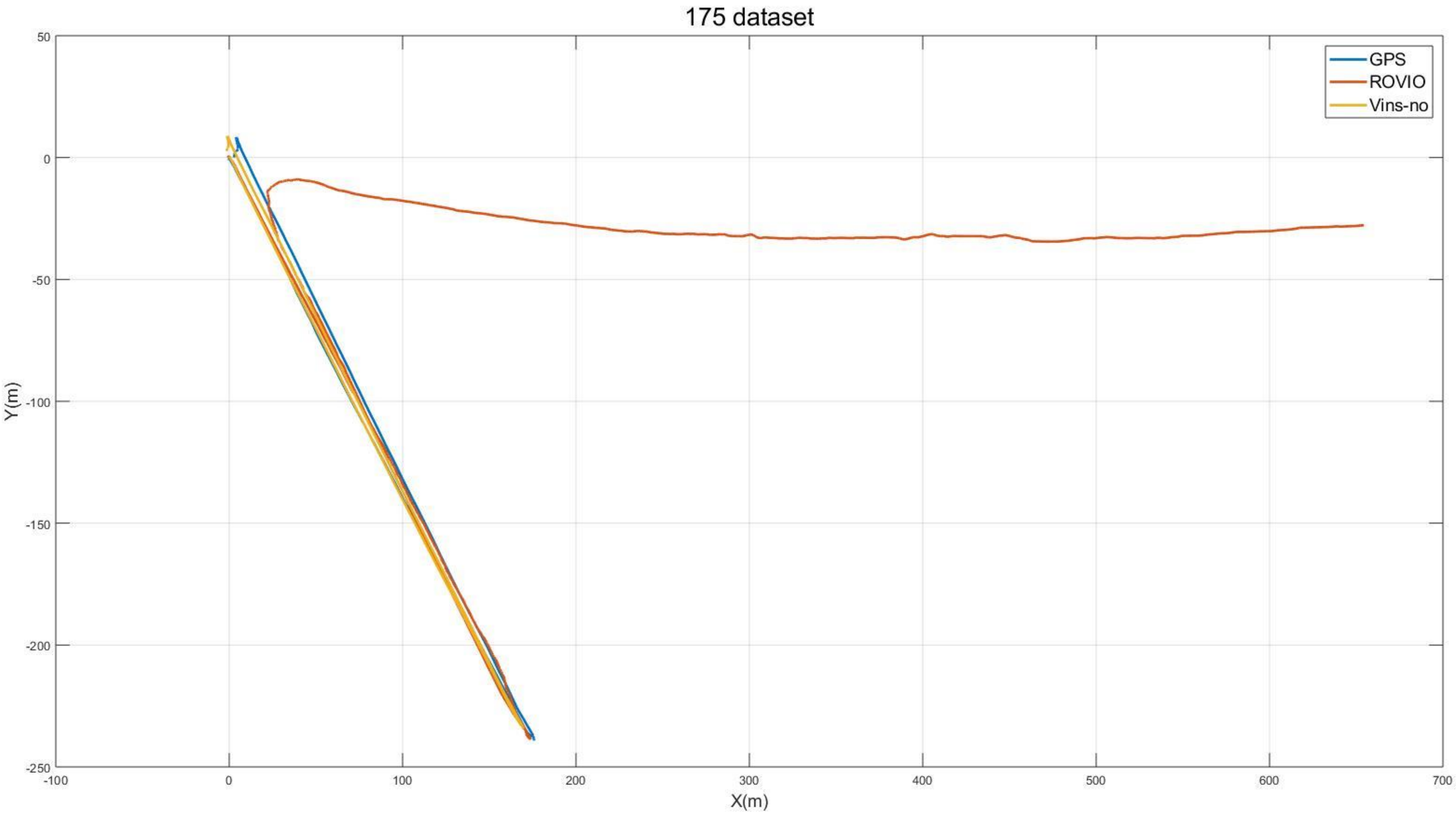
PS：待做，将MSCKF运行该数据集后，对结果进行比对



飞行高度约5.2米，运行速度为5 m/s的数据集的结果

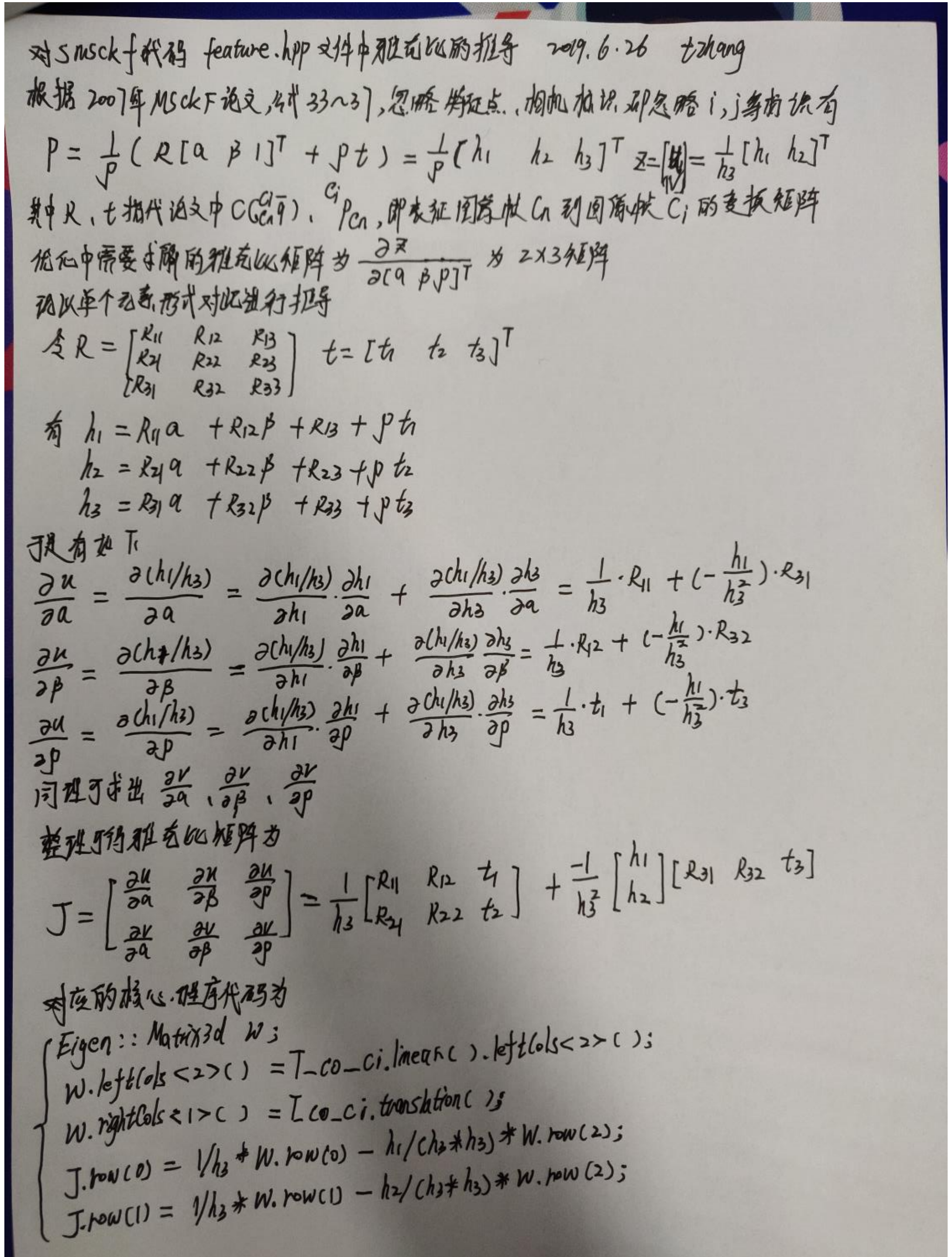


飞行高度约5.2米，运行速度为 10 m/s的数据集的结果



飞行高度约5.2米，运行速度为 17.5 m/s的数据集的结果

路标点优化过程中雅克比矩阵计算



对S-msckf代码 feature.hpp 文件中雅克比的推导 2019.6.26 tzhang

根据2007年MSCKF论文，公式33~37，忽略特征点、相机标识，即忽略 i, j 等标识有

$$P=\frac{1}{\rho}\left(R[\alpha\ \ \beta\ \ 1]^T+\rho t\right)=\frac{1}{\rho}[h_1\ \ h_2\ \ h_3]^T \quad z=\begin{bmatrix}u\\v\end{bmatrix}=\frac{1}{h_3}[h_1\ \ h_2]^T$$

其中R、t指代论文中 $C(^{C_i}_{C_n}\bar{q})$、$^{C_i}p_{C_n}$，即表征图像帧 C_n 到图像帧 C_i 的变换矩阵

优化中需要求解的雅克比矩阵为 $\frac{\partial z}{\partial[\alpha\ \ \beta\ \ \rho]^T}$ 为 2×3矩阵

现以单个元素形式对此进行推导

令 $R=\begin{bmatrix}R_{11} & R_{12} & R_{13}\\ R_{21} & R_{22} & R_{23}\\ R_{31} & R_{32} & R_{33}\end{bmatrix}$ $t=[t_1\ \ t_2\ \ t_3]^T$

有
$$h_1=R_{11}\alpha+R_{12}\beta+R_{13}+\rho t_1$$
$$h_2=R_{21}\alpha+R_{22}\beta+R_{23}+\rho t_2$$
$$h_3=R_{31}\alpha+R_{32}\beta+R_{33}+\rho t_3$$

现有如下：

$$\frac{\partial u}{\partial\alpha}=\frac{\partial(h_1/h_3)}{\partial\alpha}=\frac{\partial(h_1/h_3)}{\partial h_1}\cdot\frac{\partial h_1}{\partial\alpha}+\frac{\partial(h_1/h_3)}{\partial h_3}\cdot\frac{\partial h_3}{\partial\alpha}=\frac{1}{h_3}\cdot R_{11}+\left(-\frac{h_1}{h_3^2}\right)\cdot R_{31}$$

$$\frac{\partial u}{\partial\beta}=\frac{\partial(h_1/h_3)}{\partial\beta}=\frac{\partial(h_1/h_3)}{\partial h_1}\cdot\frac{\partial h_1}{\partial\beta}+\frac{\partial(h_1/h_3)}{\partial h_3}\frac{\partial h_3}{\partial\beta}=\frac{1}{h_3}\cdot R_{12}+\left(-\frac{h_1}{h_3^2}\right)\cdot R_{32}$$

$$\frac{\partial u}{\partial\rho}=\frac{\partial(h_1/h_3)}{\partial\rho}=\frac{\partial(h_1/h_3)}{\partial h_1}\cdot\frac{\partial h_1}{\partial\rho}+\frac{\partial(h_1/h_3)}{\partial h_3}\cdot\frac{\partial h_3}{\partial\rho}=\frac{1}{h_3}\cdot t_1+\left(-\frac{h_1}{h_3^2}\right)\cdot t_3$$

同理可求出 $\frac{\partial v}{\partial\alpha}$、$\frac{\partial v}{\partial\beta}$、$\frac{\partial v}{\partial\rho}$

整理可得雅克比矩阵为

$$J=\begin{bmatrix}\frac{\partial u}{\partial\alpha} & \frac{\partial u}{\partial\beta} & \frac{\partial u}{\partial\rho}\\ \frac{\partial v}{\partial\alpha} & \frac{\partial v}{\partial\beta} & \frac{\partial v}{\partial\rho}\end{bmatrix}=\frac{1}{h_3}\begin{bmatrix}R_{11} & R_{12} & t_1\\ R_{21} & R_{22} & t_2\end{bmatrix}+\frac{-1}{h_3^2}\begin{bmatrix}h_1\\h_2\end{bmatrix}[R_{31}\ \ R_{32}\ \ t_3]$$

对应的核心程序代码为

```
Eigen::Matrix3d W;
W.leftCols<2>() = T_co_ci.linear().leftCols<2>();
W.rightCols<1>() = T_co_ci.translation();
J.row(0) = 1/h3 * W.row(0) - h1/(h3*h3) * W.row(2);
J.row(1) = 1/h3 * W.row(1) - h2/(h3*h3) * W.row(2);
```

- 参考文献

1. Mourikis A I, Roumeliotis S I. A multi-state constraint Kalman filter for vision-aided inertial navigation [C] // Proceedings 2007 IEEE International Conference on Robotics and Automation. IEEE, 2007: 3565-3572.
2. Li M, Mourikis A I. High-precision, consistent EKF-based visual-inertial odometry[J]. The International Journal of Robotics Research, 2013, 32(6): 690-711.
3. Clement L E, Peretroukhin V, Lambert J, et al. The battle for filter supremacy: A comparative study of the multi-state constraint kalman filter and the sliding window filter[C]//2015 12th Conference on Computer and Robot Vision. IEEE, 2015: 23-30.
4. Wu K, Ahmed A, Georgiou G A, et al. A Square Root Inverse Filter for Efficient Vision-aided Inertial Navigation on Mobile Devices[C]//Robotics: Science and Systems. 2015, 2.
5. Barfoot T D. State Estimation for Robotics[M]. Cambridge University Press, 2017.
6. Paul M K, Wu K, Hesch J A, et al. A comparative analysis of tightly-coupled monocular, binocular, and stereo vins[C]//2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017: 165-172.
7. Sola J. Quaternion kinematics for the error-state Kalman filter[J]. arXiv preprint arXiv:1711.02508, 2017.
8. Sun K, Mohta K, Pfrommer B, et al. Robust stereo visual inertial odometry for fast autonomous flight[J]. IEEE Robotics and Automation Letters, 2018, 3(2): 965-972.

THANKS