

Offline GNSS/Camera Extrinsic Calibration Using RTK and Fiducials

Alex D. Jordan¹, Jacob C. Johnson², Timothy W. McLain¹, *Senior Member, IEEE*,
and Randal W. Beard¹, *Fellow, IEEE*

Abstract—Accurate robotic state estimation often requires precise knowledge of inter-sensor offsets. In this letter we present a solution for inter-sensor calibration of systems that employ a combination of GNSS and visual-inertial sensors. RTK-GNSS and fiducial measurements are utilized to produce highly precise estimates. We present an offline batch estimation approach that utilizes a continuous-time spline on the Lie group $SE(3)$. This approach simultaneously estimates the vehicle trajectory and sensor spatio-temporal offsets by choosing optimal spline control points and calibration parameters to solve a maximum likelihood estimation problem. Our calibration method is validated in both simulation and hardware, with comparison to an online extended Kalman filter. Hardware experiments are conducted in both a motion capture environment and outdoors. Results show that our method outperforms online methods and approaches millimeter-level accuracy.

Index Terms—Calibration and identification, localization, sensor fusion.

I. INTRODUCTION

A WIDE range of robotic applications benefit from the fusion of various types of sensor information. Two common choices for mobile robotics are global navigation satellite system (GNSS) and visual sensors. When available, GNSS can give a robot valuable information about both position and velocity in a global frame. However, traditional GNSS suffers from relatively low position accuracy (~ 1 m [1]) and all forms of GNSS require a clear view of the sky for incoming signals. Visual sensors yield information in the local camera frame for both position and orientation and their respective derivatives but typically require a consistent view of robust visual features. When fused together, these two measurement sources help overcome the fallbacks of using just one measurement source. Visual sensors can achieve higher accuracy than GNSS and provide odometric measurements when GNSS signals are obstructed. Similarly, systems can rely on GNSS measurements if visual features are insufficient

for robust estimation. Together these two measurement sources can lead to a variety of robust state estimation schemes (such as those in [2], [3]).

Prerequisite to the effective fusion of GNSS and visual measurements is the knowledge of spatio-temporal offsets between them, meaning the position of the GNSS receiver with respect to (w.r.t) the camera as well as the time difference between the measurement clocks. While these offsets can occasionally be measured by hand or through tuning processes (such as in [4]), the sensitive nature of many estimation methods often requires more precise values. Estimation of these parameters is non-trivial, especially when no prior information about the sensor origins is available.

Surprisingly there are only a few works in the literature that address the GNSS/camera calibration problem, and, to the knowledge of the authors, all of these works perform this calibration as part of an online pose estimation system (for example GNSS-aided visual-inertial odometry (VIO) [5], [6], [7]).¹ Online calibration has two primary limitations: 1) due to the real-time requirement of online systems, only short measurement windows can be considered during estimation, and 2) poor initial estimates of these offsets will adversely affect the online vehicle pose estimates, which could result in failure of safety-critical components such as path-planning and control. These drawbacks can be avoided by performing calibration offline and using the resulting values as fixed parameters during online estimation. Offline calibration has the advantage that it can be done as a full-batch process, where a large set of measurements is collected over an extended window of time and then all measurements are post-processed together to obtain the best possible estimate of the calibration parameters.

In this letter we present an offline full-batch optimization approach for calibration of GNSS/camera extrinsics. To our knowledge, this is the first GNSS/camera calibration scheme to be done offline. Our work is specifically motivated by the precision landing problem for unmanned aerial vehicles [8], but has a broad reach across robotic platforms that employ GNSS and visual sensors. To overcome accuracy pitfalls of traditional GNSS, we use a system equipped with real-time kinematic (RTK) GNSS receivers. RTK GNSS provides highly accurate relative measurements for calibration. However, once a system has been calibrated with the methods presented in this letter, the same receivers could be utilized for traditional GNSS in state estimation. Our system consists of a rover receiver that measures

Manuscript received 15 December 2022; accepted 3 April 2023. Date of publication 17 April 2023; date of current version 27 April 2023. This letter was recommended for publication by Associate Editor H. Kong and Editor L. Pallottino upon evaluation of the reviewers' comments. This work was supported in part by the Center for Autonomous Air Mobility and Sensing, National Science Foundation Industry/University Cooperative Research Center (I/UCRC) under NSF Grant 2139551, along with significant contributions from CAAMS industry members, and in part by Lawrence Livermore National Laboratory under subcontract B643355. (Corresponding author: Alex D. Jordan.)

Alex D. Jordan and Jacob C. Johnson are with Brigham Young University, Provo, UT 84602 USA (e-mail: a.jordan6955@gmail.com; jjohns99@byu.edu).

Timothy W. McLain and Randal W. Beard are with the Mechanical Engineering Department and Electrical and Computer Engineering Department respectively, Brigham Young University, Provo, UT 84602 USA (e-mail: mclain@byu.edu; beard@byu.edu).

Digital Object Identifier 10.1109/LRA.2023.3268014

¹By online systems we mean those that have real-time requirements so that estimates can be used by other autonomy modules (e.g. control). These systems generally use filtering or smoothing schemes to keep computation time low. Conversely, offline systems have no limits to computation time.

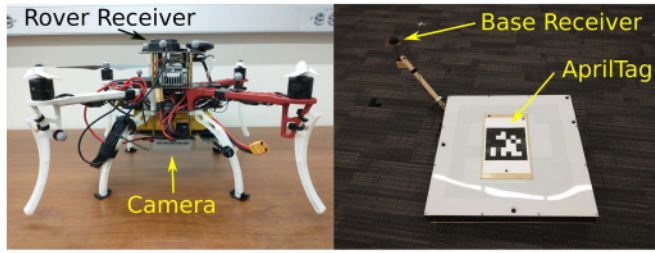


Fig. 1. Left: rover equipped with a camera-IMU and a GNSS receiver. Right: base equipped with an AprilTag fiducial marker and a GNSS receiver.

position relative to a base station receiver, as well as a camera that observes a fiducial marker attached to the base station (see Fig. 1). This setup provides drift-free, highly-accurate relative position in a global frame and camera pose measurements, allowing for high-precision spatio-temporal offset calibration. It also has the benefit that the base receiver position in the fiducial coordinate system, as well as the global rotation of the fiducial marker, can simultaneously be calibrated (these parameters are important in applications such as precision landing).

To solve the GNSS/camera calibration problem we adopt a nonlinear batch optimization approach. We parameterize the trajectory of the sensor system as a spline on the Lie group $SE(3)$ [9], allowing us to avoid rotational singularities. This continuous trajectory parameterization provides a means of connecting factors from GNSS with factors from the camera, which is essential for observing the calibration parameters. Furthermore, we add measurements from an inertial measurement unit (IMU) to find out if high frequency information will smooth the resulting trajectory estimate, which will in turn improve the calibration parameter estimates. To validate our method we compare results with an online calibration method via the error-state Kalman filter.

The contributions of the paper include the following:

- an offline full-batch estimation approach to GNSS/camera calibration, using splines on $SE(3)$ to parameterize the trajectory of the rover sensor system;
- a novel RTK-GNSS and fiducial marker setup that enables observation of the spatial and temporal offsets between the camera and the GNSS receiver on the rover, as well as the spatial offset between the fiducial marker and the GNSS receiver on the base;
- simulation, motion capture, and outdoor experiments comparing calibration accuracy using different sensor combinations, including RTK-GNSS, camera fiducial poses, GNSS Doppler velocity, and IMU.

II. RELATED WORK

A. Online Calibration

Calibration of GNSS/camera spatio-temporal offsets has received some attention in the GNSS-aided VIO literature. Researchers in [5] present a GNSS VIO filtering system that calibrates GNSS-IMU extrinsics online. Similarly, the work presented in [10] proposes a two stage filtering approach where the first 60-180 seconds of data are dedicated to calibration of camera-IMU and GNSS-IMU extrinsics, after which calibrated parameters are used in online state estimation. Both of these systems utilize the extended Kalman filter (EKF), which is

prone to inconsistencies due to linearization errors. Additionally, the time required for online filters to converge to reasonable calibration parameters may be unacceptable in deployed systems that perform time-critical tasks.

With the understanding that discrete batch-optimization approaches outperform the EKF in terms of accuracy per unit of computing time [11], the work in [6] and [7] present an optimization-based GNSS VIO system that includes GNSS to VIO extrinsics as an online calibration parameter. To fuse GNSS, IMU, and camera data in a batch optimization framework, these methods require using a complicated IMU preintegration scheme [12] to connect asynchronous and intermittent GNSS and camera factors. While each of these works provided significant contributions to the GNSS-VIO literature, we note that they do not explicitly focus on the problem of GNSS/camera calibration, thus calibration results are adequate at best and are not well-validated.

B. Offline Calibration

To our knowledge there are no prior works that calibrate GNSS/camera extrinsics in an offline system. However, there are existing offline methods in the camera/IMU calibration literature. The design of our GNSS/camera calibration method is heavily influenced by these systems. Traditionally, these systems have calibrated the transformation and time offset between the IMU and the camera using the EKF, as in [13]. These filtering-based methods use IMU measurements to propagate the filter state and camera measurements to perform the update step. The pose of the IMU and the calibration parameters are estimated simultaneously. With respect to offline calibration, the EKF suffers from two critical issues: 1) measurements are processed individually rather than considering the collection of measurements as a whole, and 2) it is especially prone to inconsistencies caused by linearization. More recently, the camera-IMU calibration literature has adopted the use of offline batch optimization techniques, where the calibration problem is solved by finding the optimal trajectory and calibration parameters given the entire collection of measurements [14], [15], [16]. In order to fuse high frequency information from the IMU with low frequency information from the camera, these methods use a continuous parameterization of the sensor trajectory (for example B-splines) that can be sampled at any measurement time, and optimize the control points that govern the shape of the trajectory. This method is advantageous over filtering-based methods because it processes all measurements simultaneously (e.g. measurements at the beginning of the trajectory affect the calibration estimate at the end of the trajectory and vice versa). Furthermore, it is less susceptible to linearization errors on account of its iterative optimization scheme. We likewise employ a batch optimization approach to estimate GNSS/camera spatio-temporal parameters, and parameterize the trajectory of the system using splines on $SE(3)$ [9].

III. SPLINES ON $SE(3)$

The cumulative form of a k^{th} order uniform B-spline in $\mathbb{R}^n$ with control points $\{\bar{\mathbf{p}}_l\}_{l=\{0,\dots,M\}}$, $\bar{\mathbf{p}}_l \in \mathbb{R}^n$ and uniform knot points $\{t_i\}_{i=\{0,\dots,M+k\}}$ (spaced by Δt) is given by

$$\mathbf{p}(t) = \bar{\mathbf{p}}_{i-1} + \sum_{j=1}^{k-1} b_j(u(t)) (\bar{\mathbf{p}}_{i+j-1} - \bar{\mathbf{p}}_{i+j-2}), \quad (1)$$

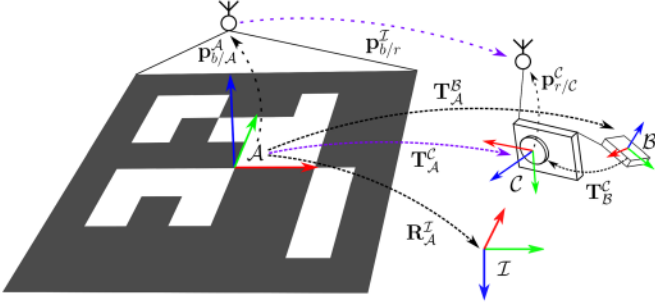


Fig. 2. Relevant variables and transformations in the RTK-GNSS/camera calibration problem. Measured quantities are shown in purple, coordinate transformations are shown with thick dotted lines, and position offsets are shown with sparse dotted lines.

where $u(t) = (t - t_i)/\Delta t \in [0, 1)$, and the index i is determined such that the sample time $t \in [t_i, t_{i+1})$. Here, $b_j(u)$ is the j^{th} index of

$$b(u) = C \begin{bmatrix} 1 & u & u^2 & \dots & u^{k-1} \end{bmatrix}^T, \quad (2)$$

where $C \in \mathbb{R}^{k \times k}$ is a constant matrix designed to satisfy bounding constraints [17].

Motivated by (1), we can define a spline on $SE(3)$ as products of geodesics, as was first done in [9]. Replacing summation with products of exponentials and subtraction with the logarithmic map, we get

$$\mathbf{T}(t) = \bar{\mathbf{T}}_{i-1} \prod_{j=1}^{k-1} \text{Exp}(b_j(u(t))\Omega_{i+j-1}), \quad (3)$$

where $\Omega_i = \text{Log}(\bar{\mathbf{T}}_{i-1}^{-1} \bar{\mathbf{T}}_i)$ and $\{\bar{\mathbf{T}}_l\}_{l=\{0, \dots, M\}}$, $\bar{\mathbf{T}}_l \in SE(3)$ are the control points of the spline. Here we have used the shorthand $\text{Exp} = \exp \circ \wedge$, where $\exp : \mathfrak{se}(3) \rightarrow SE(3)$ is the exponential map from the Lie algebra $\mathfrak{se}(3)$ to $SE(3)$ and the hat map $\wedge : \mathbb{R}^6 \rightarrow \mathfrak{se}(3)$ is the isomorphism that maps elements of Euclidean space into the Lie algebra. Also, $\text{Log} = \vee \circ \log$, where $\log : SE(3) \rightarrow \mathfrak{se}(3)$ is the inverse of the exponential map and $\vee : \mathfrak{se}(3) \rightarrow \mathbb{R}^6$ is the inverse of the hat map.

We will also need to compute the first and second time derivatives of the spline, $\dot{\varpi}(t) \in \mathbb{R}^6$ and $\ddot{\varpi} \in \mathbb{R}^6$ respectively. Due to space constraints we do not show how to compute these derivatives here, but mention that we use a recursive formulation similar to [18].

IV. PROBLEM OVERVIEW

The methods in this paper are motivated by robotic localization problems that require knowledge of positional offsets between GNSS and vision sensors. To calibrate this offset we use a vehicle equipped with a downward-facing monocular camera, an IMU, and an RTK-GNSS receiver. In addition, a stationary platform is equipped with an RTK-GNSS base receiver and an AprilTag fiducial marker [19], [20] to extract precise relative pose measurements between the platform and the vehicle from acquired camera images. The relevant sensors and coordinate frames are illustrated in Fig. 2. The camera frame $\mathcal{C}$ is centered at the sensor origin and the AprilTag frame $\mathcal{A}$ is located at the center of the marker with the k -axis normal to the marker surface. Images from the camera are passed to the AprilTag

algorithm [19] and, if the target marker is detected, the transformation from the AprilTag frame to the camera frame $\mathbf{T}_A^C$ is estimated. RTK-GNSS receivers do not have associated sensor frames, but rather represent points in space. Therefore, the output of the RTK system is a simple translation from the base receiver to the rover receiver $p_{b/r}^I$ expressed in a north-east-down inertial frame $\mathcal{I}$. The IMU frame $\mathcal{B}$ is centered at the sensor origin and is used as the body frame for the vehicle.

This problem introduces the desired unknown sensor offset of the rover receiver w.r.t. the camera expressed in the camera frame $p_{r/c}^C$. In calibrating this parameter it is also necessary to estimate the position of the base receiver w.r.t. the AprilTag expressed in the AprilTag frame $p_{b/A}^A$. This offset, while not the primary motivation behind this paper, is also useful for certain applications such as precision landing. In a calibration scenario, the rotation from the AprilTag frame to inertial frame $\mathbf{R}_A^I$ is assumed to be static and should also be estimated to convert GNSS measurements into the relevant coordinate frames. The transformation from the IMU frame to the camera frame $\mathbf{T}_B^C$ is assumed to be pre-calibrated with an off-the-shelf calibration package (such as Kalibr [16]). Lastly, the difference in time delay between the GNSS and camera measurements δt should be calibrated. Estimation of this offset is necessary because of the unknown delay between the time that a measurement is received (e.g. light reaching a camera sensor or signal reaching a GNSS receiver) and when it is timestamped. Estimation of δt allows the alignment of timestamps between GNSS and camera measurements.

V. CALIBRATION USING SPLINES ON $SE(3)$

To observe the GNSS base and rover receiver positions $p_{b/A}^A$ and $p_{r/c}^C$, we must collect relative position measurements from the RTK-GNSS system and AprilTag pose measurements from the camera simultaneously as the rover moves relative to the base. RTK-GNSS measurements have high position accuracy (~ 2 cm error), but do not give any information about the relative orientation of the rover w.r.t. the base. In addition, these measurements come at low frequencies (~ 5 Hz). AprilTag pose measurements allow the full relative pose of the rover to be observed, but have low-to-moderate accuracy and are obtained at relatively low frequencies (~ 30 Hz). Using these two measurement sources alone would allow us to fully calibrate the camera-GNSS system; however, using only low-frequency information could result in poor tracking performance and may require longer measurement collection periods in order to obtain an accurate calibration. Additionally, this sensor suite does not allow the direct observation of the rover velocity, which could result in poor pose interpolation between the low-frequency measurements. To combat these issues, we use two additional measurement sources: an IMU and relative velocity measurements from GNSS. IMU measurements come at high frequencies (~ 400 Hz) and allow for pose interpolation between camera and RTK-GNSS measurements. GNSS velocity measurements come at low frequencies but should provide smooth pose estimation results through direct observations of the rover velocity. We will investigate whether the use of these additional sensors improves the resulting calibration, and whether the extra computational cost required to include them is worthwhile.

To better facilitate the inclusion of high-frequency IMU data, we propose using continuous-time splines on $SE(3)$ to

represent the dynamic pose of the rover. The continuous nature of the spline allows us to sample the estimated pose and velocity of the rover at the exact measurement times rather than having to include a new parameter in the optimization for every measurement timestamp. Rather than optimizing the poses at the measurement times, we optimize a set of uniformly-spaced control points $\{\bar{\mathbf{T}}_j \in \text{SE}(3)\}_{j=0,\dots,M}$ that govern the shape of the spline. We let the spline represent the transformation from the AprilTag frame to the IMU frame $\mathbf{T}_A^B(t) : \mathbb{R} \rightarrow \text{SE}(3)$ for all times t between the first and last received measurement times, with first derivative $\dot{\omega}_{A/B}^B(t) = \begin{bmatrix} \mathbf{v}_{A/B}^B(t)^\top & \boldsymbol{\omega}_{A/B}^B(t)^\top \end{bmatrix}^\top : \mathbb{R} \rightarrow \mathbb{R}^6$. Here $\mathbf{v}_{A/B}^B$ and $\boldsymbol{\omega}_{A/B}^B$ are respectively the linear and angular velocity of the AprilTag frame w.r.t. the IMU frame expressed in the IMU frame. The second derivative of the spline is represented as $\ddot{\omega}_{A/B}^B(t) = \begin{bmatrix} \mathbf{a}_{A/B}^B(t)^\top & \boldsymbol{\alpha}_{A/B}^B(t)^\top \end{bmatrix}^\top : \mathbb{R} \rightarrow \mathbb{R}^6$, where $\mathbf{a}_{A/B}^B$ and $\boldsymbol{\alpha}_{A/B}^B$ are respectively the linear and angular acceleration of the AprilTag frame w.r.t. the IMU frame expressed in the IMU frame.

A. Measurement Models

1) *AprilTag Pose*: The AprilTag algorithm detects AprilTag fiducial markers from camera images and, from the projected positions of the tag corners on the sensor, computes the relative pose of the tag in the camera frame $\mathbf{T}_A^C$ using the Perspective-n-Point algorithm [21]. We model these measurements as

$$\mathbf{z}_{\text{cam}} = \tilde{\mathbf{T}}_A^C = \text{Exp}(\boldsymbol{\eta}_{\text{cam}}) h_{\text{cam}}(\mathbf{T}_A^B(t_{\text{cam}})), \quad (4a)$$

where

$$h_{\text{cam}}(\mathbf{T}) \triangleq \mathbf{T}_B^C \mathbf{T}, \quad (4b)$$

t_{cam} is the camera measurement acquisition time, and $\boldsymbol{\eta}_{\text{cam}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\text{cam}})$ is zero-mean Gaussian noise with covariance $\boldsymbol{\Sigma}_{\text{cam}} \in \mathbb{R}^{6 \times 6}$. This noise is applied in the tangent space of $\text{SE}(3)$ at the camera frame and is mapped onto $\text{SE}(3)$ via the exponential map².

2) *RTK Relative Position*: RTK-GNSS relies on carrier phase range measurements and differencing techniques to estimate the distance between two GNSS receivers. A typical RTK-GNSS system consists of one receiver designated the base and another designated the rover. Measurements represent the position of the rover w.r.t. the base expressed in the inertial frame $\mathbf{p}_{b/r}^I$, which we model as

$$\mathbf{z}_{\text{rtk}} = h_{\text{rtk}}(\mathbf{T}_A^B(t_{\text{gnss}} + \delta t), \mathbf{R}_A^I, \mathbf{p}_{b/A}^A, \mathbf{p}_{r/C}^C) + \boldsymbol{\eta}_{\text{rtk}}, \quad (5a)$$

where

$$h_{\text{rtk}}(\mathbf{T}, \mathbf{R}, \mathbf{p}_1, \mathbf{p}_2) \triangleq \mathbf{R}(\mathbf{T}^{-1} \mathbf{T}_C^B \cdot \mathbf{p}_2) - \mathbf{R} \mathbf{p}_1. \quad (5b)$$

Here $\boldsymbol{\eta}_{\text{rtk}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\text{rtk}})$ is zero-mean Gaussian noise with covariance $\boldsymbol{\Sigma}_{\text{rtk}}$, t_{gnss} is the RTK measurement acquisition time, and the operation $\mathbf{T} \cdot \mathbf{p}$ denotes the group action of $\text{SE}(3)$ on $\mathbb{R}^3$.

²We derive $\boldsymbol{\Sigma}_{\text{cam}}$ from the pixel projection covariance $\boldsymbol{\Sigma}_{\text{pix}}$, the fiducial side length s , and the camera matrix $\mathbf{K}$ to account for variations in noise given the viewing angle.

3) *GNSS Velocity*: GNSS receivers most commonly estimate velocity with Doppler measurements of receiver-satellite motion. Knowledge of the satellite velocity is then used to provide receiver velocity in an earth-centered, earth-fixed frame that we convert to a north-east-down frame at the starting longitude, latitude, and altitude. The velocity is modeled as

$$\mathbf{z}_{\text{vel}} = h_{\text{vel}}(\mathbf{T}_A^B, \boldsymbol{\omega}_{A/B}^B, \mathbf{R}_A^I, \mathbf{p}_{r/C}^C) + \boldsymbol{\eta}_{\text{vel}}, \quad (6a)$$

where

$$h_{\text{vel}}\left(\begin{bmatrix} \hat{\mathbf{R}} & \hat{\mathbf{p}} \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix}, \mathbf{R}, \mathbf{p}\right) \triangleq \mathbf{R} \hat{\mathbf{R}}^{-1} (\mathbf{v} + \boldsymbol{\omega}^\wedge [\mathbf{T}_C^B \cdot \mathbf{p}]), \quad (6b)$$

$\boldsymbol{\eta}_{\text{vel}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\text{vel}})$ is zero-mean Gaussian noise with covariance $\boldsymbol{\Sigma}_{\text{vel}}$, $\mathbf{T}_A^B$ and $\boldsymbol{\omega}_{A/B}^B$ are sampled from the spline at time $t_{\text{gnss}} + \delta t$, and t_{gnss} is the velocity measurement acquisition time assumed to be the same acquisition time of RTK measurements in Section V-A2.

4) *IMU*: The IMU consists of a gyroscope that measures the angular velocity of the IMU frame w.r.t. the inertial frame $\boldsymbol{\omega}_{B/I}^B$, as well as an accelerometer that measures the total acceleration of the IMU frame w.r.t. the inertial frame $\mathbf{a}_{B/I}^B$ (minus gravity). These sensors are plagued by walking biases $\mathbf{b}_{\text{gyro}}(t) : \mathbb{R} \rightarrow \mathbb{R}^3$ and $\mathbf{b}_{\text{acc}}(t) : \mathbb{R} \rightarrow \mathbb{R}^3$ respectively. We assume that these biases walk according to a Brownian motion model

$$\dot{\mathbf{b}}_{\text{gyro}} = \boldsymbol{\eta}_{\text{bg}}, \quad \dot{\mathbf{b}}_{\text{acc}} = \boldsymbol{\eta}_{\text{ba}}, \quad (7)$$

where $\boldsymbol{\eta}_{\text{bg}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\text{bg}})$ and $\boldsymbol{\eta}_{\text{ba}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\text{ba}})$ are zero-mean Gaussian noise with covariance $\boldsymbol{\Sigma}_{\text{bg}}, \boldsymbol{\Sigma}_{\text{ba}} \in \mathbb{R}^{3 \times 3}$ respectively.

The measurement model for the gyroscope is given by

$$\mathbf{z}_{\text{gyro}} = h_{\text{gyro}}(\boldsymbol{\omega}_{B/I}^B(t_{\text{imu}}), \mathbf{b}_{\text{gyro}}(t_{\text{imu}})) + \boldsymbol{\eta}_{\text{gyro}}, \quad (8a)$$

where

$$h_{\text{gyro}}(\boldsymbol{\omega}, \mathbf{b}) \triangleq \boldsymbol{\omega} + \mathbf{b}, \quad (8b)$$

t_{imu} is the IMU measurement time and $\boldsymbol{\eta}_{\text{gyro}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\text{gyro}})$ is zero-mean Gaussian noise with covariance $\boldsymbol{\Sigma}_{\text{gyro}} \in \mathbb{R}^{3 \times 3}$. If we assume that the base station is stationary w.r.t. the inertial frame, then $\boldsymbol{\omega}_{B/I}^B = -\boldsymbol{\omega}_{A/B}^B$, which can be recovered from the first derivative of the spline estimate.

The measurement model of the accelerometer is given by

$$\mathbf{z}_{\text{acc}} = h_{\text{acc}}(\mathbf{T}_A^B(t_{\text{imu}}), \mathbf{a}_{B/I}^B(t_{\text{imu}}), \mathbf{b}_{\text{acc}}(t_{\text{imu}}), \mathbf{R}_A^I) + \boldsymbol{\eta}_{\text{acc}}, \quad (9a)$$

where

$$h_{\text{acc}}\left(\begin{bmatrix} \hat{\mathbf{R}} & \hat{\mathbf{p}} \\ 0 & 1 \end{bmatrix}, \mathbf{a}, \mathbf{b}, \mathbf{R}\right) \triangleq \mathbf{a} - g \hat{\mathbf{R}} \mathbf{R}^{-1} \mathbf{e}_3 + \mathbf{b}, \quad (9b)$$

g is the gravitational acceleration constant, $\mathbf{e}_3 = [0 \ 0 \ 1]^\top$, and $\boldsymbol{\eta}_{\text{acc}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\text{acc}})$ is zero-mean Gaussian noise with covariance $\boldsymbol{\Sigma}_{\text{acc}} \in \mathbb{R}^{3 \times 3}$. Again, note that if the base station is stationary, $\mathbf{a}_{B/I}^B = -\mathbf{a}_{A/B}^B$, which can be recovered from the second derivative of the spline estimate. The model in (9) assumes that the k -axis of the inertial frame is aligned with the gravity vector. We feel that this is a fair assumption given that $\hat{\mathbf{I}}$ represents a north-east-down coordinate system w.r.t. the current position of the base station on the earth. Additionally,

the dependence of this model on global rotation of the AprilTag implies that including IMU measurements in the calibration should significantly improve the observability of two of the degrees of freedom in $\mathbb{R}_A^3$.

Because the IMU biases walk in time, it may not be sufficient to estimate them as constants, especially over large data collection windows. We therefore estimate them as uniform B-splines on $\mathbb{R}^3$ with control points $\{\bar{\mathbf{b}}_{g_j} \in \mathbb{R}^3\}_{j=0,\dots,M_{\text{gyro}}}$, $\{\bar{\mathbf{b}}_{a_j} \in \mathbb{R}^3\}_{j=0,\dots,M_{\text{acc}}}$, respectively.

B. Optimization Problem

Our goal is to determine the optimal values for the spline control points $\{\bar{\mathbf{T}}_j\}_{j=0,\dots,M}$, calibration parameters $\{\mathbf{p}_{r/c}^c, \mathbf{p}_{b/A}^A, \mathbf{R}_A^T, \delta t\}$, and IMU bias spline control points $\{\bar{\mathbf{b}}_{g_j}\}_{j=0,\dots,M_{\text{gyro}}}$, $\{\bar{\mathbf{b}}_{a_j}\}_{j=0,\dots,M_{\text{acc}}}$, given all the measurements we have received throughout the data collection period. This results in the maximum likelihood problem

$$\begin{aligned} \arg \min_{\Gamma} \left\{ \sum_{i_{\text{cam}}} \|\text{Log}(h_{\text{cam}}(\Gamma) \mathbf{z}_{i_{\text{cam}}}^{-1})\|_{\Sigma_{\text{cam}}^{-1}}^2 \right. \\ + \sum_{i_{\text{rtk}}} \|\mathbf{z}_{i_{\text{rtk}}} - h_{\text{rtk}}(\Gamma)\|_{\Sigma_{\text{rtk}}^{-1}}^2 + \sum_{i_{\text{vel}}} \|\mathbf{z}_{i_{\text{vel}}} - h_{\text{vel}}(\Gamma)\|_{\Sigma_{\text{vel}}^{-1}}^2 \\ \left. + \sum_{i_{\text{gyro}}} \|\mathbf{z}_{i_{\text{gyro}}} - h_{\text{gyro}}(\Gamma)\|_{\Sigma_{\text{gyro}}^{-1}}^2 + \sum_{i_{\text{acc}}} \|\mathbf{z}_{i_{\text{acc}}} - h_{\text{acc}}(\Gamma)\|_{\Sigma_{\text{acc}}^{-1}}^2 \right\} \quad (10) \end{aligned}$$

where Γ is the set containing all of the parameters to be optimized and $\|\cdot\|_{\mathbf{W}}$ indicates the 2-norm weighted by the matrix $\mathbf{W}$. This is a nonlinear least squares problem that can be solved using Gauss-Newton or Levenberg-Marquardt optimization. This problem requires the Jacobians of each of the cost terms w.r.t. the optimization parameters. We do not provide these Jacobians here due to space constraints.

Note 1 (Observability): We do not provide a formal observability analysis here but can deduce when it is possible to observe all of the calibration parameters using empirical evidence and intuition. The full pose trajectory $\mathbf{T}_A^B$ is observable from camera measurements defined by (4) alone. This allows the IMU biases to be observed, because angular velocity and acceleration can also be observed from the camera. The global rotation $\mathbf{R}_A^T$ requires that the rover translate along at least two axes relative to the base. The rover and base receiver positions $\mathbf{p}_{r/c}^c$ and $\mathbf{p}_{b/A}^A$ cannot be calibrated with translation alone, but require that the camera rotates w.r.t. the AprilTag about at least two axes. The time offset δt is observable given relative motion between the base and the rover while the AprilTag is in view of the camera.

VI. SIMULATION RESULTS

To evaluate the calibration procedure proposed in Section V, we simulated a mobile robot that receives artificial fiducial relative pose, RTK-GNSS, GNSS velocity, and IMU measurements using the models presented in Section V-A. To simulate these measurements, we used the following truth parameters:

$$\mathbf{p}_{b/A}^A = \begin{bmatrix} 1.0 \\ -1.0 \\ 1.5 \end{bmatrix} \text{ m}, \quad \mathbf{p}_{r/c}^c = \begin{bmatrix} 0.2 \\ 0.2 \\ -0.2 \end{bmatrix} \text{ m},$$

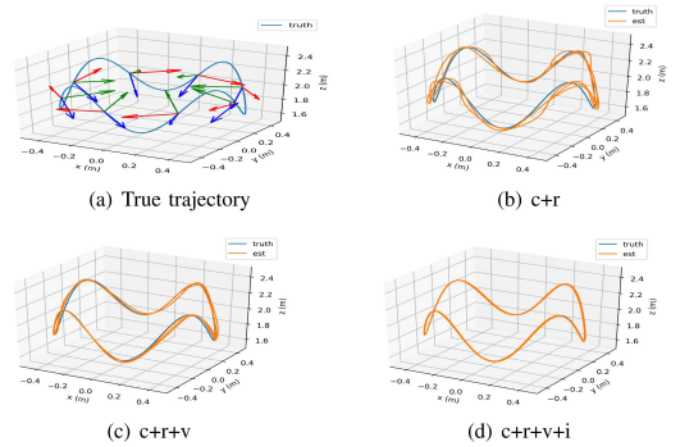


Fig. 3. Simulated trajectory estimation results for $c + r$, $c + r + v$, and $c + r + v + i$.

$$\begin{aligned} \mathbf{T}_B^c &= \begin{bmatrix} 1 & 0 & 0 & 0.2 \\ 0 & 1 & 0 & 0.1 \\ 0 & 0 & 1 & -0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{R}_A^T = \mathbf{R}\left(\frac{\pi}{12}, -\frac{\pi}{12}, \frac{2\pi}{3}\right), \\ \delta t &= -0.02 \text{ s}, \quad g = 9.81 \text{ m/s}^2, \quad s = 0.2 \text{ m}, \\ \Sigma_{\text{pix}} &= 0.25^2 \mathbf{I} \text{ px}^2, \quad \mathbf{K} = \begin{bmatrix} 320 & 0 & 320 \\ 0 & 320 & 240 \end{bmatrix}, \\ \Sigma_{\text{rtk}} &= \text{diag}(0.0004, 0.0004, 0.0016) \text{ m}^2, \\ \Sigma_{\text{vel}} &= \text{diag}(0.0004, 0.0004, 0.0016) \text{ (m/s)}^2, \\ \Sigma_{\text{gyro}} &= 0.002 \mathbf{I} \text{ (rad/s)}^2, \quad \Sigma_{\text{acc}} = 0.002 \mathbf{I} \text{ (m/s}^2\text{)}^2, \\ \Sigma_{\text{bg}} &= 0.005 \mathbf{I} \text{ (rad/s}^2\text{)}^2, \quad \Sigma_{\text{ba}} = 0.005 \mathbf{I} \text{ (m/s}^3\text{)}^2, \quad (11) \end{aligned}$$

where $\mathbf{R}(\phi, \theta, \psi)$ is a function that maps Euler angles ϕ, θ, ψ in rad to $\text{SO}(3)$. Camera, RTK-GNSS, GNSS velocity, and IMU measurements were obtained at frequencies of 10 Hz, 5 Hz, 5 Hz, and 100 Hz respectively. We simulated the motion of the IMU relative to the fiducial marker using sinusoids with a period of 4 s and amplitude of 0.5 m in the x and y directions and 1 s and 0.25 m in the z direction, at an altitude of 2 m. The pitch and roll sinusoids had a period of 2 s and amplitude of $\frac{\pi}{16}$ rad, and the yaw was varied linearly such that the fiducial marker would always stay in view of the camera. The resulting trajectory is shown in Fig. 3(a). This trajectory was carefully chosen to allow all of the calibration parameters to be observable.

We do not directly compare our offline calibration method to an existing GNSS/camera calibration method from the literature. To our knowledge, there are no other offline calibration methods in the literature to compare to. It is, however, worthwhile to compare our method to an online method, as these are prevalent in the literature. The Kalman filter is commonly used for online state estimation and has been used successfully for similar calibration scenarios (such as in [5], [10]). Therefore, we developed an error-state Kalman filter (ESKF) [22] on $\text{SE}(3)$ that uses the same measurement models in Section V-A³. This ESKF includes the

³Rather than use the IMU measurements in an update step, the ESKF uses them to drive the process model, similar to what is done in [23].

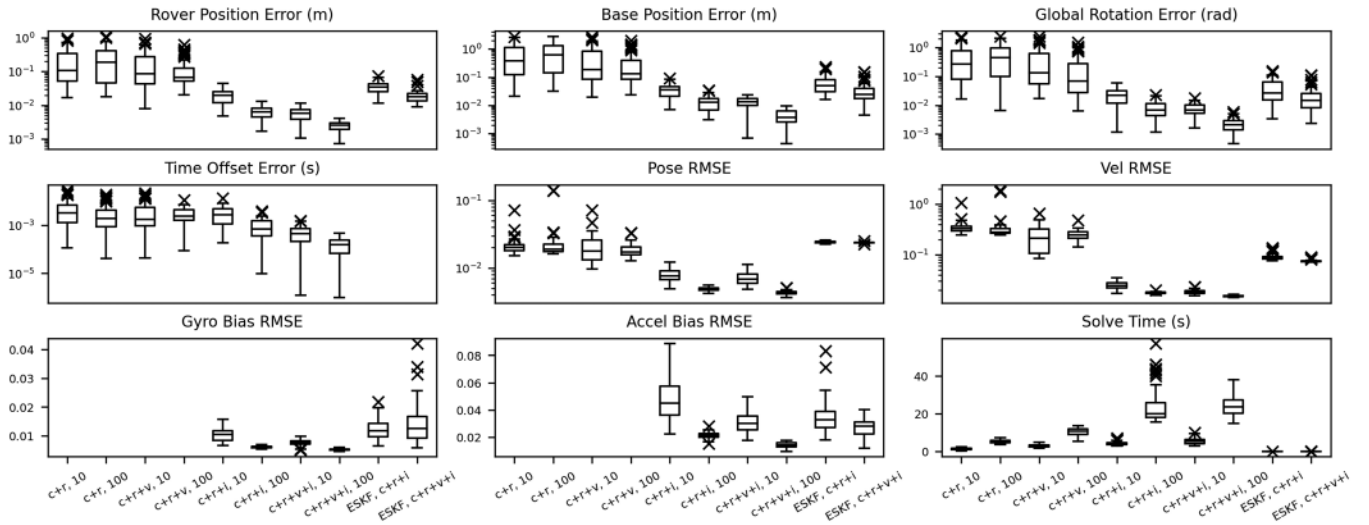


Fig. 4. Monte Carlo simulation results with 50 random trials for each sensor combination and trajectory length. Calibration errors, trajectory RMSEs, and solve times are shown. Note the logarithmic scale on the y -axis of most of the plots.

calibration parameters $p_{r/c}^c$, $p_{b/A}^A$, R_I^A as states and calibrates them while simultaneously estimating the trajectory⁴.

We tested our offline method with four different sensor combinations to calibrate the system and estimate the simulated trajectory: 1) camera and RTK-GNSS ($c + r$), 2) camera and RTK-GNSS with velocity measurements ($c + r + v$), 3) camera and RTK-GNSS with IMU measurements ($c + r + i$), and 4) camera and RTK-GNSS with velocity and IMU measurements ($c + r + v + i$). In all cases we used a spline on SE(3) with order $k = 6$ and knot spacing $\Delta t = 0.2$ s to parameterize the pose trajectory. Splines on $\mathbb{R}^3$ with order $k_b = 4$ and knot spacing $\Delta t_b = 5.0$ s were used to parameterize the IMU bias estimates. All control points were initialized to identity and 0 respectively, the receiver position offset estimates were initialized to 0, the time offset estimate was initialized to 0, and the global rotation estimate was initialized to identity. Levenberg-Marquardt optimization was implemented using Google Ceres [24].

Fig. 3 shows the trajectory estimation results for a 10 s trajectory. As can be seen from the plots, $c + r$ had the worst performance, followed by $c + r + v$. This is because these sensors all have low frequencies, so the trajectory estimate tended to drift during periods where no measurements were obtained. Adding velocity measurements appears to have reduced the drift in position and the inclusion of high frequency IMU measurements improves the trajectory estimate even further. The sensor combination $c + r + v + i$ had the best performance, likely because it had the most data available.

To evaluate the calibration performance of each sensor combination and the ESKF, we performed a Monte Carlo (MC) simulation with 50 random trials. In each trial, the noise generated by each sensor was varied with the covariances given in (11).⁵ In each case we used a single core of an Intel Core i7-10750H CPU. When including GNSS velocity and/or IMU measurements, we warm-started the optimization using the solution found with $c + r$. This is because when using velocity and/or IMU measurements the optimizer would often have trouble reaching a valid solution without a good initialization. Additionally, we tested the ESKF in the MC simulation for comparison. We noticed that the ESKF

struggled to converge to a reasonable solution when initialized with poor calibration parameters. Therefore, we initialized these parameters to within a small region of the true values (this is analogous to warm-starting with hand-tuned values). The final calibrated values for the online ESKF estimation method were taken as the average across the final 5 s of the trajectory. In addition, the first half of the trajectory was ignored when computing the trajectory error to allow the filter time to converge to a reasonable estimate. We tested the ESKF with velocity measurements ($c + r + v + i$) and without velocity measurements ($c + r + i$).

Fig. 4 shows box and whisker plots for the resulting rover receiver position, base receiver position, global rotation, and time offset error for each sensor combination, for trajectories of length 10 s and 100 s each. Also shown are the resulting pose, velocity, and IMU bias root mean squared errors (RMSE) for the entire trajectory length (with samples taken every 1 ms), as well as the total time required to solve the optimization problem (including the time to perform the warm-start when using velocity and/or IMU). Note that most of these plots use a logarithmic scale on the y -axis. From the plots it can be seen that $c + r$ and $c + r + v$ would usually converge to decimeter-level accuracy calibrations. The 100 s trajectories did not offer much improvement in these cases. The inclusion of the IMU significantly improved the calibration, providing centimeter-level accuracy, with improved trajectory quality as well. The longer trajectories offered significant improvement when the IMU was in use. The best performance was achieved when all four sensors were used on 100 s trajectories. Including IMU measurements

⁴As will be shown in the following results, online estimation is particularly difficult when measurement models are highly dependent on calibration parameters. To simplify the problem and narrow our comparison we chose to not calibrate the time offset δt . Instead the offset was set to 0 s in simulation and the estimated value from our method was provided as a precalibrated time offset in the ESKF for hardware experiments.

⁵Realistic noise parameters were chosen to emulate noise of hardware applications based on sensor datasheets.

TABLE I
MOTION CAPTURE CALIBRATION RESULTS FOR C + R AND C + R + I FOR A 50 AND 100 s TRAJECTORY ($\|P_{b/A}^A\| = 0.96$ m)

Type	$P_{b/A}^A$ err. (m)	R_{A}^T err. (rad)	$P_{r/C}^C$ (m)	δt (s)	Pose RMSE	Time (s)
c+r, 50	0.0136	0.0089	[0.100, 0.048, -0.196]	-0.0067	0.0560	2.90
c+r+i, 50	0.0098	0.0067	[0.099, 0.047, -0.198]	-0.0200	0.0458	24.88
c+r, 100	0.0039	0.0049	[0.100, 0.046, -0.201]	-0.0160	0.0460	9.90
c+r+i, 100	0.0039	0.0047	[0.100, 0.046, -0.202]	-0.0206	0.0421	35.58
ESKF, 100, GI	0.0041	0.0149	[0.101, 0.046, -0.201]	—	0.0464	0.50
ESKF, 100, PI	0.0653	0.0299	[0.095, 0.030, -0.153]	—	0.0465	0.48

increased the required solve time, but given the improvement in calibration, this seems to be worthwhile.

Also included in Fig. 4 are results for the online ESKF calibration method, which was run on a 100 s trajectory. When including all measurements and running on the 100 s trajectory, these results show that our offline method achieves a calibration error that is an order of magnitude lower than the ESKF error, illustrating that offline calibration is far superior to online calibration in this scenario.

VII. EXPERIMENTAL RESULTS

In addition to simulation we demonstrate our calibration method in real-world experiments. Our evaluation was conducted with the setup shown in Fig. 1, with an Intel RealSense D435i camera (calibrated using Kalibr [16]) and a UBLOX F9P receiver on the rover and an AprilTag fiducial marker and a UBLOX F9P on the base. To accurately show the reliability of our calibration method, we first conducted experiments using simulated RTK-GNSS in an Optitrack motion capture environment, followed by outdoor experiments using real GNSS data. Using motion capture data allows us to determine precise values for $P_{b/A}^A$ and R_{A}^T , as well as the IMU trajectory $T_I^B(t)$, which we use as ground truth⁶. In both calibration experiments the vehicle was maneuvered manually along a trajectory similar to that shown in Fig. 3. The ESKF described in Section VI was also used to attempt calibration on trajectories for both motion capture and outdoor.

A. Motion Capture

Motion capture data was used to create synthetic position measurements for RTK-GNSS. In the absence of GNSS signals indoors, motion capture dots were selected to represent the simulated GNSS sensor origins for both the rover and base receivers. Noise consistent with true RTK measurements was added and the synthetic measurements were published at the same rate as our GNSS units (5 Hz). Because motion capture does not provide velocity data, we did not simulate GNSS velocity measurements. The AprilTag was observed in the camera frame and detected poses were published at a rate of 30 Hz. To compare the estimated IMU trajectory with the motion capture trajectory, we estimated the transformation from the IMU frame to the rover motion capture coordinate frame after calibration.

We ran our calibration method on two trajectories of length 50 s and 100 s, comparing calibration and trajectory RMSE results for c + r and c + r + i. In all cases we used $k = 6$, $\Delta t = 0.075$ s, $k_b = 4$, and $\Delta t_b = 0.4$ s. The results are shown

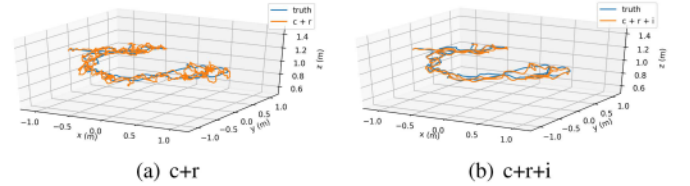


Fig. 5. Trajectory estimation results for c + r and c + r + i over a 50 s trajectory, plotted against motion capture data.

in Table I and Fig. 5. All four runs were able to calibrate $P_{b/A}^A$ to within 1.5 cm (its true value had a length of about 1 m) and R_{A}^T to within 10 mrad. The IMU offered significant improvement in calibration and pose RMSE for the 50 s trajectory, although the calibration accuracy with and without IMU was similar for the 100 s trajectory. While we do not know the true value of $P_{r/C}^C$, we can see that the calibrated values were consistent to within 6 mm. Also, we reason that because the relative receiver position is directly measured throughout the trajectory, an accurate calibration of $P_{b/A}^A$ should imply that $P_{r/C}^C$ is accurately calibrated as well. Finally, note the consistency of δt for the two c + r + i trajectories. These values are consistent with other trajectories that we tried as well, so we believe that -20 ms is close to the true value of δt . The ESKF (c + r + i) produced comparable results for the receiver offsets when given a good initialization (GI) with parameters that were close to the true values, but did not perform well when given a poor initialization (PI). Additionally, it was unable to calibrate R_{A}^T with as much accuracy as our method. In contrast, our method was able to find accurate values for all of the calibration parameters without a good initialization.

B. Outdoors

Outdoor calibration was conducted similar to that above but with real GNSS sensors and the inclusion of all available measurements (c + r + v + i). In this scenario, truth values are not available for evaluation. In motion capture it is simple to designate a single point as a simulated receiver location, but with real-world sensors it is difficult to pinpoint the exact location of sensor origins.⁷ This challenge is indeed the main motivation behind this paper. We conducted our calibration method and the ESKF on five independent 50 s trajectories using identical values for k , Δt , k_b , Δt_b as in Section VII-A. In lieu of ground truth

⁶Note that without a precise definition of the camera axes in the motion capture frame it is not possible to extract truth for the rover offset $P_{r/C}^C$.

⁷Note that simulated sensor origins in motion capture are not true sensor origins, hence the discrepancy between motion capture and outdoor offsets.

TABLE II
OUTDOOR CALIBRATION RESULTS

	$\mathbf{p}_{b/A}^A$ (m)	$\mathbf{p}_{r/C}^C$ (m)	δt (s)
mean	[0.766, -0.650, 0.308]	[0.053, 0.005, -0.184]	-0.0570
std. dev.	[0.006, 0.005, 0.004]	[0.003, 0.005, 0.006]	0.0033
ESKF mean	[0.859, -0.711, 0.389]	[0.045, 0.010, -0.162]	—
std. dev.	[0.112, 0.056, 0.079]	[0.036, 0.013, 0.037]	—
by hand	[0.750, -0.590, 0.340]	[0.055, 0.010, -0.185]	—

for evaluation, we used these separate trajectories to evaluate consistency across separate calibrations and qualitatively evaluate output values. Averages and standard deviations for these trajectories are provided in Table II. Although the calibrated offsets cannot be evaluated against the true values, results from our method were consistent across all five trajectories as shown by the millimeter-level standard deviations for each value. In addition, these values align closely with hand-tuned values from previous experiments which are also included in Table II. Therefore we consider these values to be reasonable for outdoor applications. The ESKF calibration produced similar values, but with a much higher standard deviation. We believe that this variation is because the ESKF needed a longer time for the parameters to converge than these 50 s trajectories provided, whereas our method was able to find consistent values in this short time frame.

VIII. CONCLUSION

Knowledge of inter-sensor offsets is imperative for robust state estimation in robotics. In this paper we presented a solution for calibration of GNSS and visual-inertial sensor extrinsics by employing RTK-GNSS and a fiducial marker. We presented a calibration approach that uses continuous-time splines on the Lie group SE(3). As opposed to the main body of literature that has focused on solving this problem online, we proposed an offline method. Our offline estimates benefit from a lack of real-time constraints, the use of a large batch of data, and provide precise calibration parameters that are available from the beginning for online applications. We have validated our methods in both simulation and hardware. Experiments in motion capture show millimeter-level accuracy and in the absence of ground-truth in outdoor environments we have shown reasonable and consistent results. As a method of comparison we also developed an ESKF that calibrates sensor offsets online. Results demonstrate that our proposed method is more consistent, accurate and successfully calibrates without precise initialization.

REFERENCES

- [1] C. Mekik and M. Arslanoglu, "Investigation on accuracies of real-time kinematic GPS for GIS applications," *Remote Sens.*, vol. 1, pp. 22–35, 2009.
- [2] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to MAV navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 3923–3929.
- [3] G. Cioffi and D. Scaramuzza, "Tightly-coupled fusion of global positional measurements in optimization-based visual-inertial odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5089–5095.
- [4] R. Jin, J. Liu, H. Zhang, and X. Niu, "Fast and accurate initialization for monocular vision/INS/GNSS integrated system on land vehicle," *IEEE Sensors J.*, vol. 21, no. 22, pp. 26074–26085, Nov. 2021.
- [5] W. Lee, K. Eickenhoff, P. Geneva, and G. Huang, "Intermittent GPS-aided VIO: Online initialization and calibration," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 5724–5731.
- [6] S. Han, F. Deng, T. Li, and H. Pei, "Tightly coupled optimization-based GPS-visual-inertial odometry with online calibration and initialization," 2022, *arXiv:2203.02677*.
- [7] B. Dong and K. Zhang, "A tightly coupled visual-inertial GNSS state estimator based on point-line feature," *Sensors*, vol. 22, no. 9, 2022, Art. no. 3391.
- [8] A. D. Jordan, M. Rydalch, T. McLain, and M. Williamson, "Precision maritime localization and landing with real-time kinematic GNSS," 2023. [Online]. Available: <https://arc.aiaa.org/doi/abs/10.2514/6.2023-0488>
- [9] S. Lovegrove, A. Patron-Perez, and G. Sibley, "Spline fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras," *BMVC*, vol. 2, no. 5, pp. 93.1–93.12, 2013.
- [10] Y. Zhou, S. Li, C. Xia, Z. Shen, X. Wang, and X. Li, "Online visual-inertial extrinsic calibration utilizing GNSS measurements for vehicle applications," *IEEE Sensors J.*, vol. 22, no. 5, pp. 4545–4557, Mar. 2022.
- [11] H. Strasdat, J. Montiel, and A. J. Davison, "Visual SLAM: Why filter?," *Image Vis. Comput.*, vol. 30, no. 2, pp. 65–77, 2012. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0262885612000248>
- [12] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual-inertial odometry," *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 1–21, Feb. 2017.
- [13] F. M. Mirzaei and S. I. Roumeliotis, "A Kalman filter-based algorithm for IMU-camera calibration: Observability analysis and performance evaluation," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1143–1156, Oct. 2008.
- [14] M. Fleps, E. Mair, O. Ruepp, M. Suppa, and D. Burschka, "Optimization based IMU camera calibration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2011, pp. 3297–3304.
- [15] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 1280–1286.
- [16] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, "Extending kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 4304–4311.
- [17] K. Qin, "General matrix representations for b-splines," *Vis. Comput.*, vol. 16, no. 3/4, pp. 177–186, 2000.
- [18] C. Sommer, V. Usenko, D. Schubert, N. Demmel, and D. Cremers, "Efficient derivative computation for cumulative b-splines on Lie groups," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11145–11153.
- [19] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 3400–3407.
- [20] M. Krogus, A. Haggemiller, and E. Olson, "Flexible layouts for fiducial tags," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 1898–1903.
- [21] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [22] J. Sola, "Quaternion kinematics for the error-state Kalman filter," 2017, *arXiv:1711.02508*.
- [23] M. Li and A. I. Mourikis, "3-D motion estimation and online temporal calibration for camera-IMU systems," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2013, pp. 5709–5716.
- [24] S. Agarwal, K. Mierle, and T. C. S. Team, "Ceres solver," 2022. [Online]. Available: <https://github.com/ceres-solver/ceres-solver>