An Iterative Kalman Smoother for Robust 3D Localization on Mobile and Wearable devices

Dimitrios G. Kottas and Stergios I. Roumeliotis

Abstract— **In this paper, we introduce an Iterative Kalman Smoother (IKS) for tracking the 3D motion of a mobile device in real-time using visual and inertial measurements. In contrast to existing Extended Kalman Filter (EKF)-based approaches, smoothing can better approximate the underlying nonlinear system and measurement models by re-linearizing them. Additionally, by iteratively optimizing over all measurements available, the IKS increases the convergence rate of critical parameters (e.g., IMU-camera clock drift) and improves the positioning accuracy during challenging conditions (e.g., scarcity of visual features). Furthermore, and in contrast to existing inverse filters, the proposed IKS's numerical stability allows for efficient 32-bit implementations on resource-constrained devices, such as cell phones and wearables. We validate the IKS for performing vision-aided inertial navigation on Google Glass, a wearable device with limited sensing and processing, and demonstrate positioning accuracy comparable to that achieved on cell phones. To the best of our knowledge, this work presents the first proof-of-concept real-time 3D indoor localization system on a commercial-grade wearable computer.**

I. Introduction and Related Work

Navigating in GPS-denied areas (e.g., spacecraft [1] or personal localization [2], [3]) often requires combining visual observations, from a camera, with inertial data, from an inertial measurement unit (IMU), in what is known as a vision-aided inertial navigation system (VINS). Existing approaches to VINS typically rely on filtering or batch least-squares (BLS). BLS methods optimize over the sensors' entire trajectory as well as over the map of the environment, comprising, e.g., point features. This leads to processing and memory requirements that inevitably increase with time [4]. On the other hand, filtering approaches (e.g., [1], [2], [5], [6]) optimize over a bounded-size sliding window of recent camera poses while marginalizing past poses and measurements, thus achieving constant processing time.

Filtering approaches can be further classified into extended Kalman filters (EKF) and inverse filters (INVF). EKF-based algorithms compute, up to linearization errors, the minimum mean square error (MMSE) estimate and exhibit excellent numerical properties, which make them an attractive choice for performing real-time VINS. One of the main limitations of the EKF, however, is its inability to re-linearize *all* measurements available. In particular, when applied to VINS for computing the MMSE estimate of a sliding window of camera poses [1], only the visual observations that are about to be absorbed into the estimator's prior, can be re-linearized, using the iterative form of the EKF [7], but *not* the inertial measurements. Furthermore, the EKF cannot re-process the same visual measurements multiple times (as soon as they appear in the sliding window till they are absorbed) for improving the state estimates' accuracy.

These desirable characteristics of a filtering algorithm are found in the INVF (e.g., [5], [8], [9], [10], [6]), which provides a straightforward mechanism for re-linearizing all measurements and re-processing the visual observations for as long as they are within the optimization window considered. Despite its mathematical elegance and simplicity, however, the INVF does not exhibit the numerical robustness of the EKF. Specifically, the Hessian employed by the INVF has a high condition number, which requires 64-bit numerical precision for updating it and for computing the state estimates. This can be a limiting factor, especially when considering that most current cell phones and wearable devices feature ARM processors and NEON co-processors that provide a 4-fold processing speedup when using 32-bit precision.

In order to overcome the limitations of the EKF and INVF when applied to VINS, in this work we introduce a sliding window Iterative Kalman Smoother (IKS), which offers 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 matrix) for representing prior information, thus inheriting the superior numerical properties of the EKF and leading to efficient implementations.

To validate the robustness and accuracy of the proposed approach, we demonstrate its application to indoor VINS using a wearable computer (Google Glass [11]) with limited processing and sensing capabilities, and show that it achieves positioning accuracy comparable to that of cell phones [12].

II. Vision-aided Inertial Navigation (VINS)

A. System State

The system state[1] at time t_k is given by $\mathbf{x}_k = \begin{bmatrix}\mathbf{x}_{C_k}^T & \mathbf{x}_{I_k}^T\end{bmatrix}^T$ where $\mathbf{x}_{C_k} = \begin{bmatrix}{}^{I_k}\mathbf{q}_G^T & {}^G\mathbf{p}_{I_k}^T\end{bmatrix}^T$ and $\mathbf{x}_{I_k} = \begin{bmatrix}{}^G\mathbf{v}_{I_k}^T & \mathbf{b}_{a_k}^T & \mathbf{b}_{g_k}^T\end{bmatrix}^T$,

The authors are with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455 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.

[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 [12], respectively.

while ${}^{I_k}\mathbf{q}_G$ is the quaternion representation of the orientation of the global frame $\{G\}$ in the IMU's frame of reference $\{I_k\}$, ${}^{G}\mathbf{v}_{I_k}$ 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.

B. Inertial Measurements

The IMU provides measurements of the platform's rotational velocity and linear acceleration, contaminated by white Gaussian noise and time-varying biases. The time-varying biases are modelled as random walks driven by white zero-mean Gaussian noise processes $\mathbf{n}_{wg}(t)$ and $\mathbf{n}_{wa}(t)$, with auto-correlations $\mathbb{E}[\mathbf{n}_{wg}(t)\mathbf{n}_{wg}^T(\tau)] = \mathbf{Q}_{wg}\delta(t-\tau)$ and $\mathbb{E}[\mathbf{n}_{wa}(t)\mathbf{n}_{wa}^T(\tau)] = \mathbf{Q}_{wa}\delta(t-\tau)$, respectively.[2] The gyroscope and accelerometer measurements, $\omega_m(t)$ and $\mathbf{a}_m(t)$ are given by:

$$\begin{aligned}\omega_m(t) &= {}^{I}\omega(t) + \mathbf{b}_g(t) + \mathbf{n}_g(t) \\ \mathbf{a}_m(t) &= \mathbf{C}({}^{I}\mathbf{q}_G(t))({}^{G}\mathbf{a}(t) - {}^{G}\mathbf{g}) + \mathbf{b}_a(t) + \mathbf{n}_a(t)\end{aligned} \tag{1}$$

where the rotational matrix $\mathbf{C}({}^{I}\mathbf{q}_G(t))$ represents the orientation of frame $\{G\}$, expressed in the IMU frame $\{I\}$, parameterized by the unit quaternion ${}^{I}\mathbf{q}_G(t)$. The noise terms, $\mathbf{n}_g(t)$ and $\mathbf{n}_a(t)$ are modelled as zero-mean white Gaussian noise processes, while the gravitational acceleration ${}^{G}\mathbf{g}$ is considered a deterministic constant. The platform's rotational velocity ${}^{I}\omega(t)$ and linear acceleration ${}^{G}\mathbf{a}(t)$, in (1), relate consecutive camera poses through the platform's kinematic equations and the random-walk model of the IMU's biases:

$$\begin{aligned}{}^{I}\dot{\mathbf{q}}_G(t) &= \frac{1}{2}\boldsymbol{\Omega}(\omega_m(t) - \mathbf{b}_g(t) - \mathbf{n}_g(t))\,{}^{I}\mathbf{q}_G(t) \\ {}^{G}\dot{\mathbf{v}}_I(t) &= \mathbf{C}({}^{I}\mathbf{q}_G(t))^T(\mathbf{a}_m(t) - \mathbf{b}_a(t) - \mathbf{n}_a(t)) + {}^{G}\mathbf{g} \\ {}^{G}\dot{\mathbf{p}}_I(t) &= {}^{G}\mathbf{v}_I(t) \qquad \dot{\mathbf{b}}_a(t) = \mathbf{n}_{wa}(t) \qquad \dot{\mathbf{b}}_g(t) = \mathbf{n}_{wg}(t)\end{aligned} \tag{2}$$

where, $\boldsymbol{\Omega}(\boldsymbol{\omega}) \triangleq \begin{bmatrix} -\lfloor\boldsymbol{\omega}\rfloor & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix}$ for $\boldsymbol{\omega} \in \mathbb{R}^3$. Let $\mathbf{u}_{k,k+1}$ denote inertial measurements within the time interval $\begin{bmatrix} t_k, & t_{k+1} \end{bmatrix}$. Numerical or analytical integration of (2) allows us to define a constraint (discrete-time process model) of the form:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k,\ \mathbf{u}_{k,k+1} - \mathbf{w}_{k,k+1}) \tag{3}$$

where $\mathbf{w}_{k,k+1}$ is a discrete-time zero-mean white Gaussian noise process with covariance $\mathbf{Q}_k$ [2]. Linearizing (3), at the state estimates corresponding to the two consecutive states, $\mathbf{x}_k^\star$, $\mathbf{x}_{k+1}^\star$, results in the following IMU measurement model, relating the error states $\widetilde{\mathbf{x}}_k^\star$ and $\widetilde{\mathbf{x}}_{k+1}^\star$:

$$\widetilde{\mathbf{x}}_{k+1}^\star = \mathbf{r}_{u_{k,k+1}} + \boldsymbol{\Phi}_{k+1,k}^\star \widetilde{\mathbf{x}}_k^\star + \mathbf{G}_{k+1,k}^\star \mathbf{w}_{k,k+1} \tag{4}$$

where $\mathbf{r}_{u_{k,k+1}} := \mathbf{f}(\mathbf{x}_k^\star, \mathbf{u}_{k,k+1}) - \mathbf{x}_{k+1}^\star$, and we defined the error state $\widetilde{\mathbf{x}}_k^\star$ as the difference between the true state $\mathbf{x}_k$ and the linearization point $\mathbf{x}_k^\star$ (i.e., $\widetilde{\mathbf{x}}_k^\star = \mathbf{x}_k - \mathbf{x}_k^\star$), while for the quaternion $\mathbf{q}$ we employ a multiplicative error model $\widetilde{\mathbf{q}} = \mathbf{q} \otimes \mathbf{q}^{\star-1} \simeq \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta}^T & 1 \end{bmatrix}^T$, where $\delta\boldsymbol{\theta}$ is a minimal representation of the attitude error. The Jacobians $\boldsymbol{\Phi}_{k+1,k}^\star$ and $\boldsymbol{G}_{k+1,k}^\star$ are evaluated at $\mathbf{x}_k^\star$, $\mathbf{x}_{k+1}^\star$, and are available in numerical or analytical form [2]. Note, however, that EKF-based approaches linearize (3) only *once*, during state and covariance propagation (i.e., every time a new inertial measurement becomes available). In the INVF, the inertial measurements $\mathbf{u}_{k,k+1}$, contribute a linearized cost term of the form:

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

where $\tilde{\mathbf{x}}_{k:k+1}^\star := \begin{bmatrix} \tilde{\mathbf{x}}_k^{\star T} & \tilde{\mathbf{x}}_{k+1}^{\star T} \end{bmatrix}^T$ and $\mathbf{Q}_k^\star = \mathbf{G}_{k+1,k}^\star \mathbf{Q}_k \mathbf{G}_{k+1,k}^{\star T}$, which is re-linearized multiple times.

C. 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 [14]) along with a tracking algorithm (e.g., the Kanade-Lucas-Tomasi (KLT) [15]). Once a new image arrives, point features from the previous one, are tracked to the new image, while new features are extracted from areas that just entered the camera's field of view [2]. An example of the feature tracks generated from such an image-processing pipeline is shown in Fig. 1. Consider a point feature $\mathbf{f}_j$, with Euclidean coordinates ${}^{G}\mathbf{p}_{f_j}$, observed in camera poses $\mathbb{C}_{\mathbf{f}_j} = \{\mathbf{x}_{k+1:k+N_j}\}$, where $N_j \leq M$ and M is the window's length. In VINS, such observations can be used to obtain a constraint between these poses. Specifically, 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 [16] is:

$$\begin{aligned}\mathbf{z}_{k+m,j} &= \mathbf{h}(\mathbf{x}_{k+m}, {}^{G}\mathbf{p}_{f_j}) + \mathbf{n}_{k+m,j} \\ &= \pi(\mathbf{C}({}^{I_{k+m}}\mathbf{q}_G)({}^{G}\mathbf{p}_{f_j} - {}^{G}\mathbf{p}_{I_{k+m}})) + \mathbf{n}_{k+m,j}\end{aligned} \tag{5}$$

where, $\pi(\begin{bmatrix} x & y & z \end{bmatrix}^T) = \begin{bmatrix} \frac{x}{z} & \frac{y}{z} \end{bmatrix}^T$, while $\mathbf{n}_{k+m,j}$ is zero-mean white Gaussian noise with covariance $\sigma^2\mathbf{I}_2$, and $\mathbf{I}_2$ is the 2×2 identity matrix. Linearizing (5), yields:

$$\tilde{\mathbf{z}}_{k+m,j}^\star = \mathbf{H}_{R,k+m,j}^\star \tilde{\mathbf{x}}_{k+m}^\star + \mathbf{F}_{k+m,j}^\star {}^{G}\tilde{\mathbf{p}}_{f_j}^\star + \mathbf{n}_{k+m,j}$$

Collecting all N_j feature observations in one vector, we get:

$$\tilde{\mathbf{z}}_j^\star = \mathbf{H}_{R,j}^\star \tilde{\mathbf{x}}_{k+1:k+N_j}^\star + \mathbf{F}_j^{\star G}\tilde{\mathbf{p}}_{f_j}^\star + \mathbf{n}_j \tag{6}$$

which corresponds to the cost term:

$$c_{F_{f_j}}(\tilde{\mathbf{x}}_{k+1:k+N_j}^\star, {}^{G}\tilde{\mathbf{p}}_{f_j}) = ||\tilde{\mathbf{z}}_j^\star - \mathbf{H}_{R,j}^\star \tilde{\mathbf{x}}_{k+1:k+N_j}^\star - \mathbf{F}_j^{\star G}\tilde{\mathbf{p}}_{f_j}||_{\sigma^2\mathbf{I}}^2 \tag{7}$$

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. Using Givens rotations to project (6) onto $\boldsymbol{\Theta}_j$, we partition $c_{F_{f_j}}$ into two parts:

$$\begin{aligned}&c_{F_{f_j}}(\tilde{\mathbf{x}}_{k+1:k+N_j}^\star, {}^{G}\tilde{\mathbf{p}}_{f_j}) \\ &= ||\boldsymbol{\Theta}_j^T\tilde{\mathbf{z}}_j^\star - \boldsymbol{\Theta}_j^T\mathbf{H}_{R,j}^\star\tilde{\mathbf{x}}_{k+1:k+N_j}^\star - \boldsymbol{\Theta}_j^T\mathbf{F}_j^{\star G}\tilde{\mathbf{p}}_{f_j}||_{\sigma^2\mathbf{I}_{2N_j}}^2 \\ &= ||\mathbf{r}_j^\star - \mathbf{H}_j^\star\tilde{\mathbf{x}}_{k+1:k+N_j}^\star||_{\sigma^2\mathbf{I}_{2N_j-3}}^2 \\ &+ ||\mathbf{r}_{M,j}^\star - \mathbf{H}_{M,j}^\star\tilde{\mathbf{x}}_{k+1:k+N_j}^\star - \mathbf{R}_{M,j}^{\star}{}^{G}\tilde{\mathbf{p}}_{f_j}||_{\sigma^2\mathbf{I}_3}^2 \\ &= c_{Z_{f_j}}(\tilde{\mathbf{x}}_{k+1:k+N_j}^\star) + c_{Z_{M,f_j}}(\tilde{\mathbf{x}}_{k+1:k+N_j}^\star, {}^{G}\tilde{\mathbf{p}}_{f_j})\end{aligned} \tag{8}$$

with $\mathbf{r}_j^\star = \mathbf{U}_j^T\tilde{\mathbf{z}}_j^\star$, $\mathbf{H}_j^\star = \mathbf{U}_j^T\mathbf{H}_{R,j}^\star$, and $\mathbf{r}_{M,j}^\star = \mathbf{S}_j^T\tilde{\mathbf{z}}_j^\star$, $\mathbf{H}_{M,j}^\star = \mathbf{S}_j^T\mathbf{H}_{R,j}^\star$, $\mathbf{R}_{M,j}^\star = \mathbf{S}_j^T\mathbf{F}_{R,j}^\star$. The second term, $c_{Z_{M,f_j}}$ contains *all* information regarding feature f_j, while $c_{Z_{f_j}}$ defines a multi-state constraint *only* among the poses $\mathbf{x}_{k+1:k+N_j}$. As we will see later on, such a partitioning of quadratic terms that are functions of two sets of variables will be particularly useful

[2] Their auto-correlation is estimated using an offline calibration process.

throughout the IKS' steps.

For the rest of this work, as in [1], we will consider *only* the cost term $c_{Z_{f_j}}(\tilde{\mathbf{x}}^{\star}_{k+1:k+N_j})$. Note, however, that minimizing $c_{Z_{f_j}}(\tilde{\mathbf{x}}^{\star}_{k+1:k+N_j})$ is equivalent to minimizing all terms in (8), since $\mathbf{R}^{\star}_{M,j}$ is an invertible square matrix, and *for any* $\tilde{\mathbf{x}}^{\star}_{k+1:k+N_j}$, *there exists* a ${}^{G}\tilde{\mathbf{p}}_{f_j}$, such that $c_{Z_{M,f_j}}$ is exactly zero.

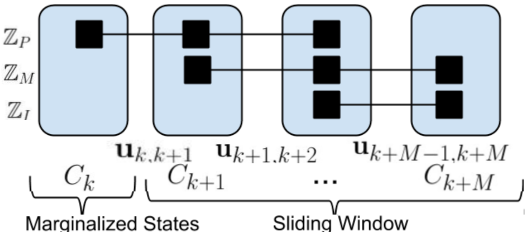


Fig. 1. At time-step $k+M$, there exist 3 categories of feature tracks: (i) $\mathbb{Z}_P$: features that were previously absorbed; (ii) $\mathbb{Z}_M$: features that have reached their maximum length and are about to be absorbed; and (iii) $\mathbb{Z}_I$: feature tracks that entered the estimator's sliding window but have not reached their maximum length. Accordingly, inertial measurements $\{\mathbf{u}_{\ell,\ell+1}\}$ can be divided into 3 sets for different values of ℓ: (i) absorbed ($\ell \leq k$); (ii) those that are about to be absorbed in the current epoch ($\ell = k+1$); (iii) those that will be absorbed in the future ($\ell \geq k+1$, $\ell \leq k+M-1$), and thus can be re-linearized.

D. Visual Observations in Sliding-Window Estimators

In order to process the multi-state constraints provided by visual feature tracks, VINS maintains a sliding window of past camera poses, $\mathbf{x}_{k+1:k+M}$ (see Fig. 1). Within an INVF, the same measurements are available for processing at different time-instants, corresponding to different epochs of the sliding window. In the EKF, however, (e.g., [1], [2]) each feature track can be used only *once* (when it has reached its maximum length) for updating the current state and covariance estimates. In particular, there exist 3 categories of feature tracks (see Fig. 1):

- **Past Features ($\mathbb{Z}_P$)**: These correspond to visual observations that were absorbed in a past epoch of the sliding window and cannot be re-processed by any filter.
- **Mature Features ($\mathbb{Z}_M$)**: These are feature tracks that have reached their maximum length. Both the EKF and the INVF linearize, process, and absorb them at a single step. Note also that the INVF, as well as the Iterative EKF (I-EKF) [7] can re-linearize these observations.
- **Immature Features ($\mathbb{Z}_I$)**: This set represents feature tracks which 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 time they are first observed till the track exits the optimization window), the EKF is not able to.

The delay in processing the $\mathbb{Z}_I$'s is a major limitation of the EKF which negatively impacts its robustness and accuracy, especially when operating under adverse conditions (e.g., areas with limited number of features). Furthermore, the EKF is unable to re-linearize past IMU measurements, which leads to loss in accuracy due to the accumulation of linearization errors.

In contrast, existing INVF-based approaches allow for the re-linearization of all measurements within a sliding window. As we determined experimentally, however, maintaining the Hessian as a representation of prior information in VINS, comes at the expense of numerical instability. Specifically, the Hessian's high condition number ($\sim 10^9$) may not allow a numerically-stable implementation on a 32-bit floating-point unit, thus significantly increasing computational cost due to the required double-precision arithmetic.

At this point, we should note that the EKF's limitations (i.e., delayed processing of $\mathbb{Z}_I$'s and inability to re-linearize the $\mathbf{u}_{k+1:k+M}$ inertial measurements) stem from the fact that at each optimization epoch the EKF constructs and uses a prior, $\mathcal{N}(\hat{\mathbf{x}}^{\ominus}_{k+1:k+M}, \mathbf{P}^{\ominus}_{k+1:k+M})$, for the entire sliding window using the inertial measurements $\mathbf{u}_{k+1:k+M}$. This is necessary for processing feature tracks that may span the entire optimization window [see (8)]. Once the covariance $\mathbf{P}_{k+1:k+M}$ has been updated, it cannot be "rolled back" to a prior $\mathbf{P}_{k+1}$ corresponding only to $\mathbf{x}_{k+1}$, which is required for re-using the inertial measurements in the next epoch. Furthermore, once a feature track has been processed for updating the covariance, it cannot be re-used in the future. As it will become evident in the next section, both these limitations of the EKF can be addressed by introducing the following two key novelties of the proposed IKS:

1) Divide the update of the EKF into a two-step process: (i) Update the state using all available inertial and visual measurements, and (ii) Update the covariance, using only $\mathbf{u}_{k+1:k+2}$ and the feature tracks $\mathbb{Z}_M$ that are about to be absorbed. This decoupling of the two processes will allow us to *re-process* the same inertial and feature measurements multiple times and hence gain in state estimation accuracy. At the same time, by considering their contribution only once when updating the covariance, we ensure consistency.
2) Define priors, as a combination of:
(i) A "regular" prior, $\mathcal{N}(\hat{\mathbf{x}}^{\ominus}_{k+1}, \mathbf{P}^{\ominus}_{k+1})$, comprising a state estimate and the corresponding covariance matrix.
(ii) A set of linearized constraints,
$$\mathbf{r}^{\star\ominus}_{L} = \mathbf{H}^{\star\ominus}_{L} \tilde{\mathbf{x}}^{\star}_{k+1:k+M} + \mathbf{n}_L \quad (9)$$
with $\mathbf{n}_L \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_{N_L})$, describing the information available for states whose covariance potentially cannot be initialized, i.e., $\mathbf{H}^{\ominus T}_{L} \mathbf{H}^{\ominus}_{L}$ is rank deficient.
Note that this division of the prior into two parts, and thus two cost terms:
$$||\tilde{\mathbf{x}}^{\star}_{k+1} - (\hat{\mathbf{x}}^{\ominus}_{k+1} - \mathbf{x}^{\star}_{k+1})||^2_{\mathbf{P}^{\ominus}_{k+1}} + ||\mathbf{r}^{\star\ominus}_{L} - \mathbf{H}^{\star\ominus}_{L} \tilde{\mathbf{x}}^{\star}_{k+1:k+M}||^2_{\sigma^2 \mathbf{I}}$$
is not necessary in the INVF, since it employs the Hessian $\mathcal{H}^{\ominus}_{k+1:k+M} = \begin{bmatrix} \mathbf{P}^{\ominus -1}_{k+1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \frac{1}{\sigma^2} \mathbf{H}^{\star\ominus T}_{L} \mathbf{H}^{\star\ominus}_{L}$, which can be formed regardless if the covariance for all the states can be initialized or not (alas, at the expense of a high condition number).

As we will show in the next section, these two key insights will allow us to appropriately modify the EKF equations so that the resulting IKS acquires the desirable characteristics of the INVF (immediate processing of all visual observations and re-linearization of all measurements) while maintaining its superior numerical properties.

III. Estimation Algorithm Description

When designing the proposed IKS our objective is twofold: (i) Process all inertial and visual observations within the current epoch of the sliding window, $\mathbf{x}_{k+1:k+M}$ (i.e., inertial measurements $\{\mathbf{u}_{\ell,\ell+1}\}$, for $k+1 \leq \ell \leq k+M-1$, and feature tracks $\mathbb{Z}_M$, and $\mathbb{Z}_I$), *and* (ii) Allow future epochs to reprocess all measurements that are independent of the sliding window's "tail" state, $\mathbf{x}_{k+1}$ (i.e., the inertial measurements $\{\mathbf{u}_{\ell,\ell+1}\}$, for $k+2 \leq \ell \leq k+M-1$, and the immature feature tracks $\mathbb{Z}_I$).

A. IKS Algorithm: Input

Before image $k+M$ arrives, the proposed IKS maintains:

- A set of linearization points $\{\mathbf{x}^\star_{k+1}, \dots, \mathbf{x}^\star_{k+M-1}\}$ that represent the estimator's best estimates given all measurements up to t_{k+M-1}.
- A prior comprising:
  (i) The pdf of the oldest state, $\mathbf{x}_{k+1}$ within the sliding window, approximated as a Gaussian, $\mathcal{N}(\hat{\mathbf{x}}^{\ominus}_{k+1}, \mathbf{P}^{\ominus}_{k+1})$.
  (ii) A set of N_L linearized constraints relating the oldest state $\mathbf{x}^\star_{k+1}$ with the rest of the poses within the sliding window, expressed as:
  $$\mathbf{r}^{\star\ominus}_L = \mathbf{H}^{\star\ominus}_L \tilde{\mathbf{x}}^\star_{k+1:k+M-1} + \mathbf{n}_L, \ \mathbf{n}_L \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_{N_L}) \quad (10)$$
  The ensemble of the pdf $\mathcal{N}(\hat{\mathbf{x}}^{\ominus}_{k+1}, \mathbf{P}^{\ominus}_{k+1})$ and the linearized constraints $\{\mathbf{r}^{\star\ominus}_L, \ \mathbf{H}^{\star\ominus}_L\}$ in (10) represent all information regarding the states $\mathbf{x}_{k+1:k+M-1}$, accumulated from previous recursions of the algorithm, through absorption of past visual observations (i.e., $\mathbb{Z}_P$ in Fig. 1) and inertial measurements (i.e., $\{\mathbf{u}_{\ell,\ell+1}, \ \ell \leq k\}$).

B. IKS Algorithm: Overview

We hereafter describe a single recursion of the IKS, which involves the following steps:

1) **Propagation:** The prior pdf, $\mathcal{N}(\hat{\mathbf{x}}^{\ominus}_{k+1}, \mathbf{P}^{\ominus}_{k+1})$, for $\mathbf{x}_{k+1}$, and the inertial measurements $\{\mathbf{u}_{\ell,\ell+1}\}$, for $k+1 \leq \ell \leq k+M-1$, are used for creating a prior $\mathcal{N}(\hat{\mathbf{x}}^{\ominus}_{k+1:k+M}, \mathbf{P}^{\ominus}_{k+1:k+M})$ for *all* the states within the sliding window.
2) **State Update:** All available feature tracks, $\mathbb{Z}_M$ *and* $\mathbb{Z}_I$, as well as the prior constraint $\{\mathbf{r}^{\star\ominus}_L, \ \mathbf{H}^{\star\ominus}_L\}$, are processed for updating the current state estimates $\mathbf{x}^\star_{k+1:k+M}$. As it is described below, this state-optimization can be performed iteratively.
3) **Covariance Update:** The measurements, $\mathbb{Z}_M$ and $\mathbf{u}_{k+1,k+2}$, which are about to be absorbed, are used to compute the posterior covariance $\mathbf{P}^{\oplus}_{k+2}$ of $\mathbf{x}_{k+2}$, which will become the new "tail" of the sliding window.
4) **Prior Constraints Update:** The prior constraints $\{\mathbf{r}^{\star\ominus}_L, \ \mathbf{H}^{\star\ominus}_L\}$ are updated such that they are independent of the state to be marginalized, $\mathbf{x}_{k+1}$, and reflect the new constraints between $\mathbf{x}_{k+2}$ and $\mathbf{x}_{k+3:k+M}$.

Algorithm 1 IKS Recursion performing N_{max} iterations

```
Input:
- Linearization points $\mathbf{x}^\star_{k+1:k+M}$
- Prior pdf $\mathcal{N}(\hat{\mathbf{x}}^{\ominus}_{k+1}, \mathbf{P}^{\ominus}_{k+1})$
- Prior constraints $\{\mathbf{r}^{\star\ominus}_L, \mathbf{H}^{\star\ominus}_L\}$
- Inertial measurements $\mathbf{u}_{k+1:k+M}$
- Mature feature tracks $\mathbb{Z}_M$
- Immature feature tracks $\mathbb{Z}_I$.

Function:
  for $j \in [1, N_{max}]$ do
    %% Propagation:
    Generate prior $\mathcal{N}(\hat{\mathbf{x}}^{\ominus}_{k+1:k+M}, \mathbf{P}^{\ominus}_{k+1:k+M})$ using all
    inertial measurements $\mathbf{u}_{k+1:k+M}$ [see (13)-(16)]
    %% State Update:
    Update linearization points using:
    i) All visual observations (i.e., $\mathbb{Z}_M$ and $\mathbb{Z}_I$)
    ii) The prior constraints ($\{\mathbf{r}^{\star\ominus}_L, \mathbf{H}^{\star\ominus}_L\}$)
    [see (19), (20)]
    if $||\mathbf{x}^\star_{k+1:k+M} - \mathbf{x}^{\star\oplus}_{k+1:k+M}||_2 \leq \varepsilon$ then
      terminate iterations
    end if
    $\mathbf{x}^\star_{k+1:k+M} \leftarrow \mathbf{x}^{\star\oplus}_{k+1:k+M}$
  end for
  %% Covariance Update:
  Evaluate next epoch's $\mathcal{N}(\hat{\mathbf{x}}^{\oplus}_{k+2}, \mathbf{P}^{\oplus}_{k+2})$ using $\mathbf{u}_{k+1,k+2}$,
  $\{\mathbf{r}^{\star\ominus}_L, \mathbf{H}^{\star\ominus}_L\}$, and $\mathbb{Z}_M$ [see (27), (28)]
  %% Prior Constraints Update:
  Evaluate next epoch's linearized constraints
  $\{\mathbf{r}^{\star\oplus}_L, \ \mathbf{H}^{\star\oplus}_L\}$ using $\{\mathbf{r}^{\star\ominus}_L, \mathbf{H}^{\star\ominus}_L\}$, and $\mathbb{Z}_M$ [see (34)]
```

C. IKS Algorithm: Detailed Description

In order to allow for a direct comparison with the INVF and the EKF (see [17]), 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:

$$c_{k+M}(\tilde{\mathbf{x}}^\star_{k+1:k+M}) = c_{P^{\ominus}_{k+1}}(\tilde{\mathbf{x}}^\star_{k+1}) + c_u(\tilde{\mathbf{x}}^\star_{k+1:k+M}) \quad (11)$$
$$+ c_L(\tilde{\mathbf{x}}^\star_{k+1:k+M-1}) + c_{Z_M}(\tilde{\mathbf{x}}^\star_{k+1:k+M}) + c_{Z_I}(\tilde{\mathbf{x}}^\star_{k+2:k+M})$$

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 (4)], c_L to prior information about the poses $\mathbf{x}_{k+1:k+M-1}$ [see (10)], c_{Z_M} and c_{Z_I} to geometric constraints between the states from the visual observations [see (8)] corresponding to the two sets of available feature tracks, $\mathbb{Z}_M$ and $\mathbb{Z}_I$, respectively. Hereafter, we employ the cost terms in (11) to describe the four main steps of the proposed IKS (see Sec. III-B).

1) Prior Propagation: The prior pdf $\mathcal{N}(\hat{\mathbf{x}}^{\ominus}_{k+1}, \mathbf{P}^{\ominus}_{k+1})$ and the inertial measurements $\mathbf{u}_{k+1:k+M}$ are used to generate a prior pdf $\mathcal{N}(\hat{\mathbf{x}}^{\ominus}_{k+1:k+M}, \mathbf{P}^{\ominus}_{k+1:k+M})$ over *all* the states, $\mathbf{x}_{k+1:k+M}$, within the sliding window. Through this process, the cost function in (11) takes the form:

$$c_{k+M}(\tilde{\mathbf{x}}^{\star}_{k+1:k+M}) = c_{P^{\ominus}_{k+1:k+M}}(\tilde{\mathbf{x}}^{\star}_{k+1:k+M}) + c_L(\tilde{\mathbf{x}}^{\star}_{k+1:k+M-1}) + c_{Z_M}(\tilde{\mathbf{x}}^{\star}_{k+1:k+M}) + c_{Z_I}(\tilde{\mathbf{x}}^{\star}_{k+2:k+M}) \tag{12}$$

where,

$$c_{P^{\ominus}_{k+1:k+M}}(\tilde{\mathbf{x}}^{\star}_{k+1:k+M}) = c_{P^{\ominus}_{k+1}}(\tilde{\mathbf{x}}^{\star}_{k+1}) + \sum_{\ell=k+1}^{k+M-1} c_{u_{\ell,\ell+1}}(\tilde{\mathbf{x}}^{\star}_{\ell:\ell+1})$$

corresponds to the prior $\mathcal{N}(\hat{\mathbf{x}}^{\ominus}_{k+1:k+M}, \mathbf{P}^{\ominus}_{k+1:k+M})$.

The mean $\hat{\mathbf{x}}^{\ominus}_{k+1:k+M}$ is computed using the standard equations of the EKF, i.e.,

$$\hat{\mathbf{x}}^{\ominus}_{k+i} = \begin{cases} \hat{\mathbf{x}}^{\ominus}_{k+1} & ,i=1 \\ \mathbf{f}(\hat{\mathbf{x}}^{\ominus}_{k+i-1}, \mathbf{u}_{k+i-1,k+i}) & ,2 \leq i \leq M \end{cases} \tag{13}$$

or,

$$\hat{\mathbf{x}}^{\ominus}_{k+i} = \begin{cases} \hat{\mathbf{x}}^{\ominus}_{k+1} & ,i=1 \\ \mathbf{f}(\mathbf{x}^{\star}_{k+i-1}, \mathbf{u}_{k+i-1,k+i}) + \mathbf{\Phi}^{\star}_{k+i,k+i-1}\delta\mathbf{x}^{\ominus}_{k+i-1} & ,2 \leq i \leq M \end{cases} \tag{14}$$

with $\delta\mathbf{x}^{\ominus}_{k+i-1} = \hat{\mathbf{x}}^{\ominus}_{k+i-1} - \mathbf{x}^{\star}_{k+i-1}$, when re-linearization is involved. Note also that, once more measurements are processed, the mean $\hat{\mathbf{x}}^{\ominus}_{k+i}$ of the prior pdf maintained by the IKS, will in general, be different from the first linearization point, $\mathbf{x}^{\star}_{k+1}$ used by the EKF in (4). Thus, in contrast to the EKF, the IKS can compensate for past linearization errors by re-computing the prior, over the sliding window $\mathbf{x}_{k+1:k+M}$, through the process described by (14).

Finally, the block diagonal-elements of the covariance $\mathbf{P}^{\ominus}_{k+1:k+M}$ are computed through the EKF covariance propagation recursion:

$$\mathbf{P}^{\ominus}_{k+i} = \mathbf{\Phi}^{\star}_{k+i,k+i-1}\mathbf{P}^{\ominus}_{k+i-1}\mathbf{\Phi}^{\star T}_{k+i,k+i-1} + \mathbf{Q}^{\star}_{k},\ i = 2\ldots,M \tag{15}$$

while the off-diagonal block elements are computed as:

$$\mathbf{P}^{\ominus}_{k+i,k+j} = \mathbf{\Phi}^{\star}_{k+i,k+j}\mathbf{P}^{\ominus}_{k+j},\ i = 2,\ldots,M,\ j = 1,\ldots,i-1 \tag{16}$$

2) State Update: All cost terms, c_L, c_{Z_M}, and c_{Z_I}, that provide linearized constraints are used for updating the state estimates for $\mathbf{x}_{k+1:k+M}$. Although each of these terms could have been used independently in three consecutive updates, we choose to first merge them and then use them in a batch form. Thus, (12) becomes:

$$c_{k+M}(\tilde{\mathbf{x}}^{\star}_{k+1:k+M}) = c_{P^{\ominus}_{k+1:k+M}}(\tilde{\mathbf{x}}^{\star}_{k+1:k+M}) + c_S(\tilde{\mathbf{x}}^{\star}_{k+1:k+M}) \tag{17}$$

where[3]

$$\begin{aligned} c_S(\tilde{\mathbf{x}}^{\star}_{k+1:k+M}) &= || \begin{bmatrix} \mathbf{r}^{\star\ominus}_L \\ \mathbf{r}^{\star}_M \\ \mathbf{r}^{\star}_I \end{bmatrix} - \begin{bmatrix} \mathbf{H}^{\star\ominus}_L \\ \mathbf{H}^{\star}_M \\ \mathbf{H}^{\star}_I \end{bmatrix} \tilde{\mathbf{x}}^{\star}_{k+1:k+M} ||^2_{\sigma^2\mathbf{I}} \\ &= ||\mathbf{r}^{\star}_s - \mathbf{H}^{\star}_s\tilde{\mathbf{x}}^{\star}_{k+1:k+M}||^2_{\sigma^2\mathbf{I}}. \end{aligned} \tag{18}$$

Minimizing (17) can be done in a numerically stable manner, following an I-EKF update step [7]:

$$\mathbf{x}^{\star\oplus}_{k+1:k+M} = \hat{\mathbf{x}}^{\ominus}_{k+1:k+M} + \mathbf{P}^{\ominus}_{k+1:k+M}\mathbf{H}^{\star T}_s\mathbf{d}_s \tag{19}$$

where $\mathbf{d}_s$ is the solution to the linear system

$$\mathbf{S}^{\star}\mathbf{d}_s = \mathbf{r}^{\star}_s - \mathbf{H}^{\star}_s\delta\mathbf{x}^{\ominus}_{k+1:k+M} \tag{20}$$

with $\mathbf{S}^{\star} = \mathbf{H}^{\star}_s\mathbf{P}^{\ominus}_{k+1:k+M}\mathbf{H}^{\star T}_s + \sigma^2\mathbf{I}$ and $\delta\mathbf{x}^{\ominus}_{k+1:k+M} = \hat{\mathbf{x}}^{\ominus}_{k+1:k+M} - \mathbf{x}^{\star}_{k+1:k+M}$.

Contrary to INVF-based approaches that involve an ill-conditioned Hessian, solving (20) is a numerically stable process, due to the low (i.e., $1-200$) condition number of $\mathbf{S}^{\star}$. Furthermore, compared to the EKF that only processes the measurements $\mathbb{Z}_M$ corresponding to mature features (see Fig. 1), the IKS can process not only $\mathbb{Z}_M$ but also $\mathbb{Z}_I$ as soon as they arrive. This way the state estimates immediately receive the gain in accuracy from recent feature tracks, while consistency is ensured by using features only *once* for updating the covariance.

Steps 1 and 2 (prior propagation and state update) can be repeated iteratively, as described in Alg. 1, to compute the state estimates $\mathbf{x}^{\star\oplus}_{k+1:k+M}$.

3) Covariance Update: Our objective is to compute the posterior $\mathcal{N}(\hat{\mathbf{x}}^{\oplus}_{k+2}, \mathbf{P}^{\oplus}_{k+2})$, 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 (11) that contain the state $\mathbf{x}_{k+1}$, which is about to be marginalized; that is the cost function:

$$c^M_{k+M}(\tilde{\mathbf{x}}^{\star}_{k+1:k+M}) = c_{P^{\ominus}_{k+1}}(\tilde{\mathbf{x}}^{\star}_{k+1}) + c_{u_{k+1,k+2}}(\tilde{\mathbf{x}}^{\star}_{k+1:k+2}) + c_L(\tilde{\mathbf{x}}^{\star}_{k+1:k+M-1}) + c_{Z_M}(\tilde{\mathbf{x}}^{\star}_{k+1:k+M}) \tag{21}$$

In particular, we follow a 4-step process:

a) Prior Propagation: Following the same procedure as in Sec. III-C.1, we use the prior $\mathcal{N}(\hat{\mathbf{x}}^{\ominus}_{k+1}, \mathbf{P}^{\ominus}_{k+1})$ and the inertial measurements $\mathbf{u}_{k+1:k+2}$ to compute the prior $\mathcal{N}(\hat{\mathbf{x}}^{\ominus}_{k+1:k+2}, \mathbf{P}^{\ominus}_{k+1:k+2})$. Thus, (21) becomes:

$$c^M_{k+M}(\tilde{\mathbf{x}}^{\star}_{k+1:k+M}) = c_{P^{\ominus}_{k+1:k+2}}(\tilde{\mathbf{x}}^{\star}_{k+1:k+2}) + c_L(\tilde{\mathbf{x}}^{\star}_{k+1:k+M-1}) + c_{Z_M}(\tilde{\mathbf{x}}^{\star}_{k+1:k+M}). \tag{22}$$

b) Measurement Compression: Following the same process as in (18), we merge the linearized constraints into one, i.e.,

$$c^M_{k+M}(\tilde{\mathbf{x}}^{\star}_{k+1:k+M}) = c_{P^{\ominus}_{k+1:k+2}}(\tilde{\mathbf{x}}^{\star}_{k+1:k+2}) + \bar{c}_C(\tilde{\mathbf{x}}^{\star}_{k+1:k+M}). \tag{23}$$

c) Partitioning of the linearized constraints: Following the same process as in (8), we partition $\bar{c}_C$ into $\bar{c}_{C_1}$ and $\bar{c}_{C_2}$, where the first term depends only on $\tilde{\mathbf{x}}^{\star}_{k+1:k+2}$:

$$\begin{aligned} \bar{c}_C(\tilde{\mathbf{x}}^{\star}_{k+1:k+M}) &= ||\bar{\mathbf{r}}^{\star}_C - \bar{\mathbf{H}}^{\star}_C\tilde{\mathbf{x}}^{\star}_{k+1:k+M}||^2_{\sigma^2\mathbf{I}} \\ &= ||\bar{\mathbf{r}}^{\star}_C - \begin{bmatrix} \bar{\mathbf{H}}^{\star}_{C_1} & \bar{\mathbf{H}}^{\star}_{C_2} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}^{\star}_{k+1:k+2} \\ \tilde{\mathbf{x}}^{\star}_{k+3:k+M} \end{bmatrix} ||^2_{\sigma^2\mathbf{I}} \\ &= ||\bar{\mathbf{r}}^{\star'}_{C_1} - \bar{\mathbf{H}}^{\star'}_{C_1}\tilde{\mathbf{x}}^{\star}_{k+1:k+2}||^2_{\sigma^2\mathbf{I}_{N_1}} + ||\bar{\mathbf{r}}^{\star'}_{C_2} - \bar{\mathbf{H}}^{\star'}_{C_2}\tilde{\mathbf{x}}^{\star}_{k+1:k+M}||^2_{\sigma^2\mathbf{I}_{N_2}} \\ &= \bar{c}_{C_1}(\tilde{\mathbf{x}}^{\star}_{k+1:k+2}) + \bar{c}_{C_2}(\tilde{\mathbf{x}}^{\star}_{k+1:k+M}) \end{aligned} \tag{24}$$

with $\bar{\mathbf{H}}^{\star'}_{C_1} = \mathbf{U}^T_{C_2}\bar{\mathbf{H}}^{\star}_{C_1}$, $\bar{\mathbf{r}}^{\star'}_{C_1} = \mathbf{U}^T_{C_2}\bar{\mathbf{r}}^{\star}_C$ and $\bar{\mathbf{H}}^{\star'}_{C_2} = \mathbf{S}^T_{C_2}\bar{\mathbf{H}}^{\star}_C$, $\bar{\mathbf{r}}^{\star'}_{C_1} = \mathbf{S}^T_{C_2}\bar{\mathbf{r}}^{\star}_C$, where the columns of $\mathbf{U}_{C_2}$ and $\mathbf{S}_{C_2}$ span the left null space and column space of $\bar{\mathbf{H}}^{\star}_{C_2}$, respectively. Substituting (24) in (23), yields:

$$c^M_{k+M}(\tilde{\mathbf{x}}^{\star}_{k+1:k+M}) = c_{P^{\ominus}_{k+1:k+2}}(\tilde{\mathbf{x}}^{\star}_{k+1:k+2}) + \bar{c}_{C_1}(\tilde{\mathbf{x}}^{\star}_{k+1:k+2}) + \bar{c}_{C_2}(\tilde{\mathbf{x}}^{\star}_{k+1:k+M}). \tag{25}$$

d) Covariance Update: At this point, we have brought the cost function c^M_{k+M} in a form that allows combining the first two terms in (25), thus updating the *prior pdf*

[3] For reducing the computational cost (linear in the number of features within the sliding window), the residual $\mathbf{r}^{\star}_s$ and Jacobian matrix $\mathbf{H}^{\star}_s$ are compressed using QR factorization as described in [1].

$\mathcal{N}(\mathbf{x}^{\ominus}_{k+1:k+2}, \mathbf{P}^{\ominus}_{k+1:k+2})$:

$$c^{M}_{k+M}(\tilde{\mathbf{x}}^{\star}_{k+1:k+M}) = c_{P^{\oplus}_{k+1:k+2}}(\tilde{\mathbf{x}}^{\star}_{k+1:k+2}) + \bar{c}_{C_2}(\tilde{\mathbf{x}}^{\star}_{k+1:k+M}). \quad (26)$$

The mean and covariance of $\mathcal{N}(\mathbf{x}^{\oplus}_{k+1:k+2}, \mathbf{P}^{\oplus}_{k+1:k+2})$, are:

$$\hat{\mathbf{x}}^{\oplus}_{k+1:k+2} = \hat{\mathbf{x}}^{\ominus}_{k+1:k+2} + \mathbf{P}^{\ominus}_{k+1:k+2}\bar{\mathbf{H}}^{\star'T}_{C_1}\mathbf{d}_C \quad (27)$$

where $\mathbf{d}_C$ is the solution to the linear system $\mathbf{S}_C\mathbf{d}_C = \mathbf{r}^{\star'}_{C_1} - \bar{\mathbf{H}}^{\star'T}_{C_1}\delta\mathbf{x}^{\ominus}_{k+1:k+2}$, with $\mathbf{S}_C = \bar{\mathbf{H}}^{\star'}_{C_1}\mathbf{P}^{\ominus}_{k+1:k+2}\bar{\mathbf{H}}^{\star'T}_{C_1} + \sigma^2\mathbf{I}_{N_1}$ and

$$\mathbf{P}^{\oplus}_{k+1:k+2} = \mathbf{P}^{\ominus}_{k+1:k+2} - \mathbf{P}^{\ominus}_{k+1:k+2}\bar{\mathbf{H}}^{\star'T}_{C_1}\mathbf{S}^{-1}_{C}\bar{\mathbf{H}}^{\star'}_{C_1}\mathbf{P}^{\ominus}_{k+1:k+2}. \quad (28)$$

4) Constructing the next epoch's prior: The last step of the IKS requires marginalizing $\mathbf{x}_{k+1}$, i.e., bringing c^{M}_{k+M} into such a form, that its minimization is independent of $\mathbf{x}_{k+1}$. To achieve this, we follow a 2-step process.

a) Partitioning of $c_{P^{\oplus}_{k+1:k+2}}$: By employing the Schur complement, the prior term $c_{P^{\oplus}_{k+1:k+2}}$, in (26) is partitioned into a prior over $\mathbf{x}_{k+2}$, $c_{P^{\oplus}_{k+2}}$, and a conditional term, $c_{k+1|k+2}$, representing linearized constraints between $\mathbf{x}_{k+1}$ and $\mathbf{x}_{k+2}$:

$$\begin{aligned} c_{P^{\oplus}_{k+1:k+2}}(\tilde{\mathbf{x}}^{\star}_{k+1:k+2}) &= || \begin{bmatrix} \tilde{\mathbf{x}}^{\star}_{k+1} - \delta\mathbf{x}^{\oplus}_{k+1} \\ \tilde{\mathbf{x}}^{\star}_{k+2} - \delta\mathbf{x}^{\oplus}_{k+2} \end{bmatrix} ||^2_{\mathbf{P}^{\oplus}_{k+1:k+2}} \\ =&||\tilde{\mathbf{x}}^{\star}_{k+2} - \delta\mathbf{x}^{\oplus}_{k+2}||_{\mathbf{P}^{\oplus}_{k+2}} \\ &+ ||\delta\mathbf{x}^{\oplus}_{k+1|k+2} - \begin{bmatrix} \mathbf{I} & -\mathbf{P}^{\oplus}_{k+1,k+2}\mathbf{P}^{\oplus-1}_{k+2} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}^{\star}_{k+1} \\ \tilde{\mathbf{x}}^{\star}_{k+2} \end{bmatrix} ||^2_{\mathbf{P}^{\oplus}_{k+1,k+2}} \\ =&c_{P^{\oplus}_{k+2}}(\tilde{\mathbf{x}}_{k+2}) + c_{k+1|k+2}(\tilde{\mathbf{x}}_{k+1:k+2}) \end{aligned} \quad (29)$$

where $\mathbf{P}^{\oplus}_{k+1|k+2} = \mathbf{P}^{\oplus}_{k+1} - \mathbf{P}^{\oplus}_{k+1,k+2}\mathbf{P}^{\oplus-1}_{k+2}\mathbf{P}^{\oplus}_{k+2,k+1}$.

Substituting (29) in (26), yields:

$$\begin{aligned} c^{M}_{k+M}(\tilde{\mathbf{x}}^{\star}_{k+1:k+M}) =& c_{P^{\oplus}_{k+2}}(\tilde{\mathbf{x}}^{\star}_{k+2}) + c_{k+1|k+2}(\tilde{\mathbf{x}}^{\star}_{k+1:k+2}) \\ &+ \bar{c}_{C_2}(\tilde{\mathbf{x}}^{\star}_{k+1:k+M}). \end{aligned} \quad (30)$$

b) Marginalization of $\mathbf{x}_{k+1}$: Firstly, we combine all terms involving $\mathbf{x}_{k+1}$, i.e., $c_{k+1|k+2}$ and $\bar{c}_{C_2}$, into a single quadratic cost:

$$\begin{aligned} c_J(\tilde{\mathbf{x}}_{k+1:k+M}) &= c_{k+1|k+2}(\tilde{\mathbf{x}}_{k+1:k+2}) + \bar{c}_{C_2}(\tilde{\mathbf{x}}^{\star}_{k+1:k+M}) \\ &= ||\mathbf{b} - \mathbf{J}\tilde{\mathbf{x}}^{\star}_{k+1:k+M}||^2_{\mathbf{I}_{N_2}} \end{aligned} \quad (31)$$

corresponding to $15+N_2$ linearized constraints:[4]

$$\mathbf{b} = \begin{bmatrix} \mathbf{J}_1 & \mathbf{J}_2 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}^{\star}_{k+1} \\ \tilde{\mathbf{x}}^{\star}_{k+2:k+M} \end{bmatrix} + \mathbf{v}, \ \mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{15+N_2}). \quad (32)$$

Next, following the same process as in (8), we project (32) onto the column space, spanned by $\mathbf{S}_{J_1}$, and left null space, spanned by $\mathbf{U}_{J_1}$, of $\mathbf{J}_1$, to split c_J into $c_{k+1|k+2:k+M}$ and $c_{L^{\oplus}}$, respectively,

$$\begin{aligned} & c_J(\tilde{\mathbf{x}}^{\star}_{k+1:k+M}) \\ &= ||\mathbf{b}'_1 - \mathbf{J}'_1\tilde{\mathbf{x}}^{\star}_{k+1:k+M}||^2_{\mathbf{I}_{15}} + ||\mathbf{r}^{\star\oplus}_{L} - \mathbf{H}^{\star\oplus}_{L}\tilde{\mathbf{x}}^{\star}_{k+2:k+M}||^2_{\mathbf{I}_{N_2}} \\ &= c_{k+1|k+2:k+M}(\tilde{\mathbf{x}}^{\star}_{k+1:k+M}) + c_{L^{\oplus}}(\tilde{\mathbf{x}}^{\star}_{k+2:k+M}) \end{aligned} \quad (33)$$

where $\mathbf{J}'_1 = \mathbf{S}^{T}_{J_1}\mathbf{J}$, $\mathbf{b}'_1 = \mathbf{S}^{T}_{J_1}\mathbf{b}$, and $\mathbf{H}^{\star\oplus}_{L} = \mathbf{U}^{T}_{J_1}\mathbf{J}_2$, $\mathbf{r}^{\star\oplus}_{L} = \mathbf{U}^{T}_{J_1}\mathbf{b}$. Furthermore, for the latter terms that define $c_{L^{\oplus}}$, as it is shown in [17], there exists an analytical form:

$$\mathbf{H}^{\star\oplus}_{L} = \mathbf{L}^{-T}\begin{bmatrix} \bar{\mathbf{H}}^{\star'}_{C_{22}} + \bar{\mathbf{H}}^{\star'}_{C_{21}}\mathbf{P}^{\oplus}_{k+1,k+2}\mathbf{P}^{\oplus-1}_{k+2} & \bar{\mathbf{H}}^{\star'}_{C_{23}} \end{bmatrix} \quad (34)$$

$$\mathbf{r}^{\star\oplus}_{L} = \mathbf{L}^{-T}\left(\bar{\mathbf{r}}^{\star'}_{C_2} - \bar{\mathbf{H}}^{\star'}_{C_{21}}\left(\delta\mathbf{x}^{\oplus}_{k+1} + \mathbf{P}^{\oplus}_{k+1,k+2}\mathbf{P}^{\oplus-1}_{k+2}\delta\mathbf{x}^{\oplus}_{k+2}\right)\right)$$

where $\mathbf{L}$ is the Cholesky factor of $\mathbf{R} = \frac{1}{\sigma^2}\bar{\mathbf{H}}^{\star'}_{C_{21}}\mathbf{P}^{\oplus}_{k+1|k+2}\bar{\mathbf{H}}^{\star'T}_{C_{21}} + \mathbf{I}_{N_2}$, while $\bar{\mathbf{H}}_{C_{21}}$, $\bar{\mathbf{H}}_{C_{22}}$, and $\bar{\mathbf{H}}_{C_{23}}$ denote the columns of $\bar{\mathbf{H}}_{C_2}$ in (24), corresponding to $\tilde{\mathbf{x}}^{\star}_{k+1}$, $\tilde{\mathbf{x}}^{\star}_{k+2}$, and $\tilde{\mathbf{x}}^{\star}_{k+3:k+M}$, respectively.

Substituting (31) and (33) in (30), yields:

$$\begin{aligned} c^{M}_{k+M}(\tilde{\mathbf{x}}^{\star}_{k+1:k+M}) =& c_{P^{\oplus}_{k+2}}(\tilde{\mathbf{x}}^{\star}_{k+2}) + c_J(\tilde{\mathbf{x}}^{\star}_{k+1:k+M}) \\ =& c_{P^{\oplus}_{k+2}}(\tilde{\mathbf{x}}^{\star}_{k+2}) + c_{L^{\oplus}}(\tilde{\mathbf{x}}^{\star}_{k+2:k+M}) \\ &+ c_{k+1|k+2:k+M}(\tilde{\mathbf{x}}^{\star}_{k+1:k+M}). \end{aligned} \quad (35)$$

The last term, $c_{k+1|k+2:k+M}$ in (35), is irrelevant for the minimization of c^{M}_{k+M} over $\tilde{\mathbf{x}}^{\star}_{k+2:k+M}$, since for any $\tilde{\mathbf{x}}^{\star}_{k+2:k+M}$, there exists a $\tilde{\mathbf{x}}^{o}_{k+1}$, that minimizes $c_{k+1|k+2:k+M}$ to *exactly* zero. Hence, all prior information from the current to the next IKS recursion, is represented completely through the terms $c_{P^{\oplus}_{k+2}}$ and $c_{L^{\oplus}}$, both of which do *not* involve $\tilde{\mathbf{x}}^{\star}_{k+1}$.

IV. 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 $\sigma = 1.5$ pixel. The platform's trajectory and dynamics resembled those of a person traversing 144 m along four corridors of an indoor environment, while new camera poses were generated every 25 cm [see Fig. 2(a)].

A. Localization Accuracy

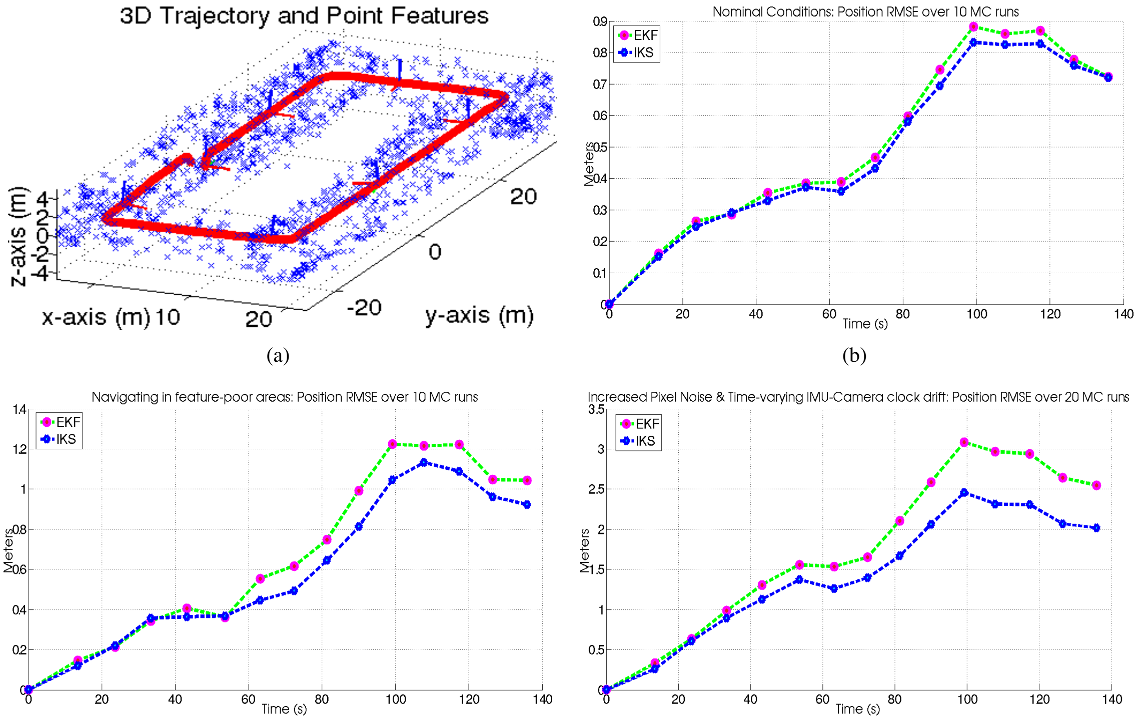
We compared the position Root Mean Square Error (RMSE) of a well-established EKF-based VINS algorithm, the MSC-KF [2] (denoted as EKF), with that of the proposed Iterative Kalman Smoother (denoted as IKS), over 10 Monte Carlo trials, for different scenarios. The simulated estimators maintained a sliding window of 10 past camera poses.

1) Nominal conditions: In this scenario, the rate that new features enter the camera's field of view resembles that of real-world experimental trials, which corresponds to approximately 100 new features tracks per second. As seen in Fig. 2(b), the performance difference between the EKF-based approach and the proposed estimator is rather small, since in the presence of many visual measurements, both estimators are able to accurately track the system's state.

2) Challenging conditions: Vision-aided inertial navigation systems often encounter challenging conditions, such as scarcity of visual observations, or time-varying parameters, which can significantly increase the importance of relinearization.

Feature-poor areas: We simulated a case where visual observations are scarce (e.g., going down texture-less corridors). Specifically, the birth rate of feature tracks was reduced to approximately 20 new feature tracks per second.

[4] The term $c_{k+1|k+2}$ contributes 15 equations (i.e., the dimension of $\tilde{\mathbf{x}}^{\star}_{k+1}$) while $\bar{c}_{C_2}$, N_2 constraints [see (24)].



(a)

(b)

(c)

(d)

Fig. 2. Monte Carlo simulations and comparisons of the proposed IKS versus the EKF: (a) The simulated trajectory of the camera-IMU and feature positions; (b) Position RMSE under nominal conditions; (c) Position RMSE while operating within a featureless area; (d) Position RMSE under large pixel noise and a time-varying drift between the camera and IMU clocks.

As seen in Fig. 2(c), during periods with few visual observations, the accuracy difference between the EKF and IKS increases, due to the ability of the latter to re-linearize visual obsevations.

Time-varying IMU-camera clocks drift: In this scenario, we simulated a time-varying drift between the camera and IMU clocks, similar to that encountered on mobile devices. An interpolation-based approach was employed for modelling the time synchronization [12], while, the σ of the camera noise was increased to 3 pixels, resembling the effect of feature tracking on scaled (down to 320×240, due to processing limitations) images. Finally the camera's field of view was reduced to 45^o, representing the imaging capabilities of commercially available devices. As seen in Fig. 2(d), under the presence of time-varying parameters, such as the drift between the clocks of the two sensors, the IKS significantly outperforms the EKF.

B. Numerical Stability

TABLE I

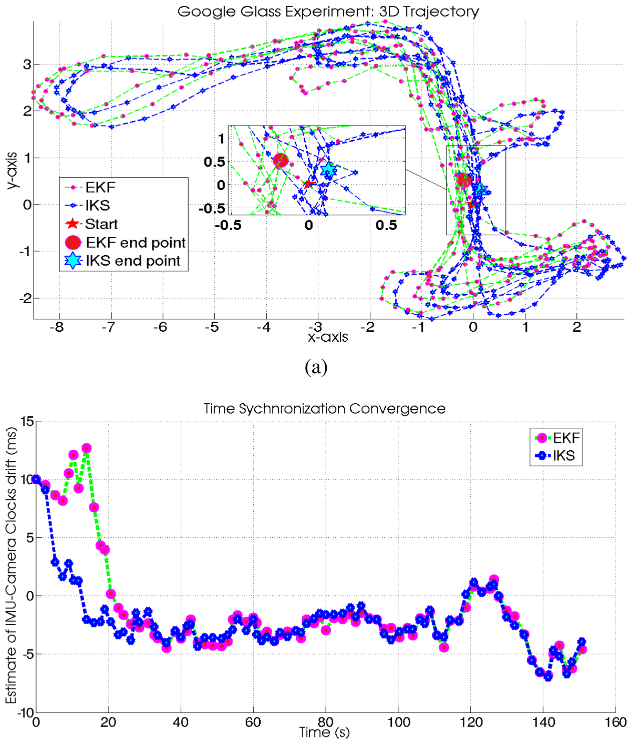
WORST CONDITION NUMBER ENCOUNTERED AT EACH RECURSION

Estimation Algorithm	Condition Number
EKF [1]	10^1
IKS (Proposed)	10^2
INVF [5]	10^9

We compared the numerical stability of three estimators applied to VINS. The EKF-based approach of [1], the sliding window Inverse Filter (INVF) of [5] and the proposed IKS, using the worst condition number that was encountered during any matrix inversion (or solution of a linear system) across each estimator's recursions. As seen in Table I, our findings confirm that the proposed estimator inherits the excellent numerical properties of Kalman filters, while the INVF requires the solution of an ill-conditioned (for floating-point precision) linear system.

V. EXPERIMENTS

We further validated the proposed IKS on real-world data, using a Google Glass wearable computer [11], which features a dual-core OMAP 4430 CPU operating at 1 GHz with 682 MB of RAM, and is equipped with an Invensense 9150 IMU and a 640×480 narrow (45^o) field of view camera. Due to computing limitations, both image processing and estimation algorithms operated on scaled (down to 320×240) images and maintained a sliding window of length 8. During the experiment, the user traversed multiple times an office space of dimensions 10×5 m, over a total distance of 127 m, returning back to the device's starting position, so as to quantitatively evaluate the accuracy of the estimators considered [see Fig. 3(a)]. As seen in Table II, the proposed algorithm provided a performance improvement of 58%, resulting in a final positioning error of 0.32 m (as opposed to 0.55 m for the EKF) corresponding to 0.25% of the total distance travelled. At this point, it is important to note, that due to the low processing and sensing capabilities of Google Glass, the images were scaled down and sampled at a relatively low frequency (i.e., 15 Hz). Despite these limitations, the achieved accuracy was comparable to that achieved on mobile devices (i.e., cell-phones) using the same modelling assumptions [12]. Furthermore, we should note that the presence of a time-varying parameter, in this case the IMU-camera clock drift, which affects the accuracy of the Jacobians employed by the estimator, can significantly reduce the performance of estimators that do not allow re-linearization, such as the EKF. Specifically, as seen in Fig. 3(b), the proposed IKS estimator converges faster to the correct time synchronization value, as compared to the EKF, hence reducing the initial accumulation of error along unobservable directions (position and yaw).



(a)

(b)

Fig. 3. Experimental results: (a) 3D trajectory end estimated end-points. (b) Convergence of the time-synchronization parameter for both estimators.

TABLE II

GOOGLE GLASS EXPERIMENT: LOOP-CLOSURE ERROR

Estimation Algorithm	Loop Closure Error
EKF	0.43% (0.55/127 m)
IKS	0.25% (0.32/127 m)

VI. CONCLUSIONS

In this work, we have presented an Iterative Kalman Smoother (IKS) for performing vision-aided inertial navigation that shares the two competing advantages of alternative approaches. On one hand, and similar to inverse-filter-based estimators, the proposed IKS can iteratively re-linearize both inertial and visual measurements over a single, or multiple overlapping, sliding windows, hence improving robustness to linearization errors and increasing positioning accuracy. On the other hand, the IKS enjoys the numerical stability of extended Kalman filter (EKF)-based estimators making it amenable to very efficient implementations (4-fold speed gain on ARM NEON co-processor) using single-precision (32 bit) arithmetic. Finally, its adjustable computational cost (based on the window size, the number of features processed, and the number of iterations employed) makes the proposed IKS an appealing solution for online localization on severely resource-constrained devices, such as wearables. As part of our validation process, we demonstrated the high accuracy of the proposed algorithm over extensive simulations and real-world experiments on the Google Glass, and achieved precision comparable to that of EKF-based estimators operating on cell phones.

REFERENCES

[1] 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.

[2] 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.

[3] "Project Tango https://www.google.com/atap/projecttango," *Online*.

[4] 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.

[5] 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.

[6] 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.

[7] A. Jazwinski, *Stochastic processes and filtering theory*. No. 64 in Mathematics in science and engineering, New York, Academic Press, 1970.

[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] D. Tue-Cuong and A. I. Mourikis, "Motion tracking with Fixed-Lag Smoothing: Algorithm and consistency analysis," in *Proc. of the IEEE International Conference on Robotics and Automation*, (Shanghai, China), pp. 5655–5662, May 9 – 13 2011.

[10] 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.

[11] "Google Glass https://www.google.com/glass/start/," *Online*.

[12] 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.

[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] 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.

[15] 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.

[16] J. Y. Bouguet, "Camera Calibration Toolbox for Matlab," 2006. Available at `http://www.vision.caltech.edu/bouguetj/calibdoc/`, version 1.6.0.

[17] D. G. Kottas and S. I. Roumeliotis, "An Iterative Kalman Smoother for robust 3D localization on mobile and wearable devices, http://www-users.cs.umn.edu/˜dkottas," tech. rep., University of Minnesota, Dept. of Comp. Sci. & Eng., MARS Lab, Sept. 2014.