An iterative Kalman smoother for robust 3D localization and mapping

Dimitrios G. Kottas and Stergios I. Roumeliotis

Abstract In this paper, we present an iterative Kalman smoother (IKS) for robust 3D localization and mapping, using visual and inertial measurements. Contrary to extended Kalman filter (EKF) methods, smoothing increases the convergence rate of critical parameters (e.g., IMU's velocity and camera's clock drift), improves the positioning accuracy during challenging conditions (e.g., scarcity of visual features), and allows the immediate processing of visual observations. As opposed to existing smoothing approaches to VINS, based on the inverse filter (INVF), the proposed IKS exhibits superior numerical properties, allowing efficient implementations on mobile devices. Furthermore, we propose a classification of visual observations, for smoothing algorithms applied to VINS, based on their: (i) Track length, allowing their efficient processing as multi-state constraints, when possible and (ii) First observation, allowing their optional re-processing. Finally, we demonstrate the robustness of the proposed approach, over challenging indoor VINS scenarios, including, system (re)-initialization, and scarcity of visual observations.

1 Introduction and Related Work

Over the past decade, localization systems fusing inertial data, from an inertial measurement unit (IMU), with visual observations, from a camera [i.e., vision-aided inertial navigation systems (VINS)] have become a popular choice for GPS-denied navigation (e.g., indoors [1], or in space [2]). The dramatic upswing in manufacturing of low-cost, miniature IMUs and cameras, combined with the increasing capabilities of embedded computers, have made mobile devices (e.g., cell phones) excellent platforms for VINS, but raised also new challenges, when designing algorithms with improved robustness and efficiency characteristics.

The Maximum a Posteriori (MAP) estimator for VINS, corresponds to a batch least-squares (BLS) problem, over the platform's trajectory, and the map of the environment. Unfortunately, BLS approaches cannot serve time-critical navigation ap-

The authors are with the Department of Computer Science and Engineering, University of Minnesota, USA (e-mail: dkottas|stergios@cs.umn.edu). This work was supported by the National Science Foundation (IIS-1328722) and UMN's Doctoral Dissertation Fellowship.

plications (e.g., augmented reality), due to their unbounded processing and memory requirements, as the problem size increases with time [3].

Filtering approaches, on the other hand, maintain bounded processing time, by optimizing over a sliding window of the most recent visual and inertial measurements, while absorbing past measurements into a prior cost term. Depending on their representation of the prior information, filtering methods can be classified into extended Kalman filters (EKFs) and inverse filters (INVFs) [4].

EKF-based algorithms exhibit excellent numerical stability, and have been demonstrated in real-time VINS implementations [5, 6]. Despite their efficiency, however, they do not allow re-processing the visual and/or inertial measurements within the optimization window, which may severely affect their performance under challenging conditions. Specifically, consider visual observations, that arrive as feature tracks, spanning a set of images within the estimator's sliding window. EKF-based approaches (e.g., the MSC-KF [2]), postpone their processing till they have reached their maximum length, which causes a delay in the state correction equal to the time between when a feature track is available for processing (i.e., when it spans at least two images), and when it is actually processed. Although small (typically less than 2 sec for personal localization), such a delay can affect the accuracy of time-critical navigation systems. Furthermore, under scenarios with increased linearization errors [e.g., (re)-initialization (after failure) of the system's operation, when the available estimates of critical quantities, such as the IMU's velocity and biases, are of low accuracy], re-processing visual and inertial measurements, can significantly improve accuracy and robustness, by increasing their rate of convergence.

Such appealing capabilities of filtering algorithms, are supported by the INVF (e.g., [7, 8, 9, 10]), which allows re-processing all inertial and visual measurements within the optimization window considered. Unfortunately, due to the high condition number of the Hessian, INVFs typically require 64-bit precision, reducing their efficiency, especially when considering that most current mobile devices feature ARM processors and NEON co-processors that provide a 4-fold processing speedup when using 32-bit precision. Note also that, with the exception of [11], existing INVFs for VINS do not classify visual observations based on their track length, not allowing their efficient processing (e.g., as in the MSC-KF) when possible.

To overcome the limitations of the EKF and INVF when applied to visual-inertial odometry (VIO), in [12], we introduced an iterative Kalman smoother (IKS), that shares the advantages of both approaches. In this paper, we extend the methodology of [12], to the most general case for VINS, which allows (re)-processing visual data either using a VIO approach (as in the MSC-KF), or as SLAM landmarks when their track length exceeds the estimator's sliding window. In particular, we introduce a sliding window IKS for VINS with the following key advantages:

- The IKS iteratively re-linearizes both inertial and camera measurements within the estimator's window, and re-processes visual data over multiple overlapping sliding-window epochs, thus improving robustness and increasing accuracy.
- The IKS employs a covariance matrix, as well as a set of linearized constraints (instead of a Hessian) for representing prior information, thus inheriting the superior numerical properties of the EKF and leading to efficient implementations.

Besides the detailed analysis and description of the proposed IKS algorithm, we demonstrate its robustness and assess its accuracy in simulations and experiments over challenging indoor VINS scenarios, including, filter initialization and scarcity of feature tracks due to sudden turns or camera occlusions. Finally, we provide a timing comparison between the IKS and the EKF, using a mobile processor.

2 Vision-aided Inertial Navigation System (VINS)

The system state[1] at time t_k is given by $\mathbf{x}_k = \left[\mathbf{x}_{I_k}^T \; {}^{I_\ell}\boldsymbol{\ell}_k^T\right]^T$ where $\mathbf{x}_{I_k}$ contains all kinematic quantities, describing the motion of the IMU's frame of reference $\{I_k\}$, while ${}^{I_\ell}\boldsymbol{\ell}_k$ comprises landmarks of the environment. In particular, $\mathbf{x}_{I_k} = \left[{}^{I_k}\mathbf{q}_G^T \; {}^G\mathbf{p}_{I_k}^T \; {}^G\mathbf{v}_{I_k}^T \; \mathbf{b}_{a_k}^T \; \mathbf{b}_{g_k}^T\right]^T$, where ${}^{I_k}\mathbf{q}_G$ is the quaternion representation of the orientation of the global frame $\{G\}$ in $\{I_k\}$, ${}^G\mathbf{v}_{I_k}$ and ${}^G\mathbf{p}_{I_k}$ are the velocity and position of $\{I_k\}$ in $\{G\}$ respectively, while $\mathbf{b}_{a_k}$ and $\mathbf{b}_{g_k}$ correspond to the gyroscope and accelerometer biases. Finally, ${}^{I_\ell}\boldsymbol{\ell}_k$ comprises N_k landmarks, i.e., ${}^{I_\ell}\boldsymbol{\ell}_k = \left[{}^{I_\ell}\mathbf{f}_1^T \ldots {}^{I_\ell}\mathbf{f}_{N_k}^T\right]^T$ where ${}^{I_\ell}\mathbf{f}_j$ denotes the inverse-depth parameterization of feature $\mathbf{f}_j$ in $\{I_\ell\}$.

2.1 Inertial Measurements

The IMU provides measurements of the platform's rotational velocity and linear acceleration, contaminated by white Gaussian noise and time-varying biases. Let $\mathbf{u}_{k,k+1}$ denote the inertial measurements within the time interval $\left[t_k,\, t_{k+1}\right]$, which through integration, define a constraint (discrete-time process model) of the form:

$$\mathbf{x}_{I_{k+1}} = \mathbf{f}(\mathbf{x}_{I_k},\, \mathbf{u}_{k,k+1} - \mathbf{w}_{k,k+1}) \tag{1}$$

where $\mathbf{w}_{k,k+1}$ is a discrete-time zero-mean white Gaussian noise process with covariance $\mathbf{Q}_k$. [2] Linearizing (1), at the state estimates corresponding to the two consecutive states, $\mathbf{x}_{I_k}^\star$, $\mathbf{x}_{I_{k+1}}^\star$, results in the following IMU measurement model, relating the error states $\widetilde{\mathbf{x}}_{I_k}^\star$ and $\widetilde{\mathbf{x}}_{I_{k+1}}^\star$:

$$\widetilde{\mathbf{x}}_{I_{k+1}}^\star = \mathbf{r}_{u_{k,k+1}}^\star + \boldsymbol{\Phi}_{k+1,k}^\star \widetilde{\mathbf{x}}_{I_k}^\star + \mathbf{G}_{k+1,k}^\star \mathbf{w}_{k,k+1} \tag{2}$$

where $\mathbf{r}_{u_{k,k+1}}^\star := \mathbf{f}(\mathbf{x}_{I_k}^\star, \mathbf{u}_{k,k+1}) - \mathbf{x}_{I_{k+1}}^\star$, and we have defined the error state $\widetilde{\mathbf{x}}_{I_k}^\star$ as the difference between the true state $\mathbf{x}_{I_k}$ and the linearization point $\mathbf{x}_{I_k}^\star$ (i.e., $\widetilde{\mathbf{x}}_{I_k}^\star = \mathbf{x}_{I_k} -$

[1] Without loss of generality, we assume that the IMU-camera extrinsic calibration is the identity transformation and the clocks of the two sensors are perfectly synchronized. In practice, both are included in the system's state following the methodologies described in [13] and [6], respectively.

[2] The interested reader is referred to [14, 1, 5], and references there-in, for details on IMU integration.

$\mathbf{x}_{I_k}^\star$).[3] The Jacobians $\boldsymbol{\Phi}_{k+1,k}^\star$ and $\boldsymbol{G}_{k+1,k}^\star$ are evaluated at $\mathbf{x}_{I_k}^\star$, $\mathbf{x}_{I_{k+1}}^\star$, and are available in numerical or analytical form [1]. Although the corresponding cost term,

$$c_{u_{k,k+1}}(\tilde{\mathbf{x}}_{k:k+1}^\star) = ||\mathbf{r}_{u_{k,k+1}}^\star - \begin{bmatrix}-\boldsymbol{\Phi}_{k+1,k}^\star & \mathbf{I}\end{bmatrix} \begin{bmatrix}\tilde{\mathbf{x}}_{I_k}^\star \\ \tilde{\mathbf{x}}_{I_{k+1}}^\star\end{bmatrix} ||_{\mathbf{Q}_k^\star}^2$$

where $\tilde{\mathbf{x}}_{I_{k:k+1}}^\star := \begin{bmatrix}\tilde{\mathbf{x}}_{I_k}^{\star T} & \tilde{\mathbf{x}}_{I_{k+1}}^{\star T}\end{bmatrix}^T$ and $\mathbf{Q}_k^\star = \mathbf{G}_{k+1,k}^\star \mathbf{Q}_k \mathbf{G}_{k+1,k}^{\star T}$, can be re-linearized multiple times within the INVF framework, current EKF-based approaches linearize (1) only *once*, during state and covariance propagation (i.e., every time a new inertial measurement becomes available). This limitation may negatively affect performance when the linearization errors are large (e.g., during system initialization).

2.2 Visual Observations

In order for a camera to provide kinematic information, a feature-tracking pipeline is required. A common choice comprises a feature-extraction method (e.g., the Harris corner [15]) along with a tracking algorithm (e.g., the Kanade-Lucas-Tomasi (KLT) [16]). Once a new image arrives, point features from the previous image, are tracked to the new one, while new features are extracted from areas that just entered the camera's field of view [1]. An example of the feature tracks generated from such an image-processing pipeline is shown in Fig. 1.

Consider a point feature $\mathbf{f}_j$, observed in camera poses $\mathbf{x}_{I_{k+1:k+N_j}}$, where $N_j \leq M$ and M is the window's length. We represent $\mathbf{f}_j$, using its homogeneous coordinates and inverse distance in $\{I_{k+1}\}$, hereafter denoted by ${}^{I_{k+1}}\mathbf{f}_j$, or using its Cartesian coordinates, ${}^{I_{k+1}}\mathbf{p}_{f_j}$. For the m-th measurement, $m \in \begin{bmatrix}1,\ldots,N_j\end{bmatrix}$, the observation $\mathbf{z}_{k+m,j}$ acquired by a calibrated camera is:

$$\mathbf{z}_{k+m,j} = \pi(C({}^{I_{k+m}}\mathbf{q}_{I_{k+1}})^{I_{k+1}}\mathbf{p}_{f_j} + {}^{I_{k+m}}\mathbf{p}_{I_{k+1}}) + \mathbf{n}_{k+m,j} \tag{3}$$

where, $\pi(\begin{bmatrix}x & y & z\end{bmatrix}^T) = \begin{bmatrix}\frac{x}{z} & \frac{y}{z}\end{bmatrix}^T$, while $\mathbf{n}_{k+m,j} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_2)$, and $\mathbf{I}_2$ is the 2×2 identity matrix. Linearizing (3), yields:

$$\tilde{\mathbf{z}}_{k+m,j}^\star = \mathbf{H}_{R,k+1,j}^\star \tilde{\mathbf{x}}_{I_{k+1}}^\star + \mathbf{H}_{R,k+m,j}^\star \tilde{\mathbf{x}}_{I_{k+m}}^\star + \mathbf{F}_{k+m,j}^\star {}^{I_{k+1}}\tilde{\mathbf{f}}_j^\star + \mathbf{n}_{k+m,j}$$

Collecting all N_j observations of feature $\mathbf{f}_j$ in one vector, yields:

$$\tilde{\mathbf{z}}_j^\star = \mathbf{H}_{R,j}^\star \tilde{\mathbf{x}}_{I_{k+1:k+N_j}}^\star + \mathbf{F}_j^{\star\, I_{k+1}}\tilde{\mathbf{f}}_j^\star + \mathbf{n}_j \tag{4}$$

which corresponds to the cost term:

$$c_{z_{f_j}}(\tilde{\mathbf{x}}_{I_{k+1:k+N_j}}^\star, {}^{I_{k+1}}\tilde{\mathbf{f}}_j) = ||\tilde{\mathbf{z}}_j^\star - \mathbf{H}_{R,j}^\star \tilde{\mathbf{x}}_{I_{k+1:k+N_j}}^\star - \mathbf{F}_j^{\star I_{k+1}}\tilde{\mathbf{f}}_j||_{\sigma^2\mathbf{I}}^2 \tag{5}$$

Consider an orthonormal matrix $\boldsymbol{\Theta}_j$, partitioned as $\boldsymbol{\Theta}_j = \begin{bmatrix}\mathbf{S}_j & \mathbf{U}_j\end{bmatrix}$, where the 3 columns of $\mathbf{S}_j$ span the column space of $\mathbf{F}_j^\star$, while the $2N_j - 3$ columns of $\mathbf{U}_j$, its left null space. Projecting (4) onto $\boldsymbol{\Theta}_j$, partitions $c_{z_{f_j}}$ into two parts:

[3] For the quaternion $\mathbf{q}$ we employ a multiplicative error model $\widetilde{\mathbf{q}} = \mathbf{q} \otimes \mathbf{q}^{\star -1} \simeq \begin{bmatrix}\frac{1}{2}\boldsymbol{\delta\theta}_{3\times 1}^T & 1\end{bmatrix}^T$.

$$
\begin{aligned}
c_{z_{f_j}}(\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+N_j}}, {}^{I_{k+1}}\tilde{\mathbf{f}}_j) &= ||\boldsymbol{\Theta}_j^T \left(\tilde{\mathbf{z}}_j^{\star} - \mathbf{H}_{R,j}^{\star}\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+N_j}} - \mathbf{F}_j^{\star\, I_{k+1}}\tilde{\mathbf{f}}_j\right)||^2_{\sigma^2\mathbf{I}_{2N_j}} \\
&= ||\mathbf{r}_j^{K\star} - \mathbf{H}_j^{K\star}\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+N_j}}||^2_{\sigma^2\mathbf{I}_{2N_j-3}} + ||\mathbf{r}_j^{M\star} - \mathbf{H}_j^{M\star}\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+N_j}} - \mathbf{R}_j^{M\star\, I_{k+1}}\tilde{\mathbf{f}}_j||^2_{\sigma^2\mathbf{I}_3} \\
&= c^K_{z_{f_j}}(\tilde{\mathbf{x}}^{\star}_{k+1:k+N_j}) + c^M_{z_{f_j}}(\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+N_j}}, {}^{I_{k+1}}\tilde{\mathbf{f}}_j) \qquad (6)
\end{aligned}
$$

with $\mathbf{r}_j^{K\star} = \mathbf{U}_j^T\tilde{\mathbf{z}}_j^{\star}$, $\mathbf{H}_j^{K\star} = \mathbf{U}_j^T\mathbf{H}_{R,j}^{\star}$, and $\mathbf{r}_j^{M\star} = \mathbf{S}_j^T\tilde{\mathbf{z}}_j^{\star}$, $\mathbf{H}_j^{M\star} = \mathbf{S}_j^T\mathbf{H}_{R,j}^{\star}$, $\mathbf{R}_j^{M\star} = \mathbf{S}_j^T\mathbf{F}_{R,j}^{\star}$. The second term, $c^M_{z_{f_j}}$ contains *all* information regarding feature f_j, while $c^K_{z_{f_j}}$ defines a multi-state constraint *only* among the poses $\mathbf{x}_{I_{k+1:k+N_j}}$. For feature tracks that do not exceed the estimator's window, as in the MSC-KF [2], we consider *only* the cost term $c^K_{z_{f_j}}$. Specifically, since $\mathbf{R}_j^{M\star}$ is an invertible square matrix and *for any* $\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+N_j}}$ *there exists* a ${}^{I_{k+1}}\tilde{\mathbf{f}}_j$, such that $c^M_{z_{f_j}}$ is exactly zero, minimizing (6) is equivalent to minimizing $c^K_{z_{f_j}}(\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+N_j}})$. As we describe later on, for feature tracks whose span exceeds the sliding window (i.e., SLAM landmarks), $c^M_{z_{f_j}}$ allows initializing them into the estimator's map, and subsequently optimizing their measurements across *non-overlapping* epochs of the sliding window.

2.3 Visual Observations in Sliding-Window Estimators

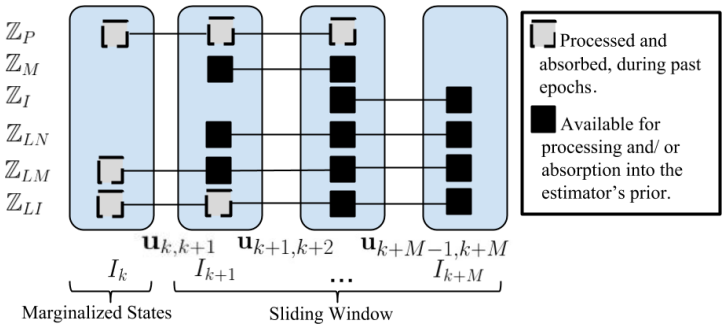


Fig. 1: At t_{k+M}, there exist 6 categories of feature tracks described in Section 2.3.

We classify visual observations, in sliding-window estimators, based on their:
i) **Track length** which distinguishes SLAM landmarks from MSC-KF features, since for the latter, their track length does not exceed the estimator's window. Thus, optimizing over the MSC-KF feature's coordinates is not required for minimizing their corresponding cost terms [see (6)].
ii) **Earliest observation** which if it involves the sliding window's "tail" (oldest) state, it does not allow postponing their absorption into the estimator's prior.
In particular, we distinguish the following categories of visual observations:

- **Past features** ($\mathbb{Z}_P$) corresponding to visual observations that were absorbed in a past epoch of the sliding window and cannot be re-processed by any filter.
- **MSC-KF features** whose tracking length does not exceed the estimator's window, i.e., $N_j \leq M$, hence they are not mapped but rather used only for providing multi-state constraints involving the camera poses observing them. Based on their earliest observation, we further classify MSC-KF features into 2 sets:
  - **Mature features** ($\mathbb{Z}_M$): These are MSC-KF features that have reached their maximum length. Both the EKF and the INVF linearize, process, and absorb them in a single step. Note also that the INVF, as well as the Iterative EKF (I-EKF) [4] can re-linearize these observations.
  - **Immature features** ($\mathbb{Z}_I$): This set represents feature tracks that have already entered the image-processing pipeline, but have not reached their maximum length, yet. Although the INVF can use (and re-linearize) these measurements multiple times, across overlapping epochs of the sliding window (from the second time they are observed till the track exits the optimization window), the EKF is not able to do so. This limitation introduces a delay, between the "birth" of a feature track and its impact on the filter's state estimates; a potential drawback of EKF-based approaches for time-critical applications.
- **SLAM landmarks** corresponding to features, whose track length exceeds the estimator's optimization window, i.e., $N_j > M$, and hence their optimal processing requires including them into the estimator's map of the environment, ${}^{I_{k+1}}\boldsymbol{\ell}_{k+M}$. Within a single epoch of the sliding window, observations to SLAM landmarks can be partitioned into 3 categories:
  - **Mature landmarks** ($\mathbb{Z}_{LM}$): These are observations to previously initialized SLAM landmarks, which include the estimator's "tail" pose, hence their absorption into the estimator's prior, cannot be postponed for future epochs of the sliding window.
  - **Immature landmarks** ($\mathbb{Z}_{LI}$): These correspond to measurements of previously initialized SLAM landmarks, which do not involve the estimator's "tail" pose, thus their absorption can be postponed to later epochs, till one of them includes the estimator's "tail" pose, i.e., they become mature landmarks.
  - **New SLAM landmarks** ($\mathbb{Z}_{LN}$): These are feature tracks that have not yet reached their maximum length, while their present observations span all cameras within the estimator's optimization window. Hence, they will be processed, absorbed, and initialized as new landmarks in the estimator's map.

3 Estimation Algorithm Description

When designing the proposed IKS our objective is two-fold: (i) Process all inertial and visual observations within the current epoch of the sliding window $\mathbf{x}_{I_{k+1:k+M}}$ (i.e., inertial measurements $\{\mathbf{u}_{\ell,\ell+1}\}$, for $k+1 \leq \ell \leq k+M-1$, and feature tracks $\mathbb{Z}_{LM}$, $\mathbb{Z}_{LI}$, $\mathbb{Z}_{LN}$, $\mathbb{Z}_M$, and $\mathbb{Z}_I$), and (ii) Allow future epochs to re-process all measurements

that are independent of the sliding window's "tail" state $\mathbf{x}_{I_{k+1}}$ (i.e., the inertial measurements $\{\mathbf{u}_{\ell,\ell+1}\}$, for $k+2 \leq \ell \leq k+M-1$, and visual observations to immature MSC-KF features and SLAM landmarks (i.e., $\mathbb{Z}_I$ and $\mathbb{Z}_{LI}$, respectively).

3.1 IKS Algorithm: Input

Before image $k+M$ arrives, the proposed IKS maintains:
1) **A set of linearization points**, over the estimator's sliding-window of camera poses $\mathbf{x}^{\star}_{I_{k+1}:k+M-1}$, and landmarks ${}^{I_{k+1}}\boldsymbol{\ell}^{\star}_{k+M-1}$ that represent the estimator's best estimates given all measurements up to t_{k+M-1}.
2) **A prior** comprising:
(a) The pdf of the oldest state, $\mathbf{x}_{I_{k+1}}$, within the sliding window approximated as a Gaussian $\mathcal{N}(\hat{\mathbf{x}}^{\ominus}_{I_{k+1}}, \mathbf{P}^{\ominus}_{I_{k+1}})$.
(b) A set of N_L linearized constraints relating the oldest state, $\mathbf{x}^{\star}_{I_{k+1}}$, with the rest of the poses within the sliding window, expressed as:

$$\mathbf{r}^{\star\ominus}_{L} = \mathbf{H}^{\star\ominus}_{L}\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+M-1}} + \mathbf{n}_L,\ \mathbf{n}_L \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_{N_L}). \tag{7}$$

(c) A set of $3N_{k+M-1}$ linearized constraints relating the oldest state, $\mathbf{x}^{\star}_{I_{k+1}}$, with the rest of the poses within the sliding window and the SLAM landmarks ${}^{I_{k+1}}\boldsymbol{\ell}_{k+M-1}$ expressed as:

$$\mathbf{r}^{\star\ominus}_{M} = \mathbf{H}^{\star\ominus}_{M}\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+M-1}} + \mathbf{F}^{\star\ominus}_{M}{}^{I_{k+1}}\widetilde{\boldsymbol{\ell}}^{\star}_{k+M-1} + \mathbf{n}_M,\ \mathbf{n}_M \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_{3N_{k+M}}). \tag{8}$$

The ensemble of the pdf $\mathcal{N}(\hat{\mathbf{x}}^{\ominus}_{I_{k+1}}, \mathbf{P}^{\ominus}_{I_{k+1}})$ and the linearized constraints $\{\mathbf{r}^{\star\ominus}_{L},\ \mathbf{H}^{\star\ominus}_{L}\}$ and $\{\mathbf{r}^{\star\ominus}_{M},\ \mathbf{H}^{\star\ominus}_{M},\ \mathbf{F}^{\star\ominus}_{M}\}$ in (7) and (8) represent all information for the poses $\mathbf{x}_{I_{k+1:k+M-1}}$ and the landmarks ${}^{I_{k+1}}\boldsymbol{\ell}_{k+M-1}$, accumulated through absorption of past visual observations (i.e., $\mathbb{Z}_P$ in Fig. 1) and inertial measurements (i.e., $\{\mathbf{u}_{\ell,\ell+1},\ \ell \leq k\}$).

3.2 IKS Algorithm: Overview

A single recursion of the IKS, involves the following steps:

1. Propagation: The prior pdf, $\mathcal{N}(\hat{\mathbf{x}}^{\ominus}_{I_{k+1}}, \mathbf{P}^{\ominus}_{I_{k+1}})$, of $\mathbf{x}_{I_{k+1}}$ and the inertial measurements $\{\mathbf{u}_{\ell,\ell+1}\}$, for $k+1 \leq \ell \leq k+M-1$, are used for creating a prior $\mathcal{N}(\hat{\mathbf{x}}^{\ominus}_{I_{k+1:k+M}}, \mathbf{P}^{\ominus}_{I_{k+1:k+M}})$ for *all* the poses within the sliding window.

2. State Update: All available feature tracks, either from SLAM landmarks, i.e., $\mathbb{Z}_{LM}$, $\mathbb{Z}_{LN}$ and $\mathbb{Z}_{LI}$, and MSC-KF features, i.e., $\mathbb{Z}_M$ and $\mathbb{Z}_I$, as well as the prior constraints $\{\mathbf{r}^{\star\ominus}_{L},\ \mathbf{H}^{\star\ominus}_{L}\}$ and $\{\mathbf{r}^{\star\ominus}_{M},\ \mathbf{H}^{\star\ominus}_{M},\ \mathbf{F}^{\star\ominus}_{M}\}$ are processed for updating the current state estimates $\mathbf{x}^{\star}_{I_{k+1:k+M}}$. This state-optimization can be performed iteratively.

3. Landmark Propagation: All landmarks are propagated to the next "tail" of the sliding window, $\mathbf{x}_{I_{k+2}}$, i.e., ${}^{I_{k+2}}\boldsymbol{\ell}_{k+M} = \left[{}^{I_{k+2}}\mathbf{f}^{T}_{1} \ldots {}^{I_{k+2}}\mathbf{f}^{T}_{N_{k+M}}\right]^{T}$.

4. Covariance Update: The measurements, $\mathbb{Z}_{LM}$, $\mathbb{Z}_{LN}$, $\mathbb{Z}_M$, and $\mathbf{u}_{k+1,k+2}$, which are about to be absorbed, are used to compute the posterior covariance $\mathbf{P}^{\oplus}_{I_{k+2}}$ of $\mathbf{x}_{I_{k+2}}$,

which will become the new "tail" of the sliding window.

5. Construction of the next epoch's prior constraints: The prior constraints $\{\mathbf{r}_L^{\star\ominus}, \mathbf{H}_L^{\star\ominus}\}$ and $\{\mathbf{r}_M^{\star\ominus}, \mathbf{H}_M^{\star\ominus}, \mathbf{F}_M^{\star\ominus}\}$ are updated so that they become independent of the state to be marginalized, $\mathbf{x}_{I_{k+1}}$, and instead reflect the new constraints between $\mathbf{x}_{I_{k+2}}$, $\mathbf{x}_{I_{k+3:k+M}}$, and ${}^{I_{k+2}}\boldsymbol{\ell}_{k+M}$.

3.3 IKS Algorithm: Detailed Description

In order to allow for a direct comparison with the INVF and the EKF, we follow a two-level presentation of the IKS: We first describe the effect that each step has on the cost function being minimized and then present the corresponding IKS equations. We start by stating that the IKS (iteratively) minimizes the cost function:

$$\begin{aligned} c_{k+M}(\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+M}}, {}^{I_{k+1}}\widetilde{\boldsymbol{\ell}}^{\star}_{k+M}) = c_{P^{\ominus}_{I_{k+1}}} &+ c_u + c_L + c^K_{z_M} + c^K_{z_I} + c^K_{z_{LM}} + c^K_{z_{LI}} + c^K_{z_{LN}} \\ &+ c_M + c^M_{z_{LM}} + c^M_{z_{LI}} + c^M_{z_{LN}} \end{aligned} \tag{9}$$

where $c_{P^{\ominus}_{k+1}}$ corresponds to the prior pdf of the oldest state within the sliding window, $\mathbf{x}_{k+1}$, $c_u = \sum_{\ell=k+1}^{k+M-1} c_{u_{\ell,\ell+1}}$ to the inertial measurements $\mathbf{u}_{k+1:k+M}$ [see (2)], c_L to prior information about the poses $\mathbf{x}_{k+1:k+M-1}$ [see (7)], $c^K_{z_M}$, $c^K_{z_I}$, $c^K_{z_{LM}}$, $c^K_{z_{LI}}$, and $c^K_{z_{LN}}$ to geometric constraints between the poses from all available visual observations [see (6)], c_M to prior information about the SLAM landmarks ${}^{I_{k+1}}\boldsymbol{\ell}_{k+M-1}$, $c^M_{z_{LM}}$ [see (8)] and $c^M_{z_{LI}}$ to feature constraints between the poses *and* the SLAM landmarks [see (6)], and finally $c^M_{z_{LN}}$ to feature constraints for the *new* SLAM landmarks, ${}^{I_{k+1}}\boldsymbol{\ell}_{\mathcal{N}}$ [see (6)].

Hereafter, we employ the cost terms in (9) to describe the four main steps of the proposed IKS (see Section 3.2).

3.3.1 Prior Propagation

The prior pdf $\mathcal{N}(\hat{\mathbf{x}}^{\ominus}_{I_{k+1}}, \mathbf{P}^{\ominus}_{I_{k+1}})$ and the inertial measurements $\mathbf{u}_{k+1:k+M}$ are used to generate a prior pdf $\mathcal{N}(\hat{\mathbf{x}}^{\ominus}_{I_{k+1:k+M}}, \mathbf{P}^{\ominus}_{I_{k+1:k+M}})$ over *all* the states, $\mathbf{x}_{I_{k+1:k+M}}$, within the sliding window. Through this process, the cost function in (9) takes the form:

$$\begin{aligned} c_{k+M}(\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+M}}, {}^{I_{k+1}}\widetilde{\boldsymbol{\ell}}^{\star}_{k+M}) = c_{P^{\ominus}_{I_{k+1:k+M}}} &+ c_L + c^K_{z_M} + c^K_{z_I} + c^K_{z_{LM}} + c^K_{z_{LI}} + c^K_{z_{LN}} \\ &+ c_M + c^M_{z_{LM}} + c^M_{z_{LI}} + c^M_{z_{LN}} \end{aligned} \tag{10}$$

where, $c_{P^{\ominus}_{I_{k+1:k+M}}} = c_{P^{\ominus}_{I_{k+1}}} + \sum_{\ell=k+1}^{k+M-1} c_{u_{\ell,\ell+1}}$ corresponds to the prior $\mathcal{N}(\hat{\mathbf{x}}^{\ominus}_{I_{k+1:k+M}}, \mathbf{P}^{\ominus}_{I_{k+1:k+M}})$. The mean $\hat{\mathbf{x}}^{\ominus}_{I_{k+1:k+M}}$ is computed as:

$$\hat{\mathbf{x}}^{\ominus}_{I_{k+i}} = \begin{cases} \hat{\mathbf{x}}^{\ominus}_{I_{k+1}} & , i = 1 \\ \mathbf{f}(\mathbf{x}^{\star}_{I_{k+i-1}}, \mathbf{u}_{k+i-1,k+i}) + \boldsymbol{\Phi}^{\star}_{k+i,k+i-1}\delta\mathbf{x}^{\ominus}_{k+i-1} & , 2 \leq i \leq M \end{cases} \tag{11}$$

where $\delta\mathbf{x}^{\ominus}_{I_{k+i-1}} = \hat{\mathbf{x}}^{\ominus}_{I_{k+i-1}} - \mathbf{x}^{\star}_{I_{k+i-1}}$. Note that for the EKF inertial measurements are linearized and processed once, soon as they become available; hence, $\delta\mathbf{x}^{\ominus}_{I_{k+i-1}} = \mathbf{0}$ and the mean of the prior pdf, $\hat{\mathbf{x}}^{\ominus}_{I_{k+i}}$, coincides with the linearization point $\mathbf{x}^{\star}_{I_{k+i}}$. In contrast, the IKS re-processes inertial measurements by re-computing the prior over the sliding window $\mathbf{x}_{I_{k+1:k+M}}$ through the process described in (11).
The block elements of the covariance $\mathbf{P}^{\ominus}_{I_{k+1:k+M}}$ are computed through the EKF covariance propagation recursion, using, however, the most recent state estimates:

$$\begin{aligned} \mathbf{P}^{\ominus}_{I_{k+i}} &= \boldsymbol{\Phi}^{\star}_{k+i,k+i-1}\mathbf{P}^{\ominus}_{I_{k+i-1}}\boldsymbol{\Phi}^{\star T}_{k+i,k+i-1} + \mathbf{Q}^{\star}_{k},\ i = 2\ldots,M \\ \mathbf{P}^{\ominus}_{I_{k+i,k+j}} &= \boldsymbol{\Phi}^{\star}_{k+i,k+j}\mathbf{P}^{\ominus}_{I_{k+j}},\ i = 2,\ldots,M,\ j = 1,\ldots,i-1 \end{aligned} \tag{12}$$

3.3.2 State Update

All cost terms, which provide multi-state (i.e., c_L, $c^K_{z_M}$, $c^K_{z_I}$, $c^K_{z_{LM}}$, $c^K_{z_{LI}}$, $c^K_{z_{LN}}$), as well as mapping (i.e., c_M, $c^M_{z_{LM}}$, $c^M_{z_{LI}}$, $c^M_{z_{LN}}$) constraints, are used for updating the state estimates for $\mathbf{x}_{I_{k+1:k+M}}$ and ${}^{I_{k+1}}\boldsymbol{\ell}_{k+M}$. Although each of these terms could have been used independently, in successive updates, we choose to first merge them into two cost terms, c^K_S and c^M_S, comprising multi-state geometric and mapping constraints, respectively, and process them in a batch form.
Measurement Compression: Combining all mapping terms, c_M, $c^M_{z_{LM}}$, $c^M_{z_{LI}}$ $c^M_{z_{LN}}$, into a single cost term, $c^{M'}_S$, yields:

$$\begin{aligned} c^{M'}_S(\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+M}}, {}^{I_{k+1}}\widetilde{\boldsymbol{\ell}}_{k+M}) &= || \begin{bmatrix} \mathbf{r}^{\star\ominus}_M \\ \mathbf{r}^{M\star}_{LI} \\ \mathbf{r}^{M\star}_{LM} \\ \mathbf{r}^{M\star}_{LN} \end{bmatrix} - \begin{bmatrix} \mathbf{H}^{\star\ominus}_M & \mathbf{F}^{\star\ominus}_M & & \\ \mathbf{H}^{M\star}_{LI} & \mathbf{F}^{M\star}_{LI} & & \\ & & \mathbf{F}^{M\star}_{LM} & \\ \mathbf{H}^{M\star}_{LN} & & & \mathbf{F}^{M\star}_{LN} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}^{\star}_{I_{k+1:k+M}} \\ {}^{I_{k+1}}\tilde{\boldsymbol{\ell}}^{\star}_{k+M-1} \\ {}^{I_{k+1}}\tilde{\boldsymbol{\ell}}_{\mathcal{N}} \end{bmatrix} ||^2_{\sigma^2\mathbf{I}} \\ &= ||\mathbf{r}^{M'}_S - \begin{bmatrix} \mathbf{H}^{M'\star}_S & \mathbf{F}^{M'\star}_S \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}^{\star}_{I_{k+1:k+M}} \\ {}^{I_{k+1}}\tilde{\boldsymbol{\ell}}^{\star}_{k+M} \end{bmatrix} ||^2_{\sigma^2\mathbf{I}} \end{aligned} \tag{13}$$

As in Section 2.2, we project the linearized constraints of (13) onto the column space and left nullspace of $\mathbf{F}^{M'\star}_S$, which partitions $c^{M'}_S$ into, $3N_{k+M}$ constraints, denoted by c^M_S, providing information *only* over the landmarks $\boldsymbol{\ell}_{k+M}$ and a cost term $c^{K'}_S$, providing geometric constraints, *only* over the poses $\mathbf{x}_{I_{k+1:k+M}}$, i.e.,

$$\begin{aligned} c^{M'}_S(\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+M}}, {}^{I_{k+1}}\widetilde{\boldsymbol{\ell}}^{\star}_{k+M}) &= ||\mathbf{r}^{M\star}_S - \begin{bmatrix} \mathbf{H}^{M\star}_S & \mathbf{F}^{M\star}_S \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}^{\star}_{I_{k+1:k+M}} \\ {}^{I_{k+1}}\tilde{\boldsymbol{\ell}}^{\star}_{k+M} \end{bmatrix} ||^2_{\sigma^2\mathbf{I}_{3N_{k+M}}} \\ &\quad + ||\mathbf{r}^{K'\star}_S - \mathbf{H}^{K'\star}_S\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+M}}||^2_{\sigma^2\mathbf{I}} \\ &= c^M_S(\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+M}}, {}^{I_{k+1}}\widetilde{\boldsymbol{\ell}}^{\star}_{k+M}) + c^{K'}_S(\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+M}}) \end{aligned}$$

which allows c_{k+M} in (10) to take the form:

$$c_{k+M}(\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+M}}, {}^{I_{k+1}}\widetilde{\boldsymbol{\ell}}^{\star}_{k+M}) = c_{P^{\ominus}_{I_{k+1:k+M}}} + c^{M}_{S} + c^{K}_{S} \tag{14}$$

where,[4] c^{K}_{S} comprises all geometric constraints, i.e., c_L, $c^{K'}_{S}$, $c^{K}_{z_M}$, $c^{K}_{z_I}$, $c^{K}_{z_{LM}}$, $c^{K}_{z_{LI}}$, and $c^{K}_{z_{LN}}$:

$$c^{K}_{S}(\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+M}}) = ||\mathbf{r}^{K\star}_{s} - \mathbf{H}^{K\star}_{s}\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+M}}||^{2}_{\sigma^2\mathbf{I}}. \tag{15}$$

Note that, since $\mathbf{F}^{M\star}_{S}$ is a square full-rank matrix and for every $\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+M}}$ there exists an ${}^{I_{k+1}}\widetilde{\boldsymbol{\ell}}^{\star}_{k+M}$ that minimizes c^{M}_{S} to zero, for minimizing (14), it suffices to first minimize $c^{\ominus}_{P_{I_{k+1:k+M}}} + c^{K}_{S}$, over $\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+M}}$, and then solve for ${}^{I_{k+1}}\widetilde{\boldsymbol{\ell}}^{\star}_{k+M}$, using c^{M}_{S}. Specifically, through an I-EKF update step [4] we first update the poses $\mathbf{x}_{I_{k+1:k+M}}$:

$$\mathbf{x}^{\star\oplus}_{I_{k+1:k+M}} = \mathbf{x}^{\star}_{I_{k+1:k+M}} + \tilde{\mathbf{x}}^{o}_{I_{k+1:k+M}}, \;\; \tilde{\mathbf{x}}^{o}_{I_{k+1:k+M}} = \delta\mathbf{x}^{\ominus}_{I_{k+1:k+M}} + \mathbf{P}^{\ominus}_{I_{k+1:k+M}}\mathbf{H}^{K\star T}_{S}\mathbf{d}_{S} \tag{16}$$

where $\mathbf{d}_S$ is the solution to the linear system $\mathbf{S}^{\star}\mathbf{d}_S = \mathbf{r}^{K\star}_{S} - \mathbf{H}^{K\star}_{S}\delta\mathbf{x}^{\ominus}_{I_{k+1:k+M}}$, with $\mathbf{S}^{\star} = \mathbf{H}^{K\star}_{S}\mathbf{P}^{\ominus}_{I_{k+1:k+M}}\mathbf{H}^{K\star T}_{S} + \sigma^2\mathbf{I}$, and $\delta\mathbf{x}^{\ominus}_{I_{k+1:k+M}} = \hat{\mathbf{x}}^{\ominus}_{I_{k+1:k+M}} - \mathbf{x}^{\star}_{I_{k+1:k+M}}$.

Second, we substitute $\tilde{\mathbf{x}}^{o}_{I_{k+1:k+M}}$ into c^{M}_{S}, and solve[5] the linear system

$$\mathbf{F}^{M\star I_{k+1}}_{S}\widetilde{\boldsymbol{\ell}}^{o}_{k+M} = \mathbf{r}^{M\star}_{S} - \mathbf{H}^{M\star}_{S}\tilde{\mathbf{x}}^{o}_{I_{k+1:k+M}} \tag{17}$$

for updating the SLAM landmarks: ${}^{I_{k+1}}\boldsymbol{\ell}^{\star\oplus} = {}^{I_{k+1}}\boldsymbol{\ell}^{\star} + {}^{I_{k+1}}\widetilde{\boldsymbol{\ell}}^{o}_{k+M}$.

It is important to note that processing immature observations (i.e., $\mathbb{Z}_I$ and $\mathbb{Z}_{LI}$) is *optional* for the IKS, allowing us to adjust its computational cost, based on the availability of computational resources.

3.3.3 Landmark Propagation

Before proceeding with computing the next epoch's prior pdf $\mathcal{N}(\hat{\mathbf{x}}^{\oplus}_{k+2}, \mathbf{P}^{\oplus}_{k+2})$ and linearized constraints, $\{\mathbf{r}^{\star\oplus}_{L}, \mathbf{H}^{\star\oplus}_{L}\}$ and $\{\mathbf{r}^{\star\oplus}_{M}, \mathbf{H}^{\star\oplus}_{M}, \mathbf{F}^{\star\oplus}_{M}\}$, we express all landmarks ${}^{I_{k+1}}\boldsymbol{\ell}_{k+M}$ w.r.t. the new "tail" pose of the sliding window, $\mathbf{x}_{I_{k+2}}$, in (9). As we describe in detail in [17], the landmark parameters in the two frames I_{k+1} and I_{k+2}, as well as the poses $\mathbf{x}_{I_{k+1:k+2}}$, are related through a non-linear *deterministic* constraint, $\mathbf{g}(\mathbf{x}_{I_{k+1:k+2}}, {}^{I_{k+1}}\boldsymbol{\ell}_{k+M}, {}^{I_{k+2}}\boldsymbol{\ell}_{k+M}) = \mathbf{0}$, which after linearization, becomes:

$$\begin{aligned} &\mathbf{G}^{\star}_{I_{k+1:k+2}}\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+2}} + \mathbf{G}^{\star}_{I_{k+1},\ell}\,{}^{I_{k+1}}\widetilde{\boldsymbol{\ell}}_{k+M} + \mathbf{G}^{\star}_{I_{k+2},\ell}\,{}^{I_{k+2}}\widetilde{\boldsymbol{\ell}}_{k+M} = \mathbf{0} \\ \Leftrightarrow {}^{I_{k+1}}&\widetilde{\boldsymbol{\ell}}^{\star}_{k+M} = -\mathbf{G}^{\star-1}_{I_{k+1},\ell}(\mathbf{G}^{\star}_{I_{k+1:k+2}}\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+2}} + \mathbf{G}^{\star}_{I_{k+2},\ell}\,{}^{I_{k+2}}\widetilde{\boldsymbol{\ell}}^{\star}_{k+M}) \end{aligned} \tag{18}$$

Substituting in (9), transforms c_{k+M} to a function of $\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+M}}$ and ${}^{I_{k+2}}\widetilde{\boldsymbol{\ell}}^{\star}_{k+M}$.

[4] For reducing the computational cost (linear in the number of MSC-KF features within the sliding window), the residual $\mathbf{r}^{K\star}_{s}$ and Jacobian matrix $\mathbf{H}^{K\star}_{s}$ are compressed using QR factorization [2].

[5] Not surprisingly, the computational cost of this step is cubic in the number of SLAM landmarks, which are observed at the present epoch, as in the corresponding EKF and INVF state-update steps.

3.3.4 Covariance Update

Our objective is to compute the posterior $\mathcal{N}(\hat{\mathbf{x}}^{\oplus}_{I_{k+2}}, \mathbf{P}^{\oplus}_{I_{k+2}})$, which will be used as the prior pdf during the next epoch. To do so, we will operate on those terms of the cost function in (9) that contain the state $\mathbf{x}_{I_{k+1}}$, which is about to be marginalized; that is the cost function:

$$c^M_{k+M} = c_{P^{\ominus}_{I_{k+1}}} + c_{u_{k+1:k+2}} + c_L + c^K_{z_M} + c^K_{z_{LM}} + c^K_{z_{LN}} + c_M + c^M_{z_{LM}} + c^M_{z_{LN}} \tag{19}$$

In particular, we follow a 4-step process:

Prior propagation and measurement compression: Following the same process, as in Sections 3.3.1 and 3.3.2, we use the prior $\mathcal{N}(\hat{\mathbf{x}}^{\ominus}_{I_{k+1}}, \mathbf{P}^{\ominus}_{I_{k+1}})$ and the inertial measurements $\mathbf{u}_{k+1:k+2}$ to compute the prior $\mathcal{N}(\hat{\mathbf{x}}^{\ominus}_{I_{k+1:k+2}}, \mathbf{P}^{\ominus}_{I_{k+1:k+2}})$, and merge the linearized constraints into two terms, c^K_C and c^M_C, comprising multi-state and mapping constraints, respectively. Thus, (19) becomes:

$$c^M_{k+M} = c_{P^{\ominus}_{I_{k+1:k+2}}} + c^K_C + c^M_C \tag{20}$$

Partitioning of the linearized constraints: Following the same process as in (6), we partition c^K_C into $c^K_{C_1}$ and $c^K_{C_2}$, where the first term depends only on $\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+2}}$, i.e., $c^K_C(\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+M}}) = c^K_{C_1}(\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+2}}) + c^K_{C_2}(\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+M}})$, which after substitution in (20), yields:

$$c^M_{k+M} = c_{P^{\ominus}_{I_{k+1:k+2}}} + c^K_{C_1}(\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+2}}) + c^K_{C_2}(\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+M}}) + c^M_C. \tag{21}$$

Covariance Update: At this point, we combine the first two terms in (21), thus updating the *prior pdf* $\mathcal{N}(\mathbf{x}^{\ominus}_{I_{k+1:k+2}}, \mathbf{P}^{\ominus}_{I_{k+1:k+2}})$:

$$c^M_{k+M} = c_{P^{\oplus}_{I_{k+1:k+2}}} + c^K_{C_2} + c^M_C. \tag{22}$$

$\mathcal{N}(\mathbf{x}^{\oplus}_{I_{k+1:k+2}}, \mathbf{P}^{\oplus}_{I_{k+1:k+2}})$ has mean $\hat{\mathbf{x}}^{\oplus}_{I_{k+1:k+2}} = \hat{\mathbf{x}}^{\ominus}_{I_{k+1:k+2}} + \mathbf{P}^{\ominus}_{I_{k+1:k+2}} \mathbf{H}^{K\star T}_{C_1} \mathbf{d}_C$, where $\mathbf{d}_C$ is the solution to the linear system $\mathbf{S}_C \mathbf{d}_C = \mathbf{r}^{K\star}_{C_1} - \mathbf{H}^{K\star T}_{C_1} \delta\mathbf{x}^{\ominus}_{k+1:k+2}$, with $\mathbf{S}_C = \mathbf{H}^{K\star}_{C_1} \mathbf{P}^{\ominus}_{I_{k+1:k+2}} \mathbf{H}^{K\star T}_{C_1} + \sigma^2 \mathbf{I}_{N_1}$, and covariance $\mathbf{P}^{\oplus}_{I_{k+1:k+2}} = \mathbf{P}^{\ominus}_{I_{k+1:k+2}} \left(\mathbf{I} - \mathbf{H}^{K\star T}_{C_1} \mathbf{S}_C^{-1} \mathbf{H}^{K\star}_{C_1} \mathbf{P}^{\ominus}_{I_{k+1:k+2}} \right)$.

3.3.5 Construction of Next Epoch's Prior

During this last step of the IKS, we will bring c^M_{k+M} into a form whose minimization is independent of $\mathbf{x}_{I_{k+1}}$. To achieve this, we follow a 2-step process.

Partitioning of $c_{P^{\oplus}_{I_{k+1:k+2}}}$: By employing the Schur complement, the prior term $c_{P^{\oplus}_{I_{k+1:k+2}}}$ in (22) is partitioned into a prior over $\mathbf{x}_{I_{k+2}}$, $c_{P^{\oplus}_{I_{k+2}}}$, and a conditional term, $c_{I_{k+1|k+2}}$, representing linearized constraints between $\mathbf{x}_{I_{k+1}}$ and $\mathbf{x}_{I_{k+2}}$, i.e.,

$$c_{P^{\oplus}_{I_{k+1:k+2}}}(\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+2}}) = ||\tilde{\mathbf{x}}^{\star}_{I_{k+2}} - \delta\mathbf{x}^{\oplus}_{I_{k+2}}||_{\mathbf{P}^{\oplus}_{I_{k+2}}} + ||\delta\mathbf{x}^{\oplus}_{I_{k+1|k+2}} - \begin{bmatrix}\mathbf{I} - \mathbf{P}^{\oplus}_{I_{k+1,k+2}}\mathbf{P}^{\oplus-1}_{I_{k+2}}\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{x}}^{\star}_{I_{k+1}}\\ \tilde{\mathbf{x}}^{\star}_{I_{k+2}}\end{bmatrix}||^2_{\mathbf{P}^{\oplus}_{I_{k+1|k+2}}}$$
$$= c_{P^{\oplus}_{I_{k+2}}}(\tilde{\mathbf{x}}_{I_{k+2}}) + c_{I_{k+1|k+2}}(\tilde{\mathbf{x}}_{I_{k+1:k+2}}) \quad (23)$$

where $\mathbf{P}^{\oplus}_{I_{k+1|k+2}} = \mathbf{P}^{\oplus}_{I_{k+1}} - \mathbf{P}^{\oplus}_{I_{k+1,k+2}}\mathbf{P}^{\oplus-1}_{I_{k+2}}\mathbf{P}^{\oplus}_{I_{k+2,k+1}}$. Substituting in (22), yields:

$$c^{M}_{k+M} = c_{P^{\oplus}_{I_{k+2}}} + c_{I_{k+1|k+2}} + c^{K}_{C_2} + c^{M}_{C}. \quad (24)$$

Marginalization of $\mathbf{x}_{I_{k+1}}$: Firstly, we combine all terms involving $\mathbf{x}_{I_{k+1}}$, i.e., $c_{I_{k+1|k+2}}$, $c^{K}_{C_2}$, and c^{M}_{C} into a single quadratic cost, corresponding to $15+N^{K}_{C_2}+3N_{k+M}$ linearized constraints:

$$c_J(\tilde{\mathbf{x}}_{I_{k+1:k+M}}, {}^{I_{k+2}}\widetilde{\boldsymbol{\ell}}^{\star}_{k+M}) = ||\mathbf{b} - \mathbf{J}_1\tilde{\mathbf{x}}^{\star}_{I_{k+1}} - \mathbf{J}_2\begin{bmatrix}\tilde{\mathbf{x}}^{\star}_{I_{k+2:k+M}}\\ {}^{I_{k+2}}\widetilde{\boldsymbol{\ell}}^{\star}_{k+M}\end{bmatrix}||^2_{\mathbf{I}_{15+N^{K}_{C_2}+3N_{k+M}}} \quad (25)$$

Following the same process as in (6), we partition c_J into $c_{I_{k+1|k+2:k+M}}$, that contains all information regarding $\mathbf{x}_{I_{k+1}}$, and $c_{L^{\oplus}}$ and $c_{M^{\oplus}}$, which are independent of $\mathbf{x}_{I_{k+1}}$:

$$c_J(\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+M}}, {}^{I_{k+2}}\widetilde{\boldsymbol{\ell}}^{\star}_{k+M}) = c_{I_{k+1|k+2:k+M}} + c_{L^{\oplus}} + c_{M^{\oplus}}. \quad (26)$$

where the detailed analytical expressions for $c_{L^{\oplus}}$ and $c_{M^{\oplus}}$ are given in [17].

Substituting (26) in (24), yields:

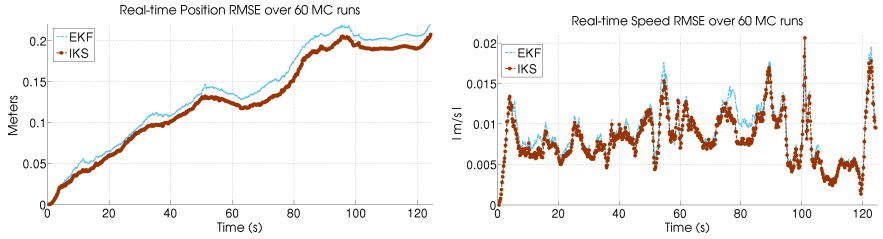
$$c^{M}_{k+M}(\tilde{\mathbf{x}}^{\star}_{I_{k+1:k+M}}, {}^{I_{k+2}}\widetilde{\boldsymbol{\ell}}^{\star}_{k+M}) = c_{P^{\oplus}_{I_{k+2}}}(\tilde{\mathbf{x}}^{\star}_{I_{k+2}}) + c_{L^{\oplus}}(\tilde{\mathbf{x}}^{\star}_{I_{k+2:k+M}}) + c_{M^{\oplus}}(\tilde{\mathbf{x}}^{\star}_{I_{k+2:k+M}}, {}^{I_{k+2}}\widetilde{\boldsymbol{\ell}}^{\star}_{k+M})$$
$$+ c_{I_{k+1|k+2:k+M}}(\tilde{\mathbf{x}}^{\star}_{I_{k+2:k+M}}, {}^{I_{k+2}}\widetilde{\boldsymbol{\ell}}^{\star}_{k+M}). \quad (27)$$

The last term, $c_{I_{k+1|k+2:k+M}}$ in (27), is irrelevant for the minimization of c^{M}_{k+M} over $\tilde{\mathbf{x}}^{\star}_{I_{k+2:k+M}}$ and ${}^{I_{k+2}}\widetilde{\boldsymbol{\ell}}^{\star}_{k+M}$ since, for any of their values, there exists a $\tilde{\mathbf{x}}^{o}_{I_{k+1}}$ that minimizes $c_{I_{k+1|k+2:k+M}}$ to *exactly* zero. Hence, all prior information from the current to the next IKS recursion, is represented completely through the terms $c_{P^{\oplus}_{k+2}}$, $c_{L^{\oplus}}$, and $c_{M^{\oplus}}$ all of which do *not* involve $\tilde{\mathbf{x}}^{\star}_{I_{k+1}}$.

4 Simulations

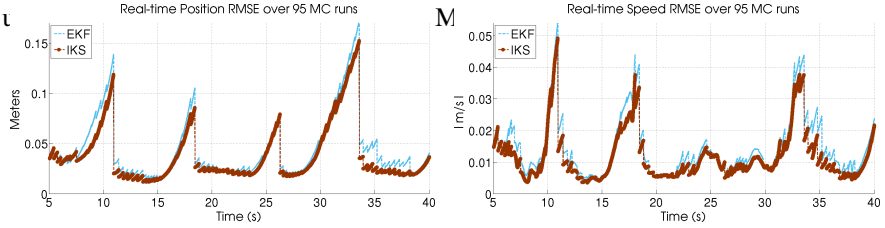
Our simulations involved a MEMS-quality commercial grade IMU, similar to those present on current mobile devices, running at 100 Hz, and a wide (175^{o} degrees) field of view camera with resolution 640×480. Visual observations were contaminated by zero-mean white Gaussian noise with $\sigma = 1.0$ pixel. We compared the Root Mean Square Error (RMSE) for the real-time estimates[6] of the MSC-KF VINS (denoted as EKF), with that of the proposed iterative Kalman smoother (denoted as IKS). Both estimators maintained a sliding window of 10 camera poses. Feature tracks that spanned beyond the sliding window were initialized as SLAM landmarks

[6] By real-time, we refer to the estimate for $\mathbf{x}_{I_k}$, right before, processing image k.



(a) (b)

Fig. 2: Monte Carlo Simulation I: Comparison of the proposed IKS versus the EKF



(a) (b)

Fig. 3: Monte Carlo Simulation II: Comparison of the proposed IKS versus the EKF during camera occlusions: (a) Position RMSE; (b) Speed RMSE

(and marginalized when lost from the camera's field of view), while shorter feature tracks were processed as MSC-KF features.

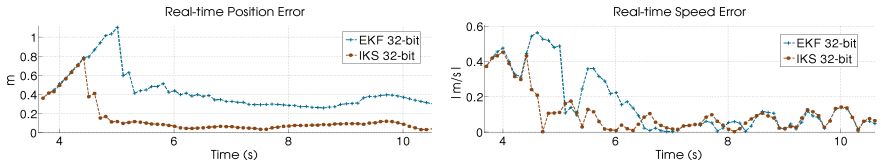
- VINS under nominal conditions: In Simulation I (see Fig. 2), the platform's trajectory and dynamics resembled those of a person traversing 120 m of an indoor environment, while new camera poses were generated every 25 cm, and the rate that new features enter the camera's field of view followed that of real-world experimental trials. As seen in Fig. 2(a), the performance difference between the EKF-based VINS and the proposed IKS is rather small, since in the presence of many visual measurements, both estimators are able to accurately track the system's state. Note however, that even under these nominal conditions the IKS always maintained a more accurate estimate of the platform's speed (see Fig. 2(b)), while for certain parts of the trajectory its estimate improved by $\sim 20\%$, over the EKF, due to the inability of the latter to process feature tracks immediately as they become available.

- Camera Occlusions: In Simulation II (see Fig. 3), we simulated the motion of a handheld device, "hovering" over the same scene, for 40 s, emulating a common scenario for augmented-reality applications. We then introduced 3 periods, of approximately 5 s each, during which the camera was occluded and no feature tracks were available. As evident from Figs. 3(a) and 3(b), by re-processing visual and inertial measurements, the IKS, converges faster to the correct position and velocity estimates, right after re-gaining access to camera measurements.

5 Experiments

We further validated the proposed IKS on real-world data, using a Project Tango developer tablet, and as ground truth the result of a batch least-squares (BLS) over the entire set of visual and inertial measurements. As in Section 4, we compared the real-time estimates of the proposed IKS, to those of the MSC-KF VINS, both of which processed measurements to SLAM landmarks, as well as MSC-KF feature tracks, and maintained a sliding window of length 14.

Initialization: In Experiment I (Fig. 4), we compared the initialization phase, of both estimators, when they start from inaccurate estimates of their initial velocity and IMU biases, as often happens in practice. As it is seen in Fig. 4(b), the IKS converged faster to its correct velocity estimates, which lead to a reduced position error, as compared to the EKF, for the rest of their trajectories [Fig. 4(a)]. Note that a BLS over a small set of initial poses, could have been used for system initialization, making both estimators equally accurate. Such a choice, however, would inevitably introduce a significant time delay, a key drawback for real-time applications, while the complexity of the corresponding implementation would significantly increase. For the IKS, however, an iterative optimization during its initial phase, seamlessly takes place, without the need to transition from BLS to filtering.



(a) (b)

Fig. 4: Experiment I: Comparison of the proposed IKS versus the EKF during filter initialization: (a) Real-time position error; (b) Real-time speed error

Fast camera turns: In Experiment II (Fig. 5), we collected a "stress" dataset, during which the camera performed quick turns, inside an office area, causing abrupt reductions in the number, quality, and length of feature tracks for short periods of time [Fig. 5(a)]. As it is evident from Fig. 5(d), the inability of the EKF to re-process visual observations, caused sudden accumulations of yaw error, e.g., at 40 s. Furthermore, on par with our simulation results, the IKS maintained an improved real-time estimate, of the platform's velocity, throughout the experiment, while at certain points, its velocity estimate was even 2 times better than the EKF's [Fig. 5(c)].

Timing analysis: We used the Samsung S4 cell phone, as a testbed for comparing the processing time of the proposed IKS, *with and without processing of immature visual observations*, denoted by IKS w/, and IKS w/o, respectively, as well as, a reference EKF implementation. Albeit the 32-bit arithmetic precision, of the NEON co-processor, present on the 1.6 GHz Cortex-A15 ARM CPU of the Samsung S4, no numerical inaccuracies were introduced, when compared to 64-bit arithmetic precision. All estimators maintained a sliding window of 14 poses and on average 5 SLAM landmarks in their state vector. As it is evident from Table 1, the proposed IKS achieves real-time performance, even under re-linearization, while it

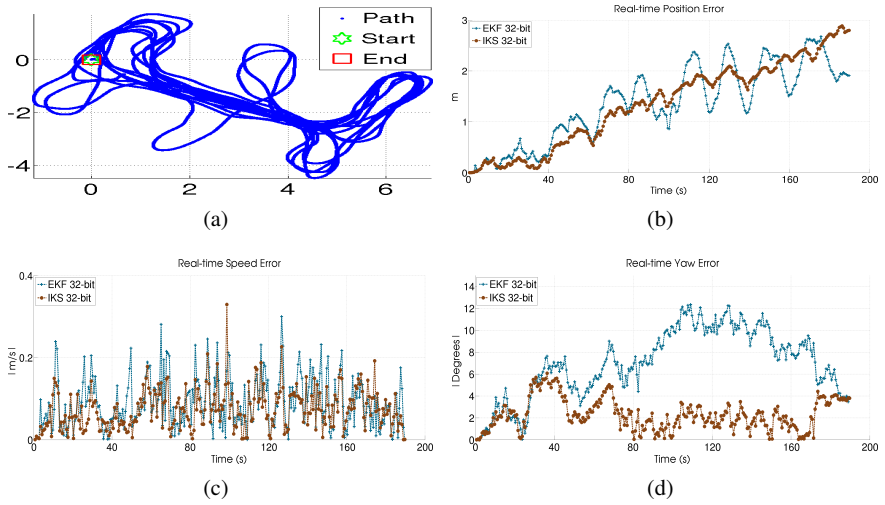


Fig. 5: Experiment II: Comparison of the proposed IKS versus the EKF during sudden turns: (a) 3D Trajectory; (b) Position error; (c) Speed error; (d) Yaw error

is able to bring its cost down to levels comparable to the EKF by temporary disabling the re-processing of visual observations.

Step \Algorithm	**IKS w/**	**IKS w/o**	**EKF**
Propagation	14	14	1
Jacobians Calculation	101	12	9
Measurement Compressions	14	6	2
State Update	12	12	12
Covariance Update	0.5	0.5	18
Prior Constraints Update	1.5	1.5	N/A
Total	166	46	42

Table 1: Timing analysis on the Samsung S4: Numbers denote average time in ms

6 Conclusion

In this paper, we have presented an iterative Kalman smoother (IKS) for vision-aided inertial navigation that incorporates advantages of competing approaches. Through smoothing, the proposed IKS iteratively re-linearizes both inertial and visual measurements over a single, or multiple overlapping, sliding windows, thus improving robustness. At the same time, the IKS inherits the excellent numerical properties of the Kalman filter, making it amenable to very efficient implementations (4-fold speed up on ARM NEON co-processor) using single-precision (32 bit) arithmetic. As part of our validation process, we demonstrated the resilience and efficiency of the proposed approach, under adverse navigation conditions.

References

1. J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Consistency analysis and improvement of vision-aided inertial navigation," *IEEE Trans. on Robotics*, vol. 30, pp. 158–176, Feb. 2014.
2. A. I. Mourikis, N. Trawny, S. I. Roumeliotis, A. E. Johson, A. Ansar, and L. Matthies, "Vision-aided inertial navigation for spacecraft entry, descent, and landing," *IEEE Trans. on Robotics*, vol. 25, pp. 264–280, Apr. 2009.
3. E. D. Nerurkar, K. J. Wu, and S. I. Roumeliotis, "C-KLAM: Constrained Keyframe Localization and Mapping for long-term navigation," in *Proc. of the IEEE International Conference on Robotics and Automation*, (Hong Kong, China), pp. 3638–3643, May 31 – June 6 2013.
4. A. Jazwinski, *Stochastic processes and filtering theory*. No. 64 in Mathematics in science and engineering, New York, Academic Press, 1970.
5. J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Towards consistent vision-aided inertial navigation," in *Proc. of the 10th International Workshop on the Algorithmic Foundations of Robotics (WAFR'12)*, (Cambridge, Massachusetts), pp. 559–574, June 13–15 2012.
6. C. Guo, D. Kottas, R. DuToit, A. Ahmed, R. Li, and S. Roumeliotis, "Efficient visual-inertial navigation using a rolling-shutter camera with inaccurate timestamps," in *Proceedings of Robotics: Science and Systems*, (Berkeley, USA), July 2014.
7. G. Sibley, L. Matthies, and G. Sukhatme, "Sliding window filter with application to planetary landing," *Journal of Field Robotics*, vol. 27, no. 5, pp. 587–608, 2010.
8. G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "An Observability-Constrained Sliding Window Filter for SLAM," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, (San Francisco, CA), pp. 65–72, Sept. 25 – 30 2011.
9. H.-P. Chiu, S. Williams, F. Dellaert, S. Samarasekera, and R. Kumar, "Robust vision-aided navigation using Sliding-Window Factor Graphs," in *Proc. of the IEEE International Conference on Robotics and Automation*, (Karlsruhe, Germany), pp. 46–53, May 6 – 10 2013.
10. S. Leutenegger, P. T. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart, "Keyframe-Based Visual-Inertial SLAM using Nonlinear Optimization," in *Proceedings of Robotics: Science and Systems*, (Berlin, Germany), June 2013.
11. K. J. Wu, A. Medhat, G. Georgiou, and S. I. Roumeliotis, "A square root inverse filter for efficient vision-aided inertial navigation on mobile devices," in *Robotics: Science and Systems*, (Rome, Italy), July 13 - 17 2015.
12. D. G. Kottas and S. I. Roumeliotis, "An iterative Kalman smoother for robust 3D localization on mobile and wearable devices," in *Proc. of the IEEE International Conference on Robotics and Automation*, (Seattle, WA), pp. 6336–6343, May 26 – 30 2015.
13. F. M. Mirzaei and S. I. Roumeliotis, "A Kalman Filter-based algorithm for IMU-camera calibration: Observability analysis and performance evaluation," *IEEE Trans. on Robotics*, vol. 24, pp. 1143–1156, Oct. 2008.
14. N. Trawny and S. I. Roumeliotis, "Indirect Kalman Filter for 3D Attitude Estimation," tech. rep., University of Minnesota, Dept. of Comp. Sci. & Eng., March 2005.
15. C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. of the Alvey Vision Conference*, (Manchester, UK), pp. 147–151, Aug. 31 – Sept. 2 1988.
16. B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. of the International Joint Conference on Artificaial Intelligence*, (Vancouver, British Columbia), pp. 674–679, Aug. 24 – 28 1981.
17. D. G. Kottas and S. I. Roumeliotis, "An iterative Kalman smoother for robust 3D localization and mapping, http://www-users.cs.umn.edu/~dkottas/pdfs/iks_slam_2014_tr.pdf," tech. rep., University of Minnesota, Dept. of Comp. Sci. & Eng., MARS Lab, Apr. 2015.