STATE ESTIMATION IN MULTI-SENSOR FUSION NAVIGATION: EQUIVALENCE ANALYSIS ON FILTERING AND OPTIMIZATION

Zhuo Xu [1], Feng Zhu [1 *], Xiaohong Zhang [2]

[1] School of Geodesy and Geomatics, Wuhan University, Wuhan 430079, China – (zhuoxu, fzhu)@whu.edu.cn
[2] Chinese Antarctic Center of Surveying and Mapping, Wuhan University, Wuhan 430079, China – xhzhang@sgg.whu.edu.cn

KEY WORDS: Multi-Sensor Fusion, State Estimation, Extended Kalman Filter, Factor Graph Optimization, Visual Odometry.

ABSTRACT:

Integration of information from multi-sensor can provide navigation systems with reliable estimates of their own states, which overcomes shortage of standalone sensors. State estimation approaches, which can be categorized into filtering-based and optimization-based methods, provide the means to fuse information from various sensors with different principles to estimate the system's position, orientation, and other navigation parameters accurately. Recent researches have shown that optimization-based frameworks outperform filtering-based ones in terms of accuracy. However, both methods are based on maximum likelihood estimation (MLE) and, assuming Gaussian noise, should be theoretically equivalent. In this paper, we comprehensively and theoretically analyse the differences between the two methods, including algorithms and strategies. Our simulated experiments based on visual odometry (VO) indicate that filtering-based approaches are equal to optimization-based ones in accuracy when employing the same strategies and under the premise of the same measurements and observation model. Therefore, future research on sensor-fusion navigation problems should concentrate on strategies rather than state estimation methods.

1. INTRODUCTION

Multi-sensor fusion, which combines sensors with different principles, is widely used due to the complementary advantages of their measurements, providing seamless, reliable, and high-precision positioning with emerging applications, such as autonomous vehicles and mobile mapping, where standalone sensors are insufficient. By appropriately modelling the observations of each sensor, multi-sensor fusion tasks are reduced to a state estimation problem that can be classified into two methods: filtering-based and optimization-based (Strasdat et al., 2010).

In the early ages, filtering-based approaches are widely used due to its high efficiency, since Kalman filter (KF) updates states with measurements sequentially inputting. Its real-time applications in sensor fusion of GNSS (Global Navigation Satellite System) and IMU (Inertial Measurements Unit) can be traced back to 90s (Grejner-Brzezinska et al., 1998). Furthermore, many excellent frameworks concentrating on SLAM (Simultaneous Localization And Mapping) have been proposed based on KF. MonoSLAM (Davison et al., 2007), which is the earliest visual SLAM that performs in real-time, estimates camera frames and landmarks by an extended Kalman filter (EKF). Gmapping adopts a particle filter to localize and map in unknown environments using a 2D lidar (Grisetti et al., 2007). Then, multi-state constrained Kalman filter (MSCKF), a benchmark of sliding window filter (SWF), was proposed to integrate camera and IMU, which maintains historic camera poses in the state vector to construct constraints by using measurements of the same landmarks across multiple cameras (Mourikis and Roumeliotis, 2007). The idea of multi-state constraints is still applied in nowadays multi-sensor fusion which integrates camera, lidar, IMU, and GNSS (Li et al., 2023).

Meanwhile, optimization-based, or graph-based, approaches optimize all measurements by solving a nonlinear least square (NLS) problem, which is time-consuming compared with filtering-based ones (Leutenegger et al., 2013). To bound computational complexity, one common strategy is to maintain a bounded-size window and marginalize out past states and measurements (Sibley et al., 2010; Yang et al., 2017), which is so-called sliding window optimization (SWO). However, marginalization corrupts sparsity in the system which has an adverse effect on efficiency. Thus, efforts are made on node removal strategies and sparsification to avoid densifying when marginalize (Eckenhoff et al., 2016; Mazuran et al., 2014; Vial et al., 2011). On the other hand, incremental smoothing using a Bayes tree has been proposed to accelerate states estimation (Kaess et al., 2011). Based on these strategies, research show that optimization-based frameworks can also operate in real-time, whether using a computer or a mobile phone (Campos et al., 2021; Qin et al., 2018; Shan et al., 2020), and perform better than filtering-based ones, where the same sensors are used (Strasdat et al., 2010). Therefore, optimization-based approaches are preferred in recent research due to the better performance in accuracy (Cao et al., 2021; Huang, 2019; Niu et al., 2023).

However, despite strategies and graph representations used in optimization-based methods, system states are estimated by NLS. It is important to note that, both NLS and EKF can be derived from MLE by assuming Gaussian noise. Furthermore, by modelling observations from sensor equivalently, EKF can be derived from NLS without additional assumptions, which indicates that EKF should be equivalent to NLS in terms of accuracy (Bell and Cathey, 1993). Thus, filtering-based approaches should achieve the same performance with optimization-based ones. Differences between the both state estimation methods are caused by different strategies rather than state estimation approaches.

In this paper, we comprehensively and theoretically analyse the differences between both the filtering-based approach and the optimization-based one from three aspects: *theory*, *Jacobi*, and *strategies*, which covers all factors that may affect estimation results. Meanwhile, simulated visual odometry experiments are conducted to evaluate estimating results between the two state estimation results. Specifically, the main contributions of this work include:

1. We analytically show that, when system states contain all frames and landmarks in the whole trajectory, reasons caused differences between the two estimators lie in linearization points. And by applying the same linearization points, both estimators achieve the same results in accuracy.
2. We show that, with sliding window applied in real-time operation, both linearization points and prior knowledge corrupts accuracy consistency of SWF and SWO due to different strategies used when window slides. Due to utilization of observations, SWO performs better than SWF. But by a three-step modification on SWF, the modified SWF is equal to SWO in accuracy.
3. Marginalization in SWO, which operates on information matrix, equals to deleting covariance in SWF if the same observations are utilized. Both strategies convey constraints to the next window without information loss.

The rest of this paper is structured as follows. In Section 2, we firstly review that NLS can be derived from MLE, and EKF can be derived from NLS. Then, Jacobian matrix and common strategies used in optimization-based and filtering-based methods are theoretically analysed. In Section 3, strategies used in SWO are applied in SWF by a 3-step modification. Section 4 validates our analysis based on simulation experiments. Finally, Section 5 concludes the paper and outline future works.

2. THEORETICAL ANALYSIS OF FILTERING-BASED AND OPTIMIZATION-BASED APPROACHES

In this section, three aspects are analysed theoretically. We first derive the formula of optimization and EKF from MLE. Then, differences in the Jacobian matrix caused by linearization points are analysed. Lastly, strategies that are most commonly used in real-time operations are analysed for both filtering-based and optimization-based approaches. Without loss of generality, all these analyses are based on visual odometry (VO).

2.1 Derivation from MLE to Optimization and KF

VO recovers camera pose $\boldsymbol{x}_{f,j}$ by matched features in camera frames, which includes 3-DOF rotation and 3-DOF translation at time t_j. Observations can be modelled as follow:

$$\boldsymbol{z}_i = \boldsymbol{h}_i\left(\boldsymbol{x}_{f,j}, \boldsymbol{x}_{l,k}\right) + \boldsymbol{\Delta}_i \tag{1}$$

Where $\boldsymbol{h}_i$ maps the *k*-th landmark $\boldsymbol{x}_{l,k}$ into measurements $\boldsymbol{z}_i$ on image plane. $\boldsymbol{\Delta}_i \sim \boldsymbol{N}(\boldsymbol{0}, \boldsymbol{d}_i)$ is zero-mean white Gaussian noise. With stacked measurements, camera poses can be estimated by maximum likelihood estimation:

$$\hat{\boldsymbol{X}} = \arg\max_{\boldsymbol{X}} \prod_{\substack{1\le j\le n, 1\le k\le m \\ i\in \boldsymbol{S}}} p\left(\boldsymbol{z}_i \mid \boldsymbol{x}_{f,j}; \boldsymbol{x}_{l,k}\right) \tag{2}$$

Where, *n* and *m* are number of camera poses and landmarks respectively. $\boldsymbol{S}$ is the set containing all observations. For priori information $\tilde{\boldsymbol{X}}_p$, it can be treated as virtual measurements with $\boldsymbol{\Delta}_p \sim \boldsymbol{N}\left(\boldsymbol{0}, \boldsymbol{D}_p\right)$:

$$\boldsymbol{z}_p = \tilde{\boldsymbol{X}}_p + \boldsymbol{\Delta}_p \tag{3}$$

Due to the assumption of Gaussian noise, the NLS, which is generally utilized in optimization methods, can be derived from MLE:

$$\begin{aligned} &\left(\boldsymbol{D}_p^{-1} + \boldsymbol{J}^T\boldsymbol{D}^{-1}\boldsymbol{J}\right)\delta\hat{\boldsymbol{X}} = \boldsymbol{D}_p\tilde{\boldsymbol{X}}_p + \boldsymbol{J}^T\boldsymbol{D}^{-1}\boldsymbol{L} \\ &\boldsymbol{J} = \begin{bmatrix}\boldsymbol{J}_1^T & \cdots & \boldsymbol{J}_i^T\end{bmatrix}^T, \boldsymbol{L} = \begin{bmatrix}\boldsymbol{l}_1^T & \cdots & \boldsymbol{l}_i^T\end{bmatrix}^T, \\ &\boldsymbol{D} = diag\begin{pmatrix}\boldsymbol{d}_1 & \cdots & \boldsymbol{d}_i\end{pmatrix}, i \in \boldsymbol{S} \end{aligned} \tag{4}$$

Where, $\boldsymbol{J}_i$ is the Jacobian matrix of mapping function $\boldsymbol{h}_i$ evaluated at the initial state $\boldsymbol{X}_0$. $\boldsymbol{l}_i = \boldsymbol{z}_i - \boldsymbol{h}_i$ represents residuals. And the estimating error-state vector $\delta\hat{\boldsymbol{X}}$ is defined as:

$$\delta\hat{\boldsymbol{X}} = \begin{bmatrix}\delta\hat{\boldsymbol{x}}_{f,1}^T & \cdots & \delta\hat{\boldsymbol{x}}_{f,n}^T & \delta\hat{\boldsymbol{x}}_{l,1}^T & \cdots & \delta\hat{\boldsymbol{x}}_{l,m}^T\end{bmatrix}^T \tag{5}$$

Then, the full-state vector can be derived by compensating the estimated error-state vector:

$$\begin{aligned} \hat{\boldsymbol{X}} &= \begin{bmatrix}\hat{\boldsymbol{x}}_{f,1}^T & \cdots & \hat{\boldsymbol{x}}_{f,n}^T & \hat{\boldsymbol{x}}_{l,1}^T & \cdots & \hat{\boldsymbol{x}}_{l,m}^T\end{bmatrix}^T \\ &= \boldsymbol{X}_0 \ominus \delta\hat{\boldsymbol{X}} \end{aligned} \tag{6}$$

Where $\ominus$ corresponds to operation on updating states including rotation and translation. Furthermore, by consideration of posterior covariance $\hat{\boldsymbol{D}}^{-1} = \boldsymbol{D}_p^{-1} + \boldsymbol{J}^T\boldsymbol{D}^{-1}\boldsymbol{J}$, Eq.(4) can be rewritten:

$$\delta\hat{\boldsymbol{X}} = \tilde{\boldsymbol{X}}_p + \hat{\boldsymbol{D}}\boldsymbol{J}^T\boldsymbol{D}^{-1}\left(\boldsymbol{L} - \boldsymbol{J}\tilde{\boldsymbol{X}}_p\right) \tag{7}$$

Since parameters are compensated once the error state $\delta\hat{\boldsymbol{X}}$ is estimated, the priori $\tilde{\boldsymbol{X}}_p$ is set to $\boldsymbol{0}$. Therefore, Eq.(7) is equal to EKF update, which shows when select the same observation model and measurements, both optimization-based and filtering-based methods are equivalent in terms of accuracy by stacking all measurements.

$$\delta\hat{\boldsymbol{X}} = \hat{\boldsymbol{D}}\boldsymbol{J}^T\boldsymbol{D}^{-1}\boldsymbol{L} \tag{8}$$

2.2 Analysis on Jacobi consistency

Note that estimating state $\boldsymbol{X}$ contains all camera poses and positions of landmarks in the whole trajectory, which is hard to realize in real-time operation. However, to avoid effects caused by other strategies, state $\boldsymbol{X}$ is kept in this section for analysis. Meanwhile, strategies applied in real-time operation will be discussed in section 2.3.

In optimization approaches, stacked measurements are used to update $\boldsymbol{X}$, which makes the most of information in observations. For *s*-th iteration, Jacobian matrix of optimization has the form:

$$\boldsymbol{J}_O^{(s)}\big|_{\boldsymbol{X}=\hat{\boldsymbol{X}}^{(s-1)}} = \begin{bmatrix}\boldsymbol{J}_1^{(s)T}\big|_{\boldsymbol{X}=\hat{\boldsymbol{X}}^{(s-1)}} & \cdots & \boldsymbol{J}_i^{(s)T}\big|_{\boldsymbol{X}=\hat{\boldsymbol{X}}^{(s-1)}}\end{bmatrix}^T \tag{9}$$

Where the Jacobian matrix is evaluated at estimating results of previous iteration, which is consistent due to the same accuracy of linearization points, as shown in Figure 1. Then, the linearized form of observation model in optimization can be written as:

$$\boldsymbol{Z} = \boldsymbol{J}_O^{(s)}\hat{\boldsymbol{X}}_O^{(s)} \tag{10}$$

However, EKF updates estimating states $\boldsymbol{X}$ using all observations by inputting them epoch by epoch instead of stacking. Linearization points for each update should be derived from states of previous update. Then stack Jacobi in each update, the whole Jacobian matrix can be written as:

$$\boldsymbol{J}_{KF} = \left[\boldsymbol{J}_1^T \big|_{\boldsymbol{X}=\boldsymbol{X}_0} \quad \cdots \quad \boldsymbol{J}_i^T \big|_{\boldsymbol{X}=\hat{\boldsymbol{X}}_{i-1}} \right]^T \tag{11}$$

On the contrary, the Jacobian matrix of KF is inconsistent due to the different linearization point, as shown in Figure 1. Similarly, the linearized form of observation model in KF is:

$$\begin{aligned} \boldsymbol{Z} &= \boldsymbol{J}_{KF} \hat{\boldsymbol{X}}_{KF} \\ &= \left(\boldsymbol{J}_O^{(s)} + \Delta \boldsymbol{J} \right) \left(\hat{\boldsymbol{X}}_O^{(s)} + \Delta \boldsymbol{X} \right) \end{aligned} \tag{12}$$

Where $\Delta \boldsymbol{J}$ and $\Delta \boldsymbol{X}$ are different part of Jacobi and wrongly estimated states due to different linearization points respectively. Then, wrongly estimated part of states $\Delta \boldsymbol{X}$ can be derived by:

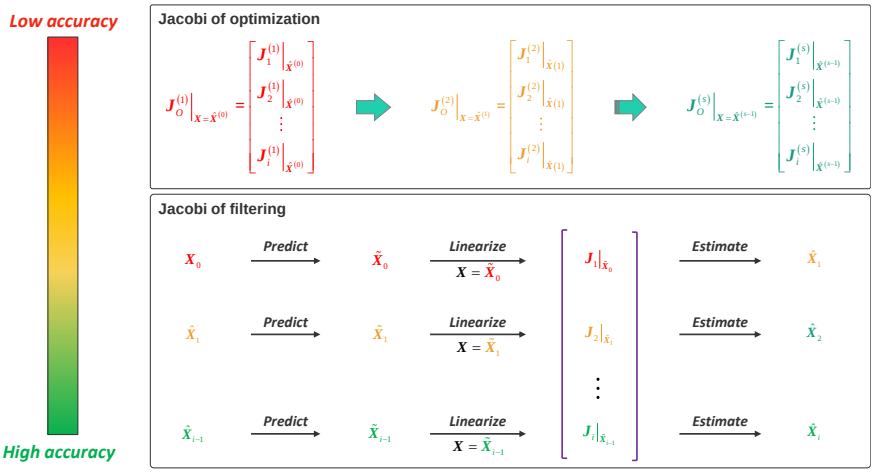


Figure 1. Jacobian matrix of optimization and EKF.

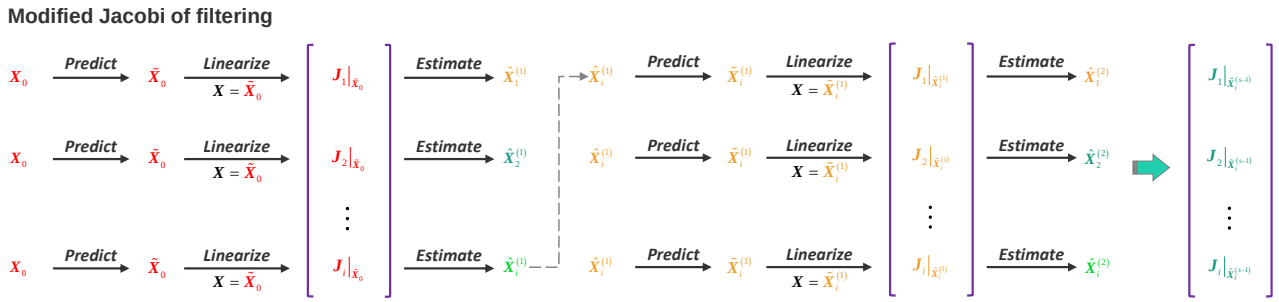


Figure 2. Modified Jacobian matrix of EKF.

$$\Delta \boldsymbol{X} = -\left(\boldsymbol{J}_O^{(s)} + \Delta \boldsymbol{J} \right)^{-g} \Delta \boldsymbol{J} \hat{\boldsymbol{X}}_O^{(s)} \tag{13}$$

Where $-g$ is the pseudo-inverse of matrix. Since $|\Delta \boldsymbol{J}|$ is proportional to the degree of non-linearity, Eq.(13) indicates that the difference between optimization and KF depends on degree of nonlinearity of the observation model. So, the linearization points are essential to problems with high degree of nonlinearity.

Thus, instead of correcting $\delta \hat{\boldsymbol{X}}$ to full states after each update in EKF, full states are to be updated when all measurements are utilized, which ensures the same linearization points at each filtering. After all measurements are used, update the full states $\boldsymbol{X}$. Then, as shown in Figure 2, do iterations which has the same number of the optimization. Note that, error states should be estimated by Eq.(7) since $\delta \hat{\boldsymbol{X}}$ is not updated. As a result, the Jacobian matrix during the s-th iteration of EKF can be expressed as:

$$\boldsymbol{J}_{KF}^{(s)} \big|_{\boldsymbol{X}=\hat{\boldsymbol{X}}_s} = \left[\boldsymbol{J}_1^{(s)\,T} \big|_{\boldsymbol{X}=\hat{\boldsymbol{X}}_{s-1}} \quad \cdots \quad \boldsymbol{J}_i^{(s)\,T} \big|_{\boldsymbol{X}=\hat{\boldsymbol{X}}_{s-1}} \right]^T \tag{14}$$

Eq.(14) indicates that, with appropriately chosen linearization point, both estimators can achieve equivalent estimating results in terms of accuracy when state vector contains all variables in the whole trajectory.

2.3 Analysis on strategies

In practical applications, the real-time performance of navigation systems is a primary concern, leading to the frequent use of sliding windows to balance efficiency and accuracy. In this section, the visual strategies employed by two representative methods: MSCKF (Mourikis and Roumeliotis, 2007) and VINS-mono (Qin et al., 2018), which respectively belongs to sliding window filter (SWF) and sliding window optimization (SWO), will be analysed. These strategies can be categorized into three aspects: utilization of observations, nodes removing strategy, as well as constraints transmission. And comparisons between SWF and SWO can be referred to Figure 3.

2.3.1 Strategies in MSCKF

A. utilization of observations

For MSCKF update, landmarks will be used if one of those conditions are satisfied[1]:

1. Landmarks are no longer observed by latest frame;
2. Landmarks are observed by three frames.

[1] Strategies are simplified for discussion.

This strategy makes full use of observations and tracks landmarks as long as possible. Therefore, as shown in Figure 3, only $x_{l,1}$ is used to filter in first window, since it is not observed by the latest frame. In the next window, $x_{l,2}$ and $x_{l,3}$ are landmarks not observed by the latest frame, and $x_{l,4}$ is observed for three frames. Thus, three landmarks are used to update states in this window.

B. nodes removing strategy

The oldest frame is to be removed from window to bound computational complexity. Meanwhile, landmarks no longer observed by frames in the window and all used landmarks should also be removed in order to avoid repeatedly using. In other words, observations are used only *once* in SWF.

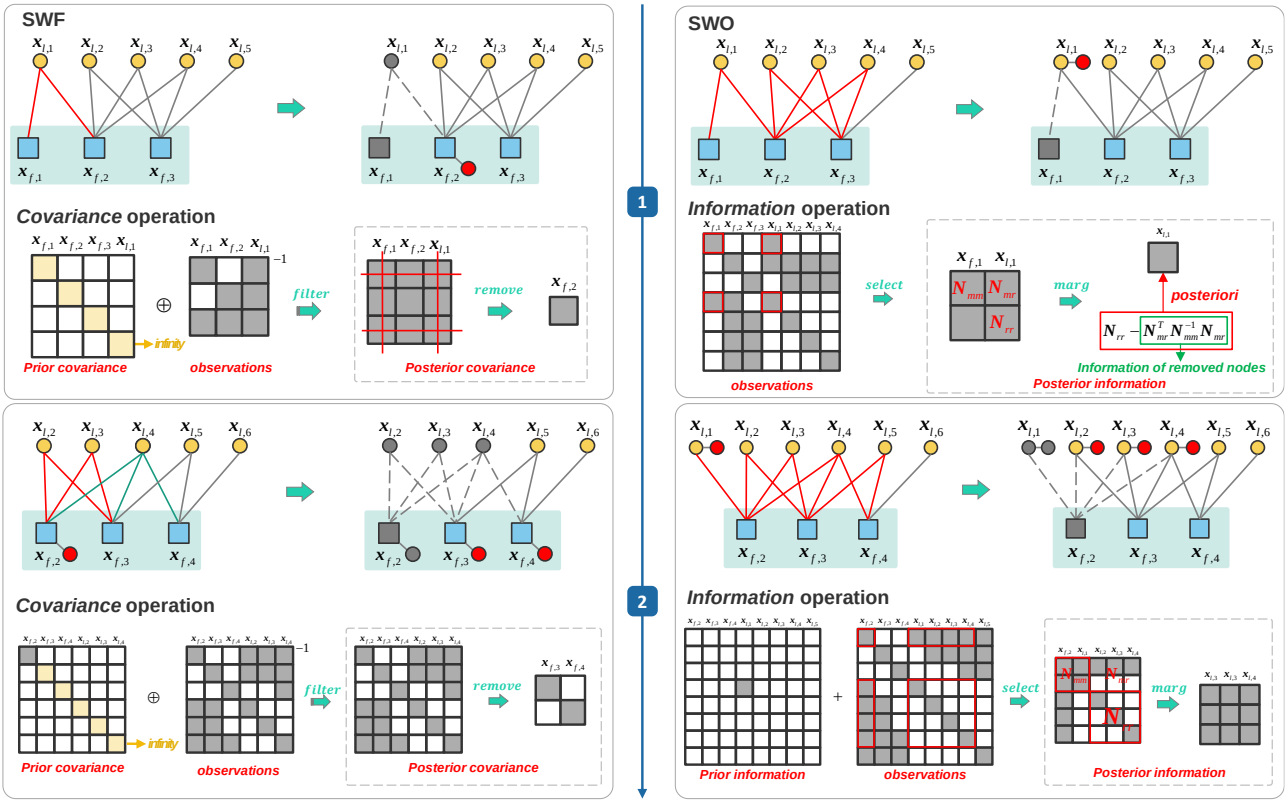


Figure 3. Comparison of SWF and SWO methods based on MSCKF and VINS-mono.

Where, $\oplus$ represents covariance operation: $(\cdot)\oplus(\cdot)=\left[(\cdot)^{-1}\oplus(\cdot)^{-1}\right]^{-1}$.

Thus, after first window finishes filtering, the oldest frame, $x_{f,1}$, and landmarks have been utilized as well as their observations in the window, $x_{l,1}$, will be removed. Then in the second window, $x_{l,2}$, $x_{l,3}$, $x_{l,4}$ are utilized so that the oldest frame $x_{f,2}$ as well as landmarks with their observations should be removed.

C. constraints transmission

In the first window, due to lack of prior knowledge of each state, diagonal elements of covariance are set to infinity and non-diagonal elements are set to 0, which represents high uncertainty of each state and there is no correlation between estimating states.

After $x_{l,1}$ is used in the first window, covariance of $x_{l,1}$ and its connected states will be updated. However, to bound window size when window slides, the oldest frame, $x_{f,1}$, and landmarks no longer observed by frames in the window, $x_{l,1}$, will be removed. Thus, covariance of removed states is to be directly deleted and only $x_{f,2}$ has its posterior covariance. Then in the second window, $x_{f,2}$ can be constrained by posteriori produced in previous window, and other estimating states remain uncertain. Likewise, after observations connected to $x_{l,2}$, $x_{l,3}$, $x_{l,4}$ are used to filter, oldest frame and no longer used landmarks are removed from window. Therefore, $x_{f,3}$ and $x_{f,4}$ will be constrained in the next filtering.

2.3.2 Strategies in VINS-mono

A. utilization of observations

In optimization-based methods, all tracked landmarks, observed by at least two frames, with their connected observations can be utilized to estimate. Therefore, all landmarks except $x_{l,5}$, which is only observed by $x_{f,3}$, with their observations are used. Likewise, in the second window, all landmarks but $x_{l,6}$ are used.

B. nodes removing strategy

Similarly to SWF, the oldest frame and its corresponding observations are always removed from the sliding window. However, landmarks will only be eliminated when they are no longer observed by any frames in the window.

Therefore, after states estimated in the first window, oldest frame, $x_{f,1}$, with its observation, connected to $x_{l,1}$, are removed. All landmarks will remain as they are connected to frames that still exist within the window. Similarly, $x_{f,2}$ with its observations as well as $x_{l,1}$ which is no longer observed by any remaining frames, are removed in the second window.

C. constraints transmission

Instead of covariance matrix, the information matrix, which represents the weights of each estimated state and is the inverse of covariance, is utilized to constrain the estimation of states in SWO. Without priori in the first window, the information matrix of first window is set to **0**. After estimation, posterior information is obtained. And then marginalization strategy is applied when window slides.

There are three steps in marginalization. Firstly, solve posteriori. The posterior information is the sum of a priori and information matrix of observation which can be derived by Jacobi $\boldsymbol{N}=\boldsymbol{J}_{SWO}^{T}\boldsymbol{J}_{SWO}$. And then, select elements of to be removed nodes and their connections in the posterior information matrix. In the first window, $\boldsymbol{x}_{f,1}$ is to be removed and $\boldsymbol{x}_{l,1}$ is connected to it, so that selected matrix is consisted of nodes $\boldsymbol{x}_{f,1}$ and $\boldsymbol{x}_{l,1}$. Finally, marginalize nodes to be removed. We denote nodes that will be removed as $\boldsymbol{x}_m$, and rest nodes in the selected information matrix as $\boldsymbol{x}_r$. According to Eq.(4), we have:

$$\begin{bmatrix}\boldsymbol{N}_{mm} & \boldsymbol{N}_{mr}\\ \boldsymbol{N}_{mr}^{T} & \boldsymbol{N}_{rr}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_m\\ \boldsymbol{x}_r\end{bmatrix}=\begin{bmatrix}\boldsymbol{b}_m\\ \boldsymbol{b}_r\end{bmatrix} \tag{15}$$

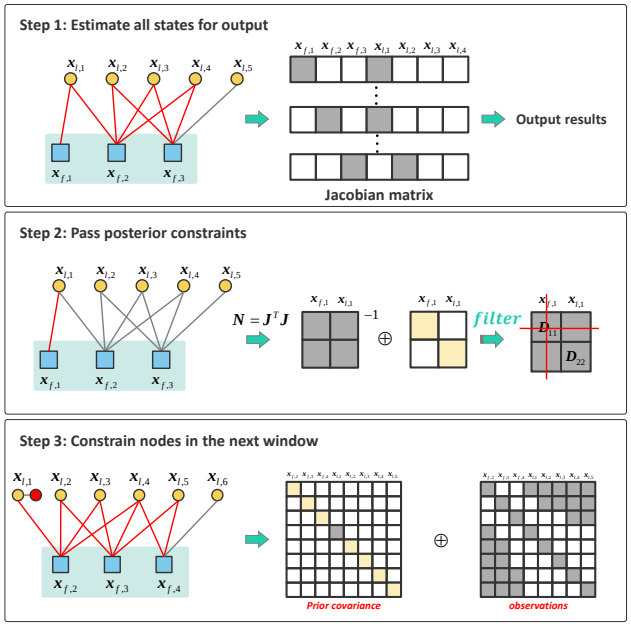


Figure 4. Workflow of modified SWF for once update.

The posterior information matrix of $\boldsymbol{x}_r$ can be easily obtained by Schur complement:

$$\boldsymbol{N}_r=\boldsymbol{N}_{mm}-\boldsymbol{N}_{mr}^{T}\boldsymbol{N}_{rr}\boldsymbol{N}_{mr} \tag{16}$$

Where, $\boldsymbol{N}_{mr}^{T}\boldsymbol{N}_{rr}\boldsymbol{N}_{mr}$ contains information of removed nodes. Thus, $\boldsymbol{x}_r$ is constrained by $\boldsymbol{N}_r$ in the next window without information loss. As illustrated in Figure 3, elements of $\boldsymbol{x}_{l,1}$ in prior information matrix is set to $\boldsymbol{N}_r$ and others remain 0 in the second window. After estimation, since $\boldsymbol{x}_{f,2}$, $\boldsymbol{x}_{l,1}$ are to be removed from window, both nodes and their connections are selected to construct information matrix as in Eq.(15). Then, constraints on $\boldsymbol{x}_{l,2}$, $\boldsymbol{x}_{l,3}$, and $\boldsymbol{x}_{l,4}$ in the next window is generated by Eq.(16).

It is important to note that, due to the strategies of nodes removing and marginalization, observations in SWO can be repeatedly using since only removed observations contribute to posteriori, which makes full use of observations to *re-linearize*.

On the other hand, information matrix is the inverse of covariance matrix. We have:

$$\underbrace{\begin{bmatrix}\boldsymbol{N}_{mm} & \boldsymbol{N}_{mr}\\ \boldsymbol{N}_{mr}^{T} & \boldsymbol{N}_{rr}\end{bmatrix}^{-1}}_{information}=\underbrace{\begin{bmatrix}\left(\boldsymbol{N}_{mm}-\boldsymbol{N}_{mr}\boldsymbol{N}_{rr}^{-1}\boldsymbol{N}_{mr}^{T}\right)^{-1} & \boldsymbol{C}\\ \boldsymbol{C}^{T} & \left(\boldsymbol{N}_{rr}-\boldsymbol{N}_{mr}^{T}\boldsymbol{N}_{mm}^{-1}\boldsymbol{N}_{mr}\right)^{-1}\end{bmatrix}}_{covariance}$$

When $\boldsymbol{x}_m$ is removed, SWF deletes covariance of $\boldsymbol{x}_m$ directly, while SWO applies marginalization. Then the information and covariance matrix of $\boldsymbol{x}_r$ have the relationship of:

$$\boldsymbol{D}_r=\left(\boldsymbol{N}_{rr}-\boldsymbol{N}_{mr}^{T}\boldsymbol{N}_{mm}^{-1}\boldsymbol{N}_{mr}\right)^{-1}=\boldsymbol{N}_r^{-1} \tag{17}$$

Where $\boldsymbol{D}_r$ is the covariance matrix of $\boldsymbol{x}_r$. Eq.(17) indicates that deleting covariance in SWF equals to marginalization in SWO, if use the same priori and observations.

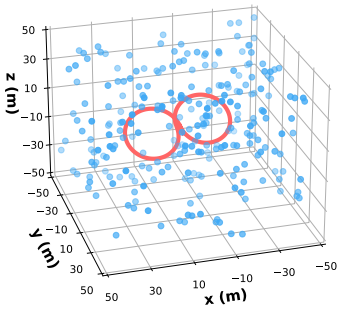


Figure 5. Simulated trajectory and landmarks.

Image width	1920
Image height	1080
f_x	1960.422
f_y	1960.422
c_x	947.492
c_y	450.813
b	801.853

Table 1. Stereo camera settings in simulated experiments.

Based on analyses in Section 2.3, we show that when sliding window is applied in both optimization-based and filtering-based methods, SWO should perform better in accuracy, since observations are utilized more than once, which overcomes effects of non-linearity. And differences between SWO and SWF are caused by prior constraints, measurements utilization, and linearization points, which can be corrected by applying the same strategies in SWF.

3. MODIFIED SLIDING WINDOW FILTER

Based on previous analysis on theories, Jacobi, and strategies, SWF should be equal to SWO theoretically. However, different linearization points and strategies applied in both methods lead to different results of SWF and SWO. Thus, by using the same linearization points and strategies in SWO, SWF can achieve the same accuracy.

As shown in Figure 4, there are three steps for modified SWF. First, same as SWO all landmarks tracked by two frames at least are used to update instead of using ones which satisfy conditions illustrated in Section 2.3.1. And then, output estimating results. Second, while SWF removes all used observations and oldest frames, the modified version selects observations connected to

removing nodes to form Jacobian matrix, $\boldsymbol{J}_{MSWF}$, where it is evaluated at linearization points which is not updated. Then derive information matrix, $\boldsymbol{N}_{MSWF} = \boldsymbol{J}_{MSWF}^{T}\boldsymbol{J}_{MSWF}$, and do once filtering to acquire posterior covariance matrix, which equals to marginalization. After this, remove observations connected to oldest frame. Thirdly, constrain nodes, $\boldsymbol{x}_r$, by covariance matrix in the next window, and repeat the same three steps.

By following these three steps, the same observations and priori will be utilized in both SWO and SWF. Consequently, once the Jacobian matrix is evaluated at identical points, the modified SWF will equal SWO in each step, resulting in equivalent estimation outcomes.

Therefore, the initial values of states in the first window can be manually set to be identical. With an equivalent number of iterations and window size, both SWO and modified SWF will yield identical estimation results in the first window since all input information and strategies are equivalent. Based on this, the subsequent windows can also achieve consistent results.

4. EXPERIMENTAL RESULTS

In this section, we present three sets of experiments to validate our analysis. First, we compare estimating results of filter and optimization at different linearization points, with states containing all frames and landmarks in the whole trajectory. We then examine accuray of SWF and SWO based on strategies of MSCKF and VINS-mono respectively. Lastly, estimating results of SWO and the modfied SWF are analysed. Meanwhile, all three experiments are conducted with simulated stereo visual data, which ensures the same observations. An *8-turn* trajectory is firstly generated, and then landmarks are simulated uniformly and randomly in space as shown in Figure 5. Stereo camera settings can be referred to in Table 1.

4.1 Linearization points

To prevent interference with other strategies, the estimation states include all frames and landmarks in the entire *8-turn* trajectory. The Jacobian matrix is then evaluated at different linearization points. Initially, manual values are given to both estimators for each measurement, which are then stacked and updated using an optimization-based method with a specific number of iterations. Meanwhile, the filtering-based approach updates states epoch by epoch and applies the same number of iterations in each update. As the filtering-based method updates states at each epoch, the linearization points are altered in the subsequent epoch, resulting in discrepancies between the filtering-based and optimization-based approaches, as illustrated in Figure 6.

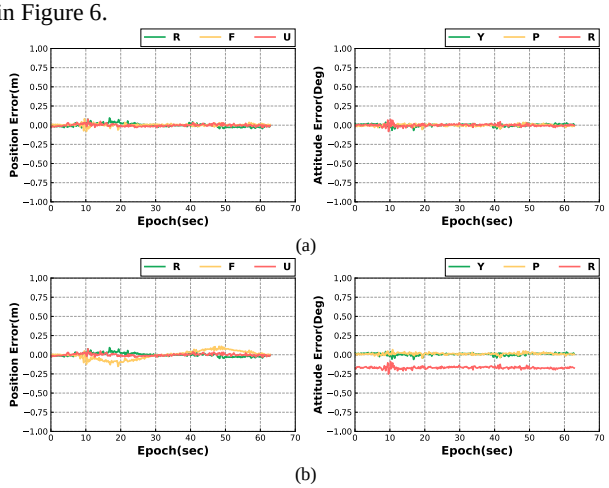


Figure 6. Residuals of position (left) and attitude (right) based on optimization (a) and filtering (b) evaluating at different points.

Subsequently, the Jacobian matrix is evaluated at the ground truth, revealing that both estimators yield comparable accuracy in estimation. This suggests that the choice of linearization points has a significant impact on estimating results, and selecting identical linearization points enables both estimators to achieve equivalent accuracy in estimation.

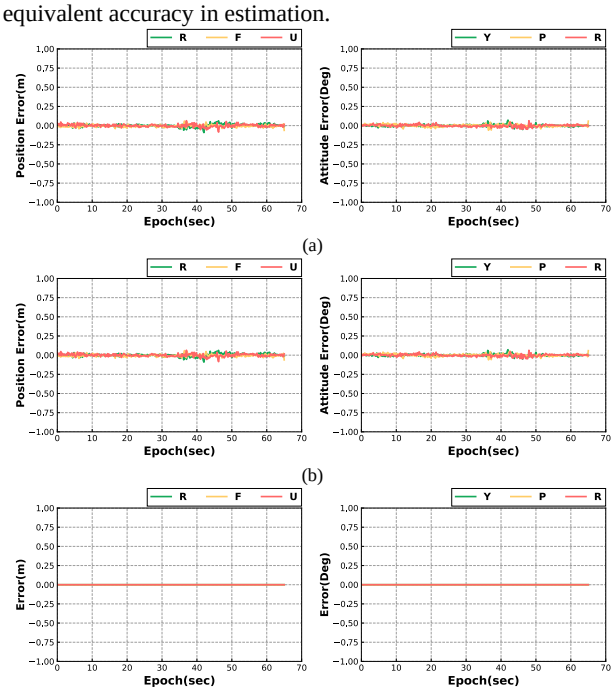
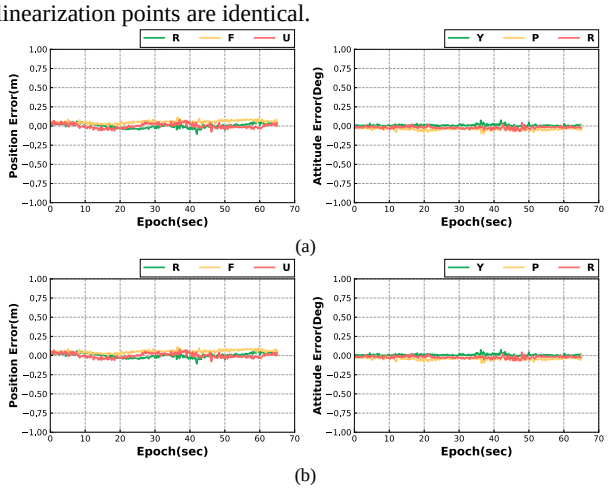
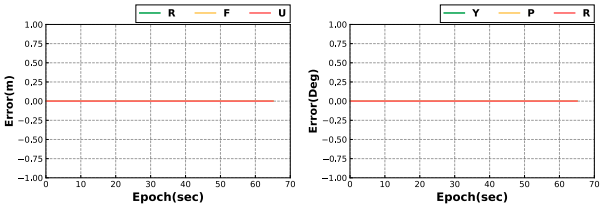


Figure 7. Residuals of position (left) and attitude (right) based on optimization (a) and filtering (b) evaluating at ground truth. (Where, (c) shows differences of estimating results between filtering and optimization.)

Lastly, instead of performing iterations at each epoch, iterate only once per epoch in filtering-based method. Additionally, when utilizing observations during each epoch, estimated error states should not be corrected. In other words, the state estimation should utilize Eq.(7), as it currently operates as an open-loop system. Once all observations have been utilized, update the states using estimated error states and then set them to **0** before proceeding with the next iteration linearized at updated states. With the same number of iterations, both estimation approaches can achieve the same accuracy as shown in Figure 8, since all linearization points are identical.



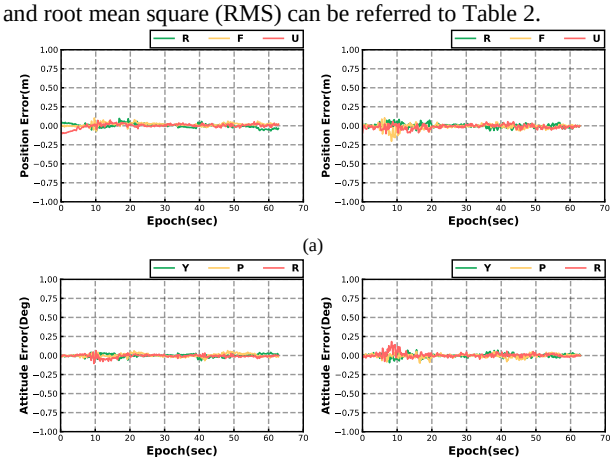


(c)

Figure 8. Residuals of position (left) and attitude (right) based on optimization (a) and filtering (b) evaluating at the same points.

4.2 SWF vs. SWO

Since impact of linearization points have already analysed, linearization points keep the same in both SWF and SWO. We set the size of window to 20 frames, and initial values and uncertainties are manually set the same. In each window, iterate only once. Residuals of both methods can be referred to Figure 9, and root mean square (RMS) can be referred to Table 2.



(a)

(b)

Figure 9. Residuals of position (left) and attitude (right) based on SWO (a) and SWF (b).

With analysis of root mean square (RMS), SWO slightly performs better than SWF when strategies are not modified in SWF, due to repeated use of observations. Since input information of both approaches are equivalent, differences are caused by strategies analysed in section 2.3.

RMS	Position (m)			Attitude (deg)		
	R	F	U	Y	P	R
SWO	0.027	**0.026**	**0.029**	**0.019**	**0.022**	**0.021**
SWF	0.026	0.036	0.031	0.023	0.024	0.031

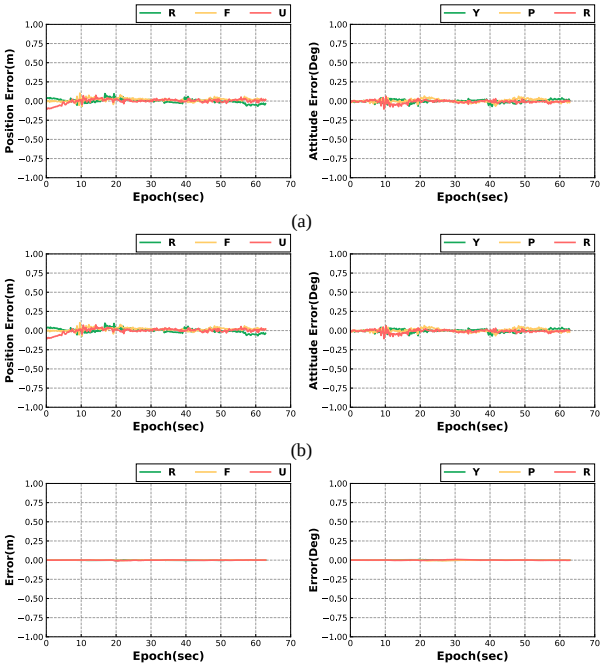
Table 2. RMS of SWO and SWF.

4.3 Modified SWF

We further follow the three steps of strategies to modify SWF as analysed in section 3, and all the input information is given the same, which ensures input information in each window of modified SWF and SWO is identical. Eventually, modified SWF achieves the same estimating results compared with SWO as shown in Figure 10. However, since covariance of unconstrained states are set to infinity, there exists slight differences between SWO and modified SWF due to numerical instability.

5. CONCLUSIONS AND FUTURE WORKs

In this paper, we have comprehensively and theoretically analysed from three aspects: formula derivation, Jacobi analysis, and strategies. With validated by stereo visual odometry simulation, we conclude that differences between optimization-based and filtering-based approaches are caused by linearization points and strategies. By inputting the same measurements, prior information, and applying the same strategies, both filtering-based and optimization-based approaches are equivalent in terms of accuracy once the same linearization points and strategies are applied in both estimators, under the assumption of Gaussian noise and the same observation model. Since strategies, rather than estimators, impact estimating results significantly, future research on multi-sensor fusion should concentrate on strategies instead of state estimation methods.



(a)

(b)

(c)

Figure 10. Residuals of position (left) and attitude (right) based on SWO (a) and modified SWF (b).

REFERENCES

Bell, B.M., Cathey, F.W., 1993. The iterated Kalman filter update as a Gauss-Newton method. IEEE Trans. Automat. Contr. 38, 294–297. https://doi.org/10.1109/9.250476

Campos, C., Elvira, R., Rodríguez, J.J.G., Montiel, J.M.M., Tardós, J.D., 2021. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM. IEEE Trans. Robot. 37, 1874–1890. https://doi.org/10.1109/TRO.2021.3075644

Cao, S., Lu, X., Shen, S., 2022. GVINS: Tightly Coupled GNSS–Visual–Inertial Fusion for Smooth and Consistent State Estimation. IEEE Trans. Robot. 38, 2004–2021. https://doi.org/10.1109/TRO.2021.3133730.

Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O., 2007. MonoSLAM: Real-Time Single Camera SLAM. IEEE Trans. Pattern Anal. Mach. Intell. 29, 1052–1067. https://doi.org/10.1109/TPAMI.2007.1049

Eckenhoff, K., Paull, L., Huang, G., 2016. Decoupled, consistent node removal and edge sparsification for graph-based SLAM, in: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Presented at the 2016 IEEE/RSJ

International Conference on Intelligent Robots and Systems (IROS), IEEE, Daejeon, South Korea, pp. 3275–3282. https://doi.org/10.1109/IROS.2016.7759505

Grejner-Brzezinska, D.A., Da, R., Toth, C., 1998. GPS error modeling and OTF ambiguity resolution for high-accuracy GPS/INS integrated system. Journal of Geodesy 72, 626–638. https://doi.org/10.1007/s001900050202

Grisetti, G., Stachniss, C., Burgard, W., 2007. Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters. IEEE Trans. Robot. 23, 34–46. https://doi.org/10.1109/TRO.2006.889486

Huang, G., 2019. Visual-Inertial Navigation: A Concise Review, in: 2019 International Conference on Robotics and Automation (ICRA). Presented at the 2019 International Conference on Robotics and Automation (ICRA), IEEE, Montreal, QC, Canada, pp. 9572–9582. https://doi.org/10.1109/ICRA.2019.8793604

Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J., Dellaert, F., 2011. iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering, in: 2011 IEEE International Conference on Robotics and Automation. Presented at the 2011 IEEE International Conference on Robotics and Automation (ICRA), IEEE, Shanghai, China, pp. 3281–3288. https://doi.org/10.1109/ICRA.2011.5979641

Leutenegger, S., Furgale, P., Rabaud, V., Chli, M., Konolige, K., Siegwart, R., 2013. Keyframe-Based Visual-Inertial SLAM using Nonlinear Optimization, in: Robotics: Science and Systems IX. Presented at the Robotics: Science and Systems 2013, Robotics: Science and Systems Foundation. https://doi.org/10.15607/RSS.2013.IX.037

Li, S., Li, X., Wang, H., Zhou, Y., Shen, Z., 2023. Multi-GNSS PPP/INS/Vision/LiDAR tightly integrated system for precise navigation in urban environments. Information Fusion 90, 218–232. https://doi.org/10.1016/j.inffus.2022.09.018

Mazuran, M., Gian Diego, T., Luciano, S., Burgard, W., 2014. Nonlinear Graph Sparsification for SLAM, in: Robotics: Science and Systems X. Presented at the Robotics: Science and Systems 2014, Robotics: Science and Systems Foundation. https://doi.org/10.15607/RSS.2014.X.040

Mourikis, A.I., Roumeliotis, S.I., 2007. A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation, in: Proceedings 2007 IEEE International Conference on Robotics and Automation. Presented at the 2007 IEEE International Conference on Robotics and Automation, IEEE, Rome, Italy, pp. 3565–3572. https://doi.org/10.1109/ROBOT.2007.364024

Niu, X., Tang, H., Zhang, T., Fan, J., Liu, J., 2023. IC-GVINS: A Robust, Real-Time, INS-Centric GNSS-Visual-Inertial Navigation System. IEEE Robot. Autom. Lett. 8, 216–223. https://doi.org/10.1109/LRA.2022.3224367

Qin, T., Li, P., Shen, S., 2018. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. IEEE Trans. Robot. 34, 1004–1020. https://doi.org/10.1109/TRO.2018.2853729

Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., Rus, D., 2020. LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Presented at the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Las Vegas, NV, USA, pp. 5135–5142. https://doi.org/10.1109/IROS45743.2020.9341176

Sibley, G., Matthies, L., Sukhatme, G., 2010. Sliding window filter with application to planetary landing: Sibley et al.: Sliding Window Filter. J. Field Robotics 27, 587–608. https://doi.org/10.1002/rob.20360

Strasdat, H., Montiel, J.M.M., Davison, A.J., 2010. Real-time monocular SLAM: Why filter?, in: 2010 IEEE International Conference on Robotics and Automation. Presented at the 2010 IEEE International Conference on Robotics and Automation (ICRA 2010), IEEE, Anchorage, AK, pp. 2657–2664. https://doi.org/10.1109/ROBOT.2010.5509636

Vial, J., Durrant-Whyte, H., Bailey, T., 2011. Conservative Sparsification for efficient and consistent approximate estimation, in: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. Presented at the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2011), IEEE, San Francisco, CA, pp. 886–893. https://doi.org/10.1109/IROS.2011.6095128

Yang, Y., Maley, J., Huang, G., 2017. Null-space-based marginalization: Analysis and algorithm, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Presented at the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Vancouver, BC, pp. 6749–6755. https://doi.org/10.1109/IROS.2017.8206592