Optimization-Based Estimator Design for Vision-Aided Inertial Navigation: Supplemental Materials

Mingyang Li and Anastasios I. Mourikis
Dept. of Electrical Engineering, University of California, Riverside
E-mail: mli@ee.ucr.edu, mourikis@ee.ucr.edu

I. Feature Initialization

In this section, we describe the initialization of SLAM features in the hybrid MSCKF/SLAM filter. We consider a feature, $\mathbf{f}_i$, which has been continuously observed for m frames, and thus needs to be initialized. To initialize the feature, we can include it in the state vector, with an infinite initial covariance matrix, and then use all the m measurements simultaneously for an EKF update. The augmented state vector and its covariance matrix will thus be:

$$\hat{\mathbf{x}}_{aug_{k+1|k}} \leftarrow \begin{bmatrix} \hat{\mathbf{x}}_{k+1|k} \\ \hat{\mathbf{f}}_i \end{bmatrix}, \quad \mathbf{P}_{aug_{k+1|k}} \leftarrow \begin{bmatrix} \mathbf{P}_{k+1|k} & \mathbf{0} \\ \mathbf{0} & \mu\mathbf{I} \end{bmatrix} \tag{1}$$

with $\mu \to \infty$. In theory, we can use any randomly selected estimate $\hat{\mathbf{f}}_i$ (since the uncertainty is infinite). However, since we will need to compute Jacobians using $\hat{\mathbf{f}}_i$, a good estimate is preferable. Therefore, as the first step in the process, we use all the feature observations to triangulate the feature via least-squares minimization, similarly to the MSCKF. Subsequently, using this feature estimate, we can compute the residuals:

$$\tilde{\mathbf{z}}_{ij} = \mathbf{z}_{ij} - \mathbf{h}(\hat{\mathbf{x}}_{C_{j,k+1|k}}, \hat{\mathbf{f}}_i) \tag{2}$$

$$\simeq \mathbf{H}_{ij}\tilde{\mathbf{x}}_{C_{j,k+1|k}} + \mathbf{H}_{f_{ij}}\tilde{\mathbf{f}}_i + \mathbf{n}_{ij} \tag{3}$$

Staking all these residuals, we obtain:

$$\tilde{\mathbf{z}}_i \simeq \mathbf{H}_i\tilde{\mathbf{x}}_{k+1|k} + \mathbf{H}_{f_i}\tilde{\mathbf{f}}_i + \mathbf{n}_i \tag{4}$$

$$= \begin{bmatrix} \mathbf{H}_i & \mathbf{H}_{f_i} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_{k+1|k} \\ \tilde{\mathbf{f}}_i \end{bmatrix} + \mathbf{n}_i \tag{5}$$

$$= \begin{bmatrix} \mathbf{H}_i & \mathbf{H}_{f_i} \end{bmatrix} \tilde{\mathbf{x}}_{aug_{k+1|k}} + \mathbf{n}_i \tag{6}$$

where $\tilde{\mathbf{z}}_i$ and $\mathbf{n}_i$ are block vectors with elements $\tilde{\mathbf{z}}_{ij}$ and $\mathbf{n}_{ij}$, respectively, and $\mathbf{H}_i$ and $\mathbf{H}_{f_i}$ are matrices with block rows $\mathbf{H}_{ij}$ and $\mathbf{H}_{f_{ij}}$, for $j = 1 \dots m$. We can now use (6) directly for an EKF update using the standard EKF equations. A difficulty that needs to be addressed here is the choice of μ in (1). If one simply selects a "large" value for μ, there is a risk of numerical problems. A better approach is to explicitly compute the limit of the EKF update equations as $\mu \to \infty$, as we show next. For this process we start by defining a square orthonormal matrix $\mathbf{W}$ as:

$$\mathbf{W} = \begin{bmatrix} \mathbf{V} & \mathbf{U} \end{bmatrix} \tag{7}$$

where columns of $\mathbf{V}$ and $\mathbf{U}$ form bases of the left nullspace and the column space of $\mathbf{H}_{f_i}$, respectively. Since $\mathbf{W}$ is a full-rank matrix, instead of using the residual $\tilde{\mathbf{z}}_i$ in the EKF update, we can equivalently use the residual:

$$\tilde{\mathbf{z}}_i^c = \mathbf{W}^T\tilde{\mathbf{z}}_i = \begin{bmatrix} \mathbf{W}^T\mathbf{H}_i & \mathbf{W}^T\mathbf{H}_{f_i} \end{bmatrix} \tilde{\mathbf{x}}_{aug_{k+1|k}} + \mathbf{W}^T\mathbf{n}_i \Rightarrow \tag{8}$$

$$\begin{bmatrix} \mathbf{V}^T\tilde{\mathbf{z}}_i \\ \mathbf{U}^T\tilde{\mathbf{z}}_i \end{bmatrix} = \begin{bmatrix} \mathbf{V}^T\mathbf{H}_i & \mathbf{V}^T\mathbf{H}_{f_i} \\ \mathbf{U}^T\mathbf{H}_i & \mathbf{U}^T\mathbf{H}_{f_i} \end{bmatrix} \tilde{\mathbf{x}}_{aug_{k+1|k}} + \mathbf{W}^T\mathbf{n}_i \Rightarrow \tag{9}$$

$$\begin{bmatrix} \tilde{\mathbf{z}}_i^o \\ \tilde{\mathbf{z}}_i^1 \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{H}_i^o & \mathbf{0} \\ \mathbf{H}_i^1 & \mathbf{H}_i^2 \end{bmatrix}}_{\mathbf{H}_i^c} \tilde{\mathbf{x}}_{aug_{k+1|k}} + \mathbf{n}_i^c \tag{10}$$

where $\mathbf{n}_i^c$ is a noise vector with covariance matrix $\mathcal{E}\{(\mathbf{n}_i^c)(\mathbf{n}_i^c)^T\} = \mathbf{W}^T\mathcal{E}\{\mathbf{n}_i\mathbf{n}_i^T\}\mathbf{W} = \mathbf{W}^T\sigma^2\mathbf{I}\mathbf{W} = \sigma^2\mathbf{I}$. Note that the effect of multiplying (6) by $\mathbf{W}^T$ is to separate the information that the feature observations provide for the camera's relative motion, and the information for the feature position. The former is expressed by the first block row of (10) (which is the MSCKF

residual – see (5) in [1]), while the latter is expressed by the second block row of (10). If we use the residual $\tilde{\mathbf{z}}_i^c$ for an EKF update, the EKF update equations are:

$$\boldsymbol{\Delta}\mathbf{x}_{aug} = \mathbf{K}\tilde{\mathbf{z}}_i^c \tag{11}$$

$$\mathbf{P}_{aug_{k+1|k+1}} = \mathbf{P}_{aug_{k+1|k}} - \mathbf{P}_{aug_{k+1|k}}(\mathbf{H}_i^c)^T\mathbf{S}^{-1}(\mathbf{H}_i^c)\mathbf{P}_{aug_{k+1|k}} \tag{12}$$

$$\mathbf{K} = \mathbf{P}_{aug_{k+1|k}}(\mathbf{H}_i^c)^T\mathbf{S}^{-1} \tag{13}$$

$$\mathbf{S} = \mathbf{H}_i^c\mathbf{P}_{aug_{k+1|k}}(\mathbf{H}_i^c)^T + \sigma^2\mathbf{I} \tag{14}$$

We next show how to compute these equations with $\mu \to \infty$. To simplify the presentation, in what follows we use $\mathbf{P}_{aug}^+ \equiv \mathbf{P}_{aug_{k+1|k+1}}$, $\mathbf{P}_{aug} \equiv \mathbf{P}_{aug_{k+1|k}}$, $\mathbf{P}^+ \equiv \mathbf{P}_{k+1|k+1}$, $\mathbf{P} \equiv \mathbf{P}_{k+1|k}$, $\mathbf{H}_\star \equiv \mathbf{H}_i^\star$, for $\star = 1, 2, o, c$, and $\tilde{\mathbf{z}}_\star \equiv \tilde{\mathbf{z}}_i^\star$, for $\star = 1, o, c$. To begin with, we write (14) as:

$$\mathbf{S} = \mathbf{H}_c\mathbf{P}_{aug}\mathbf{H}_c^T + \sigma^2\mathbf{I} \tag{15}$$

$$= \begin{bmatrix}\mathbf{H}_o & \mathbf{0}\\ \mathbf{H}_1 & \mathbf{H}_2\end{bmatrix}\begin{bmatrix}\mathbf{P} & \mathbf{0}\\ \mathbf{0} & \mu\mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{H}_o & \mathbf{0}\\ \mathbf{H}_1 & \mathbf{H}_2\end{bmatrix}^T + \sigma^2\mathbf{I} \tag{16}$$

$$= \begin{bmatrix}\mathbf{H}_o\mathbf{P}\mathbf{H}_o^T + \sigma^2\mathbf{I} & \mathbf{H}_o\mathbf{P}\mathbf{H}_1^T\\ \mathbf{H}_1\mathbf{P}\mathbf{H}_o^T & \mathbf{H}_1\mathbf{P}\mathbf{H}_1^T + \mu\mathbf{H}_2\mathbf{H}_2^T + \sigma^2\mathbf{I}\end{bmatrix} \tag{17}$$

Then, we compute $\mathbf{S}^{-1}$, which is necessary for (13):

$$\mathbf{S}^{-1} = \begin{bmatrix}\mathbf{F}_{11} & \mathbf{F}_{21}^T\\ \mathbf{F}_{21} & \mathbf{F}_{22}\end{bmatrix} \tag{18}$$

where

$$\mathbf{F}_{11} = \left(\mathbf{H}_o\mathbf{P}\mathbf{H}_o^T + \sigma^2\mathbf{I} - \mathbf{H}_o\mathbf{P}\mathbf{H}_1^T\left(\mathbf{H}_1\mathbf{P}\mathbf{H}_1^T + \mu\mathbf{H}_2\mathbf{H}_2^T + \sigma^2\mathbf{I}\right)^{-1}\mathbf{H}_1\mathbf{P}\mathbf{H}_o^T\right)^{-1} \tag{19}$$

$$\mathbf{F}_{21} = -\left(\mathbf{H}_1\mathbf{P}\mathbf{H}_1^T + \mu\mathbf{H}_2\mathbf{H}_2^T + \sigma^2\mathbf{I}\right)^{-1}\mathbf{H}_1\mathbf{P}\mathbf{H}_o^T\mathbf{F}_{11} \tag{20}$$

$$\mathbf{F}_{22} = \left(\mathbf{H}_1\mathbf{P}\mathbf{H}_1^T + \mu\mathbf{H}_2\mathbf{H}_2^T + \sigma^2\mathbf{I} - \mathbf{H}_1\mathbf{P}\mathbf{H}_o^T\left(\mathbf{H}_o\mathbf{P}\mathbf{H}_o^T + \sigma^2\mathbf{I}\right)^{-1}\mathbf{H}_o\mathbf{P}\mathbf{H}_1^T\right)^{-1} \tag{21}$$

With $\mu \to \infty$, we obtain:

$$\lim_{\mu\to\infty}\mathbf{F}_{11} = \left(\mathbf{H}_o\mathbf{P}\mathbf{H}_o^T + \sigma^2\mathbf{I}\right)^{-1} \tag{22}$$

$$\lim_{\mu\to\infty}\mathbf{F}_{21} = \mathbf{0} \tag{23}$$

$$\lim_{\mu\to\infty}\mathbf{F}_{22} = \mathbf{0} \tag{24}$$

Thus, the EKF covariance update equation (12) is written as:

$$\mathbf{P}_{aug}^+ = \mathbf{P}_{aug} - \mathbf{P}_{aug}\mathbf{H}_c^T\mathbf{S}^{-1}\mathbf{H}_c\mathbf{P}_{aug} \tag{25}$$

$$= \lim_{\mu\to\infty}\left(\begin{bmatrix}\mathbf{P} & \mathbf{0}\\ \mathbf{0} & \mu\mathbf{I}\end{bmatrix} - \begin{bmatrix}\mathbf{P} & \mathbf{0}\\ \mathbf{0} & \mu\mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{H}_o & \mathbf{0}\\ \mathbf{H}_1 & \mathbf{H}_2\end{bmatrix}^T\begin{bmatrix}\mathbf{F}_{11} & \mathbf{F}_{21}^T\\ \mathbf{F}_{21} & \mathbf{F}_{22}\end{bmatrix}\begin{bmatrix}\mathbf{H}_o & \mathbf{0}\\ \mathbf{H}_1 & \mathbf{H}_2\end{bmatrix}\begin{bmatrix}\mathbf{P} & \mathbf{0}\\ \mathbf{0} & \mu\mathbf{I}\end{bmatrix}\right) \tag{26}$$

$$= \lim_{\mu\to\infty}\begin{bmatrix}\mathbf{P} - \left(\mathbf{P}\mathbf{H}_o^T\mathbf{F}_{11}\mathbf{H}_o\mathbf{P} + \mathbf{P}\mathbf{H}_1^T\mathbf{F}_{21}\mathbf{H}_o\mathbf{P} + \mathbf{P}\mathbf{H}_o^T\mathbf{F}_{21}^T\mathbf{H}_1\mathbf{P} + \mathbf{P}\mathbf{H}_1^T\mathbf{F}_{22}\mathbf{H}_1\mathbf{P}\right) & -\left(\mu\mathbf{H}_2^T\mathbf{F}_{21}\mathbf{H}_o\mathbf{P} + \mu\mathbf{H}_2^T\mathbf{F}_{22}\mathbf{H}_1\mathbf{P}\right)^T\\ -\left(\mu\mathbf{H}_2^T\mathbf{F}_{21}\mathbf{H}_o\mathbf{P} + \mu\mathbf{H}_2^T\mathbf{F}_{22}\mathbf{H}_1\mathbf{P}\right) & \mu\mathbf{I} - \mu^2\mathbf{H}_2^T\mathbf{F}_{22}\mathbf{H}_2\end{bmatrix} \tag{27}$$

Using results from (19) - (24), we obtain:

$$\mathbf{P}_{aug}^+ = \begin{bmatrix}\mathbf{P}^+ & -\left(\mathbf{H}_2^{-1}\mathbf{H}_1\mathbf{P}^+\right)^T\\ -\mathbf{H}_2^{-1}\mathbf{H}_1\mathbf{P}^+ & \mathbf{P}_{22}^+\end{bmatrix} \tag{28}$$

where

$$\mathbf{P}^+ = \mathbf{P} - \mathbf{P}\mathbf{H}_o^T\left(\mathbf{H}_o\mathbf{P}\mathbf{H}_o^T + \sigma^2\mathbf{I}\right)^{-1}\mathbf{H}_o\mathbf{P} \tag{29}$$

$$\mathbf{P}_{22}^+ = \left(\mathbf{H}_2^{-1}\mathbf{H}_1\right)\mathbf{P}^+\left(\mathbf{H}_2^{-1}\mathbf{H}_1\right)^T + \sigma^2\mathbf{H}_2^{-1}\mathbf{H}_2^{-T} \tag{30}$$

For the state update (11), we compute:

$$\Delta\mathbf{x} = \mathbf{P}_{aug}\mathbf{H}_c^T\mathbf{S}^{-1}\tilde{\mathbf{z}}_c \tag{31}$$

$$= \lim_{\mu\to\infty}\begin{bmatrix}\mathbf{P} & \mathbf{0}\\ \mathbf{0} & \mu\mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{H}_o & \mathbf{0}\\ \mathbf{H}_1 & \mathbf{H}_2\end{bmatrix}^T\begin{bmatrix}\mathbf{F}_{11} & \mathbf{F}_{21}^T\\ \mathbf{F}_{21} & \mathbf{F}_{22}\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{z}}_o\\ \tilde{\mathbf{z}}_1\end{bmatrix} \tag{32}$$

$$= \lim_{\mu\to\infty}\begin{bmatrix}\mathbf{P}\mathbf{H}_o^T\mathbf{F}_{11} & \mathbf{0}\\ -\mathbf{H}_2^{-1}\mathbf{H}_1\mathbf{P}\mathbf{H}_o^T\mathbf{F}_{11} & \mathbf{H}_2^{-1}\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{z}}_o\\ \tilde{\mathbf{z}}_1\end{bmatrix} \tag{33}$$

$$= \begin{bmatrix}\Delta\mathbf{x}_o\\ -\mathbf{H}_2^{-1}\mathbf{H}_1\Delta\mathbf{x}_o + \mathbf{H}_2^{-1}\tilde{\mathbf{z}}_1\end{bmatrix} \tag{34}$$

where

$$\Delta\mathbf{x}_o = \mathbf{P}\mathbf{H}_o^T\left(\mathbf{H}_o\mathbf{P}\mathbf{H}_o^T + \sigma^2\mathbf{I}\right)^{-1}\tilde{\mathbf{z}}_o \tag{35}$$

It is interesting to observe that $\mathbf{P}^+$ and $\Delta\mathbf{x}_o$ are the same as the covariance and state update we would compute if we used the feature for a standard MSCKF update.

II. Feature Re-parametrization

In the hybrid filter, we employ an inverse-depth parameterization for the SLAM features. Specifically, we parameterize the feature position of the i-th feature as:

$${}^G\mathbf{p}_{f_i} = {}^G\mathbf{p}_{C_i} + \frac{1}{\rho_i}\mathbf{R}_i\begin{bmatrix}\alpha_i\\ \beta_i\\ 1\end{bmatrix} \tag{36}$$

where (i) $\mathbf{f}_i = [\alpha_i,\, \beta_i,\, \rho_i]^T$ is the inverse-depth parameter vector, which is included in the state vector (see (14) in [1]), (ii) ${}^G\mathbf{p}_{C_i}$ is the global position of one camera pose in the sliding window of the state vector, termed the anchor of this feature, and (iii) $\mathbf{R}_i$ is a constant matrix. In the feature initialization, we choose the anchor to be the latest camera position, ${}^G\mathbf{p}_{C_i}$, which allows it to be kept in the state vector for a longer time, and set $\mathbf{R}_i = {}^G_{C_i}\hat{\mathbf{R}}$. The key advantage of our feature parametrization is that, since the anchor position is chosen as the position of a camera state that is already in the state vector, we *only* need to include the 3×1 vector $\mathbf{f}_i$ in the state vector, to represent the feature. This is in contrast to the traditional inverse depth parametrization [2], which includes both an anchor and an inverse-depth parameter vector for each individual feature in the state vector.

A difficulty here is that in certain situations the anchor of SLAM features will be removed from the state vector (see Algorithm I in [1]). When this occurs, we re-parameterize this feature as $\mathbf{f}_i'$ and select the position of the current latest camera, ${}^G\mathbf{p}_{C_i}'$, as the new anchor. To compute $\mathbf{f}_i'$, we note that:

$${}^G\mathbf{p}_{f_i} = {}^G\mathbf{p}_{C_i} + \frac{1}{\rho_i}\mathbf{R}_i\begin{bmatrix}\alpha_i\\ \beta_i\\ 1\end{bmatrix} = {}^G\mathbf{p}_{C_i}' + \frac{1}{\rho_i'}\mathbf{R}_i'\begin{bmatrix}\alpha_i'\\ \beta_i'\\ 1\end{bmatrix} \tag{37}$$

$$\Longrightarrow \quad \frac{1}{\rho_i'}\begin{bmatrix}\alpha_i'\\ \beta_i'\\ 1\end{bmatrix} = (\mathbf{R}_i')^T\left(-{}^G\mathbf{p}_{C_i}' + {}^G\mathbf{p}_{C_i} + \mathbf{R}_i\cdot\frac{1}{\rho_i}\begin{bmatrix}\alpha_i\\ \beta_i\\ 1\end{bmatrix}\right) \tag{38}$$

where $\mathbf{R}_i' = {}^G_{C_i}\hat{\mathbf{R}}'$ is a constant matrix. Using the above equation, the re-parameterized feature vector $\mathbf{f}_i'$ and its covariance can be easily computed.

III. Computation Analysis

In this section, we show that, in our proposed hybrid filter, it is computationally optimal to process a feature using the MSCKF if its track is lost after m frames, and to initialize it as a SLAM feature, if it is still active after m images. Specifically, we show that if a feature is initialized after $m-t$ ($t>0$) images, the computations of the hybrid filter will be increased.

A. Computational cost

We first calculate the computational cost (number of floating-point operations) for each update of the hybrid filter. It worth noting that, for efficiency, we only compute and store the upper-triangular elements of all symmetric positive semi-definite matrices used in the hybrid filter. We break down the computational cost of an EKF update into three parts: (i) computing the MSCKF residual and Jacobian matrix, (ii) computing SLAM residuals and Jacobians, and (iii) carrying out the EKF update. Out of these, the computation of the SLAM residuals and Jacobians has a negligible cost (linear in the number of SLAM features). In what follows, we calculate the computational cost of steps (i) and (iii).

1) Cost of Computing the MSCKF residual and Jacobian matrix: The cost of processing the MSCKF features consists of three parts: computing each feature's residual and Jacobian, performing the Mahalanobis test for each feature, and computing the final MSCKF residual and Jacobian using all MSCKF features. To compute the residual and Jacobian of a feature $\mathbf{f}_i$ we stack all its measurements together and apply the nullspace projection (see Eq (5) in [1]) to obtain the $(2\ell_i - 3) \times 1$ residual $\tilde{\mathbf{z}}_i^o$, and the $(2\ell_i - 3) \times (6\ell_i)$ Jacobian matrix $\mathbf{H}_i^o$ (ℓ_i is the feature track length for $\mathbf{f}_i$):

$$\tilde{\mathbf{z}}_i^o \doteq \mathbf{V}^T \tilde{\mathbf{z}}_i \simeq \mathbf{V}^T \mathbf{H}_i \tilde{\mathbf{x}} + \mathbf{V}^T \mathbf{n}_i = \mathbf{H}_i^o \tilde{\mathbf{x}} + \mathbf{n}_i^o \tag{39}$$

Calculating $\mathbf{H}_i^o$ and $\tilde{\mathbf{z}}_i^o$ using Householder projection matrices involves C_r^i operations, where

$$C_r^i = 72\ell_i^2 + l.o.t. \tag{40}$$

Here, $l.o.t.$ stands for the lower order terms (first order terms and constant terms). Then, a Mahalanobis gating test is carried out for $\mathbf{f}_i$, by computing γ:

$$\gamma = \left(\tilde{\mathbf{z}}_i^o\right)^T \left(\mathbf{S}_i^o\right)^{-1} \tilde{\mathbf{z}}_i^o \tag{41}$$

$$\mathbf{S}_i^o = \mathbf{H}_i^o \mathbf{P} \left(\mathbf{H}_i^o\right)^T + \sigma^2 \mathbf{I} \tag{42}$$

and comparing it against a threshold given by the 95-th percentile of the χ^2 distribution with $2\ell_i - 3$ degrees of freedom. The major costs in this step are:
1) computation of $\mathbf{S}_i^o$:

$$C_{M_1}^i = \frac{1}{2}\left(2\ell_i - 3\right)\left(6\ell_i\right)^2 + \frac{1}{2}\left(2\ell_i - 3\right)^2\left(6\ell_i\right) + l.o.t = 48\ell_i^3 - 90\ell_i^2 + l.o.t. \tag{43}$$

2) Cholesky factorization of $\mathbf{S}_i^o$, to obtain $\left(\mathbf{S}_i^o\right)^{-\frac{1}{2}}$:

$$C_{M_2}^i = \frac{1}{3}\left(2\ell_i - 3\right)^3 + l.o.t = \frac{8}{3}\ell_i^3 - 6\ell_i^2 + l.o.t. \tag{44}$$

3) computing $\left(\tilde{\mathbf{z}}_i^o\right)^T \left(\mathbf{S}_i^o\right)^{-\frac{1}{2}}$:

$$C_{M_3}^i = \frac{1}{2}\left(2\ell_i - 3\right)^2 + l.o.t. = 2\ell_i^2 + l.o.t. \tag{45}$$

Therefore, the total cost for the Mahalanobis gating test is:

$$C_M^i = C_{M_1}^i + C_{M_2}^i + C_{M_3}^i = \frac{152}{3}\ell_i^3 - 94\ell_i^2 + l.o.t. \tag{46}$$

By stacking together the residuals of the n features that pass this gating test, we obtain:

$$\tilde{\mathbf{z}}^o = \mathbf{H}^o \tilde{\mathbf{x}} + \mathbf{n}^o \tag{47}$$

where $\tilde{\mathbf{z}}^o$, $\mathbf{H}^o$, and $\mathbf{n}^o$ are block vectors/matrices, with block rows $\tilde{\mathbf{z}}_i^o$, $\mathbf{H}_i^o$, and $\mathbf{n}_i^o$, for $i = 1 \dots n$, respectively. We here assume that all the features processed by the MSCKF update pass the gating test, however, in the optimization approach in [1] the failure rate is also modeled. To process the above equation efficiently, we can compute the thin QR factorization of $\mathbf{H}^o$, written as $\mathbf{H}^o = \mathbf{Q}\mathbf{H}^r$, and then employ the residual $\tilde{\mathbf{z}}^r$ for the MSCKF update, defined as:

$$\tilde{\mathbf{z}}^r \doteq \mathbf{Q}^T \tilde{\mathbf{z}}^o \simeq \mathbf{H}^r \tilde{\mathbf{x}} + \mathbf{n}^r \tag{48}$$

We here assume that there is large number of features processed, which is the general case for both the hybrid filter [1] and the MSCKF [3]. Therefore, the number of rows in $\mathbf{H}^r$ is given by (see Section IV):

$$r \simeq 6m - 7 \tag{49}$$

Thus, we can calculate the cost for the QR projection[1]:

$$C_{QR} = 2\sum_{i=1}^{r}\left(b - i\right)\left(r - i\right) + l.o.t. \tag{50}$$

$$= 2\sum_{i=1}^{r}\left(br - bi - ri + i^2\right) + l.o.t. \tag{51}$$

$$= br^2 - br - \frac{1}{3}r^3 + 2r^2 + l.o.t. \tag{52}$$

[1]In general, the matrix $\mathbf{H}^o$ is a sparse matrix, and we are able to compute the QR decomposition more efficiently. The C_{QR} shown here is the worst-case, when all the features are observed m times. However, we show that even in the worst-case, the feature processing scheme in the hybrid filter (i.e., initializing features only if they are still active after m frames) is computationally optimal. Initialing features earlier will lead to decreased computation in the QR projection. Therefore, by showing that, even with the worst-case the late initialization is faster, the proof will also hold for the general case.

where b is the number of rows of the matrix $\mathbf{H}^o$, $b = \sum_{i=1}^{n}(2\ell_i - 3)$. We thus write the cost of processing MSCKF features:

$$C_{MSCKF} = C_r + C_M + C_{QR} = \sum_{i=1}^{n}(C_r^i + C_M^i) + C_{QR} \tag{53}$$

$$= \sum_{i=1}^{n}\left(\frac{152}{3}\ell_i^3 - 22\ell_i^2 + 2\ell_i r^2 - 2\ell_i r - 3r^2 + 3r\right) - \frac{1}{3}r^3 + 2r^2 + l.o.t. \tag{54}$$

2) Cost of carrying out the EKF update: Once the residual vector $\tilde{\mathbf{z}}^r$ and the Jacobian matrix $\mathbf{H}^r$ are obtained, the hybrid filter proceeds to compute the state correction and covariance update. The measurement Jacobian matrix $\mathbf{H}$ can be represented as:

$$\mathbf{H} = \begin{bmatrix}\mathbf{H}^r & \mathbf{0} \\ \mathbf{H}^a & \mathbf{H}^b\end{bmatrix} \tag{55}$$

where $\mathbf{H}^a$ and $\mathbf{H}^b$ contain the stacked Jacobians of the SLAM features with respect to the camera poses and the feature positions, respectively. Therefore, the covariance matrix of the residual of the hybrid filter is calculated as:

$$\mathbf{S} = \begin{bmatrix}\mathbf{S}_{11} & \mathbf{S}_{12} \\ \mathbf{S}_{21} & \mathbf{S}_{22}\end{bmatrix} = \begin{bmatrix}\mathbf{H}^r & \mathbf{0} \\ \mathbf{H}^a & \mathbf{H}^b\end{bmatrix}\begin{bmatrix}\mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22}\end{bmatrix}\begin{bmatrix}\mathbf{H}^r & \mathbf{0} \\ \mathbf{H}^a & \mathbf{H}^b\end{bmatrix}^T + \sigma^2\mathbf{I} \tag{56}$$

$$= \begin{bmatrix}\mathbf{H}^r\mathbf{P}_{11}(\mathbf{H}^r)^T + \sigma^2\mathbf{I} & \mathbf{H}^r\mathbf{P}_{11}(\mathbf{H}^a)^T + \mathbf{H}^r\mathbf{P}_{12}(\mathbf{H}^b)^T \\ (\mathbf{H}^r\mathbf{P}_{11}(\mathbf{H}^a)^T + \mathbf{H}^r\mathbf{P}_{12}(\mathbf{H}^b)^T)^T & \mathbf{H}^a\mathbf{P}_{11}(\mathbf{H}^a)^T + \mathbf{H}^a\mathbf{P}_{12}(\mathbf{H}^b)^T + \mathbf{H}^b\mathbf{P}_{21}(\mathbf{H}^a)^T + \mathbf{H}^b\mathbf{P}_{22}(\mathbf{H}^b)^T + \sigma^2\mathbf{I}\end{bmatrix} \tag{57}$$

where $\mathbf{P}_{11}$ corresponds to the marginal covariance matrix of the camera poses in $\mathbf{P}$, $\mathbf{P}_{12}$ and $\mathbf{P}_{21}$ to the cross correlation between the camera poses and the SLAM features, and $\mathbf{P}_{12}$ to the marginal covariance matrix of the SLAM features. For the term $\mathbf{S}$, we first observe that the computation of $\mathbf{H}^r\mathbf{P}_{11}$ is needed for both $\mathbf{S}_{11}$ and $\mathbf{S}_{12}$, which introduces a cost:

$$C_{S_1} = \frac{1}{2}r(6m)^2 + l.o.t. \tag{58}$$

where we have used the fact that $\mathbf{H}^r$ is the "R" matrix of the QR decomposition and thus has upper-triangular structure. In addition, the matrices $\mathbf{H}^a$ and $\mathbf{H}^b$ are both sparse, since the measurement Jacobian matrix of a SLAM feature with respect to the state vector only contains nonzero elements for the anchor pose and its inverse-depth parameter vector. Therefore, we obtain computational cost:
1) for $\mathbf{S}_{11}$

$$C_{S_2} = \frac{1}{4}r^2(6m) + l.o.t. \tag{59}$$

2) for $\mathbf{S}_{21}$

$$C_{S_3} = 6rs_k + \frac{1}{2}r(6m)(3s_k) + 6rs_k + l.o.t. \tag{60}$$

3) for $\mathbf{S}_{22}$

$$C_{S_4} = 39s_k^2 + l.o.t. \tag{61}$$

Therefore, the total cost for calculating the matrix $\mathbf{S}$ is

$$C_S = C_{S_1} + C_{S_2} + C_{S_3} + C_{S_4} \tag{62}$$

$$= 18rm^2 + \frac{3}{2}r^2m + 12rs_k + 39s_k^2 + 9rms_k + l.o.t. \tag{63}$$

For the above computational cost, we stress that we only compute the upper triangular parts of $\mathbf{S}_{11}$ and $\mathbf{S}_{22}$, because they are both symmetric positive semidefinite. Then, we compute the Cholesky decomposition of the matrix $\mathbf{S}$, whose cost is

$$C_{Cholesky} = \frac{1}{3}(r + 2s_k)^3 + l.o.t. \tag{64}$$

Completing the EKF update, the hybrid filter requires the following computations for the posterior state estimate and covariance:
1) calculation of $\mathbf{S}^{-\frac{1}{2}}\mathbf{HP}$, whose cost is

$$C_{posterior_1} = \frac{1}{2}(r + 2s_k)^2(15 + 6m + 3s_k) + l.o.t. \tag{65}$$

2) state correction $(\mathbf{S}^{-\frac{1}{2}}\mathbf{HP})^T\mathbf{S}^{-\frac{1}{2}}r$, whose cost is

$$C_{posterior_2} = \frac{1}{2}(r + 2s_k)^2 + (15 + 6m + 3s_k)(r + 2s_k) + l.o.t. \tag{66}$$

3) covariance update $\mathbf{P}_+ = \mathbf{P} - (\mathbf{S}^{-\frac{1}{2}}\mathbf{H}\mathbf{P})^T(\mathbf{S}^{-\frac{1}{2}}\mathbf{H}\mathbf{P})$, whose cost is

$$C_{posterior_3} \simeq \frac{1}{2}(15+6m+3s_k)^2 + \frac{1}{2}(15+6m+3s_k)^2(r+2s_k) + l.o.t. \tag{67}$$

Therefore, we obtain:

$$\begin{aligned} C_{posterior} &= C_{posterior_1} + C_{posterior_2} + C_{posterior_3} && (68)\\ &= \frac{1}{2}(15+6m+3s_k)^2 + \frac{1}{2}(15+6m+3s_k)^2(r+2s_k) + \frac{1}{2}(r+2s_k)^2(15+6m+3s_k) + \frac{1}{2}(r+2s_k)^2 && (69)\\ &\quad + (15+6m+3s_k)(r+2s_k) + l.o.t. && (70)\end{aligned}$$

Thus, the cost of the update of the hybrid filter C_{update} can be computed as:

$$C_{update} = C_S + C_{Cholesky} + C_{posterior} \tag{71}$$

Finally, the total cost per update of the hybrid filter is:

$$C_{hybrid} = C_{MSCKF} + C_{update} \tag{72}$$

B. Initialization Choice for SLAM features

The hybrid filter initializes a feature when it has been observed for m camera frames. We here show that this is the optimal strategy of feature processing in terms of computation. For any integer t, $0 < t < m-1$, initializing a feature t frames earlier (observed for $m-t$ frames) will result in increased computation.

In what follows, we analyze the computation of the hybrid filter by assuming *one* feature, $\mathbf{f}_j$, is initialized t steps earlier. However, our conclusion could be extended to the case when multiple features are initialized earlier. As a result of initializing the feature earlier, the cost of the MSCKF at the time of initialization will be reduced. Specifically, at that time step, the MSCKF cost will be:

$$\begin{aligned} C'_{MSCKF} &= \sum_{i=1, i\neq j}^{n}\left(\frac{152}{3}\ell_i^3 - 22\ell_i^2 + 2\ell_i r^2 - 2\ell_i r - 3r^2 + 3r\right) - \frac{1}{3}r^3 + 2r^2 && (73)\\ &\quad + \left(\frac{152}{3}(m-t)^3 - 22(m-t)^2 + 2(m-t)r^2 - 2(m-t)r - 3r^2 + 3r\right) + l.o.t. && (74)\end{aligned}$$

from which we compute the reduction in the MSCKF's computation as:

$$C'_{MSCKF} - C_{MSCKF} = -(222m^2t - 152mt^2 + \frac{152}{3}t^3 - 224mt + 22t^2) + l.o.t. \tag{75}$$

In contrast, in the next t time steps, the computation of the EKF update C'_{update} is increased, since the number of the SLAM features s_k will be increased by 1. Assuming s_k remains fixed, the cost of each EKF update after the initialization of $\mathbf{f}_j$ can be computed by:

$$\begin{aligned} C'_{update} &= C'_S + C'_{Cholesky} + C'_{posterior} && (76)\\ &= \frac{1}{2}(15+6m+3s_k+3)^2 + \frac{1}{2}(15+6m+3s_k+3)^2(r+2s_k+2) + \frac{1}{2}(r+2s_k+2)^2(15+6m+3s_k+3) && (77)\\ &\quad + \frac{1}{2}(r+2s_k+2)^2 + 18rm^2 + \frac{3}{2}r^2m + 12r(s_k+1) + 39(s_k+1)^2 + 9rm(s_k+1) + \frac{1}{3}(r+2s_k+2)^3 && (78)\\ &\quad + (15+6m+3s_k+3)(r+2s_k+2) + l.o.t. && (79)\end{aligned}$$

from which we obtain:

$$C'_{update} - C_{update} = 414m^2 + 288ms_k + 57s_k^2 + 435m + \frac{441}{2}s_k + l.o.t. \tag{80}$$

To prove the optimality, we need to show that

$$(C'_{MSCKF} - C_{MSCKF}) + t(C'_{update} - C_{update}) > 0 \tag{81}$$

Combining (75) and (80) leads to:

$$\begin{aligned} &(C'_{MSCKF} - C_{MSCKF}) + t(C'_{update} - C_{update}) && (82)\\ &\simeq (414m^2 + 288ms_k + 57s_k^2 + 435m + \frac{441}{2}s_k)t - (222m^2t - 152mt^2 + \frac{152}{3}t^3 - 224mt + 22t^2) && (83)\\ &= 192m^2t + 659mt + 288mts_k + 57s_k^2t + \frac{441}{2}s_kt + 152mt^2 - 50\frac{2}{3}t^3 - 22t^2 && (84)\end{aligned}$$

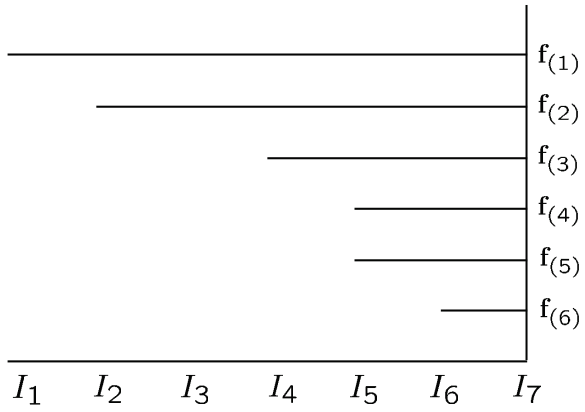

Fig. 1: Schematic representation of feature tracks. Point features $\mathbf{f}_{(i)}$, $i = 1 \dots 6$, are observed from the images, I_i, $i = 1 \dots 7$. In this example, $\ell_{(1)} = 7$, $\ell_{(2)} = 6, \dots, \ell_{(6)} = 2$. In image I_1 only one feature ($\mathbf{f}_{(1)}$) is observed, I_2 and I_3 contain two feature observations ($\mathbf{f}_{(1)}$ and $\mathbf{f}_{(2)}$), I_4 contains three image observations ($\mathbf{f}_{(1)}$, $\mathbf{f}_{(2)}$, and $\mathbf{f}_{(3)}$), and so on.

Using the fact that $t < m$, we obtain:

$$(C'_{MSCKF} - C_{MSCKF}) + t(C'_{update} - C_{update}) > \frac{427}{3}m^2t + 637mt + 288mts_k + 57s_k^2t + \frac{441}{2}s_kt + 152mt^2 > 0 \quad (85)$$

This completes the proof.

IV. Rank of the MSCKF measurement Jacobian matrix

We here prove that the rank of the MSCKF measurement Jacobian matrix $\mathbf{H}^o$ (see (47)) is:

$$r = 2(\ell_{(1)} + \ell_{(2)} + \ell_{(3)}) - d \quad (86)$$

where

$$d = \begin{cases} 9, \text{ if only three features are processed.} \\ 8, \text{ if only four features are processed, and at least one of them is only tracked for two images.} \\ 7, \text{ otherwise.} \end{cases} \quad (87)$$

In (86), $\ell_{(1)}$, $\ell_{(2)}$, and $\ell_{(3)}$ are the lengths of three longest feature tracks (corresponding to features $\mathbf{f}_{(1)}$, $\mathbf{f}_{(2)}$, and $\mathbf{f}_{(3)}$), and $\ell_{(1)} \geq \ell_{(2)} \geq \ell_{(3)}$. Fig. 1 shows a schematic of a representative case of feature tracks processed by the MSCKF. In this figure, six feature tracks, observed in a set of images I_1 to I_7, are shown. We here assume that I_1 is the oldest image, and I_7 is the latest one. Each image corresponds to one camera pose. It is important to point out that, because of the way features processed in the MSCKF (feature tracks are processed when they are complete), all features are observed by the last two poses (i.e., images I_6 and I_7) while longer feature tracks begin at earlier times. Moreover, we note that there are $\ell_{(i)}$ poses from which at least i features are observed, and there are $\ell_{(i)} - \ell_{(i+1)}$ poses from which exactly i features are observed.

To prove (86), we first note that the rank of the matrix $\mathbf{H}^o$ can be computed as:

$$\text{Rank}\,(\mathbf{H}^o) = \text{Col}\,(\mathbf{H}^o) - \text{Dim}\left(Null\big(\mathbf{H}^o\big)\right) \quad (88)$$

where $\text{Col}\,(\mathbf{H}^o) = 6\ell_{(1)}$ denotes the number of columns of the measurement Jacobian matrix $\mathbf{H}^o$, and $\text{Dim}\left(Null\big(\mathbf{H}^o\big)\right)$ is the dimension of the nullspace of $\mathbf{H}^o$. We note that all vectors $\Delta\mathbf{x}$ that satisfy $\mathbf{H}^o\Delta\mathbf{x} = \mathbf{0}$ represent the changes in the camera poses that are locally indistinguishable given the feature measurements in $\mathbf{H}^o$. In other words, the dimension of the nullspace of $\mathbf{H}^o$ can be interpreted as the dimension of the manifold $\mathcal{M}$ containing the locally indistinguishable states, and thus:

$$\text{Rank}\,(\mathbf{H}^o) = \text{Col}\,(\mathbf{H}^o) - \text{Dim}\,(\mathcal{M}) \quad (89)$$

$$= 6\ell_{(1)} - \text{Dim}\,(\mathcal{M}) \quad (90)$$

Therefore, to examine the rank of the matrix $\mathbf{H}^o$, we analyze the dimension of the manifold $\mathcal{M}$.

A. Known-feature case

We begin our analysis by considering a simplified situation, where the positions of the feature points are all *known*. In this case, we first focus on the $\ell_{(3)}$ latest camera poses (e.g., the poses corresponding to images I_4 to I_7 in Fig. 1), from which at least three features are observed. Estimating these camera poses by using the feature observations is the well-known perspective-N-point problem (PnP). It is known that when at least three features are observed from an image, barring degeneracy, a set of distinct solutions can be obtained for the camera poses [4]. Therefore, for the latest $\ell_{(3)}$ poses, for which at least three feature observations are available, no locally indistinguishable families of solutions exist.

Next, we examine the $\ell_{(2)} - \ell_{(3)}$ camera poses whose images contain measurements of exactly two features (i.e., images I_2 and I_3 in Fig. 1). For these poses, a family of solutions exists, which can be spanned by hypothesizing all the possible projections of a third known feature in the images. This projection has two unknown parameters for each image, which will result in a family of locally indistinguishable solutions with $2\left(\ell_{(2)} - \ell_{(3)}\right)$ parameters. Similarly, there exist $\ell_{(1)} - \ell_{(2)}$ camera poses whose images contain measurements of only one feature (i.e., image I_1 in Fig. 1). For each image, an indistinguishable family of solutions with four parameters exists, for a total of $4\left(l_{(1)} - l_{(2)}\right)$ parameters.

Therefore, we conclude that the dimension of the manifold $\mathcal{M}$ is:

$$\text{Dim}\left(\mathcal{M}\right) = 2\left(\ell_{(2)} - \ell_{(3)}\right) + 4\left(\ell_{(1)} - \ell_{(2)}\right) \tag{91}$$

and we thus see that, if the feature positions were known, the rank of the measurement Jacobian matrix would be equal to:

$$\begin{aligned}\text{Rank}\left(\mathbf{H}^o_{\text{known}}\right) &= 6\,\ell_{(1)} - \left(2\left(\ell_{(2)} - \ell_{(3)}\right) + 4\left(\ell_{(1)} - \ell_{(2)}\right)\right) &(92)\\ &= 2\left(\ell_{(1)} + \ell_{(2)} + \ell_{(3)}\right) &(93)\end{aligned}$$

B. Unknown-feature case

We now come back to the original problem, where the positions of the feature points are all *unknown*. In this case additional unknown parameters exist, and as a result, the dimension of both the nullspace of $\mathbf{H}^o$ and the manifold $\mathcal{M}$ will increase. Therefore, we can write the rank of the matrix $\mathbf{H}^o$ as:

$$\text{Rank}\left(\mathbf{H}^o\right) = 2\left(\ell_{(1)} + \ell_{(2)} + \ell_{(3)}\right) - d \tag{94}$$

where d is a positive integer, representing the increase in dimension of the manifold $\mathcal{M}$.

To compute the value of d, we examine three cases:

Case 1: only three features are available. In this scenario, if we hypothesize 3D positions for each of these features, the problem is transformed to a P3P, and a set of distinct solutions can be obtained, as explained earlier. Since the nine unknown parameters corresponding to the three features' positions can be freely chosen, we see that the family of valid solutions for the camera poses depends on an *additional* nine free parameters, compared to the case of the P3P. Therefore, in this case, we have $d = 9$.

Case 2: four features are available, and one of them is observed only two times. We now consider the case where a fourth feature $\mathbf{f}_{(4)}$ is observed, and $\ell_{(4)} = 2$. In this case, the observations of $\mathbf{f}_{(4)}$ introduce one additional constraint on the camera poses: the epipolar constraint between the two camera poses from which $\mathbf{f}_{(4)}$ was observed. As a result, the dimension of the manifold $\mathcal{M}$ is reduced by one compared to the case where only three features were available, and $d = 8$.

Case 3: five or more features are available, or four features are observed with $\ell_{(4)} \geq 3$. When five or more feature tracks are available, it is well known that we can determine the positions of the features in space up to seven unknown degrees of freedom, corresponding to the global 3D position, orientation, and scale [5]. The same is true if four features are available, and each feature is observed in at least three poses [6]. Thus, by freely choosing these 7 parameters, the problem can be reduced to the case of known features. As a result, we have $d = 7$.

These results lead to the expression in (87).

REFERENCES

[1] M. Li and A. I. Mourikis, "Optimization-based estimator design for vision-aided inertial navigation," in *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 9-14 2012.

[2] J. Civera, A. Davison, and J. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 932–945, 2008.

[3] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Rome, Italy, Apr. 2007, pp. 3565–3572.

[4] R. M. Haralick, C.-N. Lee, K. Ottenberg, and M. Noelle, "Review and analysis of solutions of the three point perspective pose estimation problem," *International Journal of Computer Vision*, vol. 13, no. 3, pp. 331–356, Dec. 1994.

[5] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, 2004.

[6] D. Nister and F. Schaffalitzky, "Four points in two or three calibrated views: Theory and practice," *International Journal of Computer Vision*, vol. 67, no. 2, pp. 211–231, 2006.