A Linear-Complexity EKF for Visual-Inertial Navigation with Loop Closures

Patrick Geneva*, Kevin Eckenhoff†, and Guoquan Huang†

***Abstract*— Enabling real-time visual-inertial navigation in unknown environments while achieving bounded-error performance holds great potentials in robotic applications. To this end, in this paper, we propose a novel linear-complexity EKF for visual-inertial localization, which can efficiently utilize loop closure constraints, thus allowing for long-term persistent navigation. The key idea is to adapt the Schmidt-Kalman formulation within the multi-state constraint Kalman filter (MSCKF) framework, in which we selectively include keyframes as nuisance parameters in the state vector for loop closures but do not update their estimates and covariance in order to save computations while still tracking their cross-correlations with the current navigation states. As a result, the proposed Schmidt-MSCKF has only $O(n)$ computational complexity while still incorporating loop closures into the system. The proposed approach is validated extensively on large-scale real-world experiments, showing significant performance improvements when compared to the standard MSCKF, while only incurring marginal computational overhead.**

I. Introduction

Visual-inertial navigation systems (VINS) have become one of the most promising low-cost 3D localization solutions, which fuse images from a monocular or stereo camera and IMU measurements [1–7]. This localization solution has the advantages of using sensors that are both cheap and ubiquitous and, because of the complementary nature of the sensors, has the potential to provide pose estimates which are on-par in terms of accuracy with more expensive sensors such as LiDAR. While current VINS approaches can perform well over a short period of time (e.g., see [2–4, 7, 8] and references therein), they are not robust and accurate enough for long-term and large-scale deployments, due to their limited available resources of sensing, memory and computation.

In this paper, we seek to overcome this issue by developing efficient VINS to incorporate loop closures, as loop closure constraints provide effective information to correct drift accumulated over time, thus permitting bounded localization errors even after long-term navigation in unknown environments [9–11]. In particular, instead of performing feature-based SLAM to build a large 3D map of the environment, we build upon the standard multi-state constraint Kalman filter (MSCKF) [1], and selectively keep the poses of keyframes (where loop closures take place) over the trajectory in the state vector, thereby limiting the growth of the state as a single keyframe may observe many features. By doing so, both feature measurements in the current window and loop closure feature observations can be exploited in the EKF update. However, as the length of the trajectory grows and keyframes are continuously added to the state, this naive extension of the MSCKF would quickly become computationally intractable for real-time performance due to the quadratic complexity of the EKF covariance update. For example, we empirically observed that the standard EKF could only retain on the upwards of 200 keyframes and remain real-time. To overcome this issue, we exploit the Schmidt-KF (SKF) [12] to treat the keyframe poses in the state vector as "nuisance" parameters and do not update their estimates and covariance while still properly tracking their cross-correlations with the rest of the navigation states. As a result, the proposed Schmidt-MSCKF has only $O(n)$ computational complexity with respect to the number of keyframes added, while effectively exploiting loop closure constraints.

II. Related Work

SLAM jointly estimates a robot's location and the structure of its surrounding environment and has attracted significant research efforts in the past three decades [13]. Over a trajectory, localization errors may grow unbounded unless some global information (e.g., GPS or *a priori* map) or constraints to previous locations (i.e., loop closures) are detected and used. Many methods leverage feature observations from different keyframes to limit drift over the trajectory [4, 14]. Most have a two-thread system that optimizes a small window of "local" keyframes and features limiting drift in the short-term, while a background process optimizes a long-term sparse pose graph containing loop closure constraints enforcing long-term consistency [15–19]. For example, VINS-Mono [18, 19] uses loop closure constraints in both the local sliding window and in the global batch optimization. Specifically, during the local optimization, feature observations from keyframes provide implicit loop closure constraints, while the problem size remains small by assuming the keyframe poses are perfect (thus removing them from optimization).

The inclusion of long-term loop closure information is challenging due to the inability to remain computationally efficient without making inconsistent assumptions such as treating keyframe poses to be true, or reusing information. Mourikis and Roumeliotis [20] introduced a hybrid estimator that used the MSCKF to perform real-time local estimation,

This work was partially supported by the University of Delaware College of Engineering, the NSF (IIS-1566129), the DTRA (HDTRA1-16-1-0039), and Google Daydream.

*P. Geneva is with the Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716, USA. Email: pgeneva@udel.edu

†K. Eckenhoff and G. Huang are with the Department of Mechanical Engineering, University of Delaware, Newark, DE 19716, USA. Email: {keck,ghuang}@udel.edu

and triggered global bundle adjustment (BA) on loop closure detection. This allowed for the relinearization and inclusion of loop closure constraints in a consistent manner, while requiring substantial additional overhead time where the filter waits for the bundle adjustment to finish. Lynen et al. [21] developed a large-scale map-based VINS that used a compressed prior map containing feature positions and their uncertainties and employed matches to features in the prior map to constrain the localization globally. Zheng et al. [22] developed a point-line visual-inertial odometry (VIO) system that treated the 3D positions of marginalized keypoints as true for future loop closure observations, which may lead to inconsistency. Julier [23] developed a 2D SKF that tracked the positions and covariances of landmarks while remaining computationally efficient through a sub-optimal update rule. Recently, DuToit et al. [24] introduced a Cholesky-Schmidt-Kalman filter, which, however, uses *a prior* map with its full uncertainty and relaxes all the correlations between the mapped features and the current state variables. As compared to these works, we actively estimate the correlations with the current state and only add keyframe poses, as compared to map points, limiting the computational growth.

III. Problem Statement

In this section, we describe the problem of visual-inertial localization with loop closures within the framework of the standard MSCKF [1], revealing its (at least) quadratic complexity not suitable for real-time performance.

A. MSCKF-based VIO

The state vector of the standard MSCKF-based VIO [1] contains the IMU navigation state $\mathbf{x}_I$ and a sliding window of cloned past IMU (or camera) poses $\mathbf{x}_C$, i.e.,

$$\mathbf{x}_k = \begin{bmatrix}\mathbf{x}_I^\top & \mathbf{x}_C^\top\end{bmatrix}^\top \tag{1}$$

$$\mathbf{x}_I = \begin{bmatrix}{}^{I_k}_{G}\bar{q}^\top & \mathbf{b}_{\omega_k}^\top & {}^G\mathbf{v}_{I_k}^\top & \mathbf{b}_{a_k}^\top & {}^G\mathbf{p}_{I_k}^\top\end{bmatrix}^\top \tag{2}$$

$$\mathbf{x}_C = \begin{bmatrix}{}^{I_{k-1}}_{G}\bar{q}^\top & {}^G\mathbf{p}_{I_{k-1}}^\top & \cdots & {}^{I_{k-m}}_{G}\bar{q}^\top & {}^G\mathbf{p}_{I_{k-m}}^\top\end{bmatrix}^\top \tag{3}$$

where ${}^I_G\bar{q}$ is the unit quaternion parameterizing the rotation ${}^I_G\mathbf{R}$ from the global frame of reference $\{G\}$ to the IMU local frame $\{I\}$ [25], $\mathbf{b}_\omega$ and $\mathbf{b}_a$ are the gyroscope and accelerometer biases, and ${}^G\mathbf{v}_I$ and ${}^G\mathbf{p}_I$ are the velocity and position of the IMU expressed in the global frame, respectively. The clone state $\mathbf{x}_C$ contains m historical IMU poses in a sliding window fashion.

1) Propagation: In the MSCKF, the state estimate and the corresponding covariance blocks are propagated over time by integrating the incoming IMU measurements of linear accelerations ($\mathbf{a}_m$) and angular velocities ($\boldsymbol{\omega}_m$) based on the following generic nonlinear IMU kinematics [26]:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{a}_{m_k} - \mathbf{n}_{a_k}, \boldsymbol{\omega}_{m_k} - \mathbf{n}_{\omega_k}) \tag{4}$$

where $\mathbf{n}_a$ and $\mathbf{n}_\omega$ are the zero-mean white Gaussian noise of the IMU measurements. We linearize this nonlinear model at the current estimate, and then propagate the state estimate and covariance matrix using the standard EKF [1].

2) Update: As the sensor platform moves through 3D space, environmental features are tracked on the image plane using KLT optical flow [27]. When a feature has been lost or reaches the sliding window size, we use the feature track information to perform an update. Specifically, we assume the following stacked nonlinear camera measurement model:

$$\mathbf{z}_f = \mathbf{h}(\mathbf{x}_k, {}^G\mathbf{p}_f) + \mathbf{n}_f \tag{5}$$

where $\mathbf{n}_f$ is the white Gaussian noise with covariance $\mathbf{R}_f$, and ${}^G\mathbf{p}_f$ is the 3D position of the feature. We linearize this measurement model at the current state estimate and obtain the following measurement residual:

$$\mathbf{r}_f = \mathbf{z}_f - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1}, {}^G\hat{\mathbf{p}}_f) \tag{6}$$

$$\simeq \mathbf{H}_x\tilde{\mathbf{x}}_{k|k-1} + \mathbf{H}_f{}^G\tilde{\mathbf{p}}_f + \mathbf{n}_f \tag{7}$$

where $\mathbf{H}_x$ and $\mathbf{H}_f$ are the measurement Jacobians with respect to the state and feature, and $\tilde{\mathbf{x}}$ and ${}^G\tilde{\mathbf{p}}_f$ are the errors for the state and feature, respectively. As the feature is not contained in the state vector, we marginalize ${}^G\mathbf{p}_f$ by projecting $\mathbf{r}_f$ onto the left nullspace of $\mathbf{H}_f$ (i.e., $\mathbf{N}^\top\mathbf{H}_f = \mathbf{0}$) to obtain the feature-independent measurement residual:

$$\mathbf{N}^\top\mathbf{r}_f = \mathbf{N}^\top\mathbf{H}_x\tilde{\mathbf{x}}_{k|k-1} + \mathbf{N}^\top\mathbf{H}_f{}^G\tilde{\mathbf{p}}_f + \mathbf{N}^\top\mathbf{n}_f \tag{8}$$

$$\Rightarrow\ \mathbf{r}'_f = \mathbf{H}'_x\tilde{\mathbf{x}}_{k|k-1} + \mathbf{n}'_f \tag{9}$$

where $\mathbf{n}'_f = \mathbf{N}^\top\mathbf{n}_f$ is white Guassian noise with covariance $\mathbf{R}'_f = \mathbf{N}^\top\mathbf{R}_f\mathbf{N}$, and can be used in the EKF update [1].

B. Incorporating Loop Closures

The standard MSCKF-based VIO [1] and its variants [2, 3, 7, 28] have been shown to be able to provide accurate localization solutions for relatively short periods of time and yet are not robust and accurate enough for long-term large-scale deployments, since these systems are essentially *open-loop* odometry whose navigation errors grow unbounded over time. In order to bound localization errors, we should incorporate loop closure constraints even in cases when an *a priori* map is unavailable. To this end, we may continuously include the keyframe poses where loop closure events can be detected, into the state vector:

$$\mathbf{x}_k = \begin{bmatrix}\mathbf{x}_I^\top & \mathbf{x}_C^\top & \mathbf{x}_{S_1}^\top & \cdots & \mathbf{x}_{S_n}^\top\end{bmatrix}^\top =: \begin{bmatrix}\mathbf{x}_I^\top & \mathbf{x}_C^\top & \mathbf{x}_S^\top\end{bmatrix}^\top \tag{10}$$

where $\mathbf{x}_{S_i} = \begin{bmatrix}{}^{I_i}_{G}\bar{q}^\top & {}^G\mathbf{p}_{I_i}^\top\end{bmatrix}^\top$ is the i-th keyframe pose. For example, if one naively keeps all the poses in the state vector, they essentially build the batch graph SLAM optimization and can incorporate all possible loop closures, which, however, causes the problem size to grow unbounded over time and is not amenable for real-time performance [29]. Even if we continuously include only the keyframe poses into the state vector within the MSCKF framework, performing an EKF update still yields *quadratic* cost in terms of the dimension of the state vector, $\dim(\mathbf{x}_k)$, which increases over time and may quickly become too computationally expensive.

IV. Schmidt-MSCKF VINS with Loop Closures

It is clear from the previous section that simply including keyframes in the MSCKF state vector in order to utilize loop closure information is not amenable for long-term real-time navigation. To address this issue, in this section, we present in detail the proposed novel consistent Schmidt-MSCKF of linear complexity while efficiently incorporating loop closures into our EKF-based VINS.

A. Overview

As discussed earlier, when looking to include loop closure constraints into VINS, one of the bottleneck issues is the computational complexity that arises from the storage and estimation of keyframe poses and/or keypoint features. As keyframes are added over the trajectory length, the size of states that need to be estimated would grow over time, threatening the real-time VINS performance, although it grows much slower than adding keypoint features into the state vector. One approach that prevents the need to estimate the keyframes at later times, is to simply assume that keyframe poses are "true" and ignore the uncertainty associated with these estimates, which would cause an overconfident (inconsistent) filter. In contrast, we leverage the SKF to allow for efficient estimation while still tracking the uncertainty of all keyframes in the states.

Specifically, we carefully retain by stochastic cloning [30] a set of keyframe poses where loop closures are likely to occur in the state vector (10) and consistently track their correlations with other state variables. We implicitly enforce loop closure constraints by adding additional observations from historical keyframe poses to actively-tracked features in the sliding window of the MSCKF. It is important to note that this does not require estimating the 3D feature position, since these observations are only a function of the poses in the sliding window and the historical keyframes once processed through the MSCKF update. This leads to substantial computational savings, as the number of keyframes over a trajectory is typically much smaller than the number of features seen from those frames. Moreover, we adapt the SKF [12] to treat old keyframe poses as "nuisance" parameters, which enables us to significantly limit the computational complexity incurred due to the increasing number of additional keyframes states. This is due to the fact that with its formulation, we do not update the estimates and marginal covariance of these states and instead only updated their cross-correlations with the active state variables in the sliding window. We thereby gain a significant reduction in computational complexity from quadratic to linear while remaining consistent.

B. Schmidt-MSCKF

The key idea of the proposed VINS with loop closures is to adapt the SKF within the MSCKF framework for real-time state estimation. Specifically, we first define the active state $\mathbf{x}_A := [\mathbf{x}_I^\top \ \mathbf{x}_C^\top]^\top$ as remaining of constant size over the trajectory due to sliding window marginalization, while the keyframe state $\mathbf{x}_S$ grows linearly as new keyframes are inserted [see (10)]. We accordingly partition the corresponding covariance matrix at time step k.

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_{A_k} \\ \mathbf{x}_{S_k} \end{bmatrix}, \quad \mathbf{P}_k = \begin{bmatrix} \mathbf{P}_{AA_k} & \mathbf{P}_{AS_k} \\ \mathbf{P}_{SA_k} & \mathbf{P}_{SS_k} \end{bmatrix} \tag{11}$$

1) Propagation: As in the standard MSCKF [1], the IMU state estimate is propagated via integration based on the IMU kinematics (4) (see Section III-A.1), while all other variables remain constant since they have zero dynamics.

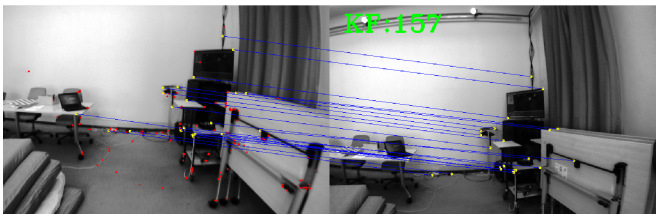

Fig. 1: An example of feature matches between the current frame (left) and the keyframe (right). Active feature tracks are seen in red on the current image and their matches to the keyframe are visualized with blue correspondence lines.

The covariance matrix is also propagated as follows:

$$\mathbf{P}_{k|k-1} = \tag{12}$$
$$\begin{bmatrix} \mathbf{\Phi}_{k-1}\mathbf{P}_{AA_{k-1|k-1}}\mathbf{\Phi}_{k-1}^\top & \mathbf{\Phi}_{k-1}\mathbf{P}_{AS_{k-1|k-1}} \\ \mathbf{P}_{SA_{k-1|k-1}}\mathbf{\Phi}_{k-1}^\top & \mathbf{P}_{SS_{k-1|k-1}} \end{bmatrix} + \begin{bmatrix} \mathbf{Q}_{k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

where $\mathbf{\Phi}_{k-1}$ and $\mathbf{Q}_{k-1}$ are respectively the system Jacobian and discrete noise covariance matrices for the active state. Note that the repeated computation of $\mathbf{\Phi}_{k-1}\mathbf{P}_{AS_{k-1|k-1}}$ is computationally expensive because of the high-rate IMU measurements (e.g., 200Hz) as compared to the low-rate camera images (e.g., 20Hz). To reduce this cost, we incrementally integrate the IMU measurements available between two image times (say m): $\mathbf{\Phi}_{k-1} = \mathbf{\Phi}_{k-1,m} \cdots \mathbf{\Phi}_{k-1,0}$, with the initial condition $\mathbf{\Phi}_{k-1,0} = \mathbf{I}$. Similarly, the noise covariance $\mathbf{Q}_{k-1}$ is recursively compounded.

2) Visual tracking and loop closing: As the IMU-camera sensor platform navigates in the environment, the sliding window sequentially shifts forward as a new image arrives. As in the standard MSCKF VIO (see Section III-A.2), KLT-based visual tracking is employed to build feature tracks in the current sliding window. However, instead of marginalizing out the old cloned camera poses as in the standard MSCKF, we select certain clones to be keyframes retained in the state vector for loop closure. While different heuristics may be used to select new keyframes, for example, based on the image parallax [18] or feature tracking quality [4], in this work, for proof-of-concept purposes, we simply add new keyframes at a fixed time interval. Once a cloned pose is chosen to be a keyframe, its corresponding state becomes part of the keyframe state [see (10)], whose cross-correlations (instead of autocovariance) will be updated at future times. This can be justified by the fact that when cloned poses reach the end of the sliding window and are selected as keyframes, their estimates are often accurate and can be assumed to have reached their steady states with matured but *non-zero* uncertainty, which will be properly and efficiently tracked in our estimator, instead of being naively assumed to be perfect with zero uncertainty, e.g., as in [18].

To perform keyframe-based loop closing, we leverage the state-of-the-art DBoW2 method [31] for finding loop closure candidates. When a new keyframe is inserted, the DBoW2 database is updated with the new keyframe image by extracting 300 FAST features [32] along with their ORB descriptors [33]. To detect loop closures with the current camera image, we query the DBoW2 database to retrieve the top keyframes that are visually similar to it. After retrieval, a geometric check of the top candidate keyframe from the

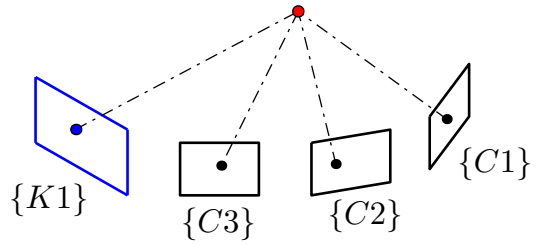

Fig. 2: Illustration a keyframe-based loop closure scenario where an active feature tracked over three clones has matched to a keyframe $\{K1\}$. An additional feature measurement from the keyframe (blue) is added to the feature track such that an implicit loop closure constraint (by viewing the same scene) is formed and can be utilized in the EKF update.

database is performed by ensuring that the fundamental matrix can be calculated between the query image and candidate keyframe (see Fig. 1).

Once a loop closure keyframe has been determined, we then incorporate the feature matches between the actively tracked features in the sliding window and those in the keyframe. For example, as illustrated in Fig. 2, if a frame $\{C3\}$ is detected as matching a keyframe $\{K1\}$, then all extracted features in $\{C3\}$ will try to match with features extracted in $\{K1\}$. Specifically, when an actively tracked feature is matched to a keyframe feature, we add the additional keyframe observation to the feature track. Special care has to be taken that an active feature can only match to a keyframe once; that is, one feature measurement can only be involved in one feature track, in order to prevent reuse of information and thus ensure consistency. Note that for computational savings, the current image is only matched to a single keyframe while this can be easily modified to allow for more loop closure keyframes. We thus use the additional observations from the loop closure keyframe along with feature tracks from the current window to perform a MSCKF update once the active feature has lost track or reaches sliding window size. Explained in the following section, we update the active state estimate $\mathbf{x}_A$, its covariance, and the cross-correlations between the active state and keyframe state $\mathbf{x}_S$.

3) Update: Once feature tracks including both active feature measurements and (if any) loop closure constraints are ready for processing in the current sliding window, we perform a SKF update within the MSCKF framework (and thus Schmidt-MSCKF) based on these visual feature measurements. Specifically, as in the standard MSCKF, we first perform BA with all the feature measurements in the current window to triangulate the 3D feature positions ${}^G\mathbf{p}_f$. Their estimated quantities ${}^G\hat{\mathbf{p}}_f$ are needed in linearizing the nonlinear camera measurements (6) [noting that features are not maintained in our state vector (10)]. For a given feature $\mathbf{z}_k$, we perform linear marginalization of its position (i.e., null space operation) [34], and partition Eq. (9) as follows:

$$\mathbf{r}'_f \simeq \mathbf{H}_{A_k}\tilde{\mathbf{x}}_{A_{k|k-1}} + \mathbf{H}_{S_k}\tilde{\mathbf{x}}_{S_{k|k-1}} + \mathbf{n}'_f \tag{13}$$

where $\mathbf{H}'_x := [\mathbf{H}_{A_k}\ \mathbf{H}_{S_k}]$ as in Equation (9).

Ideally, we can now use the above measurement residual (13) that is independent of features to perform the standard MSCKF update. However, as the keyframes in the state vector $\mathbf{x}_S$ can grow over time, it may quickly become too large to update the full state estimate and covariance in real-time, even though features are already excluded from the state in the MSCKF. Therefore, we instead perform a SKF update. Specifically, we first notice that the standard Kalman gain is computed as follows:

$$\begin{aligned}\mathbf{K}_k &= \begin{bmatrix}\mathbf{K}_{A_k}\\ \mathbf{K}_{S_k}\end{bmatrix} = \begin{bmatrix}\mathbf{P}_{AA_{k|k-1}}\mathbf{H}_{A_k}^\top + \mathbf{P}_{AS_{k|k-1}}\mathbf{H}_{S_k}^\top\\ \mathbf{P}_{SA_{k|k-1}}\mathbf{H}_{A_k}^\top + \mathbf{P}_{SS_{k|k-1}}\mathbf{H}_{S_k}^\top\end{bmatrix}\mathbf{S}_k^{-1}\\ &=: \begin{bmatrix}\mathbf{L}_{A_k}\\ \mathbf{L}_{S_k}\end{bmatrix}\mathbf{S}_k^{-1}\end{aligned} \tag{14}$$

where $\mathbf{S}_k = \mathbf{H}'_x\mathbf{P}_{k|k-1}{\mathbf{H}'_x}^\top + \mathbf{R}'_f$ is the measurement residual covariance [see (9)]. To reduce the computational complexity, we do *not* update the keyframe state nuisance parameters, and thus, as in the the SKF, we set the Kalman gain corresponding to the keyframe state to zero, i.e., $\mathbf{K}_{S_k} = \mathbf{0}$. Under this condition, it can be shown that the optimal gain for the active state is identical to its corresponding standard Kalman gain $\mathbf{K}_{A_k}$. As a result, the state estimate is updated as follows:

$$\hat{\mathbf{x}}_{A_{k|k}} = \hat{\mathbf{x}}_{A_{k|k-1}} + \mathbf{K}_{A_k}\tilde{\mathbf{z}}'_k \tag{15}$$

$$\hat{\mathbf{x}}_{S_{k|k}} = \hat{\mathbf{x}}_{S_{k|k-1}} \tag{16}$$

Note that in the standard EKF covariance is updated by:

$$\begin{aligned}\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \\ \begin{bmatrix}\mathbf{K}_{A_k}\mathbf{S}_k\mathbf{K}_{A_k}^\top & \mathbf{K}_{A_k}\mathbf{H}'^\top_k\begin{bmatrix}\mathbf{P}_{AS_{k|k-1}}\\ \mathbf{P}_{SS_{k|k-1}}\end{bmatrix}\\ \begin{bmatrix}\mathbf{P}_{AS_{k|k-1}}\\ \mathbf{P}_{SS_{k|k-1}}\end{bmatrix}^\top\mathbf{H}'^\top_k\mathbf{K}_{A_k}^\top & \mathbf{K}_{S_k}\mathbf{S}_k\mathbf{K}_{S_k}^\top\end{bmatrix}\end{aligned} \tag{17}$$

It can be seen that the $\mathbf{K}_{S_k}\mathbf{S}_k\mathbf{K}_{S_k}^\top$ term contains the largest computational cost in the standard formulation. The SKF, by contrast, allows for computational savings by setting the Schmidt-Kalman gain $\mathbf{K}_{S_k} = \mathbf{0}$, while still ensuring consistency [24]:

$$\begin{aligned}\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \\ \begin{bmatrix}\mathbf{K}_{A_k}\mathbf{S}_k\mathbf{K}_{A_k}^\top & \mathbf{K}_{A_k}\mathbf{H}'^\top_k\begin{bmatrix}\mathbf{P}_{AS_{k|k-1}}\\ \mathbf{P}_{SS_{k|k-1}}\end{bmatrix}\\ \begin{bmatrix}\mathbf{P}_{AS_{k|k-1}}\\ \mathbf{P}_{SS_{k|k-1}}\end{bmatrix}^\top\mathbf{H}'^\top_k\mathbf{K}_{A_k}^\top & \mathbf{0}\end{bmatrix}\end{aligned} \tag{18}$$

$$\overset{(14)}{=} \mathbf{P}_{k|k-1} - \begin{bmatrix}\mathbf{L}_{A_k}\mathbf{S}_k^{-1}\mathbf{L}_{A_k}^\top & \mathbf{L}_{A_k}\mathbf{S}_k^{-1}\mathbf{L}_{S_k}^\top\\ \mathbf{L}_{S_k}\mathbf{S}_k^{-1}\mathbf{L}_{A_k}^\top & \mathbf{0}\end{bmatrix} \tag{19}$$

In the above expressions, the second equality (19) reveals that the computational complexity becomes $O(n)$, instead of $O(n^2)$ as in (17), which is analyzed in detail next. In summary, the main steps of the proposed Schmidt-MSCKF VINS approach are outlined in Algorithm 1.

4) Computational complexity: During propagation, the computation is dominated by $\mathbf{\Phi}_k\mathbf{P}_{AS_{k|k}}$ [see (12)]. Since $\mathbf{\Phi}_k$ is square with constant size and $\mathbf{P}_{AS_{k|k}}$ is fat with $O(n)$ columns, this matrix multiplication has $O(n)$ cost. After propagation, we create a clone of the the IMU at the current time. This involves simply copying the corresponding rows and columns in $\mathbf{P}_{AA_{k+1|k}}$, which is an $O(1)$ operation, as well as copying the rows corresponding to the state being cloned onto the end of $\mathbf{P}_{AS_{k+1|k}}$, which is $O(n)$.

When performing feature tracking, each image associated with a feature being tracked across the sliding window

Algorithm 1 Schmidt-MSCKF VINS with Loop Closures

Require: Initial state estimate and covariance

1: **loop**
2: Propagate the state to the current image time.
3: Track features from the previous image into the current one.
4: Query the keyframe database for a loop closure and if there is a match, active features are appended with additional measurements from the loop closure keyframe.
5: Features that can be used for update are collected and processed.
6: Oldest pose in the sliding window is either marginalized out or added to the keyframe state (nuisance parameters).
7: **end loop**

can be matched to at most one old keyframe. Since each match can add one bearing measurement, this implies that all features being processed have a constant size number of measurements as seen from a constant size number of poses. Given that the number of tracked features is upper bounded, during each update, there are (at most) a constant number of measurements involving a constant number of variables, which immediately implies that computing $\mathbf{S}_k$ is $O(1)$. By exploiting the sparsity of the measurement Jacobian, we can compute $\mathbf{L}_{A_k} = \mathbf{P}_{AA_{k|k-1}}\mathbf{H}_{A_k}^\top + \mathbf{P}_{AS_{k|k-1}}\mathbf{H}_{S_k}^\top$ and $\mathbf{L}_{S_k}$ in $O(n)$ cost [see (14)], which holds similarly for $\mathbf{L}_{S_k}$. The rest of the update is dominated by $\mathbf{L}_{A_k}\mathbf{S}_k^{-1}\mathbf{L}_{S_k}^\top$ [see (19)], which again is $O(n)$ due to the tall $\mathbf{L}_{S_k}$ with $O(n)$ rows.

During marginalization, when moving a variable $\mathbf{x}_{C_i}$ from the active state to the keyframe state $\mathbf{x}_S$, care must be taken so that the operation remains $O(n)$. In particular, we manage $\mathbf{P}_{AA}$, $\mathbf{P}_{AS}$, and $\mathbf{P}_{SS}$ as separate matrices. When moving $\mathbf{x}_{C_i}$ into the Schmidt state, we first copy the associated column in $\mathbf{P}_{AA}$ as a new column onto $\mathbf{P}_{AS}$. This involves allocating $O(n)$ space followed by a cost of $O(n)$ for copying. Then, we copy the entries from the *row* of $\mathbf{P}_{AS}$ corresponding to $\mathbf{x}_{C_i}$ to form the new rows and columns on the *end* of $\mathbf{P}_{SS}$. As before, this involves allocating a new row and column followed by an $O(n)$ cost in copying. Lastly, we delete the columns and rows of $\mathbf{P}_{AA}$ corresponding to $\mathbf{x}_{C_i}$, which is $O(1)$; and delete the rows of $\mathbf{P}_{AS}$ associated with $\mathbf{x}_{C_i}$, which is $O(n)$. A complete analysis can be found in our technical report [35] with full details on the complexity of each step.

V. Experimental Results

To validate the proposed Schmidt-MSCKF VINS with loop closures, we have performed real-world experiments on different sensor platforms. In the following, we present two sets of results with hand-held sensors, demonstrating that the proposed approach achieves significantly better accuracy than the standard MSCKF (without loop closures) while only incurring marginal computational overhead.

A. Vicon Loops

We first test our VINS system on the Vicon loops dataset [4] which is a 13-minute and 1.2km long trajectory

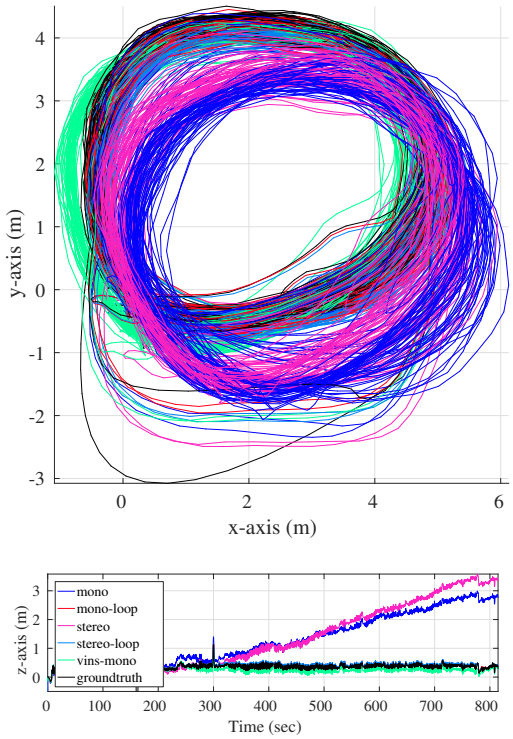

Fig. 3: The estimated trajectories of the proposed Schmidt-MSCKF, standard MSCKF [1], and VINS-Mono [18, 19]. Both monocular and stereo VINS results are presented. In particular, it is clear from the z-axis results (bottom) that the proposed approach is able to achieve bounded-error performance while the standard MSCKF has errors growing over time.

TABLE I: RMSE position errors averaged over 10 runs of the Vicon loops dataset (units are in meters).

	MSCKF	**MSCKF w. Full Loops**	**Schmidt-MSCKF**	**VINS-Mono [18, 19]**
Monocular	1.626	0.113	0.122	0.527
Stereo	1.555	0.093	0.172	-

of a hand-held VI-sensor (stereo camera and IMU) traveling in a large number of loops. Full 6DOF groundtruth is provided by the Vicon motion tracking system at 200Hz. In this test, we select loop closure keyframes at a fixed rate of one every two seconds. For the results presented below, we have developed and validated both stereo and monocular (i.e., using only one of the stereo cameras) VINS algorithms. In particular, the estimators compared are (i) the standard MSCKF without loop closures [1], (ii) the standard MSCKF with loop closures and full covariance update for the keyframes (see Section III-B), (iii) the proposed Schmidt-MSCKF VINS with loop closures and (iv) the open sourced VINS-Mono [18, 19]. The performance metrics used include (i) the root mean squared error (RMSE) that measures the estimation accuracy and (ii) the CPU run time that quantifies the computational cost.

Table I shows the RMSE values averaged over 10 runs (in

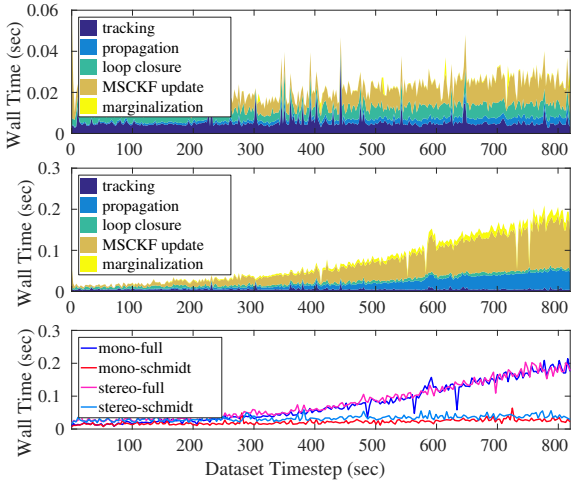

Fig. 4: The CPU run time of the different components and the total execution time (bottom). The breakdown of the proposed monocular Schmidt-MSCKF (top) and that of the standard MSCKF with full covariance updates (middle).

order to consider repeatability of the algorithms). Trajectories were aligned to the groundtruth using the first two minutes. Fig. 3 depicts the estimated trajectories, which clearly shows that the information provided by the loop closure measurements significantly limit the drift of the proposed approach over time. Of the two MSCKF systems that utilize the loop closure constraints, the system that updates the full state estimate and covariance, as expected, achieves a slightly better performance; while the proposed Schmidt-MSCKF closely follows it in accuracy but with a significant computational saving. Specifically, Fig. 4 shows the CPU run time for a single representative run of the monocular VINS on the dataset.[1] As expected, the proposed Schmidt-MSCKF (top) has only *linear* growth in the time it takes to perform an EKF update, while the standard MSCKF with full loop closure update (middle) shows *quadratic* growth in the computation time in both propagation and update. The proposed approach even stays below the real-time threshold of 0.05 sec (camera frame rate is 20Hz), processing 400 keyframes in real-time towards the end of the trajectory. In contrast, the VINS-Mono pose optimization backend (non-realtime thread) takes upwards of 2 sec by the end of the trajectory, which significantly delays the inclusion of loop closure information in the current state estimate.

B. UD Spencer

We further conducted a multi-floor indoor experiment at the University of Delaware (UD) Spencer Lab. An Intel Realsense ZR300 sensor[2] was used to collect a 20-minute and 1.5km long dataset of monocular fisheye images and IMU measurements. The extrinsic calibration provided by the manufacture was used as an initial guess and further refined online. Keyframes were inserted every 0.75 seconds, with a maximum of 886 keyframes in total, with 26% of all features containing at least one keyframe feature observation

[1]Single thread on an Intel(R) Xeon(R) E3-1505Mv6 @ 3.00GHz

[2]https://software.intel.com/en-us/realsense/zr300

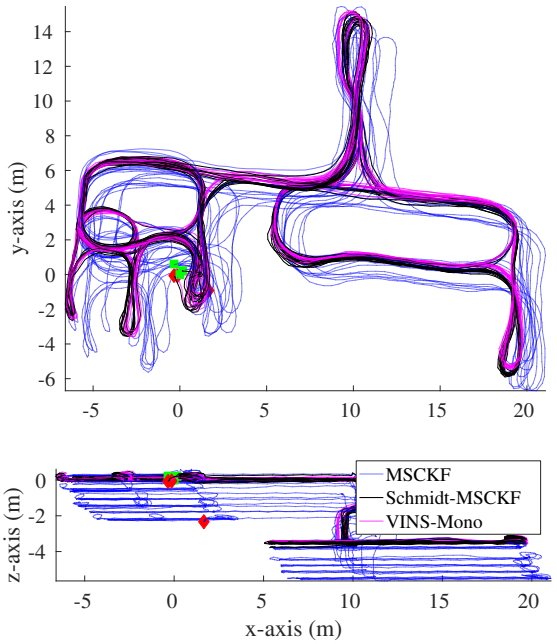

Fig. 5: The trajectories of the standard MSCKF [1] (blue), proposed Schmidt-MSCKF with loop closures (black), and VINS-Mono [18, 19] (magenta). Clearly, without loop closures, the standard MSCKF accumulates significant errors.

during update. Fig. 5 shows the estimated trajectories of the standard MSCKF, proposed monocular Schmidt-MSCKF, and VINS-Mono. Clearly, the proposed method was able to localize with minimal drift over the trajectory while the standard MSCKF drifted significantly. As ground truth was not available for this dataset, we returned to the starting location and evaluated the start-end error to be 3.58m (0.2%), 0.64m (0.04%), 0.26m (0.02%), and 0.16m (0.01%) for the standard MSCKF, the proposed Schmidt-MSCKF, MSCKF with full covariance update, and VINS-Mono (with non-realtime optimization thread). Qualitatively, the trajectory of the proposed method is on-par with that of VINS-Mono, while only requiring a single thread for realtime estimation.

VI. Conclusions and Future Work

In this paper, we have proposed the linear-complexity Schmidt-MSCKF for VINS, which is able to efficiently incorporate loop closure constraints in a single-thread estimator and provide (almost) bounded-error localization performance. The key idea of the proposed approach is to selectively keep keyframes (rather then then a large amount of map features) in the state vector in order utilize loop closure constraints and then leverage the SKF formulation without updating the keyframes (as nuisance parameters) and their covariance while still properly tracking their cross-correlations with the active navigation states. We have extensively validated both monocular and stereo VINS with the proposed Schmidt-MSCKF, showing significant improvements over the standard MSCKF while incurring marginal computational overhead. In the future, we will selectively add and delete keyframes and evaluate the performance of different keyframe selection criteria.

REFERENCES

[1] A. I. Mourikis and S. I. Roumeliotis. "A multi-state constraint Kalman filter for vision-aided inertial navigation". In: *2007 IEEE International Conference on Robotics and Automation*. IEEE. 2007, pp. 3565–3572.

[2] J. Hesch, D. Kottas, S. Bowman, and S. Roumeliotis. "Consistency Analysis and Improvement of Vision-aided Inertial Navigation". In: *IEEE Transactions on Robotics* 30.1 (2013), pp. 158–176.

[3] M. Li and A. Mourikis. "High-Precision, Consistent EKF-based Visual-Inertial Odometry". In: *International Journal of Robotics Research* 32.6 (2013), pp. 690–711.

[4] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. "Keyframe-based visual-inertial odometry using nonlinear optimization". In: *International Journal of Robotics Research* 34.3 (2015), pp. 314–334.

[5] G. Huang, M. Kaess, and J. Leonard. "Towards Consistent Visual-Inertial Navigation". In: *Proc. of the IEEE International Conference on Robotics and Automation*. Hong Kong, China, 2014, pp. 4926–4933.

[6] K. Eckenhoff, P. Geneva, and G. Huang. "Direct Visual-Inertial Navigation with Analytical Preintegration". In: *Proc. of the IEEE International Conference on Robotics and Automation*. Singapore, 2017, pp. 1429–1435.

[7] Z. Huai and G. Huang. "Robocentric Visual-Inertial Odometry". In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*. Madrid, Spain, 2018.

[8] G. Huang, K. Eckenhoff, and J. Leonard. "Optimal-State-Constraint EKF for Visual-Inertial Navigation". In: *Proc. of the International Symposium on Robotics Research*. Sestri Levante, Italy, 2015.

[9] Y. Latif, G. Huang, J. Leonard, and J. Neira. "Sparse Optimization for Robust and Efficient Loop Closing". In: *Robotics and Autonomous Systems* 93 (July 2017), pp. 13–26.

[10] F. Han, H. Wang, G. Huang, and H. Zhang. "Sequence-Based Sparse Optimization Methods for Long-Term Loop Closure Detection in Visual SLAM". In: *Autonomous Robots* (2018). (to appear).

[11] N. Merrill and G. Huang. "Lightweight Unsupervised Deep Loop Closure". In: *Proc. of Robotics: Science and Systems (RSS)*. Pittsburgh, PA, 2018.

[12] S. F. Schmidt. "Application of state-space methods to navigation problems". In: *Advances in Control Systems*. Vol. 3. Elsevier, 1966, pp. 293–340.

[13] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. D. Reid, and J. J. Leonard. "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age". In: *IEEE Transactions on Robotics* 32.6 (2016), pp. 1309–1332.

[14] E. D. Nerurkar, K. J. Wu, and S. I. Roumeliotis. "C-KLAM: Constrained keyframe-based localization and mapping". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2014, pp. 3638–3643.

[15] J. Engel, T. Schöps, and D. Cremers. "LSD-SLAM: Large-scale direct monocular SLAM". In: *European Conference on Computer Vision*. Springer. 2014, pp. 834–849.

[16] R. Mur-Artal and J. D. Tardós. "Visual-inertial monocular SLAM with map reuse". In: *IEEE Robotics and Automation Letters* 2.2 (2017), pp. 796–803.

[17] H. Liu, M. Chen, G. Zhang, H. Bao, and Y. Bao. "ICE-BA: Incremental, Consistent and Efficient Bundle Adjustment for Visual-Inertial SLAM". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1974–1982.

[18] T. Qin, P. Li, and S. Shen. "Vins-mono: A robust and versatile monocular visual-inertial state estimator". In: *IEEE Transactions on Robotics* 34.4 (2018), pp. 1004–1020.

[19] T. Qin, P. Li, and S. Shen. "Relocalization, Global Optimization and Map Merging for Monocular Visual-Inertial SLAM". In: *arXiv preprint arXiv:1803.01549* (2018).

[20] A. I. Mourikis and S. I. Roumeliotis. "A dual-layer estimator architecture for long-term localization". In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2008.* IEEE. 2008, pp. 1–8.

[21] S. Lynen, T. Sattler, M. Bosse, J. A. Hesch, M. Pollefeys, and R. Siegwart. "Get Out of My Lab: Large-scale, Real-Time Visual-Inertial Localization." In: *Robotics: Science and Systems*. 2015.

[22] F. Zheng, G. Tsai, Z. Zhang, S. Liu, C.-C. Chu, and H. Hu. "PI-VIO: Robust and Efficient Stereo Visual Inertial Odometry using Points and Lines". In: *arXiv preprint arXiv:1803.02403* (2018).

[23] S. J. Julier. "A sparse weight Kalman filter approach to simultaneous localisation and map building". In: *2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vol. 3. IEEE. 2001, pp. 1251–1256.

[24] R. C. DuToit, J. A. Hesch, E. D. Nerurkar, and S. I. Roumeliotis. "Consistent map-based 3D localization on mobile devices". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 6253–6260.

[25] N. Trawny and S. I. Roumeliotis. *Indirect Kalman Filter for 3D Attitude Estimation*. Tech. rep. University of Minnesota, Dept. of Comp. Sci. & Eng., Mar. 2005.

[26] A. B. Chatfield. *Fundamentals of High Accuracy Inertial Navigation*. Reston, VA: American Institute of Aeronautics and Astronautics, Inc., 1997.

[27] B. D. Lucas and T. Kanade. "An iterative image registration technique with an application to stereo vision". In: *International Joint Conference on Artificial Intelligence*. Vancouver, BC, 1981, pp. 674–679.

[28] J. Hesch, D. Kottas, S. Bowman, and S. Roumeliotis. "Camera-IMU-based localization: Observability analysis and consistency improvement". In: *International Journal of Robotics Research* 33 (2014), pp. 182–201.

[29] K. Eckenhoff, L. Paull, and G. Huang. "Decoupled, Consistent Node Removal and Edge Sparsification for Graph-based SLAM". In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*. Daejeon, Korea, 2016, pp. 3275–3282.

[30] S. I. Roumeliotis and J. W. Burdick. "Stochastic Cloning: A generalized framework for processing relative state measurements". In: *Proceedings of the IEEE International Conference on Robotics and Automation*. Washington, DC, 2002, pp. 1788–1795.

[31] D. Gálvez-López and J. D. Tardós. "Bags of Binary Words for Fast Place Recognition in Image Sequences". In: *IEEE Transactions on Robotics* 28.5 (2012), pp. 1188–1197.

[32] E. Rosten and T. Drummond. "Machine learning for high-speed corner detection". In: *European conference on computer vision*. Springer. 2006, pp. 430–443.

[33] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. "ORB: An efficient alternative to SIFT or SURF". In: *2011 IEEE international conference on Computer Vision (ICCV)*. IEEE. 2011, pp. 2564–2571.

[34] Y. Yang and G. Huang. "Aided Inertial Navigation with Geometric Features: Observability Analysis". In: *Proc. of the IEEE International Conference on Robotics and Automation*. Brisbane, Australia, 2018.

[35] P. Geneva, K. Eckenhoff, and G. Huang. *Complexity Analysis: A Linear-Complexity EKF for Visual-Inertial Navigation with Loop Closures*. Tech. rep. RPNG-2019-LOOP. Available: http://udel.edu/~ghuang/papers/tr_loop.pdf. University of Delaware, 2019.