Consistency Analysis for Sliding-Window Visual Odometry

Tue-Cuong Dong-Si and Anastasios I. Mourikis
Dept. of Electrical Engineering, University of California, Riverside
E-mail: tdongsi@ee.ucr.edu, mourikis@ee.ucr.edu

February 10, 2012

Abstract

In this report we focus on the problem of *visual odometry*, i.e., the task of tracking the pose of a moving platform using visual measurements. In recent years, several VO algorithms have been proposed that employ nonlinear minimization in a sliding window of poses for this task. Through the use of iterative re-linearization, these methods are capable of successfully addressing the nonlinearity of the measurement models, and have become the de-facto standard for high-precision VO. In this work, we conduct an analysis of the properties of marginalization, which is the process through which older states are removed from the sliding window. This analysis shows that the standard way of marginalizing older poses results in an erroneous change in the rank of the measurements' information matrix, and leads to underestimation of the uncertainty of the state estimates. Based on the analytical results, we also propose a simple modification of the way in which the measurement Jacobians are computed. This modification avoids the above problem, and results in an algorithm with superior accuracy, as demonstrated in both simulation tests and real-world experiments.

1 Introduction and Related Work

Accurate pose tracking is an essential requirement in a number of systems, ranging from large-scale autonomous vehicles to small hand-held devices. If a system operates in an environment where reliable GPS reception is possible, then it can use the GPS signals, possibly in conjunction with additional sensors such as an inertial measurement unit (IMU), to track its position. However, in GPS-denied environments, different sensing modalities must be employed. Among the possible choices (e.g., cameras, laser rangefinders, sonars) cameras stand out due to their low cost, size and power consumption, as well as their widespread availability on mobile devices. Recent advances in the performance of vision sensors and computing hardware have made vision-based algorithms a more attractive option. Motivated by these reasons, in this paper we focus on the task of vision-based motion estimation.

Several different methods for localization using cameras have been proposed. For instance, when landmarks or fiducial points with known positions are available, these can be used to estimate the camera's position with respect to a known frame (e.g., [1–3]). However, when cameras operate in unknown and un-instrumented environments, these methods are not applicable. In these settings, one can employ simultaneous localization and mapping (SLAM)-type methods (see. e.g., [4–6] and references therein), which jointly estimate the camera trajectory and features' positions, or visual odometry (VO)-type methods [7–12], which track the pose of the camera only. In this work, our interest is in the latter type of methods. We are not interested in the problem of loop-closure detection, or in producing a map of the environment. Instead, we focus on the task of performing VO in real time, using the measurements from a monocular or stereo camera *only*.

Several methods exist that perform VO by estimating the camera displacement using the images recorded at two consecutive time steps (e.g., [12–14]). In these methods, when a single camera is used, additional sensors, scene information, or a statistical motion model must be employed to infer scale. By using only consecutive images, these methods attain low computational cost, but this often comes at the expense of accuracy. Due to the nonlinear nature of the camera measurement models and the existence of outliers in the image data, such methods may also be prone to failure.

At the other end of the spectrum, the "golden standard" method for vision-based estimation is bundle adjustment (e.g., [15] and references therein). In bundle adjustment the entire history of camera states and feature positions is jointly optimized using nonlinear minimization methods. This can result in high precision, but bundle-adjustment methods cannot operate in real-time in large-scale environments, as their computational complexity continuously increases over time. Incremental implementation of the nonlinear minimization is possible [16], but even in this case the computational cost increases in time, and eventually becomes unsuitable for real-time applications.

As a compromise between bundle adjustment and using pairwise displacement estimates, methods that perform optimization over a *sliding window* of states have recently gained popularity (see, e.g., [9–11, 17–20] and references therein). These techniques remove older states (features and/or camera poses) from the actively estimated state vector, and carry out iterative minimization to produce estimates for the most recent states. The use of a sliding window of more than two camera poses increases the accuracy and stability of these algorithms. At the same time, the removal of older states means that these methods have a *bounded* computational cost, which makes them suitable for real-time implementation. In fact, by changing the size of the sliding window we can adaptively control the computational requirements, which is an important characteristic of sliding-window algorithms. In this paper, we focus on the properties of these methods.

In addition to differences in the visual front end (i.e., the algorithms used for feature extraction and matching) the main difference between the various sliding-window algorithms is the way in which older states are removed. It is well-known that the theoretically "correct" way of removing states from the state vector is the process of *marginalization* [18, 19, 21, 22]. When marginalization is carried out the uncertainty of the discarded states is properly modelled in the estimator's equations, which is a key requirement for precise estimation. However, several successful methods do not follow this approach, and simply fix the values of the states that are removed from the state vector, using them to "bootstrap" the trajectory [9, 10, 17, 20]. In certain cases this is done purely for simplification of the algorithms and to improve computational efficiency. However, in [9] and [10] it is reported that this is done to reduce the estimation error. This fact, which appears to be counterintuitive at first, suggests that the "standard" way of carrying out marginalization may produce inaccuracies.

We note that in our previous work [19] we have shown that when a fixed-lag smoother is employed for tracking the motion of a vehicle using a camera and an IMU, the standard marginalization approach results in *inconsistency*[1]. Specifically, due to the marginalization process, two different estimates of the same states are used in computing certain Jacobian matrices in the estimator. In [19] we showed that this causes an infusion of information along directions of the state space where no actual information is provided by the measurements (the un-observable directions). This "artificial" information causes the estimates' covariance to be underestimated, and results in inconsistency. Moreover, since the accuracy of different states is misrepresented, the estimates' accuracy is also reduced.

Motivated by the observations of [9] and [10] on sliding-window VO, and the results of [19] on fixed lag smoothing using visual and inertial measurements, in this paper we carry out an analysis of the effects of marginalization in VO. Our results show that, similarly to the case of fixed-lag smoothing, even when *only* camera measurements are used for estimation, the same infusion of "artificial information" takes place. This degrades both the consistency and the accuracy of the trajectory estimates. Additionally, building on this analysis, we present a simple solution to the problem. This solution consists of ensuring that only one estimate of any given state variable is used in computing Jacobian matrices. The resulting algorithm is shown to perform better than competing approaches, in both simulation results and real-world experiments.

2 Sliding-window visual odometry

In this section, we present the "standard" algorithm for sliding window visual odometry [18, 19]. We start by discussing bundle adjustment, which serves to introduce the notation and will also be useful for our derivations in Section 3.

2.1 Bundle adjustment

We consider the case where a monocular or stereo camera moves in space, observing unknown features. The camera state vector at time-step i, $\mathbf{c}_i$, consists of the sensor orientation and position with respect to a global frame of reference:

$$\mathbf{c}_i = \begin{bmatrix} \mathbf{q}_{C_i} \\ \mathbf{p}_{C_i} \end{bmatrix} \tag{1}$$

where we have employed a unit-quaternion description of orientation, $\mathbf{q}_{C_i}$ [24]. Assuming calibrated cameras, the observation of feature j at time-step i is described by the perspective camera model:

$$\mathbf{z}_{ij} = \mathbf{h}\left(\mathbf{C}(\mathbf{q}_{C_i})(\mathbf{p}_{L_j} - \mathbf{p}_{C_i})\right) + \mathbf{n}_{ij} \tag{2}$$

[1] A recursive estimator is termed consistent when the state estimation errors are zero mean, and their covariance equals the one reported by the estimator [23, Section 5.4].

where $\mathbf{p}_{L_j}$ is the 3D feature position vector, $\mathbf{C}(\mathbf{q}_{C_i})$ is the rotation matrix corresponding to $\mathbf{q}_{C_i}$ (i.e., the rotation matrix from the global frame to the frame of camera i), $\mathbf{h}(\cdot)$ is the perspective measurement function:

$$\mathbf{h}(\mathbf{p}) = \frac{1}{p_3}\begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \text{ with } \mathbf{p} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}, \tag{3}$$

and finally $\mathbf{n}_{ij} \sim \mathcal{N}(\mathbf{0}_{2\times 1}, \mathbf{R}_{ij})$ is the measurement noise vector, modeled as a zero-mean Gaussian variable with covariance matrix $\mathbf{R}_{ij}$. If a stereo camera is used, the state vector in (1) represents the pose of one of the two sensors (e.g., the left one), and we obtain two measurements of the form (2), one from each camera. When performing bundle adjustment at time-step k, we simultaneously estimate the entire history of camera poses, $\mathbf{c}_{0:k} = \{\mathbf{c}_0, \ldots, \mathbf{c}_k\}$, as well as the positions of all features observed by the camera, $\mathbf{l}_{1:n} = \{\mathbf{p}_{L_1}, \ldots, \mathbf{p}_{L_n}\}$. We denote the state vector containing all these quantities by $\mathbf{x}_k$. The optimal solution is computed by maximizing the following pdf:

$$p(\mathbf{x}_k|\mathbf{z}_{0:k}) = p(\mathbf{x}_k) \prod_{(i,j)\in\mathcal{S}_a(k)} p(\mathbf{z}_{ij}|\mathbf{c}_i, \mathbf{p}_{L_j}) \tag{4}$$

where the set $\mathcal{S}_a(k)$ contains the pairs of indices (i, j) that describe all the feature observations through time k, $p(\mathbf{z}_{ij}|\mathbf{c}_i, \mathbf{p}_{L_j})$ is the Gaussian measurement-likelihood pdf, and $p(\mathbf{x}_k)$ is a pdf describing the prior information for the state vector. For instance, this may express our knowledge of the first camera state, constraints on the global scale in the case of monocular VO, etc. For clarity of presentation, we assume here that this prior information is modelled as a probabilistic constraint, $\mathbf{f}(\mathbf{x}_k) \sim \mathcal{N}(\hat{\mathbf{f}}, \mathbf{R}_p)$. The maximization of the above pdf is equivalent to the minimization of the following cost function:

$$c(\mathbf{x}_k) = \frac{1}{2}||\mathbf{f}(\mathbf{x}_k) - \hat{\mathbf{f}}||_{\mathbf{R}_p} + \frac{1}{2}\sum_{(i,j)\in\mathcal{S}_a(k)} ||\mathbf{z}_{ij} - \mathbf{h}(\mathbf{c}_i, \mathbf{p}_{L_j})||_{\mathbf{R}_{ij}}$$

where we have employed the notation $||\mathbf{a}||_{\mathbf{M}} = \mathbf{a}^T\mathbf{M}^{-1}\mathbf{a}$.

$c(\mathbf{x}_k)$ is a nonlinear cost function, which can be minimized using iterative Gauss-Newton minimization [15]. At the ℓ-th iteration of this method, a correction, $\Delta\mathbf{x}^{(\ell)}$, to the current estimate, $\mathbf{x}_k^{(\ell)}$, is computed by solving the linear system:

$$\mathbf{A}^{(\ell)}\Delta\mathbf{x}^{(\ell)} = -\mathbf{b}^{(\ell)} \tag{5}$$

where

$$\mathbf{A}^{(\ell)} = \mathbf{F}^T\mathbf{R}_p^{-1}\mathbf{F} + \sum_{(i,j)\in\mathcal{S}_a(k)} \mathbf{H}_{ij}^{(\ell)T}\mathbf{R}_{ij}^{-1}\mathbf{H}_{ij}^{(\ell)} \tag{6}$$

$$\mathbf{b}^{(\ell)} = \mathbf{F}^T\mathbf{R}_p^{-1}\left(\mathbf{f}\left(\mathbf{x}_k^{(\ell)}\right) - \hat{\mathbf{f}}\right) - \sum_{(i,j)\in\mathcal{S}_a(k)} \mathbf{H}_{ij}^{(\ell)T}\mathbf{R}_{ij}^{-1}\left(\mathbf{z}_{ij} - \mathbf{h}\left(\mathbf{c}_i^{(\ell)}, \mathbf{p}_{L_j}^{(\ell)}\right)\right) \tag{7}$$

In the above expressions, $\mathbf{F}$ is the Jacobian of the function $\mathbf{f}(\mathbf{x}_k)$ with respect to $\mathbf{x}_k$, and $\mathbf{H}_{ij}^{(\ell)}$ is the Jacobian of the measurement function $\mathbf{h}(\mathbf{c}_i, \mathbf{p}_{L_j})$ with respect to $\mathbf{x}_k$, evaluated at $\mathbf{x}_k^{(\ell)}$. Since the measurement model involves only one camera pose and one feature, $\mathbf{H}_{ij}^{(\ell)}$ has the following sparse structure:

$$\mathbf{H}_{ij}^{(\ell)} = \begin{bmatrix}\mathbf{0} & \ldots & \mathbf{H}_{L_{ij}}(\mathbf{x}_k^{(\ell)}) & \ldots & \mathbf{H}_{C_{ij}}(\mathbf{x}_k^{(\ell)}) & \ldots & \mathbf{0}\end{bmatrix} \tag{8}$$

where $\mathbf{H}_{L_{ij}}$ and $\mathbf{H}_{C_{ij}}$ are the Jacobians with respect to the feature position and the camera pose, respectively:

$$\begin{aligned}\mathbf{H}_{C_{ij}}(\mathbf{x}_k) &= \mathbf{\Gamma}_{ij}\mathbf{C}(\mathbf{q}_{C_i})\begin{bmatrix} \lfloor(\mathbf{p}_{L_j} - \mathbf{p}_{C_i})\times\rfloor\mathbf{C}^T(\mathbf{q}_{C_i}) & -\mathbf{I}_3 \end{bmatrix}\\ \mathbf{H}_{L_{ij}}(\mathbf{x}_k) &= \mathbf{\Gamma}_{ij}\mathbf{C}(\mathbf{q}_{C_i})\\ \mathbf{\Gamma}_{ij} &= \left.\frac{\partial\mathbf{h}(\mathbf{p})}{\partial\mathbf{p}}\right|_{\mathbf{p}=\mathbf{C}(\mathbf{q}_{C_i})(\mathbf{p}_{L_j}-\mathbf{p}_{C_i})}\end{aligned} \tag{9}$$

where $\mathbf{I}_3$ denotes the 3×3 identity matrix, and $\lfloor\mathbf{a}\times\rfloor$ is the skew-symmetric matrix associated with a vector $\mathbf{a}$. From Eq. (9), we can write:

$$\mathbf{H}_{C_{ij}} = \mathbf{H}_{L_{ij}}\begin{bmatrix} \lfloor(\mathbf{p}_{L_j} - \mathbf{p}_{C_i})\times\rfloor\mathbf{C}^T(\mathbf{q}_{C_i}) & -\mathbf{I}_3 \end{bmatrix}$$

$$= \mathbf{H}_{L_{ij}} \bar{\mathbf{H}}_{C_{ij}} \tag{10}$$

with $\bar{\mathbf{H}}_{C_{ij}} = [\lfloor(\mathbf{p}_{L_j} - \mathbf{p}_{C_i})\times\rfloor \mathbf{C}^T(\mathbf{q}_{C_i}) \ \ -\mathbf{I}_3]$. This is a useful property, which we will use in our derivation (Section 3.1). After solving (5) the correction is applied to the state, and the process is repeated until convergence.

2.2 Marginalization of old states

By exploiting the sparse structure of $\mathbf{A}^{(\ell)}$, we can speed up the solution of the linear system (5) considerably. However, as the camera continuously moves and observes new features, the size of the state vector $\mathbf{x}_k$ constantly increases. Therefore, in order to obtain a real-time algorithm with bounded computational complexity, we marginalize out older states.

We consider the following scenario: The moving camera observes features during the time interval $[0, k]$, and bundle adjustment is carried out at time-step k. Then, the states $\mathbf{x_m} = \{\mathbf{c}_0, \ldots, \mathbf{c}_{m-1}, \mathbf{p}_{L_1}, \ldots, \mathbf{p}_{L_q}\}$ (i.e., the m oldest camera poses and the q oldest landmarks, which we can no longer observe) are marginalized out, and only the states $\mathbf{x_r} = \{\mathbf{c}_m, \ldots, \mathbf{c}_k, \mathbf{p}_{L_{q+1}}, \ldots, \mathbf{p}_{L_n}\}$ remain active in the sliding window. After marginalization, the states $\mathbf{x_m}$ and all the measurements that involve them are discarded. We will use $\mathcal{S}_m$ to denote the set of indices (i, j) describing all the camera observations that involve either marginalized camera poses or marginalized landmarks, or both. These measurements provide information that is useful for the estimation of the remaining states, and this information should *not* be completely discarded. To express this information, we maintain in memory a vector $\mathbf{b_p}(k)$ and a matrix $\mathbf{A_p}(k)$, which are defined as follows:

$$\mathbf{b_p}(k) = \mathbf{b_{mr}}(k) - \mathbf{A_{rm}}(k)\mathbf{A_{mm}}(k)^{-1}\mathbf{b_{mm}}(k) \tag{11}$$

$$\mathbf{A_p}(k) = \mathbf{A_{rr}}(k) - \mathbf{A_{rm}}(k)\mathbf{A_{mm}}(k)^{-1}\mathbf{A_{mr}}(k) \tag{12}$$

where

$$\mathbf{b_m}(k) = \begin{bmatrix} \mathbf{b_{mm}}(k) \\ \mathbf{b_{mr}}(k) \end{bmatrix} \tag{13}$$

$$= \mathbf{F}^T \mathbf{R}_p^{-1}(\mathbf{f}(\hat{\mathbf{x}}_k(k)) - \hat{\mathbf{f}}) - \sum_{(i,j)\in\mathcal{S}_m} \mathbf{H}_{ij}^T(k)\mathbf{R}_{ij}^{-1}\left(\mathbf{z}_{ij} - \mathbf{h}(\hat{\mathbf{c}}_i(k), \hat{\mathbf{p}}_{L_j}(k))\right) \tag{14}$$

$$\mathbf{A_m}(k) = \begin{bmatrix} \mathbf{A_{mm}}(k) & \mathbf{A_{mr}}(k) \\ \mathbf{A_{rm}}(k) & \mathbf{A_{rr}}(k) \end{bmatrix} \tag{15}$$

$$= \mathbf{F}^T \mathbf{R}_p^{-1} \mathbf{F} + \sum_{(i,j)\in\mathcal{S}_m} \mathbf{H}_{ij}(k)\mathbf{R}_{ij}^{-1}\mathbf{H}_{ij}(k) \tag{16}$$

In the above, the size of the matrix partitions is defined by the length of the vectors $\mathbf{x_m}$ and $\mathbf{x_r}$, and all quantities are evaluated using the state estimate $\hat{\mathbf{x}}_k(k)$ (i.e., the estimate of $\mathbf{x}_k$ computed using bundle adjustment at time-step k). We point out that the matrix $\mathbf{A_m}$ represents the information contained in the prior and the discarded measurements, and $\mathbf{A_p}(k)$ is its Schur complement. Thus, as desired, $\mathbf{A_p}(k)$ represents all the information that the prior and the discarded measurements provide for estimating $\mathbf{x_r}$.

Proceeding further, as the camera keeps moving and observing features in the time interval $[k+1, k']$, the sliding window of states is augmented by the new camera and landmark states $\mathbf{x_n} = \{\mathbf{c}_{k+1}, \ldots, \mathbf{c}_{k'}, \mathbf{p}_{L_{n+1}}, \ldots, \mathbf{p}_{L_{n'}}\}$. Now, at time-step k', the sliding window contains the states $\mathbf{x_r}$ and $\mathbf{x_n}$. To obtain an estimate for the active state vector we once again employ iterative minimization of an appropriate cost function [19]. Similarly to the previous case, at the ℓ-th iteration the correction to the active states $\{\mathbf{x_r}, \mathbf{x_n}\}$ is computed by solving the linear system $\mathbf{A}^{(\ell)}\Delta\mathbf{x} = -\mathbf{b}^{(\ell)}$, with:

$$\mathbf{b}^{(\ell)} = \mathbf{\Pi}_\mathbf{r}^T \mathbf{b_p}(k) + \mathbf{\Pi}_\mathbf{r}^T \mathbf{A_p}(k)(\mathbf{x}_\mathbf{r}^{(\ell)} - \hat{\mathbf{x}}_\mathbf{r}(k)) - \sum_{(i,j)\in\mathcal{S}_a(k')} \mathbf{H}_{ij}^{(\ell)T}\mathbf{R}_{ij}^{-1}\left(\mathbf{z}_{ij} - \mathbf{h}(\mathbf{c}_i^{(\ell)}, \mathbf{p}_{L_j}^{(\ell)})\right) \tag{17}$$

$$\mathbf{A}^{(\ell)} = \mathbf{\Pi}_\mathbf{r}^T \mathbf{A_p}(k)\mathbf{\Pi}_\mathbf{r} + \sum_{(i,j)\in\mathcal{S}_a(k')} \mathbf{H}_{ij}^{(\ell)T}\mathbf{R}_{ij}^{-1}\mathbf{H}_{ij}^{(\ell)} \tag{18}$$

where the set $\mathcal{S}_a(k')$ contains the (i, j) indices corresponding to all the active measurements at time-step k' (i.e., all measurements involving states in $\mathbf{x_r}$ and $\mathbf{x_n}$), and $\mathbf{\Pi_r} = [\mathbf{I}_{\dim \mathbf{x_r}} \ \ \mathbf{0}]$. After the iterations have converged, we can again marginalize out some older states if desired, and proceed in the same way.

3 Estimator consistency

This section presents the main results of this work, which show the effects of marginalization on the estimator's consistency. Specifically, we prove that due to the marginalization the rank of the information matrix associated with the feature measurements is erroneously increased.

We start by considering what this information matrix would be if we had *not* performed marginalization, and instead carried out bundle adjustment for the entire trajectory in the time interval $[0, k']$. In that case, the matrix describing the information given by the measurements for the camera poses and feature positions would be given by:

$$\mathbf{J}_{\mathrm{ba}}(k') = \sum_{(i,j)\in\mathcal{S}} \mathbf{H}_{ij}^T(k')\mathbf{R}_{ij}^{-1}\mathbf{H}_{ij}(k') \tag{19}$$

where $\mathcal{S} = \mathcal{S}_a(k') \bigcup \mathcal{S}_m$ is the set describing all the available measurements in $[0, k']$. The above expression is the sum of the information contribution of each of these measurements.

Let us now return to the scenario described in the preceding section, i.e., marginalization of the states $\mathbf{x_m}$ at time-step k, and a new estimation step at time-step k'. In this case the estimator uses *the same* measurements, and thus the information matrix must contain a summation of the same number of terms as above. However, the important distinction is that the information contribution of the measurements that were discarded upon marginalization was evaluated at time-step k, and expressed by the matrix $\mathbf{A_p}(k)$ (see (12) and (18)). Thus, the information matrix for the entire history of states in $[0, k']$ is given by:

$$\mathbf{J}_{\mathrm{mar}}(k') = \sum_{(i,j)\in\mathcal{S}_m} \mathbf{H}_{ij}^T(k)\mathbf{R}_{ij}^{-1}\mathbf{H}_{ij}(k) + \sum_{(i,j)\in\mathcal{S}_a(k')} \mathbf{H}_{ij}^T(k')\mathbf{R}_{ij}^{-1}\mathbf{H}_{ij}(k') \tag{20}$$

Comparing (19) and (20) we clearly see that, since $\mathcal{S} = \mathcal{S}_a(k') \bigcup \mathcal{S}_m$, the *only* difference between these two information matrices are the state estimates used for computing the Jacobians. Apart from that, the structure of the matrices is the same in both cases. Yet, perhaps surprisingly, the mere fact that the Jacobians are evaluated using different state estimates causes the rank of these two matrices to differ. Specifically, in Section 3.1, we prove that

$$\mathrm{rank}(\mathbf{J}_{\mathrm{mar}}(k')) = \mathrm{rank}(\mathbf{J}_{\mathrm{ba}}(k')) + 3 \tag{21}$$

In other words, when marginalization takes place, the estimator *appears* to have more information (i.e., information along more directions of the state space) than when we perform bundle adjustment. Clearly, this increase is incorrect, since the estimators use the same measurements in both cases, and thus have access to the same information.

Since the sliding-window VO estimator believes it has more information, it underestimates the uncertainty of the state estimates it produces, i.e., it becomes *inconsistent*. Moreover, this over-confidence in the estimates' accuracy is not uniform: as discussed in Section 3.2, the estimator mistakenly "believes" that the global orientation is observable. It therefore has undue confidence in its orientation estimates, which ultimately degrades the accuracy of the computed state estimates, as corrections are not properly applied to all states. This reasoning shows that the artificial increase in the rank of the information matrix leads both to inconsistency *and* to suboptimal estimates. This is corroborated by the experimental results of Section 5.

It is worth pointing out that the results of our analysis are not restricted to the particular choice of states to marginalize. Even though we here focus on sliding-window estimation, where the oldest states are always marginalized, the same increase in the rank of the information matrix would occur if one chooses the states to marginalize in a different way (for instance, a common approach is to keep a selection of keyframes evenly spread in time and/or space).

3.1 Proof of (21)

We now prove (21) for the case in which the visual measurements are recorded by a stereo pair of cameras. We start by noting that $\mathbf{J}_{\mathrm{ba}}(k')$ and $\mathbf{J}_{\mathrm{mar}}(k')$ can be written as follows:

$$\mathbf{J}_{\mathrm{ba}}(k') = \sum_{(i,j)\in\mathcal{S}} \mathbf{H}_{ij}^T(k')\mathbf{R}_{ij}^{-1}\mathbf{H}_{ij}(k') \tag{22}$$

$$= \underbrace{\begin{bmatrix} \vdots \\ \mathbf{H}_{ij}(k') \\ \vdots \end{bmatrix}^T}_{\mathbf{H}^T(k')} \underbrace{\begin{bmatrix} \ddots & & \mathbf{0} \\ & \mathbf{R}_{ij}^{-1} & \\ \mathbf{0} & & \ddots \end{bmatrix}}_{\mathbf{Diag}(\mathbf{R}_{ij}^{-1})} \underbrace{\begin{bmatrix} \vdots \\ \mathbf{H}_{ij}(k') \\ \vdots \end{bmatrix}}_{\mathbf{H}(k')} \tag{23}$$

$$= \mathbf{H}^T(k')\,\mathbf{Diag}(\mathbf{R}_{ij}^{-1})\,\mathbf{H}(k') \tag{24}$$

and

$$\mathbf{J}_{\mathrm{mar}}(k') = \sum_{(i,j)\in\mathcal{S}_m} \mathbf{H}_{ij}^T(k)\mathbf{R}_{ij}^{-1}\mathbf{H}_{ij}(k) + \sum_{(i,j)\in\mathcal{S}_a(k')} \mathbf{H}_{ij}^T(k')\mathbf{R}_{ij}^{-1}\mathbf{H}_{ij}(k') \tag{25}$$

$$= \underbrace{\left[\begin{array}{c} \vdots \\ \mathbf{H}_{ij}(k) \\ \vdots \\ \hline \vdots \\ \mathbf{H}_{ij}(k') \\ \vdots \end{array}\right]^T}_{\mathbf{H}^T(k,k')} \underbrace{\left[\begin{array}{ccc|ccc} \ddots & \dots & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{ij}^{-1} & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \mathbf{0} & \dots & \mathbf{0} \\ \hline \mathbf{0} & \dots & \mathbf{0} & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \mathbf{0} & \mathbf{R}_{ij}^{-1} & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \dots & \mathbf{0} & \ddots \end{array}\right]}_{\mathbf{Diag}(\mathbf{R}_{ij}^{-1})} \underbrace{\left[\begin{array}{c} \vdots \\ \mathbf{H}_{ij}(k) \\ \vdots \\ \hline \vdots \\ \mathbf{H}_{ij}(k') \\ \vdots \end{array}\right]}_{\mathbf{H}(k,k')} \tag{26}$$

$$= \mathbf{H}^T(k,k')\,\mathbf{Diag}(\mathbf{R}_{ij}^{-1})\,\mathbf{H}(k,k') \tag{27}$$

where $\mathbf{Diag}(\cdot)$ denotes a block diagonal matrix, $\mathbf{H}(k')$ is a matrix with block rows $\mathbf{H}_{ij}(k')$, for all $(i,j) \in \mathcal{S}$, while $\mathbf{H}(k,k')$ is a matrix with block rows $\mathbf{H}_{ij}(k)$, for $(i,j) \in \mathcal{S}_m$, and $\mathbf{H}_{ij}(k')$, for $(i,j) \in \mathcal{S}_a(k')$.

We see that, similarly to $\mathbf{J}_{\mathrm{ba}}(k')$ and $\mathbf{J}_{\mathrm{mar}}(k')$, the matrices $\mathbf{H}(k')$ and $\mathbf{H}(k,k')$ have the same structure, and the only difference lies in the state estimates at which the Jacobians are evaluated. In the matrix $\mathbf{H}(k,k')$ the Jacobians of all measurements that involve marginalized states (the matrices $\mathbf{H}_{ij}$ where $(i,j) \in \mathcal{S}_m$) are evaluated using the state estimates available at time-step k, while all other Jacobians are evaluated using the estimates available at time-step k'. On the other hand, in the matrix $\mathbf{H}(k')$, *all* Jacobians are evaluated using the latest state estimates at time-step k'.

Proceeding further, we note that since $\mathbf{Diag}(\mathbf{R}_{ij}^{-1})$ is a full-rank matrix, we have that $\mathrm{rank}(\mathbf{J}_{\mathrm{ba}}(k')) = \mathrm{rank}(\mathbf{H}(k'))$, and $\mathrm{rank}(\mathbf{J}_{\mathrm{mar}}(k')) = \mathrm{rank}(\mathbf{H}(k,k'))$. Thus, to prove (21), it suffices to show that

$$\mathrm{rank}(\mathbf{H}(k,k')) = \mathrm{rank}(\mathbf{H}(k')) + 3 \tag{28}$$

To this end, we utilize the structure of the measurement Jacobians (see (10)) to factorize the matrix $\mathbf{H}(k')$ as follows:

$$\mathbf{H}(k') = \mathbf{D}(k')\bar{\mathbf{H}}(k') \tag{29}$$

where $\mathbf{D}(k') = \mathbf{Diag}\left(\mathbf{H}_{L_{ij}}(k')\right)$, $(i,j) \in \mathcal{S}$, and $\bar{\mathbf{H}}(k')$ is a matrix with block rows given by

$$\begin{aligned} \bar{\mathbf{H}}_{ij}(k') &= \begin{bmatrix} \mathbf{0} & \dots & \bar{\mathbf{H}}_{C_{ij}}(k') & \dots & \mathbf{I}_3 & \dots & \mathbf{0} \end{bmatrix} \\ \bar{\mathbf{H}}_{C_{ij}}(k') &= \begin{bmatrix} \lfloor(\hat{\mathbf{p}}_{L_j}(k') - \hat{\mathbf{p}}_{C_i}(k'))\times\rfloor \mathbf{C}^T(\hat{\mathbf{q}}_{C_i}(k')) & -\mathbf{I}_3 \end{bmatrix} \end{aligned} \tag{30}$$

A similar factorization can be obtained for $\mathbf{H}(k,k') = \mathbf{D}(k,k')\bar{\mathbf{H}}(k,k')$. We can now employ the following result, which allows us to compute the rank of the product of two matrices [25, 4.5.1]:

$$\mathrm{rank}(\mathbf{AB}) = \mathrm{rank}(\mathbf{B}) - \dim(\mathcal{N}(\mathbf{A}) \bigcap \mathcal{R}(\mathbf{B})) \tag{31}$$

where $\mathcal{N}$ and $\mathcal{R}$ denote the null space and the range of a matrix, respectively. An important intermediate result, whose proof is given in Appendix A, is the following:

Lemma 1 *When stereo camera measurements are used,*

$$\dim(\mathcal{N}(\mathbf{D}(k')) \bigcap \mathcal{R}(\bar{\mathbf{H}}(k'))) = 0 \tag{32}$$

$$\dim(\mathcal{N}(\mathbf{D}(k,k')) \bigcap \mathcal{R}(\bar{\mathbf{H}}(k,k'))) = 0 \tag{33}$$

Thus, by combining the results of Lemma 1, along with the decomposition in (29) and the property (31), we see that $\operatorname{rank}(\mathbf{H}(k')) = \operatorname{rank}(\bar{\mathbf{H}}(k'))$ and $\operatorname{rank}(\mathbf{H}(k,k')) = \operatorname{rank}(\bar{\mathbf{H}}(k,k'))$. Thus, to prove (28) it suffices to show that

$$\operatorname{rank}(\bar{\mathbf{H}}(k,k')) = \operatorname{rank}(\bar{\mathbf{H}}(k')) + 3 \tag{34}$$

To prove this result, we first apply elementary row and column operations to the matrix $\bar{\mathbf{H}}(k')$, in order to compute its rank. The intermediate steps are detailed in Appendix B. There, it is shown that when at least three non-collinear features are available, it is:

$$\operatorname{rank}(\bar{\mathbf{H}}(k')) = 3n' + 6k' \tag{35}$$

Note that the state vector consists of n' landmarks and $k'+1$ camera poses, which means that the matrix $\bar{\mathbf{H}}(k')$ has $3n' + 6k' + 6$ columns, equal to the number of state variables. Thus, the above result shows that $\bar{\mathbf{H}}(k')$ is rank deficient by 6.

Proceeding with the proof of (34), we apply similar elementary row and column operations to the matrix $\bar{\mathbf{H}}(k,k')$ to compute its rank. However, recall that in this matrix some of the Jacobians are evaluated using the state estimates at time step k, and some using the state estimates at time-step k'. As a result, for certain state variables, *two different* estimates appear in the equations. This means that certain cancellations that occurred in when applying elementary row and column operations in $\bar{\mathbf{H}}(k')$ do *not* happen when these operations are applied to $\bar{\mathbf{H}}(k,k')$. As a result, the rank of the matrix is increased. Specifically, in Appendix C, it is shown that:

$$\operatorname{rank}(\bar{\mathbf{H}}(k,k')) = 3n' + 6k' + 3 \tag{36}$$

This result, in conjunction with (35), completes the proof.

3.2 Physical interpretation

Equation (21) shows that when marginalization takes place the estimator erroneously believes to have information along three more directions of the state space. To identify these directions, we can examine the nullspaces of the matrices $\mathbf{J}_{\mathrm{mar}}(k')$ and $\mathbf{J}_{\mathrm{ba}}(k')$. First, note that in the preceding section it was shown that $\operatorname{rank}(\mathbf{J}_{\mathrm{ba}}(k')) = \operatorname{rank}(\bar{\mathbf{H}}(k')) = 3n' + 6k'$. Since $\mathbf{J}_{\mathrm{ba}}(k')$ is a $(3n' + 6k' + 6) \times (3n' + 6k' + 6)$ matrix, this result means that $\mathbf{J}_{\mathrm{ba}}(k')$ has a nullspace of dimension 6. To obtain a basis for this nullspace, we define the $(3n' + 6k' + 6) \times 6$ matrix

$$\mathbf{N}(\hat{\mathbf{x}}_{\mathbf{m}}(k'), \hat{\mathbf{x}}_{\mathbf{r}}(k'), \hat{\mathbf{x}}_{\mathbf{n}}(k')) = \left[\begin{array}{cc} \mathbf{I}_3 & -\lfloor \hat{\mathbf{p}}_{L_1}(k') \times \rfloor \\ \vdots & \vdots \\ \mathbf{I}_3 & -\lfloor \hat{\mathbf{p}}_{L_{n'}}(k') \times \rfloor \\ \hline \mathbf{0}_{3\times 3} & \mathbf{C}(\hat{\mathbf{q}}_{C_0}(k')) \\ \mathbf{I}_3 & -\lfloor \hat{\mathbf{p}}_{C_0}(k') \times \rfloor \\ \vdots & \vdots \\ \mathbf{0}_{3\times 3} & \mathbf{C}(\hat{\mathbf{q}}_{C_{k'}}(k')) \\ \mathbf{I}_3 & -\lfloor \hat{\mathbf{p}}_{C_{k'}}(k') \times \rfloor \end{array}\right] \tag{37}$$

It is easy to verify that the following property holds (here we are assuming a state variable ordering of all features followed by all camera poses):

$$\mathbf{J}_{\mathrm{ba}}(k') \cdot \mathbf{N}(\hat{\mathbf{x}}_{\mathbf{m}}(k'), \hat{\mathbf{x}}_{\mathbf{r}}(k'), \hat{\mathbf{x}}_{\mathbf{n}}(k')) = \mathbf{0}$$

which means that the columns of the matrix $\mathbf{N}$ (which are linearly independent) form a basis for the nullspace of $\mathbf{J}_{\mathrm{ba}}(k')$. The nullspace of the information matrix $\mathbf{J}_{\mathrm{ba}}(k')$ describes changes in the state that cannot be detected using the available measurements (i.e., the unobservable subspace). Close examination of the columns of $\mathbf{N}$ reveals that the first block column corresponds to global translations of the entire state vector, while the second corresponds to global rotations. This should come as no surprise, since by using only measurements of unknown features only the *relative* camera motion can be determined, and not the global pose.

Let us now examine the situation when marginalization takes place. In this case, based on the results of the preceding section, we see that $\operatorname{rank}(\mathbf{J}_{\mathrm{mar}}(k')) = 3n' + 6k' + 3$, which in turn means that the nullspace of $\mathbf{J}_{\mathrm{mar}}(k')$ is only of dimension three. Multiplying $\mathbf{J}_{\mathrm{mar}}(k')$ with the first block column (first three columns) of the matrix $\mathbf{N}$ yields a zero

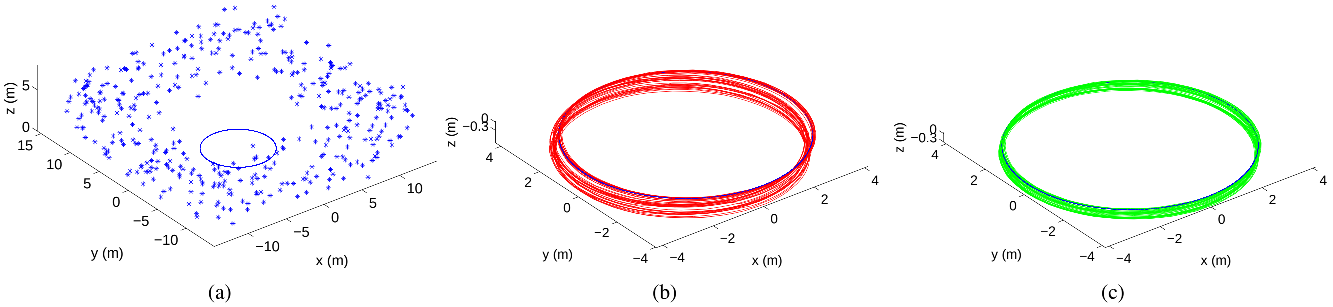


Figure 1: Example trajectories computed in one Monte-Carlo trial for stereo-based VO. (a) The simulation environment and the true trajectory (blue) (b) The estimated trajectory by S-VO (red) (c) The estimated trajectory by M-VO (green).

matrix, and thus we conclude that the nullspace of $\mathbf{J}_{\mathrm{mar}}(k')$ is spanned by the first three columns of $\mathbf{N}$. We thus see that the directions that correspond to the global orientation are "missing" from the nullspace of the information matrix. In other words, the sliding-window VO estimator incorrectly "believes" that the global orientation is observable. As discussed in Section 3, this results in inconsistent estimates, and a degradation of accuracy.

4 Improving the estimator's performance

In this section we describe a simple modification of the standard sliding-window VO algorithm that prevents the increase of the rank of the measurement information matrix, and improves the performance of the estimator. As shown in Section 3.1, the erroneous increase in the rank of the information matrix is caused by the fact that two different estimates of certain states appear in the measurement Jacobians. Specifically, these are the camera poses and/or landmarks in $\mathbf{x_r}$ that are "connected" to marginalized states via measurements in the set $\mathcal{S}_m$. For example, let us assume that the camera pose $\mathbf{c}_{i^\star}$ is one of the camera poses that remains in the sliding window after marginalization at time step k. Moreover, consider that the feature $\mathbf{p}_{L_{j^\star}}$ was observed from this camera pose and was marginalized at time step k. Then the information matrix $\mathbf{H}^T_{i^\star j^\star}(k)\mathbf{R}^{-1}_{i^\star j^\star}\mathbf{H}_{i^\star j^\star}(k)$ will appear in the summation in (12), and will be used to compute the matrix $\mathbf{A_p}(k)$. Later on, when we perform iterative estimation at time step k', the camera pose $\mathbf{c}_{i^\star}$ is still in the sliding window, but now the Jacobians that involve this pose are evaluated using the estimate $\hat{\mathbf{c}}_{i^\star}(k')$. Thus, two different estimates of $\mathbf{c}_{i^\star}$ are used for Jacobian computations.

To avoid this problem, a simple solution is to change the state estimates that are used for Jacobian computations. Specifically, when an active state is connected to already marginalized states via measurements (e.g., $\mathbf{c}_{i^\star}$ in the above example), then we use the estimate that was available at the time of marginalization (e.g., $\hat{\mathbf{c}}_{i^\star}(k)$) for all subsequent Jacobian computations. In this way only a single estimate of this state appears in the information matrix, and the increase in rank is averted. We stress that we use these "older" estimates when computing the Jacobians *only*, and we still allow the actual estimates to be updated normally in the Gauss-Newton iterations. Clearly, the use of "older" estimates for computing Jacobians will inevitably lead to larger linearization errors. However, as indicated by the results presented in the next section, the effect of this loss of linearization accuracy is not significant, while avoiding the creation of fictitious information leads to significantly improved precision.

5 Results

5.1 Simulation results

In this section, we present simulation results that demonstrate the performance of the modified sliding-window VO algorithm presented in Section 4. In our simulation setup, we consider a camera (stereo or monocular) that moves along a circular trajectory of radius 4 m in a $24 \times 24 \times 5$ m room with 600 visual point features randomly placed near the walls (as shown in Fig. 1a). The camera moves with constant velocity and angular velocity of 2 m/s and 0.5 rad/s, respectively. The camera frame rate is 10 Hz and 5 Hz for monocular and stereo camera, respectively, the field of view is 45°, the focal length is 500 pixels, and the measurement noise standard deviation is 1 pixel. For the simulations with a stereo camera, we set the stereo baseline equal to 0.12 m. The features can be observed for up to 30 consecutive camera poses.

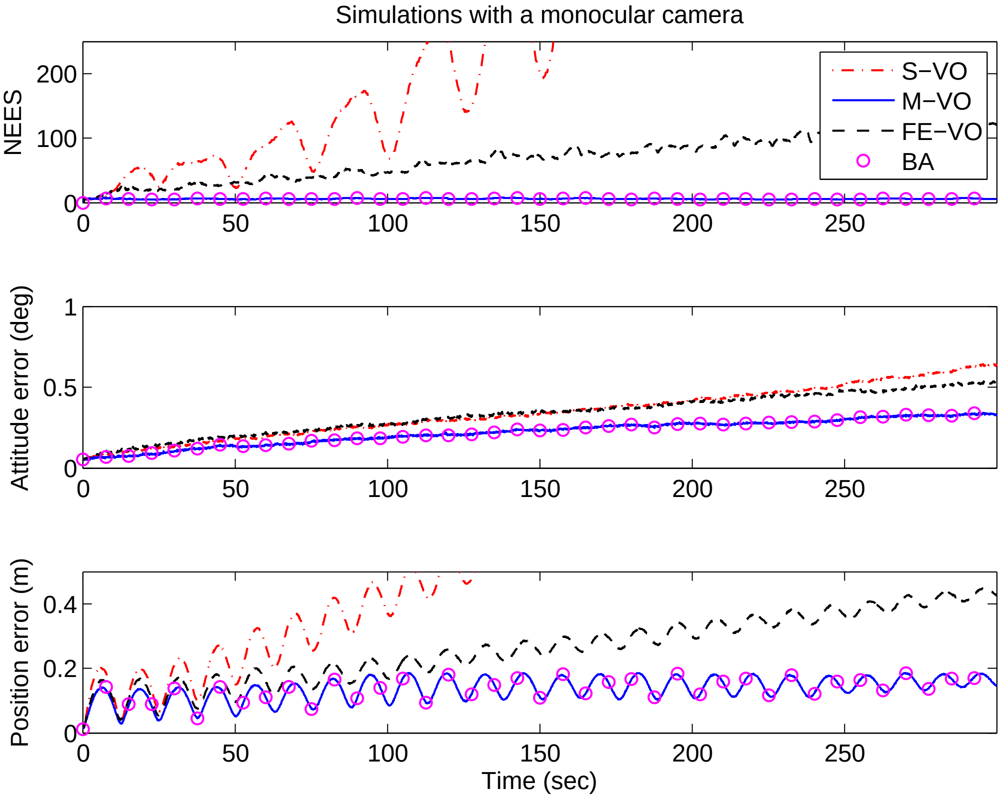


Figure 2: Simulation results for VO using a monocular camera. From top to bottom: (a) The average value of the camera-pose NEES over time (b) The RMS errors of the camera attitude over time (c) The RMS errors of the camera position over time.

Therefore, in the sliding-window VO we choose a sliding window containing the 40 latest camera poses and the landmarks seen in these poses. In these simulations, we compare the performance of (i) the modified sliding-window VO algorithm (termed M-VO) presented in Section 4, (ii) sliding-window VO with the standard linearization approach (termed S-VO), (iii) sliding-window VO with fixed estimates for the previous states, similarly to [10] (termed FE-VO), and finally, (iv) the standard bundle adjustment (termed BA) that estimates the entire history of the camera poses (i.e., no marginalization).

Fig. 2 shows results using a monocular camera for VO, and Fig. 3 shows results for the case of stereo. In both figures, the consistency of the estimators is measured by the average normalized estimation error squared (NEES) for the latest camera pose, and accuracy is measured by the root mean squared (RMS) error of the orientation and position. All results are averaged over 50 Monte-Carlo runs. In these plots, we can observe that the M-VO algorithm clearly outperforms S-VO, both in terms of consistency (i.e., NEES) and accuracy (i.e., RMS errors). When using a monocular camera, the average NEES over all Monte-Carlo runs and all timesteps equals 6.090 for M-VO, very close to 6.017, which is the value obtained for BA. The average NEES for S-VO is a staggering 2350. Since the pose error state is of dimension 6, the "ideal" NEES value would equal 6. Similarly, when using a stereo camera, the average NEES equals 6.351, 6.497, and 6435 for the BA, M-VO, and S-VO algorithms, respectively. Most importantly, in both monocular and stereo VO, we see that the performance of the new M-VO algorithm is *almost indistinguishable* from that of the "golden standard" of BA, even though its computational cost is orders of magnitude smaller.

For illustration purposes, two sample estimated trajectories, computed by the S-VO and M-VO algorithms respectively, are also provided in Fig. 1. This plot clearly demonstrates that the M-VO method can yield results that are substantially more accurate than those of the standard method. In particular, as shown in Fig. 1, the position errors in the vertical axis are significantly larger for S-VO.

We now turn our attention to the FE-VO method. This method shares similar characteristics to our M-VO, in the sense that it uses "older" estimates of previous states in computing Jacobians. This improves the consistency of the method, as shown in Figs. 2 and 3. Specifically, FE-VO has an average NEES of 68.68 and 52.48, respectively, in the monocular and stereo cases. In terms of RMS errors, FE-VO also outperforms S-VO significantly. However, both in terms of NEES

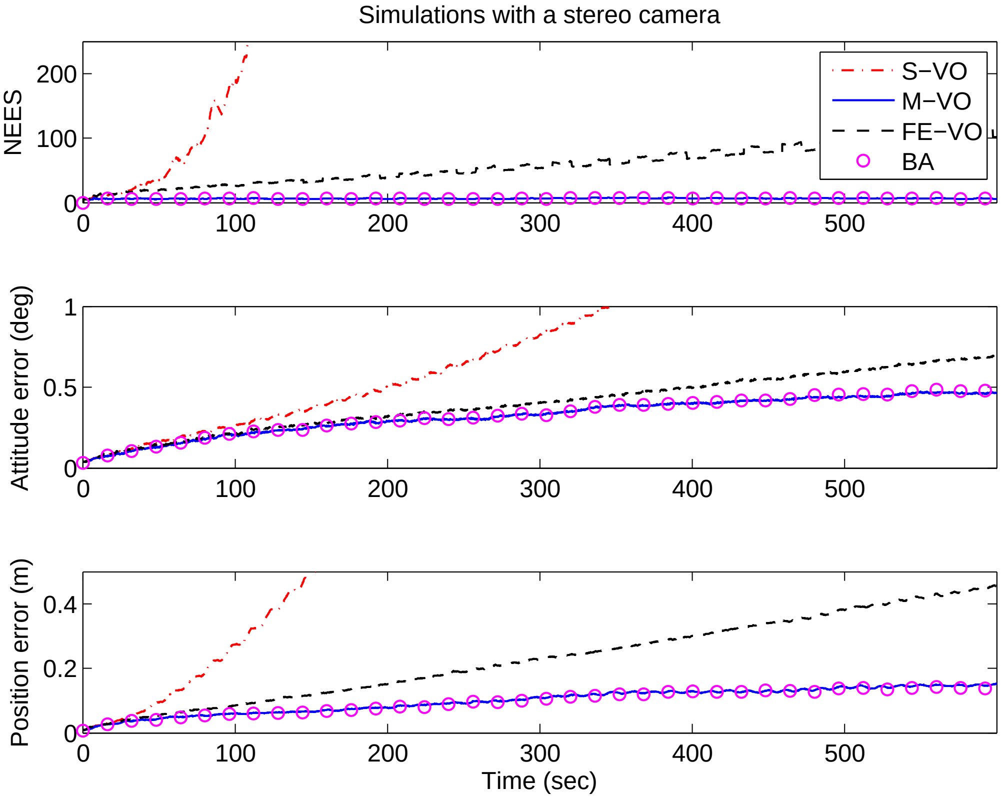


Figure 3: Simulation results for VO using a stereo camera. From top to bottom: (a) The average value of the camera-pose NEES over time (b) The RMS errors of the camera attitude over time (c) The RMS errors of the camera position over time.

and RMS errors, the FE-VO approach performs worse than the proposed M-VO. This can be explained by the fact that in FE-VO the estimates of the older states are fixed and not updated, which degrades accuracy. In contrast, in the M-VO method the older states are updated normally, thus attaining higher precision.

5.2 Real-world experiment

The performance of the proposed algorithm is also validated in a real-world setting. For this purpose, we tested the algorithm on Epoch A (Campus) of the New College dataset [26], using stereo images. Only the first epoch is used because when the robot passes through a dark tunnel to a different area, no point features can be detected in the captured images. Since the stereo camera is the sole sensor used in this experiment, it is impossible for the algorithm to recover the camera pose. In the section of the dataset that we used, the camera moved for about 7 min, performing three loops around the main oval quad in New College, Oxford. The dataset consists of more than 15000 images of resolution 512×384 pixels, captured by a PointGrey Bumblebee stereo rig at 20 Hz. Features are extracted using the Harris corner detector [27], and matched using normalized cross-correlation.

In Fig. 5, the trajectory estimates of the S-VO and the M-VO are shown in red and blue, respectively. Unfortunately, ground truth is not available for the New College dataset, but the camera was driven so that the trajectory in each loop was identical. Compared with previous results on the New College dataset [28], the results obtained by the S-VO and M-VO are similar to the best estimates. It should be noted that only ego-motion is estimated from stereo images, and we do not address loop closure, compared to [28]. By inspection of the trajectory estimates, we can deduce that the position errors of the S-VO are larger than M-VO, both in the x-y plane and along the z-axis. In particular, the side views of the trajectories in Fig. 5b and the elevation estimate plot in Fig. 5c show that the accuracy of the two VO methods is very different in the vertical direction. Since the camera moves on flat ground, its elevation (z coordinate) should remain approximately constant throughout its motion. The results show that by using the prior linearization points to preserve the observability properties of the estimator, we achieve better overall estimation accuracy.

(a) Left

(b) Right

(c) Left

(d) Right

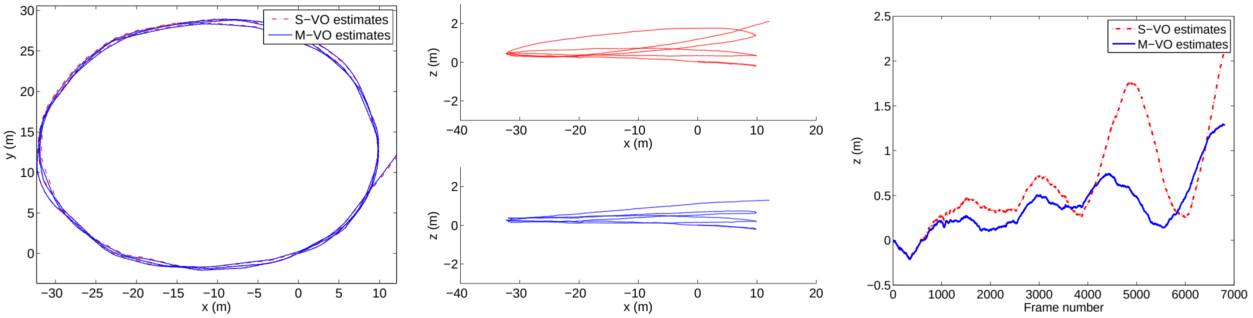
Figure 4: Sample stereo images from the New College dataset.

6 Conclusions

In this work, we presented an analysis of the properties of sliding-window minimization for visual odometry. Estimators that employ this approach attain bounded computational cost by marginalizing out older states, so as to maintain an approximately constant number of states active at all times. By analyzing the details of the Jacobian computations needed for the marginalization equations, we have proven that the standard linearization method will introduce erroneous information into the estimator, resulting in inconsistency. Based on this analysis, we proposed a modified linearization scheme, to prevent the infusion of artificial information, and improve estimation performance. Our simulation tests and real-world experiments demonstrated that this modified sliding-window VO estimator outperforms competing methods, both in terms of accuracy and consistency.

Acknowledgements

This work was supported by the National Science Foundation (grant no. IIS-1117967), the UC Riverside Bourns College of Engineering, and the Hellman Family Foundation.



(a) VO estimates, top view

(b) VO estimates, side view

(c) Elevation estimate over time

Figure 5: Real-world results for VO using a stereo camera.

Appendices

In the proofs, we make use of row and column operations to transform a matrix into another equivalent matrix with the same rank, but with structure that is more amenable to analysis. We use the sign "$\sim$" to denote a transformation using row or column operations on the matrix. In addition, we employ the following ordering of the variables: all the features come first, followed by all the camera poses in the temporal order. As a result, the matrices $\mathbf{H}(k')$ and $\mathbf{H}(k,k')$ can be divided into partitions of features and camera poses, respectively:

$$\mathbf{H} = \begin{bmatrix} \vdots \\ \mathbf{H}_{ij} \\ \vdots \end{bmatrix} = \left[\begin{array}{c|cccc} \mathbf{H}_{L_0} & \mathbf{H}_{C_0} & \dots & \dots & \mathbf{0}_{2l_0\times 6} \\ \vdots & \mathbf{0}_{2l_1\times 6} & \mathbf{H}_{C_1} & \dots & \vdots \\ \vdots & \vdots & \dots & \ddots & \mathbf{0}_{2l_{k'-1}\times 6} \\ \mathbf{H}_{L_{k'}} & \mathbf{0}_{2l_{k'}\times 6} & \dots & \dots & \mathbf{H}_{C_{k'}} \end{array}\right] \tag{38}$$

where the time-step indices (k) and (k') have been omitted, since the above structure is shared by both $\mathbf{H}(k')$ and $\mathbf{H}(k,k')$. In the above, we group all the Jacobians $\mathbf{H}_{C_{ij}}$ corresponding to a camera pose $\mathbf{c}_i$ into a block matrix $\mathbf{H}_{C_i}$, and the feature Jacobians $\mathbf{H}_{L_{ij}}$, for all j such that $(i,j)\in\mathcal{S}$, into a block matrix $\mathbf{H}_{L_i}$. For example, if at time-step i the camera observes l_i landmarks then $\mathbf{H}_{C_i}$ is an $l_i\times 1$ block vector containing the Jacobians $\mathbf{H}_{C_{ij}}$, while $\mathbf{H}_{L_i}$ contains l_i block rows, where each row contains the Jacobian $\mathbf{H}_{L_{ij}}$ at the jth position:

$$\mathbf{H}_{C_i} = \begin{bmatrix} \mathbf{H}_{C_{ij_1}} \\ \mathbf{H}_{C_{ij_2}} \\ \vdots \\ \mathbf{H}_{C_{ij_{l_i}}} \end{bmatrix}, \quad \mathbf{H}_{L_i} = \begin{bmatrix} \mathbf{0} & \mathbf{H}_{L_{ij_1}} & \dots & \dots & \mathbf{0} \\ \mathbf{H}_{L_{ij_2}} & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \dots & \mathbf{H}_{L_{ij_{l_i}}} & \dots & \mathbf{0} \end{bmatrix} \tag{39}$$

Following the factorization $\mathbf{H} = \mathbf{D}\bar{\mathbf{H}}$ in (29), the structure of matrix $\bar{\mathbf{H}}$ is similarly given by:

$$\bar{\mathbf{H}} = \left[\begin{array}{c|cccc} \bar{\mathbf{H}}_{L_0} & \bar{\mathbf{H}}_{C_0} & \dots & \dots & \mathbf{0}_{3l_0\times 6} \\ \vdots & \mathbf{0}_{3l_i\times 6} & \bar{\mathbf{H}}_{C_1} & \dots & \vdots \\ \vdots & \vdots & \dots & \ddots & \mathbf{0}_{3l_{i+1}\times 6} \\ \bar{\mathbf{H}}_{L_{k'}} & \mathbf{0}_{3l_{k'}\times 6} & \dots & \dots & \bar{\mathbf{H}}_{C_{k'}} \end{array}\right] \tag{40}$$

where

$$\bar{\mathbf{H}}_{C_i} = \begin{bmatrix} \bar{\mathbf{H}}_{C_{ij_1}} \\ \bar{\mathbf{H}}_{C_{ij_2}} \\ \vdots \\ \bar{\mathbf{H}}_{C_{ij_{l_i}}} \end{bmatrix} = \begin{bmatrix} \lfloor(\mathbf{p}_{L_{j_1}} - \mathbf{p}_{C_i})\times\rfloor \mathbf{C}^T(\mathbf{q}_{C_i}) & -\mathbf{I}_3 \\ \lfloor(\mathbf{p}_{L_{j_2}} - \mathbf{p}_{C_i})\times\rfloor \mathbf{C}^T(\mathbf{q}_{C_i}) & -\mathbf{I}_3 \\ \vdots & \\ \lfloor(\mathbf{p}_{L_{j_{l_i}}} - \mathbf{p}_{C_i})\times\rfloor \mathbf{C}^T(\mathbf{q}_{C_i}) & -\mathbf{I}_3 \end{bmatrix} \tag{41}$$

$$
\bar{\mathbf{H}}_{L_i} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_3 & \dots & \dots & \mathbf{0}_{3\times3} \\ \mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_{3\times3} & \dots & \mathbf{I}_3 & \dots & \mathbf{0}_{3\times3} \end{bmatrix} \tag{42}
$$

Here, we emphasize that the above matrix structure is shared by both $\bar{\mathbf{H}}(k')$ and $\bar{\mathbf{H}}(k, k')$.

A Proof of Lemma 1

We here prove that if stereo measurements are used, then $\dim(\mathcal{N}(\mathbf{D}) \bigcap \mathcal{R}(\bar{\mathbf{H}})) = 0$. To this end, it suffices to show that $\mathcal{N}(\mathbf{D}) = \emptyset$. In this proof, the time indices (k') and (k, k') are omitted since the analysis is similar for both $\mathbf{H}(k')$ and $\mathbf{H}(k, k')$. Given the definition of $\mathbf{D}$ in (29), we see that to prove that $\mathcal{N}(\mathbf{D}) = \emptyset$, it suffices to show that $\mathbf{H}_{L_{ij}}$ is a full-rank matrix. Assuming a calibrated stereo camera rig with perspective camera model, the measurement model for the feature observation is given by:

$$
\mathbf{h}(\mathbf{p}) = \begin{bmatrix} p_1/p_3 \\ p_2/p_3 \\ (p_1+b)/p_3 \\ p_2/p_3 \end{bmatrix} \tag{43}
$$

where

$$
\mathbf{p} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} = \mathbf{C}(\mathbf{q}_{C_i})(\mathbf{p}_{L_j} - \mathbf{p}_{C_i}) \tag{44}
$$

is the position of the feature with respect to the left camera, and b is the baseline between the two cameras' optical centers. Here, we assume that the pose of the stereo sensor is represented by the pose of the left camera. The Jacobian of the above measurement function with respect to $\mathbf{p}_{L_j}$ is given by:

$$
\mathbf{H}_{L_{ij}} = \mathbf{\Gamma}_{ij}\mathbf{C}(\mathbf{q}_{C_i}) \tag{45}
$$

where

$$
\mathbf{\Gamma}_{ij} = \frac{\partial \mathbf{h}(\mathbf{p})}{\partial \mathbf{p}} = \begin{bmatrix} 1/p_3 & 0 & -p_1/p_3^2 \\ 0 & 1/p_3 & -p_2/p_3^2 \\ 1/p_3 & 0 & -(p_1+b)/p_3^2 \\ 0 & 1/p_3 & -p_2/p_3^2 \end{bmatrix} \tag{46}
$$

By applying two elementary row operations, it is straightforward to obtain the following reduced form of $\mathbf{\Gamma}_{ij}$:

$$
\mathbf{\Gamma}_{ij} \sim \begin{bmatrix} 1/p_3 & 0 & -p_1/p_3^2 \\ 0 & 1/p_3 & -p_2/p_3^2 \\ 0 & 0 & -b/p_3^2 \\ 0 & 0 & 0 \end{bmatrix} \tag{47}
$$

From this row-reduced form of $\mathbf{\Gamma}_{ij}$, we conclude that $\mathbf{\Gamma}_{ij}$ has full column rank. Together with the fact that the rotation matrix $\mathbf{C}(\mathbf{q}_{C_i})$ has full rank, this shows that $\mathbf{H}_{L_{ij}}$ also has full column rank. Consequently, $\mathbf{D}$ is a full-rank matrix, which means that $\mathcal{N}(\mathbf{D}) = \emptyset$. This directly leads to:

$$
\dim(\mathcal{N}(\mathbf{D}(k')) \bigcap \mathcal{R}(\bar{\mathbf{H}}(k'))) = 0 \tag{48}
$$

$$
\dim(\mathcal{N}(\mathbf{D}(k, k')) \bigcap \mathcal{R}(\bar{\mathbf{H}}(k, k'))) = 0 \tag{49}
$$

B Proof of (35)

From (40), we have the following structure for the matrix $\bar{\mathbf{H}}(k')$:

$$\bar{\mathbf{H}}(k') = \left[\begin{array}{c|cccc} \bar{\mathbf{H}}_{L_0} & \bar{\mathbf{H}}_{C_0}(k') & \cdots & \cdots & \mathbf{0}_{3l_0\times 6} \\ \vdots & \mathbf{0}_{3l_i\times 6} & \bar{\mathbf{H}}_{C_1}(k') & \cdots & \vdots \\ \vdots & \vdots & \cdots & \ddots & \mathbf{0}_{3l_{i+1}\times 6} \\ \bar{\mathbf{H}}_{L_{k'}} & \mathbf{0}_{3l_{k'}\times 6} & \cdots & \cdots & \bar{\mathbf{H}}_{C_{k'}}(k') \end{array}\right] \tag{50}$$

All quantities appearing in this section are evaluated using the estimates at time step k', and therefore we omit the time index (k') and the estimate symbol “ˆ”, for notational simplicity. For each block row of the matrix $\bar{\mathbf{H}}(k')$, we apply the following column operations:

$$\bar{\mathbf{H}}_i = \left[\begin{array}{c|ccccc} \bar{\mathbf{H}}_{L_i} & \mathbf{0}_{3l_i\times 6} & \cdots & \bar{\mathbf{H}}_{C_i} & \cdots & \mathbf{0}_{3l_i\times 6} \end{array}\right]$$

$$= \left[\begin{array}{ccccc|ccccc} \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \lfloor(\mathbf{p}_{L_{j_1}} - \mathbf{p}_{C_i})\times\rfloor \mathbf{C}^T(\mathbf{q}_{C_i}) & -\mathbf{I}_3 & \cdots & \mathbf{0}_{3\times 3} \\ \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \lfloor(\mathbf{p}_{L_{j_2}} - \mathbf{p}_{C_i})\times\rfloor \mathbf{C}^T(\mathbf{q}_{C_i}) & -\mathbf{I}_3 & \cdots & \mathbf{0}_{3\times 3} \\ & & \vdots & & & \vdots & & \vdots & & & \vdots \\ \mathbf{0}_{3\times 3} & \cdots & \mathbf{0}_{3\times 3} & \cdots & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \cdots & \lfloor(\mathbf{p}_{L_{j_i}} - \mathbf{p}_{C_i})\times\rfloor \mathbf{C}^T(\mathbf{q}_{C_i}) & -\mathbf{I}_3 & \cdots & \mathbf{0}_{3\times 3} \end{array}\right]$$

Multiply the first block column of $\bar{\mathbf{H}}_{C_i}$ by $\mathbf{C}(\mathbf{q}_{C_i})$

$$\sim \left[\begin{array}{ccccc|ccccc} \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \lfloor(\mathbf{p}_{L_{j_1}} - \mathbf{p}_{C_i})\times\rfloor & -\mathbf{I}_3 & \cdots & \mathbf{0}_{3\times 3} \\ \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \lfloor(\mathbf{p}_{L_{j_2}} - \mathbf{p}_{C_i})\times\rfloor & -\mathbf{I}_3 & \cdots & \mathbf{0}_{3\times 3} \\ & & \vdots & & & \vdots & & \vdots & & & \vdots \\ \mathbf{0}_{3\times 3} & \cdots & \mathbf{0}_{3\times 3} & \cdots & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \cdots & \lfloor(\mathbf{p}_{L_{j_i}} - \mathbf{p}_{C_i})\times\rfloor & -\mathbf{I}_3 & \cdots & \mathbf{0}_{3\times 3} \end{array}\right]$$

Multiply the block column corresponding to $\mathbf{p}_{C_i}$ by $-\lfloor\mathbf{p}_{C_i}\times\rfloor$ and add to the column corresponding to $\mathbf{q}_{C_i}$

$$= \left[\begin{array}{ccccc|ccccc} \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \lfloor(\mathbf{p}_{L_{j_1}})\times\rfloor & -\mathbf{I}_3 & \cdots & \mathbf{0}_{3\times 3} \\ \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \lfloor(\mathbf{p}_{L_{j_2}})\times\rfloor & -\mathbf{I}_3 & \cdots & \mathbf{0}_{3\times 3} \\ & & \vdots & & & \vdots & & \vdots & & & \vdots \\ \mathbf{0}_{3\times 3} & \cdots & \mathbf{0}_{3\times 3} & \cdots & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \cdots & \lfloor(\mathbf{p}_{L_{j_i}})\times\rfloor & -\mathbf{I}_3 & \cdots & \mathbf{0}_{3\times 3} \end{array}\right]$$

Change column ordering of the camera states: position first, then orientation.

$$\sim \left[\begin{array}{ccccc|ccccc} \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & -\mathbf{I}_3 & \lfloor(\mathbf{p}_{L_{j_1}})\times\rfloor & \cdots & \mathbf{0}_{3\times 3} \\ \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & -\mathbf{I}_3 & \lfloor(\mathbf{p}_{L_{j_2}})\times\rfloor & \cdots & \mathbf{0}_{3\times 3} \\ & & \vdots & & & \vdots & & & \vdots & & \vdots \\ \mathbf{0}_{3\times 3} & \cdots & \mathbf{0}_{3\times 3} & \cdots & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \cdots & -\mathbf{I}_3 & \lfloor(\mathbf{p}_{L_{j_i}})\times\rfloor & \cdots & \mathbf{0}_{3\times 3} \end{array}\right]$$

Next, we rearrange $\bar{\mathbf{H}}(k')$ such that the block rows corresponding to the *first* observation of each feature are grouped into the first row partition $\bar{\mathbf{H}}_0$. The remaining feature observations are grouped into the second to $(k'+1)$-th row partitions, in the temporal order. Thus,

$$\bar{\mathbf{H}}(k') \sim \left[\begin{array}{cccc|cc|c|cc|cc}
\mathbf{I}_3 & \dots & \dots & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \lfloor(\mathbf{p}_{L_1})\times\rfloor & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\
\mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \vdots & -\mathbf{I}_3 & \lfloor(\mathbf{p}_{L_2})\times\rfloor & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\
\vdots & \mathbf{0}_{3\times3} & \ddots & \mathbf{0}_{3\times3} & \multicolumn{2}{c|}{\vdots} & \ddots & \multicolumn{2}{c|}{\ddots} & \multicolumn{2}{c}{\vdots} \\
\mathbf{0}_{3\times3} & \dots & \dots & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & -\mathbf{I}_3 & \lfloor(\mathbf{p}_{L_{n'}})\times\rfloor & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\
\hline
 & \vdots & \vdots & & \multicolumn{2}{c|}{\ddots} & \ddots & \multicolumn{2}{c|}{\vdots} & \multicolumn{2}{c}{\vdots} \\
\hline
\mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \vdots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & -\mathbf{I}_3 & \lfloor(\mathbf{p}_{L_2})\times\rfloor & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\
\vdots & \mathbf{0}_{3\times3} & \ddots & \mathbf{0}_{3\times3} & \multicolumn{2}{c|}{\vdots} & \ddots & \multicolumn{2}{c|}{\ddots} & \multicolumn{2}{c}{\vdots} \\
\hline
\mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \vdots & \multicolumn{2}{c|}{\mathbf{0}_{3\times3}} & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \lfloor(\mathbf{p}_{L_2})\times\rfloor \\
\vdots & \mathbf{0}_{3\times3} & \ddots & \mathbf{0}_{3\times3} & \multicolumn{2}{c|}{\vdots} & \ddots & \multicolumn{2}{c|}{\ddots} & \multicolumn{2}{c}{\vdots} \\
\mathbf{0}_{3\times3} & \dots & \dots & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \lfloor(\mathbf{p}_{L_{n'}})\times\rfloor
\end{array}\right]$$

We here note that the first partition of the above matrix is already in upper triangular form, with $3n'$ rows. Proceeding further, we subtract the block row corresponding to the first observation of each feature, from all block rows corresponding to observations of the same feature. Thus,

$$\bar{\mathbf{H}}(k') \sim \left[\begin{array}{c|c}
\mathbf{I}_{3n'} & \mathbf{M}_0 \\
\hline
\mathbf{0} & \mathbf{M}_1 \\
\hline
\vdots & \vdots \\
\hline
\mathbf{0} & \mathbf{M}_{k'}
\end{array}\right] \tag{51}$$

$$\sim \left[\begin{array}{c|c}
\mathbf{I}_{3n'} & \mathbf{0} \\
\hline
\mathbf{0} & \mathbf{M}_1 \\
\hline
\vdots & \vdots \\
\hline
\mathbf{0} & \mathbf{M}_{k'}
\end{array}\right] = \begin{bmatrix} \mathbf{I}_{3n'} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix} \tag{52}$$

where $\mathbf{M}_i$ for $i = 1, \dots, k'$ is a $3l_i \times 6k'$ matrix, where each $3 \times 6k'$ block row has the following structure:

$$\mathbf{M}_{ij} = \begin{bmatrix} \mathbf{0}_{3\times6} & \dots & \overbrace{\mathbf{I}_3 \quad -\lfloor \mathbf{p}_{L_{j_0}} \times \rfloor}^{j_0\text{-th block}} & \dots & \mathbf{0}_{3\times6} & \dots & \overbrace{-\mathbf{I}_3 \quad \lfloor \mathbf{p}_{L_j} \times \rfloor}^{i\text{-th block}} & \dots & \mathbf{0}_{3\times6} \end{bmatrix} \tag{53}$$

where j_0 is the time-step at which the landmark was observed for the first time. From (52), we see that $\operatorname{rank}(\bar{\mathbf{H}}(k')) = 3n' + \operatorname{rank}(\mathbf{M})$. In what follows, we will use induction to show that $\operatorname{rank}(\mathbf{M}) = 6k'$, which immediately leads to the desired result. Specifically, we will show that if after applying a sequence of elementary row operations, the matrix $\mathbf{M}$ is shown to satisfy:

$$\mathbf{M} \sim \left[\begin{array}{cc|cc|c|c|cc|cc|cc|c}
\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots \\
\mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots \\
\hline
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots \\
\hline
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \ddots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \vdots & \ddots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots \\
\hline
\vdots & \vdots & \vdots & \vdots & \dots & \dots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\hline
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{I}_3 & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots \\
\hline
\multicolumn{13}{c}{\mathbf{M}_{\kappa+1}} \\
\multicolumn{13}{c}{\vdots} \\
\multicolumn{13}{c}{\mathbf{M}_{k'}}
\end{array}\right] \begin{array}{l} \left.\begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array}\right\} 6\kappa \text{ rows} \\ \\ \\ \\ \end{array} \tag{54}$$

then it must also satisfy

$$\mathbf{M} \sim \left[\begin{array}{c}
\left.\begin{array}{cc|cc|c|c|cc|cc|cc|c}
\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots \\
\mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \dots \\
\hline
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \dots \\
\hline
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \ddots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \vdots & \ddots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \dots \\
\hline
\vdots & \vdots & \vdots & \vdots & \dots & \dots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\hline
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \dots \\
\hline
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \dots
\end{array}\right\} 6(\kappa+1)\text{ rows} \\
\hline
\mathbf{M}_{\kappa+2} \\
\vdots \\
\mathbf{M}_{k'}
\end{array}\right] \tag{55}$$

Using this results, which we prove below, we conclude that

$$\mathbf{M} \sim \left[\begin{array}{cc|cc|c|c|cc|cc}
\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} \\
\mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 \\
\hline
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 \\
\hline
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \ddots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \vdots & \ddots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 \\
\hline
\vdots & \vdots & \vdots & \vdots & \dots & \dots & \vdots & \vdots & \vdots & \vdots \\
\hline
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{I}_3 & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & -\mathbf{I}_3
\end{array}\right] \tag{56}$$

which shows that $\mathrm{rank}(\mathbf{M}) = 6k'$, since the above matrix is in row-reduced form with $6k'$ non-zero rows.

To obtain a base case for the proof by induction, we examine the matrix $\mathbf{M}_1$. This matrix contains block rows that correspond to measurements of features at time-step 1 (except for those features that are first seen at time-step 1). All these features were first seen at time-step 0, and therefore $\mathbf{M}_1$ has the following structure (see (53))

$$\mathbf{M}_1 \sim \left[\begin{array}{cc|cc|cc|c}
\mathbf{I}_3 & -\lfloor(\mathbf{p}_{L_1})\times\rfloor & -\mathbf{I}_3 & \lfloor(\mathbf{p}_{L_1})\times\rfloor & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots \\
\vdots & \vdots & \vdots & \vdots & \multicolumn{2}{c|}{\vdots} & \vdots \\
\mathbf{I}_3 & -\lfloor(\mathbf{p}_{L_{l_0}})\times\rfloor & -\mathbf{I}_3 & \lfloor(\mathbf{p}_{L_{l_0}})\times\rfloor & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots
\end{array}\right]$$

Subtract the first block row from each block row below it in $\mathbf{M}_1$

$$= \left[\begin{array}{cc|cc|cc|c}
\mathbf{I}_3 & -\lfloor(\mathbf{p}_{L_1})\times\rfloor & -\mathbf{I}_3 & \lfloor(\mathbf{p}_{L_1})\times\rfloor & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots \\
\mathbf{0}_{3\times3} & \lfloor(\mathbf{p}_{L_1} - \mathbf{p}_{L_2})\times\rfloor & \mathbf{0}_{3\times3} & -\lfloor(\mathbf{p}_{L_1} - \mathbf{p}_{L_2})\times\rfloor & \multicolumn{2}{c|}{\vdots} & \vdots \\
\vdots & \vdots & \vdots & \vdots & \multicolumn{2}{c|}{\vdots} & \vdots \\
\mathbf{0}_{3\times3} & \lfloor(\mathbf{p}_{L_1} - \mathbf{p}_{L_{l_0}})\times\rfloor & \mathbf{0}_{3\times3} & -\lfloor(\mathbf{p}_{L_1} - \mathbf{p}_{L_{l_0}})\times\rfloor & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots
\end{array}\right]$$

$$= \left[\begin{array}{cc|cc|cc|c}
\mathbf{I}_3 & -\lfloor(\mathbf{p}_{L_1})\times\rfloor & -\mathbf{I}_3 & \lfloor(\mathbf{p}_{L_1})\times\rfloor & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots \\
\mathbf{0}_{3\times3} & \mathbf{V}_1 & \mathbf{0}_{3\times3} & -\mathbf{V}_1 & \multicolumn{2}{c|}{\vdots} & \vdots
\end{array}\right]$$

where

$$\mathbf{V}_1 = \begin{bmatrix} \lfloor(\mathbf{p}_{L_1} - \mathbf{p}_{L_2})\times\rfloor \\ \lfloor(\mathbf{p}_{L_1} - \mathbf{p}_{L_3})\times\rfloor \\ \vdots \\ \lfloor(\mathbf{p}_{L_1} - \mathbf{p}_{L_{l_0}})\times\rfloor \end{bmatrix} \tag{57}$$

We now show that the columns of $\mathbf{V}_1$ are linearly independent. We have that:

$$\begin{aligned} \mathbf{V}_1\mathbf{a} = \mathbf{0} \Leftrightarrow \mathbf{V}_1\mathbf{a} = &\begin{bmatrix} \lfloor(\mathbf{p}_{L_1} - \mathbf{p}_{L_2})\times\rfloor \\ \lfloor(\mathbf{p}_{L_1} - \mathbf{p}_{L_3})\times\rfloor \\ \vdots \\ \lfloor(\mathbf{p}_{L_1} - \mathbf{p}_{L_{l_0}})\times\rfloor \end{bmatrix} \mathbf{a} = \begin{bmatrix} \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} \\ \vdots \\ \mathbf{0}_{3\times1} \end{bmatrix} \\ \Leftrightarrow \quad & \lfloor(\mathbf{p}_{L_1} - \mathbf{p}_{L_j})\times\rfloor\mathbf{a} = \mathbf{0}_{3\times1}, \ \ j = 2, \ldots, l_0 \\ \Leftrightarrow \quad & \mathbf{a} = \mathbf{0} \quad \text{or} \quad \mathbf{a} = c_j(\mathbf{p}_{L_1} - \mathbf{p}_{L_j}), \ \ j = 2, \ldots, l_0, c_j \in \mathbb{R} \end{aligned} \tag{58}$$

The second case above ($\mathbf{a} = c_j(\mathbf{p}_{L_1} - \mathbf{p}_{L_j})$) requires all the features to be collinear. Thus, when at least three non-collinear features are observed, the only valid solution is $\mathbf{a} = \mathbf{0}$. Therefore, we have $\mathbf{V}_1\mathbf{a} = \mathbf{0} \Leftrightarrow \mathbf{a} = \mathbf{0}$, which implies that the three columns of $\mathbf{V}_1$ are linearly independent, and $\mathbf{V}_1$ can be reduced to the identity matrix $\mathbf{I}_3$ by elementary row operations (Gauss-Jordan elimination). After removing the zero rows, we have:

$$\mathbf{M}_1 \sim \left[\begin{array}{cc|cc|cc|c} \mathbf{I}_3 & -\lfloor(\mathbf{p}_{L_1})\times\rfloor & -\mathbf{I}_3 & \lfloor(\mathbf{p}_{L_1})\times\rfloor & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \cdots \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \vdots \end{array}\right] \tag{59}$$

$$\sim \left[\begin{array}{cc|cc|cc|c} \mathbf{I}_3 & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \cdots \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \vdots \end{array}\right] \tag{60}$$

We can therefore write

$$\mathbf{M} \sim \left[\begin{array}{ccccccc} \mathbf{I}_3 & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \cdots \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \vdots \\ \hline \multicolumn{7}{c}{\mathbf{M}_2} \\ \multicolumn{7}{c}{\vdots} \\ \multicolumn{7}{c}{\mathbf{M}_{k'}} \end{array}\right] \tag{61}$$

which confirms with the structure of Eq. (54), with $\kappa = 1$. Thus, we have a base-case for induction, and we next show that if (54) is true, then (55) must also hold. The matrix $\mathbf{M}_{\kappa+1}$ has the following structure:

$$\mathbf{M}_{\kappa+1} \sim \left[\begin{array}{c|cc|c|cc|cc|c} \cdots & \mathbf{I}_3 & -\lfloor(\mathbf{p}_{L_j})\times\rfloor & \cdots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \lfloor(\mathbf{p}_{L_j})\times\rfloor & \cdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots & \vdots \\ \cdots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \cdots & \mathbf{I}_3 & -\lfloor(\mathbf{p}_{L_{\ell_\kappa}})\times\rfloor & -\mathbf{I}_3 & \lfloor(\mathbf{p}_{L_{\ell_\kappa}})\times\rfloor & \cdots \end{array}\right]$$

Eliminate the block columns corresponding to $\mathbf{c}_{0:\kappa-1}$, by using the non-zero rows in the partitions $\mathbf{M}_{0:\kappa}$

$$\sim \left[\begin{array}{c|cc|c|cc|cc|c} \cdots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \cdots & \mathbf{I}_3 & -\lfloor(\mathbf{p}_{L_j})\times\rfloor & -\mathbf{I}_3 & \lfloor(\mathbf{p}_{L_j})\times\rfloor & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ \cdots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \cdots & \mathbf{I}_3 & -\lfloor(\mathbf{p}_{L_{\ell_\kappa}})\times\rfloor & -\mathbf{I}_3 & \lfloor(\mathbf{p}_{L_{\ell_\kappa}})\times\rfloor & \cdots \end{array}\right] \tag{62}$$

Proceed similar to $\mathbf{M}_1$, based on similar matrix structure

$$\sim \left[\begin{array}{c|cc|c|cc|cc|c} \cdots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \cdots & \mathbf{I}_3 & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \cdots \\ \cdots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \cdots & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \cdots \end{array}\right]$$

We thus see that $\mathbf{M}_{\kappa+1}$ has also been reduced to a reduced-row echelon form with six non-zero rows. Using the above result and substituting in $\mathbf{M}$ at this point, we can write:

$$\mathbf{M} \sim \left[\begin{array}{cc|cc|c|c|cc|cc|cc|c}
\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots \\
\mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots \\
\hline
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots \\
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots \\
\hline
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \ddots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots \\
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \vdots & \ddots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots \\
\hline
\vdots & \vdots & \vdots & \vdots & \cdots & \cdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\hline
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \cdots & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots \\
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots \\
\hline
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \cdots \\
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \cdots \\
\hline
\multicolumn{13}{c}{\mathbf{M}_{\kappa+2}} \\
\multicolumn{13}{c}{\vdots} \\
\multicolumn{13}{c}{\mathbf{M}_{k'}}
\end{array}\right] \tag{63}$$

Add block row coresponding to $\mathbf{M}_{\kappa+1}$ to all block rows $\mathbf{M}_{0:\kappa}$

$$\sim \left[\begin{array}{cc|cc|c|c|cc|cc|cc|c}
\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \cdots \\
\mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \cdots \\
\hline
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \cdots \\
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \cdots \\
\hline
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \ddots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \cdots \\
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \vdots & \ddots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \cdots \\
\hline
\vdots & \vdots & \vdots & \vdots & \cdots & \cdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\hline
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \cdots & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \cdots \\
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \cdots \\
\hline
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \cdots \\
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \cdots \\
\hline
\multicolumn{13}{c}{\mathbf{M}_{\kappa+2}} \\
\multicolumn{13}{c}{\vdots} \\
\multicolumn{13}{c}{\mathbf{M}_{k'}}
\end{array}\right] \tag{64}$$

which completes the proof. We have thus shown that:

$$\operatorname{rank}(\bar{\mathbf{H}}(k')) = 3n' + 6k'$$

This result holds as long as at each time-step, we observe at least three non-collinear features that were also seen by a previous camera state.

C Proof of (36)

We now show that $\operatorname{rank}(\bar{\mathbf{H}}(k, k')) = 3n' + 6k' + 3$. We consider the case where at time-step k the first m poses are marginalized, along with all landmarks that are only observed by the marginalized states. From (40), we have the following structure for the matrix $\bar{\mathbf{H}}(k, k')$:

$$\bar{\mathbf{H}}(k,k') = \left[\begin{array}{c|ccc|ccc} \bar{\mathbf{H}}_{L_0} & \bar{\mathbf{H}}_{C_0}(k) & \dots & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \bar{\mathbf{H}}_{L_{m-1}} & \mathbf{0} & \ddots & \bar{\mathbf{H}}_{C_{m-1}}(k) & \mathbf{0} & \dots & \mathbf{0} \\ \hline \bar{\mathbf{H}}_{L_m} & \mathbf{0} & \dots & \mathbf{0} & \bar{\mathbf{H}}_{C_m}(k') & \ddots & \mathbf{0} \\ \vdots & \vdots & \dots & \dots & \mathbf{0} & \ddots & \mathbf{0} \\ \bar{\mathbf{H}}_{L_{k'}} & \mathbf{0} & \dots & \dots & \dots & \mathbf{0} & \bar{\mathbf{H}}_{C_{k'}}(k') \end{array}\right] \tag{65}$$

As already discussed, the structure of this matrix is the same as that of $\bar{\mathbf{H}}(k')$, with the only difference being the state estimates used for evaluating Jacobians. Therefore, by applying the same sequence of column and row operations as in the preceding section, we can write (see (52)):

$$\bar{\mathbf{H}}(k,k') \sim \left[\begin{array}{c|c} \mathbf{I}_{3n'} & \mathbf{M}_0 \\ \hline \mathbf{0} & \mathbf{M}_1(k) \\ \vdots & \vdots \\ \mathbf{0} & \mathbf{M}_m(k) \\ \hline \mathbf{0} & \mathbf{M}_1(k,k') \\ \vdots & \vdots \\ \mathbf{0} & \mathbf{M}_{k'}(k,k') \end{array}\right] \tag{66}$$

$$\sim \begin{bmatrix} \mathbf{I}_{3n'} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}(k,k') \end{bmatrix} \tag{67}$$

The matrices $\mathbf{M}_i(k), i = 1, \dots, m-1$ have values identical to those shown in (53). On the other hand, the matrices $\mathbf{M}_i(k,k'), i = m, \dots, k'$ have the same structure, but may use different state estimates for the Jacobians. Specifically, if a feature was seen for the first time before time step m, then the block row corresponding to this feature's measurement at time-step $i > m$ will be:

$$\mathbf{M}_{ij} = \begin{bmatrix} \mathbf{0}_{3\times6} & \dots & \mathbf{I}_3 & -\lfloor \hat{\mathbf{p}}_{L_{j_0}}(k)\times\rfloor & \dots & \mathbf{0}_{3\times6} & \dots & -\mathbf{I}_3 & \lfloor \hat{\mathbf{p}}_{L_j}(k')\times\rfloor & \dots & \mathbf{0}_{3\times6} \end{bmatrix} \tag{68}$$

If the feature was seen for the first time after time step m, then

$$\mathbf{M}_{ij} = \begin{bmatrix} \mathbf{0}_{3\times6} & \dots & \mathbf{I}_3 & -\lfloor \hat{\mathbf{p}}_{L_{j_0}}(k')\times\rfloor & \dots & \mathbf{0}_{3\times6} & \dots & -\mathbf{I}_3 & \lfloor \hat{\mathbf{p}}_{L_j}(k')\times\rfloor & \dots & \mathbf{0}_{3\times6} \end{bmatrix} \tag{69}$$

Since the first m block matrices in $\mathbf{M}(k, k')$ are the same as those in the preceding section, the result of (54) applies, and we can write

$$\mathbf{M}(k,k') \sim \left[\begin{array}{c} \left.\begin{array}{cc|cc|c|c|cc|cc|cc|c} \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots \\ \hline \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots \\ \hline \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \ddots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \vdots & \ddots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots \\ \hline \vdots & \vdots & \vdots & \vdots & \dots & \dots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \hline \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{I}_3 & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots \end{array}\right\} 6m \text{ rows} \\ \hline \mathbf{M}_m(k,k') \\ \vdots \\ \mathbf{M}_{k'}(k,k') \end{array}\right] \tag{70}$$

All block rows in $\mathbf{M}_m(k,k')$ correspond to measurements of features that were first seen in one of the marginalized states. Therefore, all rows in $\mathbf{M}_m(k,k')$ are of the form (68). As a result, by applying the same elimination procedure as in (62), we obtain:

$$\mathbf{M}_m(k,k') \sim \left[\begin{array}{ccc|cc|cc|ccc} \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & -\lfloor(\hat{\mathbf{p}}_{L_{j_1}}(k))\times\rfloor & -\mathbf{I}_3 & \lfloor(\hat{\mathbf{p}}_{L_{j_1}}(k'))\times\rfloor & \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & -\lfloor(\hat{\mathbf{p}}_{L_{j_2}}(k))\times\rfloor & -\mathbf{I}_3 & \lfloor(\hat{\mathbf{p}}_{L_{j_2}}(k'))\times\rfloor & \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & -\lfloor(\hat{\mathbf{p}}_{L_{j_{\ell_m}}}(k))\times\rfloor & -\mathbf{I}_3 & \lfloor(\hat{\mathbf{p}}_{L_{j_{\ell_m}}}(k'))\times\rfloor & \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} \end{array}\right] \tag{71}$$

Subtract the first block row from each block row

$$\sim \left[\begin{array}{ccc|cc|cc|ccc} \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & -\lfloor(\hat{\mathbf{p}}_{L_{j_1}}(k))\times\rfloor & -\mathbf{I}_3 & \lfloor(\hat{\mathbf{p}}_{L_{j_1}}(k'))\times\rfloor & \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \lfloor(\hat{\mathbf{p}}_{L_{j_1}}(k)-\hat{\mathbf{p}}_{L_{j_2}}(k))\times\rfloor & \mathbf{0}_{3\times 3} & -\lfloor(\hat{\mathbf{p}}_{L_{j_1}}(k')-\hat{\mathbf{p}}_{L_{j_2}}(k'))\times\rfloor & \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \lfloor(\hat{\mathbf{p}}_{L_{j_1}}(k)-\hat{\mathbf{p}}_{L_{j_{\ell_m}}}(k))\times\rfloor & \mathbf{0}_{3\times 3} & -\lfloor(\hat{\mathbf{p}}_{L_{j_1}}(k')-\hat{\mathbf{p}}_{L_{j_{\ell_m}}}(k'))\times\rfloor & \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} \end{array}\right]$$

$$= \left[\begin{array}{ccc|cc|cc|ccc} \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & -\lfloor(\hat{\mathbf{p}}_{L_{j_1}}(k))\times\rfloor & -\mathbf{I}_3 & \lfloor(\hat{\mathbf{p}}_{L_{j_1}}(k'))\times\rfloor & \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{V}_m & \mathbf{0}_{3\times 3} & -\mathbf{V}'{}_m & \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} \end{array}\right]$$

where

$$\mathbf{V}_m = \begin{bmatrix} \lfloor(\hat{\mathbf{p}}_{L_{j_1}}(k)-\hat{\mathbf{p}}_{L_{j_2}}(k))\times\rfloor \\ \lfloor(\hat{\mathbf{p}}_{L_{j_1}}(k)-\hat{\mathbf{p}}_{L_{j_3}}(k))\times\rfloor \\ \vdots \\ \lfloor(\hat{\mathbf{p}}_{L_{j_1}}(k)-\hat{\mathbf{p}}_{L_{j_{\ell_m}}}(k))\times\rfloor \end{bmatrix}, \quad \mathbf{V}'_m = \begin{bmatrix} \lfloor(\hat{\mathbf{p}}_{L_{j_1}}(k')-\hat{\mathbf{p}}_{L_{j_2}}(k'))\times\rfloor \\ \lfloor(\hat{\mathbf{p}}_{L_{j_1}}(k')-\hat{\mathbf{p}}_{L_{j_3}}(k'))\times\rfloor \\ \vdots \\ \lfloor(\hat{\mathbf{p}}_{L_{j_1}}(k')-\hat{\mathbf{p}}_{L_{j_{\ell_m}}}(k'))\times\rfloor \end{bmatrix} \tag{72}$$

In general, the differences $(\hat{\mathbf{p}}_{L_{j_1}}(k)-\hat{\mathbf{p}}_{L_j}(k))$ and $(\hat{\mathbf{p}}_{L_{j_1}}(k')-\hat{\mathbf{p}}_{L_j}(k'))$ with $j = j_2, j_3, \dots, j_{\ell_m}$ are arbitrary vectors. It is straightforward to show the following matrix is full column-ranked for arbitrary vectors $\mathbf{p}_i, i = 1, \dots, 6$:

$$\operatorname{rank}\left(\begin{bmatrix} \lfloor(\mathbf{p}_1)\times\rfloor & \lfloor\mathbf{p}_2\times\rfloor \\ \lfloor(\mathbf{p}_3)\times\rfloor & \lfloor\mathbf{p}_4\times\rfloor \\ \lfloor(\mathbf{p}_5)\times\rfloor & \lfloor\mathbf{p}_6\times\rfloor \end{bmatrix}\right) = 6 \tag{73}$$

Therefore, when we have multiple features observed at time-step m ($\ell_m \geq 4$):

$$\operatorname{rank}\left([\mathbf{V}_3 \;\; \mathbf{V}'_3]\right) = \operatorname{rank}\left(\begin{bmatrix} \lfloor(\hat{\mathbf{p}}_{L_{j_1}}(k)-\hat{\mathbf{p}}_{L_{j_2}}(k))\times\rfloor & \lfloor(\hat{\mathbf{p}}_{L_{j_1}}(k')-\hat{\mathbf{p}}_{L_{j_2}}(k'))\times\rfloor \\ \lfloor(\hat{\mathbf{p}}_{L_{j_1}}(k)-\hat{\mathbf{p}}_{L_{j_3}}(k))\times\rfloor & \lfloor(\hat{\mathbf{p}}_{L_{j_1}}(k')-\hat{\mathbf{p}}_{L_{j_3}}(k'))\times\rfloor \\ \vdots & \vdots \\ \lfloor(\hat{\mathbf{p}}_{L_{j_1}}(k)-\hat{\mathbf{p}}_{L_{j_{\ell_m}}}(k))\times\rfloor & \lfloor(\hat{\mathbf{p}}_{L_{j_1}}(k')-\hat{\mathbf{p}}_{L_{j_{\ell_m}}}(k'))\times\rfloor \end{bmatrix}\right) = 6 \tag{74}$$

which shows that the columns of $[\mathbf{V}_3 \;\; \mathbf{V}'_3]$ are linearly independent. Consequently, we can apply elementary row operations to obtain the following form, after removing zero block rows:

$$\mathbf{M}_m(k,k') \sim \left[\begin{array}{ccc|cc|cc|ccc} \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & -\lfloor(\hat{\mathbf{p}}_{L_{j_1}}(k))\times\rfloor & -\mathbf{I}_3 & \lfloor(\hat{\mathbf{p}}_{L_{j_1}}(k'))\times\rfloor & \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} \end{array}\right] \tag{75}$$

Add the second block row, multiplied with $\lfloor(\hat{\mathbf{p}}_{L_{j_1}}(k))\times\rfloor$, and the third block row,
multiplied with $-\lfloor(\hat{\mathbf{p}}_{L_{j_1}}(k'))\times\rfloor$, to the first block row.

$$\sim \left[\begin{array}{ccc|cc|cc|ccc} \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} \end{array}\right]$$

$$\sim \left[\begin{array}{ccc|cc|cc|ccc} \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \dots & \mathbf{0}_{3\times 3} \end{array}\right] \tag{76}$$

It is important to note that due to the presence of two estimates $\hat{\mathbf{p}}_{L_j}(k)$ and $\hat{\mathbf{p}}_{L_j}(k')$ for the feature states, the partition is reduced to 9 non-zero rows, instead of 6 as usual. We will now use the result to prove, by induction, that $\mathrm{rank}(\mathbf{M}(k,k')) = 6k' + 3$. Specifically, we show that if for $\kappa \geq m$:

$$\mathbf{M}(k,k') \sim \left[\begin{array}{c} \left.\begin{array}{cc|cc|c|c|cc|cc|cc|c}
\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots \\
\mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots \\ \hline
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots \\ \hline
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \ddots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \vdots & \ddots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots \\ \hline
\vdots & \vdots & \vdots & \vdots & \dots & \dots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \hline
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{I}_3 & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots
\end{array}\right\} 6\kappa+3 \text{ rows} \\ \hline
\mathbf{M}_{\kappa+1}(k,k') \\ \vdots \\ \mathbf{M}_{k'}(k,k')
\end{array}\right] \tag{77}$$

then it is

$$\mathbf{M}(k,k') \sim \left[\begin{array}{c} \left.\begin{array}{cc|cc|c|c|cc|cc|cc|c}
\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots \\
\mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \dots \\ \hline
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \dots \\ \hline
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \ddots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \vdots & \ddots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \dots \\ \hline
\vdots & \vdots & \vdots & \vdots & \dots & \dots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \hline
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \dots \\ \hline
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \dots \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \dots
\end{array}\right\} 6(\kappa+1)+3 \text{ rows} \\ \hline
\mathbf{M}_{\kappa+2} \\ \vdots \\ \mathbf{M}_{k'}
\end{array}\right] \tag{78}$$

The base case for induction arises from the result in (76). Specifically, if we add the first two block rows of the matrix in (76) to all the block rows corresponding to $\mathbf{M}_i(k), i = 1, \dots, m$, we obtain a matrix as in (77), with $\kappa = m$. We thus need to show that (77) implies (78). We start by following the same steps as in the preceding section (see (62)), to show that:

Eliminate the block columns corresponding to $\mathbf{c}_{0:\kappa-1}$, by using the non-zero rows in the partitions $\mathbf{M}_{0:\kappa}$

$$\mathbf{M}_{\kappa+1}(k,k') \sim \left[\begin{array}{c|cc|c|cc|cc|c}
\dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \mathbf{I}_3 & -\lfloor(\hat{\mathbf{p}}_{L_{j_1}})\times\rfloor & -\mathbf{I}_3 & \lfloor(\hat{\mathbf{p}}_{L_{j_1}})\times\rfloor & \dots \\
\dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \mathbf{I}_3 & -\lfloor(\hat{\mathbf{p}}_{L_{j_2}})\times\rfloor & -\mathbf{I}_3 & \lfloor(\hat{\mathbf{p}}_{L_{j_2}})\times\rfloor & \dots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \mathbf{I}_3 & -\lfloor(\hat{\mathbf{p}}_{L_{\ell_\kappa}})\times\rfloor & -\mathbf{I}_3 & \lfloor(\hat{\mathbf{p}}_{L_{\ell_\kappa}})\times\rfloor & \dots
\end{array}\right]$$

Use the last block row of (77) to eliminate the corresponding column

$$\sim \left[\begin{array}{c|cc|c|cc|cc|c}
\cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \lfloor(\hat{\mathbf{p}}_{L_{j_1}})\times\rfloor & \cdots \\
\cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \lfloor(\hat{\mathbf{p}}_{L_{j_2}})\times\rfloor & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \lfloor(\hat{\mathbf{p}}_{L_{\ell_\kappa}})\times\rfloor & \cdots
\end{array}\right]$$

Subtract the first block row from all the remaining ones

$$\sim \left[\begin{array}{c|cc|c|cc|cc|c}
\cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \lfloor(\hat{\mathbf{p}}_{L_{j_1}})\times\rfloor & \cdots \\
\cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \lfloor(\hat{\mathbf{p}}_{L_{j_2}} - \hat{\mathbf{p}}_{L_{j_1}})\times\rfloor & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \lfloor(\hat{\mathbf{p}}_{L_{\ell_\kappa}} - \hat{\mathbf{p}}_{L_{j_1}})\times\rfloor & \cdots
\end{array}\right]$$

$$\sim \left[\begin{array}{c|cc|c|cc|cc|c}
\cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \lfloor(\hat{\mathbf{p}}_{L_{j_1}})\times\rfloor & \cdots \\
\cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{V}_{\kappa+1} & \cdots
\end{array}\right]$$

Perform Gauss-Jordan elimination on $\mathbf{V}_{\kappa+1}$

$$\sim \left[\begin{array}{c|cc|c|cc|cc|c}
\cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \lfloor(\hat{\mathbf{p}}_{L_{j_1}})\times\rfloor & \cdots \\
\cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \cdots
\end{array}\right]$$

Multiply the second block row with $\lfloor(\hat{\mathbf{p}}_{L_{j_1}})\times\rfloor$, and add to the first block row

$$\sim \left[\begin{array}{c|cc|c|cc|cc|c}
\cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \cdots \\
\cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \cdots
\end{array}\right]$$

Add the last block row of (77) to the last block row

$$\sim \left[\begin{array}{c|cc|c|cc|cc|c}
\cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \cdots \\
\cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \cdots
\end{array}\right]$$

We have thus shown that:

$$\mathbf{M}(k,k') \sim \left[\begin{array}{cc|cc|c|c|cc|cc|cc|c}
\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots \\
\mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots \\
\hline
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots \\
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots \\
\hline
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \ddots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots \\
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \vdots & \ddots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots \\
\hline
\vdots & \vdots & \vdots & \vdots & \cdots & \cdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\hline
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \cdots & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots \\
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots \\
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots \\
\hline
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \cdots \\
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \cdots & \cdots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \cdots \\
\hline
\multicolumn{13}{c}{\mathbf{M}_{\kappa+2}} \\
\multicolumn{13}{c}{\vdots} \\
\multicolumn{13}{c}{\mathbf{M}_{k'}}
\end{array}\right] \tag{79}$$

Re-order rows

$$\sim \left[\begin{array}{cc|cc|c|c|cc|cc|cc|c}
\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots & \dots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots \\
\mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots & \dots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots \\
\hline
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \dots & \dots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots \\
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \dots & \dots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots \\
\hline
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \ddots & \dots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots \\
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \vdots & \ddots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots \\
\hline
\vdots & \vdots & \vdots & \vdots & \dots & \dots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\hline
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots & \dots & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots \\
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots & \dots & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots \\
\hline
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots & \dots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \dots \\
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots & \dots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \dots \\
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots & \dots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots \\
\hline
\multicolumn{13}{c}{\mathbf{M}_{\kappa+2}} \\
\multicolumn{13}{c}{\vdots} \\
\multicolumn{13}{c}{\mathbf{M}_{k'}}
\end{array}\right] \tag{80}$$

Add block row coresponding to $\mathbf{M}_{\kappa+1}$ to all block rows $\mathbf{M}_{0:\kappa}$

$$\sim \left[\begin{array}{cc|cc|c|c|cc|cc|cc|c}
\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots & \dots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \dots \\
\mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots & \dots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \dots \\
\hline
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \dots & \dots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \dots \\
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \dots & \dots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \dots \\
\hline
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \ddots & \dots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \dots \\
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \vdots & \ddots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \dots \\
\hline
\vdots & \vdots & \vdots & \vdots & \dots & \dots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\hline
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots & \dots & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \dots \\
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots & \dots & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \dots \\
\hline
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots & \dots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \mathbf{0}_{3\times 3} & \dots \\
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots & \dots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & -\mathbf{I}_3 & \dots \\
\mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots & \dots & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \dots \\
\hline
\multicolumn{13}{c}{\mathbf{M}_{\kappa+2}} \\
\multicolumn{13}{c}{\vdots} \\
\multicolumn{13}{c}{\mathbf{M}_{k'}}
\end{array}\right] \tag{81}$$

Subtract the second last block row from the last one

$$\sim \left[\begin{array}{cc|cc|c|c|cc|cc|cc|c}
\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots \\
\mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \dots \\
\hline
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \dots \\
\hline
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \ddots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \vdots & \ddots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \dots \\
\hline
\vdots & \vdots & \vdots & \vdots & \dots & \dots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\hline
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \dots \\
\hline
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \dots \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \dots \\
\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \dots & \dots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \dots \\
\hline
\multicolumn{13}{c}{\mathbf{M}_{\kappa+2}} \\
\multicolumn{13}{c}{\vdots} \\
\multicolumn{13}{c}{\mathbf{M}_{k'}}
\end{array}\right] \tag{82}$$

which completes the proof. By using the above result, we can show that $\mathrm{rank}(\mathbf{M}(k,k')) = 6k' + 3$. Therefore, using (67), we conclude that:

$$\mathrm{rank}(\bar{\mathbf{H}}(k,k')) = 3n' + \mathrm{rank}(\mathbf{M}(k,k')) = 3n' + 6k' + 3 \tag{83}$$

References

[1] N. Trawny, A. I. Mourikis, S. I. Roumeliotis, A. E. Johnson, and J. Montgomery, "Vision-aided inertial navigation for pin-point landing using observations of mapped landmarks," *Journal of Field Robotics*, vol. 24, no. 5, pp. 357–378, May 2007.

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

[3] A. Ansar and K. Daniilidis, "Linear pose estimation from points or lines," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 578–589, May 2003.

[4] A. J. Davison and D. W. Murray, "Simultaneous localisation and map-building using active vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 865–880, Jul. 2002.

[5] A. Davison, I. Reid, N. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[6] J. Montiel, J. Civera, and A. Davison, "Unified inverse depth parametrization for monocular SLAM," in *Proceedings of Robotics: Science and Systems*, Philadelphia, PA, Aug. 16-19 2006, pp. 81–88.

[7] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, Washington, DC, June 17-26 2004, pp. 652–659.

[8] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," in *Proc. Intl. Conf. on Intelligent Robots and Systems*, Nice, France, Sept. 22-26 2008, pp. 3946–3952.

[9] C. Engels, H. Stewenius, and D. Nister, "Bundle adjustment rules," in *Proc. Photogrammetric Computer Vision Conf.*, Bonn, Germany, Sep. 20-22 2006, pp. 266–271.

[10] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *Journal of Field Robotics*, vol. 23, no. 1, pp. 3–20, Jan. 2006.

[11] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, "Real time localization and 3D reconstruction," in *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, New York, NY, June 17-22 2006, pp. 363– 370.

[12] L. H. Matthies, “Dynamic stereo vision,” Ph.D. dissertation, School of Computer Science, Carnegie Mellon University, 1989.

[13] S. I. Roumeliotis, A. E. Johnson, and J. F. Montgomery, “Augmenting inertial navigation with image-based motion estimation,” in *Proc. IEEE Intl. Conf. on Robotics and Automation*, Washington D.C, May 2002, pp. 4326–33.

[14] D. D. Diel, “Stochastic constraints for vision-aided inertial navigation,” Master’s thesis, Massachusetts Institute of Technology, Jan. 2005.

[15] B. Triggs, P. McLauchlan, R. Hartley, and Fitzgibbon, “Bundle adjustment – a modern synthesis,” in *Vision Algorithms: Theory and Practice*. Springer Verlag, 2000, pp. 298–375.

[16] M. Kaess, A. Ranganathan, and F. Dellaert, “iSAM: Incremental smoothing and mapping,” *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.

[17] K. Konolige, M. Agrawal, and J. Sola, “Large-scale visual odometry for rough terrain,” in *Proc. Intl. Symposium on Research in Robotics*, Hiroshima, Japan, Nov. 26-29 2007, pp. 201–212.

[18] G. Sibley, L. Matthies, and G. Sukhatme, “Sliding window filter with application to planetary landing,” *Journal of Field Robotics*, vol. 27, no. 5, pp. 587–608, 2010.

[19] T.-C. Dong-Si and A. I. Mourikis, “Motion tracking with fixed-lag smoothing: Algorithm and consistency analysis,” in *IEEE International Conference on Robotics and Automation*, Shanghai, China, May 9-13 2011, pp. 5655–5662.

[20] P. McLauchlan, “The variable state dimension filter applied to surface-based structure from motion,” School of Electrical Engineering, Information Technology and Mathematics, University of Surrey, UK, Tech. Rep. VSSP-TR-4/99, 1999.

[21] A. Ranganathan, M. Kaess, and F. Dellaert, “Fast 3D pose estimation with out-of-sequence measurements,” in *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, San Diego, CA, Oct. 29 - Nov. 2 2007, pp. 2486–2493.

[22] K. Konolige and M. Agrawal, “FrameSLAM: From bundle adjustment to real-time visual mapping,” *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1066 –1077, oct. 2008.

[23] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, 2001.

[24] N. Trawny and S. I. Roumeliotis, “Indirect Kalman filter for 3D pose estimation,” Dept. of Computer Science & Engineering, University of Minnesota, Minneapolis, MN, Tech. Rep. 2005-002, Mar. 2005.

[25] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. SIAM, 2000.

[26] M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman, “The New College vision and laser data set,” *The International Journal of Robotics Research*, vol. 28, no. 5, pp. 595–599, May 2009. [Online]. Available: http://www.robots.ox.ac.uk/NewCollegeData/

[27] C. Harris and M. Stephens, “A combined corner and edge detector,” in *Proc. 4th Alvey Vision Conf.*, Manchester, UK, Aug. 31 - Sep. 2 1988, pp. 147–151.

[28] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid, “RSLAM: A system for large-scale mapping in constant-time using stereo,” *International Journal of Computer Vision*, vol. 94, no. 2, pp. 198–214, 2011.