Toward Consistent Drift-free Visual Inertial Localization on Keyframe Based Map

Zhuqing Zhang¹, Yanmei Jiao¹, Shoudong Huang², Yue Wang¹, Rong Xiong¹

Abstract—Global localization is essential for robots to perform further tasks like navigation. In this paper, we propose a new framework to perform global localization based on a filter-based visual-inertial odometry framework MSCKF. To reduce the computation and memory consumption, we only maintain the keyframe poses of the map and employ Schmidt-EKF to update the state. This global localization framework is shown to be able to maintain the consistency of the state estimator. Furthermore, we introduce a re-linearization mechanism during the updating phase. This mechanism could ease the linearization error of observation function to make the state estimation more precise. The experiments show that this mechanism is crucial for large and challenging scenes. Simulations and experiments demonstrate the effectiveness and consistency of our global localization framework.

I. INTRODUCTION AND RELATED WORKS

Autonomous mobile robot is a hot topic in contemporary research and development. To realize it, it is crucial to localize the robot in a global coordinate system. A good way to do that is employing simultaneous localization and mapping (SLAM). However, most of SLAM related works [1]–[5] estimate robot state in a local coordinate system and the estimator would have drift. Global information could be introduced to solve this problem and therefore the robot system would be localized in a global frame. A commonly used global information for localization is global position system (GPS) [6]–[8]. However, GPS information is usually noisy and not always available. The pre-built visual map is another source of global information, which is what we consider in this paper. For visual-based localization system, the map usually contains keyframes’ poses and related 2D or/and 3D features. When there are matches between the map and the online input images, the global information could be fused into the state estimator, and therefore remove the drift and increase the positional accuracy.

There are lots of literatures performing global localization with pre-built visual maps [6], [9]–[12]. However, how to use the information of the map is an open question. For a large scene, the map could contain hundreds of keyframes and tens of thousands of points, which is quite large in volume. Besides, the map information is actually not perfect and should have uncertainty information (covariance or information matrix), which makes the data much larger. To make the global localization system run in real-time, some trade-offs need to be made with map information.

A direct way of using a visual map is employing all the map information (m keyframes’ poses, n landmarks, and their covariance or information matrices; m << n in general visual SLAM scenarios). However, since the computational complexity of updating covariance is $O((m + n)^2)$, maintaining all the map information in the state vector would be computational expensive. To solve this problem, [10] proposes a system based on Schmidt-EKF [13]. With the help of Schmidt-EKF, updating covariance would be less expensive ($O(n)$). However, [10] still stores the whole map in the state vector, which makes the state dimension quite high and needs a large memory consumption ($O(n^2)$) to record information matrices. There are also some works [6], [11], [12], [16] model maps as a set of keyframes. Maplab [11] employs the ROVIO [14], [15] to realize local positioning, and combines the technique in [9] to obtain the global state. Its map is a set of pose-graph consisting of keyframes and landmarks (memory consumption $O(m + n)$). In this framework, to improve computational efficiency, it does not maintain the map information in the state (computational complexity is constant), so that the correlation between current state and map information cannot be maintained, and therefore, the system would suffer from inconsistency. Besides, this system ignores the uncertainty of the map landmarks, and utilizes them in the observation function as fixed values, which again makes the system inconsistent. VINS-fusion [5], [6], [17], [18] builds the map in a similar way to Maplab [11], however it omits the landmarks information (memory consumption $O(m)$). Its global observation function is constructed by current local 3D landmarks and matched 2D features in the map. Nevertheless, this framework cannot take the covariance of the map information into consideration, and results in inconsistency. Besides, in VINS-Fusion, local 3D landmarks are estimated online from the extracted 2D features in query images (not from the pre-built map), which is coupled with the local odometry (or state estimator). To guarantee the state estimator runs in real time, the algorithm for extracting 2D keypoints and computing descriptors needs to be lightweight (e.g. FAST [19] + Brief [20]). These kinds of features, however, are not robust for matching with map features. Furthermore, the nonlinear optimization strategy employed in this framework is also time-consuming.

In this paper, we propose a filter-based global localization framework with consideration of the balance between accuracy, consistency, computation speed and memory space. To be specific, the pre-built map is made up of m keyframes...
II. SYSTEM STATE AND OBSERVATION FUNCTION

As shown in Fig. [1] the whole system consists of three parts, feature matching with a given map, local positioning and global positioning.

A. Local Positioning Based on MSCKF

Local positioning in our work is actually a standard MSCKF [3], and this part follows the implementation of Open-VINS [4]. As is shown in Fig. [2] local positioning is estimating the current state of the robot in local reference system L. Without considering the map information, we maintain the following state vector at time step $k$:

$$x_k = \begin{bmatrix} x_l^T & x_C^T \end{bmatrix}^T,$$

where $x_l$ is the current state of the robot. $I$ refers to the IMU reference system. Here, we assume the body reference system is identical to the IMU reference system. We define $x_l$ as

$$x_l = \begin{bmatrix} I_k q_L^T & I_k p_L^T & I_k v_L^T & b_g^T & b_a^T \end{bmatrix}^T,$$

where $I_k q_L$ is an unit quaternion variable, defined by JPL [24]. Its corresponding rotation matrix $I_k R_L$ rotates a 3D vector from the local coordinate system $L$ into the IMU coordinate system $I$. $I_k p_L$ is the position of the body in the local coordinate system at time step $k$. $I_k v_L$ is the velocity of the body in the local coordinate system. $b_g$ and $b_a$ are the gyroscope and accelerometer bias.

$$x_C = \begin{bmatrix} q_L^T & p_L^T & \ldots & q_{L-k}^T & p_{L-k}^T \end{bmatrix}^T,$$

which consists of the clones of the $N$ newest historical poses. $x_C$ is the so-called sliding window.

1) State Propagation: When we receive IMU data at time step $k$, we could employ the IMU kinematics equations to predict the body state at time step $k+1$:

$$x_{k+1|k} = f(x_k, a_{mk} - n_{mk}, \omega_{mk} - n_{wk}),$$

where $f$ represents the IMU kinematics equations, $a_{mk}$ and $\omega_{mk}$ are the observations (linear acceleration and angular velocity) from the IMU. $n_{mk}$ and $n_{wk}$ are the corresponding zero-mean Gaussian white noise of the IMU observations. With this function, we could propagate the state covariance by

$$P_{k+1|k} = \Phi_k P_{k|k} \Phi_k^T + G_k Q_k G_k^T,$$

where $P_{k|k}$ and $Q_k$ are the system state covariance and the noise covariance respectively. $\Phi_k$ and $G_k$ are the Jacobians of $f$ with respect to the system state and the noise respectively. For detailed derivation, please refer to [3], [4].

2) Observation Function: In the framework of MSCKF, before fusing feature observations by the observation function, we could get a list of features $\{f_i\}$ and related cloned poses $\{T_C\}$. For each feature $f_i$ observed by $T_{Cj}$, we have the following observation function:

$$z_{fi} = h(T_{Cj}, f_i) + n_{fi},$$

where $z_{fi}$ is the 2D observation of $f_i$ in the image. $f_i$ is the 3D coordinate of $f_i$. By applying first-order Taylor expansion to the above observation function at the current estimated state, we could get the following functions:

$$z_{fi} = h(z_{fi}) + H_{fi}[\hat{x} - x_{n}],$$

where $\hat{z}_{fi}$ represents the estimated values and $z_{fi}$ represents the errors between true values and estimated values. $H_{x}$ and $H_{f}$ are the Jacobians of $h$ with respect to the estimated state and the observed feature respectively. As we do not maintain features in the state vector, $\hat{f}_i$ is marginalized by projecting $r_{fi}$ into the left null space of $H_{f}$, $N$:

$$N^T r = N^T H_{x} \hat{x} + N^T H_{f} \hat{f} + N^T n_f,$$

$$r_{fi} = H_{x} \hat{x} + n_{fi},$$

where $r_{fi}$ is the 3D observation of $f_i$. $H_{x}$ and $H_{f}$ are the Jacobians of $h$ with respect to the estimated state and the observed feature respectively. We do not maintain features in the state vector, $\hat{f}_i$ is marginalized by projecting $r_{fi}$ into the left null space of $H_{f}$, $N$:

$$N^T r_{fi} = N^T H_{x} \hat{x} + N^T H_{f} \hat{f} + N^T n_{fi},$$

$$r_{fi} = H_{x} \hat{x} + n_{fi},$$

where $r_{fi}$ is the 3D observation of $f_i$. $H_{x}$ and $H_{f}$ are the Jacobians of $h$ with respect to the estimated state and the observed feature respectively. We do not maintain features in the state vector, $\hat{f}_i$ is marginalized by projecting $r_{fi}$ into the left null space of $H_{f}$, $N$:

$$N^T r_{fi} = N^T H_{x} \hat{x} + N^T H_{f} \hat{f} + N^T n_{fi},$$

$$r_{fi} = H_{x} \hat{x} + n_{fi},$$

where $r_{fi}$ is the 3D observation of $f_i$. $H_{x}$ and $H_{f}$ are the Jacobians of $h$ with respect to the estimated state and the observed feature respectively. We do not maintain features in the state vector, $\hat{f}_i$ is marginalized by projecting $r_{fi}$ into the left null space of $H_{f}$, $N$:

$$N^T r_{fi} = N^T H_{x} \hat{x} + N^T H_{f} \hat{f} + N^T n_{fi},$$

$$r_{fi} = H_{x} \hat{x} + n_{fi},$$

where $r_{fi}$ is the 3D observation of $f_i$. $H_{x}$ and $H_{f}$ are the Jacobians of $h$ with respect to the estimated state and the observed feature respectively. We do not maintain features in the state vector, $\hat{f}_i$ is marginalized by projecting $r_{fi}$ into the left null space of $H_{f}$, $N$:

$$N^T r_{fi} = N^T H_{x} \hat{x} + N^T H_{f} \hat{f} + N^T n_{fi},$$

$$r_{fi} = H_{x} \hat{x} + n_{fi},$$

where $r_{fi}$ is the 3D observation of $f_i$. $H_{x}$ and $H_{f}$ are the Jacobians of $h$ with respect to the estimated state and the observed feature respectively. We do not maintain features in the state vector, $\hat{f}_i$ is marginalized by projecting $r_{fi}$ into the left null space of $H_{f}$, $N$:

$$N^T r_{fi} = N^T H_{x} \hat{x} + N^T H_{f} \hat{f} + N^T n_{fi},$$

$$r_{fi} = H_{x} \hat{x} + n_{fi},$$

where $r_{fi}$ is the 3D observation of $f_i$. $H_{x}$ and $H_{f}$ are the Jacobians of $h$ with respect to the estimated state and the observed feature respectively. We do not maintain features in the state vector, $\hat{f}_i$ is marginalized by projecting $r_{fi}$ into the left null space of $H_{f}$, $N$:

$$N^T r_{fi} = N^T H_{x} \hat{x} + N^T H_{f} \hat{f} + N^T n_{fi},$$

$$r_{fi} = H_{x} \hat{x} + n_{fi},$$

where $r_{fi}$ is the 3D observation of $f_i$. $H_{x}$ and $H_{f}$ are the Jacobians of $h$ with respect to the estimated state and the observed feature respectively. We do not maintain features in the state vector, $\hat{f}_i$ is marginalized by projecting $r_{fi}$ into the left null space of $H_{f}$, $N$:

$$N^T r_{fi} = N^T H_{x} \hat{x} + N^T H_{f} \hat{f} + N^T n_{fi},$$

$$r_{fi} = H_{x} \hat{x} + n_{fi},$$

where $r_{fi}$ is the 3D observation of $f_i$. $H_{x}$ and $H_{f}$ are the Jacobians of $h$ with respect to the estimated state and the observed feature respectively. We do not maintain features in the state vector, $\hat{f}_i$ is marginalized by projecting $r_{fi}$ into the left null space of $H_{f}$, $N$:

$$N^T r_{fi} = N^T H_{x} \hat{x} + N^T H_{f} \hat{f} + N^T n_{fi},$$

$$r_{fi} = H_{x} \hat{x} + n_{fi},$$

where $r_{fi}$ is the 3D observation of $f_i$. $H_{x}$ and $H_{f}$ are the Jacobians of $h$ with respect to the estimated state and the observed feature respectively. We do not maintain features in the state vector, $\hat{f}_i$ is marginalized by projecting $r_{fi}$ into the left null space of $H_{f}$, $N$:

$$N^T r_{fi} = N^T H_{x} \hat{x} + N^T H_{f} \hat{f} + N^T n_{fi},$$

$$r_{fi} = H_{x} \hat{x} + n_{fi},$$

where $r_{fi}$ is the 3D observation of $f_i$. $H_{x}$ and $H_{f}$ are the Jacobians of $h$ with respect to the estimated state and the observed feature respectively. We do not maintain features in the state vector, $\hat{f}_i$ is marginalized by projecting $r_{fi}$ into the left null space of $H_{f}$, $N$:

$$N^T r_{fi} = N^T H_{x} \hat{x} + N^T H_{f} \hat{f} + N^T n_{fi},$$

$$r_{fi} = H_{x} \hat{x} + n_{fi},$$

where $r_{fi}$ is the 3D observation of $f_i$. $H_{x}$ and $H_{f}$ are the Jacobians of $h$ with respect to the estimated state and the observed feature respectively. We do not maintain features in the state vector, $\hat{f}_i$ is marginalized by projecting $r_{fi}$ into the left null space of $H_{f}$, $N$:

$$N^T r_{fi} = N^T H_{x} \hat{x} + N^T H_{f} \hat{f} + N^T n_{fi},$$

$$r_{fi} = H_{x} \hat{x} + n_{fi},
EPnP [23] we could get the relative pose $^{kf_1}T_{C_k}$ between the keyframe and the current frame. This value, combined with the matched keyframe pose $^G{T_{kf_1}}$ and the current frame pose $^L{T_{C_k}}$, would be used to initialize $^G{T_L}$. After initialization, $^G{T_L} = x_i$ would be added into the state vector:

$$x_k = \begin{bmatrix} x_i^T & x_C^T & x_f^T & x_{KF}^T \end{bmatrix}^T.$$  (11)

Besides, the covariance of the state would also be augmented. As the initial estimation of $x_i$ may be inaccurate, its covariance could be assigned with a big value. It should be noted that every time we have matches from the map, the related keyframes’ poses should be added into the state vector:

$$x_k = \begin{bmatrix} x_i^T & x_C^T & x_f^T & x_{KF}^T \end{bmatrix}^T,$$  (12)

$$x_{KF} = \begin{bmatrix} q_{kf_1}^T & q_{pf_1}^T & \ldots & q_{kf_{M_2}}^T & p_{kf_{M_3}}^T \end{bmatrix}^T.$$  (13)

Also, the covariance of keyframes needs to be considered. For the sake of simplicity, we divide the state into two parts: the active part $x_A = \begin{bmatrix} x_i^T & x_C^T & x_f^T \end{bmatrix}^T$, and the nuisance part $x_N = x_{KF}$, so that

$$x_k = \begin{bmatrix} x_A^T & x_N^T \end{bmatrix}^T.$$  (14)

This kind of partition is very useful for Schmidt-EKF to be introduced in Sec. III-A.

2) Observation Function: As is shown in Fig. 2, supposing the current image matches with multiple frames (for example, $kf_1$ and $kf_2$), for each 3D landmark (for example, $^{kf_1}p_{f_1}$), we could project it from the anchored keyframe ($^{kf_1}f_1$) coordinate system into the 2D current image coordinate system by a nonlinear observation function $^2g$:

$$^1z_f = ^1g(^G{T_{kf_1}}, ^G{T_L}, ^L{T_{C_k}}, ^{kf_1}p_{f_1}) + ^1n_f,$$  (15)

where $^1z_f$ is the 2D pixel coordinate in the current image. Linearizing above function at the estimated values, just like [7], we could get

$$^1r_f = ^1H_{x_{A_k}}x_{A_k} + ^1H_{x_{N_k}}x_{N_k} + ^1H_{^{kf_1}p_{f_1}}^{kf_1}p_{f_1} + ^1n_f.$$  (16)

The subscript $A$ and $N$ represent the active part and the nuisance part of the state vector respectively (c.f. [14]).

Besides, the 3D landmark $^{kf_1}p_{f_1}$ could also be directly reprojected into the anchored keyframe $^{kf_1}f_1$ by a projection function $^3g$:

$$^2z_f = ^2g(^{kf_1}p_{f_1}) + ^2n_f,$$  (17)

$$^2r_f = ^2H_{^{kf_1}p_{f_1}}^{kf_1}p_{f_1} + ^2n_f.$$  (18)

Furthermore, if the 3D landmark $^{kf_1}p_{f_1}$ can be observed by the other matched keyframes(s), like $kf_2$ in Fig. 2, there is another reprojection function $^4g$ which reprojects $^{kf_1}p_{f_1}$ into the $^{kf_2}$ image plane:

$$^3z_f = ^3g(^{G{T_{kf_1}}, ^G{T_{kf_2}}, ^{kf_1}p_{f_1}}) + ^3n_f,$$  (19)

$$^3r_f = ^3H_{^{kf_1}p_{f_1}}^{kf_1}p_{f_1} + ^3n_f.$$  (20)

By stacking (15), (19) and (20) we could get the 6-dimension observation for the 3D landmark $^{kf_1}p_{f_1}$.

In summary, whenever the current image matches with a single keyframe or multiple keyframes, for each 3D landmark $^{kf_1}p_{f_1}$ anchored in the keyframe $^{kf_j}f_j$, we could get the observation function with the following form:

$$^r_f = ^H_{x_{A_k}}x_{A_k} + ^H_{x_{N_k}}x_{N_k} + ^H_{^{kf_1}p_{f_1}}^{kf_1}p_{f_1} + ^n_f.$$  (21)

It is important to note that since we regard landmarks of the map as variables with uncertainties, the Jacobian matrix $^H_{^{kf_1}p_{f_1}}$ needs to be computed. Otherwise, the landmarks would be treated as constants, like [11], which would make the system inconsistent. However, different from [10], we do not maintain map landmarks in the state vector, so we need to marginalize $^{kf_1}p_{f_1}$. For each landmark, it has at least two measurements (in the current image, and the matched one or more keyframes), so the row number of $^H_{^{kf_1}p_{f_1}}$ is more than the dimension of the landmark. This fact guarantees that we could project $^r_f$ into the left null space of $^H_{^{kf_1}p_{f_1}}$, like [7]. In this way, the items related to the landmarks are
eliminated while taking the uncertainties of the landmarks into consideration:
\[ r^*_f = H^*_{Ak} \hat{x}_{Ak} + H^*_{Nk} \hat{x}_{Nk} + n^*_f \triangleq H^*_{Ak} \hat{x}_k + n^*_f. \]  
(22)

Note that (22) only considers one landmark. We need to stack (22) for all matched landmarks to get the final observation function:
\[ r^*_k = H^*_k \hat{x}_k + n^*, \]  
(23)

where \( r^*_k \), \( H^*_k \), and \( n^* \) are the stacking of many \( r^*_f \), \( H^*_{Ak} \) and \( n^*_f \). (23) will be used to do the state update following the Schmidt-EKF as stated in the next section.

III. STATE UPDATE WITH MAP INFORMATION

In this section, we would demonstrate how to update the state with map information efficiently while maintaining the consistency. Besides, we would point out the necessity of re-linearizing the observation function.

A. Global Update by Schmidt-EKF

When there are matches between the current query frame and the map keyframes, the global observations would be used to do the global update using Schmidt-EKF. As indicated by (14), the state vector could be divided into the active part and the nuisance part. Therefore, (23) could be written as:
\[ r^*_k = H^*_k \hat{x}_k + n^* = \begin{bmatrix} H^*_{Ak} & H^*_{Nk} \end{bmatrix} \begin{bmatrix} \hat{x}_{Ak} \\ \hat{x}_{Nk} \end{bmatrix} + n^*, \]  
(24)

where \( H^*_{Ak} \) represents the Jacobians of the related poses of the matched map keyframes; \( H^*_{Ak} \) represents the Jacobians of the current state \( x_f \) and the relative transformation \( x_i \). Based on the expression of (24), we would analyze the limitation of standard EKF and introduce the updating technique of Schmidt-EKF.

1) Limitation of Standard EKF: The state and the covariance update equations for EKF are as follows:
\[ \hat{x}_k = \hat{x}_{k|k-1} + K_r r^*_k; \]  
(25)
\[ P_k = P_{k|k-1} - KH^*_k P_{k|k-1}, \]  
(26)

where \( \hat{x}_{k|k-1} \) and \( P_{k|k-1} \) are the output of the propagation step (c.f. [4], [5]), and
\[ S = H^*_k P_{k|k-1} H^*_k + R, \]  
(27)
\[ K = \frac{H^*_k P_{k|k-1} H^*_k}{S}. \]  
(28)

\( R \) is the covariance of \( n^* \). Expanding (28),
\[ K = \begin{bmatrix} K_A \\ K_N \end{bmatrix} = \begin{bmatrix} P_{AAk|k-1} H^*_k + P_{ANk|k-1} H^*_N \\ P_{NAk|k-1} H^*_N + P_{NNk|k-1} H^*_N \end{bmatrix} S^{-1} \]  
(29)
and substituting (29) into (26), we could get:
\[ P_k = P_{k|k-1} - \begin{bmatrix} K_A S K^*_A & K_A H^*_k P_{ANk|k-1} \\ P_{ANk|k-1} H^*_k & K_{NNk|k-1} \end{bmatrix} \]  
(30)

From (30), we could see that in the standard EKF update step, when we need to update the covariance, the computation is dominated by \( K_N S K^*_N \), which is \( O(n^2) \) (suppose the dimension of \( x_N \) is \( n \)). As map information is added continuously while the robot is traveling, the size of the nuisance part \( x_N \) would grow over time. Especially for a large scene, the dimension of nuisance part can be thousands easily. In this situation, the filter-based system would suffer from the huge amount of computation and even could not run in real-time. Thanks to Schmidt-EKF, this problem could be solved elegantly.

2) Update by Schmidt-EKF: As analysed before, computing \( K_N S K^*_N \) is time-consuming. Therefore, in Schmidt-EKF, we just drop this part directly when we update the covariance. More specifically, we set \( K_N = 0 \), so that (30) can be rewritten as:
\[ P_k = P_{k|k-1} - \begin{bmatrix} K_A S K^*_A & K_A H^*_k P_{ANk|k-1} \\ P_{ANk|k-1} H^*_k & K_{NNk|k-1} \end{bmatrix} \]  
(31)
and (25) would be divided into two parts:
\[ \hat{x}_{Ak} = \hat{x}_{Ak|k-1} + K_A r^*_k, \]  
(32)
\[ \hat{x}_{Nk} = \hat{x}_{Nk|k-1}. \]  
(33)

It can be seen that the active part update is identical to the standard EKF while the nuisance part would not be updated. Comparing (30) and (31), it is easy to see that:
\[ P_{SKF} = P_{EKF} + \begin{bmatrix} 0 \\ K_N \end{bmatrix} S^{-1} \begin{bmatrix} 0 & K^*_N \end{bmatrix}, \]  
(33)
where \( P_{SKF} \) and \( P_{EKF} \) are updated covariance of Schmidt-EKF and standard EKF respectively. As \( S^{-1} \) is positive definite, \( \begin{bmatrix} 0 \\ K_N \end{bmatrix} S^{-1} \begin{bmatrix} 0 & K^*_N \end{bmatrix} \) must be positive semi-definite. Therefore, \( P_{SKF} - P_{EKF} \geq 0 \), and it is not overconfident about the estimated state, i.e. maintains the system consistency. Besides, different from [9], the cross-covariance of active state and map information is considered (c.f. [31]) whose computation complexity is \( O(n) \). Therefore, for the state update with global information, we just need to linearize the global observation function to the form of (24), and substitute (24) into (31)-(32).

B. Re-linearization

In the update step, as our observation function is nonlinear, we need to linearize it at the newest estimated point:
\[ z = h(x) + n \]
\[ = h(\hat{x}) + H_{x}(x - \hat{x}) + o(x^2) + n \]  
\[ \approx h(\hat{x}) + H_{x}(x - \hat{x}) + n, \]

where \( H_{x} \) is the Jacobians of the observation function with respect to \( x \) at the estimated point \( \hat{x} \). This approximation is only effective when the estimated value (linearization point) is close to the true value. When the error between the estimated value and the true value is big, the precision of this approximation is not satisfactory.

Revisited our global observation function, i.e. \( \{15\} \), the landmark in the matched keyframe reference system is reprojected into the current image plane through a big “loop” (c.f. Fig. 2). The observation of the landmark \( k_{f}, P_{k} \) in current image plane (black dot in 2) is derived from \( G_{k}T_{k}, G_{L}T_{L} \) and \( L_{C}C_{a} \). Therefore the error of related variables would be propagated to the final estimated measurement, which could be very big. Especially, if there is a long time lack of matching information, i.e. the system does not have global information to constrain the odometry for a long time, when the next global observation occurs, the drift of the odometry and the estimation error of \( G_{T}T_{L} \) can be big. If we still directly linearize the observation function, the precision of the first-order approximation would be poor.

A simple idea to reduce the first-order approximation error is performing the Taylor expansion at a more accurate point. To be specific, when we get a match between a map keyframe and the current image, we could compute a relatively accurate pose \( k_{f}T_{C_{a}} \) by solving EPnP as mentioned in Sec. II-B. Based on this more accurate \( k_{f}T_{C_{a}} \), we could recompute \( G_{T}T_{L} \) or \( L_{T}C_{a} \). The recomputed value would therefore be more accordant with the observation function \( \{15\} \). Then, we utilize this new value as the linearization point to compute the Jacobians of the observation function.

In our implementation, when the average of the matched features’ re-projection error is bigger than a threshold, the re-linearization mechanism would be triggered. We would re-compute the linearization point of \( G_{T}T_{L} \) or \( L_{T}C_{a} \) from the results of EPnP. After that, \( H_{x} \) in \( \{34\} \) is computed with the re-computed linearization point.

IV. EXPERIMENTAL RESULTS

In this section, we would first do some simulation experiments to demonstrate the performance of our proposed framework, and then validate our approach on EuRoC [25], KAIST [26], and our own data set YQ. These three data sets cover the platforms of UAV, car and UGV. Images sampled from these three data sets are shown in Fig. 3.

A. Simulation

1) Map Data Generation: We make some adaptations to the simulator of Open-VINS [4] to make it suitable for localization. The simulator needs to be fed a ground truth trajectory so that the system could generate image features and IMU information. Here, we feed the ground truth trajectory of our own data set (denoted as YQ3, see

Fig. 3. The images sampled from three data sets. The first row is the images from EuRoC [25] MH02-05; The second row is the images from KAIST [26] urban38-39; The third row is the images from YQ YQ1-4. It has a length of 1.282km. The map keyframes’ poses come from another trajectory of our own data set YQ1, which has the similar running path as YQ3. For the details of YQ data set, please refer to Sec. IV-B.3.

To test the consistency of our proposed system, we artificially add the noise to the ground truth of YQ1 to simulate that the map is not perfect. To be specific, the position of the ground truth is perturbed with the Gaussian white noise \( n_{p} \sim \mathcal{N}(0, 0.001I_{3 \times 3}) \), and the orientation is perturbed with \( n_{o} \sim \mathcal{N}(0, 0.00025I_{3 \times 3}) \). After the perturbation, the root-mean-squared error (RMSE) of the trajectory of YQ1 is 0.179m.

For the map matching information, we first randomly generate 3D landmarks, then we utilize the global observation function \( \{15\} \) to reproject 3D landmarks into the map keyframes and the current frame so that the 2D-2D matching features are obtained, after which we add Gaussian white noise to the 2D features. Finally, for each 3D landmark, we utilize the noisy 2D features and map keyframes’ perturbed poses to triangulate its 3D position (estimated 3D landmark).

2) Results and Analysis: In our simulation, there are four settings: the original Open-VINS [4]; our method (CS-MSCKF) with single matching frame (SM); our method (CS-MSCKF) with multiple matching frames (MM); our method that treats map poses and landmarks as constants (mapconst).

Fig. 4 demonstrates the trajectories derived from different situations. The blue one is the trajectory from the Open-VINS odometry. We could see that there is an apparent drift in the latter part of the trajectory, whereas our localization result (red one for multiple matching frames) fit nicely with the ground truth (black one). The green one is the result from the algorithm treating the map as perfect. The reason why the green one fit badly with the ground truth is that our map information is not absolutely accurate, and if we do not consider the uncertainty of the map, the estimator would highly rely on the inaccurate map information and therefore leading to a bad result.

Table 1 lists the RMSE of the four settings. Noting that the initial pose of odometry is set as the initial pose of the ground truth, the trajectory of the odometry is already in the global coordinate system, i.e. the initial poses of the odometry and the ground truth are naturally aligned, so that the RMSEs between the estimated trajectories and the ground truth can
be computed directly without the need of alignment.

From Table I, we could find that CS-MSCKF+MM is better than CS-MSCKF+SM. It is in accordance with our intuition, as multiple frames could provide more information. Besides, as is shown in Table I when the map is treated as perfect (CS-MSCKF+mapconst), the performance of the algorithm is very bad, which demonstrates the necessity of taking the map’s covariance into consideration.

To demonstrate the consistency of our proposed method, we draw the error of estimation with 3-σ bounds (c.f. Fig. 5). The figure is corresponding to the results of CS-MSCKF+MM ((a),(b)) and CS-MSCKF+mapconst ((c),(d)). From the figure, we could conclude that our proposed algorithm has good consistency while the algorithm that treats the map information as perfect has poor consistency.

### B. Experiments on Real World Data Sets

For real world data sets, the matching procedure is conducted as follows: We first utilize R2D2 [21] to extract new features on the current query frame and match with features in map keyframes. When there are enough matches, 3D-2D pairs (3D from map landmarks and 2D from current frame features) are fed into the robust pose solver in [22], after which the accurate and robust 3D-2D pairs are obtained.

1) EuRoC: In this part, the effectiveness of our proposed algorithm is validated on a popular data set EuRoC [25]. To be specific, we use the sequences of machine hall (MH) to do experiments. The sequence MH01 is used to build the map, and the other sequences of MH (MH02-MH05) are used to do localization. The map contains keyframes and 3D landmarks, where 3D landmarks are triangulated and refined across multiple adjacent map images and the keyframes’ poses are given by the ground truth (we assume the uncertainties of the ground truth are 1 cm and 1°. We make the comparison between the two settings (CS-MSCKF+SM/MM) of our proposed methods with the benchmark from Open-VINS [4] and VINS-Fusion [5, 6, 17, 18].

Open-VINS is a pure odometry, and the comparison with it could demonstrate that global localization could constrain the odometry’s drift error effectively. For VINS-Fusion, its global localization mode is used. To be specific, we firstly use its SLAM mode to generate a pose-graph (map information) based on MH01, and we modified the poses of the pose-graph node as the ground truth to make the comparison fair. As the localization needs to be performed in real-time, i.e. the estimated pose in time step k has to be computed in time step k, not the result of loop closure or global optimization thereafter, we record the global localization results from the combination of the odometry $^L T_k$ (without loop closure or global optimization) and the estimated $^G T_L$. The comparison of experimental results is plotted by box-plot in Fig. 6 where the RMSE of Open-VINS is computed based on the alignment of the initial pose and the rest are computed without alignments. In Fig. 6 values along Y-axis are projected into log space, and their specific values are also given. For the localization methods, their processing times are also given in Fig. 7. All the results are the average of three runs.

From Fig. 6 and Fig. 7, we could find that our proposed algorithms perform better than the VINS-Fusion in terms of both accuracy and efficiency. This is because our framework take the uncertainty of the map information into consideration. Besides, the effective and robust matching algorithm provides us better matching information.

2) Kaist: To show the advantage of our proposed re-linearization mechanism, we do experiments on Kaist data set [26]. This data set was recorded in urban environments
Table II

| Open-VINS [4] | VINS-Fusion [5, 6, 17, 18] | CS-MSCKF +SM | CS-MSCKF +SM +R | CS-MSCKF +MM | CS-MSCKF +MM +R |
|---------------|---------------------------|--------------|-----------------|--------------|-----------------|
|               |                           | 34.303       | 25.541          | 6.927        | 6.726          | 5.812          |

where the majority of the scene can be taken up by other moving vehicles, so this is a challenging data set. In the data set, there are two sequences Urban38 and Urban39 with high similarity. We employ Urban38 to build the map and localization algorithms are tested on Urban39 (the length of the trajectory is 10.67km). The covariance of each map keyframe’s pose is given as $0.1I_{6 \times 6}$ and the re-linearization threshold is set as 20 pixels. The comparison of different algorithms are given in Table II, and the trajectories derived from CS-MSCKF+SM and CS-MSCKF+SM+R are plotted in Fig. 8.

In this challenging scene, the VINS-Fusion failed to perform global localization due to the significant drifts of the odometry. From the results, we could find that for the case of single matching frame, there is a tremendous promotion in accuracy when we employ the re-linearization mechanism. Fig. 8 could illustrate this phenomenon to some extent.

In Fig. 8, the pair of stars represents the two contiguous matches (for example, the $i^{th}$ match in red and the $(i+1)^{th}$ match in blue), and so does the pair of diamonds. The motion direction of the car from the red mark to the blue mark is shown by black arrows in the picture. We could find that the distance between the two stars or the two diamonds is far, which means the accuracy of the localization highly relies on the performance of the odometry. Unfortunately, the odometry inevitably introduces drifts and the estimated values of the state variables are not accurate enough. These two sources of error would make the first-order Taylor approximation of the observation function not good, as analyzed in Sec. [II-B]. Therefore, we introduce a re-linearization mechanism to make the first-order Taylor approximation more accurate. The effectiveness of this mechanism is shown in Fig. 8. We could find that for the CS-MSCKF+SM, the trajectory around the blue star (c.f. the area of the green dotted circle) hardly changes when there is a match from the map, i.e. since the approximated observation function is not accurate, the single matching frame can hardly correct the trajectory. On the contrary, the trajectory from CS-MSCKF+SM+R shows an obvious sign of being corrected by the single matching frame information. Besides, as shown in the bottom part of Fig. 8, the estimated $z$ value has been corrected distinctly with the help of re-linearization mechanism while the estimated value from CS-MSCKF+SM changes little (refer to the area indicated by the red and the blue dotted arrow).

In Table II, algorithms with multiple matching frames also have good performance, which is according with our intuition. This could be explained by more map information provides more constrains to the current frame so that the large drift could be corrected and bounded.

3) YQ: In this part we conduct some experiments on our own data set (YQ). This data set contains four sequences, YQ1-YQ4, where YQ1-YQ3 were recorded on three separate days with different weather in summer and
YQ4 was collected in winter after snowing (c.f. Fig. 3). For this data set, we use YQ1 to build the map, and use YQ2-YQ4 to test the performance of the algorithms. The covariance of each map keyframe’s pose is given as $0.1I_{6 \times 6}$ and the re-linearization threshold is set as 20 pixels. The comparison of different algorithms is given in Table III [3]. VINS-Fusion is failed to perform continuous and consistent global localization because its odometry has large drift and its re-localization mechanism is triggered multiple times. So, the experimental results of VINS-Fusion are not listed.

From the results, we could find that for the case of single matching frame, introducing re-linearization mechanism could significantly improve the positioning accuracy. Combining multiple frames matching information and re-linearization mechanism, we could get the best result.

V. CONCLUSION

In this paper, we propose a consistent filter-based global localization framework, where a keyframe-based map is used to reduce storage. Schmidt-EKF is employed to handle the problem of too much computation in traditional EKF when the dimension of the state vector is too high. In this way, the computation increases linearly with the number of map keyframes. Moreover, we introduce a re-linearization mechanism to improve the accuracy of the first-order approximation of the observation function, and therefore improve the algorithm performance, especially in large scenes. From extensive experiments, we validate the effectiveness of our proposed framework.

REFERENCES

[1] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," in IEEE Transactions on Robotics, vol. 33, no. 5, pp. 1255-1262, Oct. 2017, doi: 10.1109/TRO.2017.2705103.

[2] J. Engel, V. Koltun and D. Cremers, "Direct Sparse Odometry," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 3, pp. 611-625, 1 March 2018, doi: 10.1109/TPAMI.2017.2638577.

[3] A. I. Mourikis and S. I. Roumeliotis, "A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation," Proceedings 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, 2007, pp. 3565-3572, doi: 10.1109/ROBOT.2007.364024.

[4] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang and G. Huang, "OpenVINS: A Research Platform for Visual-Inertial Estimation," 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 2020, pp. 4666-4672, doi: 10.1109/ICRA40945.2020.9196524.

[5] T. Qin, P. Li and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," in IEEE Transactions on Robotics, vol. 34, no. 4, pp. 1004-1020, Aug. 2018, doi: 10.1109/TRO.2018.2833729.

[6] T. Qin, S. Cao, J. Pan, and S. Shen, “A general optimization-based framework for global pose estimation with multiple sensors," arXiv e-prints, 2019, retrieved on March 1st, 2020. [Online]. Available: https://arxiv.org/abs/1901.03624v1.

[7] R. Mascaro, L. Teixeira, Hinzmann, R. Siegwart and M. Chli, "GOMSF: Graph-Optimization Based Multi-Sensor Fusion for robust UAV Pose estimation," 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 2018, pp. 1421-1428, doi: 10.1109/ICRA.2018.8460193.

[8] W. Lee, K. Eckenhoff, P. Geneva and G. Huang, "Intermittent GPS-aided VIO: Online Initialization and Calibration," 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 2020, pp. 5724-5731, doi: 10.1109/ICRA404945.2020.9197029.

[9] S. Lynen, T. Sattler, M. Bosse, J. Hesch, and R. 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.

[10] R. C. DuToit, J. A. Hesch, E. D. Nerurkar and S. I. Roumeliotis, "Consistent map-based 3D localization on mobile devices," 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 6253-6260, doi: 10.1109/ICRA.2017.7989741.

[11] T. Schneider et al., "Maplab: An Open Framework for Research in Visual-Inertial Mapping and Localization," in IEEE Robotics and Automation Letters, vol. 3, no. 3, pp. 1418-1425, July 2018, doi: 10.1109/LRA.2018.2800113.

[12] J. Campos, R. Elvira, J. G. Rodríguez, J. M. M. Montiel and J. D. Tardós, "ORB-SLAM3: An Accurate Open-Source Library for Visual-Inertial and Multi-Map SLAM," arXiv e-prints, 2020, retrieved on July 23rd 2020. [Online]. Available: https://arxiv.org/abs/2007.11898v1.

[13] P. Geneva, K. Eckenhoff and G. Huang, "A Linear-Complexity EKF for Visual-Inertial Navigation with Loop Closures," 2019 International Conference on Robotics and Automation (ICRA), Paris, France, 2019, pp. 3535-3541, doi: 10.1109/ICRA.2019.8793836.

[14] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart,"Iterated extended kalman filter based visual-inertial odometry using direct photometric feedback," The International Journal of Robotics Research. 2017, 36(10): 1053-1072.

[15] M. Bloesch, S. Omari, M. Hutter and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, 2015, pp. 298-304, doi: 10.1109/RSJ.2015.7353389.

[16] G. Cioffi, and D. Scaramuzza, "Tightly-coupled Fusion of Global Positional Measurements in Optimization-based Visual-Inertial Odometry," arXiv e-prints, 2020, retrieved on March 9th 2020. [Online]. Available: https://arxiv.org/abs/2003.04159.

[17] T. Qin, S. Cao, J. Pan, and S. Shen, "A general optimization-based framework for global pose estimation with multiple sensors," arXiv e-prints, 2019, retrieved on January 11st, 2019. [Online]. Available: https://arxiv.org/abs/1901.03638.

[18] T. Qin and S. Shen, "Online Temporal Calibration for Monocular Visual-Inertial Systems," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, 2018, pp. 3662-3669, doi: 10.1109/IROS.2018.8593603.

[19] E. Rosten and T. Drummond,"Machine learning for high-speed corner detection," European Conference on Computer Vision (ECCV), Springer, Heidelberg, 2006, pp. 430-443.

[20] M. Calonder, V. Lepetit, C. Strecha and P. Fua, "Brief: Binary robust independent elementary features," European Conference on Computer Vision (ECCV), Springer, Berlin, Heidelberg, 2010, pp. 778-792.

[21] J. Revaud et al., "R2d2: repeatable and reliable detector and descriptor," arXiv e-prints, 2019, retrieved on June 17th 2019, [Online]. Available: https://arxiv.org/abs/1906.06195.

[22] Y. Jiao et al., "2-Entity RANSAC for robust visual localization in changing environment," 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 2019, pp. 2478-2485, doi: 10.1109/RSJ.2019.8988356.

[23] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: an accurate o(n) solution to the pnp problem", International Journal of Computer Vision, 81(2), 155-166, 2009.

[24] N. Travy and S. I. Roumeliotis., "Indirect Kalman filter for 3D attitude estimation." University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep 2 (2005): 2005.

[25] B. Siciliano et al., "EuRoC - The Challenge Initiative for Euro-
pean Robotics,” ISR/Robotik 2014; 41st International Symposium on Robotics, Munich, Germany, 2014, pp. 1-7.

[26] J. Jeong, Y. Cho, Y.S. Shin, et al. "Complex urban dataset with multi-level sensors from highly diverse urban environments”, The International Journal of Robotics Research, 2019, 38(6):0278364919843399.