Consistent Map-based 3D Localization on Mobile Devices

Ryan DuToit, Joel Hesch, Esha Nerurkar, and Stergios Roumeliotis

***Abstract*—The objective of this paper is to provide *consistent*, real-time 3D localization capabilities to mobile devices navigating within previously mapped areas. To this end, we introduce the Cholesky-Schmidt-Kalman filter (C-SKF), which explicitly considers the uncertainty of the prior map, by employing the *sparse* Cholesky factor of the map's Hessian, instead of its dense covariance–as is the case for the Schmidt-Kalman filter (SKF). By doing so, the C-SKF has memory requirements typically linear in the size of the map, as opposed to quadratic for storing the map's covariance. Moreover, and in order to bound the processing needs of the C-SKF (between linear and quadratic in the size of the map), we introduce a relaxation of the C-SKF algorithm, the sC-SKF, which operates on the Cholesky factors of independent sub-maps resulting from dividing the trajectory and observations used for constructing the map into overlapping segments. Lastly, we assess the processing and memory requirements of the proposed C-SKF and sC-SKF algorithms, and compare their positioning accuracy against other approximate map-based localization approaches that employ measurement-noise-covariance inflation to compensate for the map's uncertainty.**

I. Introduction

In many applications (e.g., surveillance, manufacturing, virtual and augmented reality), robots or people need to accurately localize within a frequently-visited indoor space. In such cases, the accuracy and efficiency of localization can be significantly improved by using a map of the area of operation. In the context of 3D visual-inertial localization, maps computed beforehand [1] have been employed by localization algorithms,[2] such as the multi-state constraint Kalman filter (MSCKF) in [15] and [18], and parallel tracking and mapping (PTAM) in [16] and [26], to improve positioning accuracy based on visual observations of mapped features. These methods achieve real-time performance but at the expense of consistency (their covariance is smaller than the system's true mean squared error). Specifically, [16] and [26] are inconsistent not only because the assumption of a perfect map but also due to the approximations invoked by the optimization algorithm used for localization; thus they cannot provide a reliable measure of their positioning uncertainty. On the other hand, [18], which also assumes that the map is perfect and [15], which ignores the correlations between the estimated state and the map, inflate the camera measurement's noise covariance so as to reduce the effect of inconsistency: overly confident and often unreliable estimates. Nevertheless, inflating the measurement noise does not alleviate the consistency issue that arises from ignoring cross-correlations between the estimated state and the map.

An alternative, approximate method, which explicitly accounts for the map's uncertainty and its correlations with the estimated state, is the Schmidt-Kalman filter (SKF) [23, 24]. The SKF has *linear*, in the map's size, processing requirements as it only needs to update the device's state, covariance, and cross-correlation with the map. Although the map's state and covariance need not be updated, maintaining its covariance has cost *quadratic* in the number of features. This has been the main drawback of the SKF, as well as of its variants applied to simultaneous localization and mapping (e.g., [6, 11]), which has restricted its use to small-size areas.

To overcome this limitation, in this work, we introduce the Cholesky (C)-SKF which has, typically, *linear* in the map's size memory requirements, while providing the same *consistency* properties as the SKF. The key insight behind our approach is that most current methods employed for constructing large-scale maps, such as batch least squares (BLS), compute and use the Cholesky factor of the problem's Hessian, which is sparse [typically linear number of non-zero elements (nnz) in the size of the map[3]] instead of the dense covariance matrix [25]. Additionally, and in order to reduce the processing requirements of the C-SKF – between linear and quadratic in the map's size – we introduce a *consistent* relaxation of the C-SKF, the sub-map (s)C-SKF, which trades localization accuracy for processing speed by operating on the Cholesky factors of the partitioned Hessians resulting from dividing the original map into independent sub-maps. Note that the sub-maps used throughout this work are generated from the method of [8], however other methods which produce sub-maps and their Cholesky factors, such as [5] or [20], could be employed as well. This approximation allows mapping larger areas and/or operating on resource-constrained mobile devices, such as cell phones and tablets.

In summary, the main contributions of this paper are:

- We introduce the Cholesky-Schmidt-Kalman filter (C-SKF), which employs the Hessian's Cholesky factor to compactly represent the map's uncertainty, and efficiently compute *consistent* map-based updates.
- We introduce the sub-map (s)C-SKF, a relaxation of the C-SKF, which employs multiple, independent sub-maps

[1] Besides batch least squares (BLS), pose-graphs [1], and PTAM [12] have also been used for reducing the processing cost of map building. Since such approximations yield *inconsistent* maps, we do not consider them further in the context of this work.

[2] Since we are interested in continuous localization, we do not consider vision-only methods that provide pose estimates only intermittently (e.g., [3]).

[3] In extreme cases, such as when the map is constructed using images taken while hovering over the same scene, the memory requirements may become quadratic. In such cases, alternative approaches, such as PTAM, should be employed instead for localization.

of the area of interest to support real-time, consistent map-based localization on mobile devices.
- We validate the accuracy and consistency of the C-SKF and sC-SKF using visual and inertial measurements from mobile devices against VICON ground truth.

In what follows, we provide an overview of our map-based localization system (Sect. II) and then present the system state and measurement models (Sect. III). In Sect. IV, we describe the limitations of the SKF when applied to map-based localization, and then introduce the C-SKF and the sC-SKF. Our method for generating reliable 2D-3D correspondences is presented in Sect. V. Lastly, we experimentally validate the proposed algorithms in Sect. VI and provide concluding remarks with a discussion on future work in Sect. VII.

II. Map-based Localization Algorithm Overview

Our objective is to design a *consistent* estimator for computing, in real-time, the 3D position and attitude (pose) of a mobile device using inertial and visual measurements, as well as a prior map of the area of operation. To do so, we require the following information from the (offline) mapping process:

- The Cholesky factor of the (sub)map's Hessian.[4]
- The 3D position estimates of the mapped features along with their descriptors (e.g., FREAKs [2] or ORBs [22]) and the vectors of their corresponding images as indexed by a vocabulary tree (VT) [21].

The former is necessary for representing the map's uncertainty, while the latter is used for recognizing mapped features and using their position estimates for updating the device's pose.

Given this information, in Fig. 1, we provide an overview of the (online) image processing and estimation components of our (s)C-SKF map-based localization algorithms. In particular, at the core of our estimator is the MSCKF [7, 18] which processes inertial measurements for propagating the device's state and covariance estimates. Intermittently, and when feature tracks (e.g., Harris corners [9] corners tracked by KLT [14]) become available, the proposed (s)C-SKF adaptation of the MSCKF consistently updates only the device's state-covariance and correlations with the map, but *not* the map itself (Sect. IV-C). Similarly, every time the 2D-to-3D feature-matching pipeline (Sect. V) finds correspondences between the, e.g., FREAK features extracted in the current image and those found in the map, the (s)C-SKF uses this information to, again, update all estimated quantities except the map and its covariance (Sect. IV-D – IV-F).

[4]In this work, we employ the cooperative mapping (CM) algorithm of [8] since, in addition to computing the Cholesky factor of the map's Hessian, it also provides a convenient mechanism for handling sub-maps and computing their corresponding Cholesky factors needed for the sC-SKF.

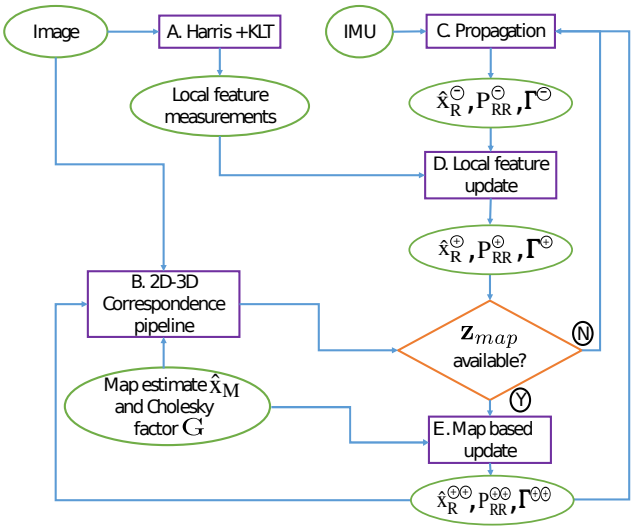

Fig. 1: Cholesky-Schmidt-Kalman filter (sub)map-based localization algorithm overview.

III. System State and Measurement Models

A. Device State

At time-step k, the estimated state is[5]:

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_E^T & \mathbf{x}_{C_{k-1}}^T & \dots & \mathbf{x}_{C_{k-N}}^T & \mathbf{x}_\tau^T \end{bmatrix}^T \quad (1)$$

where $\mathbf{x}_E$ is the evolving state of the device:

$$\mathbf{x}_E = \begin{bmatrix} {}^{I_k}\mathbf{q}_G^T & {}^G\mathbf{p}_{I_k}^T & \mathbf{b}_{g_k}^T & {}^G\mathbf{v}_{I_k}^T & \mathbf{b}_{a_k}^T & {}^G\mathbf{p}_{I_k}^T \end{bmatrix}^T \quad (2)$$

${}^{I_k}\mathbf{q}_G$ is the quaternion representation of the global, $\{G\}$, frame's orientation in the IMU's current frame, $\{I_k\}$, $\mathbf{b}_{a_k}$ and $\mathbf{b}_{g_k}$ are the accelerometer and gyroscope biases, respectively, and ${}^G\mathbf{v}_{I_k}$ and ${}^G\mathbf{p}_{I_k}$ are the velocity and position of $\{I_k\}$ in $\{G\}$. In (1), $\mathbf{x}_{C_i} = \begin{bmatrix} {}^{I_i}\mathbf{q}_G^T & {}^G\mathbf{p}_{I_i}^T & t_{s_i} \end{bmatrix}^T$, $i = k-N, \dots, k-1$ corresponds to previous IMU poses. Following [18], we maintain a sliding window of N such poses so as to process measurements to non-mapped (or local) features without incorporating them into the state vector.

Finally, our problem formulation requires estimating the 4 degree of freedom (d.o.f) transformation between the device's global frame, $\{G\}$, and one or more map's frames of reference, $\{M_i\}$:

$$\mathbf{x}_\tau = \begin{bmatrix} \mathbf{x}_{\tau_1}^T & \mathbf{x}_{\tau_2}^T & \dots & \mathbf{x}_{\tau_L}^T \end{bmatrix}^T, \quad \mathbf{x}_{\tau_i} = \begin{bmatrix} {}^{M_i}\phi_G & {}^G\mathbf{p}_{M_i}^T \end{bmatrix}^T \quad (3)$$

where ${}^G\mathbf{p}_{M_i}$ is the position of $\{M_i\}$ in $\{G\}$, and ${}^{M_i}\phi_G$ is the rotation about gravity between the two frames. Note that since the roll and pitch angles are observable for any vision-aided inertial navitagion system (VINS) system [10], we can choose the z-axis to align with gravity in both the global and map coordinate frames.

[5]To simplify the subsequent derivations, we assume the camera and IMU are time synchronized and co-located. In practice, we estimate their extrinsic calibration parameters and time offset following the approaches of [17] and [7], respectively.

B. IMU measurement model

Following [18], we propagate the state estimate of the device [see (1)] by integrating the IMU's rotational velocity and linear acceleration measurements, $\mathbf{u}_k$,

$$\mathbf{x}_{k+1} = \mathbf{g}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k \tag{4}$$

where $\mathbf{g}$ is a nonlinear function corresponding to the IMU measurement model and $\mathbf{w}_k$ is zero-mean, Gaussian noise of known covariance [4].

C. Local-feature measurement model

As the device traverses its environment, it observes and tracks (via KLT [14]) point features (Harris corners [9]) that have *not* been mapped. These local features are used as in [18] to provide measurement constraints between the $N+1$ IMU-camera poses maintained in the state vector.

The non-linear and linearized local-feature measurement models are

$$\mathbf{z} = \mathbf{h}(\mathbf{x}, \mathbf{p}_f) + \mathbf{n}\,, \qquad \mathbf{r} = \mathbf{H}_R\tilde{\mathbf{x}}_R + \mathbf{H}_f\tilde{\mathbf{p}}_f + \mathbf{n} \tag{5}$$

where $\mathbf{z}$ is the measurement, $\mathbf{h}$ is the perspective-projection camera measurement model, $\mathbf{r}$ is the linearized measurement residual, $\mathbf{x}_R$ ($\tilde{\mathbf{x}}_R$) is the device state (error-state), $\mathbf{p}_f$ ($\tilde{\mathbf{p}}_f$) is the local feature state (error-state), $\mathbf{H}_R$ and $\mathbf{H}_f$ are the measurement Jacobians corresponding to the device and feature states, respectively, and $\mathbf{n}$ is zero-mean, Gaussian noise with known covariance, $\mathbf{R}$.

As in [18], we marginalize the local feature by projecting $\mathbf{r}$ on the left null space of $\mathbf{H}_f$, $\mathbf{U}$:

$$\mathbf{r}^o = \mathbf{H}_R^o\tilde{\mathbf{x}}_R + \mathbf{n}^o \tag{6}$$

where:

$$\mathbf{r}^o = \mathbf{U}^T\mathbf{r}\,, \qquad \mathbf{H}_R^o = \mathbf{U}^T\mathbf{H}_R\,, \qquad \mathbf{n}^o = \mathbf{U}^T\mathbf{n}$$

The new measurement residual, $\mathbf{r}^o$, and Jacobian, $\mathbf{H}_R^o$, are then (Sect. IV-C2) used for updating the device state estimate.

D. Mapped-feature measurement model

When a previously-mapped point-feature, f, (expressed with respect to the IMU-camera frame, $\{I_\lambda^{M_i}\}$, that first observed it during mapping) is detected by the device, we can form the following geometric relationship (see Fig. 2):

$${}^{I_k}\mathbf{p}_f = {}^{I_k}_{G}\mathbf{C}\left({}^{G}\mathbf{p}_{M_i} - {}^{G}\mathbf{p}_{I_k} + {}^{G}_{M_i}\mathbf{C}\left[{}^{M_i}\mathbf{p}_{I_\lambda^{M_i}} + {}^{M_i}_{I_\lambda^{M_i}}\mathbf{C}\;{}^{I_\lambda^{M_i}}\mathbf{p}_f\right]\right) \tag{7}$$

where all rotation matrices, $\mathbf{C}$, are parameterized by their corresponding 3 d.o.f quaternions, except ${}^{G}_{M_i}\mathbf{C}$, which corresponds to a rotation about gravity by an angle ${}^{M_i}\phi_G$. Applying the camera perspective-projection transformation, $\boldsymbol{\pi}$, leads to the following measurement model:

$$\begin{aligned}\mathbf{z}_{map} &= \boldsymbol{\pi}\left({}^{I_k}\mathbf{p}_f\right) + \mathbf{n} \\ &= \mathbf{h}_{map}(\mathbf{x}_k, {}^{I_\lambda^{M_i}}\mathbf{p}_f) + \mathbf{n}\end{aligned} \tag{8}$$

where $\mathbf{n}$ is zero-mean, Gaussian noise with covariance $\mathbf{R}$.

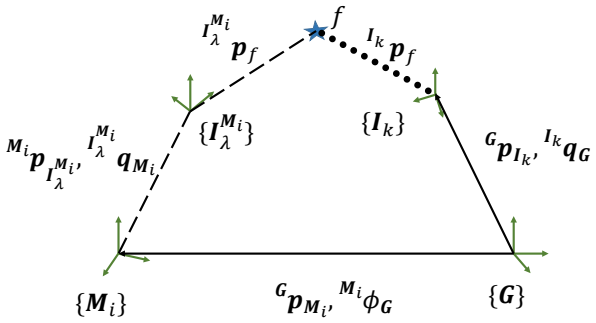

Fig. 2: The geometric relationship of a mapped feature observation. The dotted line is a bearing measurement, solid lines correspond to quantities estimated online. Dashed lines correspond to variables determined during mapping offline.

To simplify the notation, from here on, we omit time indices and denote the state $\mathbf{x}_k$ in (1) by $\mathbf{x}_R$, while we use $\mathbf{x}_M$ to represent the vector comprising of all mapped features and IMU-camera poses. Linearizing (8) yields:

$$\mathbf{r} = \mathbf{H}_R\tilde{\mathbf{x}}_R + \mathbf{H}_M\tilde{\mathbf{x}}_M + \mathbf{n} \tag{9}$$

where $\mathbf{H}_R$ and $\mathbf{H}_M$ are the device and map Jacobians, respectively. Note that both $\mathbf{H}_R$ and $\mathbf{H}_M$ are sparse, as the measurement equation only involves the position and quaternion of the current and mapped IMU-camera pairs, the 4 d.o.f map-to-global transformation, and the position of the feature.

With our system state and measurement models defined, in what follows, we present how measurements (4), (6), and (8) are processed in a consistent manner.

IV. Algorithm Description

In what follows, we first review the SKF, and then introduce the C-SKF and sC-SKF. To simplify notation, we denote updated values with $^{\oplus}$ and propagated values with $^{\ominus}$, rather than using time subscripts. [i.e., $\mathbf{P}_{k+1|k+1} = \mathbf{f}(\mathbf{P}_{k+1|k})$ is expressed as $\mathbf{P}^{\oplus} = \mathbf{f}(\mathbf{P})$, and $\mathbf{P}_{k+1|k} = \mathbf{g}(\mathbf{P}_{k|k})$ is $\mathbf{P}^{\ominus} = \mathbf{g}(\mathbf{P})$]

A. Background: Schmidt-Kalman filter

Consider the current propagated system covariance, $\mathbf{P}$, and Jacobian, $\mathbf{H}$, to be:

$$\mathbf{P} = \begin{bmatrix}\mathbf{P}_{RR} & \mathbf{P}_{RM} \\ \mathbf{P}_{MR} & \mathbf{P}_{MM}\end{bmatrix} \qquad \mathbf{H} = \begin{bmatrix}\mathbf{H}_R & \mathbf{H}_M\end{bmatrix} \tag{10}$$

The state and covariance update equations for the extended Kalman filter (EKF) are:

$$\hat{\mathbf{x}}^{\oplus} = \hat{\mathbf{x}} + \mathbf{K}\mathbf{r} \tag{11}$$

$$\mathbf{P}^{\oplus} = (\mathbf{I} - \mathbf{K}\mathbf{H})\,\mathbf{P}\,(\mathbf{I} - \mathbf{H}^T\mathbf{K}^T) + \mathbf{K}\mathbf{R}\mathbf{K}^T \tag{12}$$

where

$$\begin{aligned}\mathbf{r} &= \mathbf{z} - \mathbf{h}(\hat{\mathbf{x}}) \\ \mathbf{S} &= \mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R} \\ \mathbf{K} &= \mathbf{P}\mathbf{H}^T\mathbf{S}^{-1} = \begin{bmatrix}\mathbf{P}_{RR}\mathbf{H}_R^T + \mathbf{P}_{RM}\mathbf{H}_M^T \\ \mathbf{P}_{MR}\mathbf{H}_R^T + \mathbf{P}_{MM}\mathbf{H}_M^T\end{bmatrix}\mathbf{S}^{-1} \\ &= \begin{bmatrix}\bar{\mathbf{K}}_R \\ \bar{\mathbf{K}}_M\end{bmatrix}\mathbf{S}^{-1} = \bar{\mathbf{K}}\mathbf{S}^{-1} \end{aligned} \tag{13}$$

As shown in [24], the SKF updates only part of state by zeroing the Kalman gain associated with the state elements whose estimates are to remain the same (in our case the map, $\mathbf{x}_M$):

$$\mathbf{K}_{SKF} = \begin{bmatrix}(\bar{\mathbf{K}}_R\mathbf{S}^{-1})^T & \mathbf{0}\end{bmatrix}^T \tag{14}$$

Substituting (14) in (11) yields the following state update:

$$\hat{\mathbf{x}}_R^{\oplus} = \hat{\mathbf{x}}_R + \bar{\mathbf{K}}_R\mathbf{S}^{-1}\mathbf{r} \qquad \hat{\mathbf{x}}_M^{\oplus} = \hat{\mathbf{x}}_M \tag{15}$$

Additionally, by employing (14) in (12), we arrive at the SKF's covariance update:

$$\mathbf{P}_{SKF}^{\oplus} = \mathbf{P} - \begin{bmatrix}\bar{\mathbf{K}}_R\mathbf{S}^{-1}\bar{\mathbf{K}}_R^T & \bar{\mathbf{K}}_R\mathbf{S}^{-1}\bar{\mathbf{K}}_M^T \\ \bar{\mathbf{K}}_M\mathbf{S}^{-1}\bar{\mathbf{K}}_R^T & \mathbf{0}\end{bmatrix} \tag{16}$$

Note that if we express the updated covariance of the EKF as a function of the SKF updated covariance, it is straightforward to show:

$$\mathbf{P}_{EKF}^{\oplus} = \mathbf{P}_{SKF}^{\oplus} + \begin{bmatrix}\mathbf{0} \\ \bar{\mathbf{K}}_M\end{bmatrix}\mathbf{S}^{-1}\begin{bmatrix}\mathbf{0} & \bar{\mathbf{K}}_M^T\end{bmatrix} \tag{17}$$

$$\Rightarrow \mathbf{P}_{EKF}^{\oplus} \preceq \mathbf{P}_{SKF}^{\oplus} \tag{18}$$

Since $\mathbf{S}^{-1}$ is positive definite, and hence $\begin{bmatrix}\mathbf{0} \\ \bar{\mathbf{K}}_M\end{bmatrix}\mathbf{S}^{-1}\begin{bmatrix}\mathbf{0} & \bar{\mathbf{K}}_M^T\end{bmatrix}$ must be positive semi-definite.

Thus, we have shown that the SKF is consistent as it does not underestimate the system covariance. Furthermore, as evident from (16), the cost of an SKF covariance update is linear in the size of the map, as only the device's covariance and cross-correlation terms need to be updated, while $\mathbf{H}$ in (10) is sparse.

On the other hand, however, the SKF requires storing the covariance of the map, $\mathbf{P}_{MM}$, which is dense. To better appreciate the challenge this poses on mobile devices, we note that the covariance of a map generated from 5 min of visual and inertial data requires 16 GB of storage space. Instead, the sparse Cholesky factor of the corresponding Hessian matrix requires only 0.45 GB. This motivates us to introduce the Cholesky-SKF (C-SKF) in the next sections.

B. C-SKF device-map initialization

In what follows, we start by describing the process for estimating the 4 d.o.f transformation between the map's frame and the device's global reference frame, as well as its covariance and correlations with all estimated quantities.

Specifically, consider the first time the mobile device observes two or more previously-mapped features. An initial estimate, $\hat{\mathbf{x}}_\tau$, for the uniform 4 d.o.f transformation is obtained by employing the 2+1 pt RANSAC [13]. Furthermore, we partition the current state as

$$\mathbf{x} = \begin{bmatrix}\mathbf{x}_{R'}^T & \mathbf{x}_\tau^T & \mathbf{x}_M^T\end{bmatrix}^T \tag{19}$$

where $\mathbf{x}_{R'}$ comprises of the remaining elements of the device's state vector [see (1)], and the corresponding covariance as

$$\mathbf{P} = \begin{bmatrix}\mathbf{P}_{R'R'} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{\tau\tau} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & (\mathbf{G}\mathbf{G}^T)^{-1}\end{bmatrix}, \mathbf{P}_{\tau\tau} = \lim_{\mu\to\infty}(\mu\mathbf{I}_4) \tag{20}$$

where $\mathbf{P}_{\tau\tau}$ is the covariance of the unknown 4 d.o.f transformation, $\mathbf{G}\mathbf{G}^T$ is the Hessian of the map, and $\mathbf{G}$ is its Cholesky factor. After linearizing the measurement model in (8) and denoting the corresponding Jacobian as

$$\mathbf{H} = \begin{bmatrix}\mathbf{H}_{R'} & \mathbf{H}_\tau & \mathbf{H}_M\end{bmatrix} \tag{21}$$

we employ (13)-(15) to update the estimates of $\mathbf{x}_{R'}$ and $\mathbf{x}_\tau$:

$$\hat{\mathbf{x}}_{R'}^{\oplus} = \hat{\mathbf{x}}_{R'} + \mathbf{P}_{R'R'}\mathbf{H}_{R'}^T\mathbf{S}^{-1}\mathbf{r} \tag{22}$$

$$\hat{\mathbf{x}}_\tau^{\oplus} = \hat{\mathbf{x}}_\tau + (\mathbf{H}_\tau^T\mathbf{A}^{-1}\mathbf{H}_\tau)^{-1}\mathbf{H}_\tau^T\mathbf{A}^{-1}\mathbf{r} \tag{23}$$

$$\hat{\mathbf{x}}_M^{\oplus} = \hat{\mathbf{x}}_M \tag{24}$$

Additionally, employing (16), it can be shown that the updated SKF covariance is

$$\mathbf{P} = \begin{bmatrix}\mathbf{P}_{R'R'}^{\oplus} & \mathbf{P}_{R'\tau}^{\oplus} & \mathbf{P}_{R'M}^{\oplus} \\ \mathbf{P}_{R'\tau}^{\oplus T} & \mathbf{P}_{\tau\tau}^{\oplus} & \mathbf{P}_{\tau M}^{\oplus} \\ \mathbf{P}_{R'M}^{\oplus T} & \mathbf{P}_{\tau M}^{\oplus T} & (\mathbf{G}\mathbf{G}^T)^{-1}\end{bmatrix} = \begin{bmatrix}\mathbf{P}_{RR}^{\oplus} & \mathbf{P}_{RM}^{\oplus} \\ \mathbf{P}_{MR}^{\oplus} & (\mathbf{G}\mathbf{G}^T)^{-1}\end{bmatrix} \tag{25}$$

where

$$\begin{aligned}\mathbf{P}_{R'R'}^{\oplus} &= \mathbf{P}_{R'R'} - \mathbf{P}_{R'R'}\mathbf{H}_{R'}^T\mathbf{S}^{-1}\mathbf{H}_{R'}\mathbf{P}_{R'R'} \\ \mathbf{P}_{R'\tau}^{\oplus} &= -\mathbf{P}_{R'R'}\mathbf{H}_{R'}^T\mathbf{A}^{-1}\mathbf{H}_\tau(\mathbf{H}_\tau^T\mathbf{A}^{-1}\mathbf{H}_\tau)^{-1} \\ \mathbf{P}_{\tau\tau}^{\oplus} &= (\mathbf{H}_\tau^T\mathbf{A}\mathbf{H}_\tau)^{-1} \\ \mathbf{P}_{RM}^{\oplus} &= \begin{bmatrix}\mathbf{P}_{R'M}^{\oplus} \\ \mathbf{P}_{\tau M}^{\oplus}\end{bmatrix} = \begin{bmatrix}\mathbf{P}_{R'R'} - \mathbf{P}_{R'R'}\mathbf{H}_{R'}^T\mathbf{S}^{-1}\mathbf{J} \\ (\mathbf{H}_\tau^T\mathbf{A}^{-1}\mathbf{H}_\tau)^{-1}\mathbf{H}_\tau^T\mathbf{A}^{-1}\mathbf{J}\end{bmatrix}\mathbf{G}^{-1} \\ &= \mathbf{\Gamma}\mathbf{G}^{-1}\end{aligned} \tag{26}$$

and

$$\begin{aligned}\mathbf{S}^{-1} &= \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{H}_\tau(\mathbf{H}_\tau^T\mathbf{A}^{-1}\mathbf{H}_\tau)\mathbf{H}_\tau^T\mathbf{A}^{-1} \\ \mathbf{A} &= \mathbf{H}_{R'}\mathbf{P}_{R'R'}\mathbf{H}_{R'} + \mathbf{J}\mathbf{J}^T + \mathbf{R}\end{aligned}$$

Finally, $\mathbf{J}$ is defined as:

$$\mathbf{G}\mathbf{J}^T = \mathbf{H}_M^T \tag{27}$$

A key element of our approach is that, from this point on, instead of updating the cross-correlation term, $\mathbf{P}_{RM}$, we will represent it in a factorized form [see (26)] and apply updates on its factor, $\mathbf{\Gamma}$, as is shown in Sect. IV-C and Sect. IV-D. Following this convention, we have:

$$\mathbf{P} = \begin{bmatrix}\mathbf{P}_{RR} & \mathbf{\Gamma}\mathbf{G}^{-1} \\ (\mathbf{\Gamma}\mathbf{G}^{-1})^T & (\mathbf{G}\mathbf{G}^T)^{-1}\end{bmatrix} \tag{28}$$

Note also that we do not need to compute the inverse of the Cholesky factor, $\mathbf{G}$. Instead, all update equations involving $\mathbf{G}$

will be of the same form as (27), where a back-solve involving the sparse $\mathbf{G}$ is required for efficiently computing $\mathbf{J}^T$.

C. C-SKF propagation and local-feature update

1) IMU-based Propagation: By employing the IMU measurement model in (4), the device's state is propagated while, as is the case for the SKF, the map's state remains the same:

$$\hat{\mathbf{x}}_R^{\ominus} = \mathbf{f}(\hat{\mathbf{x}}_R, \mathbf{u}) \,, \qquad \hat{\mathbf{x}}_M^{\ominus} = \mathbf{x}_M \tag{29}$$

On the other hand, and in order to propagate the covariance of the C-SKF [see (28)], we employ the EKF covariance propagation equation:

$$\mathbf{P}^{\ominus} = \mathbf{\Phi}_C \mathbf{P} \mathbf{\Phi}_C^T + \mathbf{Q}_C = \begin{bmatrix} \mathbf{\Phi} \mathbf{P}_{RR} \mathbf{\Phi}^T + \mathbf{Q} & \mathbf{\Phi} \mathbf{\Gamma} \mathbf{G}^{-1} \\ (\mathbf{\Phi} \mathbf{\Gamma} \mathbf{G}^{-1})^T & (\mathbf{G}\mathbf{G}^T)^{-1} \end{bmatrix} \tag{30}$$

with

$$\mathbf{\Phi}_C = \begin{bmatrix} \mathbf{\Phi} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \,, \qquad \mathbf{Q}_C = \begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{31}$$

where $\mathbf{\Phi}$ and $\mathbf{Q}$ are the IMU Jacobian of the state and corresponding noise covariance, respectively.

As evident from (30), the device's propagated covariance is computed at low cost, while the cross-correlation terms only require modifying the cross-correlation factor, $\mathbf{\Gamma}$, (i.e., $\mathbf{\Gamma}^{\ominus} = \mathbf{\Phi}\mathbf{\Gamma}$) at cost linear in the size of the map.

2) C-SKF Local-Feature Measurement Update: When a local-feature-track measurement becomes available [see (6)], we arrive at the following Jacobian:

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_R^o & \mathbf{0} \end{bmatrix} \tag{32}$$

As in (14), we zero out the Kalman gain associated with the mapped states, leading to an update of the device state, while the map remains the same:

$$\hat{\mathbf{x}}_R^{\oplus} = \hat{\mathbf{x}}_R + \mathbf{P}_{RR} \mathbf{H}_R^{oT} \mathbf{S}^{-1} \mathbf{r} \,, \qquad \hat{\mathbf{x}}_M^{\oplus} = \hat{\mathbf{x}}_M \tag{33}$$

For the covariance update, we first denote $\bar{\mathbf{K}} = \mathbf{P}\mathbf{H}$:

$$\begin{bmatrix} \bar{\mathbf{K}}_R \\ \bar{\mathbf{K}}_M \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{RR} & \mathbf{\Gamma}\mathbf{G}^{-1} \\ (\mathbf{\Gamma}\mathbf{G}^{-1})^T & (\mathbf{G}\mathbf{G}^T)^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{H}_R^{oT} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{RR}\mathbf{H}_R^{oT} \\ \mathbf{G}^{-T}\mathbf{\Gamma}^T \mathbf{H}_R^{oT} \end{bmatrix} \tag{34}$$

Next, we employ (34) and (28) in (16) without evaluating $\bar{\mathbf{K}}_M$, yielding:

$$\mathbf{P}_{RR}^{\oplus} = \mathbf{P}_{RR} - \mathbf{P}_{RR}\mathbf{H}_R^{oT}\mathbf{S}^{-1}\mathbf{H}_R^o\mathbf{P}_{RR} \,, \qquad \mathbf{P}_{MM}^{\oplus} = \mathbf{P}_{MM} \tag{35}$$

and

$$\begin{aligned} \mathbf{P}_{RM}^{\oplus} &= \mathbf{\Gamma}\mathbf{G}^{-1} - \bar{\mathbf{K}}_R \mathbf{S}^{-1} \bar{\mathbf{K}}_M^T \\ &= \left(\mathbf{I} - \mathbf{P}_{RR}\mathbf{H}_R^{oT}\mathbf{S}^{-1}\mathbf{H}_R^o\right)\mathbf{\Gamma}\mathbf{G}^{-1} \\ &= \mathbf{\Gamma}^{\oplus}\mathbf{G}^{-1} \end{aligned} \tag{36}$$

As evident from (33)–(36), the local-feature update has complexity *linear* in the size of the map, as we only need to apply a standard EKF update on the device's covariance, and update the cross-correlation factor, $\mathbf{\Gamma}$.

D. C-SKF map-based updates

When the device observes previously mapped features (Sect. V), we employ the methodology of Sect. IV-A [(13)–(16)] using the measurement model of (8)–(9), and operate on the system covariance with factorized cross-correlation, as defined in (28). Specifically, we denote $\bar{\mathbf{K}} = \mathbf{P}\mathbf{H}^T$ as:

$$\begin{bmatrix} \bar{\mathbf{K}}_R \\ \bar{\mathbf{K}}_M \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{RR}\mathbf{H}_R^T + \mathbf{\Gamma}\mathbf{G}^{-1}\mathbf{H}_M^T \\ \mathbf{G}^{-T}\mathbf{\Gamma}^T\mathbf{H}_R^T + (\mathbf{G}\mathbf{G}^T)^{-1}\mathbf{H}_M^T \end{bmatrix} \tag{37}$$

Note that we do not explicitly compute $\bar{\mathbf{K}}_M$, instead we first compute $\mathbf{J}$ as:

$$\mathbf{G}\mathbf{J}^T = \mathbf{H}_M^T \tag{38}$$

Because $\mathbf{G}$ is triangular, we can compute $\mathbf{J}$ with a back-solve operation. Substituting $\mathbf{J}$ into (37) yields:

$$\begin{bmatrix} \bar{\mathbf{K}}_R \\ \bar{\mathbf{K}}_M \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{RR}\mathbf{H}_R^T + \mathbf{\Gamma}\mathbf{J}^T \\ \mathbf{G}^{-T}\mathbf{\Gamma}^T\mathbf{H}_R^T + \mathbf{G}^{-T}\mathbf{J}^T \end{bmatrix} \tag{39}$$

Next, we compute the residual covariance:

$$\mathbf{S} = \mathbf{H}_R\mathbf{P}_{RR}\mathbf{H}_R^T + \mathbf{H}_R\mathbf{\Gamma}\mathbf{J}^T + \mathbf{J}\mathbf{\Gamma}^T\mathbf{H}_R^T + \mathbf{J}\mathbf{J}^T + \mathbf{R} \tag{40}$$

and state update [see (15)]:

$$\hat{\mathbf{x}}_R^{\oplus} = \hat{\mathbf{x}}_R + \bar{\mathbf{K}}_R\mathbf{S}^{-1}\mathbf{r} \,, \qquad \hat{\mathbf{x}}_M^{\oplus} = \hat{\mathbf{x}}_M \tag{41}$$

Finally, we update the covariance using (16) with (39). Avoiding the calculation of $\bar{\mathbf{K}}_M$, and following the same process in (36), we produce a factorized form of the updated cross-correlation:

$$\begin{aligned} \mathbf{P}_{RR}^{\oplus} &= \mathbf{P}_{RR} - \bar{\mathbf{K}}_R\mathbf{S}^{-1}\bar{\mathbf{K}}_R^T \\ \mathbf{P}_{RM}^{\oplus} &= \mathbf{\Gamma}\mathbf{G}^{-1} - \bar{\mathbf{K}}_R\mathbf{S}^{-1}(\mathbf{H}_R\mathbf{\Gamma}\mathbf{G}^{-1} + \mathbf{J}\mathbf{G}^{-1}) \\ &= [\mathbf{\Gamma} - \bar{\mathbf{K}}_R\mathbf{S}^{-1}(\mathbf{H}_R\mathbf{\Gamma} + \mathbf{J})]\mathbf{G}^{-1} \\ &= \mathbf{\Gamma}^{\oplus}\mathbf{G}^{-1} \end{aligned} \tag{42}$$

At this point, we should note that unlike propagation and local-feature updates, the processing requirements of mapped-feature updates are not strictly linear in the size of the map. The main bottleneck is (38), as computing $\mathbf{J}$ has complexity between linear and quadratic in the size of the map (depending on the structure of $\mathbf{G}$). As shown in Sect. VI, as the map grows, the time to compute $\mathbf{J}$ becomes unacceptable for real-time operation on a mobile device. This limitation motivates us to introduce the sub-map relaxation of Sect. IV-E and Sect. IV-F. That is, we decrease the size of $\mathbf{G}$ in (38) by partitioning the map into independent sub-maps, while retaining consistency.

E. Sub-map relaxation

Before discussing the sC-SK, we first describe the sub-mapping relaxation process. As shown in Fig. 3 and described below, we divide the trajectory and associated features into two separate sets:[6]

First, the IMU-camera poses $\boldsymbol{\xi}_i$, are divided into two sets ($[\boldsymbol{\xi}_1$ to $\boldsymbol{\xi}_N]$ and $[\boldsymbol{\xi}_{N+1}$ to $\boldsymbol{\xi}_M]$). Our current implementation

[6]To simplify the explanation, without loss of generality, we describe the case for two sub-maps

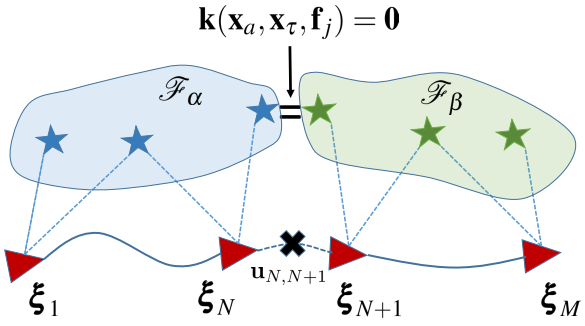

Fig. 3: A partitioning of a single map into two sub-maps, with a geometric constraint, $\mathbf{k}(\mathbf{x}_a, \mathbf{x}_\tau, \mathbf{f}_j) = \mathbf{0}$, imposed to common features.

evenly distributes these sets in time. As part of our future work, we seek to find an optimal partitioning into sub-maps. With this division defined, we partition features into two sets: (i) $\mathscr{F}_\alpha$: those observed by IMU-camera poses $\boldsymbol{\xi}_1$ to $\boldsymbol{\xi}_N$, and (ii) $\mathscr{F}_\beta$: those observed by IMU-camera poses $\boldsymbol{\xi}_{N+1}$ to $\boldsymbol{\xi}_M$. Features in $\mathscr{F}_\alpha \cap \mathscr{F}_\beta$ are common features appearing in both sub-maps. The cooperative mapping (CM) algorithm [8] "duplicates" these features such that the two sub-map's individual cost functions are independent, but also introduces a non-linear constraint between these common features

$$\mathbf{k}(\mathbf{x}_a, \mathbf{x}_\tau, \mathbf{f}_j) = \mathbf{0},\ \mathbf{f}_j \in \mathscr{F}_\alpha \cap \mathscr{F}_\beta \tag{43}$$

where $\mathbf{x}_a$ is the state of all IMU-camera poses and $\mathbf{x}_\tau$ is the 4 d.o.f transformation between the two sub-maps. For each common feature, this constraint enforces its corresponding "duplicate" in $\mathscr{F}_\alpha$ or $\mathscr{F}_\beta$ to have the same physical position. Finally, all camera and IMU measurements, $\mathbf{z}_{i,j}$ and $\mathbf{u}_{i,i+1}$, are assigned to their corresponding sub-map (with the execption of $\mathbf{u}_{N,N+1}$, which is discarded).

With such a partitioning, by ignoring the common-feature constraints, we can form two independent cost functions corresponding to each sub-map, $\mathscr{C}_1$ and $\mathscr{C}_2$:

$$\begin{aligned}
\mathscr{C}_1 &= \sum_{\substack{i=1 \\ \mathbf{f}_j \in \mathscr{F}_\alpha}}^{N} ||\mathbf{z}_{i,j} - \mathbf{h}(\boldsymbol{\xi}_i, \mathbf{f}_j)||_{\mathbf{R}} + \sum_{i=1}^{N-1} ||\boldsymbol{\xi}_{i+1} - \mathbf{g}(\boldsymbol{\xi}_i, \mathbf{u}_{i,i+1})||_{\mathbf{Q}} \\
\mathscr{C}_2 &= \sum_{\substack{i=N+1 \\ \mathbf{f}_j \in \mathscr{F}_\beta}}^{M} ||\mathbf{z}_{i,j} - \mathbf{h}(\boldsymbol{\xi}_i, \mathbf{f}_j)||_{\mathbf{R}} + \sum_{i=N+1}^{M-1} ||\boldsymbol{\xi}_{i+1} - \mathbf{g}(\boldsymbol{\xi}_i, \mathbf{u}_{i,i+1})||_{\mathbf{Q}}
\end{aligned} \tag{44}$$

where $\mathbf{g}$ and $\mathbf{h}$ are defined in (5) and (4), respectively. Summing these two cost functions and imposing the common feature constraints yields:

$$\begin{aligned}
&\mathscr{C} = \mathscr{C}_1 + \mathscr{C}_2 \\
&\text{s. t.} \quad \mathbf{k}(\mathbf{x}_a, \mathbf{x}_\tau, \mathbf{f}_j) = \mathbf{0},\ \mathbf{f}_j \in \mathscr{F}_\alpha \cap \mathscr{F}_\beta
\end{aligned} \tag{45}$$

We minimize (45) by employing the CM algorithm of [8].

F. Cholesky-Kalman-Schmidt with sub-maps (sC-SKF)

In our problem, we take advantage of the high-accuracy estimate computed from the solution of (45), but relax the information attributed to each sub-map by storing the Cholesky factors, $\mathbf{G}_i$, resulting from each corresponding cost function in (44) linearized at the CM solution of (45). Such a relaxation causes the sub-maps to become independent, but maintains consistency.

Theorem 1: The covariance of each sub-map when ignoring the common-feature constraints is larger or equal, in the positive semi-definite sense, to that computed from the CM.

Proof: See Appendix A for proof of Theorem 1. ■

To process mapped measurements with sub-maps, we represent the system covariance as:

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{RR} & \boldsymbol{\Gamma}_1 \mathbf{G}_1^{-1} & \boldsymbol{\Gamma}_2 \mathbf{G}_2^{-1} \\ (\boldsymbol{\Gamma}_1 \mathbf{G}_1^{-1})^T & (\mathbf{G}_1 \mathbf{G}_1^T)^{-1} & \mathbf{0} \\ (\boldsymbol{\Gamma}_2 \mathbf{G}_2^{-1})^T & \mathbf{0} & (\mathbf{G}_2 \mathbf{G}_2^T)^{-1} \end{bmatrix} \tag{46}$$

where $\boldsymbol{\Gamma}_i$ and $\mathbf{G}_i$ is the cross-correlation factor from (28) and the Cholesky factor of the i'th sub-map, respectively. When a mapped feature in the first sub-map is observed (without loss of generality), the measurement Jacobian is:

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_R & \mathbf{H}_1 & \mathbf{0} \end{bmatrix} \tag{47}$$

where $\mathbf{H}_1$ is the measurement Jacobian corresponding to the states in the first sub-map. We compute $\mathbf{J}_1$ and denote $\bar{\mathbf{K}}$ in similar fashion as in (38) and (39), respectively:

$$\mathbf{G}_1 \mathbf{J}_1^T = \mathbf{H}_1^T, \qquad \begin{bmatrix} \bar{\mathbf{K}}_R \\ \bar{\mathbf{K}}_1 \\ \bar{\mathbf{K}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{RR}\mathbf{H}_R^T + \boldsymbol{\Gamma}_1 \mathbf{J}_1^T \\ \mathbf{G}_1^{-T}(\boldsymbol{\Gamma}_1^T \mathbf{H}_R^T + \mathbf{J}_1^T) \\ \mathbf{G}_2^{-T} \boldsymbol{\Gamma}_2^T \mathbf{H}_R^T \end{bmatrix} \tag{48}$$

The residual covariance and state update are computed as:

$$\mathbf{S} = \mathbf{H}_R \mathbf{P}_{RR} \mathbf{H}_R^T + \mathbf{H}_R \boldsymbol{\Gamma}_1 \mathbf{J}_1^T + \mathbf{J}_1 \boldsymbol{\Gamma}_1^T \mathbf{H}_R^T + \mathbf{J}_1 \mathbf{J}_1^T + \mathbf{R}$$

$$\hat{\mathbf{x}}_R^{\oplus} = \hat{\mathbf{x}}_R + \bar{\mathbf{K}}_R \mathbf{S}^{-1} \mathbf{r}, \qquad \hat{\mathbf{x}}_1^{\oplus} = \hat{\mathbf{x}}_1, \qquad \hat{\mathbf{x}}_2^{\oplus} = \hat{\mathbf{x}}_2 \tag{49}$$

Finally, we update the device covariance and cross-correlation using the same factorization as the single-map case [see (42)]:

$$\mathbf{P}_{RR}^{\oplus} = \mathbf{P}_{RR} - \bar{\mathbf{K}}_R \mathbf{S}^{-1} \bar{\mathbf{K}}_R^T, \qquad \mathbf{P}_{11}^{\oplus} = \mathbf{P}_{11}, \qquad \mathbf{P}_{22}^{\oplus} = \mathbf{P}_{22}$$

$$\begin{aligned}
\mathbf{P}_{R1}^{\oplus} &= \boldsymbol{\Gamma}_1 \mathbf{G}_1^{-1} - \bar{\mathbf{K}}_R \mathbf{S}^{-1} \bar{\mathbf{K}}_1^T \\
&= [\boldsymbol{\Gamma}_1 - \bar{\mathbf{K}}_R \mathbf{S}^{-1} (\mathbf{H}_R \boldsymbol{\Gamma}_1 + \mathbf{J}_1)] \mathbf{G}_1^{-1} \\
&= \boldsymbol{\Gamma}_1^{\oplus} \mathbf{G}_1^{-1}
\end{aligned} \tag{50}$$

$$\begin{aligned}
\mathbf{P}_{R2}^{\oplus} &= \boldsymbol{\Gamma}_2 \mathbf{G}_2^{-1} - \bar{\mathbf{K}}_R \mathbf{S}^{-1} \bar{\mathbf{K}}_2^T \\
&= [\mathbf{I} - \bar{\mathbf{K}}_R \mathbf{S}^{-1} \mathbf{H}_R] \boldsymbol{\Gamma}_2 \mathbf{G}_2^{-1} \\
&= \boldsymbol{\Gamma}_2^{\oplus} \mathbf{G}_2^{-1}
\end{aligned} \tag{51}$$

By employing the sub-map relaxation, we have significantly lowered the computational requirements of the C-SKF. Note again that the bottleneck of our system is the computation of $\mathbf{J}_1$ [see (48)]. With the sub-map relaxation, however, we can adjust the sC-SKF to given hardware constraints by adjusting

the number of sub-maps employed. Increasing the amount of sub-maps decreases the size of the system when solving for $\mathbf{J}_1$, dramatically reducing computation time. It should be noted, however, that increasing the number of sub-maps decreases the information associated with the map, which typically leads to slightly less accurate device pose estimates (Sect. VI).

V. 2D-3D Feature Correspondence Pipeline

Before employing map-based updates, we identify correspondences between 2D feature measurements (FREAK features [2]) in the current image and 3D features which have been previously mapped (Fig. 1 Box B). Our pipeline takes a dual layer approach: the first considers the case when the estimator has no prior for the 4 d.o.f device-to-map transformation, while the second takes advantage of such a prior to improve efficiency.

A. Pose-less correspondence generation

1) VT query: We query the saved VT with the current image, which returns up to five mapped images of similar appearance.

2) Feature matching: We apply binary descriptor matching between features in the query and returned images.

3) Outlier rejection: We apply 3+1 pt RANSAC [19] on each image returned by the VT. If less than 7 correspondences remain, we classify that VT return as an outlier. Then, the 2+1 pt RANSAC [13] is applied over the remaining set of inlier correspondences, If less than 13 correspondences remain, we classify all feature measurements as outliers.

B. Pose-assisted correspondence generation

During nominal operation, the estimator will have a reliable estimate for the device-map transformation. We leverage this estimate to re-project a subset of mapped features into the current image bypassing the VT (the main bottleneck in the pose-less pipeline).

1) Pose-based matching: We find mapped images whose camera poses are nearby the current camera pose. A number of heuristics could be used to define nearby mapped images; in our case we require images to be within 3 m of the current position, and have an optical axis within 45 deg of the current optical axis. Images satisfying these criteria are the initial set of mapped image matches.

2) Co-visibility: We add mapped images which view at least 50 common features with any of the pose-based image matches to the set of mapped image matches. This step allows the pipeline to include images that are far from the current camera pose, but still view the same scene.

3) Feature re-projection and matching: The union of all features in the set of matched mapped images (through pose-based matching and the co-visibility) are reprojected onto the current image. Re-projected features are matched with current features by binary descriptor matching; projected features considered for matching are limited to a small radius (30 pixels) around the current feature.

4) Outlier rejection We apply 2+1 pt RANSAC on the set of matched 2D-3D correspondences to reject outliers. If less than 13 correspondences remain, we classify all feature measurements as outliers.

C. Additional outlier rejection

While RANSAC-based approaches generally remove the majority of any 2D-3D outliers, we further improve our system's reliability by applying a Mahalanobis distance test on a per-feature basis.

VI. Experiment Results

All experimental results are obtained on a Google Project Tango tablet, which is equipped with a fisheye, global-shutter, grayscale camera, a MEMS quality IMU, a quad-core, 2.3 GHz ARM Cortex-A15 CPU, and 4 GB of RAM.

A. Data collection and ground truth

To generate experimental results we collected two datasets: the first (DS1), which comprises of 9,305 images and is approximately 320 m long, is used to generate the map of the area of interest, while the second (DS2) (2,215 images and 75 m long) is the dataset which the tested estimators run on. All maps and sub-maps of DS1 are generated using the CM method of [8]. In addition to each map's estimate and Cholesky factor, we save FREAK binary descriptors [2] and a VT [21], which indexes all mapped images. To acquire ground truth for the trajectory in DS2, we generate a BLS estimate, where in addition to all camera and IMU measurements, we supply the BLS estimator with absolute pose measurements from a VICON system.

B. Results

Note that in the trajectory of DS2, the user moves within the VICON-room, visiting the same scene several times. Such a trajectory will emphasize the detrimental effects of an inconsistent estimator which does not track device-map cross-correlations (as the device estimate becomes strongly correlated with the map when viewing the same features multiple times). This inconsistency is illustrated in Fig. 4, which shows the error and 3σ bounds for the inflated measurement noise model (σ = 7 pixels), and the sC-SKF estimator employing 4 sub-maps. Furthermore, the pose RMSE for each method of map-based updates can be found in Table II, where, as expected, the C-SKF and sC-KF outperform inconsistent methods.

The storage requirements of the map's uncertainty information is the main limitation of the SKF addressed by the C-SKF. The sizes of the Cholesky factors used for the map of DS1 and four sub-maps are available in Table I, as well as the corresponding size of the map's covariance. As expected, the sparse Cholesky factor requires much less disk space than the dense covariance.

As mentioned earlier, the motivation for partitioning the map into sub-maps is to reduce the time required to compute $\mathbf{J}$ [see (38)]. We supply the average time for the back-solve required to compute $\mathbf{J}$ on a Project Tango tablet, for different map sizes in Table III. For large maps, it becomes impossible

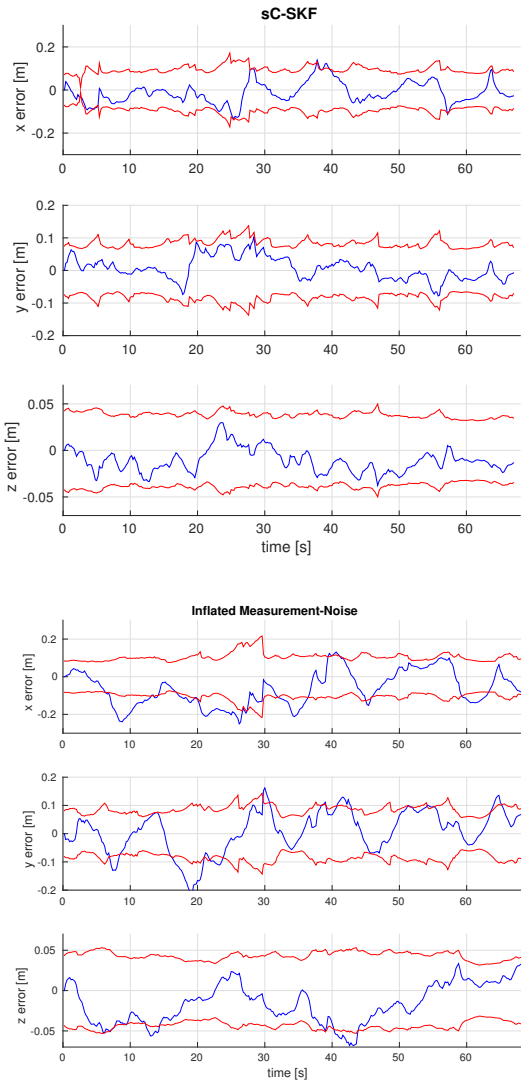

Fig. 4: The error and 3σ bounds for the sC-SKF with 4 sub-maps and map-based updates with inflated measurement noise.

Map ID	1/4	2/4	3/4	4/4	Full map
Num Feat.	3,926	3,590	4,671	3,384	12,164
Map Dim	19,683	18,590	26,008	16,952	63,692
Size ($\mathbf{G}$)	40 MB	43 MB	52 MB	39 MB	457 MB
Size ($\mathbf{P}_{MM}$)	1.5 GB	1.4 GB	2.7 GB	1.1 GB	16.2 GB

TABLE I: Sizes of sub-maps and full maps and associated memory requirements of storing their Cholesky factor, $\mathbf{G}$, and covariance $\mathbf{P}_{MM}$.

to perform real-time map-based updates with the C-SKF. On the other hand, by reducing the map size (i.e., increase number of sub-maps) and employing the sC-SKF, we significantly reduce the computation requirements.

Finally, we present the times of the most computationally-demanding-components of the sC-SKF pipeline with four sub-maps in Table IV. Specifically, the components timed

Method	Position RMSE (cm)
C-SKF, Single Map	6.2
sC-SKF, Four Sub-maps	8.4
Inflated Measurement Noise	11.5
No Map-based Updates	14.4

TABLE II: Position RMSE

Number of Features in Map	Time to Compute $\mathbf{J}$
3,926	6.7 ms
6,341	24.5 ms
12,164	233.3 ms

TABLE III: Average time to compute $\mathbf{J}$ [see (38)] for a single feature measurement in various maps.

are the 2D-3D correspondence detection pipeline, the map-based update, the local feature tracking pipeline (Harris and KLT), and the MSCKF update. We also provide the time for mapped-feature updates using the perfect-map assumption (inflate noise) for comparison. Our MSCKF runs at approximately 6Hz (depending on the user's motion). By summing the MSCKF update time and feature-tracking time listed, our local-measurements estimation requires on average 324 ms of CPU time per second, which leaves 676 ms to be devoted to map-based updates. The sC-SKF map-based update pipeline (2D-3D pose-based correspondence generation and map-based-update) takes 219 ms per iteration. Therefore, we can generally apply map-based updates while maintaining real-time operation. If the application or device has stricter processing requirements, we have the option to increase the number of sub-maps (decreasing the total information attributed to the map).

VII. Conclusion

In this paper, we focused on the problem of performing approximate, but consistent map-based localization. Specifically, and motivated from the linear (in the map's size) processing cost, but quadratic memory requirements of the Schmidt-Kalman filter (SKF) when applied to map-based localization, we introduced the Cholesky (C)-SKF, which uses the map's Cholesky factor to model the information (and thus uncertainty) in the prior map. By doing so, and given the sparsity of the Cholesky factor, the C-SKF has only linear, in the map's size, memory requirements. Moreover, its equations are factored in such a form so as to avoid inverting

Item	Mean Time (ms)
Mapped-feature Update (4 sub-maps)	183
Mapped-feature Update (inflate noise)	7
Correspondence Detection (pose-less)	52
Correspondence Detection (pose-assisted)	36
Harris + KLT Tracking	20
Propagation + Local-feature Update	34

TABLE IV: Mean times for the steps of the estimator pipeline.

the Cholesky factor of the map's Hessian matrix. Despite, however, the gains in efficiency, the processing cost of the C-SKF may grow more than linearly in the map's size. In order to bound its processing cost, we introduced a relaxation, termed the sC-SKF, which uses the sub-maps obtained by partitioning the original map, with minimal loss in accuracy. Lastly, the computational requirements of the proposed C-SKF and sC-SKF were assessed using a Project Tango tablet, while we demonstrated their superior performance against other approximate, but inconsistent, map-based approaches through real-world experiments using high accuracy ground truth.

APPENDIX A
PROOF OF SUB-MAP CONSISTENCY

We seek to prove Theorem 1 in Section IV-E.

Proof: As described in [8], for the case of two sub-maps, the KKT matrix, $\mathscr{K}$, resulting from the constrained optimization problem is:

$$\mathscr{K} = \begin{bmatrix} \mathscr{H}_1 & \mathbf{0} & \mathbf{A}_1^T & \mathbf{0} \\ \mathbf{0} & \mathscr{H}_2 & \mathbf{A}_2^T & \mathbf{0} \\ \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{0} & \mathbf{B} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}^T & \mathbf{0} \end{bmatrix}$$

where $\mathscr{H}_i$ is the Hessian of sub-map i, $\mathbf{A}_i$ corresponds to features common to the two sub-maps, and $\mathbf{B}$ is a tall matrix whose columns correspond to the 4 d.o.f transformation between the two sub-maps.

Computing the covariance of the sub-maps in the CM result requires computing the 2×2 block sub-matrix of the inverse of $\mathscr{K}$. i.e.,

$$\begin{aligned} \mathbf{P} &= (\mathscr{K}^{-1})_{1:2,1:2} \\ &= \left\{ \begin{bmatrix} \mathscr{H}_1 & \mathbf{0} \\ \mathbf{0} & \mathscr{H}_2 \end{bmatrix} - \begin{bmatrix} \mathbf{A}_1^T & \mathbf{0} \\ \mathbf{A}_2^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \right\}^{-1} \end{aligned}$$

Employing the matrix inversion lemma yields:

$$\mathbf{P} = \begin{bmatrix} \mathscr{H}_1^{-1} & \mathbf{0} \\ \mathbf{0} & \mathscr{H}_2^{-1} \end{bmatrix} - \begin{bmatrix} \mathscr{H}_1^{-1} & \mathbf{0} \\ \mathbf{0} & \mathscr{H}_2^{-1} \end{bmatrix} \mathbf{M} \begin{bmatrix} \mathscr{H}_1^{-1} & \mathbf{0} \\ \mathbf{0} & \mathscr{H}_2^{-1} \end{bmatrix}$$

where

$$\mathbf{M} = \begin{bmatrix} \mathbf{A}_1^T & \mathbf{0} \\ \mathbf{A}_2^T & \mathbf{0} \end{bmatrix} \mathbf{W} \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{52}$$

and

$$\begin{aligned} \mathbf{W}^{-1} &= \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathscr{H}_1^{-1} & \mathbf{0} \\ \mathbf{0} & \mathscr{H}_2^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{A}_1^T & \mathbf{0} \\ \mathbf{A}_2^T & \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{0} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{\Theta} & -\mathbf{B} \\ -\mathbf{B}^T & \mathbf{0} \end{bmatrix} \end{aligned}$$

with

$$\boldsymbol{\Theta} \triangleq \mathbf{A}_1 \mathscr{H}_1^{-1} \mathbf{A}_1^T + \mathbf{A}_2 \mathscr{H}_2^{-1} \mathbf{A}_2^T$$

On the other hand, if we treat each of the sub-maps as being independent (i.e., we ignore the constraints imposed by common features) but we compute their Hessians using the CM result, their covariances are:

$$\bar{\mathbf{P}} = \begin{bmatrix} \mathscr{H}_1^{-1} & \mathbf{0} \\ \mathbf{0} & \mathscr{H}_2^{-1} \end{bmatrix} \tag{53}$$

In order to prove $\mathbf{P} \preceq \bar{\mathbf{P}}$, it suffices to show $\mathbf{M}$ in (52) is symmetric positive semi-definite (PSD). We start by computing

$$\mathbf{W} = \begin{bmatrix} \mathbf{X} & \mathbf{Y} \\ \mathbf{Y}^T & \mathbf{Z} \end{bmatrix} \tag{54}$$

where:

$$\mathbf{X} = \boldsymbol{\Theta}^{-1} - \boldsymbol{\Theta}^{-1}\mathbf{B}\left(\mathbf{B}^T\boldsymbol{\Theta}^{-1}\mathbf{B}\right)^{-1}\mathbf{B}^T\boldsymbol{\Theta}^{-1} \tag{55}$$

$$\mathbf{Y} = -\boldsymbol{\Theta}^{-1}\mathbf{B}\left(\mathbf{B}^T\boldsymbol{\Theta}^{-1}\mathbf{B}\right)^{-1} \tag{56}$$

$$\mathbf{Z} = -\left(\mathbf{B}^T\boldsymbol{\Theta}^{-1}\mathbf{B}\right)^{-1} \tag{57}$$

Substituting (54) in (52) yields

$$\mathbf{M} = \begin{bmatrix} \mathbf{A}_1^T \\ \mathbf{A}_2^T \end{bmatrix} \mathbf{X} \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \end{bmatrix} \tag{58}$$

Note that $\mathbf{X}$ in (55) is PSD since it is the (1,1) Schur complement of the PSD matrix $\begin{bmatrix} \mathbf{I} \\ \mathbf{B}^T \end{bmatrix} \boldsymbol{\Theta}^{-1} \begin{bmatrix} \mathbf{I} & \mathbf{B} \end{bmatrix}$. Thus $\mathbf{M}$ is also PSD. ■

REFERENCES

[1] Motilal Agrawal and Kurt Konolige. Frameslam: From bundle adjustment to real-time visual mapping. *IEEE Trans. on Robotics*, 24(5):1066–1077, October 2008.

[2] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. FREAK: Fast retina keypoint. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 510–517, College Park, MD, June 16–21 2012.

[3] Clemens Arth, Daniel Wagner, Manfred Klopschitz, Arnold Irschara, and Dieter Schmalstieg. Wide area localization on mobile phones. In *8th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 73–82, Orlando, FL, USA, October 19–22 2009.

[4] Averil Burton Chatfield. *Fundamentals of High Accuracy Inertial Navigation*, volume 174. American Institute of Aeronautics and Astronautics, 1997.

[5] Siddharth Choudhary, Luca Carlone, Henrik I Christensen, and Frank Dellaert. Exactly sparse memory efficient slam using the multi-block alternating direction method of multipliers. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Hamburg, Germany.

[6] Jose E Guivant and Eduardo Mario Nebot. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Trans. on Robotics and Automation*, 17(3):242–257, 2001.

[7] Chao Guo, Dimitrios Kottas, Ryan DuToit, Ahmed Ahmed, Ruipeng Li, and Stergios Roumeliotis. Efficient visual-inertial navigation using a rolling-shutter camera

with inaccurate timestamps. In *Proceedings of Robotics: Science and Systems*, Berkeley, CA, USA, July 12–16 2014.

[8] Chao X. Guo, Kourosh Sartipi, Ryan DuToit, Georgios Georgiou, Ruipeng Li, John O'Leary, Esha D. Nerurkar, Joel A. Hesch, and Stergios I. Roumeliotis. Large-scale cooperative 3D visual-inertial mapping in a Manhattan world. In *Proc. of the IEEE International Conference on Robotics and Automation*, Stockholm, Sweden, May 16–21 2016. URL http://mars.cs.umn.edu/papers/CM_line.pdf.

[9] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Proc. of the Alvey Vision Conference*, pages 147–151, Manchester, UK, August 31 – September 2 1988.

[10] Joel A. Hesch, Dimitrios G. Kottas, Sean L. Bowman, and Stergios I. Roumeliotis. Consistency analysis and improvement of vision-aided inertial navigation. *IEEE Trans. on Robotics*, 30(1):158–176, February 2014.

[11] Simon J Julier. A sparse weight Kalman filter approach to simultaneous localisation and map building. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 1251–1256, Maui, HI, USA, October 29 – November 3 2001.

[12] Georg Klein and David Murray. Parallel tracking and mapping for small AR workspaces. In *6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 225–234, Nara, Japan, November 13–16 2007.

[13] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Closed-form solutions to minimal absolute pose problems with known vertical direction. In *Proc. of the Asian Conference on Computer Vision*, pages 216–229, Queenstown, New Zealand, November 8–12 2011.

[14] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of the International Joint Conference on Artificaial Intelligence*, pages 674–679, Vancouver, British Columbia, August 24–28 1981.

[15] Simon Lynen, Torsten Sattler, Michael Bosse, Joel Hesch, Marc Pollefeys, and Roland Siegwart. Get out of my lab: Large-scale, real-time visual-inertial localization. In *Proc. of Robotics: Science and Systems Conference*, Rome, Italy, July 13–17 2015.

[16] Sven Middelberg, Torsten Sattler, Ole Untzelmann, and Leif Kobbelt. Scalable 6-dof localization on mobile devices. In *Proc. of the European Conference on Computer Vision*, pages 649–663, Zurich, Switzerland, September 6–12 2014.

[17] Faraz M. Mirzaei and Stergios I. Roumeliotis. A Kalman filter-based algorithm for IMU-camera calibration: Observability analysis and performance evaluation. *IEEE Trans. on Robotics*, 24(5):1143–1156, October 2008.

[18] Anastasios I. Mourikis, Nikolas Trawny, Stergios I. Roumeliotis, Andrew E. Johson, Adnan Ansar, and Larry Matthies. Vision-aided inertial navigation for spacecraft entry, descent, and landing. *IEEE Trans. on Robotics*, 25 (2):264–280, April 2009.

[19] Oleg Naroditsky, Xun S. Zhou, Jean Gallier, Stergios I. Roumeliotis, and Kostas Daniilidis. Two efficient solutions for visual odometry using directional correspondence. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(4):818–824, April 2012.

[20] Kai Ni, Drew Steedly, and Frank Dellaert. Tectonic sam: Exact, out-of-core, submap-based slam. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 1678–1685, Rome, Italy, April 10–14 2007. IEEE.

[21] David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2161–2168, New York, NY, June 17–22 2006.

[22] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *Proc. of the IEEE International Conference on Computer Vision*, pages 2564–2571, Barcelona, Spain, November 6–13 2011.

[23] Stanley F. Schmidt. Applications of state space methods to navigation problems. *in C. T. Leondes, Editor, Advanced Control Systems*, 3:293–340, 1966.

[24] Dan Simon. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[25] Bill Triggs, Philip McLauchlan, Richard Hartley, and A. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Vision Algorithms: Theory and Practice*, pages 298–375. Springer–Verlag, 2000.

[26] Jordi Ventura, Clemens Arth, Gerhard Reitmayr, and Dieter Schmalstieg. Global localization from monocular SLAM on a mobile phone. *IEEE Trans. on Visualization and Computer Graphics*, 20(4):531–539, April 2014.