Robocentric Visual-Inertial Odometry

Zheng Huai - zhuai@udel.edu
Guoquan Huang - ghuang@udel.edu

Department of Mechanical Engineering
University of Delaware, Delaware, USA

Robot Perception and Navigation Group (RPNG)
Tech Report - RPNG-2018-RVIO
Last Updated - April 4, 2018
## Contents

1 Introduction .............................................. 1

2 Robocentric VIO ......................................... 2
   2.1 State Vector ........................................ 2
   2.2 Propagation ......................................... 3
      2.2.1 Continuous Time Error-State Model ............ 3
      2.2.2 Discrete Time State Propagation ............... 6
      2.2.3 Discrete Time Covariance Propagation ......... 7
   2.3 Update .............................................. 8
      2.3.1 Bearing-only Model ......................... 8
      2.3.2 Inverse Depth Bearing-only Model .......... 10
      2.3.3 EKF Update .................................. 13
      2.3.4 State Augmentation ......................... 14
   2.4 Composition ....................................... 14
   2.5 Observability Analysis ............................. 15
      2.5.1 Analytic State Transition Matrix ............ 16
      2.5.2 Measurement Jacobian ....................... 17
      2.5.3 Propagation and Update ....................... 17
      2.5.4 Composition ................................ 18
   2.6 Initialization ..................................... 20

3 Simulation Results ...................................... 20

4 Experimental Results .................................. 20
   4.1 Highly Dynamic Capability ........................ 22
   4.2 Long-Term Sustainability .......................... 22
   4.3 Versatility ........................................ 22

5 Conclusion ............................................ 22

Appendix A Analytic State Transition Matrix ................. 25

Appendix B Bundle Adjustment with Euclidean Coordinates 27

Appendix C Bundle Adjustment with Inverse Depth Parameters 30

References ............................................. 31
Abstract

In this report, we propose a novel robocentric formulation of visual-inertial navigation systems (VINS) within a multi-state constraint Kalman filter (MSCKF) framework and develop an efficient, lightweight, robocentric visual-inertial odometry (R-VIO) algorithm for consistent localization in challenging environments using only monocular vision. The key idea of the proposed approach is to deliberately reframe the 3D VINS with respect to a moving local frame (i.e., robocentric), rather than a fixed global frame of reference as in the standard world-centric VINS, and instead utilize high-accuracy relative motion estimates for global pose update. As an immediate advantage of using this robocentric formulation, the proposed R-VIO can start from an arbitrary pose, without the need to align its orientation with the global gravity vector. More importantly, we analytically show that the proposed robocentric EKF-based VINS does not undergo the observability mismatch issue as in the standard world-centric framework which was identified as the main cause of estimation inconsistency. The proposed R-VIO is extensively tested through both Monte Carlo simulations and the real-world experiments with different sensor platforms in different environments, and shown to achieve a competitive performance with the state-of-the-art VINS algorithms in terms of consistency and accuracy.

1 Introduction

Enabling high-precision 3D navigation on mobile devices and robots with minimal sensing of low-cost sensors holds potentially huge implications in the real applications ranging from mobile augmented reality to autonomous driving. To this end, inertial navigation offers a classical 3D localization solution which utilizes an inertial measurement unit (IMU) measuring the 3 degree-of-freedom (DOF) angular velocity and linear acceleration of the sensor platform on which it is rigidly attached. Typically, IMU works at a high frequency (e.g., 100Hz∼1000Hz) that enable it to sense highly dynamic motion, however, due to the corrupting sensor noise and bias, purely integrating IMU measurements can easily result in unusable motion estimates. This necessitates to fuse the aiding information from at least a single camera to reduce the accumulated inertial navigation drifts, which comes into the visual-inertial navigation systems (VINS).

In the past decade, we have witnessed significant progress on VINS, including both visual-inertial SLAM (VI-SLAM) and visual-inertial odometry (VIO), and many different algorithms have been developed (e.g., [1, 2, 3, 4, 5, 6, 7, 8] and references therein). However, almost all these algorithms are based on the standard world-centric formulation – that is, to estimate the absolute motion with respect to a fixed global frame of reference, such as the earth-centered earth-fixed (ECEF) or the north-east-down (NED) frame. In order to achieve accurate localization, such world-centric VINS often require a particular initialization procedure to estimate the starting pose in the global frame of reference, which, however, is hard to guarantee the accuracy in some cases (e.g., quick start, or poor vision). While an extended Kalman filter (EKF)-based world-centric VINS algorithm has the advantage of lower computational cost [1, 4] in comparing to the batch optimization-based ones (which incur high computation due to performing iterated linearization [5, 6]), it may become inconsistent, primarily due to the fact that the EKF linearized systems have different observability properties from the corresponding underlying nonlinear systems [9, 10, 4]. The remedies for mitigating this issue include enforcing the correct observability [4, 11, 12] or employing an invariant error representation [13]. Therefore, one may ask: Do we have to formulate VINS in the standard world-centric form? The answer is no. Intuitively, inspired by how humans navigate – we may not remember the starting pose after traveling a long distance while knowing well the relative motion within a recent, short time interval – we might relax the fixed global frame of VINS, instead, choose a moving local frame to better estimate relative motion which can be used later for global pose update.

Note that, this sensor-centered idea for localization can be traced back to 2D laser-based robocentric mapping [14], where the global frame is treated as a feature being observed from the moving robot frame and the relative motion measurements from an odometer are fused for pose update, while the composition step makes it possible to shift the local frame of reference during the motion. Following the similar idea, in [15] a camera-centered formulation shows the potential to fuse the visual information with the measurements from proprioceptive sensors (e.g., angular and linear velocity measurements). Both methods had been applied to the EKF-based SLAM while performing mapping in the local frames, thus limiting the global uncertainty and improving the estimation consistency. It should also be noted that a robust VINS algorithm using a different robocentric formulation and sensor-fusion scheme was recently introduced [16]. In particular, its
state vector includes both IMU states and the features which are reformulated with respect to the local IMU frame, while the camera and IMU measurements are fused in a direct fashion. Additionally, in contrast to [14, 15], it employs the iterated EKF to perform update, without the composition step to shift the local frame of reference.

In this report, following the idea [14, 15], we reformulate the VINS in 3D with respect to a local IMU frame, while in contrast to [14, 15, 16] which keep features in the state vector and have to concern the increasing computational cost as more features are observed and included, we focus on an EKF-based visual-inertial odometry (EKF-VIO) framework, i.e., multi-state constraint Kalman filter (MSCKF) [1], whose stochastic cloning enables the VINS to process hundreds of features while only keep a small number of robot poses from which the features have been observed, in the state vector, thus significantly reducing the computational cost. However, as studied in [4], the world-centric MSCKF is inconsistent. To enable consistent MSCKF-based 3D localization, a novel lightweight, robocentric VIO algorithm (R-VIO) is proposed in this report with the following key points:

- The global frame has been treated as a feature which involves the gravity effect, while the local frame of reference is shifted at every image time through a composition step.
- The relative motion estimates used for global updates are obtained by tightly fusing the camera and IMU measurements in a local frame of reference, for which instead of the features, a sliding relative pose window is included in the state vector to aid the estimation.
- A tailored inverse-depth measurement model is developed, which allows for fusing bearing information provided by the distant features, especially at motionless state.
- A constant unobservable subspace is analytically shown with the proposed robocentric formulation, which is independent of the linearization points while possessing correct dimensions and desired unobservable directions.

We perform extensive tests on both Monte Carlo simulations and the real-world experiments running real data of different sensor platforms from the micro aerial vehicle (MAV) flying indoor to the hand-held device navigating outdoor. All real-time results validate the superior performance of the proposed R-VIO algorithm.

## 2 Robocentric VIO

### 2.1 State Vector

In the proposed R-VIO, the robot frame \( \{ R \} \) is chosen as the immediate frame of reference for state estimation. For robot equipped with an IMU, the corresponding frame \( \{ I \} \) is usually considered to be aligned with the robot frame. As a result, the global frame \( \{ G \} := \{ R_0 \} \), the first robot frame) turns into a "moving" object with respect to \( \{ R \} \), and as compared to the world-centric case (e.g. [1]), we have a sequential frames of reference as the robot is moving.

At time-step \( \tau \in [t_k, t_{k+1}] \) the robocentric state with respect to \( \{ R_k \} \) (set at time-step \( k \)) consists of the global state and the IMU state, where the global state maintains the information of the start pose, while the IMU state characterizes the motion from \( \{ R_k \} \) to the current IMU pose, which is defined by the vector:

\[
R_k \mathbf{x}_r = \begin{bmatrix} R_k \mathbf{x}_G^\top & R_k \mathbf{x}_I^\top \end{bmatrix}^\top \\
= \begin{bmatrix} k \mathbf{q}^\top & R_k \mathbf{p}_G^\top & R_k \mathbf{g}^\top & \mathbf{t}_k \mathbf{q}^\top & R_k \mathbf{p}_I^\top & \mathbf{v}_I^\top & \mathbf{b}_{b_{GR}}^\top & \mathbf{b}_{b_{IR}}^\top \end{bmatrix}^\top
\]  

\( ^1 \)Throughout this report, \( k, k + 1, \ldots \) indicate the image time-steps, while \( \tau, \tau + 1, \ldots \) are the IMU time-steps between every two consecutive images. \( \{ I \} \) and \( \{ C \} \) denote the IMU frame and camera frame, respectively. \( \{ R \} \) is the robocentric frame of reference which is selected with the corresponding IMU frame at every image time-step. The subscript \( \ell i \) refers to the estimate of a quantity at time-step \( \ell \), after all measurements up to time-step \( i \) have been processed. \( \hat{x} \) is used to denote the estimate of a random variable \( x \), while \( \hat{x} = x - \hat{x} \) is the additive error in this estimate. \( \mathbf{I}_n \) and \( \mathbf{0}_n \) are the \( n \times n \) identity and zero matrices, respectively. Finally, the left superscript denotes the frame of reference with respect to which the vector is expressed.
where $k\hat{q}$ is the unit quaternion [17] describing the rotation from $\{R_k\}$ to $\{G\}$, $R_k\mathbf{p}_G$ is the position of $\{G\}$ in $\{R_k\}$, $\hat{q}$ and $R_k\mathbf{p}_{I_r}$ are the relative rotation and translation of robot from $\{R_k\}$ to $\{I_r\}$, and $\mathbf{v}_{I_r}$ is the robot velocity in the frame $\{I_r\}$. In particular, the local gravity vector is included which implicitly encodes the orientation of $\{R_k\}$.

Following Eq. (2), the error-state vector is defined as:

$$
R_k\hat{\mathbf{x}}_r = \left[ R_k\hat{\mathbf{x}}_G^T \ R_k\hat{\mathbf{x}}_{I_r}^T \right]^T
$$

where, in particular, if the quaternion estimate is $\hat{q}$, then the error quaternion $\delta\hat{q}$ is defined by:

$$
\delta\hat{q} = \delta\hat{q} \otimes \hat{q}
$$

where $\otimes$ denotes the quaternion multiplication, $\delta\theta$ is 3 degree-of-freedom (DOF) error angle associated to the error quaternion, and $C(\cdot)$ represents a $3 \times 3$ rotation matrix with $[\cdot \times]$ the skew-symmetric operator.

Assuming that at time-step $k$ when $\{I_k\}$ is set to be the frame of reference (i.e., $\{R_k\}$), a window of the relative poses between the last $N$ robocentric frames of reference is included in the state vector:

$$
\hat{x}_k = \left[ R_k\hat{\mathbf{x}}_G^T \ \hat{\mathbf{w}}_k^T \right]^T
$$

where $i^{-1}\hat{\mathbf{p}}_{R_i}$ are cloned from the IMU state estimate $R_{i^{-1}}\hat{x}_{R_i}$, $i \in \{2, \ldots, N\}$ (cf. Section 2.3.4). The error-state vector is defined according to:

$$
\hat{x}_k = \left[ R_k\hat{\mathbf{x}}_G^T \ \hat{\mathbf{w}}_k^T \right]^T
$$

Specifically, if $i \in \{k-N+1, k-N+2, \ldots, k\}$, then this window represents a short memory of the system motion. With such interpretation, everything happens at time-step $k$ could be correlated with the previous time-steps in the window, which in next is shown in the design of the measurement model (cf. Section 2.3). In our implementation, we manage this in a sliding-window fashion, i.e., delete the oldest one when a new relative pose is cloned, to save the computational cost.

### 2.2 Propagation

#### 2.2.1 Continuous Time Error-State Model

The typical measurements of IMU consist of the gyroscope and accelerometer measurements, $\omega_m$ and $a_m$, which are respectively given as:

$$
\omega_m = \omega + b_g + n_g
$$

$$
a_m = l^g a + l^g g + b_a + n_a
$$

where $\omega$ and $l^a$ are the angular velocity and linear acceleration in the IMU frame, $n_g$ and $n_a$ are the zero-mean white Gaussian sensor noise, $b_g$ and $b_a$ are the gyroscope and accelerometer biases which are modeled as random walk processes, driven by the zero-mean white Gaussian noise $n_{wg}$ and $n_{wa}$, respectively, and $l^g$ denotes the local gravity vector, whose global counterpart is constant (e.g., $Gg = [0,0,9.8]^T$, if $\{G\}$ is an earth-centered frame).

The continuous time IMU model with respect to the frame of reference $\{R_k\}$ is shown as follows:

$$
\dot{\hat{\mathbf{q}}}_k = \frac{1}{2} \Omega(\omega) \hat{\mathbf{q}}_k
$$

$$
R_k \hat{\mathbf{p}}_{I_r} = C(k\hat{q})^T \mathbf{v}_{I_r}
$$
\[ \dot{v}_{I_r} = \tau a - [\omega \times] v_{I_r} \]  
(15)

\[ \dot{b}_g = n_{wg} \]  
(16)

\[ \dot{b}_a = n_{wa} \]  
(17)

and for \( \omega = [\omega_x, \omega_y, \omega_z]^T \), we have:

\[ \Omega(\omega) = \begin{bmatrix} -[\omega \times] & \omega \\ \omega^T & 1 \end{bmatrix}, \quad [\omega \times] = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \]  
(18)

Especially, the derivation for local velocity (cf. Eq. (15)) is:

\[ \dot{v}_{I_r} = C(\hat{\kappa} \hat{q}) R_{k} v_{I_r} \]  
(19)

\[ \Rightarrow \dot{v}_{I_r} = C(\hat{\kappa} \hat{q}) R_{k} v_{I_r} + C(\hat{\kappa} \hat{q}) R_{k} a_{I_r} - [\omega \times] C(\hat{\kappa} \hat{q}) R_{k} v_{I_r} = \tau a - [\omega \times] v_{I_r} \]  
(20)

The continuous time IMU state propagation with respect to \( \{R_k\} \) is accordingly expressed as:

\[ \tau \hat{\kappa} \hat{q} = \frac{1}{2} \Omega(\hat{\omega}) \hat{\kappa} \hat{q} \]  
(21)

\[ R_{k} \dot{p}_{I_r} = R_{k} C_{\hat{q}}^T \dot{v}_{I_r} \]  
(22)

\[ \dot{v}_{I_r} = \hat{a} - \hat{\kappa} C_{\hat{q}} R_{k} \hat{g} - [\hat{\omega} \times] \dot{v}_{I_r} \]  
(23)

\[ \dot{\hat{b}}_g = 0_{3 \times 1} \]  
(24)

\[ \dot{\hat{b}}_a = 0_{3 \times 1} \]  
(25)

where \( \hat{\omega} = \omega - \hat{b}_g \) and \( \hat{a} = a - \hat{b}_a \) by denoting \( \omega = \omega - b_g - n_g \) and \( a = a - b_a - n_a \), and for brevity we have denoted \( \hat{\kappa} C_{\hat{q}} = C(\hat{\kappa} \hat{q}) \).

In EKF, the dynamics of IMU error-state is needed for which we start from the error quaternion. By the definition of Eq. (5):

\[ \frac{d}{dt} \hat{\kappa} \hat{q} = \delta \hat{q} \otimes \hat{\kappa} \hat{q} \]  
(26)

\[ \frac{d}{dt} \hat{\kappa} \hat{q} = \delta \hat{q} \otimes \hat{\kappa} \hat{q} + \delta \hat{q} \otimes \hat{\kappa} \hat{q} \]  
(27)

Substituting Eq. (13) and Eq. (21) leads to:

\[ \frac{1}{2} \begin{bmatrix} [\omega] \\ 0 \end{bmatrix} \otimes \hat{\kappa} \hat{q} = \delta \hat{q} \otimes \hat{\kappa} \hat{q} + \delta \hat{q} \otimes \left( \frac{1}{2} \begin{bmatrix} [\omega] \\ 0 \end{bmatrix} \otimes \hat{\kappa} \hat{q} \right) \]  
(28)

By rearranging the terms in above equation, we have:

\[ \delta \hat{q} \otimes \hat{\kappa} \hat{q} = \frac{1}{2} \begin{bmatrix} [\omega] \\ 0 \end{bmatrix} \otimes \hat{\kappa} \hat{q} - \delta \hat{q} \otimes \left( \frac{1}{2} \begin{bmatrix} [\omega] \\ 0 \end{bmatrix} \otimes \hat{\kappa} \hat{q} \right) \]  
(29)

\[ \delta \hat{q} = \frac{1}{2} \begin{bmatrix} [\omega] \\ 0 \end{bmatrix} \otimes \delta \hat{q} - \delta \hat{q} \otimes \left( \frac{1}{2} \begin{bmatrix} [\omega] \\ 0 \end{bmatrix} \right) \]  
(30)

\[ \delta \hat{q} = \frac{1}{2} \begin{bmatrix} [\omega] \\ 0 \end{bmatrix} \otimes \delta \hat{q} - \delta \hat{q} \otimes \left( \frac{1}{2} \begin{bmatrix} [\omega] \\ 0 \end{bmatrix} \right) - \frac{1}{2} \begin{bmatrix} \hat{b}_g + n_g \\ 0 \end{bmatrix} \otimes \delta \hat{q} \]  
(31)

\[ \delta \hat{q} = \frac{1}{2} \left( \begin{bmatrix} -[\dot{\omega} \times] \\ -\dot{\omega}^T \end{bmatrix} \otimes \delta \hat{q} - \begin{bmatrix} \dot{\omega} \times \\ -\dot{\omega}^T \end{bmatrix} \delta \hat{q} \right) - \frac{1}{2} \begin{bmatrix} \hat{b}_g + n_g \\ 0 \end{bmatrix} \otimes \delta \hat{q} \]  
(32)

\[ = \frac{1}{2} \begin{bmatrix} -2[\dot{\omega} \times] + (\hat{b}_g + n_g) \times \hat{b}_g + n_g \\ (\hat{b}_g + n_g)^T \hat{b}_g + n_g \end{bmatrix} \delta \hat{q} \]  
(33)
After neglecting the second order error terms in above equation, we get:

\[ \delta \dot{q} \simeq \begin{bmatrix} \frac{1}{2} |\dot{\omega} \times \delta \theta_I | - \frac{1}{2} \left( \dot{\delta} g + n_g \right) + \frac{1}{2} \left( \dot{\delta} g + n_g \right) \times |\delta \theta_I | \end{bmatrix} \left( \frac{1}{2} \left( \dot{\delta} g + n_g \right) \right) \] (34)

After neglecting the second order error terms from above equation, we get:

\[ \delta \dot{q} \simeq \begin{bmatrix} \frac{1}{2} |\dot{\omega} \times \delta \theta_I | - \frac{1}{2} \left( \dot{\delta} g + n_g \right) \end{bmatrix} \] (35)

and the target linear equation is the upper part:

\[ \delta \dot{\theta}_I = -|\dot{\omega} \times \delta \theta_I | - \dot{\delta} g - n_g \] (36)

Next for the error position, we have:

\[ \dot{\hat{p}}_I = R_k \dot{p}_I - R_k \dot{R}_k \] (37)
\[ = \hat{\dot{p}}_I = \kappa C_q^T v_I - \kappa C_q^T \dot{\hat{v}}_I \] (38)
\[ = \kappa C_q^T (1 + [\delta \theta_I x]) (v_I + \dot{\hat{v}}_I) - \kappa C_q^T \dot{v}_I \] (39)
\[ = \kappa C_q^T [\delta \theta_I x] \dot{v}_I + \kappa C_q^T \dot{v}_I \] (40)

After neglecting the second order error terms, the resulting linear equation is:

\[ \dot{\hat{p}}_I = \kappa C_q^T [\delta \theta_I x] \dot{v}_I + \kappa C_q^T \dot{v}_I \] (41)
\[ = -\kappa C_q^T [\delta \theta_I x] \dot{v}_I + \kappa C_q^T \dot{v}_I \] (42)

Similarly for the error velocity, we have:

\[ \dot{\hat{v}}_I = \dot{v}_I - \dot{\hat{v}}_I \] (43)
\[ = a - \hat{\dot{a}} = \hat{\dot{a}} \] (44)
\[ = -\left( \dot{\delta} a + n_a \right) \right) \kappa C_q^T (R_k \dot{g} + R_k \dot{g}) + \kappa C_q^T \dot{R}_k \] (45)
\[ = -\left( \dot{\delta} a + n_a \right) \kappa C_q^T (R_k \dot{g} + R_k \dot{g}) + \kappa C_q^T \dot{R}_k \] (46)

Again, after neglecting the second order error terms from above equation, we get:

\[ \dot{\hat{v}}_I = -\left( \dot{\delta} a + n_a \right) - \kappa C_q^T (R_k \dot{g} + R_k \dot{g}) - |\dot{\omega} \times \dot{\hat{v}}_I| + \left( \dot{\delta} g + n_g \right) \times |\dot{\hat{v}}_I| \] (47)
\[ = -\kappa C_q^T (R_k \dot{g} - [\dot{g} \times \delta \theta_I] \dot{v}_I - |\dot{\omega} \times \dot{v}_I| - |\dot{\omega} \times \dot{v}_I| \times (\dot{\hat{b}}_g + n_g) - (\dot{\hat{b}}_g + n_g) \] (48)

where for brevity we have denoted \( \dot{R}_k \dot{g} \). The remaining IMU error-states can be obtained as:

\[ \dot{\hat{b}}_g = \dot{\hat{b}}_g - \dot{\hat{b}}_g = n_{wg} \] (49)
\[ \dot{\hat{b}}_a = \dot{\hat{b}}_a - \dot{\hat{b}}_a = n_{wa} \] (50)

For the error propagation of \( R_k x_G \), we consider it as a zero dynamic process, then:

\[ \delta \dot{\theta}_G = 0_{3 \times 1} \] (51)
\[ \dot{\hat{P}}_G = R_k \dot{P}_G - R_k \dot{P}_G = 0_{3 \times 1} \] (52)
\[ \dot{\hat{g}} = R_k \dot{g} - R_k \dot{g} = 0_{3 \times 1} \] (53)
After combining above results, we obtain the continuous time, linear model of the robocentric error-state as:

\[
\begin{bmatrix}
\delta \theta_G \\
\dot{p}_G \\
\dot{g} \\
\delta \theta_I \\
\dot{p}_I \\
\dot{v}_I \\
\dot{b}_g \\
\dot{b}_a
\end{bmatrix}
= \begin{bmatrix}
0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
0_3 & 0_3 & 0_3 & -[\hat{\omega} \times] & 0_3 & 0_3 & -I_3 & 0_3 \\
0_3 & 0_3 & 0_3 & -\frac{\tau}{k} C_q ^T [v_L \times] & 0_3 & \frac{\tau}{k} C_q ^T & 0_3 & 0_3 \\
0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
0_4 & 0_4 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & b_g \\
0_4 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & b_a
\end{bmatrix}
\begin{bmatrix}
\delta \theta_G \\
\dot{p}_G \\
\dot{g} \\
\delta \theta_I \\
\dot{p}_I \\
\dot{v}_I \\
\dot{b}_g \\
\dot{b}_a
\end{bmatrix}
\]

\[
\Rightarrow \ddot{x} = F\dot{x} + Gn
\]  

2.2.2 Discrete Time State Propagation

For an actual EKF implementation, a discrete time propagation model is needed. From time-step \( k \) to \( \tau \), \( \hat{\theta} _k ^{\hat{\theta} _k ^*} \) can be obtained using the zeroth order quaternion integrator [18]:

\[
\hat{\theta} _k ^{\hat{\theta} _k ^*} = \int _{t_k} ^{t_\tau} \hat{\theta} _k ^* ds
\]

\[
= \int _{t_k} ^{t_\tau} \frac{1}{2} \Omega (\hat{\omega}) ^{\hat{\theta} _k ^*} ds
\]

\[
= \int _{t_k} ^{t_\tau} \frac{1}{2} \Omega (\omega _{m,s} - \hat{b}_g) ^{\hat{\theta} _k ^*} ds
\]

and \( R_k \hat{p}_L \) and \( R_k \hat{v}_L \) can be computed respectively using IMU preintegration:

\[
R_k \hat{p}_L = \hat{p}_L \Delta t + \int _{t_k} ^{t_\tau} \int _{t_k} ^{s} \frac{\mu}{k} C_q ^T \dot{a} d\mu ds
\]

\[
= \hat{p}_L \Delta t + \int _{t_k} ^{t_\tau} \int _{t_k} ^{s} \frac{\mu}{k} C_q ^T (a_{m,\mu} - \dot{b}_a - \mu \dot{g}) d\mu ds
\]

\[
= \hat{p}_L \Delta t - \frac{1}{2} R_k \hat{g} \Delta t^2 + \int _{t_k} ^{t_\tau} \int _{t_k} ^{s} \frac{\mu}{k} C_q ^T (a_{m,\mu} - \dot{b}_a) d\mu ds
\]

\[
\Delta p_{k,\tau}
\]

\[
R_k \hat{v}_L = \hat{v}_L + \int _{t_k} ^{t_\tau} \int _{t_k} ^{s} \dot{q} C_q ^T \dot{a} ds
\]

\[
= \hat{v}_L + \int _{t_k} ^{t_\tau} \int _{t_k} ^{s} C_q ^T (a_{m,s} - \dot{b}_a - \dot{g}) ds
\]

\[
= \hat{v}_L - R_k \hat{g} \Delta t + \int _{t_k} ^{t_\tau} \int _{t_k} ^{s} C_q ^T (a_{m,s} - \dot{b}_a) ds
\]

\[
\Delta v_{k,\tau}
\]
where \( \hat{b}_g = \hat{b}_{g_k} \), \( \hat{b}_a = \hat{b}_{a_k} \), and \( \Delta t = t_{\tau} - t_k \). We can find that the preintegrated terms, \( \Delta p_{k,\tau} \) and \( \Delta \mathbf{v}_{k,\tau} \), can be computed recursively with the IMU measurements \cite{19}. Therefore, the estimate for the local velocity, \( \hat{\mathbf{v}}_{I_{\tau}} \), can be obtained using the preceding results as:

\[
\hat{\mathbf{v}}_{I_{\tau}} = \hat{\mathbf{v}}_{I_{\tau+1}} + \mathbf{R}_k \hat{\mathbf{v}}_{I_{\tau}}.
\]

The bias estimates are assumed to be constant within the time interval \([t_k, t_{\tau}]\), which are

\[
\hat{b}_{g_{\tau+1}} = \hat{b}_{g_k}, \quad \hat{b}_{a_{\tau+1}} = \hat{b}_{a_k}.
\]

Thus, the discrete time state propagation from time-step \( \tau \) to \( \tau + 1 \) can be described as:

\[
\mathbf{R}_k \hat{\mathbf{x}}_{G_{\tau+1}} = \mathbf{R}_k \hat{\mathbf{x}}_{G_{\tau}}
\]

and for the IMU state

\[
\mathbf{R}_k \hat{\mathbf{p}}_{I_{\tau+1}} = \mathbf{R}_k \hat{\mathbf{p}}_{I_{\tau}} + \Delta \mathbf{p}_{\tau,\tau+1},
\]

\[
\hat{\mathbf{v}}_{I_{\tau+1}} = \hat{\mathbf{v}}_{I_{\tau+1}} + \mathbf{R}_k \hat{\mathbf{v}}_{I_{\tau}} + \Delta \mathbf{v}_{\tau,\tau+1},
\]

\[
\hat{b}_{g_{\tau+1}} = \hat{b}_{g_{\tau}}, \quad \hat{b}_{a_{\tau+1}} = \hat{b}_{a_{\tau}}.
\]

In addition, we consider the estimate of the clone states, \( \hat{\mathbf{w}} \), is static within \([t_k, t_{k+1}]\).

### 2.2.3 Discrete Time Covariance Propagation

In EKF, the covariance propagation from time-step \( \tau \) to \( \tau + 1 \) with respect to \( \{\mathbf{R}_k\} \) can be expressed as:

\[
\mathbf{P}_{\tau+1|k} = \Phi_{\tau+1,\tau} \mathbf{P}_{\tau|k} \Phi_{\tau+1,\tau}^\top + \mathbf{Q}_{\tau}
\]

where \( \mathbf{P}_{\tau|k} \) and \( \mathbf{P}_{\tau+1|k} \) are the estimated and predicted covariance matrices, respectively, \( \Phi_{\tau+1,\tau} \) is the state transition matrix from \( \tau \) to \( \tau + 1 \), and \( \mathbf{Q}_{\tau} \) is the noise covariance matrix. Assuming that \( \dot{\mathbf{w}} \) and \( \dot{\mathbf{a}} \) are constant for time interval \([t_\tau, t_{\tau+1}]\) with \( \delta t = t_{\tau+1} - t_\tau \), \( \mathbf{F} \) is constant and the state transition matrix can be found as:

\[
\Phi(\tau + 1, \tau) = \exp(\mathbf{F} \delta t) = \mathbf{I}_{24} + \mathbf{F} \delta t + \frac{1}{2!} \mathbf{F}^2 \delta t^2 + \cdots
\]

Considering that IMU has a very high frequency (e.g., \( \delta t \) is \( 10^{-3} \sim 10^{-2} \)s for 100 ~ 1000Hz), we can obtain the discrete time error-state transition matrix by just using the first order approximation of Eq. (75):

\[
\Phi_{\tau+1,\tau} = \mathbf{I}_{24} + \mathbf{F} \delta t
\]

\[
\mathbf{F} = \begin{bmatrix}
0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
0_3 & 0_3 & 0_3 & -[\dot{\omega} \times] & 0_3 & 0_3 & -\mathbf{I}_3 & 0_3 \\
0_3 & 0_3 & -\mathbf{C}_q^\top \dot{\mathbf{g}} \times & -[\ddot{\mathbf{g}} \times] & 0_3 & 0_3 & -[\dot{\omega} \times] & -\dot{\mathbf{v}}_{I_{\tau}} \times & -\mathbf{I}_3 \\
0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
\end{bmatrix}
\]

In order to obtain the noise covarian matrix \( \mathbf{Q}_{\tau} \), we first show the discrete-time solution of Eq. (55):

\[
\ddot{x}_{\tau+1} = \Phi(\tau + 1, \tau) \dot{x}_{\tau} + \int_{t_\tau}^{t_{\tau+1}} \Phi(\tau + 1, s) \mathbf{G}(s) \mathbf{n} \, ds
\]
\[ G(s) = \begin{bmatrix}
0_3 & 0_3 & 0_3 & 0_3 \\
0_3 & 0_3 & 0_3 & 0_3 \\
0_3 & 0_3 & 0_3 & 0_3 \\
-\bar{v}_t & 0_3 & 0_3 & 0_3 \\
-\frac{1}{2} & 0_3 & 0_3 & 0_3 \\
0_3 & 0_3 & 0_3 & I_3 \\
0_3 & 0_3 & 0_3 & I_3 \\
\end{bmatrix}, \quad n = \begin{bmatrix}
n_p \\
n_{wg} \\
n_q \\
n_{wa}
\end{bmatrix} \quad (79) \]

If all matrices are constant over time interval \([t, t+1]\), the noise covariance for Eq. (78) can be computed as:

\[ Q_z = \mathbb{E} [\Phi (\tau + 1, \tau) G(\tau) n (\Phi (\tau + 1, \tau) G(\tau) n)^	op] \delta t \]
\[ = \Phi_{\tau+1,\tau} G \mathbb{E} [nn^\top] G^\top \Phi_{\tau+1,\tau} \delta t \]
\[ = \Phi_{\tau+1,\tau} G \Sigma G^\top \Phi_{\tau+1,\tau} \delta t \quad (80) \]
\[ \Sigma = \begin{bmatrix}
\sigma^2_w & 0_3 & 0_3 & 0_3 \\
0_3 & \sigma^2_{wg} & I_3 & 0_3 \\
0_3 & 0_3 & \sigma^2_{wg} & I_3 \\
0_3 & 0_3 & 0_3 & \sigma^2_{wg}
\end{bmatrix} \quad (81) \]

where \( \Sigma \) is the continuous time noise covariance matrix. Substituting Eq. (76) into Eq. (80) and neglecting the second order terms of \( \delta t \), we have the discrete time noise covariance matrix:

\[ Q_z = G \Sigma G^\top \delta t \quad (82) \]

We introduce the following partition for the augmented covariance matrix at time-step \( k \):

\[ P_k = \begin{bmatrix}
P_{xx_k} & P_{xw_k} \\
P_{xw_k}^\top & P_{ww_k}
\end{bmatrix} \quad (83) \]

where \( P_{xx_k} \) is the covariance matrix of the evolving robocentric state, \( P_{ww_k} \) is the covariance matrix of the states in the window, and \( P_{xw_k} \) is the correlation between the errors in the robocentric state and the states in the window. With this notation, at time-step \( \tau + 1 \in [t_k, t_{k+1}] \) the propagated covariance is given by:

\[ P_{\tau+1|k} = \begin{bmatrix}
P_{xx_{\tau+1|k}} & \Phi_{\tau+1,k} P_{xw_k} \\
P_{xw_k}^\top & P_{ww_k}
\end{bmatrix} \Phi_{\tau+1,k} \quad (84) \]

where \( P_{xx_{\tau+1|k}} \) can be recursively computed using Eq. (74), and the evolving error-state transition matrix:

\[ \Phi_{\tau+1,k} = \Phi_{\tau+1,\tau} \cdots \Phi_{k+\delta t,k} = \prod_{\ell=k}^{\tau} \Phi_{\ell+\delta t,\ell} \quad (85) \]

with \( \Phi_{k,k} = I_24 \) at time-step \( k \).

### 2.3 Update

#### 2.3.1 Bearing-only Model

We present this measurement model by considering the case of a single landmark \( L_j \) which has been observed from a set of \( n_j \) robocentric frames of reference \( R_j \). For \( i \in R_j \), the observation of landmark \( L_j \) is described by the following bearing-only model in the \( xyz \) coordinates:

\[ z_{j,i} = \frac{1}{z_j} \begin{bmatrix}
x_{j,i} \\
y_{j,i}
\end{bmatrix} + n_{j,i}, \quad c_i p_{L_j} = \begin{bmatrix}
x^j_i \\
y^j_i \\
z^j_i
\end{bmatrix} = c^i_j R^R_{L_j} p_{L_j} + c^i_p p_I \quad (86) \]

where \( n_{j,i} \sim \mathcal{N}(0, \sigma^2_{ni} I_3) \) represents the image noise, and \( c^i_j p_{L_j} \) is the position of landmark expressed in the camera frame \( \{C_i\} \). Assume that \( R_j = \{1, 2, \ldots, n\} \) and \( \{R_1\} \) is the first robocentric frame from which the landmark is observed, we begin to derive the expression for \( R^R_{L_j}, i \in \{R_j\} \) following the steps:
1. If \( i = 2 \), then
\[
R_2 p_{L_j} = \frac{2}{1} C_q R_1 p_{L_j} - R_1 p_{R_2}
\]
(87)
\[
= \frac{2}{1} C_q R_1 p_{L_j} - \frac{2}{1} C_q R_1 p_{R_2}
\]
(88)

2. If \( i = 3 \), then
\[
R_3 p_{L_j} = \frac{3}{2} C_q R_1 p_{L_j} - R_1 p_{R_3}
\]
(89)
\[
= \frac{3}{2} C_q R_1 p_{L_j} - \frac{3}{2} C_q R_2 p_{R_3} - \frac{3}{2} C_q R_1 p_{R_2}
\]
(90)

3. If \( i = 4 \), then
\[
R_4 p_{L_j} = \frac{4}{3} C_q R_1 p_{L_j} - R_1 p_{R_4}
\]
(91)
\[
= \frac{4}{3} C_q R_1 p_{L_j} - \frac{4}{3} C_q R_2 p_{R_4} - \frac{4}{3} C_q R_3 p_{R_4} - \frac{4}{3} C_q R_1 p_{R_2}
\]
(92)

4. 

5. From above results, we can conclude that
\[
R_i p_{L_j} = \frac{i}{i-1} C_q R_{i-1} p_{L_j} - R_{i-1} p_{R_i}
\]
(93)
\[
= \frac{i}{i-1} C_q R_{i-1} p_{L_j} - \frac{i}{i-1} C_q R_{i-2} p_{R_{i-1}} - \cdots - \frac{i}{i-1} C_q R_1 p_{R_{i-1}} - \cdots - \frac{i}{i-1} C_q R_1 p_{R_2}
\]
(94)

Note that in Eq. (94) all relative poses between \( \{R_i\} \) and \( \{R_i\}, i \in \{R_j \setminus 1\} \) have been used to express the \( i \)-th observation of \( L_j \). Then, the corresponding Jacobians with respect to the error-state are derived as:

1. If \( i = 2 \), then
\[
\frac{\partial R_2 p_{L_j}}{\partial \theta_2} = \frac{2}{1} C_q R_1 p_{L_j} - \frac{2}{1} C_q R_1 p_{R_2}
\]
(95)
\[
\frac{\partial R_2 p_{L_j}}{\partial R_2 p_{L_j}} = \frac{2}{1} C_q
\]
(96)

2. If \( i = 3 \), then
\[
\frac{\partial R_3 p_{L_j}}{\partial \theta_3} = \frac{3}{2} C_q R_1 p_{L_j} - \frac{3}{2} C_q R_2 p_{R_3}
\]
(97)
\[
\frac{\partial R_3 p_{L_j}}{\partial R_2 p_{L_j}} = \frac{3}{2} C_q
\]
(98)

3. If \( i = 4 \), then
\[
\frac{\partial R_4 p_{L_j}}{\partial \theta_4} = \frac{4}{3} C_q R_1 p_{L_j} - \frac{4}{3} C_q R_3 p_{R_4}
\]
(99)
\[
\frac{\partial R_4 p_{L_j}}{\partial R_3 p_{L_j}} = \frac{4}{3} C_q
\]
(100)
From above, we can find that the landmark observation from one robocentric frame has been correlated to a chain of relative poses through its tracking history, which is contrast to the world-centric case (e.g., [1]) where the landmark observation was only correlated to the pose from which it was obtained.

Linearizing about the estimates of the state and landmark position, the residual of Eq. (86) can be approximated as:

$$\mathbf{r}_{j,i} = \mathbf{z}_{j,i} - \hat{\mathbf{z}}_{j,i} \approx \mathbf{H}_{x_{j,i}} \hat{\mathbf{x}} + \mathbf{H}_{L_{j,i}} \hat{\mathbf{p}}_{L_j} + \mathbf{n}_{j,i}$$

(107)

where \(\mathbf{H}_{x_{j,i}} (\mathbf{H}_{x_{j,i}} = 0)\) and \(\mathbf{H}_{L_{j,i}} (\mathbf{H}_{L_{j,i}} = \mathbf{H}_{p_{j,i}} \mathbf{C}_q \mathbf{C}_q')\) are the Jacobians with respect to the errors of the state and landmark position, respectively. While \(R^1 \hat{\mathbf{p}}_{L_j}\) is the position of landmark \(L_j\) in \(\{R_1\}\) whose value is unknown in computation, for which we first solve a nonlinear least-squares minimization using the measurements \(\mathbf{z}_{j,i}, i \in \mathcal{R}_j\) and the relative pose estimates \(\mathbf{w}\) (cf. Appendix B). We stack all \(\mathbf{r}_{j,i}, i \in \mathcal{R}_j\) to obtain the following equation for landmark \(L_j\):

$$\mathbf{r}_j \approx \mathbf{H}_x \hat{\mathbf{x}} + \mathbf{H}_{L_j} \hat{\mathbf{p}}_{L_j} + \mathbf{n}_j$$

(109)

As \(R^1 \hat{\mathbf{p}}_{L_j}\) is computed using the state estimate \(\hat{\mathbf{x}}\) (precisely, \(\hat{\mathbf{w}}\)), \(R^1 \hat{\mathbf{p}}_{L_j}\) is correlated with \(\hat{\mathbf{x}}\). However, \(R^1 \hat{\mathbf{p}}_{L_j}\) is not in the state vector, in order to find a valid residual for EKF update, we project \(\mathbf{r}_j\) to the left nullspace of \(\mathbf{H}_{L_j}\). Let \(\mathbf{O}_{L_j}\) denotes the matrix consisting of the basis of the left nullspace of \(\mathbf{H}_{L_j}\), we have:

$$\tilde{\mathbf{r}}_j = \mathbf{O}_{L_j}^\top \mathbf{r}_j \approx \mathbf{O}_{L_j}^\top \mathbf{H}_x \hat{\mathbf{x}} + \mathbf{O}_{L_j}^\top \mathbf{n}_j = \tilde{\mathbf{H}}_x \hat{\mathbf{x}} + \tilde{\mathbf{n}}_j$$

(110)

After rejecting the observations of the landmarks at infinity \((x_j^i \gg x_j^i, z_j^i \gg y_j^i, i \in \mathcal{R}_j)\) and the observations near the principal point of image \((x_j^i \rightarrow 0, y_j^i \rightarrow 0, i \in \mathcal{R}_j)\), we have the full column rank 2n_j × 3 matrix \(\mathbf{H}_{L_j}\), of which the nullspace is of dimension 2n_j − 3. In Eq. (110) without explicitly finding \(\mathbf{O}_{L_j}\), the projection of \(\mathbf{r}_j\) and \(\mathbf{H}_x\) on the nullspace of \(\mathbf{H}_{L_j}\) is efficiently computed using Givens rotation [20] with complexity \(O(n_j^2)\). Also, \(\tilde{\mathbf{n}}_j = \mathbf{O}_{L_j}^\top \mathbf{n}_j\) is the noise vector with the covariance \(\mathbf{R} = \mathbf{O}_{L_j}^\top \mathbf{E} [\mathbf{n}_j \mathbf{n}_j^\top] \mathbf{O}_{L_j} = \sigma_{im}^2 \mathbf{I}_{2n_j-3}\).

### 2.3.2 Inverse Depth Bearing-only Model

We still consider the bearing-only model and the case of a single landmark \(L_j\) which are used in the preceding section. However, instead of using the \(xyz\) coordinates, we introduce the following tailored inverse depth [21] parametrization for the landmark position:

$$\mathbf{C}_q^1 \mathbf{p}_{L_j} = \mathbf{C}_q \mathbf{C}_q^1 \mathbf{p}_{L_j} + \mathbf{i} \mathbf{p}_1 := f_r(\phi, \psi, \rho)$$

$$\mathbf{C}_q^1 \mathbf{p}_{L_j} = \frac{1}{\rho} \mathbf{e}(\phi, \psi), \quad \mathbf{e} = \begin{bmatrix} \cos \phi \sin \psi \\ \sin \phi \\ \cos \phi \cos \psi \end{bmatrix}$$

(111)

(112)

RPNG-2018-RVIO 10
where we use \( \mathbf{e} \) (i.e., the unit vector of \( \mathbf{C}_q \mathbf{p}_{L_j} \)) and \( 1/\rho \) (i.e., the depth of of \( L_j \) in \{C\}) to express \( \mathbf{C}_q \mathbf{p}_{L_j} \) with \( \phi \) and \( \psi \) the elevation and azimuth expressed in \{C\}. Especially, the relative poses between \{C\} and \{\mathcal{R}_j \}, \( i \in \mathcal{R}_j \{1\} \) are expressed using the calibration result, \{\( \mathcal{C}_q \), \( \mathbf{C}_q \mathbf{p}_I \)}, and the relative poses between the robocentric frames of reference, \( \mathbf{w} \), as

\[
\mathbf{C}_q \mathbf{p}_{I} = \mathbf{C}_q \mathbf{p}_{L_j} \left( \mathbf{R}_j \mathbf{C}_q \mathbf{p}_I + \mathbf{C}_q \mathbf{R}_j \mathbf{p}_{R_l} + \mathbf{C}_q \mathbf{p}_I \right)
\]  

(114)

Considering the observation of landmark at infinity (i.e., \( \rho \to 0 \)), we can normalize Eq. (111) with \( \rho \) to avoid numerical issues. Note that such operation does not affect the bearing expression because the scale is always eliminated in the fraction form:

\[
\mathbf{h}_i(\mathbf{w}, \phi, \psi, \rho) = \left[ \frac{h_{i,1}(\mathbf{w}, \phi, \psi, \rho)}{h_{i,2}(\mathbf{w}, \phi, \psi, \rho)} \right] \mathbf{h}_i = i \mathbf{q}_i \mathbf{e}(\phi, \psi) + \rho \mathbf{p}_I
\]  

(115)

This equation retains the geometry of Eq. (111), meanwhile compasses two degenerate cases: (i) observing the landmarks at infinity (i.e., \( \rho \to 0 \)), and (ii) having low parallax between \{C\} and \{C\} (i.e., \( \mathbf{p}_I \to 0 \)). For both cases, Eq. (115) can be approximated by \( \mathbf{h}_i \simeq i \mathbf{C}_q \mathbf{e}(\phi, \psi) \), and hence the corresponding measurements can still provide the information about the camera rotation. Thus, we introduce the following inverse depth measurement model for robocentric VIO:

\[
\mathbf{z}_{j,i} = \frac{1}{h_{i,3}(\mathbf{w}, \phi, \psi, \rho)} \left[ h_{i,1}(\mathbf{w}, \phi, \psi, \rho) \right] + \mathbf{n}_{j,i}
\]  

(116)

Assume that \( \mathcal{R}_j = \{1,2,\ldots,n\} \), and \( \{\mathcal{R}_j \} \) is the robocentric frame from which \( L_j \) is firstly observed, then:

\[
\mathbf{C}_q \mathbf{p}_{L_j} = i \mathbf{C}_q \mathbf{C}_q \mathbf{p}_{L_j} + i \mathbf{p}_I
\]  

(117)

1. If \( i = 2 \), then

\[
\mathbf{h}_2 = i \mathbf{C}_q \mathbf{p}_I \left( \mathbf{L}_q \mathbf{e}(\phi, \psi) + \rho \mathbf{p}_C - \rho_R \mathbf{p}_{R_2} \right) + \rho \mathbf{p}_C
\]  

(119)

2. If \( i = 3 \), then

\[
\mathbf{h}_3 = i \mathbf{C}_q \mathbf{p}_I \left( \mathbf{L}_q \mathbf{e}(\phi, \psi) + \rho \mathbf{p}_C - \rho_R \mathbf{p}_{R_2} \right) + \rho \mathbf{p}_C
\]  

(121)

3. If \( i = 4 \), then

\[
\mathbf{h}_4 = i \mathbf{C}_q \mathbf{p}_I \left( \mathbf{L}_q \mathbf{e}(\phi, \psi) + \rho \mathbf{p}_C - \rho_R \mathbf{p}_{R_4} \right) + \rho \mathbf{p}_C
\]  

(123)

4. \ldots...

5. Finally, we can conclude that

\[
\mathbf{h}_5 = i \mathbf{C}_q \mathbf{p}_I \left( \mathbf{L}_q \mathbf{e}(\phi, \psi) + \rho \mathbf{p}_C - \rho_R \mathbf{p}_{R_5} \right) + \rho \mathbf{p}_C
\]  

(125)
Similarly, we can obtain the Jacobians of $h_i, i \in \{R, \lambda\}$ with respect to the error-state and $[\phi, \psi, \rho]^T$:

1. If $i = 2$, then
   \[
   \frac{\partial h_2}{\partial \theta_2} = \hat{C} C_q \frac{\partial}{\partial \theta_2} [C_q \dot{e} + \dot{p}_C - \dot{p}_R^T \dot{p}_R] \times, \quad \frac{\partial h_2}{\partial p_R} = -\dot{p}_T^2 C_q \hat{C}_q \tag{127}
   \]

2. If $i = 3$, then
   \[
   \frac{\partial h_3}{\partial \delta_3} = \hat{C} C_q \frac{\partial}{\partial \delta_3} [C_q \dot{e} + \dot{p}_C - \dot{p}_R^T \dot{p}_R] \times, \quad \frac{\partial h_3}{\partial p_R} = -\dot{p}_T^2 C_q \hat{C}_q \tag{128}
   \]

3. If $i = 4$, then
   \[
   \frac{\partial h_4}{\partial \delta_4} = \hat{C} C_q \frac{\partial}{\partial \delta_4} [C_q \dot{e} + \dot{p}_C - \dot{p}_R^T \dot{p}_R] \times, \quad \frac{\partial h_4}{\partial p_R} = -\dot{p}_T^2 C_q \hat{C}_q \tag{130}
   \]

4. ……

5. In conclusion, we have the general form as follows. Specifically, the Jacobians with respect to $[\phi, \psi, \rho]^T$ are obtained from Eq. (115):
   \[
   \frac{\partial h_\ell}{\partial \theta_\ell} = \hat{C} C_q \frac{\partial}{\partial \theta_\ell} \left[ (C_q \dot{e} + \dot{p}_C - \dot{p}_R^T \dot{p}_R) \times \right] \tag{133}
   \]
   \[
   = \frac{\partial h_\ell}{\partial p_R} = -\dot{p}_T^2 C_q \hat{C}_q \tag{134}
   \]
   \[
   \frac{\partial h_i}{\partial (\phi, \psi, \rho)} = \left[ \begin{array}{c} \frac{\partial h_i}{\partial \phi} \\ \frac{\partial h_i}{\partial \psi} \\ \frac{\partial h_i}{\partial \rho} \end{array} \right] = \left[ \begin{array}{c} \hat{C}_q \left[ \begin{array}{c} -\sin \phi \sin \psi \\ \cos \phi \cos \psi \\ \cos \phi \sin \psi \\ -\sin \phi \cos \psi \\ -\cos \phi \sin \psi \end{array} \right] \end{array} \right] \right] \hat{p}_i \tag{136}
   \]

Let us denote $\lambda = [\phi, \psi, \rho]^T$, then the residual of Eq. (116) by linearizing about the estimates of the robot state and $\lambda$ can be approximated as:

\[
r_{j,i} = z_{j,i} - \hat{z}_{j,i} \simeq H_{x_{j,i}} \hat{x} + H_{\lambda_{j,i}} \hat{\lambda} + n_{j,i} \tag{137}
\]

\[
H_{x_{j,i}} = H_{p_{j,i}}^T C_q \frac{\partial}{\partial p_R} \left[ \begin{array}{c} \rho_{3\times24} \\ \rho_{3\times6} \\ \ldots \\ \rho_{3\times6} \end{array} \right], \quad H_{\lambda_{j,i}} = H_{p_{j,i}}^T H_{inv_{j,i}},
\]

\[
H_{p_{j,i}} = \frac{1}{h_{i,3}} \begin{bmatrix} 1 & 0 & -h_{i,1} \\ 0 & 1 & -h_{i,2} \end{bmatrix},
\]

\[
H_{inv_{j,i}} = \left[ i \hat{C}_q \left[ \begin{array}{c} -\sin \phi \sin \hat{\psi} \\ \cos \phi \cos \hat{\psi} \\ \cos \phi \sin \hat{\psi} \\ -\sin \phi \cos \hat{\psi} \\ -\cos \phi \sin \hat{\psi} \end{array} \right] \right] \hat{p}_1,
\]

\[
H_{w_{j,i}} = \left[ (\frac{1}{i} C_q \dot{e} + \dot{p}_C - \dot{p}_R^T \dot{p}_R) \times \right] C_q^T \mbox{ } -\dot{p}_1^T \left[ \begin{array}{c} i \hat{C}_q \left[ \begin{array}{c} -\sin \phi \sin \hat{\psi} \\ \cos \phi \cos \hat{\psi} \\ \cos \phi \sin \hat{\psi} \\ -\sin \phi \cos \hat{\psi} \\ -\cos \phi \sin \hat{\psi} \end{array} \right] \end{array} \right] \right] \right] \hat{p}_i \tag{138}
\]
Here let us take a break to discuss the rank deficiency in $H_{\chi_j}$, which determines the way to get a nominal residual as using the Givens rotations (cf. Eq. (110)). We consider the rank deficiency happened in either $H_{p_j}$, or $H_{\text{nav}_{j,i}}$. For the former, we find two possible conditions for $h_i$: (i) $\check{h}_{i,1} = \check{h}_{i,2} = \check{h}_{i,3}$, which means that the image size should be at least $2f \times 2f$ (f is the focal length), or (ii) $\check{h}_{i,1} \rightarrow 0$ and $\check{h}_{i,2} \rightarrow 0$, which means that the measurement of $L_j$ is near the principal point of image; however, for the latter we can also find two conditions: (iii) $\cos \hat{\phi} \rightarrow 0$, which means that we have either infinitely small focal length or infinitely large image size for the camera so that $[\hat{\phi}] \rightarrow \pi/2$ can happen, or (iv) $\hat{r}_1 \rightarrow 0$, which means a small parallax between $\{C_1\}$ and $\{C_i\}$. Among those conditions, (i) is about the selection for the lens (e.g., a wide-angle lens) which is restricted by the image size of the camera, and (ii) is too ideal to be realized in the real world. While (ii) and (iv) are common in the vision-aided navigation and can be conveniently detected from the pixel values of measurements and the relative poses, respectively. We can discard the measurements which meet (ii) when computing the Jacobians. While (iv) is the case that we can utilize, however, makes the last column of $H_{\chi_j}$ approaches zero, for which we perform Givens rotation on the first two columns to guarantee the image size should be at least $2H \times 2H$. We consider the rank deficiency happened in either $H_{\text{nav}_{j,i}}$, or $H_{\text{nav}_{j,i}}$. For the former, we find two possible conditions for $\check{h}_i$: (i) $\check{h}_{i,1} = \check{h}_{i,2} = \check{h}_{i,3}$, which means that the image size should be at least $2f \times 2f$ (f is the focal length), or (ii) $\check{h}_{i,1} \rightarrow 0$ and $\check{h}_{i,2} \rightarrow 0$, which means that the measurement of $L_j$ is near the principal point of image; however, for the latter we can also find two conditions: (iii) $\cos \hat{\phi} \rightarrow 0$, which means that we have either infinitely small focal length or infinitely large image size for the camera so that $[\hat{\phi}] \rightarrow \pi/2$ can happen, or (iv) $\hat{r}_1 \rightarrow 0$, which means a small parallax between $\{C_1\}$ and $\{C_i\}$. Among those conditions, (i) is about the selection for the lens (e.g., a wide-angle lens) which is restricted by the image size of the camera, and (ii) is too ideal to be realized in the real world. While (ii) and (iv) are common in the vision-aided navigation and can be conveniently detected from the pixel values of measurements and the relative poses, respectively. We can discard the measurements which meet (ii) when computing the Jacobians. While (iv) is the case that we can utilize, however, makes the last column of $H_{\chi_j}$ approaches zero, for which we perform Givens rotation on the first two columns to guarantee a valid nullspace projection numerically. Also, in Eq. (137) an estimate of $\lambda$ is needed, for which a nonlinear least-squares minimization is solved at first (cf. Appendix C). To the end, we stack all $\check{r}_{j,i}$, $i \in \mathcal{R}_j$ to obtain the residual $\check{r}_j$ for EKF update (cf. Eq. (110)).

2.3.3 EKF Update

Assuming at time-step $k+1$, we have the observations of $M$ landmarks to process, we can stack $\check{r}_j$, $j = 1 \ldots M$ to have:

$$\check{r} = H_{\chi} \dot{x} + \check{n}$$  \hfill (139)

of which the resulting row dimension is $d = \sum_{j=1}^{M} (2n_j - 3)$. However, in practice, $d$ can be a large number even if $M$ is small (e.g., $d = 170$, when 10 landmarks are observed from 10 robot poses). In order to reduce the computational complexity in EKF update, we perform QR decomposition on $H_{\chi}$ [1] before stepping into the standard procedure. Inheriting Eq. (108), $H_{\chi}$ is rank deficient due to the zero columns corresponding to the robecentric state, however, the rest of columns corresponding to the clone states are linearly independent (i.e., assuming independent observations). Therefore, we can perform QR decomposition as:

$$H_{\chi} = \begin{bmatrix} 0_{d \times 24} & H_{w} \end{bmatrix}$$

$$= \begin{bmatrix} 0_{d \times 24} & \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} T_w & 0_{(d-6(N-1)) \times 6(N-1)} \end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} 0_{d(N-1) \times 24} & T_w & 0_{(d-6(N-1)) \times 6(N-1)} \end{bmatrix}$$  \hfill (141)

where $Q_1$ and $Q_2$ are unitary matrices of dimension $d \times 6(N-1)$ and $d \times (d-6(N-1))$, respectively, and $T_w$ is a $6(N-1) \times 6(N-1)$ upper triangular matrix. With this definition, Eq. (139) yields:

$$\check{r} = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} 0 & T_w \end{bmatrix} \dot{x} + \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \check{n}$$

$$\Rightarrow \begin{bmatrix} Q_1^T & Q_2^T \end{bmatrix} \check{r} = \begin{bmatrix} 0 & T_w \end{bmatrix} \dot{x} + \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \check{n}$$  \hfill (144)

For which after discarding the lower rows which are only about the measurement noise, we employ the upper $6(N-1)$ rows, instead of Eq. (139), as the residual for the following EKF update:

$$\check{r} = Q_1^T \check{r} = \begin{bmatrix} 0 & T_w \end{bmatrix} \dot{x} + Q_1^T \check{n} = \check{H}_x \dot{x} + \check{n}$$

where $\check{n} = Q_1^T \check{n}$ is the noise vector with the covariance $\check{R} = Q_1^T \mathbb{E}[\check{n}\check{n}^T] Q_1 = \sigma_m^2 I_{6(N-1)}$. In such a way, the dimension of measurement model is reduced to $6(N-1)$. Especially, when $d \gg 6(N-1)$ these can be done by Givens rotations with complexity $O(N^2d)$. Next, the EKF update is conducted as:

$$K = P_{\chi}^{T} ( \check{H}_x P_{\chi}^{T} + \check{R} )^{-1}$$

$$\hat{x}_{k+1} = \hat{x}_{k+1|k} + K\check{r}$$

$$P_{k+1|k} = ( I_{24+6(N-1)} - K\check{H}_x ) P_{k+1|k} ( I_{24+6(N-1)} - K\check{H}_x )^T + K\check{R} K^{T}$$

$$\text{RPNG-2018-RVIO}$$

13
2.3.4 State Augmentation

In contrast to [1] which clones the pose estimates at the time of receiving a new image, we perform a delayed clone after the EKF update. Assuming that at time-step \( k + 1 \) the EKF update is done, we append the copy of the relative pose estimate, \( \left\{ q_{k+1}^0, p_{k+1}^0, R_k^r \right\} \), to the end of the state vector \( \mathbf{x}_{k+1|k+1} \). Moreover, for covariance matrix we introduce the following Jacobian to facilitate matrix augmentation:

\[
P_{k+1|k+1} = \begin{bmatrix} \text{I}_{24+6(N-1)} \end{bmatrix}_J P_{k+1|k+1} \begin{bmatrix} \text{I}_{24+6(N-1)} \end{bmatrix}^T
\]

(149)

\[
\mathbf{J} = \begin{bmatrix} \mathbf{0}_{3 \times 9} & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_{3 \times 9} & \mathbf{0}_{3 \times 6(N-1)} \\
\mathbf{0}_{3 \times 9} & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_{3 \times 9} & \mathbf{0}_{3 \times 6(N-1)}
\end{bmatrix}
\]

(150)

2.4 Composition

After the state augmentation we have the state vector in the form of:

\[
\mathbf{x}_{k+1|k+1} = \begin{bmatrix} R_k^x \mathbf{x}_{k+1|k+1}^T & \mathbf{w}_{k+1|k+1}^T \end{bmatrix}^T
\]

(151)

\[
= \begin{bmatrix} R_k^x & \mathbf{w}_{k+1|k+1} \end{bmatrix}^T
\]

(152)

As we shift local frame of reference at every image time, the corresponding IMU frame, \( \{ I_{k+1} \} \), is selected as the frame of reference (i.e., \( \{ R_k \} \)). The state estimates in the new frame of reference is obtained through:

\[
\dot{\mathbf{x}}_{k+1} = \begin{bmatrix} R_{k+1}^x \dot{\mathbf{x}}_{k+1}^T \end{bmatrix} = \begin{bmatrix} R_{k+1}^x \dot{\mathbf{x}}_{k+1|k+1}^T & \mathbf{w}_{k+1|k+1}^T \end{bmatrix}
\]

(153)

where \( \vec{q}_0 = [0, 0, 0, 1]^T \). Accordingly, the covariance matrix is transformed through:

\[
P_{k+1} = U_{k+1} P_{k+1|k+1} U_{k+1}^T
\]

(155)

\[
U_{k+1} = \frac{\partial \mathbf{x}_{k+1}}{\partial \mathbf{x}_{k+1|k+1}^T} = \begin{bmatrix} \mathbf{V}_{k+1} & \mathbf{0}_{24 \times 6N} \\
\mathbf{0}_{6N \times 24} & \mathbf{I}_{6N}
\end{bmatrix},
\]

(156)

\[
V_{k+1} = \frac{\partial R_{k+1}^x \dot{\mathbf{x}}_{k+1|k+1}^T}{\partial R_k^x \mathbf{x}_{k+1|k+1}^T} = \begin{bmatrix} k+1^1 C_\dot{q}^k & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\
\mathbf{0}_3 & k+1^1 C_\dot{q}^k & \mathbf{0}_3 & \mathbf{0}_3 \\
\mathbf{0}_3 & \mathbf{0}_3 & k+1^1 C_\dot{q}^k & \mathbf{0}_3 \\
\mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3
\end{bmatrix}_{k+1|k+1}
\]

(157)

Note that, the relative pose in IMU state is reset to the origin of the new frame of reference, \( \{ R_k \} \), however, the velocity and biases are evolving in the sensor frame (cf. Eq. (15), (16), and (17)), thus are not affected by the change of frame of reference. In particular, through (157) the covariance of the relative pose is also reset to zero, i.e., no uncertainties for robocentric frame of reference itself. We outline the proposed approach in Algorithm 1.
Algorithm 1 Robocentric Visual-Inertial Odometry

\textbf{Input:} camera images, and IMU measurements
\textbf{Output:} 6DOF real-time robot poses
\textbf{R-VIO:} Initialize the first frame of reference \{R_0\} (=\{G\}) when the first IMU measurement(s) comes in. Then, when a new camera image comes in, do

- **Feature tracking:** extract corner features from the image, then perform KLT tracking and 2-point RANSAC outlier rejection. Record the tracking history for all inliers.

- **Propagation:** propagate state and covariance matrix using preintegration with all IMU measurements starting from the last image time-step.

- **Update:**
  (i) \textit{EKF update:} compute the inverse-depth measurement model matrices for the features whose track are complete (i.e., lost track, or reach the maximum tracking length). Use the features passed the Mahalanobis test for an EKF update.
  (ii) \textit{State augmentation:} augment state vector and covariance matrix with the updated relative pose estimates (state and covariance).

- **Composition:** shift local frame of reference to current IMU frame, update \textit{global} state and covariance using the relative pose estimates. After that, reset relative pose (state and covariance).

2.5 Observability Analysis

System observability reveals whether the information provided by the measurements is sufficient to estimate the state without ambiguity. Since the observability matrix describes the information available in the measurements, by studying its nullspace we gain insight about the directions in the state space from which the system acquires information. To this end, we perform the observability analysis of the proposed robocentric VIO within the EKF-SLAM framework that has the same observability properties as the EKF-VIO provided the same linearization points used [4]. For brevity of presentation, we employ the case that only a single landmark is included in the state vector, while the conclusion can be easily generalized to the case of multiple landmarks. In this case, the state vector at time-step \(k\) includes a landmark \(L\):

\[
x_k = \begin{bmatrix} R_k^T x_G & R_k^T x_{l_k} & R_k^T p_L^T \end{bmatrix}^T = \begin{bmatrix} k^{-1} q & R_k^T p_G^T & k^{-1} q & R_k^T p_{l_k}^T & v_{l_k}^T & b_{gh}^T & b_{ah}^T & R_k^T p_L^T \end{bmatrix}^T
\]

(158)

(159)

and the observability matrix \(M\) is computed as [22]:

\[
M = \begin{bmatrix} H_k \\ \vdots \\ H_{l} \Psi_{l,k} \\ \vdots \\ H_{k+m} \Psi_{k+m,k} \end{bmatrix}
\]

(160)

where \(\Psi_{l,k}\) is the state transition matrix from time-step \(k\) to \(\ell\), and \(H_{l}\) is the measurement Jacobian for the landmark observation at time-step \(\ell\). Each row is evaluated at state, \(x_i\), for \(i = k, \ldots, \ell, \ldots, k + m\). The nullspace of \(M\) describes the directions of the state space for which no information is provided by the measurements, i.e., the unobservable state subspace. Since the proposed robocentric EKF includes three steps: propagation, update, and composition, and the composition step changes the frame of reference, we analyze the observability for a complete cycle of (i) propagation and update, and (ii) composition. We analytically show that the proposed robocentric VIO system has a constant unobservable subspace, and dose \textit{not} have inconsistency issue which is caused by observability mismatch [9, 11], thus improving consistency.
2.5.1 Analytic State Transition Matrix

The analytic form state transition matrix is used for the analysis, for which we integrate the following differential equation over time interval \([t_k, t_\ell]\):

\[
\dot{\Psi}(\ell, k) = \tilde{F}\Psi(\ell, k) \tag{161}
\]

\[
\tilde{F} = \begin{bmatrix}
    0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
    0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
    0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
    0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
    0_3 & 0_3 & 0_3 & -[\hat{\omega} \times] & 0_3 & 0_3 & 0_3 & 0_3 \\
    0_3 & 0_3 & 0_3 & -\frac{\ell^2}{k} C_q^T \{\hat{\nu}_s \times\} & 0_3 & 0_3 & 0_3 & 0_3 \\
    0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
    0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
    \end{bmatrix} \tag{162}
\]

where \(\tilde{F}\) is the augmented state transition matrix assuming zero dynamics for the landmark position in \(\{R_k\}\). By checking Eq. (161) with initial condition \(\Psi(k, k) = I_{27}\), we can get the following structure of \(\Psi\):

\[
\Psi(\ell, k) = \begin{bmatrix}
    I_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
    0_3 & I_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
    0_3 & 0_3 & I_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
    0_3 & 0_3 & 0_3 & I_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
    0_3 & 0_3 & 0_3 & 0_3 & I_3 & 0_3 & 0_3 & 0_3 \\
    0_3 & 0_3 & 0_3 & 0_3 & 0_3 & I_3 & 0_3 & 0_3 \\
    0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & I_3 & 0_3 \\
    0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & I_3 \\
    \end{bmatrix} \tag{163}
\]

We follow an analogous approach to derive the expressions for the block entries of \(\Phi(\ell, k)\) (cf. Appendix A), and the results are listed as follows:

\[
\Psi_{44}(\ell, k) = \frac{\ell}{k} C_q \tag{164}
\]

\[
\Psi_{47}(\ell, k) = -\frac{\ell}{k} C_q \int_{t_k}^{t_\ell} \frac{k}{t_k} C_q^T ds = -\int_{t_k}^{t_\ell} \frac{k}{t} C_q^T ds \tag{165}
\]

\[
\Psi_{53}(\ell, k) = -\frac{1}{2} I_3 \Delta \dot{t}_{k, \ell} \tag{166}
\]

\[
\Psi_{54}(\ell, k) = -[\left(\sum_{R_k} P_{\ell} + \frac{1}{2} \sum_{R_k} \tilde{g} \Delta \dot{t}_{k, \ell}^2\right) \times] \tag{167}
\]

\[
\Psi_{56}(\ell, k) = I_3 \Delta t_{k, \ell} \tag{168}
\]

\[
\Psi_{57}(\ell, k) = \int_{t_k}^{t_\ell} \frac{k}{t} C_q^T [\hat{\nu}_s \times] \int_{t_k}^{t_\ell} \frac{k}{t} C_q^T d\mu ds + [\sum_{R_k} \tilde{g} \times] \int_{t_k}^{t_\ell} \int_{t_k}^{t_\ell} \int_{t_k}^{t_\ell} \frac{k}{t} C_q^T d\lambda d\mu ds 
\]

\[
-\int_{t_k}^{t_\ell} \int_{t_k}^{t_\ell} \int_{t_k}^{t_\ell} \frac{k}{t} C_q^T [\hat{\nu}_s \times] d\mu ds \tag{169}
\]

\[
\Psi_{58}(\ell, k) = -\int_{t_k}^{t_\ell} \int_{t_k}^{t_\ell} \frac{k}{t} C_q^T d\mu ds \tag{170}
\]

\[
\Psi_{63}(\ell, k) = -\frac{\ell}{k} C_q \Delta t_{k, \ell} \tag{171}
\]

\[
\Psi_{64}(\ell, k) = -\frac{\ell}{k} C_q [\sum_{R_k} \tilde{g} \times] \Delta t_{k, \ell} \tag{172}
\]

\[
\Psi_{66}(\ell, k) = \Psi_{44}(\ell, k) \tag{173}
\]

\[
\Psi_{67}(\ell, k) = \frac{\ell}{k} C_q [\sum_{R_k} \tilde{g} \times] \int_{t_k}^{t_\ell} \int_{t_k}^{t_\ell} \frac{k}{t} C_q^T d\mu ds - \int_{t_k}^{t_\ell} \frac{k}{t} C_q [\hat{\nu}_s \times] d\mu ds \tag{174}
\]
\[
\Psi_{0k}(\ell, k) = -\ell_k C_{\bar{q}} \int_{t_k}^{t_\ell} s C_{\bar{q}}^T ds = -\int_{t_k}^{t_\ell} s C_{\bar{q}}^T ds
\] (175)

2.5.2 Measurement Jacobian

At time-step \( \ell \in [t_k, t_{k+m}] \) the estimate of landmark position in \( \{I_\ell\} \) can be expressed as:

\[
I_\ell \hat{p}_L = \ell_k C_{\bar{q}} (R_k \hat{p}_L - R_k \hat{p}_{t_\ell})
\] (176)

According to Eq. (86), the bearing-only measurement model can be expressed as:

\[
z_\ell = \frac{1}{\ell} \begin{bmatrix} x \\ y \end{bmatrix} + n_\ell, \quad C_{\ell} p_L = \begin{bmatrix} x \\ y \end{bmatrix} = \ell_k C_{\bar{q}} \ell_k C_{\bar{q}}^T (R_k \hat{p}_L - R_k \hat{p}_{t_\ell}) + C_{p_L}
\] (177)

where \( \{C_{\bar{q}}, C_{p_L}\} \) is the result of camera-IMU calibration. The corresponding measurement Jacobian is:

\[
H_\ell = H_{\text{proj}} C_{\bar{q}} \ell_k C_{\bar{q}} \begin{bmatrix} 0_3 & 0_3 & \ell_k \end{bmatrix} - I_3 0_{3 \times 9} I_3,
\]

\[
H_{\text{proj}} = \frac{1}{\ell} \begin{bmatrix} 1 & 0 & \ell_x \\ \ell_z & 0 & \ell_y \end{bmatrix},
\]

\[
H_{\theta_\ell} = [\ell_k (R_k \hat{p}_L - R_k \hat{p}_{t_\ell}) \times] \ell_k C_{\bar{q}}^T
\] (178)

2.5.3 Propagation and Update

Based on the results we have, we can write out the \( \ell \)-th row, \( M_\ell \), of \( M \) as follows:

\[
M_\ell = \Pi \Psi_{\ell,k}
\] (179)

\[
H_\ell = H_{\text{proj}} C_{\bar{q}} \ell_k C_{\bar{q}} \begin{bmatrix} 0_3 & 0_3 & \ell_k \end{bmatrix} - I_3 0_{3 \times 9} I_3
\] (180)

\[
H_{\theta_\ell} = [\ell_k (R_k \hat{p}_L - R_k \hat{p}_{t_\ell}) \times] \ell_k C_{\bar{q}}^T
\] (181)

\[
\Rightarrow M_\ell = \Pi = \Pi \begin{bmatrix} 0_3 & 0_3 & \ell_1 & \ell_2 & I_3 \end{bmatrix}
\] (182)

where

\[
\Pi = H_{\text{proj}} C_{\bar{q}} \ell_k C_{\bar{q}}
\] (183)

\[
\Gamma_1 = -\Psi_{53} = \frac{1}{2} \Delta t_{k,\ell}^2
\] (184)

\[
\Gamma_2 = [\ell_k (R_k \hat{p}_L - R_k \hat{p}_{t_\ell}) \times] \ell_k C_{\bar{q}}^T \Psi_{44} - \Psi_{54} = [\ell_k (R_k \hat{p}_L - R_k \hat{p}_{t_\ell}) \times] + \frac{1}{2} [\ell_k g \times] \Delta t_{k,\ell}^2
\] (185)

\[
\Gamma_3 = -\Psi_{56} = -I_3 \Delta t_{k,\ell}
\] (186)

\[
\Gamma_4 = [\ell_k (R_k \hat{p}_L - R_k \hat{p}_{t_\ell}) \times] \ell_k C_{\bar{q}}^T \Psi_{47} - \Psi_{57} = -[\ell_k (R_k \hat{p}_L - R_k \hat{p}_{t_\ell}) \times] + \int_{t_k}^{t_\ell} \ell_k C_{\bar{q}}^T ds - \Psi_{57}
\] (187)

\[
\Gamma_5 = -\Psi_{58}
\] (188)

We note that for generic motion, i.e., \( \omega \neq 0_{3 \times 1} \) and \( a \neq 0_{1 \times 1} \), the values of \( \Psi_{57} \) and \( \Psi_{58} \) are time-varying, then \( \Gamma_4 \) and \( \Gamma_5 \) are linearly independent. Moreover, the value of \( \Delta t_{k,\ell} \) is varying for different time interval, then the stacked \( \Gamma_1, \Gamma_2, \) and \( \Gamma_3 \) are linearly independent. Based on that, we perform the following Gaussian elimination on \( M_\ell \) to facilitate the search for the nullspace:

\[
M_\ell = \Pi \begin{bmatrix} 0_3 & 0_3 & \ell_1 & \ell_2 & I_3 \end{bmatrix}
\] (189)

\[
\Rightarrow M_\ell \sim \Pi \begin{bmatrix} 0_3 & 0_3 & \ell_1 & \ell_2 & I_3 \end{bmatrix}
\] (190)
from which we can find that $\mathbf{M}_\ell$ is rank deficient by 9, and accordingly the nullspace is of rank 9. Specifically, $\forall \ell \geq k$, we can find that the nullspace of $\mathbf{M}$ consists of the following 

$$\text{null}(\mathbf{M}) = \text{span}_{\text{col}} \begin{bmatrix} I_3 & 0_3 & 0_3 \\ 0_3 & I_3 & 0_3 \\ 0_3 & 0_3 & I_3 \end{bmatrix}$$

### Remark 1

The first 6DOF correspond to the orientation (3) and position (3) of the global frame.

### Remark 2

The last 3DOF belong to the same translation (3) simultaneously applied to the landmark(s) and sensor.

#### 2.5.4 Composition

Assuming that the estimates of $R_k \mathbf{x}_\ell$ and $R_k \mathbf{p}_L$ are obtained at time-step $\ell$ and $k$, we have the following linear dynamic model from time-step $k$ to $\ell$ including the state composition:

$$\mathbf{x}_\ell = \mathbf{V}_\ell \mathbf{C}_{\ell,k} \mathbf{x}_k = \mathbf{\Psi}_{\ell,k} \mathbf{x}_k$$

$$\mathbf{V}_\ell = \begin{bmatrix} \ell \mathbf{C}_{\ell,q} & 0_3 & 0_3 & I_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & \ell \mathbf{C}_{\ell,q} & 0_3 & [R_{k} \mathbf{p}_G \times] & -\ell \mathbf{C}_{\ell,q} & 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & \ell \mathbf{C}_{\ell,q} & [R_{k} \mathbf{g} \times] & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & I_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & I_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & I_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & I_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
\end{bmatrix}$$

$$\mathbf{\Psi}_{\ell,k} = \begin{bmatrix} \mathbf{\Psi}_{11} & 0_3 & 0_3 & \mathbf{\Psi}_{14} & 0_3 & 0_3 & \mathbf{\Psi}_{17} & 0_3 & 0_3 \\ 0_3 & \mathbf{\Psi}_{22} & \mathbf{\Psi}_{23} & \mathbf{\Psi}_{24} & \mathbf{\Psi}_{25} & \mathbf{\Psi}_{26} & \mathbf{\Psi}_{27} & \mathbf{\Psi}_{28} & 0_3 \\ 0_3 & 0_3 & \mathbf{\Psi}_{33} & \mathbf{\Psi}_{34} & 0_3 & 0_3 & \mathbf{\Psi}_{37} & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & \mathbf{\Psi}_{36} & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & \mathbf{\Psi}_{67} & \mathbf{\Psi}_{68} & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & \mathbf{\Psi}_{67} & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & \mathbf{\Psi}_{97} & \mathbf{\Psi}_{98} \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & \mathbf{\Psi}_{99} \\
\end{bmatrix}$$

$$\mathbf{\Psi}_{11} = \ell \mathbf{C}_{\ell,q}$$

$$\mathbf{\Psi}_{14} = \mathbf{\Psi}_{44}$$

$$\mathbf{\Psi}_{17} = \mathbf{\Psi}_{47}$$

$$\mathbf{\Psi}_{22} = \ell \mathbf{C}_{\ell,q}$$

$$\mathbf{\Psi}_{23} = -\ell \mathbf{C}_{\ell,q} \mathbf{\Psi}_{53}$$

$$\mathbf{\Psi}_{24} = [R_{k} \mathbf{p}_G \times] \mathbf{\Psi}_{44} - \ell \mathbf{C}_{\ell,q} \mathbf{\Psi}_{54}$$

$$\mathbf{\Psi}_{25} = -\ell \mathbf{C}_{\ell,q}$$
\[ \dot{\Psi}_{26} = -\frac{\ell}{k} C_q \Psi_{56} \]  
\[ \dot{\Psi}_{27} = [R_t \hat{p}_G \times] \Psi_{47} - \frac{\ell}{k} C_q \Psi_{57} \]  
\[ \dot{\Psi}_{28} = -\frac{\ell}{k} C_q \Psi_{58} \]  
\[ \dot{\Psi}_{33} = \frac{\ell}{k} C_q \]  
\[ \dot{\Psi}_{34} = [R_t \hat{g} \times] \Psi_{44} \]  
\[ \dot{\Psi}_{37} = [R_t \hat{g} \times] \Psi_{47} \]  
\[ \dot{\Psi}_{63} = \Psi_{63} \]  
\[ \dot{\Psi}_{64} = \Psi_{64} \]  
\[ \dot{\Psi}_{66} = \Psi_{66} \]  
\[ \dot{\Psi}_{67} = \Psi_{67} \]  
\[ \dot{\Psi}_{68} = \Psi_{68} \]  
\[ \dot{\Psi}_{93} = -\frac{\ell}{k} C_q \Psi_{53} \]  
\[ \dot{\Psi}_{94} = [R_t \hat{p}_L \times] \Psi_{44} - \frac{\ell}{k} C_q \Psi_{54} \]  
\[ \dot{\Psi}_{95} = -\frac{\ell}{k} C_q \]  
\[ \dot{\Psi}_{96} = -\frac{\ell}{k} C_q \Psi_{56} \]  
\[ \dot{\Psi}_{97} = [R_t \hat{p}_L \times] \Psi_{47} - \frac{\ell}{k} C_q \Psi_{57} \]  
\[ \dot{\Psi}_{98} = -\frac{\ell}{k} C_q \Psi_{58} \]  
\[ \dot{\Psi}_{99} = \frac{\ell}{k} C_q \]  

Accordingly at time-step \( \ell \) the measurement model of Eq. (177) becomes linear:

\[ z_\ell = R_t \hat{p}_L + n_\ell, \quad \dot{R}_t \hat{p}_L = \frac{\ell}{k} C_q (R_t \hat{p}_L - R_t \hat{p}_L) \]

for which the measurement Jacobian with respect to \( \hat{x}_\ell \) is in the form of:

\[ \dot{H}_\ell = [0_3 \quad 0_3 \quad 0_3 \quad 0_3 \quad 0_3 \quad 0_3 \quad 0_3 \quad 0_3 \quad I_3] \]

Therefore, for time interval \([t_k, t_\ell]\), after composition we have the corresponding row \( M_\ell \), of \( M \) as follows:

\[ M_\ell = \dot{H}_\ell \Psi_{\ell, k} \]

\[ = [0_3 \quad 0_3 \quad \dot{\Psi}_{93} \quad \dot{\Psi}_{94} \quad \dot{\Psi}_{95} \quad \dot{\Psi}_{96} \quad \dot{\Psi}_{97} \quad \dot{\Psi}_{98} \quad \dot{\Psi}_{99}] \]

\[ = [0_3 \quad 0_3 \quad \dot{\Psi}_{93} \quad \dot{\Psi}_{94} \quad -\frac{\ell}{k} C_q \quad \dot{\Psi}_{96} \quad \dot{\Psi}_{97} \quad \dot{\Psi}_{98} \quad \frac{\ell}{k} C_q] \]

for which in the generic motion case, i.e., \( \omega \neq 0_{3 \times 1} \), and \( a \neq 0_{3 \times 1} \), \( \dot{\Psi}_{93} \), \( \dot{\Psi}_{94} \), \( \dot{\Psi}_{96} \), \( \dot{\Psi}_{97} \), and \( \dot{\Psi}_{98} \) are linearly independent, and the same nullspace as in the propagation and update can be obtained:

\[ \text{null}(M) = \text{span}_{col.} \begin{bmatrix} I_3 & 0_3 & 0_3 \\ 0_3 & I_3 & 0_3 \\ 0_3 & 0_3 & I_3 \\ 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 \end{bmatrix} \]

**Remark 3.** Changing frame of reference does not change the unobservable subspace.

**Remark 4.** The proposed robitricentric system has a constant unobservable subspace, i.e., independent of the linearization points, which not only guarantees the correct unobservable dimensions, but also the desired unobservable directions, thus improving consistency.
2.6 Initialization

It is important to point out that using the proposed robocentric formulation, the filter initialization becomes much simpler, as the states, by design, are simply relative to the sensor’s local frame of reference and typically start from zero without the need to align the initial pose with the fixed global frame. In particular, in our implementation, (i) the initial global pose and IMU relative pose are both set to \( \{ \hat{q}_0, 0_{3 \times 1} \} \), (ii) the initial local gravity is the average of first available accelerometer measurement(s) before moving, and (iii) the initial acceleration bias is obtained by removing gravity effects while the initial gyroscope bias is the average of the corresponding stationary measurements. Similarly, the corresponding uncertainties for the poses are set to zero, while for the local gravity and biases are set to: \( \Sigma_g = \Delta T \sigma_a^2 I_3 \), \( \Sigma_{b_g} = \Delta T \sigma_{wg}^2 I_3 \), and \( \Sigma_{b_a} = \Delta T \sigma_{wa}^2 I_3 \), where \( \Delta T \) is the time length of initialization.

3 Simulation Results

We conducted a series of Monte Carlo simulations under realistic conditions to validate the proposed algorithm. Two metrics are employed to evaluate the performance: (i) the root mean squared error (RMSE), and (ii) the normalized estimation error squared (NEES). The RMSE provides a concise metric of the accuracy of a given filter, while the NEES is a standard criterion for evaluating the consistency. We compared with two world-centric counterparts: the standard MSCKF [1], and the state-of-the-art state-transition observability constrained (STOC)-MSCKF [12] by enforcing correct observability to improve consistency. To ensure a fair comparison, we implemented the compared filters using the same parameters, such as the sliding-window size, and processed the same data (i.e., 50 trails for the scenario of Figure 1 under real MEMS sensor noise and bias level [12]). The comparison results are in Figure 2 (the calculation details refers to [9]) and Table 1 provides average RMSE and NEES values for all of the algorithms, which clearly show that the proposed R-VIO significantly outperforms the standard MSCKF as well as the STOC-MSCKF, in terms of both RMSE (accuracy) and NEES (consistency), attributed to the novel reformulation of the system. Note that in Figure 2c the orientation NEES of proposed method has a jump at the beginning which is primarily due to the small covariance we used for initialization (cf. Section 2.6), however, it can quickly recover and perform consistently only after a short period of time.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Orientation RMSE (°)</th>
<th>Position RMSE (m)</th>
<th>Orientation NEES</th>
<th>Position NEES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Std-MSCKF</td>
<td>3.4700</td>
<td>0.4774</td>
<td>7.0487</td>
<td>5.8103</td>
</tr>
<tr>
<td>STOC-MSCKF</td>
<td>2.5232</td>
<td>0.4305</td>
<td>4.0964</td>
<td>3.7936</td>
</tr>
<tr>
<td>Proposed</td>
<td>0.6811</td>
<td>0.0715</td>
<td>2.4146</td>
<td>1.9061</td>
</tr>
</tbody>
</table>

4 Experimental Results

We further validate the proposed R-VIO experimentally in both indoor and outdoor environments, using both the benchmark public datasets and the datasets collected by our own sensor platform. As described in Algorithm 1, we implemented a C++ multithread framework. In front end, the visual tracking thread extracts features from the image using the Shi-Tomasi corner detector [23], then tracks them pairwise from the previous image using the KLT algorithm [24]. To remove the outliers from tracking result, we realized an IMU-aided two-point RANSAC algorithm [25]. Especially, for the feature lost track we use all its measurements within the sliding window for EKF update, while for the one reaching the maximum tracking length (e.g., the window size) we use its subset (e.g., 1/2) of measurements and maintain the rest for next update. Once the visual tracking is done, the back end processes the measurements using the proposed robocentric EKF. In particular, we present extensive comparison results against a state-of-the-art implementation, Okvis (i.e., open keyframe-based visual-inertial SLAM) \(^2\), which does nonlinear optimization in the back end lit-

\(^2\)https://github.com/ethz-asl/okvis
Figure 1: The scenario (i.e., trajectory and landmarks) in simulation. The camera/IMU pair moves along a circular path of radius 5m (black) at an average speed 1m/s. The camera with 45° field of view observes point features randomly distributed on a circumscribing cylinder of radius 6m (pink). The standard deviation of image noise was set to 1.5 pixels, while IMU measurements were modelled with MEMS sensor quality.

Figure 2: The statistical results of 50 Monte Carlo simulations. Moderately incurring higher computational cost, however, taking the advantage of iterated linearization to limit linearization errors [5]. All the tests run real time on a Core i7-4710MQ @ 2.5GHz × 8, 16GB RAM laptop.
4.1 Highly Dynamic Capability
To show the robustness of the proposed R-VIO under highly dynamic motion, we consider the EuRoC micro aerial vehicle (MAV) dataset where the VI-sensor (ADIS16448 IMU @ 200Hz, MT9V034 cameras @ 20Hz) mounted on a FireFly hex-rotor helicopter was used for data collection [26]. During the tests, only the left camera (cam0) images are used as vision input. The Okvis was set with default parameters, while the R-VIO was set up to maintain 20 relative poses (i.e., about 1 second relative motion record), and uniformly extracted 200 corner features per image. Note that the computational complexity of R-VIO is linear to the number of features for EKF update (cf. Section 2.3), thus significantly reduces the computational cost. The estimated trajectories are shown in Figure 3 and we provide the corresponding pose error results in Figure 4. In general, the proposed R-VIO performs comparably to the Okvis, even better in some sequences.

4.2 Long-Term Sustainability
Aiming at long-term navigation, we further evaluated the accuracy and consistency of the proposed R-VIO with a Long distance dataset, which contains a 14 minutes, 1.2km circular path recorded by the VI-Sensor and has been employed in [5]. We compared the results with [5] which included the results of Okvis and an MSCKF implementation of [27]. Figure 5 shows the estimated trajectories after 6DOF pose alignment, where the z-drifts are also presented to illustrate the altitude performance.

4.3 Versatility
We have also validated the proposed R-VIO both indoor and outdoor, using our own sensor platform that consists of: a MicroStrain 3DM-GX3-35 IMU (@ 500Hz) and a PointGrey Chameleon3 camera (@ 30Hz), and is mounted onto the laptop. Both daytime and nighttime datasets were collected for the indoor test, where we traveled about 150m loop at an average speed of 0.539m/s, covering two floors in a building (with white wall, variant illumination, and strong glare in the hallway), while the outdoor dataset recorded about 360m loop at an average speed of 1.216m/s (with uneven terrain, and opportunistic moving objects). In both tests, due to the lack of the ground truth, to visualize the performance we overlay the results onto the floor plan and the Google Map, respectively (see Figure 6b,6d). The final position errors are about 0.349% (daytime) and 0.615% (nighttime) over the travelled distance in the indoor tests, and about 1.173% in the outdoor test.

5 Conclusion
In this report, we have reformulated the 3D VINS with respect to a moving local frame and developed a lightweight, high-precision, robocentric visual-inertial odometry (R-VIO) algorithm. With this novel reformulation, the resulting VINS does not suffer from the observability mismatch issue encountered in the world-centric systems, thus improving consistency and accuracy. Extensive Monte Carlo simulations and the real-world experiments with different sensor platforms navigating in different environments and using only monocular vision validate our theoretical analysis and show that the proposed R-VIO is versatile and robust to different types of motions and environments. In future, we will focus on improving the proposed approach further, for example, by integrating online calibration and loop closure to deal with sensor parameter variations and bound localization errors.
Figure 3: Trajectory estimates vs. ground truth in EuRoC dataset.

Figure 4: Estimation accuracy (RMSE) in EuRoC dataset.
Figure 5: The results vs. ground truth in Long distance dataset

Figure 6: The results in our Hand-held experiments
Appendix A: Analytic State Transition Matrix

In this part, we analytically derive the block entries of $\Psi(\ell, k)$ employed by Eq. (164)-(175) in Section 2.5.1:

\[
\dot{\Psi}(\ell, k) = \ddot{F} \Psi(\ell, k) \tag{226}
\]

\[
\ddot{F} = \begin{bmatrix}
0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
\end{bmatrix}
\tag{227}
\]

\[
\Psi(\ell, k) = \begin{bmatrix}
I_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
I_3 & I_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
I_3 & 0_3 & I_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
I_3 & I_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
\Psi_{44}(\ell, k) & \Psi_{54}(\ell, k) & \Psi_{64}(\ell, k) & I_3 & \Psi_{56}(\ell, k) & \Psi_{57}(\ell, k) & \Psi_{58}(\ell, k) & \Psi_{68}(\ell, k) \\
\Psi_{63}(\ell, k) & \Psi_{64}(\ell, k) & \Psi_{66}(\ell, k) & I_3 & \Psi_{66}(\ell, k) & \Psi_{67}(\ell, k) & \Psi_{68}(\ell, k) & \Psi_{68}(\ell, k) \\
I_3 & I_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
I_3 & I_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\
\end{bmatrix} \tag{228}
\]

$\Psi_{44}$:

\[
\dot{\psi}_{44}(\ell, k) = -[\dot{\omega} \times] \Psi_{44}(\ell, k) \tag{229}
\]

\[
\Rightarrow \Psi_{44}(\ell, k) = \Psi_{44}(k, k) \exp \left( \int_{t_k}^{t_\ell} -[\dot{\omega} \times] \ ds \right) = \exp \left( \int_{t_k}^{t_\ell} -[\dot{\omega} \times] \ ds \right) = \frac{\dot{\ell}}{k} \dot{\ell}_q \tag{230}
\]

$\Psi_{47}$:

\[
\dot{\psi}_{47}(\ell, k) = -[\dot{\omega} \times] \Psi_{47}(\ell, k) - I_3 \tag{231}
\]

\[
\Rightarrow \dot{\psi}_{47}(\ell, k) + [\dot{\omega} \times] \Psi_{47}(\ell, k) = -I_3 \tag{232}
\]

\[
\Rightarrow \exp \left( \int_{t_k}^{t_\ell} [\dot{\omega} \times] \ ds \right) \Psi_{47}(\ell, k) + \exp \left( \int_{t_k}^{t_\ell} [\dot{\omega} \times] \ ds \right) [\dot{\omega} \times] \Psi_{47}(\ell, k) = -\exp \left( \int_{t_k}^{t_\ell} [\dot{\omega} \times] \ ds \right) \tag{233}
\]

\[
\Rightarrow \frac{d}{dt} \left( \exp \left( \int_{t_k}^{t_\ell} [\dot{\omega} \times] \ ds \right) \Psi_{47}(\ell, k) \right) = -\exp \left( \int_{t_k}^{t_\ell} [\dot{\omega} \times] \ ds \right) \tag{234}
\]

\[
\Rightarrow \frac{d}{dt} \left( \frac{\dot{\ell}}{k} \dot{C}_{q}^T \Psi_{47}(\ell, k) \right) = -\frac{\dot{\ell}}{k} \dot{C}_{q}^T \tag{235}
\]

\[
\Rightarrow \frac{\dot{\ell}}{k} \dot{C}_{q}^T \Psi_{47}(\ell, k) + \Psi_{47}(\ell, k) = -\int_{t_k}^{t_\ell} \frac{\dot{\ell}}{k} \dot{C}_{q}^T \ ds \tag{236}
\]

\[
\Rightarrow \Psi_{47}(\ell, k) = -\int_{t_k}^{t_\ell} \frac{\dot{\ell}}{k} \dot{C}_{q} \ ds = -\int_{t_k}^{t_\ell} \frac{\dot{\ell}}{k} \dot{C}_{q}^T \ ds \tag{237}
\]

$\Psi_{63}$:

\[
\dot{\psi}_{63}(\ell, k) = -\frac{\dot{\ell}}{k} \dot{C}_{q}^T - [\dot{\omega} \times] \Psi_{63}(\ell, k) \tag{238}
\]

\[
\Rightarrow \dot{\psi}_{63}(\ell, k) + [\dot{\omega} \times] \Psi_{63}(\ell, k) = -\frac{\dot{\ell}}{k} \dot{C}_{q} \tag{239}
\]

\[
\Rightarrow \frac{d}{dt} \left( \frac{\dot{\ell}}{k} \dot{C}_{q}^T \Psi_{63}(\ell, k) \right) = -\frac{\dot{\ell}}{k} \dot{C}_{q}^T \dot{C}_{q} = I_3 \tag{240}
\]
\[ \Rightarrow \ell_k C_q^T \Psi_{63}(\ell, k) + \Psi_{63}(k, k) = - \int_{t_k}^{t_\ell} I_3 \, ds \]
\[ \Rightarrow \Psi_{63}(\ell, k) = - \ell_k C_q \Delta t_{k, \ell} \] (241)
\[ \Psi_{64}: \]
\[ \dot{\Psi}_{64}(\ell, k) = - \ell_k C_q \left[ - \ell_k C_q \right] \]
\[ \Rightarrow \dot{\Psi}_{64}(\ell, k) + \dot{\Psi}_{64}(k, k) = - \ell_k C_q \left[ - \ell_k C_q \right] \] (242)
\[ \Psi_{66}: \]
\[ \dot{\Psi}_{66}(\ell, k) = - \ell_k C_q \] (243)
\[ \Rightarrow \dot{\Psi}_{66}(\ell, k) = \ell_k C_q \] (244)
\[ \Psi_{67}: \]
\[ \dot{\Psi}_{67}(\ell, k) = - \ell_k C_q \] (245)
\[ \Rightarrow \dot{\Psi}_{67}(\ell, k) + \dot{\Psi}_{67}(k, k) = - \ell_k C_q \] (246)
\[ \Psi_{68}: \]
\[ \dot{\Psi}_{68}(\ell, k) = - \ell_k C_q \] (247)
\[ \Rightarrow \dot{\Psi}_{68}(\ell, k) = \ell_k C_q \] (248)
\[ \Psi_{69}: \]
\[ \dot{\Psi}_{69}(\ell, k) = - \ell_k C_q \] (249)
\[ \Rightarrow \dot{\Psi}_{69}(\ell, k) = \ell_k C_q \] (250)
\[ \Psi_{70}: \]
\[ \dot{\Psi}_{70}(\ell, k) = - \ell_k C_q \] (251)
\[ \Rightarrow \dot{\Psi}_{70}(\ell, k) + \dot{\Psi}_{70}(k, k) = - \ell_k C_q \] (252)
\[ \Psi_{71}: \]
\[ \dot{\Psi}_{71}(\ell, k) = - \ell_k C_q \] (253)
\[ \Rightarrow \dot{\Psi}_{71}(\ell, k) + \dot{\Psi}_{71}(k, k) = - \ell_k C_q \] (254)
\[ \Psi_{72}: \]
\[ \dot{\Psi}_{72}(\ell, k) = - \ell_k C_q \] (255)
\[ \Rightarrow \dot{\Psi}_{72}(\ell, k) + \dot{\Psi}_{72}(k, k) = - \ell_k C_q \] (256)
\[ \Psi_{73}: \]
\[ \dot{\Psi}_{73}(\ell, k) = \ell_k C_q^T \Psi_{63}(\ell, k) \] (257)
\[ \Rightarrow \dot{\Psi}_{73}(\ell, k) = \ell_k C_q^T \Psi_{63}(\ell, k) \] (258)
\[ \Psi_{74}: \]
\[ \dot{\Psi}_{74}(\ell, k) = - \ell_k C_q^T \Psi_{64}(\ell, k) + \ell_k C_q \Psi_{64}(\ell, k) \] (259)
\[ \Rightarrow \dot{\Psi}_{74}(\ell, k) = \ell_k C_q^T \Psi_{64}(\ell, k) + \ell_k C_q \Psi_{64}(\ell, k) \] (260)
\[ \dot{\Psi}_{54}(\ell, k) = -\ell k C_q D [\dot{v}_I x] k C_q - \{C_q D R_k \dot{g} \times \} \Delta t_{k, \ell} \] (263)

\[ \dot{\Psi}_{54}(\ell, k) = -[k C_q D [\dot{v}_I x] - [R_k \dot{g} \times] \Delta t_{k, \ell} \] (264)

\[ \Psi_{54}(\ell, k) + \Psi_{54}(k, k) = -\int_{t_k}^{t_f} [k C_q D [\dot{v}_I x] \} ds - \int_{t_k}^{t_f} [R_k \dot{g} \times] \Delta t_{k, s} ds \] (265)

\[ \dot{\Psi}_{54}(\ell, k) = -\int_{t_k}^{t_f} [R_k \dot{v}_I x] ds - \frac{1}{2} [R_k \dot{g} \times] \Delta t_{k, \ell}^2 \] (266)

\[ \dot{\Psi}_{54}(\ell, k) = -([R_k \dot{p}_I + \frac{1}{2} R_k \ddot{g} \Delta t_{k, \ell}^2]) \times \] (267)

\[ \Psi_{56}: \]

\[ \dot{\Psi}_{56}(\ell, k) = \ell k C_q D \Psi_{66}(\ell, k) \] (268)

\[ \Rightarrow \dot{\Psi}_{56}(\ell, k) = \ell k C_q D q = I_3 \] (269)

\[ \Rightarrow \Psi_{56}(\ell, k) = [k C_q D q] ds \] (270)

\[ \Rightarrow \Psi_{56}(\ell, k) = I_3 \Delta t_{k, \ell} \] (271)

\[ \Psi_{57}: \]

\[ \dot{\Psi}_{57}(\ell, k) = -\ell k C_q D [\dot{v}_I x] q C_q \] (272)

\[ \Rightarrow \dot{\Psi}_{57}(\ell, k) = \ell k C_q D [\dot{v}_I x] q C_q \] (273)

\[ \Rightarrow \dot{\Psi}_{57}(\ell, k) = \ell k C_q D [\dot{v}_I x] q C_q \] (274)

\[ \Rightarrow \Psi_{57}(\ell, k) + \Psi_{57}(k, k) = \int_{t_k}^{t_f} [k C_q D q] ds + [R_k \dot{g} \times] q C_q \] (275)

\[ \Rightarrow \Psi_{57}(\ell, k) = \int_{t_k}^{t_f} \int_{t_k}^{t_f} [k C_q D q] ds + [R_k \dot{g} \times] q C_q d\lambda ds \] (276)

\[ \Psi_{58}: \]

\[ \dot{\Psi}_{58}(\ell, k) = \ell k C_q D \Psi_{68}(\ell, k) \] (277)

\[ \Rightarrow \dot{\Psi}_{58}(\ell, k) = -\ell k C_q D \] (278)

\[ \Rightarrow \Psi_{58}(\ell, k) = -\int_{t_k}^{t_f} \int_{t_k}^{t_f} k C_q D ds \] (279)

\[ \Rightarrow \Psi_{58}(\ell, k) = -\int_{t_k}^{t_f} \int_{t_k}^{t_f} k C_q D ds \] (280)

**Appendix B: Bundle Adjustment with Euclidean Coordinates**

In this section, we show the nonlinear least-squares method for computing the position of landmark \( L_j \) which has been observed from a set of time-ordered robo-centric frames of reference, \( R_j \). Assume that \( \{R_k\} \) is the
first robocentric frame from which the landmark is observed, the landmark position for \( i \in \mathcal{R}_j \):

\[
R_k \mathbf{p}_{L_j} = i C_q R^i R_{L_j} + R_k \mathbf{p}_{R_i}
\]  
(281)

\[
R_k \mathbf{p}_{L_j} = i C_q C_i \mathbf{p}_{L_j} + i \mathbf{p}_C
\]  
(282)

for which we have the estimate of the relative pose, \( \{ i C_q, R^i R_{R_i} \} \), and the bearing measurement for \( C_i \mathbf{p}_{L_j} \):

\[
\eta_j = \begin{bmatrix}
x_j^i \\
y_j^i \\
z_j^i
\end{bmatrix} = \begin{bmatrix}
x_j
y_j
1
\end{bmatrix}, \quad C_i \mathbf{p}_{L_j} = \begin{bmatrix}
x^i_j \\
y^i_j \\
z^i_j
\end{bmatrix}
\]  
(283)

where \((u_i, v_i)\) is the pixel measurement of \( L_j \) centered at the principal point of image, \( f \) is the focal length. If we denote the depth of landmark as \( \lambda^i_j \), then Eq. (281) can be rewritten as:

\[
R_k \mathbf{p}_{L_j} = i C_q \left( i C_q C_i \mathbf{p}_{L_j} + i \mathbf{p}_C \right) + R_k \mathbf{p}_{R_i}
\]  
(284)

\[
= i C_q \left( i C_q \lambda^i_j \eta_j + i \mathbf{p}_C \right) + R_k \mathbf{p}_{R_i}
\]  
(285)

\[
= \lambda^i_j \eta_j + \mathbf{p}_i + R_k \mathbf{p}_{R_i}
\]  
(286)

\[
= \lambda^i_j \eta_j + \mathbf{p}_i + R_k \mathbf{p}_{R_i}
\]  
(287)

where for brevity we have denoted \( \eta_j = C_q \mathbf{p}_{L_j} \) and \( \mathbf{p}_i = C_q \mathbf{p}_C + R_k \mathbf{p}_{R_i} \). We can find the perpendicular counterpart of \( \eta_j \) as \( \perp \eta_j \), and premultiply it to the above equation, we have:

\[
\perp \eta_j^T R_k \mathbf{p}_j = \perp \eta_j^T \mathbf{p}_i
\]  
(288)

from which we can form two linearly independent constraints for \( R_k \mathbf{p}_{L_j} \) (e.g., for \( \eta_j = [\eta_x, \eta_y, \eta_z]^T \), we have \( \perp \eta_{j,1} = [\eta_y, -\eta_x, 0]^T \) and \( \perp \eta_{j,2} = [\eta_z, 0, -\eta_z]^T \)):

\[
\begin{bmatrix}
\perp \eta_{j,1}^T \\
\perp \eta_{j,2}^T
\end{bmatrix}
\Rightarrow \mathbf{A}_i R_k \mathbf{p}_{L_j} = \mathbf{y}_i
\]  
(289)

(290)

If we stack all the resulting equations for \( i \in \mathcal{R}_j \), then the following linear least-squares problem is formulated:

\[
\mathbf{A}_{2N \times 3} R_k \mathbf{p}_{L_j} = \mathbf{y}_{2N \times 1}
\]  
(291)

where the size of \( \mathcal{R}_j \): \( N \geq 2 \) (including \( k \)) for solving the problem, which means the positions of landmarks observed from at least two robocentric frames of reference can be solved: \( R_k \mathbf{p}_{L_j} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{y} \).

Once obtaining an initial estimate for \( R_k \mathbf{p}_{L_j} \), we can form a nonlinear least-squares problem using the measurements of \( L_j \) to refine the value of \( R_k \mathbf{p}_{L_j} \). If we denote \( R_k \mathbf{p}_{L_j} = [x, y, z]^T \), then for \( i \in \{ \mathcal{R}_j \backslash k \} \):

\[
C_i \mathbf{p}_{L_j} = i C_q i R^i R_{L_j} + C_i \mathbf{p}_I
\]  
(292)

\[
= i C_q \left( i C_q (R_k \mathbf{p}_{L_j} - R_k \mathbf{p}_{R_i}) \right) + C_i \mathbf{p}_I
\]  
(293)

\[
:= \begin{bmatrix}
h_{i,1}(x, y, z) \\
h_{i,2}(x, y, z) \\
h_{i,3}(x, y, z)
\end{bmatrix}
\]  
(294)

However, for seeking the numerical stability of computation, we adopt an equivalent representation of \( R_k \mathbf{p}_{L_j} \):

\[
[x, y, z]^T = z[\frac{x}{z}, \frac{y}{z}, 1]^T
\]  
(assuming depth \( z \) is the most variant direction), then:

\[
C_i \mathbf{p}_{L_j} = i C_q i C_q \left( \frac{x}{z} - R_k \mathbf{p}_{R_i} \right) + C_i \mathbf{p}_I
\]  
(295)

RPNG-2018-RVIO
Once the result of Eq. (304) get converged, we find the optimal parameters: \( X \) coordinates (i.e., divided by the focal length \( f \) where \( n \)), such as multiplying \( \beta \), \( \rho \) in the fraction form), such as multiplying \( \beta \), \( \rho \) be reformulated with arbitrary scale while still fits for the bearing expression (as the scale will be cancelled in the fraction form), such as multiplying \( \beta \), \( \rho \). Given the measurements \( z_i = \frac{\hat{\delta}_X}{\hat{\delta}_Y} \), we can form the following nonlinear least-squares problem for solving \( X = [\alpha, \beta, \rho]^T \):

\[
X^* = \arg \min_X \sum_{i \in \{R_i \backslash k\}} \|h_i(X) - z_i\|_A^2 = \arg \min_X \sum_{i \in \{R_i \backslash k\}} \|\varepsilon_i(X)\|_A^2
\]

By substituting Eq. (300) into Eq. (86), we express the measurement model as the function of \( \alpha, \beta, \rho \):

\[
z_i = \frac{1}{h_{i,3}(\alpha, \beta, \rho)} \begin{bmatrix} h_{i,1}(\alpha, \beta, \rho) \\ h_{i,2}(\alpha, \beta, \rho) \\ h_{i,3}(\alpha, \beta, \rho) \end{bmatrix} + n_i
\]

where \( n_i \sim \mathcal{N}(0, A_i) \) is a zero-mean white Gaussian noise corresponding to the scale of the normalized pixel coordinates (i.e., divided by the focal length \( f \)). Given the measurements \( z_i = (\frac{\hat{\delta}_X}{\hat{\delta}_Y}, \frac{\hat{\delta}_Y}{\hat{\delta}_X}) \), we can form the following nonlinear least-squares problem for solving \( X = [\alpha, \beta, \rho]^T \):

\[
X^* = \arg \min_X \sum_{i \in \{R_i \backslash k\}} \|h_i(X) - z_i\|_A^2 = \arg \min_X \sum_{i \in \{R_i \backslash k\}} \|\varepsilon_i(X)\|_A^2
\]

where \( \|\cdot\|_A \) denotes the \( A \)-weighted energy norm, and we define \( \varepsilon_i \) as the residual associated to \( z_i \). This problem can be solved iteratively via Gauss-Newton approximation about the initial estimate of \( \hat{X} \):

\[
\delta X^* = \arg \min_{\delta X} \sum_{i \in \{R_i \backslash k\}} \|\varepsilon_i(\hat{X} + \delta X)\|_A^2 \approx \arg \min_{\delta X} \sum_{i \in \{R_i \backslash k\}} \|\varepsilon_i(\hat{X}) + H_i \delta X\|_A^2
\]

For which the Jacobian of residual \( H_i = \frac{\partial \varepsilon_i(\hat{X} + \delta X)}{\partial \delta X} \) evaluated at \( \hat{X} \) can be obtained following the chain rule:

\[
H_i = \frac{\partial \varepsilon_i}{\partial \hat{p}_{L_i}} \frac{\partial \hat{p}_{L_i}}{\partial \hat{X}} \frac{\partial \hat{X}}{\partial \delta X} = \frac{\partial \varepsilon_i}{\partial \hat{p}_{L_i}} \frac{\partial \hat{p}_{L_i}}{\partial \hat{X}} \frac{\partial \hat{X}}{\partial \delta X}
\]

\[
\frac{\partial \varepsilon_i}{\partial \hat{p}_{L_i}} = \frac{1}{h_{i,3}} \begin{bmatrix} 1 & 0 & -\frac{h_{i,1}}{h_{i,3}} \\ 0 & 1 & -\frac{h_{i,2}}{h_{i,3}} \end{bmatrix},
\]

\[
\frac{\partial \hat{p}_{L_i}}{\partial \hat{X}} = \begin{bmatrix} \frac{\partial \hat{p}_{L_i}}{\partial [\alpha, \beta]^T} \\ \frac{\partial \hat{p}_{L_i}}{\partial \rho} \end{bmatrix} = \begin{bmatrix} C_{q^i}^i \beta & 0 & \alpha \end{bmatrix} \frac{R_i \hat{p}_{R_i} + \hat{C}_p}{\hat{C}_p} + \frac{\hat{C}_p}{\hat{C}_p} \hat{C}_p
\]

Once the result of Eq. (304) get converged, we find the optimal parameters: \( X^* = \hat{X} + \delta X^* \), thus \( R_i \hat{p}_{L_i} \).
Appendix C: Bundle Adjustment with Inverse Depth Parameters

Assuming that we have the same landmark $L_j$ as in Appendix B, the expression for $R_k p_{L_j}$ using our tailored inverse depth parametrization (cf. Section 2.3.2) is shown as:

\[ R_k p_{L_j} = \frac{I}{C} C_q^i C_k p_{L_j} + \frac{I}{C} p_C \]  
\[ \frac{1}{C} p_C = \frac{1}{\rho} \mathbf{e}(\phi, \psi), \quad \mathbf{e} = \begin{bmatrix} \cos \phi \sin \psi \\ \sin \phi \\ \cos \phi \cos \psi \end{bmatrix} \]  

where $\mathbf{e}$ is the unit directional vector of the ray from $\{ C_k \}$ with $\phi$ and $\psi$ the elevation and azimuth expressed in $\{ C_k \}$, respectively, and $\rho$ denotes the inverse depth of landmark along $\mathbf{e}$. For $i \in \{ R_j \backslash k \}$, we have:

\[ R^i p_{L_j} = \frac{1}{k} C_q^i C_q^k R_k p_{L_j} + R_k p_{R_k} \]  
\[ = \frac{1}{k} C_q^i \left( C_q^k \frac{C_k}{C} p_{L_j} + \frac{I}{C} p_C + R_k p_{R_k} \right) \]  
\[ = \frac{1}{k} C_q^i \left( C_q^k \frac{1}{\rho} - e + \frac{I}{C} p_C + R_k p_{R_k} \right) \]  
\[ = \frac{1}{k} C_q^i C_q^k \frac{C_k}{C} p_C + \frac{1}{k} C_q^i C_q^k R_k p_{R_k} + \frac{1}{k} C_q^i p_C + \frac{1}{k} p_k \]  

Then, the position of landmark in the corresponding camera frame can be expressed as:

\[ C_i p_{L_j} = \frac{C_i}{C} p_{C_q^k} + C_i p_C \]  
\[ C_i p_C = \left( \frac{1}{\rho} C_q^i C_q^k \frac{C_k}{C} p_C + \frac{1}{\rho} C_q^i C_q^k R_k p_{R_k} \right) \]  
\[ + \frac{1}{\rho} C_q^i C_q^k \frac{C_k}{C} p_C + \frac{1}{\rho} C_q^i \frac{1}{\rho} - e + \frac{1}{\rho} p_C + \frac{1}{\rho} p_k \]  

where $\frac{1}{k} C_q^i = \frac{1}{i} C_q^i C_q^k C_q^k C_q^i$ and $\frac{1}{k} p_k = \frac{1}{i} C_q^i C_q^k C_q^k \frac{1}{C} p_C + \frac{1}{i} C_q^i R_k p_{R_k} + \frac{1}{i} p_C$ represent the relative pose between the camera frame $\{ C_k \}$ and $\{ C_i \}$. If we only consider the geometry in the above equation, then it is able to be reformulated with arbitrary scale, such as being normalized by $\rho$, which results in the following expression:

\[ \rho \frac{C_i}{C} p_{L_j} = \frac{1}{\rho} C_q^i C_q^k \frac{C_k}{C} e + \rho \frac{1}{\rho} p_k \]  
\[ \Rightarrow \frac{1}{\rho} p_{L_j} = \begin{bmatrix} h_{i,1}(\phi, \psi, \rho) \\ h_{i,2}(\phi, \psi, \rho) \\ h_{i,3}(\phi, \psi, \rho) \end{bmatrix} \]  

where $h_{i,1}$, $h_{i,2}$, and $h_{i,3}$ are the scalar functions of the entries of $\frac{1}{\rho} p_{L_j}$. By substituting Eq. (319) into Eq. (86), the measurement model can be expressed as the function of $\phi$, $\psi$, and $\rho$:

\[ z_i = \frac{1}{h_{i,3}(\phi, \psi, \rho)} \begin{bmatrix} h_{i,1}(\phi, \psi, \rho) \\ h_{i,2}(\phi, \psi, \rho) \end{bmatrix} \]  
\[ = h_i(\phi, \psi, \rho) + n_i \]  

where $n_i \sim \mathcal{N}(0, \Lambda_i)$ is a zero-mean white Gaussian noise corresponding to the scale of the normalized pixel coordinates (i.e., divided by the focal length $f$). Given the measurements $z_i = (\frac{u_i}{f}, \frac{v_i}{f})$, $i \in \{ R_j \backslash k \}$, we can form the following nonlinear least-squares problem for solving $\lambda = [\phi, \psi, \rho]^T$:

\[ \lambda^* = \arg\min_{\lambda} \sum_{i \in \{ R_j \backslash k \}} \| h_i(\lambda) - z_i |_{\Lambda_i} = \arg\min_{\lambda} \sum_{i \in \{ R_j \backslash k \}} \| c_i(\lambda) |_{\Lambda_i} \]  

RPNG-2018-RVIO 30
where $\| \cdot \|_\Lambda$ denotes the $\Lambda$-weighted energy norm, and we define $\epsilon_i$ as the residual associated to $z_i$. This problem can be solved iteratively via Gauss-Newton approximation about the initial estimate of $\hat{\lambda}$:

$$
\delta \lambda^* = \arg \min_{\delta \lambda} \sum_{i \in \{R \setminus k\}} \| \epsilon_i(\hat{\lambda} + \delta \lambda) \|_{\Lambda}, \simeq \arg \min_{\delta \lambda} \sum_{i \in \{R \setminus k\}} \| \epsilon_i(\hat{\lambda}) + H_i \delta \lambda \|_{\Lambda},
$$

(323)

For the initial estimate $\hat{\lambda}$, we can get $[\hat{\phi}, \hat{\psi}]^T$ using the pixel measurement $(u_k, v_k)$ and the focal length $f$ as:

$$
\begin{bmatrix}
\hat{\phi} \\
\hat{\psi}
\end{bmatrix} = \begin{bmatrix}
\arctan(v_k, \sqrt{u_k^2 + f^2}) \\
\arctan(u_k, f)
\end{bmatrix} = \begin{bmatrix}
\arctan\left(\frac{u_k}{f}, \sqrt{\left(\frac{u_k}{f}\right)^2 + 1}\right) \\
\arctan\left(\frac{v_k}{f}, 1\right)
\end{bmatrix}
$$

(324)

however, the initial value for $\hat{\rho}$ is empirically chosen (e.g., 0.1 [21]), for which we choose 0 to put landmark at infinity first, and let it converge by performing iteration. The Jacobian of residual $H_i = \frac{\partial \epsilon_i(\hat{\lambda} + \delta \lambda)}{\partial \delta \lambda}$ evaluated at $\hat{\lambda}$ can be obtained following the chain rule:

$$
H_i = \frac{\partial \epsilon_i}{\partial \hat{\phi}_j} \frac{\partial \hat{\phi}_j}{\partial \lambda} \frac{\partial \lambda}{\delta \lambda} + \frac{\partial \epsilon_i}{\partial \hat{\psi}_j} \frac{\partial \hat{\psi}_j}{\partial \lambda} \frac{\partial \lambda}{\delta \lambda} + \frac{\partial \epsilon_i}{\partial \hat{\phi}_j} \frac{\partial \hat{\phi}_j}{\partial \lambda} \frac{\partial \lambda}{\delta \lambda}
$$

(325)

$$
\frac{\partial \epsilon_i}{\partial \hat{\phi}_j} = \frac{1}{h_{i,3}} \begin{bmatrix}
1 & 0 & \frac{h_{i,1}}{h_{i,3}} \\
0 & 1 & \frac{h_{i,2}}{h_{i,3}}
\end{bmatrix},
$$

(326)

$$
\frac{\partial \hat{\phi}_j}{\partial \lambda} = \left[ \frac{\partial \hat{\phi}_j}{\partial \phi} \frac{\partial \hat{\phi}_j}{\partial \psi} \right] = \begin{bmatrix}
- \sin \hat{\phi} \sin \hat{\psi} & \cos \hat{\phi} \cos \hat{\psi} & 0 \\
\cos \hat{\phi} & - \sin \hat{\phi} \cos \hat{\psi} & - \cos \hat{\phi} \sin \hat{\psi}
\end{bmatrix},
$$

(327)

Once the result of Eq. (323) get converged, we find the optimal parameters: $\lambda^* = \hat{\lambda} + \delta \lambda^*$, thus $R_k p^*_L$.

References


