

Visual-inertial Self-calibration on Informative Motion Segments

Thomas Schneider*, Mingyang Li[†], Michael Burri*, Juan Nieto*, Roland Siegwart* and Igor Gilitschenski*
*Autonomous Systems Lab, ETH Zurich, [†]Google Inc., Mountain View, CA

Abstract—Environmental conditions and external effects, such as shocks, have a significant impact on the calibration parameters of visual-inertial sensor systems. Thus long-term operation of these systems cannot fully rely on factory calibration. Since the observability of certain parameters is highly dependent on the motion of the device, using short data segments at device initialization may yield poor results. When such systems are additionally subject to energy constraints, it is also infeasible to use full-batch approaches on a big dataset and careful selection of the data is of high importance.

In this paper, we present a novel approach for resource efficient self-calibration of visual-inertial sensor systems. This is achieved by casting the calibration as a segment-based optimization problem that can be run on a small subset of informative segments. Consequently, the computational burden is limited as only a predefined number of segments is used. We also propose an efficient information-theoretic selection to identify such informative motion segments. In evaluations on a challenging dataset, we show our approach to significantly outperform state-of-the-art in terms of computational burden while maintaining a comparable accuracy.

I. INTRODUCTION

In this work, we address the problem of sensor self-calibration of a visual-inertial tracking system, i.e., a state estimation system that fuses measurements from an inertial measurement unit (IMU) and one/multiple cameras to compute pose (position and orientation) estimates of a moving platform. In recent years visual-inertial tracking has witnessed an ever increasing gain in popularity and is used in numerous mobile devices, virtual and augmented reality systems, and robotic platforms. This success story results in large-scale projects such as Google Tango or Microsoft’s HoloLens promising to make these complex systems available as part of consumer devices with a limited energy supply and may be operated by inexperienced users over a potential lifespan of several years. These developments pose novel technical challenges to ensure accurate calibration of extrinsics and intrinsics of the underlying sensor systems.

Outside a lab environment, varying environmental conditions (such as temperature) and a long lifespan result in changing calibration parameters that make permanent use of factory calibration infeasible even when assuming all parameters to be constant over a short or medium timespan. In the absence of experienced engineers with access to special calibration routines and calibration patterns, the systems need to be capable of calibrating automatically in a potentially unknown environment. Even though it was shown that calibration is also possible by using natural visual landmarks only [1], parameters such as axis misalignment of the IMU can only be observed under certain motion. One

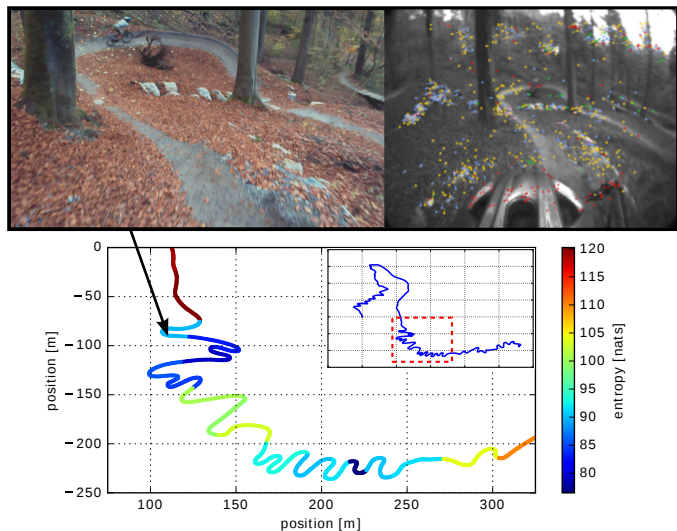


Fig. 1: Riding down Mount Uetliberg on a mountain-bike with a camera and IMU attached to the rider’s helmet: This dataset is a good illustration of the vastly varying amount of information available in different segments of the trajectory. The color indicates the information content of the segment w.r.t. the sensor calibration parameters (intrinsics and extrinsics of camera/IMU) where a lower value indicates more information. Consequently, the information measure is used to sparsify the sensor self-calibration problem by excluding less informative portions of the dataset.

possible solution is to run a full-batch calibration procedure over as much data as possible. However, this results in a huge computational load making this methodology infeasible for consumer devices with limited computational resources and a limited power supply.

This work makes use of the fact that visual-inertial estimation systems typically run for a sufficiently long time to perform a lot of different types of motion eventually. Therefore, our system is designed to automatically select informative motion segments, that are well suited for calibration. The information measure, used for this identification, is illustrated on an example trajectory in Fig. 1. The most informative segments are then stored in a database and used to refine the calibration from time to time. This not only helps in getting good calibration data, but also reduces the size of the calibration problem considerably. Furthermore, we show that the results of our calibration, using only a small number of segments, is comparable in accuracy to the results obtained with a full batch approach over all the data collected.

This paper makes the following contributions:

- We present an efficient information-theoretic procedure to identify the most informative segments of a trajectory.
- We propose a segment-based method for self-calibration of the intrinsic and extrinsic parameters of visual-inertial sensor systems.
- In thorough evaluations we show that the proposed methodology achieves comparable results to a full batch approach and state-of-the-art while at the same time requires a significant lower complexity and computational effort.

II. RELATED WORK

Over the last decade, visual-inertial SLAM has received great attention from the research community and tremendous progress has been achieved. For example, the work of [2] demonstrates a fixed-lag-smoother based VIO framework, that achieves accuracies in the sub-percent range over the travelled distance. However, on constrained platforms such as mobile phones, filtering based algorithms are preferred such as [3] and [4] that show similar accuracies at lower computational complexity.

To achieve such accuracies, precise calibration of the sensor models is required. Traditionally, camera models are calibrated using a calibration target such as in the work of [5]. It has also been shown that camera models can be obtained using natural features only [6]. The increasing usage of low-cost MEMS IMUs further requires calibration of the inertial sensors, referred to as IMU intrinsics. The work of [7] presents an inertial model, which we will adopt in this work, that considers scale inaccuracies and misalignments of individual sensors axes. In [8] a batch estimator is presented that calibrates the latter model relying on a calibration pattern. The model of [9] additionally considers the location of individual accelerometer axes where the parameters are estimated in a continuous-time formulation using a parametric estimation framework.

The recent roll-out of advanced SLAM systems to a wide audience creates a need for simple calibration algorithms accessible to users without access to special equipment such as calibration targets. The work of [1] mitigates these shortcomings by including the calibration parameters directly into an EKF-based VIO estimator and performing visual and inertial self-calibration solely based on natural features. Visual inertial systems, however, require special motion in order to render all calibration parameters observable [10]. Therefore, observability-aware calibration methods have been developed to aid non-expert users in collecting a complete dataset of minimal size and improve the estimation quality. In [11], a set of informative segments is selected using an information-gain measure to consequently perform a calibration over this set. Further, a truncated QR solver is used to constrain parameter updates to the observable sub-space. The generality of this method makes it applicable to a wide-range of estimation problems. Unfortunately, the evaluation of the utilized information metric is expensive and can prevent its use especially on resource constrained platforms. In our work, we follow a similar approach and identify informative

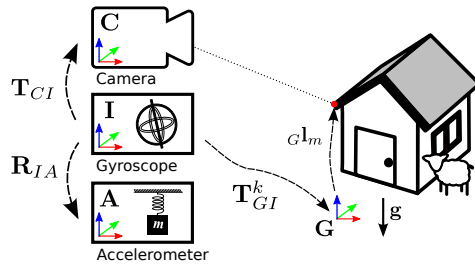


Fig. 2: Frame of reference definitions for the visual-inertial system. A camera, 3-dof accelerometer and 3-dof gyroscope are rigidly attached to an agent. The estimated pose of the agent at timestep k is expressed by the transformation \mathbf{T}_{GI}^k . A 6-dof transformation matrix \mathbf{T}_{CI} relates the gyroscope’s frame I to the camera’s frame C . The accelerometer frame A is only rotated w.r.t. gyroscope’s frame I by \mathbf{R}_{IA} , since ${}_I\mathbf{P}_{IA}$ in single-chip MEMS-IMUs is typically close to zero.

motion segments to build a sparser but complete calibration dataset. Similarly to the work of [12], we use the entropy to efficiently approximate the information content of segments but calibrate the full visual-inertial model instead of just a camera. Additionally, we extend the information measure and evaluate the informativeness of segments w.r.t. to subgroups of the high-dimensional parameter vector and thus mitigate the drawback of using a scalar measure. Another interesting approach approximates the information of a trajectory segment by the local observability Gramian, as described in [13], where it is used in an active calibration setting.

III. VISUAL-INERTIAL MODELS AND CALIBRATION

In this section, we will introduce the sensor models for the camera and IMU and formulate the batch estimation problem for self-calibration.¹

A. Notation and Frames of Reference

A transformation matrix $\mathbf{T}_{AB} \in \mathbb{SE}^3$ takes vector ${}_B\mathbf{p} \in \mathbb{R}^3$ from the frame of reference B to the frame of reference A and can be further partitioned into a rotation matrix $\mathbf{R}_{AB} \in \mathbb{SO}^3$ and a translation vector ${}_A\mathbf{p}_{AB} \in \mathbb{R}^3$ as follows:

$$\begin{bmatrix} {}_A\mathbf{p} \\ 1 \end{bmatrix} = \mathbf{T}_{AB} \cdot \begin{bmatrix} {}_B\mathbf{p} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{AB} & {}_A\mathbf{p}_{AB} \\ \mathbf{0} & 1 \end{bmatrix} \cdot \begin{bmatrix} {}_B\mathbf{p} \\ 1 \end{bmatrix} \quad (1)$$

Further, the unit quaternion \mathbf{q}_{AB} represents the rotation corresponding to \mathbf{R}_{AB} as defined in [14]. The operator $\mathbf{T}_{AB}(\cdot)$ is defined to transform a vector in \mathbb{R}^3 from B to the frame of reference A as ${}_A\mathbf{p} = \mathbf{T}_{AB}({}_B\mathbf{p})$ according to Eq. (1).

The Fig. 2 illustrates the relevant coordinate frames used within this work. The frame G denotes a gravity aligned (${}_G\mathbf{e}_z = -\mathbf{g}$) inertial frame and is used to express the estimated pose of the agent \mathbf{T}_{GI}^k and the position of the estimated landmarks ${}_G\mathbf{l}_m$. The frame I coincides with the sensing axes of the gyroscope and is chosen as the body frame of the agent. The camera frame C and accelerometer

¹It is important to note that the method described in this paper generalizes to arbitrary problems, however it is presented on the application of visual-inertial self-calibration.

frame A are rigidly attached to the body frame. The extrinsic calibration transformations for the camera \mathbf{T}_{CI} and the rotation matrix for the accelerometer \mathbf{R}_{IA} are to be estimated and are both defined relative to the frame of the gyroscope I that is used as the body frame.

B. Inertial Model

A triad of (ideally) orthogonal gyroscopes are used to sense the true angular velocities ${}_I\boldsymbol{\omega}_{GI}$ of the body frame I w.r.t. the world-fixed inertial frame G . The gyroscope measurements $\tilde{\boldsymbol{\omega}}$ are modeled similar to [7, 1] as:

$$\tilde{\boldsymbol{\omega}} = \mathbf{T}_g \cdot {}_I\boldsymbol{\omega}_{GI} + \mathbf{b}_g + \boldsymbol{\eta}_g \quad (2)$$

where the bias \mathbf{b}_g follows a random walk process as $\dot{\mathbf{b}}_g = \boldsymbol{\eta}_{bg}$ and $\boldsymbol{\eta}_g$ and $\boldsymbol{\eta}_{bg}$ are zero-mean, white Gaussian noise processes. The matrix \mathbf{T}_g accounts for scale errors and sensor axis misalignments present in cheaper sensors. It is assumed to be a constant over time and is structured as:

$$\mathbf{T}_g = \begin{bmatrix} s_g^x & m_g^x & m_g^y \\ 0 & s_g^y & m_g^z \\ 0 & 0 & s_g^z \end{bmatrix}, \mathbf{s}_g = \begin{bmatrix} s_g^x \\ s_g^y \\ s_g^z \end{bmatrix}, \mathbf{m}_g = \begin{bmatrix} m_g^x \\ m_g^y \\ m_g^z \end{bmatrix} \quad (3)$$

with \mathbf{s}_g and \mathbf{m}_g denoting the collection of all parameters from \mathbf{T}_g .

Similarly, the specific force measurements $\tilde{\mathbf{a}}$ of the accelerometer are modeled as:

$$\tilde{\mathbf{a}} = \mathbf{T}_a \cdot \mathbf{R}_{AI} \cdot \mathbf{R}_{IG}^k \cdot ({}_G\mathbf{a}_{GI} - {}_G\mathbf{g}) + \mathbf{b}_a + \boldsymbol{\eta}_a \quad (4)$$

where \mathbf{T}_a is a calibration matrix and \mathbf{b}_a defines a random walk process analog to the gyroscope model. The calibration states for the IMU models can be summarized as:

$$\boldsymbol{\theta}_i = [\mathbf{s}_g^T \quad \mathbf{m}_g^T \quad \mathbf{s}_a^T \quad \mathbf{m}_a^T \quad \mathbf{q}_{AI}^T]^T \quad (5)$$

It is important to note that the values of the scale parameters s_i can't be used directly to correct the scales of each individual axis, instead a linear combination of all factors applies. Further details can be found in [1].

C. Camera Model

Let ${}_G\mathbf{l}_m$ denote a 3-d landmark observed from keyframe k that is projected into a 2-d point $\mathbf{z}_{k,m}$ on the image plane of the camera as follows:

$$\mathbf{z}_{k,m}(\mathbf{T}_{IG}, \mathbf{l}_m, \boldsymbol{\theta}_c) = f_p(\boldsymbol{\theta}_c, \mathbf{T}_{CI}(\mathbf{T}_{IG}({}_G\mathbf{l}_m))) + \boldsymbol{\eta}_c \quad (6)$$

where $f_p(\cdot)$ denotes the perspective projection function and $\boldsymbol{\eta}_c \sim \mathcal{N}(\mathbf{0}, \sigma_c^2 \cdot \mathbf{I}_2)$ a white Gaussian noise process.

For the evaluations, we parametrize the projection function f_p using a pinhole camera model and field-of-view (FOV) distortion model of [6]. The calibration state relevant for the camera model then is:

$$\boldsymbol{\theta}_c = [\mathbf{q}_{CI}^T \quad {}_C\mathbf{p}_{CI}^T \quad \mathbf{f}^T \quad \mathbf{c}^T \quad w]^T$$

where \mathbf{q}_{CI} and ${}_C\mathbf{p}_{CI}$ are the extrinsic calibration of the camera w.r.t. IMU, $\mathbf{f} = [f_x \quad f_y]^T$ the focal lengths, $\mathbf{c} = [c_x \quad c_y]^T$ the principal point and w a distortion parameter.

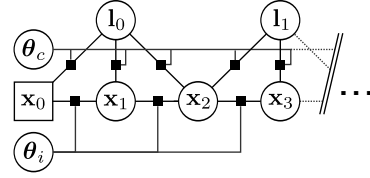


Fig. 3: Calibration problem in factor graph representation that contains visual-inertial keyframe states \mathbf{x}_k (pose, velocity and IMU biases), landmarks \mathbf{l}_m and calibration states for the camera $\boldsymbol{\theta}_c$ and IMU $\boldsymbol{\theta}_i$. The square around the initial node \mathbf{x}_0 denotes a gauge fix of the position ${}_G\mathbf{p}_{GI}$ and rotation around the gravity vector. Integrated IMU measurements constitute the inertial factor $g_k^{imu}(\mathbf{x}_k, \mathbf{x}_{k-1}, \boldsymbol{\theta}_i, \mathbf{u}_k)$ and the landmark projection factors $g_{k,m}^{cam}(\mathbf{x}_k, \mathbf{l}_m, \boldsymbol{\theta}_c, \mathbf{z}_{k,m})$ models the camera measurements.

D. Maximum-likelihood Estimator

The framework of maximum-likelihood estimation is used to jointly estimate all keyframe states \mathbf{x}_k (Eq. (7)), the scene as a set of observed point landmarks ${}_G\mathbf{l}_m$, the calibration parameters of the camera $\boldsymbol{\theta}_c$ and the IMU $\boldsymbol{\theta}_i$ with the keyframe state \mathbf{x}_k being defined as:

$$\mathbf{x}_k = [\mathbf{q}_{GI}^k{}^T \quad {}_G\mathbf{p}_{GI}^k{}^T \quad {}_G\mathbf{v}_I^k{}^T \quad \mathbf{b}_a^k{}^T \quad \mathbf{b}_g^k{}^T]^T \quad (7)$$

where \mathbf{q}_{GI}^k and ${}_G\mathbf{p}_{GI}^k$ denote the pose of the agent, ${}_G\mathbf{v}_I^k$ the velocity of the IMU expressed in frame G , and \mathbf{b}^k the biases for the gyroscope or accelerometer. For convenience of notation, the individual states are stacked into vectors as follows:

$$\hat{\mathbf{X}} = [\hat{\mathbf{x}}_0^T \quad \dots \quad \hat{\mathbf{x}}_K^T]^T, \quad \hat{\mathbf{L}} = [{}_G\hat{\mathbf{l}}_M^T \quad \dots \quad {}_G\hat{\mathbf{l}}_M^T]^T, \\ \hat{\boldsymbol{\theta}} = [\hat{\boldsymbol{\theta}}_c^T \quad \hat{\boldsymbol{\theta}}_i^T]^T$$

where K denotes the number of keyframes and M the number of landmarks. Additionally $\hat{\boldsymbol{\pi}}$ defines the collection of all estimated quantities as:

$$\hat{\boldsymbol{\pi}} = [\hat{\boldsymbol{\theta}}^T \quad \hat{\mathbf{X}}^T \quad \hat{\mathbf{L}}^T]^T$$

We want to infer $\boldsymbol{\pi}$ from measurements $\mathbf{z}_{k,m}$ made by a camera and measurements \mathbf{u}_k of an IMU. The stacked vector forms of the measurements are defined as follows:

$$\mathbf{Z} = \{\mathbf{z}_{k,m} | k \in [0, K], m \in [0, M(k)]\} \\ \mathbf{U} = \{\mathbf{u}_k | k \in [0, K-1]\}$$

Following the sensor models described in Section III-B-III-C, a probability model is defined as shown in Fig. 3. Probabilistic inertial constraints g_k^{imu} between consecutive keyframe states k and $k+1$ are formed as a function of the integrated IMU measurements and the corresponding measurement uncertainties [3]. The likelihood $p(\cdot)$ of this model can be expressed as:

$$p(\boldsymbol{\pi} | \mathbf{Z}, \mathbf{U}) \propto \prod_{k=1}^K p(\mathbf{x}_k | \mathbf{x}_{k-1}, \boldsymbol{\theta}_i, \mathbf{u}_k) \\ \cdot \prod_{k=0}^K \prod_{m=0}^{M(k)} p(\mathbf{z}_{k,m} | \mathbf{x}_k, \mathbf{l}_m, \boldsymbol{\theta}_c) \quad (8)$$

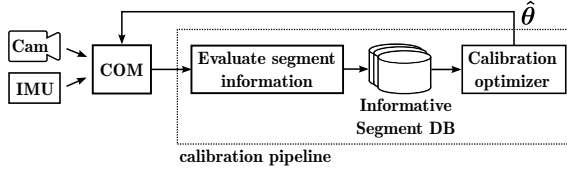


Fig. 4: System overview and context: informative motion segments are identified from the output of an existing ego-motion estimator (COM) and maintained in a database for future calibration.

where $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \boldsymbol{\theta}_i, \mathbf{u}_i)$ denotes the inertial constraints between two consecutive keyframe states as a function of integrated IMU measurements \mathbf{u}_k and $p(\mathbf{z}_{k,m}|\mathbf{x}_k, \mathbf{l}_m)$ the measurement model of the point landmark observation $\mathbf{z}_{k,m}$ of the m -th landmark observed from the k -th keyframe. More details on the derivation can be found in [8].

The maximum-likelihood (ML) estimate $\hat{\boldsymbol{\pi}}_{ML}$ is obtained by solving the optimization problem that maximizes the likelihood of Eq. (8):

$$\hat{\boldsymbol{\pi}}_{ML} = \underset{\boldsymbol{\pi}}{\operatorname{argmax}} p(\boldsymbol{\pi}|\mathbf{Z}, \mathbf{U}) \quad (9)$$

With the assumptions of Gaussian noise for all sensor models, as discussed in Section III-B and Section III-C, the optimization problem defined in Eq. (9) is equivalent to a non-linear least squares problem. This problem can be solved using numerical minimization approaches, where standard methods include Gauss-Newton, Levenberg-Marquardt, Dog-leg, etc. In our implementation, we use the Levenberg-Marquardt implementation of the Ceres framework [15].

IV. METHOD

The proposed self-calibration method aims at being run in parallel to an existing visual-inertial SLAM system that provides motion estimates as shown in Fig. 4. In our implementation we use a concurrent odometry and mapping (COM) framework consisting of [3], [16] and [17] but it is important to note that the proposed algorithms are not tied to a particular SLAM formulation. The SLAM system uses a calibration from previous runs or nominal values for the device at hand.² The stream of estimated keyframes $\hat{\mathbf{x}}_i$ and landmarks $\hat{\mathbf{l}}_i$ leaving the COM module is partitioned into motion segments \mathcal{S}^i of a predefined size N as follows:

$$\begin{aligned} \hat{\mathbf{X}}_{\mathcal{S}}^i &= \left[\hat{\mathbf{x}}_i^T, \dots, \hat{\mathbf{x}}_{i+(N-1)}^T \right]^T \\ \hat{\mathbf{L}}_{\mathcal{S}}^i &= \left[\hat{\mathbf{l}}_i^T, \dots, \hat{\mathbf{l}}_{i+(N-1)}^T \right]^T \end{aligned} \quad (10)$$

where $\hat{\mathbf{X}}_{\mathcal{S}}^i$ denotes the keyframes within the i -th segment and $\hat{\mathbf{L}}_{\mathcal{S}}^i$ the landmarks observed by the i -th segment. An efficient information-theoretic measure is used to evaluate each new candidate segment for their information content w.r.t. the calibration parameters and the most informative segments are maintained in a database. Once enough segments have been collected, an ML-based calibration is triggered to estimate

²If no priors are available, a complete self-calibration may be difficult and specialized initialization techniques should be used beforehand e.g. [5, 18].

the calibration parameters. An overview of the algorithm is shown in Alg. 1. The remainder of this section will discuss the algorithm in more detail.

Algorithm 1 Method shown for a single parameter group

Input: Initial calibration: $\hat{\boldsymbol{\theta}}_{init}$

Output: Updated calibration: $\hat{\boldsymbol{\theta}}$

Loop

// Initialize motion segments of size N from COM output.

$\mathcal{S}_i \leftarrow \{\}$

repeat

$data = \text{WaitForNewSensorData}()$

$\hat{\mathbf{x}}_j, \hat{\mathbf{l}}_j \leftarrow \text{RunCOM}(data, \hat{\boldsymbol{\theta}}_{init})$

$\mathcal{S}_i \leftarrow \mathcal{S}_i \cup (\hat{\mathbf{x}}_j, \hat{\mathbf{l}}_j)$

until $\dim(\mathcal{S}_i) == N$;

$H(\boldsymbol{\theta}) \leftarrow \text{EvaluateSegmentInformation}(\mathcal{S}_i)$ *// Section IV-A*

$\text{UpdateDatabase}(\mathcal{S}_i, H(\boldsymbol{\theta}))$ *// Section IV-B*

if $\text{EnoughSegmentsInDatabase}()$ **then**

$\mathbf{S}_{info} \leftarrow \text{GetAllSegmentsFromDatabase}()$

$\hat{\boldsymbol{\theta}} \leftarrow \text{RunOptimization}(\mathbf{S}_{info})$ *// Section IV-C*

return $\hat{\boldsymbol{\theta}}$

end

$i \leftarrow i + 1$

EndLoop

A. Evaluating Information Content of Segments

We use the differential entropy to quantify the information content of the i -th candidate segment \mathcal{S}_i w.r.t. the calibration parameters $\boldsymbol{\theta}$ by considering only the constraints within each segment. Using the entropy to evaluate the information of a candidate segments, as a score that is independent of all other segments, makes its evaluation very efficient at the cost that information coming from other segments is neglected. For example, loop-closure constraints cannot be considered in the score, however, loop-closures are considered during calibration.

To calculate the segment entropy, we first approximate the covariance matrix of all states in the segment by the inverse of the Fisher Information Matrix as:

$$\boldsymbol{\Sigma}_{\mathbf{X}\mathbf{L}\boldsymbol{\theta}} = \text{Cov} [p(\mathbf{X}_{\mathcal{S}}^i, \mathbf{L}_{\mathcal{S}}^i, \boldsymbol{\theta}|\mathbf{U}_i, \mathbf{Z}_i)] = (\mathbf{J}_i^T \mathbf{T}_i^{-1} \mathbf{J}_i)^{-1} \quad (11)$$

where \mathbf{J}_i denotes the Jacobian of all error-terms in the segment and \mathbf{T}_i the stacked error-term covariance where the column ordering is chosen that the calibration parameters $\boldsymbol{\theta}$ lie on the right side. To avoid a costly inversion of Eq. (11), which becomes intractable for larger problems, we make use of a rank-revealing QR decomposition to obtain $\mathbf{Q}_i \mathbf{R}_i = \mathbf{L}_i \mathbf{J}_i$ where $\mathbf{T}_i^{-1} = \mathbf{L}_i^T \mathbf{L}_i$ denotes the Cholesky decomposition of the error-term covariance matrix. Eq. (11) can then be rewritten as:

$$\boldsymbol{\Sigma}_{\mathbf{X}\mathbf{L}\boldsymbol{\theta}} = (\mathbf{R}_i^T \mathbf{R}_i)^{-1} = \begin{bmatrix} \boldsymbol{\Sigma}_{\mathbf{X}\mathbf{L}} & \boldsymbol{\Sigma}_{\mathbf{X}\mathbf{L},\boldsymbol{\theta}} \\ \boldsymbol{\Sigma}_{\mathbf{X}\mathbf{L},\boldsymbol{\theta}}^T & \boldsymbol{\Sigma}_{\boldsymbol{\theta}} \end{bmatrix} \quad (12)$$

In the context of sensor calibration, the keyframe $\mathbf{X}_{\mathcal{S}}^i$ and landmark states $\mathbf{L}_{\mathcal{S}}^i$ are considered nuisance variables and we are only interested in the marginal covariance $\boldsymbol{\Sigma}_{\boldsymbol{\theta}} = \text{Cov} [p(\boldsymbol{\theta}|\mathbf{U}_i, \mathbf{Z}_i)]$ of the calibration parameters $\boldsymbol{\theta}$. As \mathbf{R}_i is an upper-triangular matrix, we can efficiently obtain the marginal covariance $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ by back-substitution.

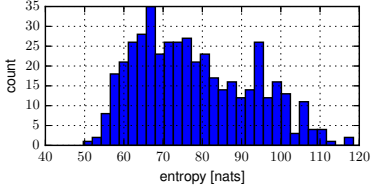


Fig. 5: Histogram of normalized segment entropies $H(\theta)$ over 450 segments from 15 datasets.

Before calculating the entropy, we normalize the marginal covariance Σ_{θ} to account for the different scales of the calibration parameters. The normalized covariance $\bar{\Sigma}_{\theta}$ is calculated as:

$$\bar{\Sigma}_{\theta} = \text{diag}(\sigma_{ref})^{-1} \cdot \Sigma_{\theta} \cdot \text{diag}(\sigma_{ref})^{-1} \quad (13)$$

where σ_{ref} is the expected standard deviation of $\hat{\theta}$ and was obtained from statistics over multiple reference segments. The differential entropy $H(\theta)$ of the normalized multivariate normal distribution $\bar{p}_{\theta}(\theta) = \bar{p}(\theta|U_i, Z_i)$ can then be calculated as:

$$\begin{aligned} H(\theta) &= - \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \bar{p}_{\theta}(\theta) \ln \bar{p}_{\theta}(\theta) d\theta \\ &= \frac{1}{2} \ln \left((2\pi e)^k \cdot \det(\bar{\Sigma}_{\theta}) \right), \end{aligned} \quad (14)$$

where k is the dimension of the normal distribution.

The segment entropy $H(\theta)$ is not a directional measure and thus summarizes the information of all parameters θ in a single scalar value. For high-dimensional calibration vectors θ , however, the contribution of well-observable modes to the entropy might shadow weaker modes despite normalization. This effect causes the distribution of the entropies to remain multimodal (as shown in Fig. 5) because the number of informative segments vs. less informative segments is in general not distributed equally within a given dataset.

For this reason, the vector of calibration parameters θ is partitioned into Q sub-vectors θ_q as:

$$\theta = \left[\tilde{\theta}_0^T \quad \dots \quad \tilde{\theta}_Q^T \right]^T \quad (15)$$

The marginal entropy is calculated for each parameter group q using the corresponding marginal covariance $\bar{\Sigma}_{\tilde{\theta}_q}$ as described in Eq. (14). The marginal segment entropies $H(\theta_q)$ are then directly used as a measure of information contained in the segment w.r.t. to the parameters of group q (where a lower entropy corresponds to richer information).

In this work, we partition the parameters θ into three groups by sensor:

$$\begin{aligned} \tilde{\theta}_0 &= [\mathbf{s}_g^T \quad \mathbf{m}_g^T \quad \mathbf{s}_a^T \quad \mathbf{m}_a^T \quad \mathbf{q}_{AI}^T]^T \\ \tilde{\theta}_1 &= [\mathbf{f}^T \quad \mathbf{c}^T \quad w]^T \\ \tilde{\theta}_2 &= [\mathbf{q}_{CI}^T \quad \mathbf{c}_{PCI}^T]^T \end{aligned} \quad (16)$$

This follows the intuition that the problem exhibits different co-observability structures i.e. a set of parameters is always observable as a group e.g. the camera model requires only

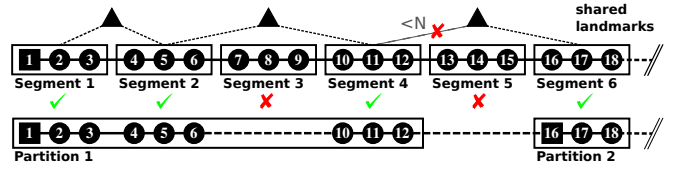


Fig. 6: The upper graph shows the full keyframe/landmark graph where the keyframes are assigned into fixed-size segments and the segments found to be informative are marked by green check-marks. The motion segments of the sparsified problem (shown below) can be partitioned into disjoint sets that are neither connected through inertial constraints nor share more than N landmark observations with other partitions. During optimization, the structurally unobservable states are fixed for exactly one keyframe per partition (marked by a square: e.g. 1)

minimal motion (once the landmarks are initialized) whereas the inertial model requires sufficient excitation. A more thorough analysis of how to identify the co-observability structure and thus optimally group the parameters should be part of future work.

B. Collecting Informative Segments in a DB

A database with Q tables is maintained where each table retains the N most informative segments for the corresponding parameter group q . Segments can be in multiple tables if it is informative w.r.t. multiple parameter groups. Therefore, the complexity of the calibration problem has an upper bound, as the max. number of segments in the database can be $Q \cdot N$ (or less if segments are in multiple tables).

It is important to note that the sum of segment entropies is a conservative approximation to the true information in the database for two reasons: First, the entropy is a scalar that “summarizes“ the information of several parameters and thus does not contain any directional information. Second, for efficiency, the segment entropy is calculated by neglecting the cross-terms to other segments. This approximation of the information in the database can lead to the collection of redundant segments in the database. Nevertheless, the very efficient evaluation of the segment entropies outweighs the run-time penalty from including such redundant segments into the optimization

C. Sparsified Problem using Informative Segments

The calibration over the set of informative segments differs from the full batch problem, described in Section III, in that non-informative segments have been removed. This results in missing inertial constraints between the remaining segments as shown in Fig. 6 (e.g. between keyframe 6/10 and 12/16). The set of segments can then contain partitions that are neither constrained to other partitions through inertial constraints nor by joint landmark observations. Each of these partitions can be seen as a (nearly) independent calibration problem only sharing calibration states with other partitions.

If we assume the availability of sufficient landmark constraints and non-degenerate motion (e.g. only rotation), then the visual-inertial calibration problem contains two structurally unobservable states: the global orientation around

the gravity vector and the global position. To ensure an optimal and efficient convergence of the iterative solvers these redundant degrees of freedom need to be held constant during optimization for exactly one keyframe in each of the partitions.

Algorithm 2 Partitioning segments on landmark co-visibility

Input: Set of motion segments $\mathbf{S} = \{\mathcal{S}_0, \dots, \mathcal{S}_K\}$

Input: Max. co-observed landmarks between partitions N

Result: Set of motion segment partitions \mathcal{P}

```

P ← {}
foreach  $\mathcal{S}_k \in \mathbf{S}$  do
  C ← {{ $\mathcal{S}_k$ }}
  foreach  $p \in \mathbf{P}$  do
    if CountSharedLandmarks( $p, \mathcal{S}_k$ ) > N then
      C ← C ∪ { $p$ }
    end
  end
end
 $p_C \leftarrow \text{MergePartitions}(C)$ 
P ← (P \ C) ∪ { $p_C$ }
end

```

Consequently, we identify these partitions by first joining all motion segments that have direct temporal neighbors into bigger segments (Fig. 6: e.g. segment 1 and 2). At this point, all keyframes within the joined segments are constrained through inertial constraints. The union-find algorithm, shown in Alg. 2, is then used to iteratively partition the segments into disjoint sets such that the count of co-observed landmarks between the partitions lies below a given threshold N (here: 15). This ensures that all keyframes within these partitions are either connected through inertial constraints or share sufficient landmark observations with other keyframes of the same partition. Degenerate landmark configurations are theoretically possible, when using such a heuristic landmark threshold, but are highly unlikely and would only affect the convergence rate but not bias the estimates.

Additionally, a constraint between two bias states is introduced if keyframes were removed between the two (Fig. 6: e.g. between keyframe 6-10 and 12-16). The bias evolution is modeled using a random walk as described in Eq. (III-B).

V. EXPERIMENTS AND RESULTS

A collection of 15 datasets is used to assess the performance of the proposed visual-inertial self-calibration method. The datasets were collected using a Google Tango Dev. Kit. tablet equipped with a MEMS IMU and a global shutter fisheye camera. The device was hand-held while recording multiple trajectories of 3 min duration while freely moving in a room of approx. 8x6 m with a height of 4 m. The trajectories consist of calmer sections and sections that excite all rotational and translational degrees of freedom. Fig. 7 shows an image of the experimental environment together with a top-down view on one of the recorded trajectories.

In this section, we discuss our evaluation results based on these datasets along the following questions:

- Does the sparsified calibration problem yield comparable results to the batch solution (Section V-A)?

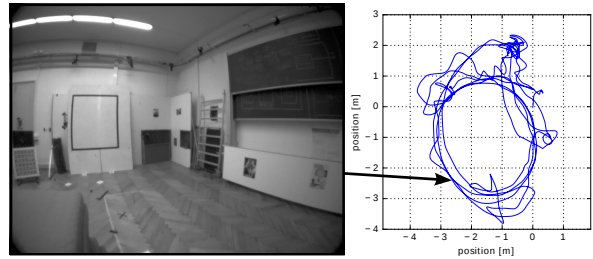


Fig. 7: Top-down view on the estimated trajectory of one of the evaluation datasets.

- Is the proposed measure capable of identifying informative segments (Section V-B)?
- Can the estimation be improved by grouping certain parameters and collecting segments for each group separately (Section V-C)?
- How does the proposed approach perform against comparable state-of-the-art methods in terms of run-time and estimation results (Section V-D)?

A. Performance and Repeatability of the Calibration

We compare the estimated parameters of the sparsified problem to the batch solution that uses all keyframes. The sparsified problem, here, denotes the calibration problem that only contains the most informative segments as described in Section IV. The initial calibration states were set to the CAD values, if available, otherwise to the expected nominal values (i.e. no sensor misalignment, unit scale factors). Table I shows the mean and standard deviation over the estimated parameters of the 15 different datasets and the convergence of the estimator is shown in Fig. 8. The mean of rotation parameters corresponds to the Rodrigues angle $\gamma(\cdot)$ of the averaged quaternion [19] over all data points and the standard deviation is calculated from the Rodrigues angles between the data points and the averaged quaternion.

These experiments show that the deviation between the sparsified estimation and the batch solution remains insignificant, in both the mean and standard deviation, even though large portions of the trajectory have been removed. This indicates that the proposed method can sparsify the problem while retaining an estimation quality close the batch solution at a drastically reduced run-time. It is important to note, that we cannot evaluate the accuracy of the estimated parameters as no ground-truth data is available, the statistics, however, give a good indication of the precision that can be achieved.

B. Evaluation of Segment Entropy to Select Informative Segments

In this section, we conduct an experiment to investigate the suitability of the segment entropy to identify informative segments. For this reason, we let the estimator from Section IV collect the least informative segments and compare the convergence of two parameters with results obtained by selecting the most informative segments. In both cases, we collect 8 segments each of which consists of 40 keyframes resulting in a total of 320 keyframes which is equal to

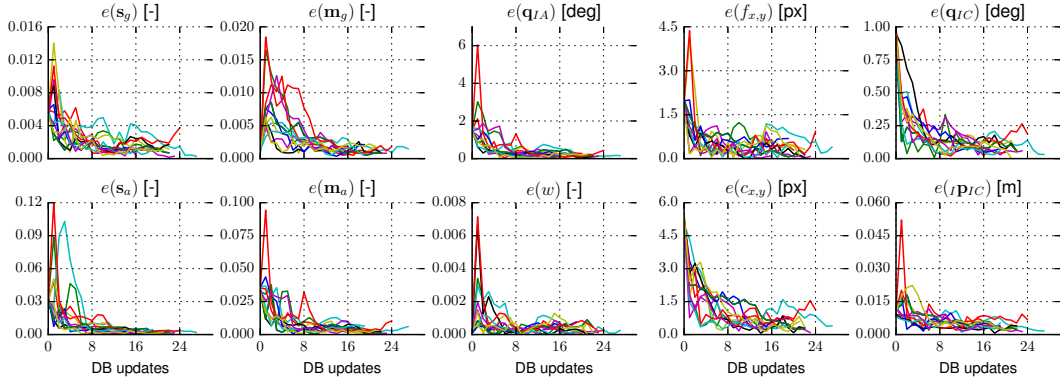


Fig. 8: Convergence of the calibration parameters for each update of the database i.e. when a new segment is added or a less informative segment is replaced. The y-axis shows the deviation to the batch solution as: $e(\mathbf{x}) = \|\hat{\mathbf{x}}(k) \boxminus \hat{\mathbf{x}}_{batch}\|$.

TABLE I: Estimated calibration parameters for three different algorithms. The statistics are taken over 15 datasets and show the mean and standard deviation. The number of used segments and run-time can be found in Table III.

parameter	informative (proposed method)	all segments (batch)	related work [11]
f [px]	254.71 \pm 0.28 254.63 \pm 0.29	254.50 \pm 0.13 254.47 \pm 0.14	254.89 \pm 0.35 254.68 \pm 0.32
c [px]	317.26 \pm 0.32 244.61 \pm 0.45	317.51 \pm 0.18 244.56 \pm 0.21	317.74 \pm 0.35 242.87 \pm 0.67
w [-]	0.9222 \pm 0.0003	0.9222 \pm 0.0003	0.9227 \pm 0.0007
$s_g - 1$ [-]	6.17e-04 \pm 9.61e-04 5.80e-03 \pm 8.51e-04 8.54e-04 \pm 3.89e-04	4.45e-05 \pm 3.52e-04 5.56e-03 \pm 5.36e-04 8.44e-04 \pm 1.97e-04	5.40e-04 \pm 1.10e-03 4.78e-03 \pm 1.53e-03 4.00e-04 \pm 6.74e-04
$s_a - 1$ [-]	-2.07e-02 \pm 2.28e-03 -1.73e-02 \pm 1.25e-03 -1.42e-02 \pm 1.34e-03	-2.07e-02 \pm 1.47e-03 -1.77e-02 \pm 5.21e-04 -1.49e-02 \pm 5.39e-04	-2.15e-02 \pm 2.33e-03 -1.82e-02 \pm 1.01e-03 -1.45e-02 \pm 1.08e-03
m_g [-]	3.44e-04 \pm 6.48e-04 1.07e-03 \pm 8.49e-04 7.38e-04 \pm 8.36e-04	7.42e-05 \pm 3.23e-04 1.23e-03 \pm 4.75e-04 4.31e-04 \pm 5.02e-04	2.94e-04 \pm 7.52e-04 1.42e-03 \pm 1.16e-03 6.75e-04 \pm 5.51e-04
$\gamma(q_{GA})$ [deg]	1.467 \pm 0.141	1.498 \pm 0.056	1.501 \pm 0.060
m_a [-]	1.78e-02 \pm 4.58e-03 -2.91e-02 \pm 3.02e-03 -1.18e-05 \pm 1.80e-03	1.79e-02 \pm 2.10e-03 -2.95e-02 \pm 1.35e-03 1.13e-04 \pm 1.14e-03	1.82e-02 \pm 1.88e-03 -3.00e-02 \pm 1.75e-03 -1.62e-03 \pm 1.32e-03
c_{PIC} [m]	2.92e-03 \pm 2.79e-03 1.25e-02 \pm 2.55e-03 -5.47e-03 \pm 2.86e-03	4.12e-03 \pm 1.01e-03 1.34e-02 \pm 1.37e-03 -5.68e-03 \pm 1.14e-03	-3.43e-03 \pm 3.42e-03 1.38e-02 \pm 1.94e-03 -2.81e-03 \pm 3.01e-03
$\gamma(q_{IC})$ [deg]	0.311 \pm 0.062	0.306 \pm 0.019	0.170 \pm 0.047

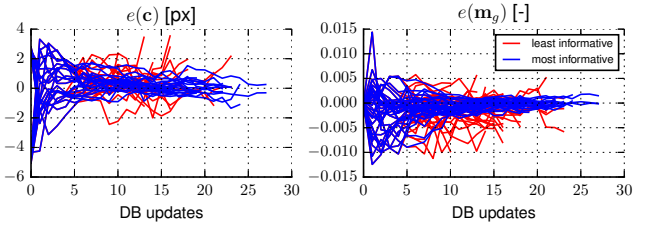
the data of 32 s. The convergence is shown in Fig. 9 together with the statistics on the final calibration states. The calibration using the set of least-informative segments yields a higher estimation error w.r.t. the batch solution and a higher variance than the estimation using the most-informative segments. This can be seen as an indication that:

- 1) the selection using the entropy identifies segments containing relevant information for sensor calibration,
- 2) the ratio of the number of selected segments to the total count of segments in the dataset is sufficiently low such that a careful selection is actually necessary.

C. Influence of Multiple Parameter Groups

In this section, we analyze the effect on the estimation performance when collecting segments for individual parameter groups. The estimator has been run with the parameter groups described in Section IV-B and as a comparison with a single group that contains all calibration parameters.

The Table II lists statistics of two estimated calibration parameters over 15 datasets. The results show that the



	most informative	least informative	batch
c [px]	317.26 \pm 0.32 244.61 \pm 0.45	318.24 \pm 0.96 244.85 \pm 0.81	317.51 \pm 0.18 244.56 \pm 0.21
m_g [-]	3.44e-04 \pm 6.48e-04 1.07e-03 \pm 8.49e-04 7.38e-04 \pm 8.36e-04	-9.71e-04 \pm 2.60e-03 1.57e-04 \pm 2.59e-03 -1.25e-03 \pm 3.75e-03	7.42e-05 \pm 3.23e-04 1.23e-03 \pm 4.75e-04 4.31e-04 \pm 5.02e-04

Fig. 9: Estimator performance when collecting the most-informative vs. the least-informative segments over 15 datasets. The plots show the deviation $e(\cdot)$ of the proposed method to the batch solution.

TABLE II: Estimated calibration parameters: single parameter group vs. multiple groups.

	multiple groups	single group	batch
c [px]	317.35 \pm 0.21 244.61 \pm 0.29	317.31 \pm 0.35 244.51 \pm 0.52	317.51 \pm 0.18 244.56 \pm 0.21
m_g [-]	1.64e-04 \pm 4.54e-04 1.09e-03 \pm 5.78e-04 5.33e-04 \pm 5.18e-04	2.60e-04 \pm 7.69e-04 1.05e-03 \pm 9.96e-04 8.68e-04 \pm 9.33e-04	7.42e-05 \pm 3.23e-04 1.23e-03 \pm 4.75e-04 4.31e-04 \pm 5.02e-04

variance of the estimates can be reduced by using multiple groups whereas the averaged error remains less affected. Intuitively, this effect can be explained as follows: If the problem structure contains groups of parameters that are rendered observable by different motion patterns and only a single informative database table is used then the chances are higher that only one type of motion is kept. If multiple groups are used, however, the content in the database gets more stable and thus leads to a lower variance of the estimates.

D. Comparison to Related Method and Run-time

In this section, we compare the proposed method against the work of [11] in terms of estimation performance and run-time. The latter work follows a similar approach that maintains a database of informative segments. A calibra-

TABLE III: Run-time and number of processed segments for the three estimators. Each segment contains 40 keyframes and corresponds to the data of 4 s. Statistics are collected over 15 datasets.

	our method	related work [11]	batch
num. segments	8.7 \pm 1.5	9.0	38.0 \pm 3.2
run-time [s]	31.2 \pm 5.6	395.9 \pm 319.6 (7745.3 \pm 4601.7)	178.5 \pm 94.0

tion is run over the candidate segment and all segments already contained in the database. The candidate is found to be informative if the information gain w.r.t. a calibration without the candidate segment lies above a certain threshold. Since a complete calibration must be run for each candidate evaluation the complexity grows with each new segment in the database. In contrast to the proposed method, this algorithm does consider all constraints when evaluating the information content of a candidate segment and does not make the assumption of segment independence as outlined in Section IV-A. Furthermore, they use a truncated QR instead of the Cholesky solver therefore it is more general and applicable for a wider range of problems although at a higher computational cost.

Two time points are given for the related work as it doesn't use an upper bound on the number of selected informative segments. The first until the same amount of informative segments are collected as in the proposed method (≈ 9) and the second (in brackets) until the information measure has been evaluated for each segment which is done in the proposed method by default. The same 15 datasets, used in the previous sections, have been processed with both methods. The run-times are shown in Table III and the estimated parameters in Table I. The results show that the run-time of our algorithm is considerably lower than the full-batch and related work at very similar estimation performance.

VI. CONCLUSIONS

In this work, we presented a novel method for efficient self-calibration of visual-inertial sensor systems that runs in parallel to an existing SLAM system. An information-theoretic measure is introduced to evaluate the information content of motion segments keeping a fixed number of the most-informative segments in a database. The proposed measure can be efficiently evaluated without running an expensive batch calibration beforehand. Once the database contains enough data, an optimization is run over these segments to update the calibration parameters.

Real-world experiments show that the sparsified problem yields similar results to the full batch solution at a significantly reduced computational cost. Even, when compared to previous work on segment based calibration, our approach shows a reduction of the run-time by a factor of approx. 10. Therefore, the proposed method is well suited for performing self-calibration on resource constrained platforms and can enable accurate operation over the entire lifespan.

VII. ACKNOWLEDGEMENT

The research leading to these results has received funding from Google's project Tango.

REFERENCES

- [1] M. Li, H. Yu, X. Zheng, and A. I. Mourikis, "High-fidelity sensor modeling and self-calibration in vision-aided inertial navigation," in *ICRA*, pp. 409–416, IEEE, 2014.
- [2] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," *IJRR*, vol. 34, no. 3, pp. 314–334, 2015.
- [3] M. Li and A. I. Mourikis, "High-precision, consistent EKF-based visual-inertial odometry," *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013.
- [4] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *IROS*, pp. 298–304, IEEE, 2015.
- [5] Z. Zhang, "A flexible new technique for camera calibration," *TPAMI*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [6] F. Devernay and O. Faugeras, "Straight lines have to be straight," *Machine vision and applications*, vol. 13, no. 1, pp. 14–24, 2001.
- [7] C. Krebs and J. Rehder, "Generic imu-camera calibration algorithm," *Semester thesis*, 2012.
- [8] J. Nikolic, M. Burri, I. Gilitschenski, J. Nieto, and R. Siegwart, "Non-parametric extrinsic and intrinsic calibration of visual-inertial sensor systems," *Sensors Journal*, vol. 16, no. 13, pp. 5433–5443, 2016.
- [9] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmänn, and R. Siegwart, "Extending kalibr: Calibrating the extrinsics of multiple imus and of individual axes," in *ICRA*, IEEE, 2016.
- [10] F. M. Mirzaei and S. I. Roumeliotis, "A Kalman filter-based algorithm for imu-camera calibration: Observability analysis and performance evaluation," *IEEE TOR*, vol. 24, no. 5, pp. 1143–1156, 2008.
- [11] J. Maye, P. Furgale, and R. Siegwart, "Self-supervised calibration for robotic systems," in *Intelligent Vehicles Symposium (IV)*, pp. 473–480, IEEE, 2013.
- [12] N. Keivan and G. Sibley, "Constant-time monocular self-calibration," in *ROBIO*, pp. 1590–1595, IEEE, 2014.
- [13] K. Hausman, J. Preiss, G. Sukhatme, and S. Weiss, "Observability-aware trajectory optimization for self-calibration with application to uavs," *arxiv*, 2016.
- [14] N. Trawny and S. I. Roumeliotis, "Indirect kalman filter for 3d attitude estimation," *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep.*, vol. 2, 2005.
- [15] S. Agarwal, K. Mierle, and Others, "Ceres solver." <http://ceres-solver.org>.
- [16] E. D. Nerurkar, K. J. Wu, and S. I. Roumeliotis, "C-klam: Constrained keyframe-based localization and mapping," in *ICRA*, pp. 3638–3643, IEEE, 2014.
- [17] S. Lynen, T. Sattler, M. Bosse, J. Hesch, M. Pollefeys, and R. Siegwart, "Get out of my lab: Large-scale, real-time visual-inertial localization," in *RSS*, 2015.
- [18] C. Hughes, P. Denny, M. Glavin, and E. Jones, "Equidistant fish-eye calibration and rectification by vanishing point extraction," *TPAMI*, vol. 32, no. 12, pp. 2289–2296, 2010.
- [19] F. L. Markley, Y. Cheng, J. L. Crassidis, and Y. Oshman, "Averaging quaternions," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 4, pp. 1193–1197, 2007.