Received: 6 December 2022 | Revised: 25 March 2023 | Accepted: 17 May 2023

DOI: 10.1002/rob.22212

RESEARCH ARTICLE

WILEY

Inertial preintegration for VI-SLAM by the screw motion theory

Nassim Bessaad[1] | Bao Qilian[1] | Zhao Jiankang[1] | Nardjess Benoudina[2] | Shuodong Sun[3] | Xuwei Zhang[3]

[1]Department of Instrument Science and Engineering, Shanghai Jiao Tong University, Shanghai, China

[2]Department of Mathematics and Computer Science, Zhejiang Normal University, Jinhua, China

[3]Shanghai Aerospace Control Technology Institute, Shanghai, China

Correspondence
Nassim Bessaad, Department of Instrument Science and Engineering, Shanghai Jiao Tong University, Shanghai 20240, China.
Email: bessaad_nassim@sjtu.edu.cn

Abstract

Most smoothing-based visual-inertial simultaneous localization and mapping algorithms (VI-SLAM) rely on the Lie algebra processing of the inertial measurements. This approach is limited in its decoupled update of the attitude by using SO_3 and velocity increments by SE_3. In addition to limitations on only point transformation between frames. We present a novel approach to handling inertial measurement unit (IMU) measurements between two camera frames by the screw motion theory. Where rigid body dynamics are concisely represented by the compact unit dual quaternion. With this approach, the limitations of point transformation are mitigated by the superior Plücker line transformation and the states update is achieved by a single coupled operation. To harness this consistent framework for a smoothing-based VI-SLAM, the screw motion twist parameter is based on the raw IMU measurements. Then, a consistent residual cost function with the corresponding Jacobian and covariance updates is derived for graph-optimization algorithm respecting the screw motion paradigm. A transition method is proposed to overcome the issues of over-parametrization by the unit dual quaternion. solving all singularity threats while saving the advantages of adopting the twist operator. Finally, the loftier performance of the proposed algorithms is attested by simulation and real-world experiments.

KEYWORDS

dual quaternions, IMU, preintegration, screw motion, VI-SLAM

1 | INTRODUCTION

In recent years, Simultaneous Localization and Mapping (SLAM) has received considerable attention from researchers and developers. The applications have expanded to most technological and industrial fields because of their low cost and high robustness (Huang, Thrun, et al., 2006). Visual-Inertial SLAM (VI-SLAM) is the most reliable and efficient method for performing an automated task such as exploration and/or navigation. This integrated navigation system consists of two main sensors and a powerful computer to achieve the best performance (Cadena et al., 2016; Huang & Dissanayake, 2016). Cameras and inertial measurement units (IMUs) have been best partners for their easy integration and compact nature (Delmerico & Scaramuzza, 2018). Because the IMU is a high-frequency standalone sensor, it can bridge scenarios where cameras are ineffective (Bessaad et al., 2021). On the other hand, its deteriorating performance is improved by the superiority of the camera's estimation of homography (F-E matrices) and the camera pose (Triggs et al., 2000). This superiority is due to the maturity of the computer vision algorithms and the accumulation of errors in the inertial algorithms. However, the traditional integration methods employ the Kalman filter or one of its descendants, which

suffers from lower accuracy in comparison to the smoothing-based integration counterpart (Delmerico & Scaramuzza, 2018; Huang, 2019; Thrun et al., 2006).

A graph-based SLAM rectifies the weakness of the filtering-based algorithm by the re-linearization of the trajectory's keyframe poses (Grisetti et al., 2010). Hence, it keeps track of all the vehicle's past states and measurements, thus consumes more memory and CPU time. Nonetheless, modern algorithms reduce the frame count to only those keyframes that have significant data while the marginalization routine harnesses all the important information (Kaess, 2008). Also, the IMU measurements are preintegrated to form a single update by the rotation and the translation increment (Forster et al., 2015). As mentioned earlier, there is a lack in the preintegration theory of the inertial aspect of the graph-based VI-SLAM compared with the more mature visual algorithms. Only recently has a continuous form of the preintegration based on Lie algebra emerged along with an analytic form (Eckenhoff et al., 2018). This is considered a step in the right direction but still insufficient to reach the level of computer vision algorithms. To that end, we present in this paper a novel approach to combine the raw IMU measurements and perform preintegration in the screw motion framework. Utilizing the unit dual quaternion (UDQ) as the pose representation for edges between keyframes. Building the corresponding optimization problem with the right cost function, weights update, Jacobians, and bias Jacobians in the Maximum A Posteriori (MAP) estimation sense for the graph-based SLAM. We derive all the necessary parameters for the graph-solver whilst making a smooth transition from the screw motion paradigm to the minimum representation of states. Achieving a novel method to propagate the measurements in an overly-parametrized representation and then extract the covariances and Jacobians in a minimal representation for the solver while keeping the screw motion consistency.

After reviewing recent related literature in Section 2, we summarize the underlying theory behind the screw motion and the UDQ representation of the twist parameter. Followed by the derivations of the IMU preintegration and its Jacobians necessary for the graph-SLAM algorithm in Section 3. Then in Section 4, we introduce the optimization in the screw motion framework. A residual cost function and its variant are defined alongside the corresponding Jacobians. We benchmark the proposed algorithm against the state-of-the-art preintegration on-manifold by simulations in Section 5. We also test the algorithm in a real-world scenario by means of publicly available data sets. We then conclude the paper with our intended future direction in Section 6.

2 | RELATED WORKS

Lupton in Lupton and Sukkarieh (2011) have proposed summing all the IMU data between two consecutive frames into a single pose increment to achieve computational tractability. Later work improved on it by introducing the Lie algebra formulation of preintegration, which overcomes Euler angle singularities (Forster et al., 2016). Since then, the IMU preintegration theory has been dominated by this method, adopting the direction cosine matrix and quaternion representation for attitude and a three-dimensional (3D) vector for the velocity (Leutenegger et al., 2013). Understandably so for its suitable minimum representation that can be modeled in the state space, avoiding singularities in the Hessian and the covariance inversion. Later on, continuous and closed-form analytical solutions for the IMU measurement propagation were presented (Eckenhoff et al., 2019, 2020; Le Gentil & Vidal-Calleja, 2021). Contributing to the completion of the theory of the version based on Lie algebra. This formulation, however, is still incoherent and ignores motion disturbances and irregularities such as coning and sculling. Such batch-based VI-SLAM algorithms are nonetheless successful and have been introduced to the industry with great impact. One of the first implementations is the ORB-SLAM2 (Mur-Artal & Tardos, 2017). This VI-SLAM algorithm made use of Forster's work and successfully provided an additional boost to the research. It represents the poses with a homogenous matrix (direction cosine matrix for attitude and a 3D vector for position). The algorithm was further improved, especially for the map and IMU-camera initialization, by an optimized MAP initialization routine that accounts for IMU parameters (Campos et al., 2019). However, this formulation does not account for commutativity errors (Bessaad et al., 2018), still consumes memory and CPU time to convert to minimal representation, and it is found to be lacking in consistency. Another version of the preintegration on manifold is by the quaternion update of the attitude and its optimization by only the 3D vector part of the orientation error (Qin et al., 2018). This method was proven faster and more robust, but it still updates the pose by the direction cosine matrix in a decoupled manner. Aside from these works, there are several successful VI-SLAM algorithms that use the same preintegration scheme. Please refer to (Delmerico & Scaramuzza, 2018; Huang, 2019) for more details. All the above-mentioned methods are limited to only point transformations. However, with screw motion theory, we can achieve line transformation between frames (Adorno, 2017). This property can boost the research on scene reconstruction by point clouds to a shape-based scene understanding by Plücker lines (Zhao et al., 2022). Figure 1 shows a comparison between the $SO_3 + SE_3$ point-only transformation and the proposed method.

In contrast to the Lie algebra formulation by $SO_3 + SE_3$, employing the screw motion theory for graph-SLAM has not been investigated thoroughly (Kim et al., 2015). In fact, the problem of over-parametrization and the induced singularity have discouraged researchers from pursuing such a direction. On the other hand, the IMU update by UDQ has been successfully implemented for simple integrated navigation systems (Yuanxin Wu et al., 2005). Additionally, the screw motion theory has been introduced to the robotics and automation fields with great impact (Gouasmi, 2012). Where robot arms and joints are modeled and controlled by the UDQ framework. More details on formulating the kinematics of a planar parallel robot have been published in Adorno (2017), Balmaceda-Santamaría and García-Murillo (2019). Our work aligns with the general direction of the robotics and SLAM communities to enrich the theory and improve the performance by such notions. We introduce the full graph-based VI-SLAM inertial aspect of the algorithm, that is, the

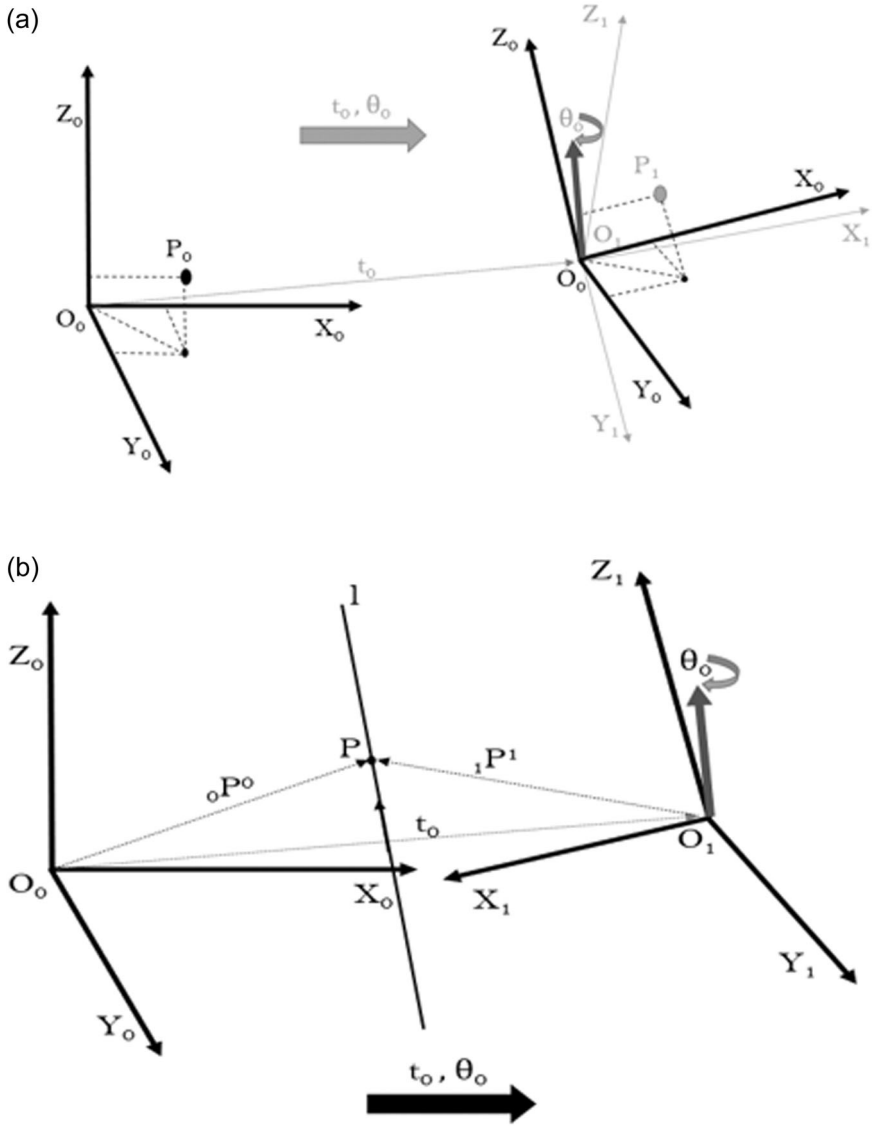

FIGURE 1 $SO_3 + SE_3$ versus screw motion for rigid body transformation. (a) Point transformation by $SO_3 + SE_3$, (b) Plücker line transformation by screw motion theory.

preintegration and the optimization based on the screw motion theory.

In the traditional Lie algebra formulation Figure 1a, each point P_0 in the frame (O_0, X_0, Y_0, Z_0) endure a translation then rotation to the new frame (O_1, X_1, Y_1, Z_1). In contrast, as shown in Figure 1b, a point P_0 in the frame (O_0, X_0, Y_0, Z_0) is part of a set of points defined by the Plücker line **l**. where the entirety of line is transformed to the frame (O_1, X_1, Y_1, Z_1).

3 | METHODOLOGY

3.1 | Basic theory on screw motion and unit dual quaternion

The motion kinematics of a rigid body in free space is efficiently represented by a rotation about an axis and a linear displacement relative to that axis. This motion is analogous to the motion of a screw moving on a solid plane before it finds equilibrium (Yuanxin Wu et al., 2005). Thus, the name "screw motion" originated. The screw motion axis and pitch are sufficient to model any motion in free space. And to represent such behavior, the UDQ has been found so far to be the most compact and efficient method. This operator must be of unit magnitude and inherit the characteristics of quaternions (Gouasmi, 2012), it actually consists of two quaternions: $\hat{\mathbf{q}} = (\mathbf{q}, \mathbf{q}')$.

$$\begin{aligned} \mathbf{q} &= (s, \mathbf{v}), \\ \hat{\mathbf{q}} &= \left[\mathbf{q} + \frac{\varepsilon}{2}(\mathbf{t} \otimes \mathbf{q})\right], \end{aligned} \tag{1}$$

where s is the scalar part of a unit quaternion $\mathbf{q}$, and $\mathbf{v}$ is the vector part. ε is the nilpotent parameter with the following properties: $\varepsilon \neq 0$, $\varepsilon^2 = 0$. The UDQ $\hat{\mathbf{q}}$ is composed of two parts as mentioned before. The rotation is encoded by unit quaternion $\mathbf{q}$ and the translation $\mathbf{t}$ is embedded within $\mathbf{q}'$ where $\mathbf{q}' = \mathbf{t} \otimes \mathbf{q}$. The special operator $\otimes$ is the quaternion product operator.

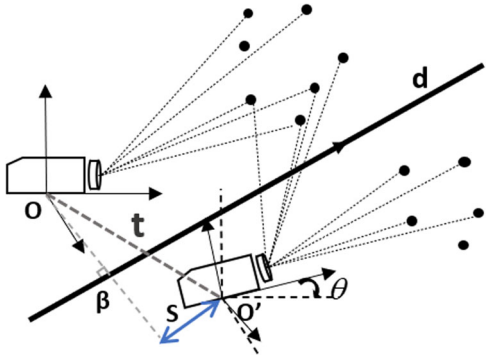

FIGURE 2 Screw motion representation of camera trajectory. [Color figure can be viewed at wileyonlinelibrary.com]

One very important property for $\hat{\mathbf{q}}$ is the unity magnitude constraint, henceforth, Equation (2) must be satisfied for every UDQ.

$$|\hat{\mathbf{q}}| = (q_1 + \varepsilon q_1')^2 + (q_2 + \varepsilon q_2')^2 + (q_3 + \varepsilon q_3')^2 + (q_4 + \varepsilon q_4')^2 = \mathbf{1}, \tag{2}$$

where the subscripts represent the elements of a rotational $\mathbf{q}$ and translation quaternion $\mathbf{q'}$.

The above equation is further expanded to formulate two distinct constraints; the first is inherited from quaternions and the second constraint is typical for the unit dual quaternion.

$$\begin{aligned} (q_1)^2 + (q_2)^2 + (q_3)^2 + (q_4)^2 &= \mathbf{1}, \\ q_1 q_1' + q_2 q_2' + q_3 q_3' + q_4 q_4' &= \mathbf{0}, \end{aligned} \tag{3}$$

Knowing that there are 8 parameters in the unit dual quaternion, the above constraints will result in a system with 6-DOF which is sufficient to model the rigid body motion in free space. Figure 2 illustrates the screw motion representation of keyframe displacement in smoothing-based SLAM. The point $\boldsymbol{\beta}$ is the projection of the origin of frame $\mathbf{O}$ onto the screw axis $\mathbf{d}$.

$$\boldsymbol{\beta} = \frac{1}{2}\left(\mathbf{t} - \mathbf{S}.\mathbf{d} + \cot\left(\frac{\theta}{2}\right)\mathbf{d} \times \mathbf{t}\right). \tag{4}$$

The screw motion pitch $\mathbf{S}$ is determined by the projection of translation vector $\mathbf{t}$ onto the screw axis $\mathbf{d}$ i.e., $\mathbf{S} = \mathbf{t}.\mathbf{d}$. Hence, $\mathbf{S}$ is the perpendicular distance between the axis of the projected point $\boldsymbol{\beta}$ and the origin of body frame $\mathbf{O'}$ and $\mathbf{t}$ is the distance traveled. θ is the screw rotation angle.

Naturally, the unit dual quaternion inherits the properties of the unit quaternion, including conjugate and inverse. The conjugate of a quaternion is $\mathbf{q}^* = (s, \mathbf{-v}) = (q_s, -q_x, -q_y, -q_z)$ while the inverse is computed as

$$\begin{aligned} \mathbf{q}^{-1} &= \left(\frac{\mathbf{1}}{|\mathbf{q}|}\right)^2 \mathbf{q}^*, \\ |\mathbf{q}|^2 &= q_s^2 + q_x^2 + q_y^2 + q_z^2 = \mathbf{q} \otimes \mathbf{q}^* = \mathbf{q}^* \otimes \mathbf{q}, \end{aligned} \tag{5}$$

To rotate a 3D vector $\mathbf{v}$ using a unit quaternion, the vector is placed in the vector part of a quaternion with a vanishing scalar part (Gouasmi, 2012).

$$\mathbf{v} = (v_x, v_y, v_z) \rightarrow \mathbf{q_v} = (v_s, v_x, v_y, v_z). \tag{6}$$

The resulting rotated vector $\mathbf{v'}$ or $\mathbf{q_{v'}} = (v_s', v_x', v_y', v_z')$ is obtained by the transformation applying the quaternion product from left and right.

$$\mathbf{q_{v'}} = \mathbf{q} \otimes \mathbf{q_v} \otimes \mathbf{q}^{-1} = \mathbf{q} \otimes \mathbf{q_v} \otimes \mathbf{q}^*. \tag{7}$$

The unit quaternion $\mathbf{q}$ of the UDQ defines a rotation about an axis $\mathbf{d}$ in the 3D frame while the rotation angle is θ. $\mathbf{d}$ is called the screw axis and θ is the coupled screw dual angle as shown in Figure 2. Then, a simplified nonlinear definition of the quaternion is as follows:

$$\begin{aligned} \mathbf{q} &= \left(\cos\frac{\theta}{2}, \sin\frac{\theta}{2}(d_x, d_y, d_z)\right), \\ \mathbf{q} &= -\mathbf{q}. \end{aligned} \tag{8}$$

The dual part in the screw motion operator is identified by the dual nilpotent parameter ε, satisfying the constraint $\varepsilon^2 = 0$. It follows that the standard notation for the unit dual quaternion in the literature employs this parametrization in the following equation:

$$\hat{\mathbf{q}} = \mathbf{q} + \frac{\varepsilon}{2}\mathbf{q'}. \tag{9}$$

The UDQ product is achieved easily by the special operator $\overline{\otimes}$, it is a composite of quaternion product and quaternion addition.

$$\begin{aligned} \hat{\mathbf{q}} &= \hat{\mathbf{q}}_1 \overline{\otimes}\, \hat{\mathbf{q}}_2, \\ &= \mathbf{q_1} \otimes \mathbf{q_2} + \tfrac{\varepsilon}{2}(\mathbf{q_1} \otimes \mathbf{q'}_2 + \mathbf{q'}_1 \otimes \mathbf{q_2}). \end{aligned} \tag{10}$$

In contrast to the unit quaternion, the UDQ conjugate $\overline{\hat{\mathbf{q}}}$ and UDQ inverse $\hat{\mathbf{q}}^{-1}$ are distinct operators.

$$\begin{aligned} \overline{\hat{\mathbf{q}}} &= \mathbf{q} - \frac{\varepsilon}{2}\mathbf{q'}, \\ \hat{\mathbf{q}}^{-1} &= \frac{1}{\mathbf{q}} - \frac{\varepsilon}{2}\frac{\mathbf{q'}}{\mathbf{q}^2}. \end{aligned} \tag{11}$$

In addition to the UDQ conjugate in Equation (11), there exists a complementary operator $\hat{\mathbf{q}}^*$ and its expression is in Equation (12). Where the unit quaternion inverse is employed.

$$\hat{\mathbf{q}}^* = \mathbf{q}^* + \frac{\varepsilon}{2}\mathbf{q}^{*'}. \tag{12}$$

This complementary conjugate is the element-wise sign flip of quaternions in the UDQ representation as follows: $\hat{\mathbf{q}}^* = (q_s, -q_x, -q_y, -q_z, -q_s', q_x', q_y', q_z')$.

As seen from Equation (1), the rigid body translation is embedded within the dual quaternion. However, it can be easily extracted from

the UDQ by Equation (13). This property is essential for our solution to overcome the parametrization issue for the optimization scheme.

$$\mathbf{t} = 2\mathbf{q}' \otimes \mathbf{q}^*. \tag{13}$$

From all of the above, the rigid body transformation following the screw motion paradigm can easily be represented by the unit dual quaternion. Nevertheless, this representation is also dependent on the nature of body motion. Consider the body frame before and after applying a displacement (rotation plus translation), the frames are related by the dual quaternion where the order of motion types (rotation then translation vice versa) decides the order of the dual part product with the attitude quaternion.

$$\widehat{\mathbf{q}} = \mathbf{q} + \frac{\varepsilon}{2}\mathbf{q}' = \mathbf{q} + \frac{\varepsilon}{2}\mathbf{t}_1 \otimes \mathbf{q} = \mathbf{q} + \frac{\varepsilon}{2}\mathbf{q} \otimes \mathbf{t}_2, \tag{14}$$

where $\mathbf{t}_1 \otimes \mathbf{q}$ is for translation followed by rotation and $\mathbf{q} \otimes \mathbf{t}_2$ is a rotation followed by a translation. It is then evident that to extract the translation vector by Equation (13) depends on the order of operations defined by Equation (14). Then, Equation (13) is further expanded to include a dependency on the order of motions.

$$\begin{aligned} \mathbf{t}_1 &= 2\mathbf{q}' \otimes \mathbf{q}^*, \\ \mathbf{t}_2 &= 2\mathbf{q}^* \otimes \mathbf{q}', \end{aligned} \tag{15}$$

A primordial advantage of using the UDQ is that the Lie algebra methods by rotation matrix or quaternions performs point transformations. However, UDQ performs Plücker line transformations. Thus, any line represented by means of the Plücker lines is transformed between 3D frames. This property is not really explored for VI-SLAM.

3.2 | IMU algorithm by dual quaternion

The idea behind the IMU preintegration is to express the pose-graph constraints with inertial measurements in the local frame. This accomplishes three goals: optimization is applied directly to the preintegrated IMU measurements, frame transformation error is reduced, and computations are minimized. However, the representation using quaternions for attitude, vectors for position, and velocity suffers from a form of loosely coupled integration. The attitude is processed separately while the specific force is integrated into a velocity and position displacement first. Then, a transformation to the current frame is introduced by the updated attitude. Hence, the dual quaternion formulation of the preintegration offers compact and concise motion constraints with an efficient coupled update scheme. It is critical to note that there are no conflicts or assumptions on the motion constraints that contradict the traditional quaternion-vector representation. Thus, the dual quaternion only improves and enriches the pose-graph constraints by introducing the screw motion theory. The basic continuous IMU preintegration for position and velocity in the $SO_3 + SE_3$ formulation is expressed in the camera and body frames as follows:

$$\begin{aligned} {}^{c}\mathbf{p}_{k+1} &= \int_{t_k}^{t_{k+1}} \int_{t_k}^{s} {}^{c}_{\tau}\mathbf{R}(\mathbf{a}_m - \mathbf{b}_a - \mathbf{n}_a)d\tau ds, \\ {}^{c}\mathbf{v}_{k+1} &= \int_{t_k}^{t_{k+1}} {}^{c}_{\tau}\mathbf{R}(\mathbf{a}_m - \mathbf{b}_a - \mathbf{n}_a)d\tau, \\ {}^{b+1}_{b}\mathbf{R} &= \int_{t_k}^{t_{k+1}} (\boldsymbol{\omega}_m - \mathbf{b}_g - \mathbf{n}_g)d\tau, \\ {}^{c}_{b}\mathbf{R} &= {}^{c}_{b}\mathbf{R}\,{}^{b+1}_{b}\mathbf{R}^{T}. \end{aligned} \tag{16}$$

where ${}^{c}_{\tau}\mathbf{R}$ is the rotation matrix from current accelerometer measurement to the camera local frame for the VI-SLAM and ${}^{b+1}_{b}\mathbf{R}$ is the rotation update. ${}^{c}_{b}\mathbf{R}$ is the fixed transformation from the IMU body to camera frame obtained by the calibration step. $\mathbf{a}_m$ and $\boldsymbol{\omega}_m$ are the accelerometers and gyros measurements, $\mathbf{b}_a$ and $\mathbf{b}_g$ are bias and the drift, $\mathbf{n}_a$ and $\mathbf{n}_g$ are the accompanying sensor noise.

In IMU mechanization with screw motion theory, the pre-processed measurements are forth integrated to obtain pose and velocity in the local frame. The quaternion formulation of the attitude integration is straight forward by using the exponential map of the rotation angles. Then, the attitude is used to transform the acceleration or velocity increment to the global or navigation frame depending on the application. However, the update of both states (attitude, translation) is achieved in a single operation with the UDQ product as seen in Equation (10). By applying the screw motion rule to update the twist parameter $\widehat{\boldsymbol{\omega}}^{b}_{ib}$ in Equation (17), we can derive the dual quaternion update of the IMU kinematics. Notice that since the displacement vector is the velocity vector, the original equation is updated considering for velocity and its first derivative.

$$\widehat{\boldsymbol{\omega}}^{b}_{ib} = Exp(\boldsymbol{\omega}^{b}_{ib}) + \frac{\varepsilon}{2}(\mathbf{a}^{b} + \boldsymbol{\omega}^{b}_{ib} \times \mathbf{v}^{b}) \otimes Exp(\boldsymbol{\omega}^{b}_{ib}), \tag{17}$$

where the superscripts define the update frame and the operator × is the cross product. It is clear that the twist $\widehat{\boldsymbol{\omega}}^{b}_{ib}$ is in the UDQ form. This makes it simpler to update the IMU kinematics in a concise and compact manner as shown in Equation (18). As it can be seen, both IMU ideal measurements are coupled in a single update under the screw motion generality.

$$2\dot{\widehat{\mathbf{q}}}_{ib} = \widehat{\mathbf{q}}_{ib} \,\overline{\otimes}\, \widehat{\boldsymbol{\omega}}^{b}_{ib}. \tag{18}$$

The attitude update by rotation vector is analogous to the screw update to be used for the unit dual quaternion. While the rotation vector update by direction cosines requires the two-speed approach to account for the noncommutativity error, the quaternion update does not require such explicit error correction. Because any quaternion product performs a cross product of the vector parts implicitly.

Finally, we can formulate the preintegrated twist increment update $\Delta\widehat{\boldsymbol{\sigma}}_{ij}$ of the duration Δt_{ij} between two consecutive keyframes by Equation (19). The update by UDQ increment is analogous to the mapping inherited from the quaternions as shown in Equation (20). In the following equations, any variable state notated with "Δ" refers to a state increment during the time period. Similarly, the notation with "δ" refers to the error in the state.

$$\Delta\widehat{\boldsymbol{\sigma}}_{ij} = \sum_{k=i}^{j-1}\left[\left(\boldsymbol{\omega}^{b}_{ib_k} - \mathbf{b}^{g}_{k} - \boldsymbol{\eta}^{g}_{k}\right)\Delta t_{ik} + \frac{\varepsilon}{2}\left(\left(\mathbf{a}^{b}_{k} - \mathbf{b}^{a}_{k} - \boldsymbol{\eta}^{a}_{k}\right)\Delta t_{ik} \otimes \mathbf{q}_{ik}\right)\right], \tag{19}$$

$$\widehat{\mathbf{q}}(\Delta t_{ik}) = \left[\cos\left(\frac{\|\Delta\widehat{\boldsymbol{\sigma}}_{ik}\|}{2}\right), \frac{\Delta\widehat{\boldsymbol{\sigma}}_{ik}}{\|\Delta\widehat{\boldsymbol{\sigma}}_{ik}\|}\sin\left(\frac{\|\Delta\widehat{\boldsymbol{\sigma}}_{ik}\|}{2}\right)\right], \tag{20}$$

where $\boldsymbol{\omega}_{ib_k}^b$, $\mathbf{a}_k^b$ are the kth gyroscopes and accelerometer measurements, $\mathbf{b}_i^g$, $\mathbf{b}_k^a$ are the piecewise drift and bias between two frames (i-j), $\boldsymbol{\eta}_k^g$, $\boldsymbol{\eta}_k^a$ are the associate noise terms. $\mathbf{q}_{ik}$ is the attitude variation (from kth to ith keyframe).

The introduced twist increment summarizes the IMU measurements within the current preintegration interval in the local frame. Here, we only focus on the velocity and attitude computations and leave the integration of position aside. Although it may seem convenient to sum all gyroscopes measurements as they arrive, it is necessary to update the rotation quaternion at every time step to transform the acceleration to the current frame as it is the case with on-manifold formulation.

From previous equations, it is clear that the twist increment update is dependent on biases and noise terms. Therefore, we first isolate the noise terms in the dual quaternion update to simplify the covariance update. Considering that the bias and drift should be stable during the current preintegration interval and considering the piecewise is discretized. We assume piecewise constant gyroscopes and acceleration measurements to simplify the integration process in the navigation computer. This step is important to build the nonlinear optimization system.

3.3 | Noise covariance

It is evident that the formulation of the noise propagation by quaternion is not very convenient for the covariance factorization. Also, the matrix becomes large during the coding process, for each quaternion is represented by 4 × 4 matrix. Nevertheless, the simplicity of conversion between quaternion and rotation matrix for attitude can be an advantage to formulate the minimal representation of the errors. Additionally, we propose a flexible mapping between the formulation by the screw motion in the UDQ paradigm to the SE_3 minimal formula. This mapping discussed in the next section proves the duality between the Lie algebra and the screw motion theory. Where a simple quaternion product by Equation (15) performs the transition smoothly, allowing to harness the advantages of UDQ without singularity issues.

The resulting linearized system of the covariance is simplified in Equation (21), to be implemented in the minimum representation. Therefore, no over-parametrization issues occur by the quaternion representation of the rotation and the translation.

where $\delta\boldsymbol{\varphi}, \delta\mathbf{v}, \delta\mathbf{p}$ are the attitude, velocity and position errors, respectively. $\mathbf{I}$ is the identity matrix, $\mathbf{J}^r$ is the right Jacobian that maps from parameter space to the tangent of the space manifold, $\Delta\mathbf{R}$ is the direction matrix and $\mathbf{Q}^R_{(0:2,0:2)}$ is the 3 × 3 right quaternion matrix that can be extracted from a unit quaternion. Similarly, the left quaternion matrix is extracted from the quaternion as follows.

$$\mathbf{Q}^L = \begin{bmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & q_w & -q_z & q_y \\ q_y & q_z & q_w & -q_x \\ q_z & -q_y & q_x & q_w \end{bmatrix}, \mathbf{Q}^R = \begin{bmatrix} q_w & -q_x & -q_y & -q_z \\ q_x & q_w & q_z & -q_y \\ q_y & -q_z & q_w & q_x \\ q_z & q_y & -q_x & q_w \end{bmatrix}. \tag{22}$$

3.4 | Solving for over-parametrization

The derivation of the optimization problem is similar to that based on SO_3 but with an extra step that is necessary to overcome the over-parametrization by unit dual quaternion. Eventually, this is the first work to formulate a full optimization problem based on unit dual quaternion but also avoid the singularity induced by it. The steps are the following:

1. Propagate the inertial measurements using the screw motion theory by Equations (19) and (20).
2. Isolate the quaternion in question (any factor to error terms in the residual function) and simplify it to basic quaternion products.
3. The resulting translation should be a rotation quaternion product with a translation 3D vector, that is, $\mathbf{q}' = \mathbf{t} \otimes \mathbf{q}$.
4. Transform the dual part to SE_3 by using Equations (13) or (15).
5. Extract the parameters by the conventional derivative rule of quaternion.

The noise covariance and bias Jacobians, in addition to the residual Jacobians, were derived by respecting the above steps.

3.5 | Bias correction

The first order expansion is employed to update the pose information after the bias value is optimized. Since gyro bias correction is an additional rotation and the acceleration bias is a translation correction obtained by the least-squares method, it is more convenient to be applied separately. As a result, their individual Jacobians are critical for carrying out the first-order bias update. Note that at the end of the

$$\begin{bmatrix} \delta\boldsymbol{\varphi}_{ij} \\ \delta\mathbf{v}_{ij} \\ \delta\mathbf{p}_{ij} \end{bmatrix} = \begin{bmatrix} Exp\left[\left(\boldsymbol{\omega}_{ib_{j-1}}^b - \mathbf{b}_i^g\right)\Delta t\right]^T & \mathbf{0} & \mathbf{0} \\ \left(-\Delta\mathbf{R}_{ij-1}\left[\left(\mathbf{a}_{j-1}^b - \mathbf{b}_{j-1}^a\right)\Delta t\right]_\times + [\Delta\mathbf{v}_{ij-1}]_\times\Delta\mathbf{R}_{ij-1}\right) & \mathbf{I} & \mathbf{0} \\ \left(-\frac{1}{2}\Delta\mathbf{R}_{ij-1}\left[\left(\mathbf{a}_{j-1}^b - \mathbf{b}_{j-1}^a\right)\Delta t^2\right]_\times + [\Delta\mathbf{p}_{ij-1}]_\times\Delta\mathbf{R}_{ij-1}\right) & [\mathbf{I}\Delta t]_\times & \mathbf{I} \end{bmatrix} \begin{bmatrix} \delta\boldsymbol{\varphi}_{ij-1} \\ \delta\mathbf{v}_{ij-1} \\ \delta\mathbf{p}_{ij-1} \end{bmatrix}$$
$$+ \begin{bmatrix} \mathbf{J}^r\left(\boldsymbol{\omega}_{ib_{j-1}}^b - \mathbf{b}_i^g\right)\Delta t & \mathbf{0} \\ \left(\Delta\mathbf{R}_{ij-1}\left[\left(\mathbf{a}_{j-1}^b - \mathbf{b}_{j-1}^a\right)\Delta t\right]_\times + [\Delta\mathbf{v}_{ij-1}]_\times\Delta\mathbf{R}_{ij-1}\right)\left[\mathbf{Q}^R_{(0:2,0:2)}\mathbf{J}^R\right]\Delta t & \Delta\mathbf{R}_{ij-1}\Delta t \\ \frac{1}{2}\left(\Delta\mathbf{R}_{ij-1}\left[\left(\mathbf{a}_{j-1}^b - \mathbf{b}_{j-1}^a\right)\Delta t\right]_\times + [\Delta\mathbf{v}_{ij-1}]_\times\Delta\mathbf{R}_{ij-1}\right)\left[\mathbf{Q}^R_{(0:2,0:2)}\mathbf{J}^R\right]\Delta t^2 & \frac{1}{2}\Delta\mathbf{R}_{ij-1}\Delta t^2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_{j-1}^g \\ \boldsymbol{\eta}_{j-1}^a \end{bmatrix}, \tag{21}$$

derivation we obtain a rotation matrix because the Jacobian of a quaternion with respect to a vector is always a rotation matrix. Fortunately, it is more suitable to use such notion to achieve the update by conversion $\Delta\mathbf{R}_{ij} = f(\mathbf{q}_{ij})$. Additionally, utilizing the classic rotation metric is more concise to the classic control theory. Making the update of the covariance and the Jacobians a straightforward operation. The resulting bias Jacobians are introduced in the following.

$$
\begin{aligned}
\frac{\partial \mathbf{q}_{ik}}{\partial \mathbf{b}^g} &\simeq -\sum_{k=i}^{j-1}\left(\Delta\mathbf{R}_{jk+1}\mathbf{J}_k^r\Delta t\right),\\
\frac{\partial \mathbf{q}'_{ik}}{\partial \mathbf{b}^a} &\simeq -\sum_{k=i}^{j-1}(\Delta\mathbf{R}_{ik}\Delta t),\\
\frac{\partial \mathbf{q}'_{ik}}{\partial \mathbf{b}^g} &\simeq -\sum_{k=i}^{j-1}\left(\begin{array}{l}\left(\Delta\mathbf{R}_{ik}\left[\left(\mathbf{a}_k^b-\mathbf{b}_k^a\right)\Delta t\right]_\times-[\Delta\mathbf{v}_{ik}]_\times\Delta\mathbf{R}_{ik}\right)\left(\frac{\partial \mathbf{q}_{ik}}{\partial \mathbf{b}^g}\right)\\ +\left(\begin{array}{l}\left(\Delta\mathbf{R}_{ik}\left(\mathbf{Q}^R_{(0:2,0:2)}\left(\boldsymbol{\omega}_{ib_k}^b-\mathbf{b}_k^g\right)^*\right)\left[\left(\mathbf{a}_k^b-\mathbf{b}_k^a\right)\Delta t\right]_\times\right)\left[\mathbf{J}_k^R\Delta t\right]\\ +[\Delta\mathbf{v}_{ik}]_\times\Delta\mathbf{R}_{ik}\left(\mathbf{Q}^R_{(0:2,0:2)}\left(\boldsymbol{\omega}_{ib_k}^b-\mathbf{b}_k^g\right)^*\right)\end{array}\right)\end{array}\right),\\
\frac{\partial \mathbf{p}'_{ik}}{\partial \mathbf{b}^a} &= -\sum_{k=i}^{j-1}\frac{3}{2}(\mathbf{q}_{ik}\otimes\Delta t^2)\simeq-\sum_{k=i}^{j-1}\frac{3}{2}(\Delta\mathbf{R}_{ik}\Delta t^2),\\
\frac{\partial \mathbf{p}'_{ik}}{\partial \mathbf{b}^g} &\simeq -\sum_{k=i}^{j-1}\left(\begin{array}{l}\frac{3}{2}\left(\Delta\mathbf{R}_{ik}\left[\left(\mathbf{a}_k^b-\mathbf{b}_k^a\right)\Delta t^2\right]_\times-[\Delta\mathbf{p}_{ik}]_\times\Delta\mathbf{R}_{ik}\right)\left(\frac{\partial \mathbf{q}_{ik}}{\partial \mathbf{b}^g}\right)\\ +\left(\begin{array}{l}\left(\frac{3}{2}\Delta\mathbf{R}_{ik}\left(\mathbf{Q}^R_{(0:2,0:2)}\left(\boldsymbol{\omega}_{ib_k}^b-\mathbf{b}_k^g\right)^*\right)\left[\left(\mathbf{a}_k^b-\mathbf{b}_k^a\right)\Delta t^2\right]_\times\right)\left[\mathbf{J}_k^R\Delta t\right]\\ +[\Delta\mathbf{p}_{ik}]_\times\Delta\mathbf{R}_{ik}\left(\mathbf{Q}^R_{(0:2,0:2)}\left(\boldsymbol{\omega}_{ib_k}^b-\mathbf{b}_k^g\right)^*\right)\end{array}\right)\end{array}\right).
\end{aligned}
\tag{23}
$$

This derivation is comparable to the work of Forster in (Forster et al., 2016) but with richer information coded within the gyros bias Jacobians that accounts for the screw motions' nature. (Making it clearer that there is a duality between the dual quaternion formulation and the classical homogenous form). However, some terms can be safely ignored to reduce the computational cost of the update.

4 | OPTIMIZATION IN THE SCREW MOTION THEORY FRAMEWORK

The SLAM system is divided into two ends: front-end and back-end. The front-end is responsible for constructing a graph using the sensors' raw measurements and is also responsible for estimating the data association. The backend determines the most likely configuration of nodes, edges, and vertices of the constructed graph through the optimization algorithm (nonlinear least square, bundle adjustment) as shown in Figure 3. However, the Gaussian assumption of the measurement model does not hold because of the multimodality, that is, the observation can be fitted to multiple edges connecting different poses. Thus, the resulting intractable multi-modal estimation is restricted to a basis for the data association problem solved by the front end (Grisetti et al., 2010). The goal of the graph-SLAM is to obtain a mean for the Gaussian that represents the posterior of the robot's trajectory, represented in our case by the screw motion constraints. Where the mean is the maximum likelihood configuration of nodes that is concise with the constructed edges. Hence, the graph-SLAM solves a minimization problem of the log-likelihood that a measurement $\mathbf{Z}_{ij}$ relates two frames or nodes $\mathbf{x}_i$ and $\mathbf{x}_j$ which is equal to the transformation of their origin by the twist.

In a probabilistic fashion, we set the function $\mathbf{E}_{ij}$ as the difference between the observation and the expected measurement. $\mathbf{E}_{ij} = \mathbf{Z}_{ij} - \widehat{\mathbf{Z}}_{ij}(\mathbf{x}_i, \mathbf{x}_j)$. Then, the maximum likelihood approach aims to minimize this cost function's negative log format. Therefore, it should obtain a concise optimal configuration of the nodes in the graph.

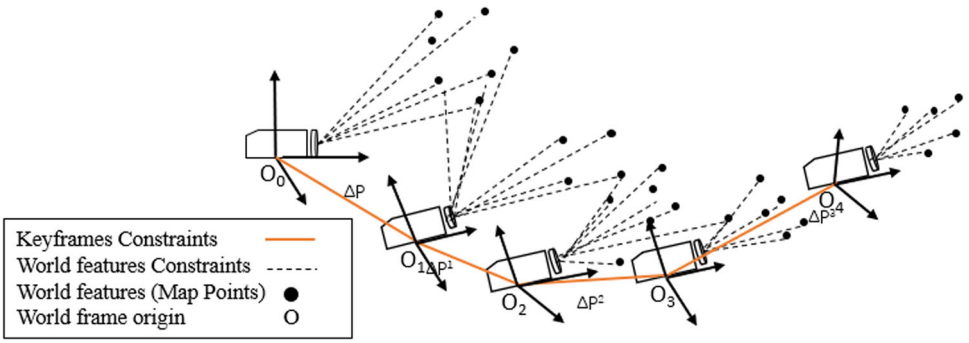

FIGURE 3 Graph of the camera trajectory by keyframes and inertial measurement unit constraints. [Color figure can be viewed at wileyonlinelibrary.com]

$$\tilde{\mathbf{x}} = \underset{\delta(\boldsymbol{\varphi}_i,\boldsymbol{\varphi}_j,\mathbf{v}_i,\mathbf{v}_j,\mathbf{p}_i,\mathbf{p}_j)}{\operatorname{argmin}} \sum_{i}^{j-1} \mathbf{E}_{ij}{}^{T}\boldsymbol{\Lambda}_{ij}\mathbf{E}_{ij}, \tag{24}$$

where $\boldsymbol{\Lambda}_{ij}$ is the information matrix that weighs the residual vector $\mathbf{E}_{ij}$. The solution of this nonlinear cost function using the least squares algorithm is computed by linearizing the error function around the current guess. Subsequently, the residual to be minimized becomes a function of the Jacobian, the Hessian, and the information matrix.

4.1 | Error function

The unit dual quaternion formulation of the residual error between two vertices of the constraint can be extracted by basic dual quaternion algebra. Equation (25) extracts the residual of the coupled attitude and velocity. The residual of the position is also derived using the Newtonian algebra to optimize the computation cost. It is worth noting that the residual parameter $\mathbf{E}_{ij}$ only contains the vector part of the quaternions, which reduces the size of the matrices and enables linear control reasoning. Another advantage of discarding the scalar part is the consistency with the covariance matrix, which if over parametrized will be singular.

$$\begin{aligned}\mathbf{E}_{ij} &= 2\left(\hat{\mathbf{q}}_{ij}^{*} \otimes \hat{\mathbf{q}}_i^{*} \otimes \hat{\mathbf{q}}_j\right) = \mathbf{E}_{ij}^{q} + \mathbf{E}_{ij}',\\ &\approx 2\Big[\mathbf{q}_{ij}^{*} \otimes \mathbf{q}_i^{*} \otimes \mathbf{q}_j + \frac{\varepsilon}{2}\Big(\mathbf{q}_j^{*} \otimes (\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}) \otimes \mathbf{q}_j\\ &\quad - \mathbf{q}_{ij}^{*} \otimes \Delta\mathbf{v}_{ij} \otimes \mathbf{q}_{ij}\Big)\Big],\end{aligned} \tag{25}$$

where $\mathbf{E}_{ij}^{q}$ is the attitude residual error and $\mathbf{E}_{ij}'$ is the dual residual error.

The gravity contribution must be accounted for before the residual extraction. Also, the corrections to the screw motion constraint increment are applied beforehand in Equation (26). The bias estimation is updated using the bias Jacobians, which should be a 3 3 matrix. Hence, the rotation quaternion correction is actually a lift of the bias error to the exponential map. While the update of velocity and position is a simple vector addition.

$$\begin{aligned}&\mathbf{q}_{ij} = \mathbf{q}_{ij} \otimes Exp\left(\frac{\partial \mathbf{q}_{ij}}{\partial \mathbf{b}^{g}}\delta\mathbf{b}^{g}\right),\\ &2\mathbf{q}_{ij}' \otimes \mathbf{q}_{ij} = 2\mathbf{q}_{ij}' \otimes \mathbf{q}_{ij} + \left(\frac{\partial \mathbf{q}_{ij}'}{\partial \mathbf{b}^{g}}\delta\mathbf{b}^{g} + \frac{\partial \mathbf{q}_{ij}'}{\partial \mathbf{b}^{a}}\delta\mathbf{b}^{a}\right) \otimes \mathbf{q}_{ij},\\ &2\mathbf{p}_{ij}' \otimes \mathbf{q}_{ij} = 2\mathbf{p}_{ij}' \otimes \mathbf{q}_{ij} + \left(\frac{\partial \mathbf{p}_{ij}'}{\partial \mathbf{b}^{g}}\delta\mathbf{b}^{g} + \frac{\partial \mathbf{p}_{ij}'}{\partial \mathbf{b}^{a}}\delta\mathbf{b}^{a}\right) \otimes \mathbf{q}_{ij}.\end{aligned} \tag{26}$$

4.2 | Jacobians of the residuals

The Jacobian matrices enable the use of optimization methods to achieve an optimal correction. To obtain the correct Jacobians for the unit dual quaternion cost function, first, a perturbation of the parameters by an added error is required. Where we introduce an error term to the parameter space we want to optimize.

4.2.1 | Quaternion error

Before we proceed to the derivation of the residual Jacobians, we must clarify the notation for the quaternion error. Similar to that of the rotation matrix, the quaternion error uses an exponential mapping. Ideally, the quaternion is a product of an ideal state $\tilde{\mathbf{q}}$ and an error term $\delta\mathbf{q}$ which consist of the mapping of the infinitesimal 3 directional error dependent on $\delta\boldsymbol{\varphi}$: $\mathbf{q} = \tilde{\mathbf{q}} \otimes \delta\mathbf{q}$ or $\mathbf{q} = \tilde{\mathbf{q}} \otimes e^{\delta\boldsymbol{\varphi}}$. Henceforth, quaternion error is expressed as follows.

$$\delta\mathbf{q} = \begin{bmatrix}1\\ \frac{1}{2}\delta\varphi_x\\ \frac{1}{2}\delta\varphi_y\\ \frac{1}{2}\delta\varphi_z\end{bmatrix} = \begin{bmatrix}1\\0\\0\\0\end{bmatrix} + \begin{bmatrix}0\\ \frac{1}{2}\delta\varphi_x\\ \frac{1}{2}\delta\varphi_y\\ \frac{1}{2}\delta\varphi_z\end{bmatrix}. \tag{27}$$

Following Equation (27), the error in the quaternion is simplified to an identity quaternion with only scalar part added to the half angle rotation of the misalignment. This property is proven useful for the optimization of UDQ graph-based SLAM as shall be proven in the following section.

4.2.2 | Perturbations of attitude

As previously discussed, the attitude represented by the unit quaternion is dependent only on the bias change. Note that the bias is considered constant or piecewise constant during the IMU preintegration. The corresponding residual Jacobians are presented next, where we get a sparse matrix block.

$$\begin{aligned}&\frac{\partial \mathbf{E}_{ij}^{q}}{\partial\delta\boldsymbol{\varphi}_i} = -\left[\left(\mathbf{q}_{ij} \otimes Exp\left(\frac{\partial \mathbf{q}_{ij}}{\partial \mathbf{b}^{g}}\delta\mathbf{b}^{g}\right)\right)^{*}\right]_{(0:2,0:2)}^{L}\left[\mathbf{q}_i^{*} \otimes \mathbf{q}_j\right]_{(0:2,0:2)}^{R},\\ &\frac{\partial \mathbf{E}_{ij}^{q}}{\partial\delta\boldsymbol{\varphi}_j} = \left[\mathbf{E}_{ij}^{q}\right]_{(0:2,0:2)}^{L},\\ &\frac{\partial \mathbf{E}_{ij}^{q}}{\partial\delta\mathbf{p}_i} = \mathbf{0}, \frac{\partial \mathbf{E}_{ij}^{q}}{\partial\delta\mathbf{p}_j} = \mathbf{0}, \frac{\partial \mathbf{E}_{ij}^{q}}{\partial\delta\mathbf{v}_i} = \mathbf{0}, \frac{\partial \mathbf{E}_{ij}^{q}}{\partial\delta\mathbf{v}_j} = \mathbf{0},\\ &\frac{\partial \mathbf{E}_{ij}^{q}}{\partial\delta\delta\mathbf{b}^{g}} = -2^{*}\left[\mathbf{q}_{ij}^{*} \otimes (\mathbf{q}_i^{*} \otimes \mathbf{q}_j)\right]_{(0:2,0:2)}^{R}\left(\frac{\partial \mathbf{q}_{ij}}{\partial \mathbf{b}^{g}}\right),\\ &\frac{\partial \mathbf{E}_{ij}^{q}}{\partial\delta\delta\mathbf{b}^{a}} = 0.\end{aligned} \tag{28}$$

4.2.3 | Perturbation of the translation

The perturbation of translation takes the unit dual quaternion where the SE_3 translation vector is transformed to be the dual by the half rotation quaternion product. Thereafter, it is simpler to perturb the original local frame translation vector and do the derivations in the screw motion paradigm.

$$\mathbf{q}_i' \leftrightarrow \frac{\varepsilon}{2}((\mathbf{t}_i + \delta\mathbf{t}) \otimes \mathbf{q}_i). \tag{29}$$

Adopting this derivation makes it easier to define the retraction then update the estimates of vertices by a simple addition in the minimal representation by SE_3. The following perturbations are introduced to Equation (25) or Equation (32): $\mathbf{E}'_{ij}(\mathbf{v}_i + \delta\mathbf{v}_i)$, $\mathbf{E}'_{ij}(\mathbf{v}_j + \delta\mathbf{v}_j)$, $\mathbf{E}'_{ij}(\mathbf{q}_i \otimes e^{\delta\boldsymbol{\varphi}_i})$, $\mathbf{E}'_{ij}(\mathbf{q}_j \otimes e^{\delta\boldsymbol{\varphi}_j})$, $\mathbf{E}'_{ij}(\mathbf{b}_i^a + \delta\mathbf{b}_i^a)$, $\mathbf{E}'_{ij}(\mathbf{b}_i^g + \delta\mathbf{b}_i^g)$. The translation is dependent on the orientation states, bias and drift. Hence, the Jacobians with respect to ***i-th*** and ***j-th*** frames are both present.

$$\begin{aligned}
\frac{\partial \mathbf{E}'_{ij}}{\partial \delta\boldsymbol{\varphi}_i} &= \Delta\mathbf{R}_{ij}^T[\Delta\mathbf{v}_{ij}]_\times,\\
\frac{\partial \mathbf{E}'_{ij}}{\partial \delta\boldsymbol{\varphi}_j} &= [\mathbf{E}'_{ij}]_\times,\\
\frac{\partial \mathbf{E}'_{ij}}{\partial \delta\mathbf{p}_i} &= \mathbf{0}, \frac{\partial \mathbf{E}'_{ij}}{\partial \delta\mathbf{p}_j} = \mathbf{0},\\
\frac{\partial \mathbf{E}'_{ij}}{\partial \delta\mathbf{v}_i} &= -\mathbf{R}_j^T,\\
\frac{\partial \mathbf{E}'_{ij}}{\partial \delta\mathbf{v}_j} &= \mathbf{R}_j^T,\\
\frac{\partial \mathbf{E}'_{ij}}{\partial \delta\delta\mathbf{b}^g} &= -\Delta\mathbf{R}_{ij}^T\frac{1}{2}\frac{\partial \widehat{\mathbf{q}}'_{ij}}{\partial \mathbf{b}^g},\\
\frac{\partial \mathbf{E}'_{ij}}{\partial \delta\delta\mathbf{b}^a} &= -\Delta\mathbf{R}_{ij}^T\frac{1}{2}\frac{\partial \widehat{\mathbf{q}}'_{ij}}{\partial \mathbf{b}^a}.
\end{aligned} \tag{30}$$

4.2.4 | Perturbation of the position

Similar to the translation perturbation, the perturbation of the position takes the dual quaternion where the constraint $\mathbf{p}'^*_{ij}$ is the position increment in dual form. In fact, the position formulation is the integrated velocity during the period of the preintegration and this is evident from the Newton formula.

$$\begin{aligned}
\frac{\partial \mathbf{E}'_{ij}[\mathbf{p}]}{\partial \delta\boldsymbol{\varphi}_i} &= \Delta\mathbf{R}_{ij}^T[\Delta\mathbf{p}_{ij}]_\times,\\
\frac{\partial \mathbf{E}'_{ij}[\mathbf{p}]}{\partial \delta\boldsymbol{\varphi}_j} &= [\mathbf{E}'_{ij}[\mathbf{p}]]_\times,\\
\frac{\partial \mathbf{E}'_{ij}[\mathbf{p}]}{\partial \delta\mathbf{p}_i} &= -\mathbf{R}_j^T,\\
\frac{\partial \mathbf{E}'_{ij}[\mathbf{p}]}{\partial \delta\mathbf{p}_j} &= \mathbf{R}_j^T,\\
\frac{\partial \mathbf{E}'_{ij}[\mathbf{p}]}{\partial \delta\mathbf{v}_i} &= -\mathbf{R}_j^T\Delta t_{ij},\\
\frac{\partial \mathbf{E}'_{ij}[\mathbf{p}]}{\partial \delta\mathbf{v}_j} &= \mathbf{0},\\
\frac{\partial \mathbf{E}'_{ij}[\mathbf{p}]}{\partial \delta\delta\mathbf{b}^g} &= -\Delta\mathbf{R}_{ij}^T\frac{1}{2}\frac{\partial \mathbf{p}'_{ij}}{\partial \mathbf{b}^g},\\
\frac{\partial \mathbf{E}'_{ij}[\mathbf{p}]}{\partial \delta\delta\mathbf{b}^a} &= -\Delta\mathbf{R}_{ij}^T\frac{1}{2}\frac{\partial \mathbf{p}'_{ij}}{\partial \mathbf{b}^a}.
\end{aligned} \tag{31}$$

4.3 | Formulation in an optimized frame

The derived Jacobians are comparable to the derivation based on Lie algebra. This is due to choices regarding truncation and optimization frame. However, the screw motion paradigm has additional terms of cross products that account for translational-rotational relativity. These terms add computation throughput but enhance the perception of the real-world motion. It is also noticeable that the residual function in the case of unit dual quaternion formulation transforms the states to the ***j***th frame, contrary to other implementations that use the ***i***th frame to reduce computation. Additionally, the sparsity of the residual Jacobians and Hessian is affected by this extra step. This, however, can be mitigated by using an alternative residual function shown in the following equation:

$$\begin{aligned}
\mathbf{E}_{ij} &= 2\left(\widehat{\mathbf{q}}_i^* \otimes \widehat{\mathbf{q}}_j \otimes \widehat{\mathbf{q}}_{ij}^*\right) = \mathbf{E}_{ij}^q + \mathbf{E}'_{ij},\\
&\approx 2\left[\mathbf{q}_i^* \otimes \mathbf{q}_j \otimes \mathbf{q}_{ij}^* + \frac{\varepsilon}{2}\left(\mathbf{q}_i^* \otimes (\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}) \otimes \mathbf{q}_i - \Delta\mathbf{v}_{ij}\right)\right].
\end{aligned} \tag{32}$$

The above equation is similar to that based on $SO_3 + SE_3$, which means that the Jacobians of the residual will be closely related, proving even more the duality. We chose to do our testing with the first cost function to evaluate its convergence properties.

5 | EXPERIMENTAL RESULTS

5.1 | Simulations

The method based on unit dual quaternion should have an effect on the convergence of the smoothing-based SLAM, the execution time, the memory cost, and the accuracy of the final solution. We test the efficacy of the proposed method by simulation in the MATLAB environment, where an unmanned aerial vehicle (UAV) flies to a high altitude. The simulation executes the preintegration based on the proposed screw update and the method based on Lie algebra for comparison.

The first metric that we investigate is the memory cost, where the homogenous matrix is replaced by the UDQ representation of the screw motion. Figure 4 shows that the memory requirement is

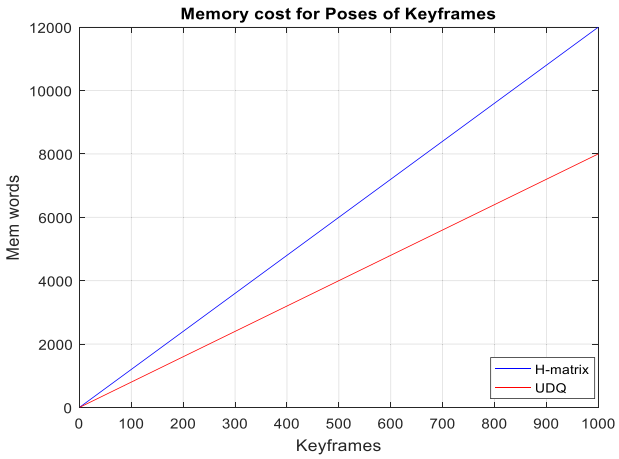

FIGURE 4 Memory cost of the homogenous matrix and the unit dual quaternion. [Color figure can be viewed at wileyonlinelibrary.com]

reduced by 33% when saving the 8 pose parameters instead of the 12 parameters of the homogenous matrix. This aspect is useful when the path of the robot is long, such as in a large city scenario. Next, we compare the integration by screw vector to the existing method by Forster et al. (2015). Figure 5 shows that the integration of the IMU data during the flight test is comparable by the two methods. The position error is reduced by the UDQ formulation by an order of ~3% compared to the $SO_3 + SE_3$ algorithm. While the attitude error is reduced by ~5% due to the commutativity advantage of the quaternions. However, as shall be seen next, the computation cost is heavier for the UDQ algorithm because of additional terms in the integration of measurements, covariance, and Jacobian updates. The average computational cost is about 25% increase over that of the Lie algebra algorithm.

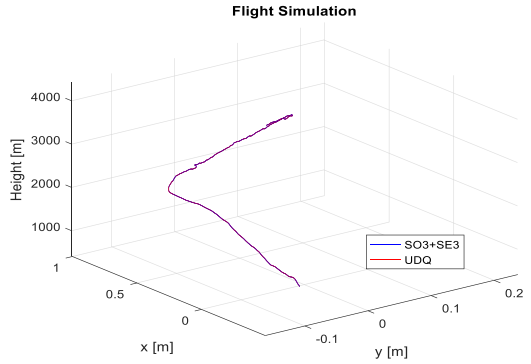

FIGURE 5 Inertial measurement unit preintegration by unit dual quaternion versus preintegration by SO_3 during a simulated unmanned aerial vehicle flight to high altitude. [Color figure can be viewed at wileyonlinelibrary.com]

5.2 | Real-world experiment

The real-world experiment using the publicly available EuRoC MAV data sets (Burri et al., 2016) validates the proposed algorithm. The data sets use two Aptina MT9V034 global shutter cameras at 20 FPS with an image resolution of 752 480 pixels and a MEMS ADIS16448 IMU at 200 Hz rate. Images and IMU are synchronized properly, so we can ignore their time offset. Additionally, a suitable spatial calibration is provided from left-to-right camera and IMU to camera. The implementation of the proposed SLAM algorithm is based on ORB-SLAM3, which uses the g2o optimization algorithm and ORB feature extractor. Another implementation is based on VINS-FUSION (Qin et al., 2018), where ceres optimization library and Shi-Tomasi features with KLT-tracking are employed. This method of testing allows a fair comparison of the algorithm based on screw motion to the other versions. Where the algorithm optimization library and features extractors and all bells and whistles are mutual and no longer influential.

The UDQ formulation rectifies the original smoothing-based SLAM algorithm in the IMU preintegration routine, Jacobians, and optimization graph for initialization and bundle adjustment (BA). While keeping the back-end structure based on g2o/ceres graph-solver and the visual aspect of the front-end. The optimization of inertial edges in the sliding window executes 200 iterations in the ORB-SLAM3, but in the case of the UDQ formulation it is set between 20 and 40 iterations only. A comparison of both algorithms' metrics, including accuracy, and execution times, is provided for the Stereo-Inertial case of the SLAM algorithm.

Table 1 is the comparison of the algorithm based on UDQ and the method based on (Forster et al., 2016) of ORB-SLAM3. The results are for the Stereo-Inertial case, where all the metrics are provided. The performance is comparable for both methods, where

TABLE 1 Error comparison of ORB-SLAM ($SO_3 + SE_3$) and proposed method.

	$SO_3 + SE_3$ (cm)					UDQ (cm)				
Data set	RMSE	Mean	Median	Std	Max	RMSE	Mean	Median	Std	Max
MH01	6.629	5.844	5.477	3.128	14.58	**5.85**	**5.146**	**4.833**	**2.783**	**13.88**
MH02	4.250	3.738	3.286	2.022	9.570	**3.946**	**3.570**	**3.234**	**1.682**	**9.144**
MH03	**4.863**	**4.319**	**3.874**	**2.235**	**10.69**	5.688	4.997	4.422	2.716	12.28
MH04	**6.584**	**6.063**	**6.127**	**2.565**	17.18	7.153	7.420	7.109	3.065	**15.83**
MH05	8.357	7.429	7.196	3.826	18.68	**6.307**	**5.745**	**5.313**	**2.601**	**14.18**
V101	**2.906**	**2.615**	**2.637**	**1.266**	**5.940**	3.244	2.943	2.895	1.365	6.852
V102	**6.008**	**5.752**	**5.754**	**1.736**	**9.441**	7.349	7.074	7.042	1.991	13.39
V103	**5.072**	**4.377**	**3.476**	**2.562**	**11.31**	8.088	7.158	6.813	3.766	24.43
V201	**6.436**	**6.114**	**6.699**	**2.010**	**9.637**	7.125	6.773	7.364	2.211	10.17
V202	5.823	5.330	4.971	2.344	11.03	**5.596**	**5.187**	**4.915**	**2.261**	**10.79**
V203	6.409	6.047	5.558	2.121	17.53	**6.300**	**6.000**	**5.625**	**1.920**	**11.68**

Note: The best performance results are in bold.

Abbreviations: RMSE, root mean square error; UDQ, unit dual quaternion.

UDQ outperforms the $SO_3 + SE_3$ method in some cases in which the motion is initially rich and the initialization of the IMU is straightforward. The Lie algebra method still outperforms the UDQ method in some cases by an order of millimeters. Nevertheless, the computational advantage and convergence properties in Table 2 is in favor of the latter.

Similar pattern is observed by VINS-FUSION experiments with EuRoC MAV data sets. The algorithm employs different optimization and feature extraction packages than ORB-SLAM3. Table 3 witnesses that the screw motion paradigm can be effectively introduced to any smoothing-based VI-SLAM and have a positive effect in general. The convergence properties by the proposed algorithm are superior and the memory cost is reduced.

Table 2 approves that the computational cost mean of the proposed preintegration is heavier than the method based on Lie algebra. It is evident from the formulation that additional terms account for the types of motion, where both linear and angular velocity have an interchangeable effect on the pose (position, orientation). However, the prediction cost by the SLAM algorithm is reduced, and the overall tracking cost is also reduced for the entire EuRoC MAV datasets. This improvement is mainly due to the efficiency of the motion model based on the UDQ.

Figure 6 validates the formulation of the inertial algorithm by the screw motion. It has successfully reduced the error while suppressing the computation cost. Even though the inertial optimization in the VI-SLAM backend performs fewer optimization iterations in the proposed method, it has surpassed the algorithm on-manifold. The estimated trajectory in the EuRoC MAV MH01 data set is depicted in Figure 7. Both methods have a good estimation of the body poses, but UDQ aligns closer to the ground truth trajectory than its counterpart based on $SO_3 + SE_3$ of ORB-SLAM3.

TABLE 2 Execution time cost comparison of ($SO_3 + SE_3$) and UDQ algorithms.

	$SO_3 + SE_3$ (ms)			UDQ (ms)		
Data set	Integration	Pred.	Tracking	Integration	Pred.	Tracking
MH01	**0.579**	0.099	30.89	0.703	**0.094**	**30.63**
MH02	**0.568**	0.091	30.55	0.702	**0.088**	**29.84**
MH03	**0.555**	0.258	29.90	0.686	**0.224**	**29.38**
MH04	**0.571**	0.250	29.79	0.718	**0.109**	**28.74**
MH05	**0.570**	0.242	29.86	0.722	**0.184**	**29.34**
V101	**0.592**	**0.138**	31.13	0.731	0.140	**30.40**
V102	**0.559**	0.182	29.03	0.697	**0.181**	**28.80**
V103	**0.551**	0.149	27.54	0.687	**0.141**	**26.60**
V201	**0.590**	0.197	30.71	0.742	**0.196**	**30.44**
V202	**0.576**	0.167	29.27	0.712	**0.165**	**28.69**
V203	**0.634**	0.144	26.53	0.803	**0.143**	**25.63**

Note: The best performance results are in bold.
Abbreviation: UDQ, unit dual quaternion.

TABLE 3 Error comparison of VINS-FUSION ($SO_3 + SE_3$) and proposed method.

	VINS-FUSION (cm)					VINS-UDQ (cm)				
Data set	RMSE	Mean	Median	std	max	RMSE	Mean	Median	Std	Max
MH01	**26.68**	**25.45**	**26.16**	**7.99**	**38.43**	30.92	29.1	31.63	10.45	55.85
MH02	**21.5**	**19.85**	**19.5**	**8.27**	**46**	29.04	27.1	27.63	10.3	61
MH03	29.27	25.98	22.24	13.48	**49.75**	**25.67**	**22.40**	**20.43**	**12.54**	59.8
MH04	42.54	39.37	37.05	16.13	70.54	**42.11**	**39.15**	**36.23**	**16.06**	**70.4**
MH05	30.47	27.93	22.97	12.05	57.93	**15.86**	**14.21**	**14.52**	**7.05**	**31.33**
V101	**11.28**	**10.53**	**9.96**	40.51	**18.87**	13.9	12.44	11.09	**6.22**	31.13
V201	**12.36**	**9.86**	**6.96**	**7.45**	**26.14**	20.86	15.73	10.27	13.7	55.5

Note: The best performance results are in bold.
Abbreviation: UDQ, unit dual quaternion.

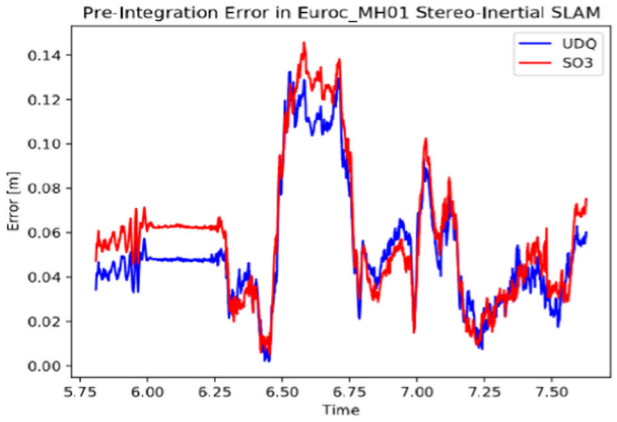

FIGURE 6 Comparison of position error in the EuRoC MAV MH01 data set. [Color figure can be viewed at wileyonlinelibrary.com]

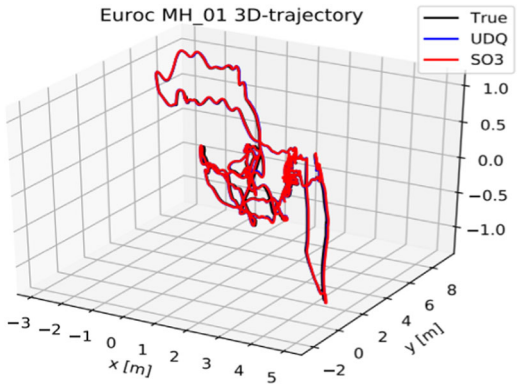

FIGURE 7 Estimated trajectory of the EuRoC MAV MH01 data set. [Color figure can be viewed at wileyonlinelibrary.com]

6 | CONCLUSIONS AND FUTURE WORK

Introducing the screw motion theory to the VI-SLAM improves the consistency and compactness of the rigid body dynamics. The memory cost is reduced and the performance is enhanced due to fewer parameters and faster convergence. This work has proved that such VI-SLAM can be futuristic and robust. Firstly, we formulated the new preintegration scheme. Unit dual quaternions are employed to construct the graphs' inertial edges and the states are updated by the screw motion framework. Secondly, the optimization problem for the SLAM back-end is assured to be singularity-free in all scenarios while maintaining all the desired advantages of such framework. Thirdly, experiments based on popular smoothing-based SLAM on EuRoC MAV datasets validate this claim. The algorithms convergence property is improved. An optimal solution is reached within 20% of the iterations (25-40 iterations suffice). Additionally, the memory requirement is reduced by 33% and the preintegration algorithm cost is approximately 25% higher due to richer kinematics. However, the tracking by the new motion model is more efficient and compensates for this increase. Lastly, we intend to incorporate this new algorithm to a project for an Unmanned Surface Vehicle (USV) that explores its environment based on the VI-SLAM. Where the Plücker line transformation is employed to reconstruct 3D scenes.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

ORCID

Nassim Bessaad http://orcid.org/0000-0002-0125-5246

REFERENCES

Adorno, B.V. (2017) Robot Kinematic Modeling and Control Based on Dual Quaternion Algebra—Part I: Fundamentals. (December).

Balmaceda-Santamaría, A.L. & García-Murillo, M.A. (2019) Kinematics of a planar parallel robot via screw theory: Details Not Mentioned." In: *IFToMM International Symposium on Robotics and Mechatronics*, pp. 91–102.

Bessaad, N., Bao, Q., Du, Y. & Liu, L. (2018) An adaptive multi-sample sins attitude algorithm. In: *2018 IEEE International Conference on Information and Automation, ICIA 2018*, pp. 888–892.

Bessaad, N., Bao, Q., Jiangkang, Z. & Eliker, K. (2021) On SINS\Star tracker geo-localization. *Engineering Research Express*, 3(4), 045040.

Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S. et al. (2016) The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10), 1157–1163.

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J. et al. (2016) Past, present, and future of simultaneous localization and mapping: toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6), 1309–1332.

Campos, C., Montiel, J.M.M. & Tardós, J.D. (2019) Fast and robust initialization for visual-inertial SLAM. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 1288–1294.

Delmerico, J. & Scaramuzza, D. (2018) A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots. In: *Proceedings—IEEE International Conference on Robotics and Automation*, pp. 2502–2509.

Eckenhoff, K., Geneva, P. & Huang, G. (2018) Continuous preintegration theory for graph-based visual-inertial navigation. *arXIv*. http://arxiv.org/abs/1805.02774.

Eckenhoff, K., Geneva, P. & Huang, G. (2019) Closed-form preintegration methods for graph-based visual–inertial navigation. *The International Journal of Robotics Research*, 38(5), 563–586.

Eckenhoff, K., Geneva, P. & Huang, G. (2020) High-accuracy preintegration for visual-inertial navigation. In: *Algorithmic Foundations of Robotics XII*. Cham: Springer, pp. 48–63.

Forster, C., Luca, C., Frank, D. & Davide, S. (2016) On-manifold preintegration for real-time visual—inertial odometry. *IEEE Transactions on Robotics*, 33(1), 1–21.

Forster, C., Carlone, L., Dellaert, F. & Scaramuzza, D. (2015) *IMU Preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation*. Georgia Institute of Technology.

Le Gentil, C. & Vidal-Calleja, T. (2021) Continuous integration over SO(3) for IMU preintegration. *Robotics: Science and Systems*, July 12.

Gouasmi, M. (2012) Robot kinematics, using dual quaternions. *IAES International Journal of Robotics and Automation (IJRA)*, 1(1), 13.

Grisetti, G., Kummerle, R., Stachniss, C. & Burgard, W. (2010) A tutorial on graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine*, 2(4), 31–43.

Huang, G. (2019) Visual-inertial navigation: a concise review. In: *Proceedings—IEEE International Conference on Robotics and Automation*. Institute of Electrical and Electronics Engineers Inc., pp. 9572–9582.

Huang, S. & Dissanayake, G. (2016) A critique of current developments in simultaneous localization and mapping. *International Journal of Advanced Robotic Systems*, 13(5), 172988141666948.

Kaess, M. (2008) *Incremental smoothing and mapping*. Georgia Tech. Available at: http://www.cs.cmu.edu/~kaess/pub/Kaess08thesis.html [Accessed March 2022].

Kim, J., Cheng, J. & Shim, H. (2015) Unit dual-quaternion parametrisation for graph SLAM. In: *Australasian Conferene on Robotics and Automation, ACRAc*, vol. 3.

Leutenegger, S., Furgale, P., Rabaud, V., Chli, M., Konolige, K. & Siegwart, R. (2013) Keyframe-based visual-inertial SLAM using nonlinear optimization. In: *Proceedings of Robotics Science and Systems (RSS)*.

Lupton, T. & Sukkarieh, S. (2012) Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. *IEEE Transactions on Robotics*, 28(1), 61–76.

Mur-Artal, R. & Tardos, J.D. (2017) ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5), 1255–1262.

Qin, T., Li, P. & Shen, S. (2018) VINS-Mono: a robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4), 1004–1020.

Thrun, W., Burgard, D. & Fox, S. (2006) *Probabilistic robotics*.

Triggs, B., Andrew, Z. & Richard, S. (Eds.) (2000) Vision algorithms: theory and practice. In: *International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999*. Berlin Heidelberg: Springer. Available from: https://doi.org/10.1007/978-3-642-29066-4_11

Yuanxin, W, Xiaoping, H.u, Dewen, H., Tao, L.i & Junxiang, L. (2005) Strapdown inertial navigation system algorithms based on dual quaternions. *IEEE Transactions on Aerospace and Electronic Systems*, 41(1), 110–132.

Zhao, Z., Song T., Xing B., Lei Y. & Wang Z. (2022) PLI-VINS: visual-inertial SLAM based on point-line feature fusion in indoor environment. *Sensors* 22, 5457(14).

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Bessaad, N., Qilian, B., Jiankang, Z., Benoudina, N., Sun, S. & Zhang, X. (2023) Inertial preintegration for VI-SLAM by the screw motion theory. *Journal of Field Robotics*, 1–13. https://doi.org/10.1002/rob.22212