On-Line Initialization and Extrinsic Calibration of an Inertial Navigation System With a Relative Preintegration Method on Manifold

Dongshin Kim, Seunghak Shin, *Student Member, IEEE*, and In So Kweon, *Member, IEEE*

***Abstract*— Inertial measurement units (IMUs) are successfully utilized to compensate localization errors in sensor fused inertial navigation systems. An IMU generally produces high-frequency signals ranging from 100 to 1000 Hz, and preintegration methods are applied to effectively process these high-frequency signals for inertial navigation systems. The main problem with an existing preintegration method is that the inertial propagation models in the method are only generated at the IMU's coordinate system. Hence, the models have to be converted to the coordinate system of the other sensor in order to apply its constraint. So, the iterative optimization framework using the conventional method takes large amount of time. In addition, since a general rigid body transformation cannot transfer a velocity propagation model to the other coordinate system, the concept of relative motion analysis needs to be considered. To solve the problems above, in this paper, we propose a novel relative preintegration (RP) method that can generate inertial propagation models at any sensor's coordinate system in a rigid body. This permits accurate and fast IMU processing in sensor fused inertial navigation systems. We applied new nonlinear optimization frameworks to solve initialization and extrinsic calibration problems for the IMU–IMU, IMU–Camera, and IMU–LiDAR pair based on the proposed RP method in an on-line manner, and the superior results of the mentioned processes are presented as well.**

***Note to Practitioners*—In the proposed method, extrinsic calibration parameters can be estimated during the localization process without any initial manual precalibration procedure. Besides, an IMU is small and inexpensive, so small AGVs in warehouses, or mobile manipulators in factories are good applications for the proposed method, where they require a certain level of localization accuracy. The localization module using the proposed method can be mounted anywhere in a robot without restrictions. The initialization procedure has been executed for a few seconds with arbitrary movement in the field of view of the first frame, then the initial gravity and velocity vector, and extrinsic parameters, which are estimated during the initialization phase, are applied to a general SLAM problem.**

Manuscript received September 26, 2017; accepted November 5, 2017. This paper was recommended for publication by Associate Editor Z. Li and Editor J. Wen upon evaluation of the reviewers' comments. This work was supported in part by the Korea Evaluation Institute of Industrial Technology, Republic of Korea, through the project "Development of core technology for advanced locomotion/manipulation based on high-speed/power robot platform and robot intelligence," and in part by the Ministry of Trade, Industry and Energy funded by the Technology Innovation Program, South Korea, under Grant 2017-10069072. *(Corresponding author: In So Kweon.)*

The authors are with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea (e-mail: dongshin@kaist.ac.kr; shshin@rcv.kaist.ac.kr; iskweon77@kaist.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2017.2773515

***Index Terms*— Extrinsic calibration, inertial measurement unit (IMU) initialization, inertial navigation system, localization, SLAM.**

I. Introduction

AN INERTIAL measurement unit (IMU) has been widely used in sensor fused inertial navigation systems. The most popular system is a visual-aided inertial navigation system (VINS), which uses a camera with an IMU as in [1]–[8]. An IMU can also be used to control manipulators in robotic exoskeletons. Because it generates high-frequency signals and has complementary sensing capability with exteroceptive sensors such as force sensors, it is suitable to track human motions compensating errors. With the precisely tracked motions, the results of manipulator controls as shown in [9]–[12] can be improved.

To estimate a position of a robot using linear acceleration and angular velocity signals, they have to be double integrated; hence, a tiny noise or bias in either of the signals will generate large inaccuracies. This issue is usually resolved by modeling the inaccuracies using filtering or optimization processes, but the computational cost becomes high because the updating steps in the processes inevitably require many reintegrations of the signals. To avoid the computational penalty, a preintegration theory was introduced, as in [13], and has been widely used in the mobile robot research area, as shown in [1] and [14]–[18].

Most inertial navigation systems apply inertial propagation models to solve localization problems. The models consist of evolutions in orientation, velocity, and position. Since the models are generated at the IMU's coordinate system using the existing preintegration method, each model has to be converted to the coordinate system of the other sensor in order to apply its constraint. Then, it must be converted back to the coordinate system of the IMU again to apply the inertial constraint. Transferring orientation and position propagation models from the IMU's coordinate system to the other coordinate system can be done by applying a simple rigid body transformation, but a velocity propagation model cannot be easily transformed. So, the concept of relative motion analysis needs to be considered.

To solve this issue, we propose a novel relative preintegration (RP) method that can generate inertial propagation models at any location in a rigid body. This permits accurate and fast

IMU processing in sensor fused inertial navigation systems. We also present the results of initialization and extrinsic calibration processes that apply to various navigation systems, to demonstrate the advantages of the proposed method.

In this paper, three contributions are claimed as follows.

1) We propose a new inertial preintegration method that includes the concept of relative motion analysis. This provides a way to transform inertial propagation models to another coordinate system in an appropriate manner.
2) We present nonlinear optimization frameworks to estimate initial condition variables and extrinsic calibration parameters based on the proposed preintegration method.
3) We present the results of the initialization and extrinsic calibration process in the IMU–IMU, IMU–Camera, and IMU–LiDAR navigation system.

The remainder of this paper is organized as follows. In Section II, current attempts to solve initialization and extrinsic calibration problems in inertial navigation systems are introduced. The development of the RP approach is described in Section III. In the same section, the handling of noise and bias inherently included in the IMU is also introduced. The initialization including the extrinsic calibration process of each system is described in Section IV. The results of the processes in the IMU–IMU, IMU–Camera, and IMU–LiDAR navigation system are introduced in Section V. Finally, the conclusions of this paper and the future research directions are discussed in Section VI.

II. Related Works

The hand/eye calibration technique first introduced in [19] has been modified and used in the extrinsic calibration process of the IMU–Camera navigation system. The accuracy of the hand/eye method depends on the accuracy of each pose that is computed by the camera and the IMU. Since the estimated pose obtained using inertial information can be easily deviated, it is necessary to have external corrections which require extrinsic calibration parameters in advance.

In [20], the rotation between the camera and the IMU was obtained by comparing the camera's attitude with a gravity vector. The attitude of the camera was estimated using visual markers, and the gravity vector was estimated using acceleration signals. The translation between the two sensors was obtained from orientations of the camera and the IMU. The orientation of the camera was estimated using the visual markers as well, and the orientation of the IMU was estimated using the turntable equipped with the IMU and encoder in the center. The encoder is not supposed to be affected by centripetal acceleration because it is located in the center of the axis. The exact orientation value of the encoder is known, and subsequently the translation can be obtained. But, the IMU may not lie precisely on the turntable's rotation axis due to the task settings. In addition, the framework is not able to handle time-correlation effects generated from continuous IMU signals with bias.

In [21], an extended Kalman filter was used to explicitly model IMU bias and provide the figure of merit (covariance) of the model to handle time-correlation effects. They also demonstrated that the extrinsic calibration process of the IMU–Camera navigation system was observable using observability-ranking criteria based on Lie derivatives. Kelly and Sukhatme [22] adapted an unscented Kalman filter to linearize the calibration system. An optimization process generally outperforms a filtering process because the filtering process only provides approximations of models, and it describes the current state only by the latest state. As a result, evolving parameters cannot be reapplied to the entire system again after they have matured. To address this, a nonlinear optimization process was proposed in [23] to solve the calibration problem. B-spline curves were used to model continuous motion and solved time synchronization issue between the camera and the IMU. Their work was not carried out in real time. A B-spline interpolation essentially involves some level of motion approximation. Hence, the number of control points in the B-spline curve needs to be increased when the system deals with highly dynamic motion. The number of knots is determined empirically. Also, estimating the optimal number and position of the control points is not trivial.

In order to resolve the timing issue, Furgale *et al.* [24] proposed a framework that jointly estimated temporal offsets between the measurements of different sensors and their extrinsic parameters. The use of B-spline curves for motion model interpolation is still limited.

In [25], the rotation between the camera and the IMU was first obtained using a convex optimization method, then the other initial conditions were obtained by a quadratic constrained least squares optimization. Because of the large error bound in the estimated translation between the two sensors, this approach may not be applied to practical applications. In [18], the VINS's initialization and extrinsic calibration problem with an on-line approach was studied. A conventional preintegration (CP) method was applied. This prevented frequent reintegrations of high-speed IMU signals and mitigated timing problems, ensuring a real-time process during the iterative optimization phase. Their work is similar to our works in that they used a preintegration method. Because of the limitation of the CP method, inertial propagation models were limited to the location of the IMU. As a result, the visual features based on the camera's coordinate system were transformed to the IMU's coordinate system, then they were applied to the inertial propagation model to estimate a translation between the camera and the IMU. Their work also first obtained the rotation, and translation later, as in [25].

In contrast to the previous works, our new preintegration method involves the concept of relative motion analysis, so the propagation model using our method can be used explicitly at another location; hence, when iterative operations are performed during the optimization process, no transforms are required to apply another sensor's constraints.

In [1], the manifold preintegration theory was introduced, and our work is similar to their work in the sense that the same rotation group is used.

III. Relative Preintegration

The RP method is inspired by the concept of relative motion analysis as in [26]. As shown in Fig. 1, the total acceleration

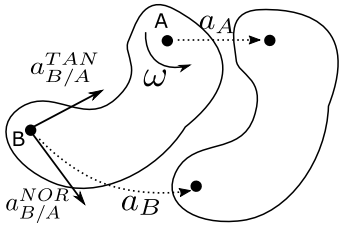

Fig. 1. Relative motion analysis. The body moves with the linear acceleration, a_A, and the angular velocity, ω around the point, A. The total acceleration of B can be obtained by the equation, $a_B = a_A + a_{B/A}^{\text{TAN}} + a_{B/A}^{\text{NOR}}$. The relative acceleration consists of tangential($a_{B/A}^{\text{TAN}}$) and normal($a_{B/A}^{\text{NOR}}$) components of accelerations.

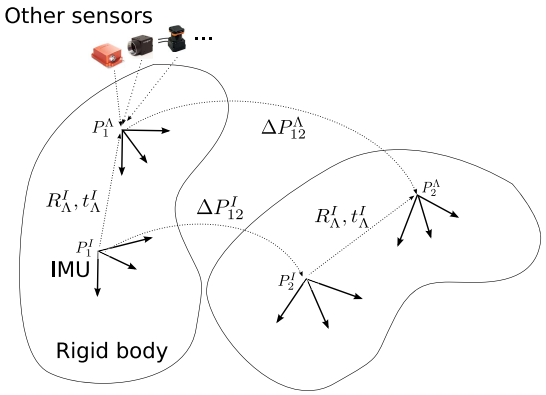

Fig. 2. Rigid body movement. R_Λ^I, t_Λ^I are the rotation and translation between the IMU and the other sensor, respectively. ΔP_{12}^I is the CP measurement and ΔP_{12}^Λ represents the RP measurement.

of the point B is the sum of the relative accelerations, $a_{B/A}^{\text{TAN}}, a_{B/A}^{\text{NOR}}$, which are caused by circular motion with the angular velocity, ω, and the linear acceleration of the body, a_A. The RP measurements are generated by adding new measurements that integrate the relative accelerations to the existing preintegration measurements. The new measurements are expressed by the terms, $\Delta v_{ij}^R t_\Lambda^I$ and $\Delta p_{ij}^R t_\Lambda^I$ as in (1), and their derivations are shown in Appendix A.

A. Relative Preintegration Measurements and Inertial Propagation Models

In Fig. 2, the coordinate of the IMU is P^I and the coordinate of the other sensor is P^Λ where Λ represents the other sensor. When the IMU moves from P_1^I to P_2^I, Λ will move from P_1^Λ to P_2^Λ. The rotation and translation between the IMU and Λ are described by R and t, assuming that the two sensors are attached to the rigid body and move together. The existing preintegration measurement is denoted by ΔP_{12}^I, which describes the motion from P_1^I to P_2^I, and the RP measurement is ΔP_{12}^Λ, which describes the motion from P_1^Λ to P_2^Λ. Assuming that Λ cannot produce the linear acceleration and angular velocity at that location directly, we use relative accelerations to convert the linear acceleration and angular velocity of P^I to those of P^Λ. The preintegration measurement is $\Delta P = \{\Delta p, \Delta v, \Delta\phi\}$, where Δp is a position change, Δv is a velocity change, and $\Delta\phi$ is a change in rotation. The following equations describe RP measurements:

$$\begin{aligned}
\Delta\phi_{ij}^\Lambda &= \Delta\phi_{ij}^I \\
\Delta v_{ij}^\Lambda &= \Delta v_{ij}^I + \Delta v_{ij}^R t_\Lambda^I \\
\Delta p_{ij}^\Lambda &= \Delta p_{ij}^I + \Delta p_{ij}^R t_\Lambda^I
\end{aligned} \tag{1}$$

where $\Delta\phi_{ij}^I$, Δv_{ij}^I, and Δp_{ij}^I represent rotation, velocity, and position change at the IMU's coordinate system between frames i and j, respectively. They are CP measurements as described in (2). t_Λ^I is the translation between two sensors, called a lever arm. Likewise, $\Delta\phi_{ij}^\Lambda$, Δv_{ij}^Λ, and Δp_{ij}^Λ represent rotation, velocity, and position change between frames i and j, respectively, at Λ's location. Note that $\Delta\phi_{ij}^I$ is the same as $\Delta\phi_{ij}^\Lambda$. This is because orientations of any two points in a rigid body are the same. The foundations of relative motion are described by Δv_{ij}^R and Δp_{ij}^R as in (3), and their derivations are given in Appendix A

$$\begin{aligned}
\Delta\phi_{ij}^I &= \prod_{k=i}^{j-1} \exp\left(\omega_k \Delta t\right) \\
\Delta v_{ij}^I &= \sum_{k=i}^{j-1} \Delta\phi_{ik}^I a_k \Delta t \\
\Delta p_{ij}^I &= \sum_{k=i}^{j-1} \left(\Delta v_{ik}^I \Delta t + \frac{1}{2} \Delta\phi_{ik}^I a_k \Delta t^2 \right)
\end{aligned} \tag{2}$$

where a_k is a linear acceleration at k and ω_k is an angular velocity at k in the IMU's coordinate system. Δt is elapsed time at k

$$\begin{aligned}
\Delta v_{ij}^R &= \sum_{k=i}^{j-1} \Delta\phi_{ik}^I \left(\left[\alpha_k^I\right]_\times + \left[\omega_k^I\right]_\times^2 \right) \Delta t \\
\Delta p_{ij}^R &= \sum_{k=i}^{j-1} \left(\Delta v_{ik}^R \Delta t + \frac{1}{2} \Delta\phi_{ik}^I \left(\left[\alpha_k^I\right]_\times + \left[\omega_k^I\right]_\times^2 \right) \Delta t^2 \right)
\end{aligned} \tag{3}$$

where α_k^I is the angular acceleration at k. A $[\cdot]_\times$ operation represents a skew symmetric matrix.

New inertial propagation models which are based on the RP measurements can be obtained as follows:

$$\begin{aligned}
\phi_j^\Lambda &= \phi_i^\Lambda R_I^\Lambda \Delta\phi_{ij}^I R_\Lambda^I \\
v_j^\Lambda &= v_i^\Lambda + R_I^\Lambda g_0^I \Delta t_{ij} + \phi_i^\Lambda R_I^\Lambda \Delta v_{ij}^\Lambda \\
p_j^\Lambda &= p_i^\Lambda + v_i^\Lambda \Delta t_{ij} + \frac{1}{2} R_I^\Lambda g_0^I \Delta t_{ij}^2 + \phi_i^\Lambda R_I^\Lambda \Delta p_{ij}^\Lambda
\end{aligned} \tag{4}$$

where ϕ_j^Λ, v_j^Λ, and p_j^Λ represent the orientation, velocity, and position state at Λ, respectively. $R_I^\Lambda \in \text{SO}(3)$ is the rotation of the frame I with respect to the frame Λ, R_Λ^I is the inverse of R_I^Λ, and g_0^I represents the initial gravity at the IMU's coordinate system.

B. Why Relative Preintegration?

Generally, a system, which uses inertial propagation models, operates in the IMU's coordinate system, so it is necessary to transfer Λ's constraints to the IMU's coordinate system in order to apply inertial propagation constraints. After the

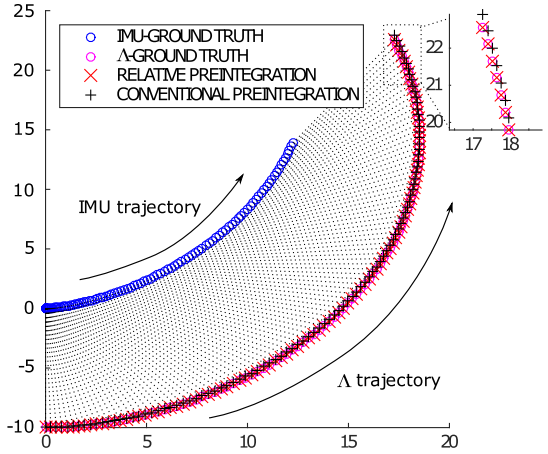

Fig. 3. Ground truth positions of the IMU and Λ, and the predicted positions of Λ using the CP method and the RP method.

constraints are applied, Λ's constraints might be converted back to its own frame to perform its own process, as shown in the linear camera measurement models in [25]. In that case, at least two conversions are required. Using the RP method, the time complexity can be reduced by removing the conversions during the optimization phase, because the inertial models are created in the Λ's coordinate system.

Another advantage is that the proposed method is perfectly applicable to a unified framework of any sensor fused inertial navigation system. In general, to work with existing methods in an integrated framework, relative motion dynamics have to be added separately as shown in (5), which degrades performance. However, the RP method already contains circular motion so that our preintegration prevents the performance degradation.

The following simulation compares the propagation models of the existing and RP method. Assume that there is a system which has two sensors, the IMU and Λ. A linear acceleration, $a^I = [0.05\ \mathrm{m/s^2}, 0\ \mathrm{m/s^2}, 0\ \mathrm{m/s^2}]^\intercal$, and an angular velocity, $\omega^I = [0°, 0°, 1°]^\intercal$ are continuously simulated to the IMU at intervals of 0.001 s for 30 s. The initial gravity is ignored for simplicity because the gravity variable has no effect on the results of this simulation. The orientation and translation between the IMU and Λ are given, such as $R^I_\Lambda = \mathbf{I}^{3\times3}$ and $t^I_\Lambda = [0\ \mathrm{m}, -10\ \mathrm{m}, 0\ \mathrm{m}]^\intercal$. The Λ's ground truth poses can be obtained with the basic rigid body transformation, as follows:

$$\begin{aligned}\phi^\Lambda_i &= \phi^I_i \\ p^\Lambda_i &= R^\Lambda_I\big(\phi^I_i t^I_\Lambda + p^I_i\big).\end{aligned}$$

The predicted positions using the inertial propagation models, which are derived by the conventional and RP methods, are shown in Fig. 3. The estimated positions using the CP measurements are approximately 0.3 m away from the ground truth positions. The error bound continues to increase. This is because, in the conventional method, relative motion dynamics must be added separately as previously described and shown in (5).

The propagation models using the conventional method are depicted as follows:

$$\begin{aligned}\phi^{\Lambda,cv}_j &= \phi^{\Lambda,cv}_i R^\Lambda_I \Delta\phi^I_{ij} R^I_\Lambda \\ v^{\Lambda,cv}_j &= v^{\Lambda,cv}_i + R^\Lambda_I \phi^I_i \big(\Delta v^I_{ij} + \underline{\big([\alpha_{ij}]_\times + [\omega_{ij}]^2_\times\big) t^I_\Lambda \Delta t_{ij}}\big) \\ p^{\Lambda,cv}_j &= p^{\Lambda,cv}_i + v^{\Lambda,cv}_i \Delta t_{ij} + R^\Lambda_I \phi^I_i \\ &\quad \times \big(\Delta p^I_{ij} + \underline{\big([\alpha_{ij}]_\times + [\omega_{ij}]^2_\times\big) t^I_\Lambda} \Delta t^2_{ij}\big)\end{aligned} \tag{5}$$

where $v^{\Lambda,cv}_j$ and $p^{\Lambda,cv}_j$ are the velocity and position propagation models using the CP method. As in (19), the separately added term, $([\alpha_{ij}]_\times + [\omega_{ij}]^2_\times) t^I_\Lambda$ represents the relative acceleration between frames i and j. α_{ij} and ω_{ij} represent the average angular acceleration and velocity between frames i and j.

The propagation models using the proposed method are described as follows:

$$\begin{aligned}\phi^{\Lambda,re}_j &= \phi^{\Lambda,\mathrm{re}}_i R^\Lambda_I \Delta\phi^I_{ij} R^I_\Lambda \\ v^{\Lambda,re}_j &= v^{\Lambda,re}_i + R^\Lambda_I \phi^I_i \Delta v^\Lambda_{ij} \\ p^{\Lambda,re}_j &= p^{\Lambda,re}_i + v^{\Lambda,re}_i \Delta t_{ij} + R^\Lambda_I \phi^I_i \Delta p^\Lambda_{ij}\end{aligned}$$

where $v^{\Lambda,re}_j$ and $p^{\Lambda,re}_j$ are the velocity and position propagation models using the RP method. As shown in (3), Δv^R_{ij} and Δp^R_{ij} represent circular motion, so the terms related with relative motion dynamics are not necessarily added.

C. Noise Propagation in the Relative Preintegration Method

In the RP method, noise is treated similar to the conventional case, but it is much more complicated to transfer the noise function from the IMU's coordinate system to the other sensor's coordinate system because the relationship between the two location is nonlinear.

The noise bound of circular motion depends on the length of a lever arm. Intuitively, when the lever arm's length grows, the noise bound is increased, and when the lever arm becomes shorter, the noise bound decreases. Hence, the lever arm term, t^I_Λ has to be added to the noise functions, as in (6). For the sake of simplicity we assume that there is no bias update in this section. Bias handling is covered in the later section

$$\begin{aligned}\Delta\phi^\Lambda_{ij} &= \Delta\bar\phi^\Lambda_{ij} \exp\big(-\delta\Delta\phi^\Lambda_{ij}\big) \\ \Delta v^\Lambda_{ij} &= \Delta\bar v^\Lambda_{ij} - \big(\delta\Delta v^I_{ij} + \delta\Delta v^R_{ij} t^I_\Lambda\big) \\ &= \Delta\bar v^\Lambda_{ij} - \delta\Delta v^\Lambda_{ij} \\ \Delta p^\Lambda_{ij} &= \Delta\bar p^\Lambda_{ij} - \big(\delta\Delta p^I_{ij} + \delta\Delta p^R_{ij} t^I_\Lambda\big) \\ &= \Delta\bar p^\Lambda_{ij} - \delta\Delta p^\Lambda_{ij}.\end{aligned} \tag{6}$$

Note that $\Delta\phi^\Lambda_{ij}$ is the same as $\Delta\phi^I_{ij}$ in the CP method, so it is not explained here. The $\bar{\cdot}$ operation indicates that the preintegration measurement has been calculated from the bias-corrected IMU signals, as shown in (10), and $\Delta\phi^\Lambda_{ij}$, Δv^Λ_{ij}, and Δp^Λ_{ij} are noise-compensated measurements. The measurement noise vector, $\delta\Delta\eta^\Lambda$ is approximated to be zero-mean normally distributed as shown in the following distribution:

$$\delta\Delta\eta^\Lambda_{ij} = \big[\big(\delta\Delta\phi^\Lambda_{ij}\big)^\intercal, \big(\delta\Delta v^\Lambda_{ij}\big)^\intercal, \big(\delta\Delta p^\Lambda_{ij}\big)^\intercal\big]^\intercal \sim \mathcal{N}\big(\mathbf{0}_{9\times1}, \Sigma^\Lambda_{ij}\big).$$

Let

$$\Upsilon_k = \bar{a}_k^{\mathrm{TAN}} + \bar{a}_k^{\mathrm{NOR}}$$
$$\Gamma_k = \left[t_\Lambda^I\right]_\times \frac{1}{\Delta t} + \left[\bar{\omega}_k^I\right]_\times \left[t_\Lambda^I\right]_\times + \left[\bar{v}_k^{\mathrm{TAN}}\right]_\times.$$

Then, the noise function of the velocity preintegration measurement can be derived as in the following equation:

$$\begin{aligned}\delta\Delta v_{ij}^\Lambda &= \delta\Delta v_{ij}^I + \delta\Delta v_{ij}^R t_\Lambda^I \\ &= -\sum_{k=i}^{j-1}\left(\Delta\bar{\phi}_{ik}^I\left(\left[\bar{a}_k^I\right]_\times + \left[\Upsilon_k\right]_\times\right)\Delta t\right)\delta\Delta\phi_{ik}^I \\ &\quad + \sum_{k=i}^{j-1}\left(\Delta\bar{\phi}_{ik}^I\,\Delta t\right)\eta^{a,I} - \sum_{k=i}^{j-1}\left(\Delta\bar{\phi}_{ik}^I\,\Gamma_k\,\Delta t\right)\eta^{g,I}. \quad (7)\end{aligned}$$

The noise function of the position preintegration measurement can be derived as follows. It is also similar to the velocity case mentioned above

$$\begin{aligned}\delta\Delta p_{ij}^\Lambda &= \delta\Delta p_{ij}^I + \delta\Delta p_{ij}^R t_\Lambda^I \\ &= \sum_{k=i}^{j-1}\delta\Delta v_{ik}^I\,\Delta t + \sum_{k=i}^{j-1}\delta\Delta v_{ik}^R t_\Lambda^I\,\Delta t \\ &\quad - \frac{1}{2}\sum_{k=i}^{j-1}\left(\Delta\bar{\phi}_{ik}^I\left(\left[\bar{a}_k^I\right]_\times + \left[\Upsilon_k\right]_\times\right)\Delta t^2\right)\delta\Delta\phi_{ik}^I \\ &\quad + \frac{1}{2}\sum_{k=i}^{j-1}\left(\Delta\bar{\phi}_{ik}^I\,\Delta t^2\right)\eta^{a,I} - \frac{1}{2}\sum_{k=i}^{j-1}\left(\Delta\bar{\phi}_{ik}^I\,\Gamma_k\,\Delta t^2\right)\eta^{g,I}.\end{aligned} \quad (8)$$

The noise covariance of the [RP] measurements is evolved through the following linear system:

$$\delta\Delta\eta_{ij}^\Lambda = \mathbf{A}_{j-1}^\Lambda\delta\Delta\eta_{ij-1}^\Lambda + \mathbf{B}_{j-1}^\Lambda\delta\eta_{j-1}^I.$$

From the above linear model, and given the covariance of the IMU noise vector, $\Sigma_{\eta^I} \in \mathbb{R}^{6\times6}$, it is now possible to compute the covariance of $\delta\Delta\eta_{ij}^\Lambda$ iteratively as follows:

$$\Sigma_{ij}^\Lambda = \mathbf{A}_{j-1}^\Lambda\Sigma_{ij-1}^\Lambda\left(\mathbf{A}_{j-1}^\Lambda\right)^\intercal + \mathbf{B}_{j-1}^\Lambda\Sigma_{\eta^I}\left(\mathbf{B}_{j-1}^\Lambda\right)^\intercal$$

where $\Sigma_{ii}^\Lambda = \mathbf{0}_{9\times9}$.

The transition functions are defined as follows:

$$\mathbf{A}_{j-1}^\Lambda = \begin{bmatrix} \left(\Delta\bar{\phi}_{j-1j}^I\right)^\intercal & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ -\Delta\bar{\phi}_{ij-1}^I\left(\left[\bar{a}_{j-1}^I\right]_\times + \left[\Upsilon_{j-1}\right]_\times\right)\Delta t & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ -\frac{1}{2}\Delta\bar{\phi}_{ij-1}^I\left(\left[\bar{a}_{j-1}^I\right]_\times + \left[\Upsilon_{j-1}\right]_\times\right)\Delta t^2 & \mathbf{I}_{3\times3}\Delta t & \mathbf{I}_{3\times3} \end{bmatrix}$$

$$\mathbf{B}_{j-1}^\Lambda = \begin{bmatrix} \mathrm{J}_\mathrm{r}\left(\bar{\omega}_{j-1}^I\Delta t\right)\Delta t & \mathbf{0}_{3\times3} \\ -\Delta\bar{\phi}_{ij-1}^I\Gamma_{j-1}\Delta t & \Delta\bar{\phi}_{ij-1}^I\Delta t \\ -\frac{1}{2}\left(\Delta\bar{\phi}_{ij-1}^I\Gamma_{j-1}\Delta t^2\right) & \frac{1}{2}\Delta\bar{\phi}_{ij-1}^I\Delta t^2 \end{bmatrix}. \quad (9)$$

where the Right Jacobian is $\mathrm{J}_\mathrm{r}(\theta) = \mathbf{I}_{3\times3} - \frac{1-\cos\|\theta\|}{\|\theta\|^2}[\theta]_\times + \frac{\|\theta\|-\sin\|\theta\|}{\|\theta\|^3}[\theta]_\times^2$.

D. Bias Handling for the Relative Preintegration Method

The bias corrected measurements are described as follows:

$$\begin{aligned}\Delta\bar{\phi}_{ij}^\Lambda &= \prod_{k=i}^{j-1}\exp\left(\left(\tilde{\omega}_k^I - b^g\right)\Delta t\right) \\ \Delta\bar{v}_{ij}^\Lambda &= \Delta\bar{v}_{ij}^I + \Delta\bar{v}_{ij}^R t_\Lambda^I \\ &= \sum_{k=i}^{j-1}\Delta\bar{\phi}_{ik}^I\left(\tilde{a}_k^I - b^a\right)\Delta t \\ &\quad + \sum_{k=i}^{j-1}\Delta\bar{\phi}_{ik}^I\left(\left[\tilde{\alpha}_k^I - \frac{b^g}{\Delta t}\right]_\times + \left[\tilde{\omega}_k^I - b^g\right]_\times^2\right)\Delta\mathrm{tt}_\Lambda^I \\ \Delta\bar{p}_{ij}^\Lambda &= \Delta\bar{p}_{ij}^I + \Delta\bar{p}_{ij}^R t_\Lambda^I \\ &= \sum_{k=i}^{j-1}\left(\Delta\bar{v}_{ik}^I\,\Delta t + \frac{1}{2}\Delta\bar{\phi}_{ik}^I\left(\tilde{a}_k^I - b^a\right)\Delta t^2\right) \\ &\quad + \sum_{k=i}^{j-1}\left(\Delta\bar{v}_{ik}^R\,\Delta t + \frac{1}{2}\Delta\bar{\phi}_{ik}^I\right. \\ &\quad \left.\times\left(\left[\tilde{\alpha}_k^I - \frac{b^g}{\Delta t}\right]_\times + \left[\tilde{\omega}_k^I - b^g\right]_\times^2\right)\Delta t^2\right)t_\Lambda^I \quad (10)\end{aligned}$$

where $\tilde{\omega}_k^I$ and $\tilde{\alpha}_k^I$ are produced by the raw IMU signals. As in the CP case, when bias is updated by adding perturbations, δb^g and δb^a to the already bias-corrected preintegration measurements with given biases b^g and b^a, the associated partial derivatives are used, as follows:

$$\begin{aligned}\Delta\bar{v}_{ij}^\Lambda\left(\delta b^g, \delta b^a\right) &\approx \Delta\bar{v}_{ij}^I + \frac{\partial\Delta\bar{v}_{ij}^I}{\partial\delta b^a}\delta b^a + \frac{\partial\Delta\bar{v}_{ij}^I}{\partial\delta b^g}\delta b^g \\ &\quad + \Delta\bar{v}_{ij}^R t_\Lambda^I + \frac{\partial\Delta\bar{v}_{ij}^R t_\Lambda^I}{\partial\delta b^g}\delta b^g \\ \Delta\bar{p}_{ij}^\Lambda\left(\delta b^g, \delta b^a\right) &\approx \Delta\bar{p}_{ij}^I + \frac{\partial\Delta\bar{p}_{ij}^I}{\partial\delta b^a}\delta b^a + \frac{\partial\Delta\bar{p}_{ij}^I}{\partial\delta b^g}\delta b^g \\ &\quad + \Delta\bar{p}_{ij}^R t_\Lambda^I + \frac{\partial\Delta\bar{p}_{ij}^R t_\Lambda^I}{\partial\delta b^g}\delta b^g.\end{aligned}$$

Equation (11) shows the partial derivatives with respect to the bias updates of the RP measurements

$$\begin{aligned}\frac{\partial\Delta\bar{v}_{ij}^R t_\Lambda^I}{\partial\delta b^g} &= -\sum_{k=i}^{j-1}\left(\Delta\bar{\phi}_{ik}^I\left([\Upsilon_k]_\times\frac{\partial\Delta\bar{\phi}_{ik}^I}{\partial\delta b^g} - \Gamma_k\right)\Delta t\right) \\ \frac{\partial\Delta\bar{p}_{ij}^R t_\Lambda^I}{\partial\delta b^g} &= \sum_{k=i}^{j-1}\left(\frac{\partial\Delta\bar{v}_{ik}^R t_\Lambda^I}{\partial\delta b^g}\Delta t\right) \\ &\quad - \frac{1}{2}\sum_{k=i}^{j-1}\left(\Delta\bar{\phi}_{ik}^I\left([\Upsilon_k]_\times\frac{\partial\Delta\bar{\phi}_{ik}^I}{\partial\delta b^g} - \Gamma_k\right)\Delta t^2\right).\end{aligned} \quad (11)$$

IV. Initialization and Extrinsic Calibration Process

This section describes the way to solve initialization and extrinsic calibration problems in various navigation systems with an IMU, using the RP method described so far. The residual functions of the nonlinear optimization frameworks of the IMU–IMU, IMU–Camera, and IMU–LiDAR navigation

system are described in this section. Later, in Appendix B, the derivations of Jacobians are also described. These Jacobians are applied to an iterative optimization process to minimize cost in general, and some off-the-shelf optimization toolkits require Jacobians of residual functions.

Since the basic framework of the initialization process of each system is identical, a generalization of the framework is explained first.

Bias updates and position estimates are also carried out during the process, thus, it is easy to expand to the localization problem from the initialization problem by adding a relocalization mechanism and key-frame management.

A. Basic Nonlinear Least Squares Framework

An optimization problem is to estimate parameters, $\mathbf{x}$s given residual functions, fs. The basic form of the nonlinear least squares problem is shown in (12). Let $\mathbf{x} \in \mathbb{R}^n$ be an n-dimensional parameter to be estimated, and f be an M-dimensional residual function. An optimization process iteratively finds a parameter at the local minima holding the following constraints:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \sum_{i=1}^{M} \| f_i(\mathbf{x}) \|^2_{\Sigma_{f_i}}. \tag{12}$$

The inertial propagation models are applied to the error functions known as factors, and basically, the following generalized error functions are applied to initialization and extrinsic calibration processes:

$$\begin{aligned} f_\phi &= \log\left(\left(\Delta\phi^I_{ij}\right)^\intercal R^I_\Lambda \left(\phi^\Lambda_i\right)^\intercal \phi^\Lambda_j R^\Lambda_I\right) \\ f_v &= \Delta v^\Lambda_{ij} - R^I_\Lambda \left(\phi^\Lambda_i\right)^\intercal \left(v^\Lambda_j - v^\Lambda_i - g^\Lambda_0 \Delta t_{ij}\right) \\ f_p &= \Delta p^\Lambda_{ij} - R^I_\Lambda \left(\phi^\Lambda_i\right)^\intercal \left(p^\Lambda_j - p^\Lambda_i - v^\Lambda_i \Delta t_{ij} - \frac{1}{2} g^\Lambda_0 \Delta t^2_{ij}\right). \end{aligned} \tag{13}$$

The RP measurements are defined as follows:

$$\begin{aligned} \Delta v^\Lambda_{ij} &= \Delta v^I_{ij} + \Delta \bar{v}^R_{ij} t^I_\Lambda + \frac{\partial \Delta \bar{v}^R_{ij} t^I_\Lambda}{\partial \delta b^{g,I}} \delta b^{g,I} \\ \Delta p^\Lambda_{ij} &= \Delta p^I_{ij} + \Delta \bar{p}^R_{ij} t^I_\Lambda + \frac{\partial \Delta \bar{p}^R_{ij} t^I_\Lambda}{\partial \delta b^{g,I}} \delta b^{g,I}. \end{aligned} \tag{14}$$

B. IMU–IMU Navigation System's Initialization and Extrinsic Calibration Process

1) Nonlinear Least Squares Framework:

$$\begin{aligned} \mathbf{x}^* &= \arg\min_{\mathbf{x}} \sum_{i=1}^{M} \left\| f^{\text{IMU}}_{i,j=i+1}(\mathbf{x}_{i,j}, \mathbf{y}_{i,j}) \right\|^2_{\Sigma_{f^{\text{IMU}}_i}} \\ \mathbf{x}_{i,j} &= \left\{ R^{I_1}_{I_2}, t^{I_1}_{I_2}, g^{I_1}_0, g^{I_2}_0, v^{I_1}_0, v^{I_2}_0, \delta\mathbf{b}, \mathbf{p}_i, \mathbf{p}_j \right\} \\ \mathbf{y}_{i,j} &= \left\{ \Delta\phi^{I_1}_{ij}, \Delta\phi^{I_2}_{ij}, \Delta v^{I_{12}}_{ij}, \Delta p^{I_{12}}_{ij}, \Delta v^{I_{21}}_{ij}, \Delta p^{I_{21}}_{ij} \right\} \\ \delta\mathbf{b}_i &= \left\{ \delta b^{g,I_1}_i, \delta b^{g,I_2}_i, \delta b^{a,I_1}_i, \delta b^{a,I_2}_i \right\} \\ \mathbf{p}_i &= \left\{ \phi^{I_1}_i, v^{I_1}_i, p^{I_1}_i, \phi^{I_2}_i, v^{I_2}_i, p^{I_2}_i \right\} \\ \mathbf{p}_j &= \left\{ \phi^{I_1}_j, v^{I_1}_j, p^{I_1}_j, \phi^{I_2}_j, v^{I_2}_j, p^{I_2}_j \right\} \end{aligned} \tag{15}$$

where the superscripts, I_1 and I_2 represent the first and second IMU, respectively. The estimating parameter, $\mathbf{x}$ consists of the rotation, $R^{I_1}_{I_2}$ and the translation, $t^{I_1}_{I_2}$ between the two IMUs, the initial gravity vectors, $g^{I_1}_0$ and $g^{I_2}_0$, and the initial velocity vectors, $v^{I_1}_0$ and $v^{I_2}_0$. The bias perturbation is $\delta\mathbf{b}$. $\mathbf{p}$ consists of the state of the system which is composed of the poses and velocities of the two IMUs. $\mathbf{y}$ is the observation vector which contains the RP measurements of the two IMUs.

2) Residual Functions: There are two IMUs in the system and there is a symmetrical set of equations; hence, $1/2$ and $12/21$ notations are used to save pages. IMU_1 and IMU_2 are abbreviated as numbers 1 and 2, respectively. The following are the substitutions. Let

$$\begin{aligned} \Psi &= \left(\phi^{I_{1/2}}_i\right)^\intercal \left(v^{I_{1/2}}_j - v^{I_{1/2}}_i - g^{I_{1/2}}_0 \Delta t_{ij}\right) \\ \Xi &= \left(\phi^{I_{1/2}}_i\right)^\intercal \left(p^{I_{1/2}}_j - p^{I_{1/2}}_i - v^{I_{1/2}}_i \Delta t_{ij} - \frac{1}{2} g^{I_{1/2}}_0 \Delta t^2_{ij}\right). \end{aligned}$$

Residual functions are described as follows:

$$\begin{aligned} f^{\text{IMU}}_{\phi_{1/2}} &= \log\left(\left(\phi^{I_{1/2}}_i \Delta\phi^{I_{1/2}}_{ij}\right)^\intercal \phi^{I_{1/2}}_j\right) \\ f^{\text{IMU}}_{\phi_{12/21}} &= \log\left(\left(\Delta\phi^{I_{1/2}}_{ij}\right)^\intercal R^{I_{1/2}}_{I_{2/1}} \Delta\phi^{I_{2/1}}_{ij} R^{I_{2/1}}_{I_{1/2}}\right) \\ f^{\text{IMU}}_{v_{1/2}} &= \Delta v^{I_{1/2}}_{ij} - \Psi \\ f^{\text{IMU}}_{v_{12/21}} &= \Delta v^{I_{12/21}}_{ij} - R^{I_{2/1}}_{I_{1/2}} \Psi \\ f^{\text{IMU}}_{p_{1/2}} &= \Delta p^{I_{1/2}}_{ij} - \Xi \\ f^{\text{IMU}}_{p_{12/21}} &= \Delta p^{I_{12/21}}_{ij} - R^{I_{1/2}}_{I_{2/1}} \Xi \end{aligned} \tag{16}$$

where $f^{\text{IMU}}_{\phi_{1/2}}$ is the difference between the orientation change produced by the two orientations of the frames i and j, and the one produced by the conventional rotation preintegration measurement. This error function is related to the independent evolution of each IMU. $f^{\text{IMU}}_{\phi_{12/21}}$ is the difference between the relative rotation preintegration measurements of the IMU_1 and IMU_2. $f^{\text{IMU}}_{v_{1/2}}$ is the difference between the velocity change produced by the two velocities of the frames i and j, and the one produced by the conventional velocity preintegration measurement. $f^{\text{IMU}}_{v_{12/21}}$ is the difference between the relative velocity preintegration measurement of one IMU and the velocity change produced by the two velocities of the frames i and j. Likewise, $f^{\text{IMU}}_{p_{1/2}}$ and $f^{\text{IMU}}_{p_{12/21}}$ are related to the differences in positions of the two IMUs.

C. IMU–Camera Navigation System's Initialization and Extrinsic Calibration Process

1) Map Initialization: As shown in [27], the best model is selected from the two computed models, a fundamental one and homography one. The model using a fundamental matrix assumes that a world is a nonplanar, and the model using homography assumes that a world is a planar. Using a heuristic method, 3-D initial map is generated. After the 3-D initial map is created, then the following bundle adjustment style tracking approach is applied to get the pose of the camera.

2) Nonlinear Least Squares Framework:

$$\begin{aligned} \mathbf{x}^* &= \arg\min_{\mathbf{x}} \sum_{i=1}^{M} \left\| f^{\text{CAM}}_{i,j=i+1}(\mathbf{x}_{i,j}, \mathbf{y}_{i,j}) \right\|^2_{\Sigma_{f^{\text{CAM}}_i}} \\ \mathbf{x}_{i,j} &= \left\{ R^I_C, t^I_C, g^C_0, v^C_0, \rho, \delta\mathbf{b}, \mathbf{p}_i, \mathbf{p}_j \right\} \end{aligned}$$

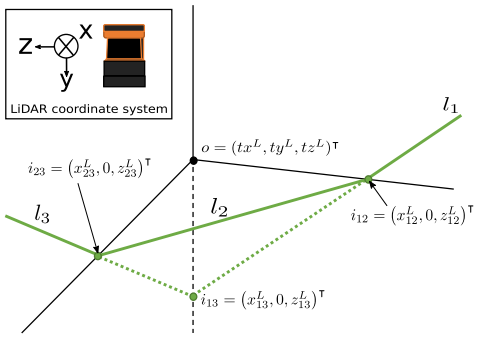

Fig. 4. LiDAR coordinates and the three orthogonal plane scanning assumption for LiDAR pose estimation.

$$\begin{aligned}
\mathbf{y}_{i,j} &= \left\{\Delta\phi^I_{ij}, \Delta v^I_{ij}, \Delta p^I_{ij}, \Delta v^C_{ij}, \Delta p^C_{ij}, \mathbf{I}_i, \mathbf{I}_j, \mathbf{L}\right\} \\
\delta\mathbf{b}_i &= \left\{\delta b^{g,I}_i, \delta b^{a,I}_i\right\}, \quad \mathbf{p}_i = \left\{\phi^C_i, v^C_i, p^C_i\right\} \\
\mathbf{p}_j &= \left\{\phi^C_j, v^C_j, p^C_j\right\}, \quad \mathbf{L} = \left\{l^1, \ldots, l^n\right\} \\
\mathbf{I}_i &= \left\{i^1_i, \ldots, i^n_i\right\}, \quad \mathbf{I}_j = \left\{i^1_j, \ldots, i^n_j\right\}
\end{aligned} \tag{17}$$

where R^I_C and t^I_C represent the extrinsic calibration parameters between the camera and the IMU. g^C_0 and v^C_0 are the initial gravity and velocity vectors. ρ is the initial scale of the landmarks. $\mathbf{p}$ consists of the camera's poses and velocities, $\mathbf{L}$ is composed of the 3-D landmark points originated by the world coordinate system, and $\mathbf{I}$ contains the projected 2-D image points of the landmarks. The superscripts I and C represent an IMU coordinate system and a camera coordinate system, respectively.

3) Residual Functions:

$$\begin{aligned}
f^{\text{CAM}}_{\phi} &= \log\left(\left(\Delta\phi^I_{ij}\right)^\intercal R^I_C \left(\phi^C_i\right)^\intercal \phi^C_j R^C_I\right) \\
f^{\text{CAM}}_{v_{\text{IC}}} &= \Delta v^C_{ij} - R^I_C \left(\phi^C_i\right)^\intercal \left(v^C_j - v^C_i - g^C_0 \Delta t_{ij}\right) \\
f^{\text{CAM}}_{p_{\text{IC}}} &= \Delta p^C_{ij} - R^I_C \left(\phi^C_i\right)^\intercal \left(\rho p^C_j - \rho p^C_i - v^C_i \Delta t_{ij} - \frac{1}{2} g^C_0 \Delta t^2_{ij}\right) \\
f^{\text{CAM}}_{pC_n} &= i^n_i - K\left(\phi^C_i\right)^\intercal \left(l^n - p^C_i\right) + i^n_j - K\left(\phi^C_j\right)^\intercal \left(l^n - p^C_j\right)
\end{aligned} \tag{18}$$

where $f^{\text{CAM}}_{pC_n}$ is a general bundle adjustment equation and K is a camera matrix. Note that $f^{\text{CAM}}_{pC_n}$ is processed independently. $f^{\text{CAM}}_{v_{\text{IC}}}$ is the difference between the relative velocity preintegration measurement of the IMU and the velocity change produced by the two velocities of the frames i and j of the camera. Likewise, $f^{\text{CAM}}_{p_{\text{IC}}}$ is the difference between the relative position preintegration measurement of the IMU and the position change produced by the two positions of the frames i and j of the camera.

D. IMU–LiDAR Navigation System's Initialization and Extrinsic Calibration Process

1) LiDAR Pose Estimation: In order to estimate the relative pose of the LiDAR, we assume that the LiDAR scans three orthogonal planes simultaneously. As shown in Fig. 4, the projected scan lines on the three planes are defined as l_1, l_2, and l_3, respectively, and the intersection points of the lines are defined as i_{12}, i_{23}, and i_{13}. Let o be the point at which the three planes meet in the current LiDAR coordinate system, then the three straight lines $\overline{oi_{12}}$, $\overline{oi_{13}}$, and $\overline{oi_{23}}$ are orthogonal to each other, based on the assumption that the three planes are orthogonal. If we define a world coordinate system with o as the origin and three vectors $\overrightarrow{oi_{12}}$, $\overrightarrow{i_{13}o}$, and $\overrightarrow{oi_{23}}$ as the x, y, and z axes, respectively, then the transformation from the world coordinate system to the current LiDAR coordinate system is

$$\begin{aligned}
R_{\text{WL}} &= [r_x, r_y, r_z], \quad T_{\text{WL}} = o \\
r_x &= \frac{i_{12} - o}{|i_{12} - o|}, \quad r_y = \frac{o - i_{13}}{|o - i_{13}|}, \quad r_z = \frac{i_{23} - o}{|i_{23} - o|}
\end{aligned}$$

where r_x, r_y, r_z, o are each a 3×1 matrix.

As shown in the LiDAR coordinate system in Fig. 4 (top left), the LiDAR's scan points exist only on the xz plane of the LiDAR coordinate system. Thus, the intersection points i_{12}, i_{23}, and i_{13} of each line can be solved by 2-D line equations, and the y value of the solution is always zero.

The position of o can be calculated using the following three spheres, which have i_{12}, i_{23}, and i_{13} as the center of each:

$$\begin{aligned}
\left(x - x^L_{12}\right)^2 + y^2 + \left(z - z^L_{12}\right)^2 &= r^2_{12} \\
\left(x - x^L_{13}\right)^2 + y^2 + \left(z - z^L_{13}\right)^2 &= r^2_{13} \\
\left(x - x^L_{23}\right)^2 + y^2 + \left(z - z^L_{23}\right)^2 &= r^2_{23}
\end{aligned}$$

where $i_{12} = (x^L_{12}, 0, z^L_{12})^T$, $i_{13} = (x^L_{13}, 0, z^L_{13})^T$, $i_{23} = (x^L_{23}, 0, z^L_{23})^T$, and r_{12}, r_{13}, and r_{23} is the distance between o and i_{12}, i_{13}, and i_{23}, respectively. r_{12}, r_{13}, and r_{23} can be obtained simply using the Pythagorean theorem

$$\begin{aligned}
r^2_{12} + r^2_{23} &= |i_{23} - i_{12}|^2 \\
r^2_{12} + r^2_{13} &= |i_{13} - i_{12}|^2 \\
r^2_{13} + r^2_{23} &= |i_{23} - i_{13}|^2.
\end{aligned}$$

The relative pose of the LiDAR can be estimated by converting the coordinate system as follows:

$$\begin{aligned}
R_{L_t L_{t-1}} &= R_{\text{WL}_{t-1}} R^{-1}_{\text{WL}_t} \\
T_{L_t L_{t-1}} &= R_{\text{WL}_{t-1}} R^{-1}_{\text{WL}_t} (-T_{\text{WL}_t}) + T_{\text{WL}_{t-1}}.
\end{aligned}$$

2) Nonlinear Least Squares Framework: The IMU–LiDAR case is similar to the IMU–Camera case, except the tracking part of the camera. Hence, f^{CAM}_{ϕ}, $f^{\text{CAM}}_{v_{\text{IC}}}$, and $f^{\text{CAM}}_{p_{\text{IC}}}$ in (18) are used as they are, and the last part, $f^{\text{CAM}}_{pC_n}$ is replaced by the LiDAR's pose estimation. Note that a LiDAR provides absolute scale poses; hence, the initial landmarks' scale factor, ρ, is removed.

V. Experimental Results

The initialization and extrinsic calibration problems for the IMU–IMU, IMU–Camera, and IMU–LiDAR navigation system were solved using the proposed RP approach. The experimental results are introduced in this section.

As shown in Fig. 5, the IMU, camera, and LiDAR were installed in the apparatus that can measure an angle and a distance to obtain the extrinsic calibration ground truth

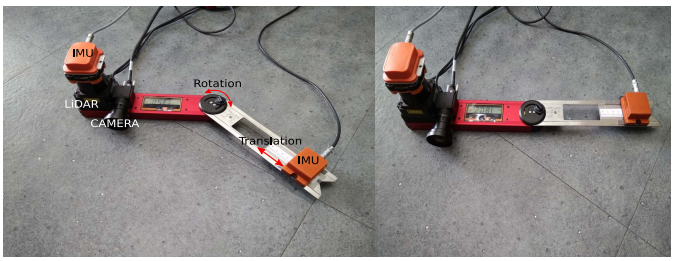

Fig. 5. Experimental setup. The device measures an angle and translation between sensors.

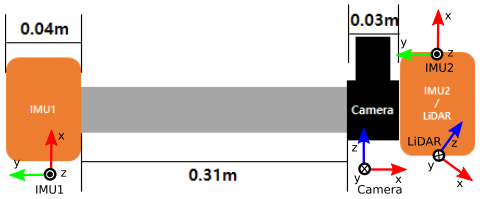

Fig. 6. Metric information of the sensor configuration. Note that the center of the sensor in each sensor is unknown.

between the sensors. The sensor configuration at 180° is shown in Fig. 6.

The IMUs were Xsens MTi-300 and MTi-G 28, and PointGrey's Flea3 camera was used. The LiDAR sensor was Hokuyo's UTM-30LX.

An Intel Core i5-3550 CPU at 3.30 GHz was used for the experiments, and the Ceres solver in [28] was used to perform a real-time optimization process.

During the initialization phase, the experimental instrument shown in Fig. 5 was moved arbitrarily by hand for 1 min, and that action was repeated three times.

The ground truths of the initial gravity and velocity vector were not obtainable because the experimental instrument was moved arbitrarily by hand; therefore, we made it stationary initially and measured an attitude by the external hand-held device. The experiments for the IMU–IMU and IMU–Camera system used the same recorded log file, so the scalar values of the initial gravity vector should be identical, and only differ in their orientations.

Since the exact axis in the sensor is not known, we compared two experiments to determine the accuracy of the extrinsic calibration parameters. The rotation between the two experiments was 45° and the translation between the two experiments was 0.05 m.

A. IMU–IMU Navigation System's Experimental Results

The estimated initial gravity and velocity are plotted in Fig. 7. They are both converged. The ground truths of the gravity and velocity were not obtainable because the experimental apparatus was moved arbitrarily by hand as mentioned earlier. A hand-held accelerometer and a gyroscope marked the gravity vector, $[1.7, 2.6, 9.3]$ at the initial position, which was similar to the initial gravity estimate. The mean value of the initial gravity obtained using the RP method was $[1.6615, 2.6402, 9.3651]$, and $[1.6857, 2.6656, 9.3496]$ using the CP method. The initial velocity vectors were plotted to be

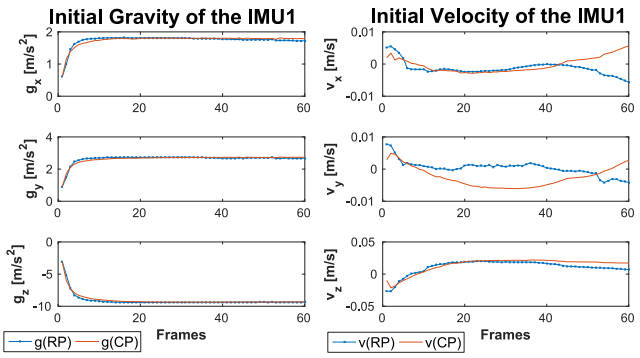

Fig. 7. Estimated initial gravity and velocity vector of the IMU–IMU system (RP and CP).

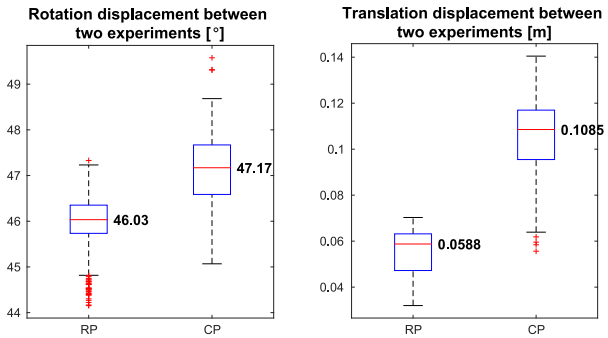

Fig. 8. Estimated extrinsic calibration parameters of the IMU–IMU system (RP and CP).

TABLE I

ERRORS IN THE ESTIMATED EXTRINSIC PARAMETERS OF THE IMU–IMU SYSTEM

Method	Translation error	Rotation error
Conventional Preintegration	$\epsilon_T = 0.0585m$	$\epsilon_R = 2.17°$
Relative Preintegration	$\epsilon_T = 0.0088m$	$\epsilon_R = 1.03°$

around $[0, 0, 0]$, but the CP method produced a more deviated initial velocity vector than the RP method did.

Fig. 8 shows the estimated extrinsic parameters between the two IMUs. As mentioned before, two experiments were executed because the exact center axis inside the sensor was unknown. The difference in rotation between the two experiments was 46.03° using the proposed method, and 47.17° using the conventional method. The difference in translation between the two experiments was 0.0588 m using the proposed method, and 0.1085 m using the conventional method. The RP method produces more accurate and consistent results than the CP method, as shown in Table I.

If only one IMU is used, the pose tends to drift in a few seconds as shown in Fig. 9, but there is no tendency to drift when using the two IMUs. Fig. 10 shows the estimated poses of the apparatus with the two IMUs. The results are shown in the video `https://youtu.be/As8NY2tsZE4`.

B. IMU–Camera Navigation System's Experimental Results

This section shows the initialization results of VINS, the most common type of navigation systems. No man-made markers are needed in this experiment.

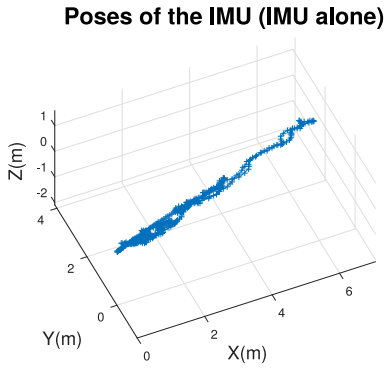

Fig. 9. Poses of the IMU have drifted in a few seconds.

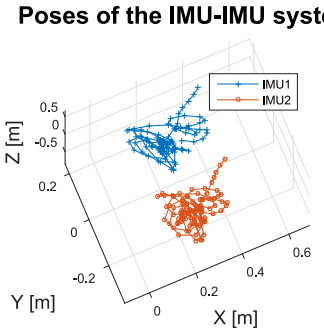

Fig. 10. Poses of the IMU–IMU system during the initialization phase (the last 100 frames).

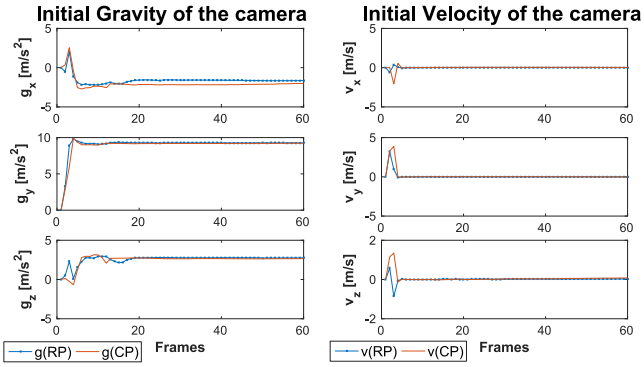

Fig. 11. Estimated initial gravity and velocity vector of the IMU–Camera system (RP and CP).

The estimated initial gravity and velocity are plotted in Fig. 11. The initial gravity vectors estimated by both methods were converged. A hand-held accelerometer marked the gravity vector, $[-1.7, 9.3, 2.6]$ at the initial position. The scalar values are the same as those in the IMU–IMU case, but differ in orientations. The initial gravity, $[-1.7122, 9.2927, 2.6216]$ was obtained using the RP method and $[-1.9678, 9.2490, 2.5983]$ using the CP method.

Note that the IMU–Camera case requires map initialization beforehand, so even when the instrument starts from a stationary state, the system is initialized while it is moving. Accordingly, the initial velocity will not exactly be zero. Precisely, the initial velocity of the RP method was $[0.0748, 0.0225, -0.0267]$ and $[0.0644, -0.0005, 0.0496]$ for the CP method.

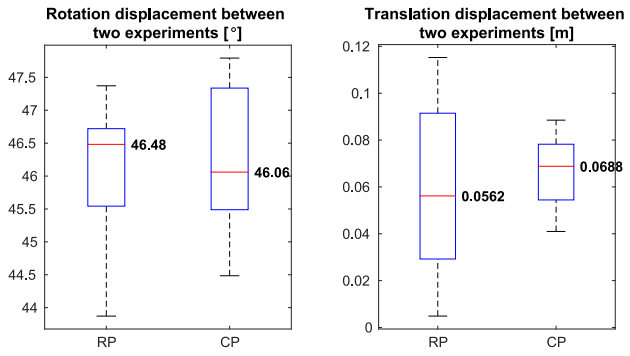

Fig. 12. Estimated extrinsic calibration parameters of the IMU–Camera system (RP and CP).

TABLE II
ERRORS IN THE ESTIMATED EXTRINSIC PARAMETERS OF THE IMU–CAMERA SYSTEM

Method	Translation error	Rotation error
Conventional Preintegration	$\epsilon_T = 0.0188m$	$\epsilon_R = 1.06°$
Kalibr	$\epsilon_T = 0.0257m$	$\epsilon_R = 3.34°$
Relative Preintegration	$\epsilon_T = 0.0062m$	$\epsilon_R = 1.48°$

Fig. 12 shows the estimated extrinsic parameters between the camera and the IMU. As mentioned before, two experiments were employed because the exact center axis inside the sensor was unknown. The difference in rotation between the two experiments was 46.48° by the proposed method, and 46.06° by the conventional method. The difference in translation between the two experiments was 0.0562 m by the proposed method, and 0.0688 m by the conventional method. In translation case, the RP produced much more accurate result than the CP did, as shown in Table II.

The rotation errors should be the same in both experiments because the rotation measurements of the CP and RP are the same as in (1). But the optimization processing time of each experiment is different, that means the processed frames are not the same for both experiments. Hence, there might be slight differences in rotation errors for both experiments which can be ignored.

The results of applying the Kalibr calibration tool in [24] are also stated. The errors produced by the Kalibr tool were larger than the errors produced by the proposed method.

As shown in Fig. 13, the estimated initial landmarks' scale of the RP method was 0.0466, which means the motion between the first two frames was 0.0466 m. The estimated scale of the CP method was 0.1967, but that did not make sense because the camera frame rate was 19.9 Hz and the visual pose tracking worked at the same rate. Hence, the speed of the hand-moved device would have to be 3.934 m/sec, which was an improbable scalar value. The RP method produced a more probable result of estimating the initial landmarks' scale.

Fig. 14 shows the estimated poses of the IMU–Camera system in the initialization stage. This result is shown in the video `https://youtu.be/2qKPDxdUDIM`, and the estimating absolute scale of the landmarks is also presented in the video by plotting the 3-D landmarks' points.

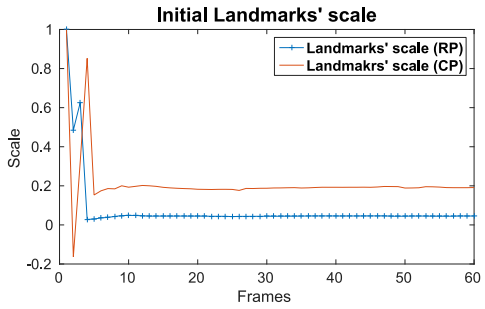

Fig. 13. Estimated scale of the landmarks.

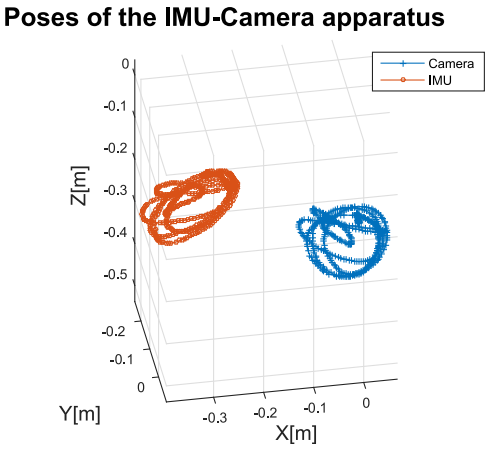

Fig. 14. Poses of the IMU–Camera system during the initialization phases (the last 500 frames).

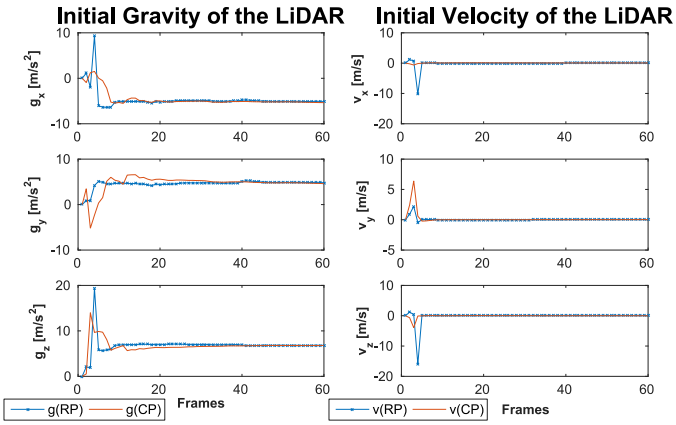

Fig. 15. Estimated initial gravity and velocity vectors of the IMU–LiDAR system (RP and CP).

C. IMU–LiDAR Navigation System's Experimental Results

The estimated initial gravity and velocity are plotted in Fig. 15. The initial gravity vectors of both the methods are converged. A hand-held accelerometer marked the gravity vector, $[-6.8, 4.5, 5.4]$ at the initial position. In order to hold three planes to get a proper LiDAR pose, the apparatus in this case should face down toward the ground. The results show the apparatus facing down initially. The initial gravity, $[-6.8741, 4.5797, 5.2490]$ was obtained using the RP method, and $[-6.4280, 4.5383, 5.8241]$ using the CP method.

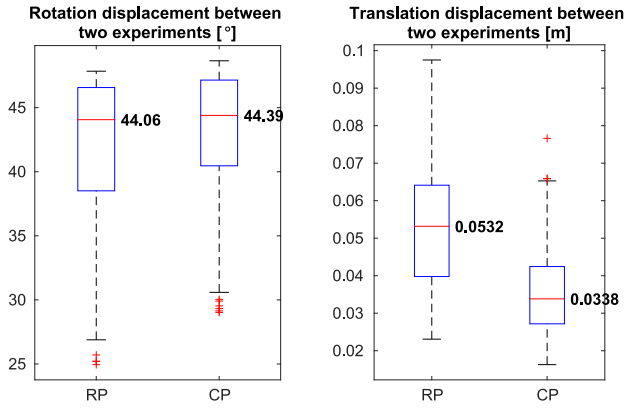

Fig. 16. Estimated extrinsic calibration parameters of the IMU–LiDAR system (RP and CP).

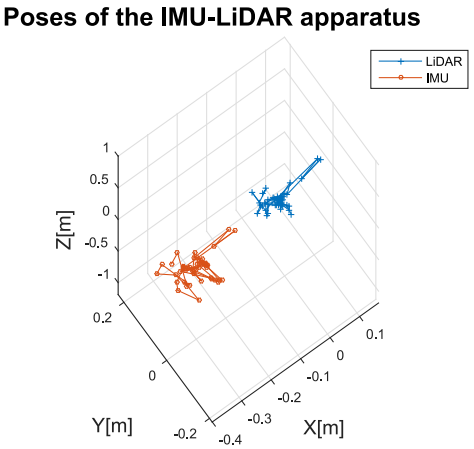

Fig. 17. Poses of the IMU–LiDAR system during the initialization phases (the last 50 frames).

Note that the IMU–LiDAR case requires five or more consistent detections of the scanned lines to filter out outliers, so that even when the instrument starts from a stationary state, the initial conditions are estimated while the apparatus is slightly moving like in the case of the IMU–Camera system. Accordingly, the initial velocity is not exactly zero. The initial velocity vectors of both the methods were converged. Precisely, the initial velocity of the RP method was $[0.0171, 0.0284, 0.0262]$ and $[0.0647, 0.0288, -0.0186]$ for the CP method.

Fig. 16 shows the estimation results of the extrinsic parameters between the LiDAR and the IMU. As mentioned before, two experiments were employed because the exact center axis inside the sensor was unknown. The difference in rotation between the two experiments was 44.06° using the proposed method, and 44.39° using the conventional method. As in the IMU–Camera case, the small difference in rotation errors of both the methods can be ignored. The difference in translation between the two experiments was 0.0532 m using the proposed method, and 0.0338 m using the conventional method. In translation case, the RP produced much more accurate result than the CP did, as shown in Table III.

Fig. 17 shows the estimated poses of the IMU–LiDAR system in the initialization stage. The results are also shown in

TABLE III
ERRORS IN THE ESTIMATED EXTRINSIC PARAMETERS OF THE IMU–LiDAR SYSTEM

Method	Translation error	Rotation error
Conventional Preintegration	$\epsilon_T = 0.0162m$	$\epsilon_R = 0.61^\circ$
Relative Preintegration	$\epsilon_T = 0.0032m$	$\epsilon_R = 0.94^\circ$

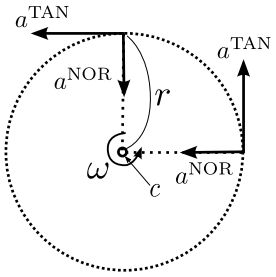

Fig. 18. Circular motion. a^{TAN} is the tangential acceleration, a^{NOR} is the normal (centripetal) acceleration, c is the center of rotation, r is the radius of the circle, and ω is the angular velocity.

the video `https://youtu.be/3slf7lGSlLE`. Note that the LiDAR processing was slower than the other cases, so only the last 50 frames were cropped. The proposed method works even if the time difference between frames is large.

VI. CONCLUSION

We proposed an relative preintegration method which includes the concept of relative motion analysis, and applied the method to the problems of initialization and extrinsic calibration in sensor fused inertial navigation systems.

The superior results in estimating the initial conditions, which are the initial gravity, initial velocity, and extrinsic calibration parameters, were shown in the various inertial navigation systems. The results also showed that the proposed method produced better results than the existing method, especially in obtaining the translation between the sensors.

In the future work, we will use the proposed method to develop solutions to general navigation problems. Because of the small size and lightweight of the MEMS IMU and camera, a Micro UAV can use the proposed method to solve its navigation problems. By exploiting its fast and accurate processing, a motion capture application which only contains multiple IMUs might be studied.

APPENDIX A
DERIVATION OF THE RELATIVE PREINTEGRATION EQUATIONS

A. Relative Motion

Relative motion dynamics are explained thoroughly in [26]. Circular motion can be explained by a tangential and normal (centripetal) acceleration, as shown in Fig. 18

$$\begin{aligned} v^{\mathrm{TAN}} &= [\omega]_\times r \\ a^{\mathrm{TAN}} &= \dot{v}^{\mathrm{TAN}} \\ a^{\mathrm{NOR}} &= [\omega]_\times v^{\mathrm{TAN}} \end{aligned}$$

where v^{TAN} is the tangential velocity, a^{TAN} is the tangential acceleration, and a^{NOR} is the normal acceleration. They are kinematics of an arbitrary location connected at the center of motion. The total acceleration at P^Λ can be obtained by adding the linear acceleration, tangential acceleration, and normal acceleration at that point, as in the following equation:

$$\begin{aligned} a^\Lambda &= a^I + a^{\mathrm{TAN}} + a^{\mathrm{NOR}} \\ &= a^I + \dot{v}^{\mathrm{TAN}} + [\omega]_\times v^{\mathrm{TAN}} \\ &= a^I + \dot{v}^{\mathrm{TAN}} + [\omega]_\times^2 t_\Lambda^I \\ &= a^I + [\alpha^I]_\times t_\Lambda^I + [\omega]_\times^2 t_\Lambda^I \end{aligned} \tag{19}$$

where α^I and ω^I represent angular acceleration and angular velocity at the IMU's coordinate system, respectively. t_Λ^I is the translation between the IMU and Λ. Once the relative accelerations are determined at a certain location, the RP measurements at that location can be obtained.

B. Derivation of a Velocity Relative Preintegration Measurement

$$\begin{aligned} \Delta v_{ij}^\Lambda &= \sum_{k=i}^{j-1} \Delta\phi_{ik}^\Lambda a_k^\Lambda \Delta t \\ &= \sum_{k=i}^{j-1} \Delta\phi_{ik}^I \left(a_k^I + a_k^{\mathrm{TAN}} + a_k^{\mathrm{NOR}}\right)\Delta t \\ &= \sum_{k=i}^{j-1} \Delta\phi_{ik}^I a_k^I \Delta t + \sum_{k=i}^{j-1} \left(\Delta\phi_{ik}^I a_k^{\mathrm{TAN}} \Delta t + \Delta\phi_{ik}^I a_k^{\mathrm{NOR}} \Delta t\right) \\ &= \sum_{k=i}^{j-1} \Delta\phi_{ik}^I a_k^I \Delta t \\ &\quad + \sum_{k=i}^{j-1} \left(\Delta\phi_{ik}^I \left[\alpha_k^I\right]_\times t_\Lambda^I \Delta t + \Delta\phi_{ik}^I \left[\omega_k^I\right]_\times^2 t_\Lambda^I \Delta t\right) \\ &= \Delta v_{ij}^I + \sum_{k=i}^{j-1} \Delta\phi_{ik}^I \left(\left(\left[\alpha_k^I\right]_\times + \left[\omega_k^I\right]_\times^2\right) t_\Lambda^I\right)\Delta t \\ &= \Delta v_{ij}^I + \sum_{k=i}^{j-1} \Delta\phi_{ik}^I \left(\left[\alpha_k^I\right]_\times + \left[\omega_k^I\right]_\times^2\right)\Delta t t_\Lambda^I \\ &= \Delta v_{ij}^I + \Delta v_{ij}^R t_\Lambda^I. \end{aligned} \tag{20}$$

Hence, the velocity RP measurement is derived as follows:

$$\Delta v_{ij}^R = \sum_{k=i}^{j-1} \Delta\phi_{ik}^I \left(\left[\alpha_k^I\right]_\times + \left[\omega_k^I\right]_\times^2\right)\Delta t. \tag{21}$$

C. Derivation of a Position Relative Preintegration Measurement

$$\begin{aligned} \Delta p_{ij}^\Lambda &= \sum_{k=i}^{j-1} \left(\Delta v_{ik}^\Lambda \Delta t + \frac{1}{2}\Delta\phi_{ik}^\Lambda a_k^\Lambda \Delta t^2\right) \\ &= \sum_{k=i}^{j-1} \Delta v_{ik}^\Lambda \Delta t + \sum_{k=i}^{j-1} \left(\frac{1}{2}\Delta\phi_{ik}^I \left(a_k^I + a_k^{\mathrm{TAN}} + a_k^{\mathrm{NOR}}\right)\Delta t^2\right) \end{aligned}$$

$$
\begin{aligned}
&= \sum_{k=i}^{j-1} \left(\Delta v_{ik}^I + \Delta v_{ik}^R t_\Lambda^I\right)\Delta t \\
&\quad + \sum_{k=i}^{j-1} \left(\left(\frac{1}{2}\Delta\phi_{ik}^I a_k^I \Delta t^2 + \frac{1}{2}\Delta\phi_{ik}^I a_k^{\mathrm{TAN}}\Delta t^2 \right.\right. \\
&\qquad \left.\left. + \frac{1}{2}\Delta\phi_{ik}^I a_k^{\mathrm{NOR}}\Delta t^2\right)\right) \\
&= \sum_{k=i}^{j-1} \left(\Delta v_{ik}^I \Delta t + \frac{1}{2}\Delta\phi_{ik}^I a_k^I \Delta t^2\right) \\
&\quad + \sum_{k=i}^{j-1} \left(\left(\Delta v_{ik}^R t_\Lambda^I \Delta t + \frac{1}{2}\Delta\phi_{ik}^I a_k^{\mathrm{TAN}}\Delta t^2 \right.\right. \\
&\qquad \left.\left. + \frac{1}{2}\Delta\phi_{ik}^I a_k^{\mathrm{NOR}}\Delta t^2\right)\right) \\
&= \Delta p_{ij}^I + \sum_{k=i}^{j-1} \left(\left(\Delta v_{ik}^R t_\Lambda^I \Delta t + \frac{1}{2}\Delta\phi_{ik}^I \left[\alpha_k^I\right]_\times t_\Lambda^I \Delta t^2 \right.\right. \\
&\qquad \left.\left. + \frac{1}{2}\Delta\phi_{ik}^I \left[\omega_k^I\right]_\times^2 t_\Lambda^I \Delta t^2\right)\right) \\
&= \Delta p_{ij}^I + \sum_{k=i}^{j-1} \left(\Delta v_{ik}^R \Delta t + \frac{1}{2}\Delta\phi_{ik}^I \left(\left[\alpha_k^I\right]_\times + \left[\omega_k^I\right]_\times^2\right)\Delta t^2\right) t_\Lambda^I \\
&= \Delta p_{ij}^I + \Delta p_{ij}^R t_\Lambda^I. \qquad (22)
\end{aligned}
$$

Then, the position RP measurement is derived as follows:

$$
\Delta p_{ij}^R = \sum_{k=i}^{j-1} \left(\Delta v_{ik}^R \Delta t + \frac{1}{2}\Delta\phi_{ik}^I \left(\left[\alpha_k^I\right]_\times + \left[\omega_k^I\right]_\times^2\right)\Delta t^2\right). \qquad (23)
$$

Appendix B
Jacobians and Noise Functions of the Residual Functions

A. Jacobians in the IMU–IMU Navigation System

Let

$$
\begin{aligned}
\xi_v &= v_j^{I_{1/2}} - v_i^{I_{1/2}} - g_0^{I_{1/2}}\Delta t_{ij} \\
\xi_p &= p_j^{I_{1/2}} - p_i^{I_{1/2}} - v_i^{I_{1/2}}\Delta t_{ij} - \frac{1}{2} g_0^{I_{1/2}}\Delta t_{ij}^2.
\end{aligned}
$$

Then, we get the following Jacobians:

$$
\begin{aligned}
&\frac{\partial f_{\phi_{12}}^{\mathrm{IMU}}}{\partial \delta R_{I_2}^{I_1}} = \mathtt{J}_\mathtt{r}{}^{-1}\left(f_{\phi_{12}}^{\mathrm{IMU}}\right)\left(\Delta\phi_{ij}^{I_2} R_{I_1}^{I_2}\right)^\mathtt{T} - \mathtt{J}_\mathtt{r}{}^{-1}\left(f_{\phi_{12}}^{\mathrm{IMU}}\right) \\
&\frac{\partial f_{\phi_{12}}^{\mathrm{IMU}}}{\partial \delta b^{g,I_1}} = -\mathtt{J}_\mathtt{r}{}^{-1}\left(f_{\phi_{12}}^{\mathrm{IMU}}\left(\delta b^{g,I_1}\right)\right)\exp \\
&\qquad \times \left(f_{\phi_{12}}^{\mathrm{IMU}}\left(\delta b^{g,I_1}\right)\right)^\mathtt{T} \mathtt{J}_\mathtt{r}\left(\frac{\partial \Delta\bar{\phi}_{ij}^{I_1}}{\partial \delta b^{g,I_1}}\delta b^{g,I_1}\right)\frac{\partial \Delta\bar{\phi}_{ij}^{I_1}}{\partial \delta b^{g,I_1}} \\
&\frac{\partial f_{\phi_{12}}^{\mathrm{IMU}}}{\partial \delta b^{g,I_2}} = \mathtt{J}_\mathtt{r}{}^{-1}\left(f_{\phi_{12}}^{\mathrm{IMU}}\left(\delta b^{g,I_2}\right)\right) R_{I_2}^{I_1} \mathtt{J}_\mathtt{r}\left(\frac{\partial \Delta\bar{\phi}_{ij}^{I_2}}{\partial \delta b^{g,I_2}}\delta b^{g,I_2}\right) \\
&\qquad \times \frac{\partial \Delta\bar{\phi}_{ij}^{I_2}}{\partial \delta b^{g,I_2}} \\
&\frac{\partial f_{\phi_{1/2}}^{\mathrm{IMU}}}{\partial \delta \phi_i^{I_{1/2}}} = -\mathtt{J}_\mathtt{r}{}^{-1}\left(f_{\phi_{1/2}}^{\mathrm{IMU}}\right)\left(\phi_j^{I_{1/2}}\right)^\mathtt{T}\phi_i^{I_{1/2}} \\
&\frac{\partial f_{\phi_{1/2}}^{\mathrm{IMU}}}{\partial \delta \phi_j^{I_{1/2}}} = \mathtt{J}_\mathtt{r}{}^{-1}\left(f_{\phi_{1/2}}^{\mathrm{IMU}}\right), \quad \frac{\partial f_{v_{1/2}}^{\mathrm{IMU}}}{\partial \delta \phi_i^{I_{1/2}}} = -\left[\left(\phi_i^{I_{1/2}}\right)^\mathtt{T}\xi_v\right]_\times \\
&\frac{\partial f_{v_{1/2}}^{\mathrm{IMU}}}{\partial \delta v_j^{I_{1/2}}} = -\left(\phi_i^{I_{1/2}}\right)^\mathtt{T}, \quad \frac{\partial f_{v_{1/2}}^{\mathrm{IMU}}}{\partial \delta v_i^{I_{1/2}}} = \left(\phi_i^{I_{1/2}}\right)^\mathtt{T} \\
&\frac{\partial f_{v_{1/2}}^{\mathrm{IMU}}}{\partial \delta g_0^{I_{1/2}}} = \left(\phi_i^{I_{1/2}}\right)^\mathtt{T}\Delta t_{ij}, \quad \frac{\partial f_{v_{1/2}}^{\mathrm{IMU}}}{\partial \delta b^{a,I_{1/2}}} = \frac{\partial \Delta\bar{v}_{ij}^{I_{1/2}}}{\partial \delta b^{a,I_{1/2}}} \\
&\frac{\partial f_{v_{1/2}}^{\mathrm{IMU}}}{\partial \delta b^{g,I_{1/2}}} = \frac{\partial \Delta\bar{v}_{ij}^{I_{1/2}}}{\partial \delta b^{g,I_{1/2}}} \\
&\frac{\partial f_{v_{12/21}}^{\mathrm{IMU}}}{\partial \delta R_{I_{2/2}}^{I_{1/1}}} = (-/+) R_{I_{1/2}}^{I_{2/1}}\left[\left(\phi_i^{I_{1/2}}\right)^\mathtt{T}\xi_v\right]_\times \\
&\frac{\partial f_{v_{12/21}}^{\mathrm{IMU}}}{\partial \delta t_{I_{2/2}}^{I_{1/1}}} = (-/+)\Delta v_{ij}^{R_{I_{2/1}}}\left(R_{I_1}^{I_2}/\cdot\right) \\
&\frac{\partial f_{v_{12/21}}^{\mathrm{IMU}}}{\partial \delta \phi_i^{I_{1/2}}} = -R_{I_{1/2}}^{I_{2/1}}\left[\left(\phi_i^{I_{1/2}}\right)^\mathtt{T}\xi_v\right]_\times \\
&\frac{\partial f_{v_{12/21}}^{\mathrm{IMU}}}{\partial \delta v_j^{I_{1/2}}} = -R_{I_{1/2}}^{I_{2/1}}\left(\phi_i^{I_{1/2}}\right)^\mathtt{T}, \quad \frac{\partial f_{v_{12/21}}^{\mathrm{IMU}}}{\partial \delta v_i^{I_{1/2}}} = R_{I_{1/2}}^{I_{2/1}}\left(\phi_i^{I_{1/2}}\right)^\mathtt{T} \\
&\frac{\partial f_{v_{12/21}}^{\mathrm{IMU}}}{\partial \delta g_0^{I_{1/2}}} = R_{I_{1/2}}^{I_{2/1}}\left(\phi_i^{I_{1/2}}\right)^\mathtt{T}\Delta t_{ij}, \quad \frac{\partial f_{v_{12/21}}^{\mathrm{IMU}}}{\partial \delta b^{a,I_{2/1}}} = \frac{\partial \Delta\bar{v}_{ij}^{I_{2/1}}}{\partial \delta b^{a,I_{2/1}}} \\
&\frac{\partial f_{v_{12/21}}^{\mathrm{IMU}}}{\partial \delta b^{g,I_{2/1}}} = \frac{\partial \Delta\bar{v}_{ij}^{I_{2/1}}}{\partial \delta b^{g,I_{2/1}}} + \frac{\partial \Delta\bar{v}_{ij}^{R_{I_{2/1}}} t_{I_{1/2}}^{I_{2/1}}}{\partial \delta b^{g,I_{2/1}}} \\
&\frac{\partial f_{p_{1/2}}^{\mathrm{IMU}}}{\partial \delta \phi_i^{I_{1/2}}} = -\left[\left(\phi_i^{I_{1/2}}\right)^\mathtt{T}\xi_p\right]_\times, \quad \frac{\partial f_{p_{1/2}}^{\mathrm{IMU}}}{\partial \delta p_j^{I_{1/2}}} = -\left(\phi_i^{I_{1/2}}\right)^\mathtt{T} \\
&\frac{\partial f_{p_{1/2}}^{\mathrm{IMU}}}{\partial \delta p_i^{I_{1/2}}} = \left(\phi_i^{I_{1/2}}\right)^\mathtt{T}, \quad \frac{\partial f_{p_{1/2}}^{\mathrm{IMU}}}{\partial \delta v_i^{I_{1/2}}} = \left(\phi_i^{I_{1/2}}\right)^\mathtt{T}\Delta t_{ij} \\
&\frac{\partial f_{p_{1/2}}^{\mathrm{IMU}}}{\partial \delta g_0^{I_{1/2}}} = \frac{1}{2}\left(\phi_i^{I_{1/2}}\right)^\mathtt{T}\Delta t_{ij}^2, \quad \frac{\partial f_{p_{1/2}}^{\mathrm{IMU}}}{\partial \delta b^{g,I_{1/2}}} = \frac{\partial \Delta\bar{p}_{ij}^{I_{1/2}}}{\partial \delta b^{g,I_{1/2}}} \\
&\frac{\partial f_{p_{1/2}}^{\mathrm{IMU}}}{\partial \delta b^{a,I_{1/2}}} = \frac{\partial \Delta\bar{p}_{ij}^{I_{1/2}}}{\partial \delta b^{a,I_{1/2}}} \\
&\frac{\partial f_{p_{12/21}}^{\mathrm{IMU}}}{\partial \delta R_{I_{2/2}}^{I_{1/1}}} = (-/+) R_{I_{1/2}}^{I_{2/1}}\left[\left(\phi_i^{I_{1/2}}\right)^\mathtt{T}\xi_p\right]_\times \\
&\frac{\partial f_{p_{12/21}}^{\mathrm{IMU}}}{\partial \delta t_{I_{2/2}}^{I_{1/1}}} = (-/+)\Delta p_{ij}^{R_{I_{2/1}}}\left(R_{I_1}^{I_2}/\cdot\right) \\
&\frac{\partial f_{p_{12/21}}^{\mathrm{IMU}}}{\partial \delta \phi_i^{I_{1/2}}} = R_{I_{1/2}}^{I_{2/1}}\left(\phi_i^{I_{1/2}}\right)^\mathtt{T}[\xi_p]_\times \\
&\frac{\partial f_{p_{12/21}}^{\mathrm{IMU}}}{\partial \delta p_j^{I_{1/2}}} = -R_{I_{1/2}}^{I_{2/1}}\left(\phi_i^{I_{1/2}}\right)^\mathtt{T}, \quad \frac{\partial f_{p_{12/21}}^{\mathrm{IMU}}}{\partial \delta p_i^{I_{1/2}}} = R_{I_{1/2}}^{I_{2/1}}\left(\phi_i^{I_{1/2}}\right)^\mathtt{T} \\
&\frac{\partial f_{p_{12/21}}^{\mathrm{IMU}}}{\partial \delta v_i^{I_{1/2}}} = R_{I_{1/2}}^{I_{2/1}}\left(\phi_i^{I_{1/2}}\right)^\mathtt{T}\Delta t_{ij} \\
&\frac{\partial f_{p_{12/21}}^{\mathrm{IMU}}}{\partial \delta g_0^{I_{1/2}}} = \frac{1}{2} R_{I_{1/2}}^{I_{2/1}}\left(\phi_i^{I_{1/2}}\right)^\mathtt{T}\Delta t_{ij}^2
\end{aligned}
$$

$$\frac{\partial f^{\text{IMU}}_{p_{12/21}}}{\partial \delta b^{a,I_{2/1}}} = \frac{\partial \Delta \bar{p}^{I_{2/1}}_{ij}}{\partial \delta b^{a,I_{2/1}}}$$
$$\frac{\partial f^{\text{IMU}}_{p_{12/21}}}{\partial \delta b^{g,I_{2/1}}} = \frac{\partial \Delta \bar{p}^{I_{2/1}}_{ij}}{\partial \delta b^{g,I_{2/1}}} + \frac{\partial \Delta \bar{p}^{R_{I_{2/1}}}_{ij} t^{I_{2/1}}_{I_{1/2}}}{\partial \delta b^{g,I_{2/1}}}$$

where the inverse of the Right Jacobian is $\mathrm{J}_r{}^{-1}(\theta) = \mathbf{I}_{3\times 3} + \frac{1}{2}[\theta]_\times + (\frac{1}{\|\theta\|^2} - \frac{1+\cos\|\theta\|}{2\|\theta\|\sin\|\theta\|})[\theta]^2_\times$ introduced in [29].

B. Noise Functions in the IMU–IMU Navigation System

$$\begin{aligned}
\delta f^{\text{IMU}}_{\phi_{1/2}} &= -\mathrm{J}_r\big(f^{\text{IMU}}_{\phi_{1/2}}\big)\exp\big(f^{\text{IMU}}_{\phi_{1/2}}\big)^\intercal \delta\Delta\phi^{I_{1/2}}_{ij}\\
\delta f^{\text{IMU}}_{\phi_{12}} &= -\mathrm{J}_r\big(f^{\text{IMU}}_{\phi_{12}}\big)\exp\big(f^{\text{IMU}}_{\phi_{12}}\big)^\intercal \delta\Delta\phi^{I_1}_{ij}\\
&\quad + \mathrm{J}_r\big(f^{\text{IMU}}_{\phi_{12/21}}\big) R^{I_{1/2}}_{I_{2/1}} \delta\Delta\phi^{I_{2/1}}_{ij}\\
\delta f^{\text{IMU}}_{v_{1/2}} &= \delta\Delta v^{I_{1/2}}_{ij}, \quad \delta f^{\text{IMU}}_{v_{12/21}} = \delta\Delta v^{I_{12/21}}_{ij}\\
\delta f^{\text{IMU}}_{p_{1/2}} &= \delta\Delta p^{I_{1/2}}_{ij}, \quad \delta f^{\text{IMU}}_{p_{12/21}} = \delta\Delta p^{I_{12/21}}_{ij}.
\end{aligned}$$

C. Jacobians in the IMU–Camera Navigation System

Following equations are Jacobians of residual functions of IMU–Camera initialization. Let

$$\begin{aligned}
\xi_v &= \big(v^C_j - v^C_i - g^C_0 \Delta t_{ij}\big)\\
\xi_p &= \left(\rho p^C_j - \rho p^C_i - v^C_i \Delta t_{ij} - \frac{1}{2} g^C_0 \Delta t^2_{ij}\right).
\end{aligned}$$

Then, we get the following Jacobians:

$$\begin{aligned}
\frac{\partial f^{\text{CAM}}_\phi}{\partial \delta R^I_C} &= \mathrm{J}_r{}^{-1}\big(f^{\text{CAM}}_\phi\big)\big(\big(\phi^C_i\big)^\intercal \phi^C_j R^C_I\big)^\intercal - \mathrm{J}_r{}^{-1}\big(f^{\text{CAM}}_\phi\big)\\
\frac{\partial f^{\text{CAM}}_\phi}{\partial \delta \phi^C_i} &= -\mathrm{J}_r{}^{-1}\big(f^{\text{CAM}}_\phi\big)\big(\big(\phi^C_i\big)^\intercal \phi^C_j R^C_I\big)^\intercal\\
\frac{\partial f^{\text{CAM}}_\phi}{\partial \delta \phi^C_j} &= \mathrm{J}_r{}^{-1}\big(f^{\text{CAM}}_\phi\big) R^I_C\\
\frac{\partial f^{\text{CAM}}_\phi}{\partial \delta b^{g,I}} &= -\mathrm{J}_r{}^{-1}\left(f^{\text{CAM}}_\phi\big(\delta b^{g,I}\big)\right)\exp\left(f^{\text{CAM}}_\phi\big(\delta b^{g,I}\big)\right)^\intercal \mathrm{J}_r\\
&\quad \times\left(\frac{\partial \Delta\bar{\phi}^I_{ij}}{\partial \delta b^{g,I}}\delta b^{g,I}\right)\frac{\partial \Delta\bar{\phi}^I_{ij}}{\partial \delta b^{g,I}}\\
\frac{\partial f^{\text{CAM}}_{v_{\text{IC}}}}{\partial \delta R^I_C} &= -R^C_I\big[\big(\phi^C_i\big)^\intercal \xi_v\big]_\times, \quad \frac{\partial f^{\text{CAM}}_{v_{\text{IC}}}}{\partial \delta t^I_C} = \Delta v^R_{ij}\\
\frac{\partial f^{\text{CAM}}_{v_{\text{IC}}}}{\partial \delta \phi^C_i} &= -R^I_C\big[\big(\phi^C_i\big)^\intercal \xi_v\big]_\times, \quad \frac{\partial f^{\text{CAM}}_{v_{\text{IC}}}}{\partial \delta v^C_j} = -R^I_C\big(\phi^C_i\big)^\intercal\\
\frac{\partial f^{\text{CAM}}_{v_{\text{IC}}}}{\partial \delta v^C_i} &= R^I_C\big(\phi^C_i\big)^\intercal, \quad \frac{\partial f^{\text{CAM}}_{v_{\text{IC}}}}{\partial \delta g^C_0} = R^I_C\big(\phi^C_i\big)^\intercal \Delta t_{ij}\\
\frac{\partial f^{\text{CAM}}_{v_{\text{IC}}}}{\partial \delta b^{a,I}} &= \frac{\partial \Delta\bar{v}^I_{ij}}{\partial \delta b^{a,I}}, \quad \frac{\partial f^{\text{CAM}}_{v_{\text{IC}}}}{\partial \delta b^{g,I}} = \frac{\partial \Delta\bar{v}^I_{ij}}{\partial \delta b^{g,I}} + \frac{\partial \Delta\bar{v}^R_{ij} t^I_C}{\partial \delta b^{g,I}}\\
\frac{\partial f^{\text{CAM}}_{p_{\text{IC}}}}{\partial \delta R^I_C} &= -R^C_I\big[\big(\phi^C_i\big)^\intercal \xi_p\big]_\times, \quad \frac{\partial f^{\text{CAM}}_{p_{\text{IC}}}}{\partial \delta t^I_C} = \Delta p^R_{ij}\\
\frac{\partial f^{\text{CAM}}_{p_{\text{IC}}}}{\partial \delta \phi^C_i} &= -R^C_I\big[\big(\phi^C_i\big)^\intercal \xi_p\big]_\times, \quad \frac{\partial f^{\text{CAM}}_{p_{\text{IC}}}}{\partial \delta p^C_j} = -R^I_C\big(\phi^C_i\big)^\intercal\\
\frac{\partial f^{\text{CAM}}_{p_{\text{IC}}}}{\partial \delta p^C_i} &= R^I_C\big(\phi^C_i\big)^\intercal, \quad \frac{\partial f^{\text{CAM}}_{p_{\text{IC}}}}{\partial \delta v^C_i} = R^I_C\big(\phi^C_i\big)^\intercal \Delta t_{ij}\\
\frac{\partial f^{\text{CAM}}_{p_{\text{IC}}}}{\partial \delta g^C_0} &= \frac{1}{2} R^I_C\big(\phi^C_i\big)^\intercal \Delta t^2_{ij}, \quad \frac{\partial f^{\text{CAM}}_{p_{\text{IC}}}}{\partial \delta b^{a,I}} = \frac{\partial \Delta\bar{p}^I_{ij}}{\partial \delta b^{a,I}}\\
\frac{\partial f^{\text{CAM}}_{p_{\text{IC}}}}{\partial \delta b^{g,I}} &= \frac{\partial \Delta\bar{p}^I_{ij}}{\partial \delta b^{g,I}} + \frac{\partial \Delta\bar{p}^R_{ij} t^I_C}{\partial \delta b^{g,I}}\\
\frac{\partial f^{\text{CAM}}_{p_{\text{IC}}}}{\partial \delta \rho} &= -R^I_C\big(\phi^C_i\big)^\intercal\big(p^C_j - p^C_i\big).
\end{aligned}$$

D. Noise Functions in the IMU–Camera Navigation System

$$\begin{aligned}
\delta f^{\text{CAM}}_\phi &= -\mathrm{J}_r\big(f^{\text{CAM}}_\phi\big)\exp\big(f^{\text{CAM}}_\phi\big)^\intercal \delta\Delta\phi^I_{ij}\\
\delta f^{\text{CAM}}_{v_{\text{IC}}} &= \delta\Delta v^C_{ij}, \quad \delta f^{\text{CAM}}_{p_{\text{IC}}} = \delta\Delta p^C_{ij}.
\end{aligned}$$

REFERENCES

[1] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation," in *Proc. Robot., Sci. Syst.*, Rome, Italy, Jul. 2015, pp. 1–10.

[2] A. I. Mourikis, N. Trawny, S. I. Roumeliotis, A. E. Johnson, A. Ansar, and L. Matthies, "Vision-aided inertial navigation for spacecraft entry, descent, and landing," *IEEE Trans. Robot.*, vol. 25, no. 2, pp. 264–280, Apr. 2009.

[3] J.-P. Tardif, M. George, M. Laverne, A. Kelly, and A. Stentz, "A new approach to vision-aided inertial navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 4161–4168.

[4] G. Huang, M. Kaess, and J. J. Leonard, "Towards consistent visual-inertial navigation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May/Jun. 2014, pp. 4926–4933.

[5] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Camera-IMU-based localization: Observability analysis and consistency improvement," *Int. J. Robot. Res.*, vol. 33, no. 1, pp. 182–201, Jan. 2014.

[6] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 314–334, Mar. 2015.

[7] A. Concha, G. Loianno, V. Kumar, and J. Civera, "Visual-inertial direct SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 1331–1338.

[8] V. Usenko, J. Engel, J. Stückler, and D. Cremers, "Direct visual-inertial odometry with stereo cameras," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 1885–1892.

[9] W. He, Y. Ouyang, and J. Hong, "Vibration control of a flexible robotic manipulator in the presence of input deadzone," *IEEE Trans. Ind. Informat.*, vol. 13, no. 1, pp. 48–59, Feb. 2017.

[10] Z. Li, Z. Huang, W. He, and C. Y. Su, "Adaptive impedance control for an upper limb robotic exoskeleton using biological signals," *IEEE Trans. Ind. Electron.*, vol. 64, no. 2, pp. 1664–1674, Feb. 2017.

[11] W. He, Y. Chen, and Z. Yin, "Adaptive neural network control of an uncertain robot with full-state constraints," *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 620–629, Mar. 2016.

[12] W. He, Y. Dong, and C. Sun, "Adaptive neural impedance control of a robotic manipulator with input saturation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 3, pp. 334–344, Mar. 2016.

[13] T. Lupton and S. Sukkarieh, "Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions," *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 61–76, Feb. 2012.

[14] V. Indelman, S. Williams, M. Kaess, and F. Dellaert, "Information fusion in navigation systems via factor graph based incremental smoothing," *Robot. Auto. Syst.*, vol. 61, no. 8, pp. 721–738, 2013.

[15] W. Xu and D. Choi, "Direct visual-inertial odometry and mapping for unmanned vehicle," in *Proc. 12th Int. Symp. Adv. Vis. Comput. (ISVC)*, Las Vegas, NV, USA, Dec. 2016, pp. 595–604.

[16] K. Eckenhoff, P. Geneva, and G. Huang, "High-accuracy preintegration for visual-inertial navigation," in *Proc. Int. Workshop Algorithmic Found. Robot.*, San Francisco, CA, USA, Dec. 2016, pp. 1–16.

[17] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 1–21, Feb. 2017.

[18] Z. Yang and S. Shen, "Monocular visual–inertial state estimation with online initialization and camera–IMU extrinsic calibration," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 39–51, Jan. 2017.
[19] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3D robotics hand/eye calibration," *IEEE Trans. Robot. Autom.*, vol. 5, no. 3, pp. 345–358, Jun. 1989.
[20] J. Lobo and J. Dias, "Relative pose calibration between visual and inertial sensors," *Int. J. Robot. Res.*, vol. 26, no. 6, pp. 561–575, 2007.
[21] F. M. Mirzaei and S. I. Roumeliotis, "A Kalman filter-based algorithm for IMU-camera calibration: Observability analysis and performance evaluation," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1143–1156, Oct. 2008.
[22] J. Kelly and G. S. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *Int. J. Robot. Res.*, vol. 30, no. 1, pp. 56–79, 2011.
[23] M. Fleps, E. Mair, O. Ruepp, M. Suppa, and D. Burschka, "Optimization based IMU camera calibration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2011, pp. 3297–3304.
[24] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2013, pp. 1280–1286.
[25] T. C. Dong-Si and A. I. Mourikis, "Estimator initialization in vision-aided inertial navigation with unknown camera-IMU calibration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2012, pp. 1064–1071.
[26] R. Hibbeler, *Engineering Mechanics: Dynamics* (Mastering Engineering Series). Englewood Cliffs, NJ, USA: Prentice-Hall, 2010, ch. 16.7.
[27] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.
[28] S. Agarwal *et al.*, *Ceres Solver*. [Online]. Available: http://ceres-solver.org
[29] G. Chirikjian, *Stochastic Models, Information Theory, and Lie Groups: Analytic Methods and Modern Applications* (Applied and Numerical Harmonic Analysis), vol. 2. Boston, MA, USA: Birkhäuser 2011.

Dongshin Kim received the B.S. degree in computer science from Iowa State University, Ames, IA, USA, in 2003, and the M.S. degree in computer science from the Georgia Institute of Technology, Atlanta, GA, USA, in 2005. He is currently pursuing the Ph.D. degree in electrical engineering with the Korea Advanced Institute of Science and Technology, Daejeon, South Korea.

He is currently at LG electronics, Seoul, South Korea, developing autonomous driving system. His research interests include computer vision, SLAM framework, visual inertial navigation, and robot application.

Seunghak Shin (S'16) received the B.S. degree in electrical engineering from Yonsei University, Seoul, South Korea, in 2009, and the M.S. degree in robotics program from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2011, where he is currently pursuing the Ph.D. degree in electrical engineering.

His research interests include geometric vision, sensor fusion, object detection, and robot application.

Mr. Shin was a member of "Team KAIST," which won the First Place in the DARPA Robotics Challenge Finals 2015.

In So Kweon (M'93) received the B.S. and M.S. degrees in mechanical design and production engineering from Seoul National University, Seoul, South Korea, in 1981 and 1983, respectively, and the Ph.D. degree in robotics from the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, in 1990.

He has been with the Toshiba Research and Development Center, Kawasaki, Japan, since 1992, where he is currently a KEPCO Chair Professor with the Department of Electrical Engineering.

Dr. Kweon was a member of "Team KAIST," which won the First Place in the DARPA Robotics Challenge Finals 2015. He is a member of the KROS. He served as a Program Co-Chair of ACCV'07 and a General Chair of ACCV'12. He is on the Honorary Board of the *International Journal of Computer Vision*.