The International Journal of
Robotics Research
30(1) 56–79
© The Author(s) 2011
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/0278364910382802
ijr.sagepub.com
SAGE

Visual–Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Self-calibration

Jonathan Kelly and Gaurav S Sukhatme

Abstract

Visual and inertial sensors, in combination, are able to provide accurate motion estimates and are well suited for use in many robot navigation tasks. However, correct data fusion, and hence overall performance, depends on careful calibration of the rigid body transform between the sensors. Obtaining this calibration information is typically difficult and time-consuming, and normally requires additional equipment. In this paper we describe an algorithm, based on the unscented Kalman filter, for self-calibration of the transform between a camera and an inertial measurement unit (IMU). Our formulation rests on a differential geometric analysis of the observability of the camera–IMU system; this analysis shows that the sensor-to-sensor transform, the IMU gyroscope and accelerometer biases, the local gravity vector, and the metric scene structure can be recovered from camera and IMU measurements alone. While calibrating the transform we simultaneously localize the IMU and build a map of the surroundings, all without additional hardware or prior knowledge about the environment in which a robot is operating. We present results from simulation studies and from experiments with a monocular camera and a low-cost IMU, which demonstrate accurate estimation of both the calibration parameters and the local scene structure.

Keywords

Self-calibration, sensor fusion, observability, computer vision, inertial navigation

1. Introduction

The majority of future robots will be mobile, and will need to navigate in dynamic and unknown environments. Inertial sensors have been used very successfully over the past several decades for both terrestrial (Farrell and Barth 1998) and space navigation (Woods 2008). The standard inertial sensing device, an *inertial measurement unit* (IMU), employs an orthogonal triad of single-axis accelerometers and an orthogonal triad of angular rate gyroscopes to measure rigid body motion. Advances in the fabrication of microelectromechanical systems (MEMS) have led to the development of reliable, solid-state IMUs (Titterton and Weston 2004); the small size and low power consumption of these MEMS-based instruments makes them ideal for many robotics applications.

Solid-state IMUs are designed for use in a *strapdown* configuration, with the accelerometers and gyroscopes attached to a common chassis and not actively gimbaled to maintain a fixed orientation. The change in pose of a strapdown IMU can, in principle, be determined by integrating the gyroscope and accelerometer outputs. In practice, all inertial sensors are subject to low-frequency drift, which limits the accuracy of inertial dead-reckoning over extended periods. The existence of varying drift terms (biases) also implies that IMU measurements are correlated in time. Further, IMU accelerometers sense the force of gravity in addition to forces that accelerate the platform. The magnitude of the gravity vector (nominally 9.81 m s^{-2}) is often large enough to dominate other measured accelerations. If the orientation of the IMU with respect to gravity is unknown, or is misestimated, the integrated sensor pose will diverge rapidly from the true pose.

To maintain the long-term integrity of inertial positioning information, some form of *aiding* is required (Farrell 2008). Typically, aiding updates are provided by GPS/GNSS sensors; however, GPS has limited accuracy, suffers from problems with multipath interference, and cannot be used in many environments of interest to roboticists, e.g. indoors, underwater, or on remote planetary surfaces. An alternative is *visual aiding*, in which pose corrections are derived from observations made by one or more cameras. The complementary frequency response and noise characteristics of

Robotic Embedded Systems Laboratory, University of Southern California, Los Angeles, USA

Corresponding author:
Jonathan Kelly
Robotic Embedded Systems Laboratory, University of Southern California, Los Angeles, CA 90089-0781, USA
Email: jonathsk@usc.edu

cameras and IMUs make the sensors suitable for use in combination to estimate ego-motion with high fidelity and at a high update rate (Strelow and Singh 2003; Mourikis and Roumeliotis 2007; Kelly et al. 2008).

In order to properly fuse visual and inertial measurements in a single navigation frame, precise calibration of the six-degree-of-freedom (6-DOF) transform(s) between the camera(s) and the IMU is required. Incorrect calibration will introduce a systematic bias in motion estimates, degrading overall navigation performance. The calibration process is usually complex and time consuming, however, and must be repeated whenever the sensors are repositioned or significant mechanical stresses are applied. This is inconvenient, and may be impossible in some cases. Ideally, we would like to build "power-on-and-go" robots that are able to operate autonomously for long periods without requiring tedious manual (re-)calibration.

In this paper, we describe our work on combining visual and inertial sensing for navigation, with an emphasis on the ability to self-calibrate the camera–IMU transform *in the field*. Self-calibration refers to the use of measurements (exclusively) from the sensors themselves to improve our estimates of related system parameters. As an exteroceptive sensor, the camera must view a set of static visual landmarks during calibration. Without additional information, the three-dimensional positions of the landmarks will initially be unknown and so must also be estimated, i.e. we must localize the camera–IMU platform, build a map of the landmarks, and determine the calibration parameters. As a first step herein, we perform an analysis of the *observability* of self-calibration. The analysis shows that the 6-DOF camera–IMU transform, IMU biases, gravity vector, and the metric scene structure are all observable and can be estimated simultaneously. To the best of the authors' knowledge, this result is more general than those reported previously in the literature. Full observability requires the camera–IMU platform to undergo rotation about at least two IMU axes and acceleration along two IMU axes.

Based on the observability analysis, we develop a practical algorithm for camera–IMU sensor-to-sensor self-calibration[1]. We formulate this task as a filtering problem, and estimate the transform between the sensors using an unscented Kalman filter (UKF). Our approach is designed to enable online, anytime calibration updates, potentially during an ongoing navigation or mapping task.[2] The calibration algorithm provides a measure of the uncertainty associated with the transform (i.e. a covariance matrix), and can easily be integrated with other estimators.

Initially, we consider *target-based calibration*, in which the camera views a known calibration target or object. We then extend the filtering algorithm to handle *target-free self-calibration*, where the positions of the landmarks corresponding to image features are not *a priori* known. We present results from simulation studies and from experiments with real hardware, which demonstrate that accurate estimates of the calibration parameters and the metric scene structure can be obtained using a monocular camera and an inexpensive solid-state IMU, *without* extra apparatus.

The remainder of the paper is organized as follows. We examine related research in Section 2. In Section 3 we review the notation used in the paper, and briefly discuss quaternions and their algebra. In Section 4 we describe the camera–IMU system model. Section 5 presents our observability analysis; we then develop our UKF-based estimator in Section 6. Results from simulation studies and from experiments with a low-cost IMU are given in Sections 7 and 8, respectively. Finally, we offer some conclusions and several directions for future work in Section 9.

2. Related Research

There are very substantial bodies of independent research on both visual and inertial navigation. Of particular relevance to our work is the recent demonstration of real-time localization and mapping based solely on the input from a monocular camera. For example, Davison et al. (2007) describe an extended Kalman filter (EKF)-based system that is able to localize a camera in a room-sized environment. To scale to larger scenes, Eade and Drummond (2006) use a fast particle filtering algorithm, which takes advantage of the independence of landmark position estimates given a known camera trajectory. Klein and Murray (2007) propose *parallel tracking and mapping*, where localization and mapping are run as separate, asynchronous processes to improve performance. We note that vision-only techniques must make assumptions about camera motion, however, and can have difficultly recovering when all tracked features are lost. In addition, to determine the absolute scene scale, monocular algorithms require initialization using a known reference object. Our approach makes use of IMU data to determine absolute scale and to fill-in during periods when visual tracking fails.

A variety of hybrid visual–inertial systems have been developed, for applications from short-range sensor localization (Strelow and Singh 2002) to long-range motion estimation for spacecraft entry, descent and landing (Mourikis et al. 2009). Rehbinder and Ghosh (2003) demonstrate accurate motion and structure estimation, using a combination of IMU measurements and camera observations of line features in the environment. Gemeiner et al. (2007) fuse IMU and camera data in an EKF for motion estimation, and use an additional bank of EKFs to estimate the positions of a series of landmarks. More broadly, improvements in available computing power have recently enabled real-time visual–inertial navigation (Strelow 2004; Mourikis and Roumeliotis 2007). While all of these algorithms estimate motion and structure, they do not incorporate calibration, which is a primary focus of our work.

Several visual–inertial calibration techniques have also been presented in the literature. Lang and Pinz (2005) use a constrained non-linear optimization algorithm to solve for the fixed rotation between a camera and an IMU. By comparing camera measurements of the relative angles to several external markers with integrated gyroscope outputs, the optimization algorithm determines the rotation which best aligns the sensor frames.

Lobo and Dias (2007) describe a camera–IMU calibration procedure in which the relative orientation and the relative translation of the sensors are estimated separately. First, the relative orientation is found by rotating the camera–IMU platform while the camera captures images of a vertical planar target. The relative translation is then determined by spinning the camera and the IMU on a turntable, positioning the IMU such that its measured horizontal acceleration is zero. A drawback of the method is that separate calibration of orientation and translation ignores any correlations that exist between the parameters, and hence does not properly account for error propagation.

More closely related to our own work is the algorithm proposed by Mirzaei and Roumeliotis (2008) for calibrating the camera–IMU transform and simultaneously estimating the IMU biases. Their algorithm uses an iterated EKF to fuse IMU data with camera measurements of known corner points on a planar calibration target. A similar algorithm for calibrating the relative pose of a spherical camera and an IMU is discussed in Hol et al. (2008). Our approach, in the spirit of Foxlin (2002) and unlike those above, does not require *any* additional equipment in the general case.

Jones et al. (2007) present an observability analysis which shows that the camera–IMU relative pose, gravity vector, and metric scene structure can be determined using camera and IMU measurements only. Their analysis assumes that the IMU biases are known and static over the calibration interval, although bias drift may be significant even for short durations – particularly for the low-cost MEMS inertial sensors that we consider in this paper. We account for and explicitly model the IMU biases. This is critical for longer motion sequences, or when estimates of the biases are not available from other sources.

3. Preliminaries

In the following, we introduce the notation used throughout the remainder of the paper. We also briefly review quaternions, which we employ as a singularity-free orientation parameterization.

3.1. Notation

Our notation follows the general conventions outlined in Britting (1971). We denote vectors and matrices in boldface, and indicate that a vector is expressed with respect to a specific reference frame by appending a superscript identifying the frame, e.g. $\mathbf{v}^A$ for the vector $\mathbf{v}$ expressed in frame $\{A\}$. If a vector describes the relative translation of one reference frame with respect to another frame, we append a subscript identifying the translated frame and a superscript identifying the base frame, e.g. $\mathbf{p}_B^A$ for the vector that defines the translation of frame $\{B\}$ in frame $\{A\}$.

In our analysis, we make frequent use of both the identity matrix, $\mathbf{I}$, and the zero matrix, $\mathbf{0}$. We indicate the sizes of these matrices using subscripts, e.g. $\mathbf{I}_3$ for the 3×3 identity matrix, and $\mathbf{0}_{4\times3}$ for the 4×3 matrix that contains zeros only.

Finally, for the 3×1 vector $\mathbf{a} = [a_x \quad a_y \quad a_z]^{\mathrm{T}}$, we write $[\mathbf{a}\times]$ to denote the 3×3 skew-symmetric cross-product matrix

$$[\mathbf{a}\times] = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}, \tag{1}$$

such that for another 3×1 vector $\mathbf{b}$, $\mathbf{a} \times \mathbf{b} \triangleq [\mathbf{a}\times]\,\mathbf{b}$.

3.2. Quaternions

We use quaternions to parameterize the rotation group SO(3). A quaternion is a four-component hyper-complex number, consisting of both a *scalar part* q_0 and a *vector part* $\mathbf{q}$,

$$q = q_0 + \mathbf{q} = q_0 + q_1\mathbf{i} + q_2\,\mathbf{j} + q_3\mathbf{k}, \tag{2}$$

where q_0, q_1, q_2, and q_3 are real numbers, and $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$ are the quaternion basis vectors, which satisfy

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{i}\,\mathbf{j}\,\mathbf{k} = -1. \tag{3}$$

Quaternion addition is performed by summing the respective components of each quaternion separately. With use of (3), the product of two quaternions, $p = p_0 + \mathbf{p}$ and $q = q_0 + \mathbf{q}$, is

$$\begin{aligned} p \otimes q &= (p_0 + \mathbf{p}) \otimes (q_0 + \mathbf{q}) \\ &= p_0q_0 - \mathbf{p}^{\mathrm{T}}\mathbf{q} + p_0\,\mathbf{q} + q_0\,\mathbf{p} + \mathbf{p} \times \mathbf{q}, \end{aligned} \tag{4}$$

where the symbol '$\otimes$' denotes quaternion multiplication. It will often be convenient to express (4) as the product of a 4×4 matrix and a quaternion (represented as a four-vector). For two quaternions, p and q, let[3]

$$p \otimes q \triangleq \mathbf{S}(p)\; q = \begin{bmatrix} p_0 & -\mathbf{p}^{\mathrm{T}} \\ \mathbf{p} & p_0\,\mathbf{I}_3 + [\mathbf{p}\times] \end{bmatrix} \begin{bmatrix} q_0 \\ \mathbf{q} \end{bmatrix} \tag{5}$$

and

$$p \otimes q \triangleq \mathbf{D}(q)\; p = \begin{bmatrix} q_0 & -\mathbf{q}^{\mathrm{T}} \\ \mathbf{q} & q_0\,\mathbf{I}_3 - [\mathbf{q}\times] \end{bmatrix} \begin{bmatrix} p_0 \\ \mathbf{p} \end{bmatrix}. \tag{6}$$

The *conjugate* of the quaternion q is

$$q^* = q_0 - \mathbf{q} = q_0 - q_1\mathbf{i} - q_2\,\mathbf{j} - q_3\mathbf{k}, \tag{7}$$

and the *norm* of q is defined as the square root of the product of the quaternion and its conjugate,

$$\|\,q\| = \sqrt{q \otimes q^*} = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}. \tag{8}$$

The *inverse* of q is

$$q^{-1} = \frac{q^*}{\|\,q\|^2}. \tag{9}$$

Any rotation on SO(3) can be represented by a unit quaternion (Kuipers 2002), i.e. by a quaternion with unit norm

$$\|\,\bar{q}\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1, \tag{10}$$

where we use an overbar to indicate that $\bar{q}$ has norm one. The quaternion $\bar{q}$ represents a rotation by the angle θ about an axis defined by unit vector $\bar{\mathbf{a}} \in \mathbb{R}^3$,

$$\bar{q} = \left[\cos(\theta/2) \quad \bar{\mathbf{a}}^{\mathrm{T}} \sin(\theta/2)\right]^{\mathrm{T}}. \tag{11}$$

The direction cosine (rotation) matrix corresponding to $\bar{q}$ is

$$\mathbf{C}(\bar{q}) = (2q_0^2 - 1)\,\mathbf{I}_3 + 2\,\mathbf{q}\,\mathbf{q}^{\mathrm{T}} - 2\,q_0\,[\mathbf{q}\times]. \tag{12}$$

Following our notation for translation vectors, we denote the unit quaternion that describes the orientation of frame $\{B\}$ with respect to frame $\{A\}$ as[4] $\bar{q}^{A}_{B}$.

Unit quaternions have several advantages over other orientation parameterizations, e.g. the mapping from the unit sphere S^3 in $\mathbb{R}^4$ to SO(3) is smooth and singularity-free. Quaternions are also less susceptible than rotation matrices to round-off errors (Chou 1992). We note that unit quaternions are a non-minimal parameterization, however, with four components but only three DOFs. This constraint requires special treatment in our estimation algorithm (cf. Section 6.2).

4. System Modeling

The goal of calibration is to determine the 6-DOF rigid body transform between the camera and the IMU. Our approach formulates this task as a causal filtering problem, in which we refine an initial estimate of the calibration parameters using observations acquired sequentially over time. In addition, if the environment is unknown, we also recover the local scene structure. We consider three separate reference frames:

1. the *IMU frame* $\{I\}$, with its origin at the center of the IMU body, in which linear accelerations and angular rates are measured;
2. the *camera frame* $\{C\}$, with its origin at the optical center of the camera and with the z-axis aligned with the optical axis of the lens; and
3. the *world frame* $\{W\}$, an inertial frame that serves as an absolute reference for both the camera and the IMU[5].

As a first step, we must choose an origin for the world frame. When a calibration target is available, we will select one of the corner points on the target as the origin of the world frame. For target-free self-calibration, we consider the initial camera position as the origin of the world frame (cf. Section 5.2.1). Figure 1 illustrates the relationship between the reference frames, for target-based calibration.

4.1. System Parameterization

Target-based calibration involves sensor-related quantities only. Our 26 × 1 *sensor* state vector is

$$\mathbf{x}_s(t) = [(\mathbf{p}^W_I(t))^{\mathrm{T}}\,(\bar{q}^{\,W}_I(t))^{\mathrm{T}}\,(\mathbf{v}^W(t))^{\mathrm{T}}\,(\mathbf{b}_g(t))^{\mathrm{T}} \\ (\mathbf{b}_a(t))^{\mathrm{T}}\,(\mathbf{g}^W)^{\mathrm{T}}\,(\mathbf{p}^I_C)^{\mathrm{T}}\,(\bar{q}^{\,I}_C)^{\mathrm{T}}\,]^{\mathrm{T}}, \tag{13}$$

where $\mathbf{p}^W_I(t)$ is the position of the IMU in the world frame, $\bar{q}^{\,W}_I(t)$ is the (unit quaternion) orientation of the IMU frame with respect to the world frame, $\mathbf{v}^W(t)$ is the linear velocity of the IMU in the world frame, $\mathbf{b}_g(t)$ and $\mathbf{b}_a(t)$ are the IMU gyroscope and accelerometer biases, respectively, and $\mathbf{g}^W$ is the gravity vector in the world frame. The remaining entries, $\mathbf{p}^I_C$ and $\bar{q}^{\,I}_C$, are the position of the camera in the IMU frame and the (unit quaternion) orientation of the camera frame with respect to the IMU frame. Note that the last three terms in the sensor state vector are *parameters*, i.e. quantities that are not time varying. For brevity, we do not indicate dependence on time in the sections that follow.

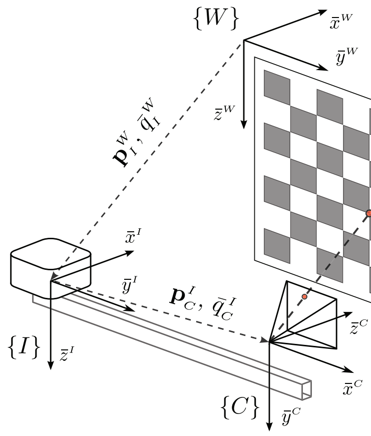


Fig. 1. Relationship between the world $\{W\}$, IMU $\{I\}$, and camera $\{C\}$ reference frames, for target-based calibration. The IMU (left) and camera (right) are rigidly attached to a common chassis. Our goal is to determine the transform $(\mathbf{p}^I_C, \bar{q}^{\,I}_C)$ between the camera and IMU.

When performing target-free self-calibration, we also estimate the positions of a series of static point landmarks in the environment. The complete target-free state vector is

$$\mathbf{x}(t) = \left[\mathbf{x}^{\mathrm{T}}_s(t) \quad \mathbf{x}^{\mathrm{T}}_m\right]^{\mathrm{T}}, \mathbf{x}_m = \left[(\mathbf{p}^W_{l_1})^{\mathrm{T}} \quad \dots \quad (\mathbf{p}^W_{l_n})^{\mathrm{T}}\right]^{\mathrm{T}}, \tag{14}$$

where $\mathbf{x}_m$ is the *map* (structure) state vector. Each entry in $\mathbf{x}_m$ is a 3 × 1 vector, $\mathbf{p}^W_{l_i}$, that defines the position of landmark i in the world frame, $i = 1 \dots n$, for $n \geq 3$. The target-free state vector has size $(26 + 3n) \times 1$. In our laboratory calibration experiments, we have found it sufficient to use Cartesian coordinates to specify landmark positions. If the true landmark depths relative to the camera vary significantly, it may be more appropriate to use an inverse-depth parameterization (Montiel et al. 2006). We discuss this issue in more detail in Section 8.

We define the axes of the world frame with respect to either the calibration target or the first camera pose. As such, the relationship between the gravity vector and the world frame depends entirely on how the target or the camera is oriented. For target-based calibration, we can manually align the vertical axis of the target with gravity: this alignment will not in general be exact, however. Similarly, for target-free self-calibration, the vertical axis of the camera image plane is unlikely to be exactly parallel to the gravity vector initially. To avoid introducing unmodeled biases in the calibration solution, we therefore also estimate the components of the local gravity vector in the world frame.

4.2. Process Model

Our process model uses the IMU angular velocity and linear acceleration measurements as control inputs in the system dynamics equations (Chatfield 1997). The gyroscope and accelerometer biases are modeled as Gaussian random walk processes, driven by the white, zero-mean noise vectors $\mathbf{n}_{gw}$ and $\mathbf{n}_{aw}$, with covariance matrices $\mathbf{Q}_{gw}$ and $\mathbf{Q}_{aw}$, respectively. Gyroscope and accelerometer measurements are assumed to be corrupted by zero-mean Gaussian noise defined by vectors $\mathbf{n}_g$ and $\mathbf{n}_a$, with covariance matrices $\mathbf{Q}_g$ and $\mathbf{Q}_a$, respectively. The system state evolves in continuous time according to

$$\dot{\mathbf{p}}_I^W = \mathbf{v}^W, \quad \dot{\bar{q}}_I^W = \frac{1}{2}\Omega(\boldsymbol{\omega}^I)\,\bar{q}_I^W, \quad \dot{\mathbf{v}}^W = \mathbf{a}^W, \quad \dot{\mathbf{g}}^W = \mathbf{0}_{3\times1}, \tag{15}$$

$$\dot{\mathbf{b}}_g = \mathbf{n}_{gw}, \quad \dot{\mathbf{b}}_a = \mathbf{n}_{aw}, \quad \dot{\mathbf{p}}_C^I = \mathbf{0}_{3\times1}, \quad \dot{\bar{q}}_C^I = \mathbf{0}_{4\times1}, \tag{16}$$

where $\Omega(\boldsymbol{\omega}^I)$ is the quaternion kinematic matrix (Stevens and Lewis 2003),

$$\Omega(\boldsymbol{\omega}^I) = \begin{bmatrix} 0 & -\left(\boldsymbol{\omega}^I\right)^{\mathrm{T}} \\ \boldsymbol{\omega}^I & -\left[\boldsymbol{\omega}^I\times\right] \end{bmatrix}, \tag{17}$$

which relates the time rate of change of the orientation quaternion to the IMU angular velocity, $\boldsymbol{\omega}^I$ is the angular velocity of the IMU expressed in the IMU frame, and $\mathbf{a}^W$ is the acceleration of the IMU expressed in the world frame. The *measured* IMU angular velocity and linear acceleration are

$$\boldsymbol{\omega}_m = \boldsymbol{\omega}^I + \mathbf{b}_g + \mathbf{n}_g, \tag{18}$$

$$\mathbf{a}_m = \mathbf{C}^{\mathrm{T}}(\bar{q}_I^W)\left(\mathbf{a}^W - \mathbf{g}^W\right) + \mathbf{b}_a + \mathbf{n}_a, \tag{19}$$

where $\mathbf{C}(\bar{q}_I^W)$ is the direction cosine matrix corresponding to the unit quaternion $\bar{q}_I^W$. We stack the individual noise term to form the complete process noise vector $\mathbf{n}$ and the associated block-diagonal process noise covariance matrix $\mathbf{Q}$,

$$\mathbf{n} = \begin{bmatrix} \mathbf{n}_{gw} \\ \mathbf{n}_{aw} \\ \mathbf{n}_g \\ \mathbf{n}_a \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} \mathbf{Q}_{gw} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{Q}_{aw} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{Q}_g & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{Q}_a \end{bmatrix}. \tag{20}$$

Also, we assume that any landmarks used for target-free calibration remain stationary over time,

$$\dot{\mathbf{p}}_{l_i}^W = \mathbf{0}_{3\times1}, \quad i = 1 \ldots n. \tag{21}$$

4.3. Measurement Model

We use an ideal projective (pinhole) camera model, and rectify each camera image to remove lens distortions. The camera intrinsic and distortion parameters may either be calibrated separately beforehand, or calibrated using a subset of the images acquired for the target-based camera–IMU procedure (if the procedure is run offline). Self-calibration of the camera is also possible, although this is beyond the scope of the work presented here.

Measurement $\mathbf{z}_i$ is the projection of landmark i, at position $\mathbf{p}_{l_i}^C = [x_i \;\; y_i \;\; z_i]^{\mathrm{T}}$ in the camera frame, onto the image plane,

$$\mathbf{p}_{l_i}^C = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \mathbf{C}^{\mathrm{T}}(\bar{q}_C^I)\,\mathbf{C}^{\mathrm{T}}(\bar{q}_I^W)\left(\mathbf{p}_{l_i}^W - \mathbf{p}_I^W\right) - \mathbf{C}^{\mathrm{T}}(\bar{q}_C^I)\,\mathbf{p}_C^I\,, \tag{22}$$

$$\mathbf{z}_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} x_i' \\ y_i' \end{bmatrix} + \boldsymbol{\eta}_i, \quad \begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} = \mathcal{K}\begin{bmatrix} x_i/z_i \\ y_i/z_i \\ 1 \end{bmatrix}, \tag{23}$$

where x_i/z_i and y_i/z_i are the normalized image plane coordinates, u_i and v_i are the observed pixel coordinates, $\mathcal{K}$ is the 3 × 3 camera intrinsic calibration matrix (Ma et al. 2004), and $\boldsymbol{\eta}_i$ is a Gaussian measurement noise vector with covariance matrix $\mathbf{R}_i = \sigma_i^2\,\mathbf{I}_2$. Our measurement residuals consist of the difference between the observed coordinates of the projected landmark points and their predicted coordinates, based on the current state estimate.

When several landmarks are visible in one image, we stack the individual measurements to form a single measurement vector $\mathbf{z} = \left[\mathbf{z}_1^{\mathrm{T}} \;\; \ldots \;\; \mathbf{z}_n^{\mathrm{T}}\right]^{\mathrm{T}}$ and the associated block-diagonal covariance matrix $\mathbf{R} = \mathrm{diag}(\mathbf{R}_1 \ldots \mathbf{R}_n)$. This vector can then be processed by our filtering algorithm in one step.

5. Non-linear Observability Analysis

In order to calibrate the transform between the camera and the IMU, the relevant system states must be *observable*. That is, we must be able to recover the state values from the measured system outputs, the known control inputs, and a finite number of their time derivatives (Conte et al. 2006). Observability is a necessary condition for any filtering algorithm to converge to an unbiased estimate of the true system state (Lee et al. 1982).

For linear time-invariant systems, observability can be determined using any of several straightforward tests, such as the well-known linear observability matrix rank test (Maybeck 1979) or the Popov–Belevitch–Hautus eigenvector test (Verhaegen and Verdult 2007). Establishing the observability of linear time-varying systems generally involves evaluating the observability Gramian matrix, which must usually be done numerically. An alternative, for time-varying systems that can be approximated as piecewise constant, is to use the stripped observability matrix (Goshen-Meskin and Bar-Itzhack 1992). Analyzing the observability of non-linear systems is significantly more difficult.

In Section 5.2, we present a fully non-linear analysis of the observability of target-free camera–IMU self-calibration. The analysis is based on a differential geometric characterization of observability, and relies on a matrix rank test originally introduced by Hermann and Krener (1977). The method has been used to analyze the observability of a series of robotic calibration problems, including camera extrinsic parameter calibration (Martinelli et al. 2006a) and online odometry self-calibration with an exteroceptive sensor (Martinelli et al. 2006b; Martinelli and Siegwart 2006).

5.1. Differential Geometry and Non-linear Observability

Our analysis follows the differential geometric approach described by Hermann and Krener (1977). We treat the state space as a smooth manifold M of dimension m, and consider the non-linear system

$$\mathcal{S}\begin{cases}\dot{\mathbf{x}} = \mathbf{f}_0(\mathbf{x}) + \sum_{i=1}^{p} \mathbf{f}_i(\mathbf{x})\, u_i \\ \mathbf{y} = \mathbf{h}(\mathbf{x})\end{cases}, \tag{24}$$

where $\mathbf{x} \in M \subseteq \mathbb{R}^m$ is the state vector, $\mathbf{y} \in \mathbb{R}^n$ is the vector of observed system outputs, and $\mathbf{u} \in \mathbb{R}^p$, $\mathbf{u} = [u_1 \ldots u_p]^{\mathrm{T}}$ is the vector of control inputs. Systems of the form defined by (24) are called *input-linear* or *control-affine* systems (Sastry 1999), as each $\mathbf{f}_i(\mathbf{x})$, $i = 1 \ldots p$, defines a vector field on M that is a linear function of a single control input. Together with the drift vector field $\mathbf{f}_0(\mathbf{x})$, the fields $\mathbf{f}_i(\mathbf{x})$, $i = 1 \ldots p$, and the control vector $\mathbf{u}$ determine the evolution of the system state over time.

A system is observable, in a global sense, if there are no two points $\mathbf{x}_0, \mathbf{x}_1 \in M$ that are *indistinguishable*, or that share the same input–output map for all possible control inputs. It may be necessary to move a significant distance on M, or for a significant amount of time, however, in order to distinguish between the two points. We will instead determine whether our camera–IMU system is *locally weakly observable*, i.e. if, for each point $\mathbf{x}_0 \in M$, there exists an open neighborhood $U \subseteq M$ such that for every open neighborhood V of $\mathbf{x}_0$, $V \subseteq U$, there is no other state $\mathbf{x}_1 \in V$ that is indistinguishable from $\mathbf{x}_0$ (Hermann and Krener 1977). Intuitively, this definition expresses the fact that it is possible to instantaneously distinguish one point in the state space from its neighbors.[6] In turn, if a system is locally weakly observable, the system state can be estimated from the measured outputs and the known control inputs (Conte et al. 2006). Note that the definition of local weak observability does *not* imply that every set of control inputs distinguishes between states.

We now develop the formalism required for our analysis. Given the smooth vector field $\mathbf{f} : M \to TM$, where TM is the tangent bundle[7] of M, and a smooth scalar function $h : M \to \mathbb{R}$, the *Lie derivative* of h with respect to $\mathbf{f}$ at $\mathbf{x} \in M$ is

$$L_{\mathbf{f}} h(\mathbf{x}) = \nabla_{\mathbf{f}} h(\mathbf{x}) = \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}), \tag{25}$$

where $\partial h(\mathbf{x})/\partial \mathbf{x}$ is a row vector. The Lie derivative is the directional derivative of h along $\mathbf{f}$, evaluated at $\mathbf{x}$. Lie differentiation can be defined recursively, so that if we, for example, differentiate along vector field $\mathbf{f}$ and then along vector field $\mathbf{g}$, we obtain

$$L_{\mathbf{g}} L_{\mathbf{f}} h(\mathbf{x}) = \frac{\partial\, L_{\mathbf{f}} h(\mathbf{x})}{\partial \mathbf{x}} \mathbf{g}(\mathbf{x}). \tag{26}$$

Note that the Lie derivative is not commutative in general. If the function h is differentiated k times along the vector field $\mathbf{f}$, we use the notation $L_{\mathbf{f}}^k h(\mathbf{x})$, according to the recurrence relation

$$L_{\mathbf{f}}^k h(\mathbf{x}) = \frac{\partial \left(L_{\mathbf{f}}^{k-1} h(\mathbf{x})\right)}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}). \tag{27}$$

By definition, $L^0 h(\mathbf{x})$ is simply the function h itself,

$$L^0 h(\mathbf{x}) \triangleq h(\mathbf{x}). \tag{28}$$

For the column observation vector $\mathbf{h}$ of size n, we will stack the individual row Lie derivatives of h_j, $j = 1 \ldots n$ to form an $n \times m$ matrix.

Now, consider the system $\mathcal{S}$ above, and let $\mathcal{O}$ be the *observability matrix* for $\mathcal{S}$, whose rows are formed from the gradients of the Lie derivatives of $\mathbf{h}(\mathbf{x})$. The system $\mathcal{S}$ is locally weakly observable at $\mathbf{x}_0$ if $\mathcal{O}$ has full column rank at $\mathbf{x}_0$; we say that $\mathcal{S}$ satisfies the *observability rank condition at* $\mathbf{x}_0$ (Hermann and Krener 1977, Theorem 3.1). If $\mathcal{O}$ has full column rank for all $\mathbf{x} \in M$, then $\mathcal{S}$ satisfies the *observability rank condition* generically and is *locally weakly observable* (Isidori 1995). Note that the matrix $\mathcal{O}$ may in general have an infinite number of rows because M is (infinitely) smooth, however it is sufficient to show there are m rows which are always linearly independent.

5.2. On the Observability of Structure, Motion and Sensor Relative Pose

We now show that the target-free camera–IMU system is locally weakly observable. In Section 5.2.3 we describe a number of simplifications that can be made to the analysis when a known calibration target is available. Prior work on the observability of camera–IMU relative pose calibration includes Jones et al. (2007) and Mirzaei and Roumeliotis (2008).

5.2.1. Structure and Motion The problem of estimating camera motion and scene structure has been studied extensively in computer vision (as structure from motion [SFM]) and in robotics (as visual simultaneous localization and mapping [SLAM]). An analysis presented in Chiuso et al. (2002) shows that monocular SFM is observable up to an unknown *similarity transform*, i.e. an arbitrary rigid motion and a scaling. The reasons for this are twofold: first, a camera is a bearing-only sensor, which cannot provide absolute information about the depths of landmark points; parallax provides a baseline-dependent measure of distance, but the baseline will be unknown unless this information is provided by another sensor. Second, all of the camera observations are *relative*: the landmark positions are unknown, and the camera position is also unknown, so there is no fixed, absolute reference. As noted by Jones et al. (2007), simply initializing the first camera pose with zero uncertainty is not sufficient to ensure full observability.

Following Chiuso et al. (2002), it *is* possible to make monocular SFM observable by fixing the directions to three non-collinear features in the first camera image. We achieve this through a pseudo-measurement technique, in which the uncertainty associated with each of the three features (which we call *anchors*) is artificially reduced. As a result, only scale remains indistinguishable. The scale can be selected arbitrarily, by choosing the depth of one of the features.

We prove in the following that, if we anchor the world frame in this way, it is possible to simultaneously observe the pose of the IMU, the camera–IMU transform, the gyroscope and accelerometer biases, the gravity vector, *and* the local scene structure. This is the basis for the self-calibration algorithm we propose. Information about absolute depth is provided by the integrated IMU accelerometer measurements. The result holds as long as the same set of anchor features remains visible throughout the calibration process. Fixing the initial camera pose can introduce a small bias in the structure measurements: by averaging several observations at the start of calibration, we have found that it is possible to make this bias negligible.

5.2.2. Observability of Self-calibration We assume that visual measurements are provided by an independent SFM algorithm, and that the world frame has been anchored relative to three point features. Given these assumptions, monocular SFM is able to provide the camera pose in the world frame up to an unknown scale. Further, if the camera pose in the world frame can be recovered, it is immediately possible to determine the landmark positions also (by, for example, triangulation). For the observability analysis, we therefore define an expanded 27 × 1 state vector for the continuous-time system model,

$$\begin{aligned}\mathbf{x} = [(\mathbf{p}_I^W)^{\mathrm{T}}\,(\bar{q}_I^W)^{\mathrm{T}}\,(\mathbf{v}^W)^{\mathrm{T}}\,(\mathbf{b}_g)^{\mathrm{T}}\,(\mathbf{b}_a)^{\mathrm{T}}\\ (\mathbf{g}^W)^{\mathrm{T}}\,(\mathbf{p}_C^I)^{\mathrm{T}}\,(\bar{q}_C^I)^{\mathrm{T}}\,\alpha]^{\mathrm{T}}.\end{aligned} \tag{29}$$

The first 26 states of the vector above are identical to those described in Section 4.1, and the final state, α, is the scale factor that appears in the measurement model. We assume that the scene scale does not vary with time, i.e. the time derivative of the scale factor is zero,

$$\dot{\alpha} = 0. \tag{30}$$

Rearranging (15)–(16) from Section 4.2, and incorporating (30), the system process model is described by the matrix differential equation

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{p}}_I^W \\ \dot{\bar{q}}_I^W \\ \dot{\mathbf{v}}^W \\ \dot{\mathbf{b}}_g \\ \dot{\mathbf{b}}_a \\ \dot{\mathbf{g}}^W \\ \dot{\mathbf{p}}_C^I \\ \dot{\bar{q}}_C^I \\ \dot{\alpha} \end{bmatrix} = \overbrace{\begin{bmatrix} \mathbf{v}^W \\ -\frac{1}{2}\Xi\left(\bar{q}_I^W\right)\mathbf{b}_g \\ \mathbf{g}^W - \mathbf{C}(\bar{q}_I^W)\,\mathbf{b}_a \\ \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} \\ \mathbf{0}_{4\times1} \\ 0 \end{bmatrix}}^{\mathbf{f}_0}$$

$$+ \overbrace{\begin{bmatrix} \mathbf{0}_{3\times3} \\ \frac{1}{2}\Xi\left(\bar{q}_I^W\right) \\ \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} \\ \mathbf{0}_{4\times3} \\ \mathbf{0}_{1\times3} \end{bmatrix}}^{\breve{\mathbf{f}}_1} \boldsymbol{\omega}_m + \overbrace{\begin{bmatrix} \mathbf{0}_{3\times3} \\ \mathbf{0}_{4\times3} \\ \mathbf{C}(\bar{q}_I^W) \\ \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} \\ \mathbf{0}_{4\times3} \\ \mathbf{0}_{1\times3} \end{bmatrix}}^{\breve{\mathbf{f}}_2} \mathbf{a}_m. \tag{31}$$

Note that $\breve{\mathbf{f}}_1$ and $\breve{\mathbf{f}}_2$ are both composed of three individual column vectors, where each column is multiplied by a single control input. For the unit quaternion $\bar{q}$, the matrix $\Xi\left(\bar{q}\right)$ is

$$\Xi\left(\bar{q}\right) = \begin{bmatrix} -\mathbf{q}^{\mathrm{T}} \\ q_0\mathbf{I}_3 - [\mathbf{q}\times] \end{bmatrix}, \tag{32}$$

which expresses the time rate of change of $\bar{q}$ as a linear function of the angular velocity vector $\boldsymbol{\omega}$

$$\dot{\bar{q}} = \frac{1}{2}\Omega(\boldsymbol{\omega})\,\bar{q} = \frac{1}{2}\Xi\left(\bar{q}\right)\boldsymbol{\omega}. \tag{33}$$

We can write the measurement functions for the camera position and the camera orientation as, respectively,

$$\mathbf{h}_1 = \alpha\left(\mathbf{p}_I^W + \mathbf{C}(\bar{q}_I^W)\;\mathbf{p}_C^I\right), \tag{34}$$

$$\mathbf{h}_2 = \bar{q}_I^W \otimes \bar{q}_C^I, \tag{35}$$

where the terms in brackets in (34) define the absolute position of the camera in the world frame, which is scaled according to the factor α. We enforce the unit quaternion constraint for $\bar{q}_I^W$ with the measurement function

$$h_3 = (\bar{q}_I^W)^{\mathrm{T}}\,\bar{q}_I^W = 1. \tag{36}$$

To show that the non-linear system described by (31) is locally weakly observable, given measurements defined by (34)–(36), we must prove that the matrix formed from the gradients of the Lie derivatives of the measurement functions has full column rank. We begin by defining the zeroth-order Lie derivatives of $\mathbf{h}_1$, $\mathbf{h}_2$, and h_3, which are simply the measurement functions themselves:

$$L^0\mathbf{h}_1 = \alpha\left(\mathbf{p}_I^W + \mathbf{C}(\bar{q}_I^W)\;\mathbf{p}_C^I\right), \tag{37}$$

$$L^0\mathbf{h}_2 = \bar{q}_I^W \otimes \bar{q}_C^I, \tag{38}$$

$$L^0 h_3 = (\bar{q}_I^W)^{\mathrm{T}}\,\bar{q}_I^W. \tag{39}$$

Their gradients are

$$\begin{aligned}\nabla L^0\mathbf{h}_1 = \big[&\alpha\,\mathbf{I}_3\,\alpha\,\boldsymbol{\Gamma}\left(\bar{q}_I^W,\mathbf{p}_C^I\right)\mathbf{0}_{3\times3}\,\mathbf{0}_{3\times3}\\ &\mathbf{0}_{3\times3}\,\mathbf{0}_{3\times3}\,\alpha\,\mathbf{C}(\bar{q}_I^W)\mathbf{0}_{3\times4}\,\mathbf{U}_1(\mathbf{p}_I^W,\bar{q}_I^W,\mathbf{p}_C^I)\big]\end{aligned} \tag{40}$$

$$\begin{aligned}\nabla L^0\mathbf{h}_2 = [&\mathbf{0}_{4\times3}\;\;\mathbf{D}(\bar{q}_C^I)\;\;\mathbf{0}_{4\times3}\;\;\mathbf{0}_{4\times3}\;\;\mathbf{0}_{4\times3}\\ &\mathbf{0}_{4\times3}\;\;\mathbf{0}_{4\times3}\;\;\mathbf{S}(\bar{q}_I^W)\;\;\mathbf{0}_{4\times1}]\end{aligned} \tag{41}$$

$$\begin{aligned}\nabla L^0 h_3 = [&\mathbf{0}_{1\times3}\;\;2\left(\bar{q}_I^W\right)^T\;\;\mathbf{0}_{1\times3}\;\;\mathbf{0}_{1\times3}\\ &\mathbf{0}_{1\times3}\;\;\mathbf{0}_{1\times3}\;\;\mathbf{0}_{1\times3}\;\;\mathbf{0}_{1\times4}\;\;0],\end{aligned} \tag{42}$$

where

$$\mathbf{\Gamma}\left(\bar{q}_I^W, \mathbf{p}_C^I\right) = \frac{\partial\, \mathbf{C}(\bar{q}_I^W)\, \mathbf{p}_C^I}{\partial\, \bar{q}_I^W}. \tag{43}$$

Matrices $\mathbf{U}_i(\cdot)$ are not required for our analysis, and so we do not expand them completely. Continuing, the first-order Lie derivatives of $\mathbf{h}_1$ and $\mathbf{h}_2$ with respect to $\mathbf{f}_0$ are

$$L^1_{\mathbf{f}_0}\mathbf{h}_1 = \nabla L^0\mathbf{h}_1 \cdot \mathbf{f}_0 = \alpha\, \mathbf{v}^W - \frac{1}{2}\alpha\, \mathbf{\Gamma}\left(\bar{q}_I^W, \mathbf{p}_C^I\right)\, \Xi\left(\bar{q}_I^W\right)\, \mathbf{b}_g \tag{44}$$

$$L^1_{\mathbf{f}_0}\mathbf{h}_2 = \nabla L^0\mathbf{h}_2 \cdot \mathbf{f}_0 = -\frac{1}{2}\mathbf{D}(\bar{q}_C^I)\, \Xi\left(\bar{q}_I^W\right)\, \mathbf{b}_g. \tag{45}$$

The gradients are

$$\begin{aligned}\nabla L^1_{\mathbf{f}_0}\mathbf{h}_1 = [\mathbf{0}_{3\times3}\;\; &\mathbf{U}_2(\bar{q}_I^W, \mathbf{b}_g, \mathbf{p}_C^I, \alpha)\;\; \alpha\,\mathbf{I}_3\;\; \mathbf{U}_3(\bar{q}_I^W, \mathbf{p}_C^I, \alpha)\\ &\mathbf{0}_{3\times3}\;\; \mathbf{0}_{3\times3}\;\; \mathbf{U}_4(\bar{q}_I^W, \mathbf{b}_g, \alpha)\\ &\quad \mathbf{0}_{3\times4}\;\; \mathbf{U}_5(\bar{q}_I^W, \mathbf{v}^W, \mathbf{b}_g, \mathbf{p}_C^I)\,]\end{aligned} \tag{46}$$

$$\begin{aligned}\nabla L^1_{\mathbf{f}_0}\mathbf{h}_2 = [\mathbf{0}_{4\times3}\;\; &\mathbf{U}_6(\mathbf{b}_g, \bar{q}_C^I)\;\; \mathbf{0}_{4\times3}\;\; -\frac{1}{2}\mathbf{D}(\bar{q}_C^I)\, \Xi\left(\bar{q}_I^W\right)\\ &\mathbf{0}_{4\times3}\;\; \mathbf{0}_{4\times3}\;\; \mathbf{0}_{4\times3}\;\; \mathbf{U}_7(\bar{q}_I^W, \mathbf{b}_g)\;\; \mathbf{0}_{4\times1}].\end{aligned} \tag{47}$$

We now consider Lie derivatives with respect to $\breve{\mathbf{f}}_1$ and $\breve{\mathbf{f}}_2$; we stack the gradients of the individual columns of the resulting matrices, where each column corresponds to a single control. The first-order Lie derivatives of $\mathbf{h}_1$ with respect to the columns of $\breve{\mathbf{f}}_1$ are

$$\begin{aligned}L^1_{\breve{\mathbf{f}}_1}\mathbf{h}_1 = \nabla L^0\mathbf{h}_1 \cdot \breve{\mathbf{f}}_1 &= \frac{1}{2}\alpha\, \mathbf{\Gamma}\left(\bar{q}_I^W, \mathbf{p}_C^I\right)\, \Xi\left(\bar{q}_I^W\right)\\ &= \left[L^1_{\mathbf{f}_{1,1}}\mathbf{h}_1\;\; L^1_{\mathbf{f}_{1,2}}\mathbf{h}_1\;\; L^1_{\mathbf{f}_{1,3}}\mathbf{h}_1\right],\end{aligned} \tag{48}$$

which is a 3 × 3 matrix[8]. Computing the gradients of the individual Lie derivatives (columns), we obtain

$$\begin{aligned}\nabla L^1_{\breve{\mathbf{f}}_1}\mathbf{h}_1 &= \begin{bmatrix}\nabla L^1_{\mathbf{f}_{1,1}}\mathbf{h}_1\\ \nabla L^1_{\mathbf{f}_{1,2}}\mathbf{h}_1\\ \nabla L^1_{\mathbf{f}_{1,3}}\mathbf{h}_1\end{bmatrix}\\ &= \left[\mathbf{0}_{9\times3}\;\; \mathbf{U}_8(\bar{q}_I^W, \mathbf{p}_C^I, \alpha)\;\; \mathbf{0}_{9\times3}\;\; \mathbf{0}_{9\times3}\right.\\ &\quad \left.\mathbf{0}_{9\times3}\;\; \mathbf{0}_{9\times3}\;\; \alpha\,\mathbf{G}_1(\bar{q}_I^W)\;\; \mathbf{0}_{9\times4}\;\; \mathbf{U}_9(\bar{q}_I^W, \mathbf{p}_C^I)\right],\end{aligned} \tag{49}$$

where the matrix[9] $\mathbf{G}_1(\bar{q}_I^W)$ is formed from the stacked partial derivatives of the columns of $\frac{1}{2}\mathbf{\Gamma}\left(\bar{q}_I^W, \mathbf{p}_C^I\right)\,\Xi\left(\bar{q}_I^W\right)$ with respect to $\mathbf{p}_C^I$. We also require the second-order Lie derivative of $\mathbf{h}_1$ with respect to $\mathbf{f}_0$

$$\begin{aligned}L^2_{\mathbf{f}_0}\mathbf{h}_1 = \nabla L^1_{\mathbf{f}_0}\mathbf{h}_1 \cdot \mathbf{f}_0 = &-\frac{1}{2}\mathbf{U}_2(\bar{q}_I^W, \mathbf{b}_g, \mathbf{p}_C^I, \alpha)\, \Xi\left(\bar{q}_I^W\right)\, \mathbf{b}_g\\ &-\alpha\, \mathbf{C}(\bar{q}_I^W)\, \mathbf{b}_a + \alpha\, \mathbf{g}^W.\end{aligned} \tag{50}$$

The gradient is

$$\begin{aligned}\nabla L^2_{\mathbf{f}_0}\mathbf{h}_1 = [\mathbf{0}_{3\times3}\;\; &\mathbf{U}_{10}(\bar{q}_I^W, \mathbf{b}_g, \mathbf{p}_C^I, \alpha) - \alpha\, \mathbf{\Gamma}(\bar{q}_I^W, \mathbf{b}_a)\;\; \mathbf{0}_{3\times3}\\ &\mathbf{U}_{11}(\bar{q}_I^W, \mathbf{b}_g, \mathbf{p}_C^I, \alpha)\\ &-\alpha\, \mathbf{C}(\bar{q}_I^W)\;\; \alpha\,\mathbf{I}_3\;\; \mathbf{U}_{12}(\bar{q}_I^W, \mathbf{b}_g, \alpha)\;\; \mathbf{0}_{3\times4}\\ &\mathbf{U}_{13}(\bar{q}_I^W, \mathbf{b}_g, \mathbf{b}_a, \mathbf{p}_C^I, \mathbf{g}^W)\,],\end{aligned} \tag{51}$$

where

$$\mathbf{\Gamma}(\bar{q}_I^W, \mathbf{b}_a) = \frac{\partial\, \mathbf{C}(\bar{q}_I^W)\, \mathbf{b}_a}{\partial\, \bar{q}_I^W}. \tag{52}$$

Next, we require the second-order Lie derivatives of $\nabla L^1_{\mathbf{f}_0}\mathbf{h}_1$ with respect to the columns of $\breve{\mathbf{f}}_2$,

$$\begin{aligned}L^1_{\breve{\mathbf{f}}_2}L^1_{\mathbf{f}_0}\mathbf{h}_1 = \nabla L^1_{\mathbf{f}_0}\mathbf{h}_1 \cdot \breve{\mathbf{f}}_2 &= \alpha\, \mathbf{C}(\bar{q}_I^W)\\ &= \left[L^1_{\mathbf{f}_{2,1}}L^1_{\mathbf{f}_0}\mathbf{h}_1\;\; L^1_{\mathbf{f}_{2,2}}L^1_{\mathbf{f}_0}\mathbf{h}_1\;\; L^1_{\mathbf{f}_{2,3}}L^1_{\mathbf{f}_0}\mathbf{h}_1\right],\end{aligned} \tag{53}$$

which is also a 3 × 3 matrix. Stacking the gradients of the individual columns, we obtain

$$\begin{aligned}\nabla L^1_{\breve{\mathbf{f}}_2}L^1_{\mathbf{f}_0}\mathbf{h}_1 &= \begin{bmatrix}\nabla L^1_{\mathbf{f}_{2,1}}L^1_{\mathbf{f}_0}\mathbf{h}_1\\ \nabla L^1_{\mathbf{f}_{2,2}}L^1_{\mathbf{f}_0}\mathbf{h}_1\\ \nabla L^1_{\mathbf{f}_{2,3}}L^1_{\mathbf{f}_0}\mathbf{h}_1\end{bmatrix}\\ &= [\mathbf{0}_{9\times3}\;\; \alpha\,\mathbf{G}_2(\bar{q}_I^W)\;\; \mathbf{0}_{9\times3}\;\; \mathbf{0}_{9\times3}\;\; \mathbf{0}_{9\times3}\\ &\quad\; \mathbf{0}_{9\times3}\;\; \mathbf{0}_{9\times3}\;\; \mathbf{0}_{9\times4}\;\; \mathbf{G}_3(\bar{q}_I^W)\,],\end{aligned} \tag{54}$$

where the matrices $\mathbf{G}_2(\bar{q}_I^W)$ and $\mathbf{G}_3(\bar{q}_I^W)$ are formed from the stacked partial derivatives of the columns of $\alpha\mathbf{C}(\bar{q}_I^W)$ with respect to $\bar{q}_I^W$ and α, respectively. Finally, we require the third-order Lie derivatives of $\nabla L^2_{\mathbf{f}_0}\mathbf{h}_1$ with respect to the columns of $\breve{\mathbf{f}}_1$

$$\begin{aligned}L^1_{\breve{\mathbf{f}}_1}L^2_{\mathbf{f}_0}\mathbf{h}_1 = \nabla L^2_{\mathbf{f}_0}\mathbf{h}_1 \cdot \breve{\mathbf{f}}_1 &= \frac{1}{2}\mathbf{U}_9(\bar{q}_I^W, \mathbf{b}_g, \mathbf{p}_C^I, \alpha)\; \Xi(\bar{q}_I^W)\\ &\quad -\frac{1}{2}\alpha\, \mathbf{\Gamma}(\bar{q}_I^W, \mathbf{b}_a)\; \Xi\left(\bar{q}_I^W\right)\\ &= \left[L^1_{\mathbf{f}_{1,1}}L^2_{\mathbf{f}_0}\mathbf{h}_1 L^1_{\mathbf{f}_{1,2}}L^2_{\mathbf{f}_0}\mathbf{h}_1 L^1_{\mathbf{f}_{1,3}}L^2_{\mathbf{f}_0}\mathbf{h}_1\right].\end{aligned} \tag{55}$$

We proceed as before and stack the gradients of the individual columns,

$$\begin{aligned}\nabla L^1_{\breve{\mathbf{f}}_1}L^2_{\mathbf{f}_0}\mathbf{h}_1 &= \begin{bmatrix}\nabla L^1_{\mathbf{f}_{1,1}}L^2_{\mathbf{f}_0}\mathbf{h}_1\\ \nabla L^1_{\mathbf{f}_{1,2}}L^2_{\mathbf{f}_0}\mathbf{h}_1\\ \nabla L^1_{\mathbf{f}_{1,3}}L^2_{\mathbf{f}_0}\mathbf{h}_1\end{bmatrix}\\ &= [\mathbf{0}_{9\times3}\;\; \mathbf{U}_{14}(\bar{q}_I^W, \mathbf{b}_g, \mathbf{b}_a, \mathbf{p}_C^I, \alpha)\;\; \mathbf{0}_{9\times3}\\ &\qquad \mathbf{U}_{15}(\bar{q}_I^W, \mathbf{b}_g, \mathbf{p}_C^I, \alpha)\\ &\quad -\alpha\,\mathbf{G}_1(\bar{q}_I^W)\;\; \mathbf{0}_{9\times3}\;\; \mathbf{U}_{16}(\bar{q}_I^W, \mathbf{b}_g, \alpha)\;\; \mathbf{0}_{9\times4}\\ &\qquad \mathbf{U}_{17}(\bar{q}_I^W, \mathbf{b}_g, \mathbf{b}_a, \mathbf{p}_C^I)\,],\end{aligned} \tag{56}$$

where $\mathbf{G}_1(\bar{q}_I^W)$ is (again) formed from stacked partial derivatives of the columns of $\frac{1}{2}\mathbf{\Gamma}\left(\bar{q}_I^W, \mathbf{b}_a\right)\,\Xi\left(\bar{q}_I^W\right)$ with respect to $\mathbf{b}_a$.

We form the observability matrix $\mathcal{O}$ by stacking the gradient matrices above:

$$\mathcal{O} = \begin{bmatrix} \nabla L^0 \mathbf{h}_1 \\ \nabla L^0 \mathbf{h}_2 \\ \nabla L^1_{\mathbf{f}_0} \mathbf{h}_1 \\ \nabla L^1_{\mathbf{f}_0} \mathbf{h}_2 \\ \nabla L^1_{\breve{\mathbf{f}}_1} \mathbf{h}_1 \\ \nabla L^2_{\mathbf{f}_0} \mathbf{h}_1 \\ \nabla L^1_{\breve{\mathbf{f}}_2} L^1_{\mathbf{f}_0} \mathbf{h}_1 \\ \nabla L^0 h_3 \\ \nabla L^1_{\breve{\mathbf{f}}_1} L^2_{\mathbf{f}_0} \mathbf{h}_1 \end{bmatrix} = \begin{bmatrix} \alpha\,\mathbf{I}_3 & \alpha\,\boldsymbol{\Gamma}\left(\bar{q}^W_I, \mathbf{p}^I_C\right) & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \alpha\,\mathbf{C}(\bar{q}^W_I) & \mathbf{0}_{3\times4} & \mathbf{U}_1 \\ \mathbf{0}_{4\times3} & \mathbf{D}(\bar{q}^I_C) & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{S}(\bar{q}^W_I) & \mathbf{0}_{4\times1} \\ \mathbf{0}_{3\times3} & \mathbf{U}_2 & \alpha\,\mathbf{I}_3 & \mathbf{U}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{U}_4 & \mathbf{0}_{3\times4} & \mathbf{U}_5 \\ \mathbf{0}_{4\times3} & \mathbf{U}_6 & \mathbf{0}_{4\times3} & -\frac{1}{2}\mathbf{D}(\bar{q}^I_C)\ \Xi\left(\bar{q}^W_I\right) & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{U}_7 & \mathbf{0}_{4\times1} \\ \mathbf{0}_{9\times3} & \mathbf{U}_8 & \mathbf{0}_{9\times3} & \mathbf{0}_{9\times3} & \mathbf{0}_{9\times3} & \mathbf{0}_{9\times3} & \alpha\,\mathbf{G}_1(\bar{q}^W_I) & \mathbf{0}_{9\times4} & \mathbf{U}_9 \\ \mathbf{0}_{3\times3} & \mathbf{U}_{10} - \alpha\,\boldsymbol{\Gamma}\left(\bar{q}^W_I, \mathbf{b}_a\right) & \mathbf{0}_{3\times3} & \mathbf{U}_{11} & -\alpha\,\mathbf{C}(\bar{q}^W_I) & \alpha\,\mathbf{I}_3 & \mathbf{U}_{12} & \mathbf{0}_{3\times4} & \mathbf{U}_{13} \\ \mathbf{0}_{9\times3} & \alpha\,\mathbf{G}_2(\bar{q}^W_I) & \mathbf{0}_{9\times3} & \mathbf{0}_{9\times3} & \mathbf{0}_{9\times3} & \mathbf{0}_{9\times3} & \mathbf{0}_{9\times3} & \mathbf{0}_{9\times4} & \mathbf{G}_3(\bar{q}^W_I) \\ \mathbf{0}_{1\times3} & 2\,(\bar{q}^W_I)^{\mathrm{T}} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times4} & 0 \\ \mathbf{0}_{9\times3} & \mathbf{U}_{14} & \mathbf{0}_{9\times3} & \mathbf{U}_{15} & -\alpha\,\mathbf{G}_1(\bar{q}^W_I) & \mathbf{0}_{9\times3} & \mathbf{U}_{16} & \mathbf{0}_{9\times4} & \mathbf{U}_{17} \end{bmatrix}, \tag{57}$$

where the complete matrix has size 45 × 27. Our choice of the ordering of the Lie derivatives is made to facilitate the proof of the following theorem about the rank of $\mathcal{O}$.

Theorem 1. The matrix $\mathcal{O}$, defined by (57), has full column rank when $\alpha \neq 0$.

Proof. We use block Gaussian elimination to show that $\mathcal{O}$ has full column rank when $\alpha \neq 0$. Lemmas 1–4 and their abbreviated proofs are provided in the Appendix.

Step 1: By Lemma 1, the matrix $\left[\alpha\,\mathbf{G}_2(\bar{q}^W_I)\ \ \mathbf{G}_3(\bar{q}^W_I)\right]$ has full column rank, with the constraint that $\bar{q}^W_I$ is a unit quaternion. This allows us to eliminate the remaining non-zero entries in column blocks 2 and 9.

Step 2: By Lemma 2, the matrix $\mathbf{S}(\bar{q}^W_I)$ has full column rank. We eliminate the remaining non-zero entries in column block 8.

Step 3: By Lemma 3, the matrix $\mathbf{D}(\bar{q}^W_I)\ \Xi\left(\bar{q}^W_I\right)$ has full column rank, and so we eliminate the remaining non-zero entry in column block 4.

Step 4: Finally, by Lemma 4, the matrix $\alpha\,\mathbf{G}_1(\bar{q}^W_I)$ has full column rank, and we eliminate the remaining non-zero entries in column blocks 5 and 7.

After applying Lemmas 1–4 and removing any rows of $\mathcal{O}$ that consist entirely of zeros, we obtain

$$\mathcal{O} = \begin{bmatrix} \alpha\,\mathbf{I}_3 & \mathbf{0}_{3\times4} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{4\times3} & \mathbf{0}_{4\times4} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{I}_4 & \mathbf{0}_{4\times1} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} & \alpha\,\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \alpha\mathbf{I}_3 & \mathbf{0}_{3\times4} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \alpha\,\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{4\times3} & \alpha\,\mathbf{I}_4 & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times4} & \mathbf{0}_{4\times1} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times4} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times4} & 1 \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \alpha\,\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} & \mathbf{0}_{3\times1} \end{bmatrix} \tag{58}$$

which has full column rank whenever $\alpha \neq 0$. This completes the proof. □

Note that $\mathcal{O}$ has full column rank only when the scale factor α is non-zero. This constraint is not an issue in practice, as any non-zero values for the initial landmark depths will result in a non-zero scene scale. Our analysis, based on Lemmas 1–4, requires the IMU to undergo rotation about at least two axes and acceleration along at least two axes (cf. the Appendix).

5.2.3. Observability of Target-Based Calibration For target-based calibration, the true distances between landmarks (corner points on the calibration target) are *a priori* known. This allows us to replace (34) with the observation function

$$\mathbf{h}^*_1 = \mathbf{p}^W_I + \mathbf{C}(\bar{q}^W_I)\ \mathbf{p}^I_C, \tag{59}$$

where the scale factor, α, does not appear. We can modify the observability analysis by removing the α term from any entries in $\mathcal{O}$, and eliminating the last column of the matrix. Using the same sequence of steps described in Section 5.2.2, it follows immediately that target-based calibration is locally weakly observable, as expected. Note also that, because the true landmark positions are known, it not necessary to anchor the world frame as described in Section 5.2.1.

6. Filtering for Structure, Motion, and Relative Pose Self-calibration

In this section, we develop a practical algorithm for recovering the local scene structure, IMU motion, and the sensor-to-sensor transform, from camera and IMU measurements. The algorithm is based on the UKF, an alternative to the well-known EKF (Julier and Uhlmann 2004).

Our choice of the UKF is motivated by its superior performance compared with the EKF for many non-linear problems. For Gaussian state distributions, the posterior estimate produced by the UKF is accurate to the third order, while the EKF estimate is accurate to the first order only[10] (van der Merwe and Wan 2004). Recent work has shown, for a variety of navigation problems, that the UKF produces more accurate and consistent estimates than the EKF. For example, Huster and Rock (2003) demonstrate that the UKF is able to accurately estimate ego-motion by fusing inertial data with monocular visual measurements, while the EKF produces inconsistent and biased results. An additional benefit of the UKF is that it is *derivative-free*, i.e. the filter does not require Jacobian matrices to be computed. Although the computational demands of the UKF are greater than those of the EKF, we note that the UKF is amenable to parallelization.

We give a brief overview of the filter in Section 6.1 below, and refer the reader to Julier and Uhlmann (2004) for a more detailed treatment. In Section 6.2 we discuss several changes to the standard UKF algorithm that are necessary to correctly propagate and update unit quaternions in the state vector. We then review filter initialization in Section 6.3 and outlier rejection in Section 6.4.

6.1. Unscented Filtering

The UKF is a Bayesian filtering algorithm which updates the system state using a set of deterministic sample points drawn from the prior distribution. These *sigma points* lie on the covariance contours in state space. The filter propagates each sigma point through the (non-linear) process and measurement models, and computes the weighted averages of the transformed points to determine the posterior mean and covariance. This technique, known as the *unscented transform*, is a type of statistical local linearization, which produces more accurate estimates than the analytic local linearization employed by the EKF.

We use a continuous-discrete formulation of the UKF, in which the sigma points are propagated forward in time using fourth-order Runge–Kutta integration of (15)–(16). Measurement updates occur at discrete time steps. Our filter implementation augments the state vector and state covariance matrix with a process noise component, as described by Julier and Uhlmann (2004),

$$\mathbf{x}_a(t_k) = \begin{bmatrix} \mathbf{x}(t_k) \\ \mathbf{n}(t_k) \end{bmatrix}, \tag{60}$$

where $\mathbf{x}_a(t_k)$ is the augmented state vector, of size N, at time t_k, and $\mathbf{n}(t_k)$ is the 12×1 process noise vector defined by (20).

At time t_{k-1}, immediately after the last measurement update, the augmented state mean $\hat{\mathbf{x}}_a^+(t_{k-1})$ and augmented state covariance matrix $\mathbf{P}_a^+(t_{k-1})$ are

$$\hat{\mathbf{x}}_a^+(t_{k-1}) = \begin{bmatrix} \hat{\mathbf{x}}^+(t_{k-1}) \\ \mathbf{0}_{12\times 1} \end{bmatrix},$$

$$\mathbf{P}_a^+(t_{k-1}) = \begin{bmatrix} \mathbf{P}^+(t_{k-1}) & \mathbf{0}_{m\times 12} \\ \mathbf{0}_{12\times m} & \mathbf{Q} \end{bmatrix}, \tag{61}$$

where $\mathbf{Q}$ is the covariance matrix for the noise vector $\mathbf{n}$, and the state vector $\hat{\mathbf{x}}^+(t_{k-1})$ has size $m = 26 + 3n$ (when n landmarks are included). We employ the scaled form of the unscented transform Julier (2002), which requires a scaling term

$$\lambda = \alpha^2(N + \beta) - N. \tag{62}$$

Here, the α parameter controls the spread of the sigma points about the state mean,[1] α is usually set to a small positive value. The β parameter is used to incorporate corrections to higher-order terms in the Taylor series expansion of the state distribution; setting $\beta = 2$ minimizes the fourth-order error for jointly Gaussian distributions. For state estimate $\hat{\mathbf{x}}^+(t_{k-1})$, the augmented state vector $\hat{\mathbf{x}}_a^+(t_{k-1})$ is used to generate the set of sigma points according to

$$^0\boldsymbol{\chi}_a(t_{k-1}) = \hat{\mathbf{x}}_a^+(t_{k-1}) \tag{63}$$

$$^l\boldsymbol{\chi}_a(t_{k-1}) = \hat{\mathbf{x}}_a^+(t_{k-1}) + {}^j\mathbf{S}(t_{k-1}),$$

$$j = l = 1, \ldots, N \tag{64}$$

$$^l\boldsymbol{\chi}_a(t_{k-1}) = \hat{\mathbf{x}}_a^+(t_{k-1}) - {}^j\mathbf{S}(t_{k-1}),$$

$$j = 1, \ldots, N, \quad l = N+1, \ldots, 2N \tag{65}$$

$$\mathbf{S}(t_{k-1}) = \sqrt{(\lambda + N)\ \mathbf{P}_a^+(t_{k-1})}, \tag{66}$$

where ${}^j\mathbf{S}$ denotes the jth column of the matrix $\mathbf{S}$. The matrix square root of $\mathbf{P}_a^+(t_{k-1})$ is found by Cholesky decomposition (Golub and Loan 1996). The associated sigma point weight values are

$$^0W_m = \lambda/(\lambda + N), \tag{67}$$

$$^0W_c = \lambda/(\lambda + N) + (1 - \alpha^2 + \beta), \tag{68}$$

$$^jW_m = {}^jW_c = \frac{1}{2(\lambda + N)}, \quad j = 1, \ldots, 2N. \tag{69}$$

Individual sigma points are propagated through the augmented non-linear process model function $\mathbf{f}_a$ (which incorporates process noise in the propagation equations) and the weights above are used to calculate the *a priori* state estimate and covariance matrix at time t_k,

$$^i\boldsymbol{\chi}_a(t_k) = \mathbf{f}_a\left({}^i\boldsymbol{\chi}_a(t_{k-1})\right), \quad i = 0, \ldots, 2N \tag{70}$$

$$\hat{\mathbf{x}}^-(t_k) = \sum_{i=0}^{2N} {}^iW_m\, {}^i\boldsymbol{\chi}(t_k) \tag{71}$$

$$\mathbf{P}^-(t_k) = \sum_{i=0}^{2N} {}^iW_c \left({}^i\boldsymbol{\chi}(t_k) - \hat{\mathbf{x}}^-(t_k)\right)\left({}^i\boldsymbol{\chi}(t_k) - \hat{\mathbf{x}}^-(t_k)\right)^{\mathrm{T}}. \tag{72}$$

When a measurement arrives (in our case, an observation of one of the landmarks in the environment), we determine the predicted measurement vector by propagating each sigma point through the non-linear measurement model function $\mathbf{h}$,

$$^i\boldsymbol{\gamma}(t_k) = \mathbf{h}\left({}^i\boldsymbol{\chi}(t_k)\right), \quad i = 0, \ldots, 2N \tag{73}$$

$$\hat{\mathbf{z}}(t_k) = \sum_{i=0}^{2N} {}^iW_m\, {}^i\boldsymbol{\gamma}(t_k). \tag{74}$$

We then perform a state update by computing the Kalman gain matrix $\mathbf{K}(t_k)$ and the *a posteriori* state vector and state covariance matrix

$$\mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{z}}}(t_k) = \sum_{i=0}^{2N} {}^iW_c \left({}^i\boldsymbol{\chi}(t_k) - \hat{\mathbf{x}}^-(t_k)\right)\left({}^i\boldsymbol{\gamma}(t_k) - \hat{\mathbf{z}}(t_k)\right)^{\mathrm{T}} \tag{75}$$

$$\mathbf{P}_{\hat{\mathbf{z}}\hat{\mathbf{z}}}(t_k) = \sum_{i=0}^{2N} {}^iW_c \left({}^i\boldsymbol{\gamma}(t_k) - \hat{\mathbf{z}}(t_k)\right)\left({}^i\boldsymbol{\gamma}(t_k) - \hat{\mathbf{z}}(t_k)\right)^{\mathrm{T}} \tag{76}$$

$$\mathbf{K}(t_k) = \mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{z}}}(t_k)\left(\mathbf{P}_{\hat{\mathbf{z}}\hat{\mathbf{z}}}(t_k) + \mathbf{R}(t_k)\right)^{-1} \tag{77}$$

$$\hat{\mathbf{x}}^+(t_k) = \hat{\mathbf{x}}^-(t_k) + \mathbf{K}(t_k)\left(\mathbf{z}(t_k) - \hat{\mathbf{z}}(t_k)\right) \tag{78}$$

$$\mathbf{P}^+(t_k) = \mathbf{P}^-(t_k) - \mathbf{K}(t_k)\,\mathbf{P}_{\hat{\mathbf{z}}\hat{\mathbf{z}}}(t_k)\,\mathbf{K}^{\mathrm{T}}(t_k), \tag{79}$$

[1] note that this α is not related to the scale factor used in the observability analysis;

where $\mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{z}}}(t_k)$ and $\mathbf{P}_{\hat{\mathbf{z}}\hat{\mathbf{z}}}(t_k)$ are the state-measurement cross-covariance matrix and the predicted measurement covariance matrix, respectively, while $\mathbf{R}(t_k)$ is the measurement covariance matrix for the current observation. Note that because our measurement noise is additive, the innovation covariance (in brackets in (77)) is simply the sum of the matrices $\mathbf{P}_{\hat{\mathbf{z}}\hat{\mathbf{z}}}(t_k)$ and $\mathbf{R}(t_k)$.

6.2. The Unscented Quaternion Estimator

The UKF updates the system state by computing the *barycenteric mean* of the sigma points. For unit quaternions, which lie on the unit sphere S^3 in $\mathbb{R}^4$, the barycenter of the transformed sigma points will often *not* represent the correct mean. In particular, the weighted average of several unit quaternions may not be a unit quaternion.

Special consideration is therefore required when propagating unit quaternions through the unscented transform. There are several possible ways to enforce the unit norm constraint within the UKF, for example by incorporating pseudo-measurements, or by projecting the unconstrained time and measurement updates onto the quaternion constraint surface (Julier and LaViola 2007). We follow the method described by Crassidis and Markely (2003) and employ a re-parameterization which incorporates multiplicative, three-parameter orientation *error state* vectors in addition to $\bar{q}^{\,W}_{I}$ and $\bar{q}^{\,I}_{C}$. This approach, called the unscented quaternion estimator (USQUE) in Crassidis and Markely (2003), is similar to the algorithm described by Kraft (2003), but does not require an iterated gradient descent step.

To propagate and update unit quaternions in the state vector, the USQUE filter defines a multiplicative local error quaternion,

$$\delta q = \begin{bmatrix} \delta q_0 & \delta \mathbf{q}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, \tag{80}$$

and the following three-component vector of modified Rodrigues parameters (MRPs), derived from the error quaternion

$$\delta\,\mathbf{e} = \frac{\delta \mathbf{q}}{1 + \delta q_0}. \tag{81}$$

The MRP vector is an unconstrained three-parameter orientation representation, which is singular at 2π, and can be expressed in axis-angle form as

$$\delta\,\mathbf{e} = \bar{\mathbf{a}} \tan(\theta/4), \tag{82}$$

where $\bar{\mathbf{a}}$ defines the rotation axis, and θ is the rotation angle. The inverse transformation, from the MRP vector to the local error quaternion δq, is

$$\delta q_0 = \frac{1 - \|\,\delta\,\mathbf{e}\|}{1 + \|\,\delta\,\mathbf{e}\|} \tag{83}$$

$$\delta \mathbf{q} = (1 + \delta q_0)\,\delta\,\mathbf{e}. \tag{84}$$

From the full sensor state vector (13), we define the modified 24 × 1 sensor error state vector $\mathbf{x}_{se}(t_k)$ as

$$\begin{aligned}\mathbf{x}_{se}(t_k) = [&(\mathbf{p}^{W}_{I}(t))^{\mathrm{T}}\ (\delta\,\mathbf{e}^{W}_{I}(t))^{\mathrm{T}}\ (\mathbf{v}^{W}(t))^{\mathrm{T}}\ (\mathbf{b}_g(t))^{\mathrm{T}} \\ &(\mathbf{b}_a(t))^{\mathrm{T}}\ (\mathbf{g}^{W})^{\mathrm{T}}\ (\mathbf{p}^{I}_{C})^{\mathrm{T}}\ (\delta\,\mathbf{e}^{I}_{C})^{\mathrm{T}}\,]^{\mathrm{T}}\end{aligned} \tag{85}$$

where $\delta\,\mathbf{e}^{W}_{I}$ and $\delta\,\mathbf{e}^{I}_{C}$ are the MRP error state vectors for the orientation quaternions $\bar{q}^{\,W}_{I}$ and $\bar{q}^{\,I}_{C}$, respectively.

Throughout the calibration procedure, the filter maintains an estimate of the full 26 × 1 sensor state vector and the 24 × 24 sensor error state covariance matrix. The initialization procedure (cf. Section 6.3.1) yields an estimate of the full state vector and the error state covariance; for the orientation quaternions $\hat{\bar{q}}^{\,W}_{I}$ and $\hat{\bar{q}}^{\,I}_{C}$ we store the 3 × 3 covariance matrices for the MRP error state representations.

At the start of each propagation step, we compute the sigma points for the error state according to (63)–(65), setting the mean error state MRP vectors to

$${}^{0}\delta\,\hat{\mathbf{e}}^{W}_{I}(t_{k-1}) = \mathbf{0}_{3\times 1}, \quad {}^{0}\delta\,\hat{\mathbf{e}}^{I}_{C}(t_{k-1}) = \mathbf{0}_{3\times 1}, \tag{86}$$

where we indicate the component of the state vector that "belongs" to a specific sigma point by prefixing the vector with a superscripted index (zero above). We follow this convention throughout the section.

To propagate the IMU orientation quaternion $\bar{q}^{\,W}_{I}$ forward in time, we compute the local error quaternion ${}^{j}\delta\hat{\bar{q}}^{\,W}_{I}(t_{k-1})$ from the MRP vector in sigma point j using (83) and (84), and then the full orientation quaternion from the error quaternion

$${}^{j}\hat{\bar{q}}^{\,W}_{I}(t_{k-1}) = {}^{j}\delta\hat{\bar{q}}^{\,W}_{I}(t_{k-1}) \otimes \hat{\bar{q}}^{\,W+}_{I}(t_{k-1}),\ j = 1, \ldots, 2N \tag{87}$$

The other components of the sigma points are determined by addition or subtraction directly. Each sigma point, including the full IMU orientation quaternion, is then propagated through the augmented process model function $\mathbf{f}_a$ from time t_{k-1} to time t_k, with process noise directly incorporated in the non-linear propagation equations. We determine the orientation error quaternions at time t_k by reversing the procedure above, using the propagated mean quaternion

$${}^{j}\delta\hat{\bar{q}}^{\,W}_{I}(t_k) = {}^{j}\hat{\bar{q}}^{\,W}_{I}(t_k) \otimes \left({}^{0}\hat{\bar{q}}^{\,W}_{I}(t_k)\right)^{-1},\ j = 1, \ldots, 2N \tag{88}$$

and finally compute the orientation error MRP vectors using (81). Note that this is required only for the IMU orientation quaternion during the propagation step, as the camera–IMU orientation quaternion does not change. We can then compute the updated *a priori* error state vector and error state covariance matrix using (71) and (72).

We store the orientation quaternions from the last propagation step; when a measurement arrives, we compute the predicted measurement vector for each sigma point using the non-linear measurement function $\mathbf{h}$. The error quaternions for the camera–IMU orientation are determined according to

$${}^{j}\delta\hat{\bar{q}}^{\,I}_{C}(t_k) = {}^{j}\hat{\bar{q}}^{\,I}_{C}(t_k) \otimes \left({}^{0}\hat{\bar{q}}^{\,I}_{C}(t_k)\right)^{-1} \quad j = 1, \ldots, 2N, \tag{89}$$

and the MRP error state vectors are found using (81). Next, we compute the state-measurement cross-covariance matrix, the predicted measurement covariance matrix, the

Kalman gain matrix, and the updated *a posteriori* error state vector and error state covariance matrix. As a final step, we use the updated mean MRP error state vectors to compute the mean error quaternions, and the full state vector orientation quaternions.

6.3. Filter Initialization

At the start of the calibration procedure, we generate an initial estimate of the sensor state (i.e. the IMU pose in the world frame). For target-free self-calibration, we also compute an initial estimate of the map state (i.e. landmark positions in the world frame). After initialization is complete, we have an estimate of the full state vector and the error state covariance matrix, which are the inputs to the UKF.

6.3.1. Sensor State Initialization For target-based calibration, we initially compute estimates of the camera position $\hat{\mathbf{p}}_C^W$ and orientation $\hat{\bar{q}}_C^W$ in the world frame. Given the known positions of the corner points on the calibration target and their image projections, we calculate a solution in closed form for the orientation of the camera using Horn's method[11] (Horn 1987). This is followed by an iterative non-linear least-squares refinement step (using modified Rodrigues parameters), which also provides the 3 × 3 MRP error covariance matrix.

An initial estimate of the camera pose relative to the IMU is also required. We use hand measurements of the relative pose for the experiments described in this paper; however, this information in many cases may be available from CAD drawings or other sources. Given the estimate of the camera pose in the world frame and an estimate of the relative pose of the camera with respect to the IMU, we compute an initial estimate of the IMU pose in the world frame, according to

$$\hat{\mathbf{p}}_I^W = \hat{\mathbf{p}}_C^W - \mathbf{C}(\hat{\bar{q}}_C^W)\,\mathbf{C}^T(\hat{\bar{q}}_C^I)\,\hat{\mathbf{p}}_C^I \tag{90}$$

$$\hat{\bar{q}}_I^W = \hat{\bar{q}}_C^W \otimes \left(\hat{\bar{q}}_C^I\right)^{-1}. \tag{91}$$

We determine the covariance matrix for the IMU pose by computing the Jacobians of (90) and (91), again using modified Rodrigues parameters. For the target-free case, the initialization procedure is the same, except that we use $\hat{\mathbf{p}}_C^W = \mathbf{0}_{3\times 1}$ as the initial camera position, and set the initial camera orientation arbitrarily, with zero initial uncertainty.

6.3.2. Landmark Position Initialization As part of the target-free calibration procedure, we estimate the map state (landmark positions) as well as the sensor state. We typically choose approximately 40–50 salient features in the first camera image as point landmarks. At present, we do not add new landmarks when the camera moves (since we expect to perform calibration in a limited area).

As noted in Section 4.1, we parameterize landmark positions in the state vector using Cartesian coordinates, and initialize the positions using a nominal depth value, along the corresponding camera rays in the first camera image. The positions of the n landmarks in the world frame are estimated as

$$\hat{\mathbf{p}}_{l_i}^W = d\,\mathcal{K}^{-1}\,[u_i \quad v_i \quad 1]^{\mathrm{T}}, \quad i = 1 \ldots n. \tag{92}$$

where d is the nominal depth value, $\mathcal{K}^{-1}$ is the inverse of the camera intrinsic calibration matrix, and $[u_i \quad v_i]^{\mathrm{T}}$ is the vector of observed image coordinates for landmark i.

To anchor the world frame, we select three highly distinct features from the initial set. We use very small image plane uncertainties when computing the three-dimensional covariance ellipses for these landmarks. This effectively "locks down" the initial camera pose and ensures that the full system state is observable (cf. Section 5.2).

6.4. Feature Detection, Tracking, and Outlier Rejection

We use different feature detection and tracking methods for target-based and target-free calibration. For the target-based case, we first locate candidate corner points in each image using a fast template-matching algorithm. This is followed by a homography-based check to ensure that all of the points lie on the planar surface of the target. Once we have coarse estimates of the locations of the corner points, we refine those estimates to subpixel accuracy using a saddle point detector (Lucchese and Mitra 2002). The integrated IMU accelerometer measurements are used to bound the search for the calibration target in the next image.

Target-free calibration is typically performed in a previously unseen and unknown environment. In this case, we select a set of salient and well-localized point features to use as landmarks. We employ SIFT Lowe (2004) as our feature detector. SIFT features are invariant to changes in scale and rotation, and partially invariant to changes in illumination; a fast C implementation of SIFT is available (Vedaldi and Fulkerson 2009).

To detect outliers due to mismatched features in consecutive image frames, we use a simple threshold test based on the χ^2 statistic for the measurement residuals. We discard any feature measurements whose residuals are above the threshold.

7. Simulation Studies

We initially carried out a series of simulation studies to characterize the performance of the calibration algorithm, with ground truth available. We modeled a sensor beam, 15 cm in length, moving through space according to accurate rigid body kinematics. During each simulation, the beam rotated while traveling along the corkscrew-like trajectory

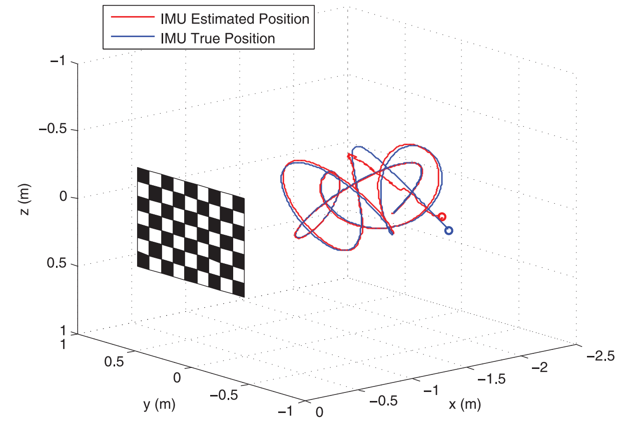


Fig. 2. Estimated trajectory of the IMU (light line) versus true trajectory (dark line), over the 25-second simulation time interval, for trial TB-1. The estimated initial position and true initial position of the IMU are indicated by light and dark circles, respectively.

shown in Figure 2, in order to excite all six degrees of translational and rotational freedom. The IMU had a maximum angular rotation rate of 36.0° s^{-1} and a maximum linear acceleration of 1.0 m s^{-2}.

Our aim was to make the simulation as realistic as possible: the characteristics of our simulated camera and IMU are the same as those of the real hardware described in Section 8. Simulated IMU updates occurred at a rate of 100 Hz, while camera images arrived at either 7.5 or 15 Hz (for target-based and target-free calibration, respectively). The process noise for the simulated IMU matched the noise properties of the IMU available in our laboratory. Our simulated camera had a horizontal field of view (FOV) of 58° and a resolution of 640 × 480 pixels.

For both the target-based and target-free simulations, we used a set of 48 corner points (8 × 6) on a simulated planar target as landmarks. The points were spaced 10.4 cm apart (again, accurately modeling our real calibration target). To generate each simulated image frame, we projected the known corner points onto the camera image plane and added uncorrelated, zero-mean Gaussian noise to the u (horizontal) and v (vertical) image coordinates. The noise had a standard deviation of 1.0 pixel in both the u and v directions. We constrained the rotation of the camera along the trajectory to ensure that the majority of the points on the calibration target remained visible throughout each simulation run.

In the sections that follow, we present results from four different simulation trials: three simulations of target-based calibration (with varying amounts of initial error in the estimate of the camera–IMU relative orientation, and with intentional misestimation of the gravity vector), and one simulation of target-free self-calibration. These results are a representative sample of the performance of the calibration algorithms across a large number of simulations.

7.1. Target-based Calibration

Simulation results for target-based calibration are shown in Figures 3 and 4, and in Tables 1 and 2. During the simulation, 2,500 IMU measurements and 187 camera images (in which all 48 corner points on the target were visible) were processed, over 25 seconds. The initial IMU position and orientation estimates were computed using the procedure described in Section 6.3.1.

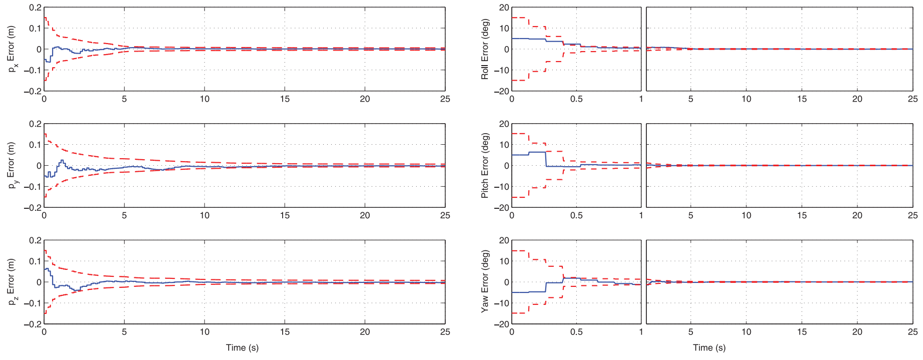
For the first simulation trial (designated “TB-1” in Tables 1 and 2), the initial translation and orientation errors were $\tilde{\mathbf{p}}^I_C = [-5 \quad -5 \quad 6]^{\mathrm{T}}$ cm and $\tilde{\boldsymbol{\xi}}^I_C = [5 \quad 5 \quad -5]^{\mathrm{T}}$ degrees (roll, pitch and yaw), respectively[12]. The plot in Figure 3(b) indicates that the orientation estimates converge rapidly, with residual errors of less than 0.1° in roll, pitch, and yaw[13] after 25 seconds. Likewise, the translation errors, shown in Figure 3(a), decrease to approximately 3 mm or less along each axis with an uncertainty (3σ) of ±7 mm or less, over the 25-second simulation time interval.

We also investigated calibration with larger initial orientation errors. For the second simulation trial (designated “TB-2” in Tables 1 and 2) we used the same translation error as for the TB-1 simulation, but an initial orientation error of $\tilde{\boldsymbol{\xi}}^I_C = [10 \quad -10 \quad -10]^{\mathrm{T}}$ degrees (roll, pitch, and yaw), respectively. The plot in Figure 4(b) shows that the orientation estimates again converge very rapidly, with residual errors of less than 0.1° in roll, pitch, and yaw after 25 seconds. The translation errors decrease to less than 3 mm along each axis, with an uncertainty (3σ) of ±8 mm or less within the 25-second simulation time interval. These results indicate that the filter is able to accurately estimate the calibration parameters despite large initial errors in the relative camera orientation.

7.2. Target-free Self-calibration

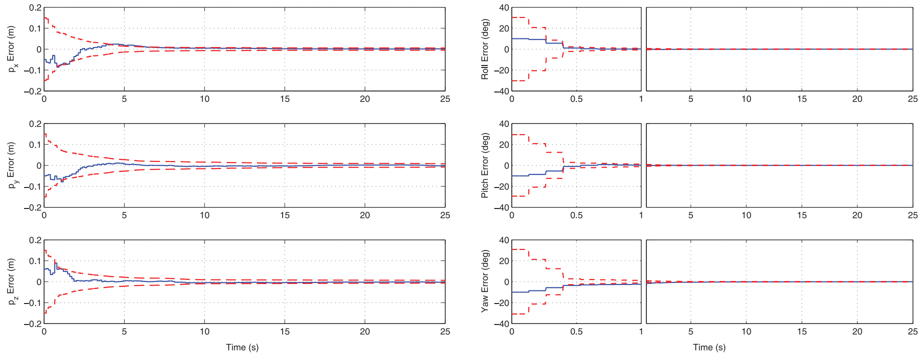
Simulation results for target-free self-calibration are shown in Figure 5 and Tables 3 and 4. We used the same IMU trajectory as for the target-based simulation; however, in this case, we treated the corner points on the calibration target as landmarks with *unknown* positions in the world frame. The landmark positions were initialized using the procedure described in Section 6.3.2; we chose a nominal distance of 2.5 m from the first camera pose for each landmark, with a standard deviation of 0.4 m, along the respective camera ray axis. The upper left and right corner points and the lower left corner point were selected as anchors to lock down the world frame. To reduce any bias in the estimates of the positions of the anchor landmarks, the locations of the corresponding points on the image plane were averaged over 60 seconds (900 image frames), while the beam was stationary, before we started the filtering algorithm.

For the target-free case, we used 375 camera images (twice as many as for target-based calibration), acquired at a rate of 15 Hz instead of 7.5 Hz. We have found that an increase in the frame rate is necessary in many situations



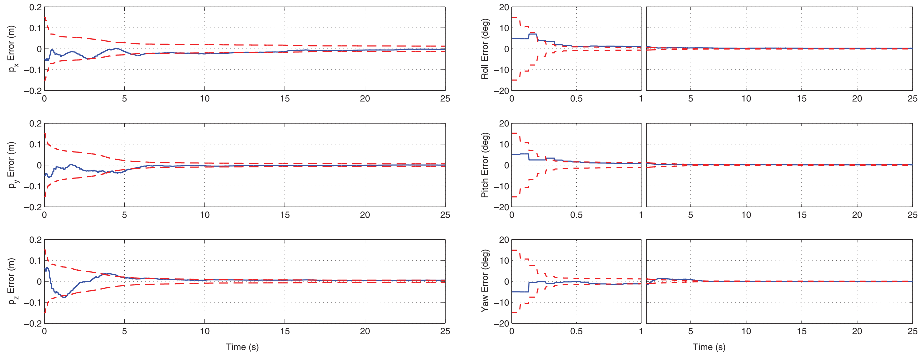
(a) (b)

Fig. 3. Simulation results for target-based calibration trial TB-1, showing the estimation error (solid line) and 3σ bounds (dashed line) for the camera–IMU (a) relative translation and (b) relative orientation. Initial translation and orientation errors were $\tilde{\mathbf{p}}^I_C = [-5 \quad -5 \quad 6]^\mathrm{T}$ cm and $\tilde{\boldsymbol{\xi}}^I_C = [5 \quad 5 \quad -5]^\mathrm{T}$ degrees, respectively.



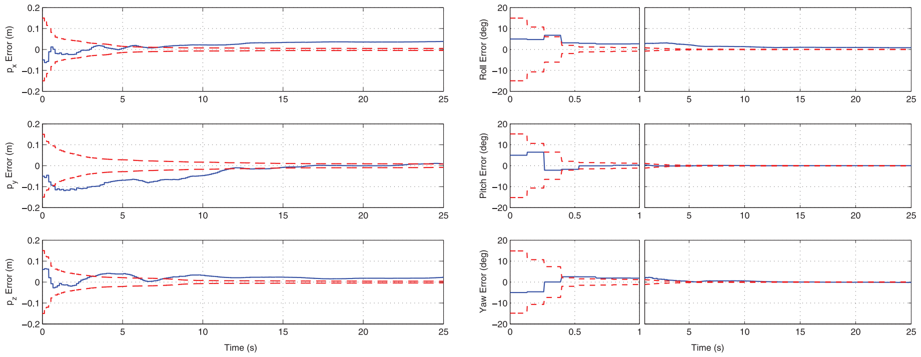
(a) (b)

Fig. 4. Simulation results for target-based calibration trial TB-2. The estimation error and 3σ bounds for the camera–IMU (a) relative translation and (b) relative orientation are shown. Initial translation and orientation errors were $\tilde{\mathbf{p}}^I_C = [-5 \quad -5 \quad 6]^\mathrm{T}$ cm and $\tilde{\boldsymbol{\xi}}^I_C = [10 \quad -10 \quad -10]^\mathrm{T}$ degrees, respectively.



(a) (b)

Fig. 5. Simulation results for target-free self-calibration. The estimation error and 3σ bounds for the camera–IMU (a) relative translation and (b) relative orientation are shown. Initial translation and orientation errors were $\tilde{\mathbf{p}}^I_C = [-5 \quad -5 \quad 6]^\mathrm{T}$ cm and $\tilde{\boldsymbol{\xi}}^I_C = [5 \quad 5 \quad -5]^\mathrm{T}$ degrees, respectively.



(a) (b)

Fig. 6. Simulation results for target-based calibration. The estimation error and 3σ bounds for the camera–IMU (a) relative translation and (b) relative orientation are shown. Gravity was *not* estimated as part of the calibration procedure in this case; the gravity vector was intentionally misaligned by 2° relative to the z-axis of the calibration target. Initial translation and orientation errors were $\tilde{\mathbf{p}}^I_C = [-5 \quad -5 \quad 6]^{\mathrm{T}}$ cm and $\tilde{\boldsymbol{\xi}}^I_C = [5 \quad 5 \quad -5]^{\mathrm{T}}$ degrees, respectively. Note that there are steady-state translation errors along p_x and p_z.

Table 1. Simulation Results for Target-based Calibration of the Camera–IMU Relative Translation. The Initial and Final Estimation Errors are Listed, Along with Their Respective 3σ Error Bounds.

Simulation		p_x Error $\pm 3\sigma$ (cm)	p_y Error $\pm 3\sigma$ (cm)	p_z Error $\pm 3\sigma$ (cm)
TB-1	Initial	-5.00 ± 15.00	-5.00 ± 15.00	6.00 ± 15.00
	Final	-0.01 ± 0.53	-0.28 ± 0.62	-0.32 ± 0.70
TB-2	Initial	-5.00 ± 15.00	-5.00 ± 15.00	6.00 ± 15.00
	Final	0.13 ± 0.53	-0.19 ± 0.79	-0.27 ± 0.65

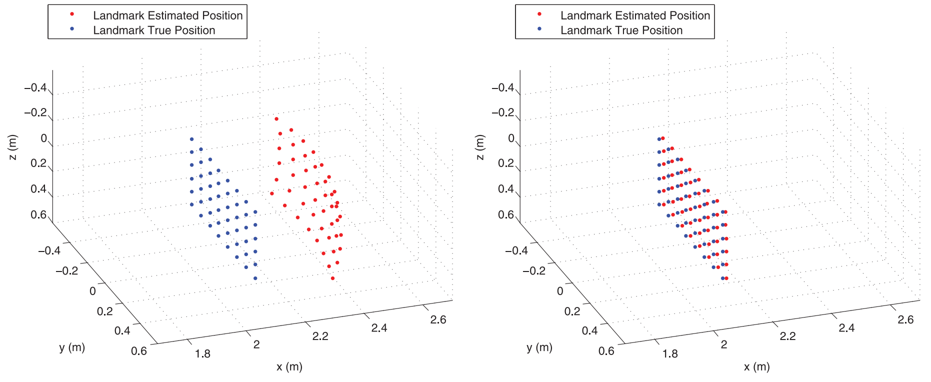
Table 2. Simulation Results for Target-based Calibration of the Camera–IMU Relative Orientation. The Initial and Final Estimation Errors are Listed, Along with Their Respective 3σ Error Bounds

Simulation		Roll Error ± 3σ (°)	Pitch Error ± 3σ (°)	Yaw Error ± 3σ (°)
TB-1	Initial	5.00 ± 15.00	5.00 ± 15.00	-5.00 ± 15.00
	Final	-0.02 ± 0.06	-0.06 ± 0.06	0.05 ± 0.05
TB-2	Initial	-10.00 ± 30.00	-10.00 ± 30.00	10.00 ± 30.00
	Final	-0.04 ± 0.10	-0.01 ± 0.10	-0.03 ± 0.08

because of the large number of DOFs in the system. This is particularly important when the initial orientation error is large[14].

The final estimates of the calibration parameters are very close to the values obtained for target-based calibration. The residual orientation errors are less than 0.2° in roll, pitch, and yaw after 25 seconds. The translation errors, shown in Figure 5, decrease to less than 6 mm along each axis, with an uncertainty (3σ) of ±12 mm or less within the 25-second simulation time interval.

To verify our ability to accurately estimate the local scene structure, we compared the estimated positions of the landmark points with their actual positions, after the filter had run for 25 seconds. The results are shown in Figure 7. The root mean square (RMS) error between the true and estimated landmark positions is only 1.39 cm, and the majority of this error is along the depth direction. By running the calibration algorithm for a longer period of time, it is possible to refine these estimates and further reduce the RMS error.



(a) (b)

Fig. 7. Simulation results for target-free self-calibration. Estimated (a) initial positions of landmarks in the world frame and (b) positions of landmarks after the calibration algorithm has run for 25 s. The RMS error between the true and estimated positions of the 48 landmarks is 1.39 cm, after 25 s.

Table 3. Simulation Results for Target-free Self-calibration of the Camera–IMU Relative Translation. The Initial and Final Estimation Errors are Listed, Along with Their Respective 3σ Error Bounds

Simulation		p_x Error $\pm 3\sigma$ (cm)	p_y Error $\pm 3\sigma$ (cm)	p_z Error $\pm 3\sigma$ (cm)
TF	Initial	-5.00 ± 15.00	-5.00 ± 15.00	6.00 ± 15.00
	Final	-0.36 ± 1.22	-0.01 ± 0.56	0.57 ± 0.60

Table 4. Simulation Results for Target-free Self-calibration of the Camera–IMU Relative Orientation. The Initial and Final Estimation Errors are Listed, Along with Their Respective 3σ Error Bounds

Simulation		Roll Error $\pm 3\sigma$ (°)	Pitch Error $\pm 3\sigma$ (°)	Yaw Error $\pm 3\sigma$ (°)
TF	Initial	5.00 ± 15.00	5.00 ± 15.00	-5.00 ± 15.00
	Final	0.19 ± 0.20	0.12 ± 0.14	-0.14 ± 0.15

7.3. *Gravity and Calibration*

We ran an additional simulation to determine the effect of *not* estimating gravity as part of the calibration procedure, when there is a small misalignment between the true gravity vector and the vector used by the filtering algorithm. For target-based calibration, this situation occurs when the vertical axis of the target is not exactly parallel to the local $\mathbf{g}^W$ vector.

In the simulation, we set the initial translation and orientation errors to the same values used for the first target-based trial, $\tilde{\mathbf{p}}^I_C = [-5 \quad -5 \quad 6]^{\mathrm{T}}$ cm and $\tilde{\boldsymbol{\xi}}^I_C = [5 \quad 5 \quad -5]^{\mathrm{T}}$ degrees, as described in Section 7.1. We then rotated the gravity vector through a pitch angle of 2°, which resulted in a component of $\mathbf{g}$ acting along the positive x-axis in the world frame. We note that misalignments of the order of 1–2° occur often in practice.

The time evolution of the camera–IMU relative translation and orientation estimates are shown in Figure 6. Note that roll and yaw orientation estimates are initially inconsistent. The pitch and yaw estimates do converge (within the estimation uncertainty) after approximately 5 seconds. However, there is a small residual bias in the roll estimate, even after 25 seconds. The relative translation estimates, in contrast, remain biased: both p_x and p_z diverge from the true translation values. Although the value of p_y does eventually converge, there is a constant bias of 3.8 cm along p_x and 2.3 cm along p_z.

An analysis of these results indicates that the constant translation biases exists because the filter is able to compensate for the erroneous orientation of the gravity vector by either a) adjusting the accelerometer bias, or b) introducing a constant translational offset in the camera position. The initial uncertainty in the accelerometer bias is small relative to the uncertainty in the p_x and p_z translation values in this case. As the camera–IMU translation covariance decreases, the translation parameters converge to the incorrect, biased values, after which they are insensitive to further changes.

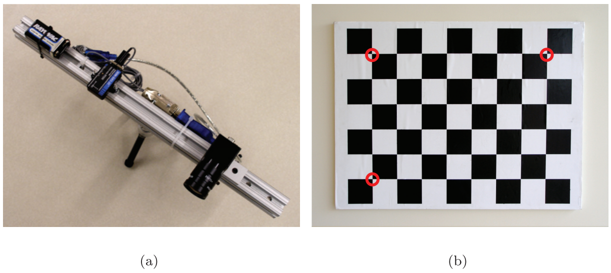


Fig. 8. (a) Sensor beam, showing Flea camera (lower right) and 3DM-GX3 IMU (center left). The beam is 30 cm in length. (b) Our planar camera calibration target. Each target square is 10.4 cm on a side. There are 48 interior corner points, which we use as landmarks for calibration. The small circles in the figure identify the three points that we use as anchors for the self-calibration experiments described in Section 8.1.

8. Experiments

We performed a series of experiments to quantify the accuracy and performance of the target-based calibration and the target-free self-calibration algorithms with real hardware. Our test rig consists of a 30-cm long 8020 aluminum beam, with an IMU mounted near the center and a camera mounted at one end (cf. Figure 8(a)). The camera is a black and white Flea FireWire model from Point Grey Research (640 × 480 pixel resolution), mated to a 4 mm Navitar lens (58° horizontal FOV, 45° vertical FOV). Our IMU is a MEMS-based 3DM-GX3 unit, manufactured by MicroStrain, which provides three-axis angular rate and linear acceleration measurements at 100 Hz. Axis scale and non-orthogonality effects are compensated for internally by the IMU.

In the following sections, we present results from three different experiments: one target-based calibration experiment and two target-free calibration experiments. These results are a representative sample of the performance of the calibration algorithms across a large number of trials.

8.1. *Experimental Procedure*

For the target-based experiments, we placed the sensor beam in front of a stand-mounted planar camera calibration target in our laboratory, such that the entire target was visible in the camera image. Our planar calibration target is 100 cm × 80 cm in size, with 48 (8 × 6) interior corner points that are each 10.4 cm apart.

At the start of an experiment, we initialized the filter biases while holding the sensor beam still for approximately 1 minute. After this settling time, we moved the beam manually through a series of rotation and translation maneuvers, at distances between 0.5 m and 2.5 m from the target. Typically, approximately 50–60 seconds of data are required for accurate calibration, with the sensor beam rotating and translating during this time.

Table 5. Results for Target-based Calibration and Target-free Self-calibration of the Camera–IMU Relative Translation. The Initial Hand-measured (HM) Estimate of the Translation and the Final Target-based (TB) and Target-free, Self-calibrated (TF-1, TF-2) Estimates are Listed, Along with Their Respective 3σ Error Bounds

Trial	$p_x \pm 3\sigma$ (cm)	$p_y \pm 3\sigma$ (cm)	$p_z \pm 3\sigma$ (cm)
HM	0.00 ± 6.00	15.00 ± 15.00	0.00 ± 9.00
TB	-3.14 ± 0.98	16.60 ± 0.64	-2.85 ± 0.70
TF-1	-3.57 ± 0.98	16.56 ± 0.64	-2.98 ± 0.69
TF-2	-3.97 ± 0.43	16.56 ± 0.23	-2.89 ± 0.24

Table 6. Results for Target-based Calibration and Target-free Self-calibration of the Camera–IMU Relative Orientation. The Initial Hand-measured (HM) Estimate of the Orientation (Roll, Pitch, and Yaw Angles) and the Final Target-based (TB) and Target-free, Self-calibrated (TF-1, TF-2) Estimates are Listed, Along with Their Respective 3σ Error Bounds

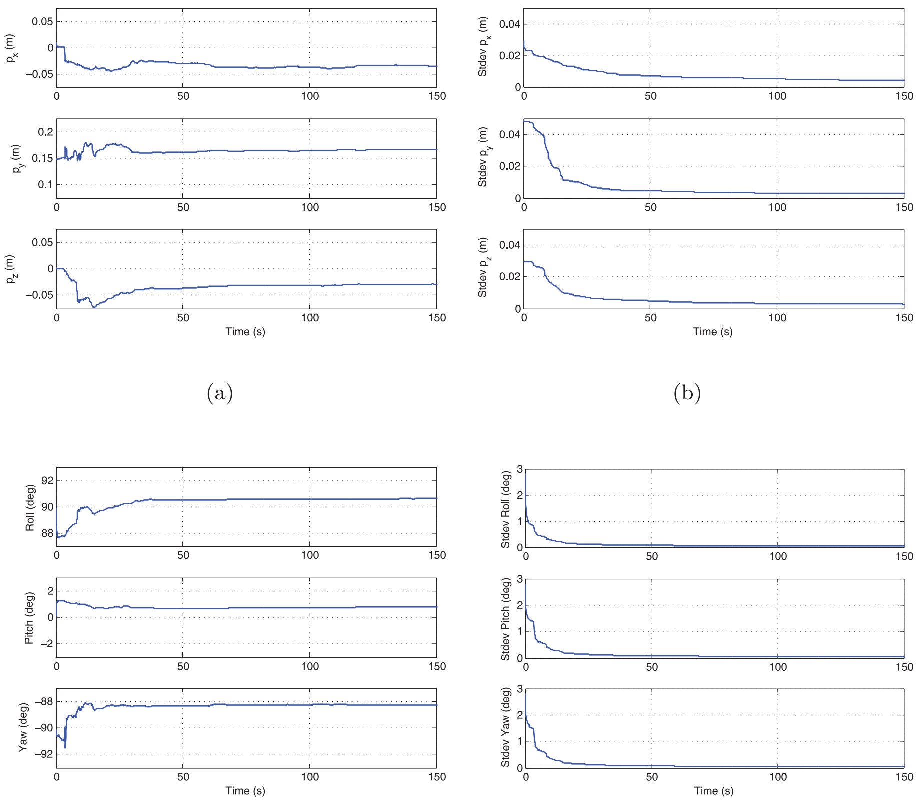
Trial	Roll $\pm 3\sigma$ (°)	Pitch $\pm 3\sigma$ (°)	Yaw $\pm 3\sigma$ (°)
HM	90.00 ± 15.00	0.00 ± 15.00	-90.00 ± 15.00
TB	90.62 ± 0.10	0.78 ± 0.11	-88.29 ± 0.10
TF-1	90.57 ± 0.11	0.70 ± 0.12	-88.33 ± 0.09
TF-2	90.60 ± 0.06	0.77 ± 0.06	-88.39 ± 0.06

The camera–IMU transform parameters were initialized using hand measurements of the relative position and orientation of the sensors. We used a subset of 25 images acquired during the camera–IMU procedure to calibrate the camera intrinsic parameters. We assume that each image measurement is corrupted by independent white Gaussian noise with a standard deviation of 2.0 pixels along the u and v image axes.

Self-calibration involves first anchoring the world frame by fixing the directions to three points on the image plane (Chiuso et al. 2002). For our self-calibration experiments using the calibration target, we chose to fix the directions to the upper left, upper right, and lower left points (as shown in Figure 8(b)). We computed the three-dimensional covariances for these points using very small image plane uncertainties, after averaging the image coordinates over 400 frames to reduce noise. Applying standard linearized error propagation, this effectively meant that only the depths of the points were undetermined. We selected an initial depth of 3.0 m for each of the 48 points on the target, along the corresponding camera rays, with a standard deviation of 0.8 m.

8.2. *Target-based Calibration*

We initially evaluated the accuracy of the camera–IMU calibration algorithm using data acquired while the camera viewed our (known) planar calibration target. This provided



(a)

(b)

(c)

(d)

Fig. 9. Experimental results for target-based calibration (TB). The 48 interior corner points on the calibration target were treated as *known* landmarks in this case. (a) Estimate of the relative translation of the camera along the IMU *x*-, *y*-, and *z*-axes, and (b) the corresponding standard deviations, over time. (c) Estimate of the relative orientation (roll, pitch, and yaw angles) of the camera frame with respect to the IMU frame, and (d) the corresponding standard deviations, over time.

a set of reference calibration values for comparison with the values obtained from the target-free procedure. Images were acquired[15] at a rate of 7.5 Hz. The data set included 15,025 IMU measurements and 1,123 image frames; the 48 corner points on the calibration target were completely visible in all but three of the frames. We simply discarded these frames, leaving 1,120 frames for use by the calibration algorithm.

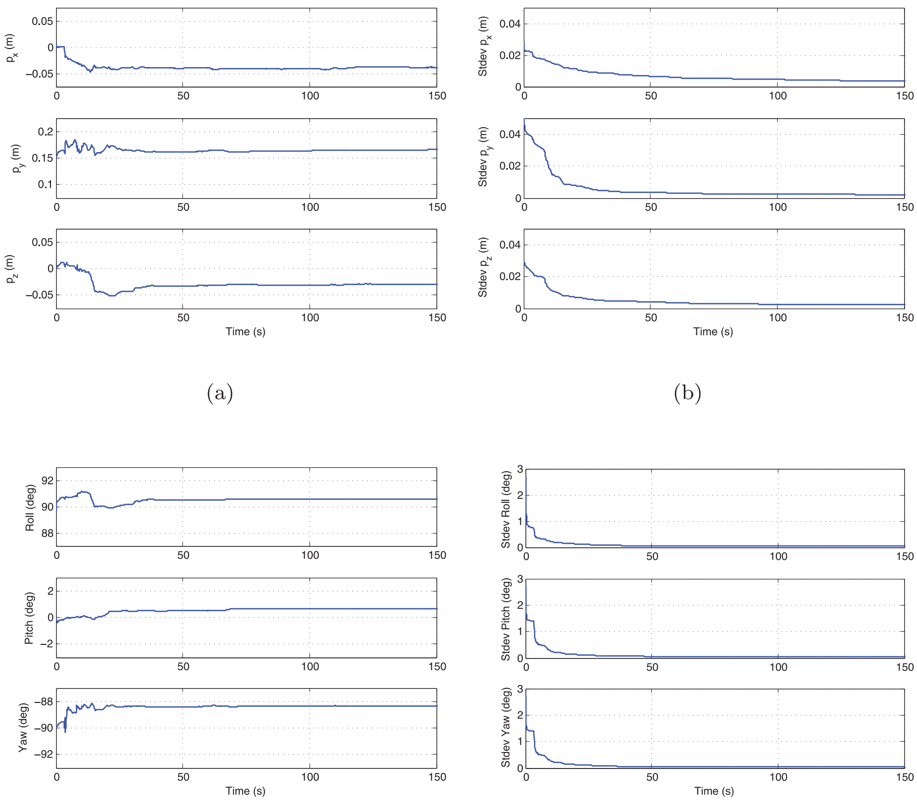
Calibration results are shown in Figure 9 and in the first row of Tables 5 and 6 (designated as “TB”). The plots indicate that both the translation and orientation parameters converge to their steady-state values within approximately 50 seconds. The final uncertainty for the parameter estimates is on the order of 0.1° for orientation and less than 1 cm for translation.

Importantly, we note that the final estimate of the gravity vector in the world frame is $\mathbf{g}^W = [-0.150 \quad 0.038 \quad 9.790]$ m s^{-2}, with magnitude $\| \mathbf{g}^W \| = 9.791$ m s^{-2}. The magnitude of the vector agrees with the known value for **g** in Los Angle of 9.796 m s^{-2}. However, the gravity vector has a pitch angle of approximately 0.9° relative to the *z*-axis of the calibration target. This misalignment was present despite our best efforts to orient the target exactly vertically (using a bubble level), and indicates the importance of estimating the gravity vector as part of the calibration process.

8.3. *Target-free Calibration*

We performed two separate target-free calibration experiments, and compared the results with those from the target-based procedure. For the first target-free experiment, we used the same target-based data set (“TB”), but treated the landmark positions as *unknown* initially[16]. In this case, we also increased the frame rate to 15 Hz. The (updated) data set (designated as “TF-1”) included 15,025 IMU measurements and 2,225 image frames; the 48 corner points on the calibration target were completely visible in all but five of these frames (which we discarded).

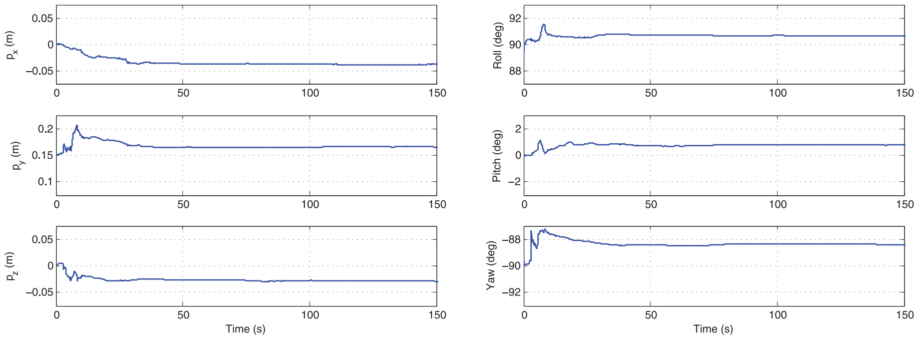
Calibration results are shown in Figure 10 and in the second row of Tables 5 and 6 (designated as “TF-1”). The plots indicate that both the translation and orientation parameters converge to their steady-state values within approximately 50 seconds. As a measure of the accuracy of the structure estimate, we computed the best-fit alignment



(a) (b)

(c) (d)

Fig. 10. Experimental results for target-free calibration (TF-1). The 48 interior corner points on the calibration target were treated as *unknown* landmarks in this case. (a) Estimate of the relative translation of the camera along the IMU *x*-, *y*-, and *z*-axes, and (b) the corresponding standard deviations, over time. (c) Estimate of the relative orientation (roll, pitch, and yaw angles) of the camera frame with respect to the IMU frame, and (d) the corresponding standard deviations, over time.

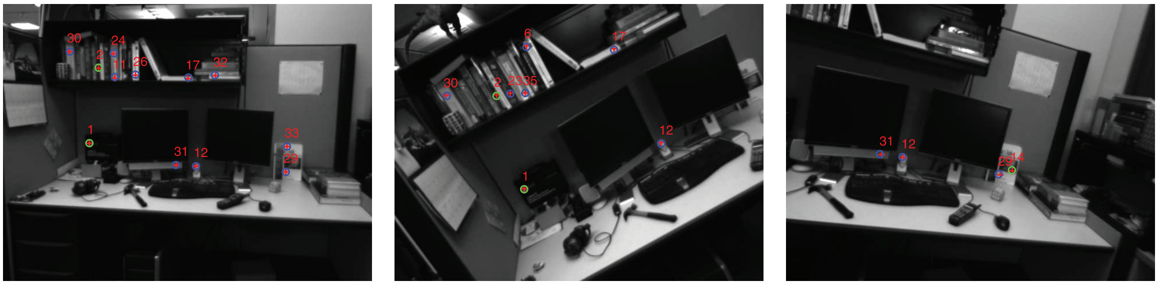


(a) (b)

Fig. 11. Experimental results for target-free calibration using the “desk” data set (TF-2). A total of 40 landmarks were tracked. (a) Estimate of the relative translation of the camera along the IMU *x*-, *y*-, and *z*-axes, over time. (b) Estimate of the relative orientation (roll, pitch, and yaw angles) of the camera frame with respect to the IMU frame, over time.

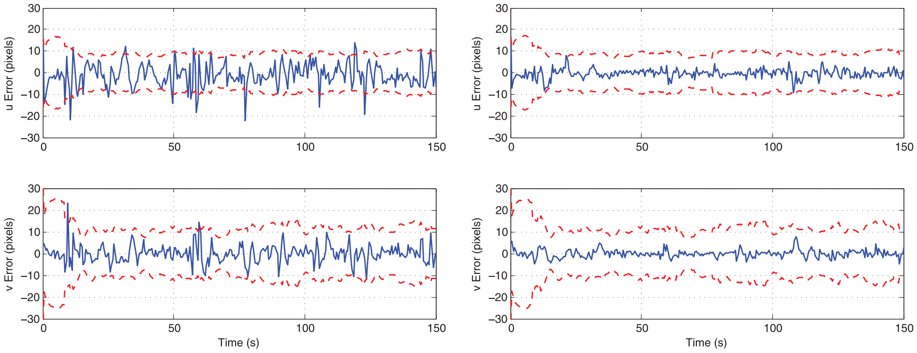
between the known target corner points and the positions estimated by the calibration algorithm using iterative, non-linear least squares. The RMS error was only 6.1 mm, over the 48 points.

As ground truth measurements of the camera–IMU transform were not available, we chose instead to evaluate the calibration accuracy by comparing motion estimation results. We ran the target-based algorithm twice, *without*



(a) (b) (c)

Fig. 12. Images from the target-free ‘desk’ data set (TF-2), at times (a) $t = 5.4$ s, (b) $t = 59.5$ s and (c) $t = 109.4$ s. The measured image locations of the identified landmarks are shown as red crosses, and the corresponding predicted locations (based on the current state estimate) are shown as blue circles. Green circles identify anchor landmarks, which we used to fix the orientation of the world frame.



(a) (b)

Fig. 13. Pixel residuals for camera–IMU motion estimation using (a) hand-measured transform parameters, and (b) self-calibrated transform parameters. Note that the residuals for the calibrated case are more than two times smaller, on average.

estimating the calibration parameters: for the first trial, we used our initial hand-measured estimate of the camera–IMU transform, while for the second trial we used the calibrated TF-1 estimate. To emphasize the importance of accurate calibration, we also reduced the camera frame rate to 1.875 Hz. We then recorded and plotted the measurement residuals at each time step, along the u and along the v image plane axes, for both cases, as shown in shown in Figures 13(a) and (b). Note that the RMS residual for the calibrated case is 3.08 pixels, which is less than half of the RMS value of 7.10 pixels for the hand-measured estimate.

For the second target-free experiment, we ran the calibration algorithm using data acquired from a scene with unknown structure (a cluttered desk in our laboratory). Several frames from this sequence (designated as “TF-2”) are shown in Figure 12. We extracted 40 well-localized SIFT features from the first camera image, and used the corresponding landmarks for calibration. The SIFT features were selected automatically, based on their distribution in the first image and their frequent appearance in the next 150 frames (10 seconds of calibration data).

We used exactly the same number of image frames and IMU measurements for the TF-1 and TF-2 trials. However, for the desk data set, only 19.5 features were visible per frame, on average. This is a significantly lower number (less than half) than for the TF-1 trial. Despite the smaller number of visible features, we note that the calibration results are almost identical to those for our target-based experiment (except for the p_x translation value, which we discuss below). This shows that it is possible to recover accurate calibration in *unknown environments* and without *any* additional hardware.

The only significant difference between the results obtained across all three trials is the value of the p_x translation parameter. It is more difficult to obtain an accurate estimate for p_x because of the camera’s limited FOV, its position relative to the IMU, and the fact that we typically calibrate using landmarks which lie in a constrained region in front of the camera. We are currently investigating the use of a wider distribution of landmarks, in order to better calibrate the p_x translational offset. Also, we note, that for most navigation applications (which involve moving primarily forward through the world), the p_x offset has significantly less influence on the accuracy of motion estimates than the other translation parameters. That is, p_x is more difficult to calibrate precisely *because* its exact value does not usually have a large effect on motion estimation accuracy.

9. Conclusions and Future Work

In this paper, we have presented a localization, mapping, and self-calibration algorithm for visual and inertial sensors. The algorithm employs an UKF to estimate the relative pose of the sensors, the metric scene structure, and the motion of the IMU over time. Our results show that it is possible to accurately calibrate the sensor-to-sensor transform *without* using a known calibration target or other calibration object. This work is a step towards building power-on-and-go robotic systems that are able to self-calibrate in the field. In addition, our approach enables rapid re-calibration when the poses of the sensors must be changed (e.g. to enable operation in different environments).

There are several directions for future research. We are currently investigating how the trajectory of the camera–IMU platform affects calibration accuracy and convergence time, in an effort to define trajectories which quickly produce accurate calibration results. We are also planning to extend the algorithm to include calibration of the camera intrinsic and lens distortion parameters. Finally, we are working to deploy our visual–inertial calibration system for several platforms, including an unmanned aerial vehicle, an autonomous underwater vehicle and a humanoid robot.

Acknowledgements

This work was funded in part by the National Science Foundation under grants CNS-0619937, CNS-0325875, CNS-0520305, and CCF-0120778 and gifts from the Okawa Foundation and Futtock Foundation. Jonathan Kelly was supported in part by an Annenberg Fellowship from the University of Southern California and by an NSERC PGS-D award from the Government of Canada.

Notes

1. We use the terms *sensor-to-sensor calibration* and *relative pose calibration* synonymously throughout the paper.
2. Note that our technique could also be adapted for use in a batch framework.
3. We use the symbols **S** and **D** as abbreviations for the Latin terms *sinister* (left) and *dexter* (right).
4. For clarity, and in a slight abuse of our notation, we do not use boldface to identify unit quaternions.
5. In fact, the world frame is not strictly an inertial frame, since it is attached to the surface of the rotating Earth. The effects of the Earth's rotation are very small over the calibration time interval, however, and we ignore them in our treatment here.
6. In summary, "if the states are different, the measurements are different".
7. The tangent bundle of M is the disjoint union of all of the tangent spaces of M.
8. For brevity, we use the notation $L^1_{\breve{\mathbf{f}}_1}\mathbf{h}_1$ for the three individual column Lie derivatives of $\breve{\mathbf{f}}_1$, and $\nabla L^1_{\breve{\mathbf{f}}_1}\mathbf{h}_1$ for their stacked gradients.
9. We fully expand the matrices $\mathbf{G}_i(\cdot)$ in the Appendix.
10. The UKF estimate is accurate to the third order of the Taylor series expansion of the posterior state distribution.
11. Here, we assume that the target is positioned in front of and approximately parallel to the camera image plane.
12. We use $\tilde{\boldsymbol{\xi}}^{I}_{C}$ to denote a 3×1 vector of Euler angles, which are more easily visualized than the equivalent MRP representation.
13. We have expanded the time axis in the plots over the interval from $t = 0$ to $t = 1.0$ seconds to more clearly show the convergence of the orientation estimate.
14. The relationship between the IMU trajectory, the initial camera–IMU orientation error, and required camera frame rate is complex in general. We are currently investigating this relationship in more detail.
15. We logged camera data at a rate of 15 Hz, however, for target-based calibration we used every second image only.
16. It is important to reemphasize that the self-calibration algorithm *does not require* a calibration target. We used the target only to compare the accuracy of the approaches, with known ground truth available.

References

Britting, K. R. (1971). *Inertial Navigation Systems Analysis*. New York, John Wiley & Sons.

Chatfield, A. B. (1997). *Fundamentals of High Accuracy Inertial Navigation* (*Progress in Astronautics and Aeronautics*, Vol. 174). Reston, VA, American Institute of Aeronautics and Astronautics.

Chiuso, A., Favaro, P., Jin, H. and Soatto, S. (2002). Structure from motion causally integrated over time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4): 523–535.

Chou, J. C. K. (1992). Quaternion kinematic and dynamic differential equations. *IEEE Transactions on Robotics and Automation*, 8(1): 53–64.

Conte, G., Moog, C. H. and Perdon, A. M. (2006). *Algebraic Methods for Nonlinear Control Systems*, 2nd edition. Berlin, Springer.

Crassidis, J. L. and Markely, F. L. (2003). Unscented filtering for spacecraft attitude estimation. *Proceedings of the AIAA Guidance, Navigation and Control Conference (GN&C'03)*, Austin, TX, paper AIAA-2003-5484.

Davison, A. J., Reid, I. D., Molton, N. D. and Stasse, O. (2007). MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6): 1052–1067.

Eade, E. and Drummond, T. (2006). Scalable monocular SLAM. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, New York, Vol. 1, pp. 469–476.

Farrell, J. A. (2008). *Aided Navigation: GPS with High Rate Sensors* (*Electronic Engineering*). New York, McGraw-Hill.

Farrell, J. A. and Barth, M. (1998). *The Global Positioning System and Inertial Navigation*, 1st edition. New York, McGraw-Hill.

Foxlin, E. M. (2002). Generalized architecture for simultaneous localization, auto-calibration, and map-building. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'02)*, Lausanne, Switzerland, Vol. 1, pp. 527–533.

Gemeiner, P., Einramhof, P. and Vincze, M. (2007). Simultaneous motion and structure estimation by fusion of inertial and vision data. *The International Journal of Robotics Research*,26(6): 591–605.

Golub, G. H. and Loan, C. F. V. (1996). *Matrix Computations*, 3rd edition (*Johns Hopkins Studies in Mathematical Sciences*). Baltimore, MD, The Johns Hopkins University Press.

Goshen-Meskin, D. and Bar-Itzhack, I. Y. (1992). Observability analysis of piece-wise constant systems – part I: Theory. *IEEE Transactions on Aerospace and Electronic Systems* 28(4): 1056–1067.

Hermann, R. and Krener, A. J. (1977). Nonlinear controllability and observability. *IEEE Transactions on Automatic Control* AC-22(5): 728–740.

Hol, J. D., Schon, T. B., and Gustafsson, F. (2008). Relative pose calibration of a spherical camera and an IMU. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR'08)*, 21–24. Cambridge, United Kingdom.

Horn, B. K. P. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A* 4(4): 629–642.

Huster, A. and Rock, S. M. (2003). Relative position sensing by fusing monocular vision and inertial rate sensors. *Proceedings of the 11th International Conference on Advanced Robotics (ICAR'03)*, Coimbra, Portugal, Vol. 3, pp. 1562–1567.

Isidori, A. (1995). *Nonlinear Control Systems*, 3rd edition (*Communications and Control Engineering*). Berlin, Springer.

Jones, E., Vedaldi, A. and Soatto, S. (2007). Inertial structure from motion with autocalibration. *Proceedings of the IEEE International Conference on Computer Vision Workshop on Dynamical Vision*, Rio de Janeiro, Brazil.

Julier, S. J. (2002). The scaled unscented transform. *Proceedings of the IEEE American Control Conference (ACC'02)*, Anchorage, AK, Vol. 6, pp. 4555–4559.

Julier, S. J. and LaViola, J. J. (2007). On Kalman filtering with nonlinear equality constraints. *IEEE Transactions on Signal Processing*, 55(6): 2774–2784.

Julier, S. J. and Uhlmann, J. K. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3): 401–422.

Kelly, J., Saripalli, S. and Sukhatme, G. S. (2008). Combined visual and inertial navigation for an unmanned aerial vehicle. *Field and Service Robotics: Results of the 6th International Conference (FSR'07)*, Laugier, C. and Siegwart, R. (eds) (*Springer Tracts in Advanced Robotics*, Vol. 42/2008). Berlin, Springer, pp. 255–264.

Klein, G. and Murray, D. (2007). Parallel tracking and mapping for small AR workspaces. *Proceedings of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, pp. 225–234.

Kraft, E. (2003). A quaternion-based unscented Kalman filter for orientation tracking. *Proceedings of the 6th International Conference on Information Fusion*, Cairns, Australia, Vol. 1, 47–54.

Kuipers, J. B. (2002). *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality*. Princeton, NJ, Princeton University Press.

Lang, P. and Pinz, A. (2005). Calibration of hybrid vision/inertial tracking systems. *Proceedings of the 2nd Workshop on Integration of Vision and Inertial Sensors (INERVIS'05)*, Barcelona, Spain.

Lee, T. S., Dunn, K. P. and Chang, C. B. (1982). On the observability and unbiased estimation of nonlinear systems. *System Modeling and Optimization: Proceedings of the 10th IFIP Conference* (*Lecture Notes in Control and Information Sciences*, Vol. 38/1982). Berlin, Springer, pp. 258–266.

Lobo, J. and Dias, J. (2007). Relative pose calibration between visual and inertial sensors. *The International Journal of Robotics Research*, 26(6): 561–575.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2(60): 91–110.

Lucchese, L. and Mitra, S. K. (2002). Using saddle points for subpixel feature detection in camera calibration targets. *Proceedings of the Asia-Pacific Conference on Circuits and Systems (APCCAS'02)*, Singapore, Vol. 2, pp. 191–195.

Ma, Y., Soatto, S., Košecká, J. and Sastry, S. (2004). *An Invitation to 3-D Vision: From Images to Geometric Models*, 1st edition (*Interdisciplinary Applied Mathematics*, Vol. 26) Berlin, Springer.

Martinelli, A., Scaramuzza, D. and Siegwart, R. (2006a). Automatic self-calibration of a vision system during robot motion. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'06)*, Orlando, FL, pp. 43–48.

Martinelli, A. and Siegwart, R. (2006). Observability properties and optimal trajectories for on-line odometry self-calibration. *Proceedings of the IEEE Conference on Decision and Control*, San Diego, CA, pp. 3065–3070.

Martinelli, A., Weingarten, J., and Siegwart, R. (2006b). Theoretical results on on-line sensor self-calibration. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'06)*, Beijing, China, pp. 43–48.

Maybeck, P. S. (1979). *Stochastic Models, Estimation and Control* (*Mathematics in Science and Engineering*, Vol. 141-1). New York, Academic Press.

Mirzaei, F. M. and Roumeliotis, S. I. (2008). A Kalman filter-based algorithm for IMU-camera calibration: Observability analysis and performance evaluation. *IEEE Transactions on Robotics*, 24(5): 1143–1156.

Montiel, J. M. M., Civera, J. and Davison, A. J. (2006). Unified inverse depth parametrization for monocular SLAM. *Proceedings of Robotics: Science and Systems (RSS'06)*, Philadelphia, PA.

Mourikis, A. I. and Roumeliotis, S. I. (2007). A multi-state constraint Kalman filter for vision-aided inertial navigation. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'07)*, Rome, Italy, pp. 3565–3572.

Mourikis, A. I., Trawny, N., Roumeliotis, S. I., Johnson, A. E., Ansar, A. and Matthies, L. H. (2009). Vision-aided inertial navigation for spacecraft entry, descent, and landing. *IEEE Transactions on Robotics*, 25(2): 264–280.

Rehbinder, H. and Ghosh, B. K. (2003). Pose estimation using line-based dynamic vision and inertial sensors. *IEEE Transactions on Automatic Control*, 48(2): 186–199.

Sastry, S. (1999). *Nonlinear Systems: Analysis, Stability and Control* (*Interdisciplinary Applied Mathematics*, Vol. 10). Berlin, Springer.

Stevens, B. L. and Lewis, F. L. (2003). *Aircraft Control and Simulation*, 2nd edition. New York, Wiley-Interscience.

Strelow, D. (2004). *Motion Estimation from Image and Inertial Measurements*. PhD Thesis, Carnegie Mellon University, Pittsburgh, PA.

Strelow, D. and Singh, S. (2002). Optimal motion estimation from visual and inertial data. *Proceedings of the Sixth IEEE Workshop on the Applications of Computer Vision (WACV'02)*, Orlando, FL, pp. 314–319.

Strelow, D. and Singh, S. (2003). Online motion estimation from visual and inertial measurements. *Proceedings of the 1st Workshop on Integration of Vision and Inertial Sensors (INERVIS'03)*, Coimbra, Portugal.

Titterton, D. and Weston, J. (2004). *Strapdown Inertial Navigation Technology*, 2nd edition (*IEE Radar, Sonar and Navigation Series*, 17(5)). London, The Institution of Electrical Engineers.

van der Merwe, R. and Wan, E. A. (2004). Sigma-point Kalman filters for integrated navigation. *Proceedings of the 60th Annual Meeting of The Institute of Navigation (ION)*, Dayton, OH, pp. 641–654.

Vedaldi, A. and Fulkerson, B. (2009). VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/.

Verhaegen, M. and Verdult, V. (2007). *Filtering and System Identification: A Least Squares Approach*. Cambridge, Cambridge University Press.

Woods, D. W. (2008). *How Apollo Flew to the Moon*. New York, Springer Praxis Books.

Appendix: Lemmas and Their Proofs

We use curly braces, { and }, to denote the submatrix formed from several rows of a larger matrix, e.g. $\mathbf{B}\{1,2,6\}$ for the submatrix formed from rows 1, 2, and 6 of matrix **B**. The determinant of a matrix is denoted by a set of vertical bars, e.g. $|\mathbf{A}|$.

Lemma 1. For any unit quaternion $\bar{q}$, the 10×5 matrix:

$$\mathbf{A} = \begin{bmatrix} \alpha\,\mathbf{G}_2(\bar{q}) & \mathbf{G}_3(\bar{q}) \\ 2\,\bar{q}^{\mathrm{T}} & 0 \end{bmatrix} \tag{93}$$

has full column rank when α is non-zero and at least two components of $\mathbf{a}_m = [a_x \;\; a_y \;\; a_z]^{\mathrm{T}}$ are excited.

Proof. Let $\bar{q} = \left[q_0 \;\; \mathbf{q}^{\mathrm{T}}\right]^{\mathrm{T}} = [q_0 \;\; q_1 \;\; q_2 \;\; q_3]^{\mathrm{T}}$. The rows of **A** correspond to the individual components of the measured linear acceleration vector $\mathbf{a}_m$, as follows:

$$\mathbf{A} = \left[\begin{array}{ccccc} 0 & 0 & -4\alpha q_2 & -4\alpha q_3 & 1-2q_2^2-2q_3^2 \\ 2\alpha q_3 & 2\alpha q_2 & 2\alpha q_1 & 2\alpha q_0 & 2q_0q_3+2q_1q_2 \\ -2\alpha q_2 & 2\alpha q_3 & -2\alpha q_0 & 2\alpha q_1 & 2q_1q_3-2q_0q_2 \\ -2\alpha q_3 & 2\alpha q_2 & 2\alpha q_1 & -2\alpha q_0 & 2q_1q_2-2q_0q_3 \\ 0 & -4\alpha q_1 & 0 & -4\alpha q_3 & 1-2q_1^2-2q_3^2 \\ 2\alpha q_1 & 2\alpha q_0 & 2\alpha q_3 & 2\alpha q_2 & 2q_0q_1+2q_2q_3 \\ 2\alpha q_2 & 2\alpha q_3 & 2\alpha q_0 & 2\alpha q_1 & 2q_0q_2+2q_1q_3 \\ -2\alpha q_1 & -2\alpha q_0 & 2\alpha q_3 & 2\alpha q_2 & 2q_2q_3-2q_0q_1 \\ 0 & -4\alpha q_1 & -4\alpha q_2 & 0 & 1-2q_1^2-2q_2^2 \\ 2q_0 & 2q_1 & 2q_2 & 2q_3 & 0 \end{array}\right] \begin{array}{l} \\ \}a_x \\ \\ \\ \}a_y \\ \\ \\ \}a_z \\ \\ \\ \end{array} \tag{94}$$

where the series of braces to the right of the matrix indicate which components of the control input $\mathbf{a}_m$ must be excited in order to use the respective rows of **A**.

To prove that **A** has full column rank, it is sufficient to show that there always exists a combination of five rows of the matrix for which the determinant is non-zero. Observe that with only one component of $\mathbf{a}_m$ available, a maximum of four rows of **A** can be used, and the rank of **A** will be at most four. So at least two of a_x, a_y, and a_z must be excited.

Without loss of generality, let the control inputs a_x and a_y be excited; we can prove the same result when other possible combinations of control inputs (a_x and a_z, a_y and a_z) are used, and we do not present those cases here. Selecting rows $\{1,2,3,6,10\}$ and $\{1,2,4,5,10\}$, we compute the determinants of the respective submatrices, and after simplification we obtain

$$|\,\mathbf{A}\{1,2,3,6,10\}| = \;\; 16\,\alpha^3\left(q_1^2 - q_0^2 + q_2^2 - q_3^2\right), \tag{95}$$

$$|\,\mathbf{A}\{1,2,4,5,10\}| = -64\,\alpha^3\left(q_0^2 + q_3^2\right)\left(q_1^2 + q_2^2\right). \tag{96}$$

Assume that (95) and (96) are both equal to zero and that $\alpha \neq 0$. If we consider (96), then either $q_0 = q_3 = 0$ or $q_1 = q_2 = 0$. We handle each case in turn.

Case 1: Let $q_0 = q_3 = 0$. Substituting these values in (95) and dividing by $16\alpha^3$, we obtain

$$q_1^2 + q_2^2 = 0, \tag{97}$$

which is zero only if $q_1 = q_2 = 0$. However, $\bar{q}$ is a unit quaternion, and so at least one component must be non-zero.

Case 2: Let $q_1 = q_2 = 0$. Substituting these values in (95) and dividing by $-16\alpha^3$, we obtain

$$q_0^2 + q_3^2 = 0, \tag{98}$$

which is zero only if $q_0 = q_3 = 0$. However, $\bar{q}$ is a unit quaternion, and so (again) at least one component must be non-zero.

The matrix therefore always has full column rank. □

Lemma 2. For any unit quaternion $\bar{q}$, the 4×4 matrix $\mathbf{S}(\bar{q})$ is full rank.

Proof. Let $\bar{q} = \left[q_0 \;\; \mathbf{q}^{\mathrm{T}}\right]^{\mathrm{T}} = [q_0 \; q_1 \; q_2 \; q_3]^{\mathrm{T}}$. The determinant of $\mathbf{S}(\bar{q})$ is

$$\left|\,\mathbf{S}(\bar{q})\right| = \left(q_0^2 + q_1^2 + q_2^2 + q_3^2\right)^2 = \|\,\bar{q}\|^4 = 1, \tag{99}$$

which shows that the matrix is full rank. □

Lemma 3. For any two unit quaternions $\bar{p}$ and $\bar{q}$, the 4×3 matrix $\mathbf{B} = \mathbf{D}(\bar{p})\;\Xi\,(\bar{q})$ has full column rank.

Proof. We use the approach described by Mirzaei and Roumeliotis (2008), and multiply $\mathbf{B}$ by its transpose,

$$\mathbf{B}^{\mathrm{T}}\mathbf{B} = \Xi\,(\bar{q})^{\mathrm{T}}\,\mathbf{D}(\bar{p})^{\mathrm{T}}\,\mathbf{D}(\bar{p})\;\Xi\,(\bar{q}) = \mathbf{I}_3, \tag{100}$$

noting that $\mathbf{D}(\bar{p})^{\mathrm{T}}\,\mathbf{D}(\bar{p}) = \mathbf{I}_4$ and $\Xi\,(\bar{q})^{\mathrm{T}}\;\Xi\,(\bar{q}) = \mathbf{I}_3$. The matrix therefore always has full column rank. □

Lemma 4. For any unit quaternion $\bar{q}$, the 9×3 matrix $\mathbf{G}_1(\bar{q})$ has full column rank when at least two components of $\boldsymbol{\omega}_m = [\omega_x \quad \omega_y \quad \omega_z]^{\mathrm{T}}$ are excited.

Proof. Let $\bar{q} = \left[q_0 \quad \mathbf{q}^{\mathrm{T}}\right]^{\mathrm{T}} = [q_0 \; q_1 \; q_2 \; q_3]^{\mathrm{T}}$ and $\mathbf{v} = [v_x \quad v_y \quad v_z]^{\mathrm{T}}$. The matrix $\mathbf{G}_1(\bar{q})$ is

$$\mathbf{G}_1(\bar{q}) = \left[\begin{array}{c} \dfrac{\partial\,\boldsymbol{\Gamma}\left(\bar{q}_I^W, \mathbf{v}\right)\,\Xi\left(\bar{q}_I^W\right)}{\partial\, v_x} \\ \dfrac{\partial\,\boldsymbol{\Gamma}\left(\bar{q}_I^W, \mathbf{v}\right)\,\Xi\left(\bar{q}_I^W\right)}{\partial\, v_y} \\ \dfrac{\partial\,\boldsymbol{\Gamma}\left(\bar{q}_I^W, \mathbf{v}\right)\,\Xi\left(\bar{q}_I^W\right)}{\partial\, v_z} \end{array}\right]\begin{array}{l} \}\omega_x \\ \\ \}\omega_y, \\ \\ \}\omega_z \end{array} \tag{101}$$

where series of braces to the right of the matrix indicate which components of the control input $\boldsymbol{\omega}_m$ must be excited in order to use the respective rows of $\mathbf{G}_1$.

When at least two of ω_x, ω_y, and ω_z are excited, it is always possible to choose three rows of $\mathbf{G}_1$ (e.g. rows 2, 3, and 8) such that the absolute value of the determinant of the resulting 3×3 matrix is

$$|\,\mathbf{G}_1\{2,3,8\}| = \left(q_0^2 + q_1^2 + q_2^2 + q_3^2\right)^3 = \|\,\bar{q}\|^6 = 1. \tag{102}$$

The matrix therefore always has full column rank. □