Attitude Tracking from a Camera and an Accelerometer on Gyro-less Devices

Tien Do, Leo Neira, Yang Yang, and Stergios I. Roumeliotis

The authors are with the University of Minnesota, Minneapolis, MN 55455,
Emails: {doxxx104|neira005|yang5276}@umn.edu, stergios@cs.umn.edu

Abstract. In this paper, we address the problem of estimating the 3-DOF attitude of a mobile device using measurements from a camera and an accelerometer; two sensors that are typically found on most mobile devices. In contrast to previous approaches that combine measurements from both the gyroscopes and the accelerometers of an inertial measurement unit (IMU) to estimate attitude, we restrict ourselves to the case of *gyro-less devices*, of which over one billion are currently in use mostly in developing countries. Furthermore, and in order to support virtual and augmented reality (VR/AR) applications on low-cost devices with limiting processing, we introduce an efficient and robust algorithm where attitude is first estimated locally over a sliding window of three keyframes and subsequently is integrated within an extended Kalman filter (EKF) to track the device's global orientation. Additionally, gravity direction estimates, which are extracted from the accelerometer measurements when the device is not in motion, are used to update the roll and pitch attitude estimates, as well as the accelerometers biases, all of which, as we prove, are observable. The accuracy of the proposed method is assessed using the MAV EuRoC datasets [3], and it is shown to outperform alternative approaches relying on only visual data or the IMU, over a wide range of motions and conditions. Lastly, the efficiency of our algorithm is demonstrated on a Huawei 7A cell phone where it is able to run at 20 Hz on a single 1.4 GHz ARM Cortex A53 processor core.

Keywords: 3D attitude estimation, gyro-less devices, accelerometer-camera observability

1 Introduction and Related Work

Estimating a device's 3D attitude is necessary in many applications such as augmented reality (AR) and virtual reality (VR), on cell phones and tablets. Attitude can be simply estimated by integrating the signals of a 3-axis gyroscope that measures instantaneous rotational velocity. There exist, however, numerous low-cost devices that are not equipped with gyroscopes (e.g., Alcatel PulseMix, Galaxy J2, Galaxy J5, HTC Desire, Huawei 7A, Oppo A37 - note that over one billion such devices exist in the market, mostly in the developing world), which makes it very challenging to create AR/VR applications for them. To address this limitation, in this paper, we introduce an algo-

rithm for orientation estimation in gyro-less mobile devices based solely on camera and accelerometer measurements.[1]

One way to estimate attitude, when a gyroscope is not available, is to formulate and solve a 6 degree-of-freedom (DOF) localization and mapping problem where images from a camera are used to track its position and orientation (pose), while incrementally building a map (e.g., PTAM [13] or ORB SLAM [18]). Then, one can extract the 3-DOF attitude estimates and use them for the purpose of their application. Such an approach, however, has very high computational overhead, due to the requirement to maintain a scale-consistent 3D point features' map. It is, therefore, prohibitive for real-time attitude-based applications on devices with limited processing.

On the other hand, methods that focus on estimating *only* attitude from visual information can be classified into two categories: Line-based [1], [14], and [17]; and feature-based [8] and [11]. Due to their high dependence on (i) the line detector's robustness and (ii) the existence of a sufficient number of line segments in the environment, line-based methods usually provide intermittent and often unreliable orientation estimates. Hence, in this work, we will solely focus on feature-based approaches. Specifically, [11] uses a 2-point (2pt) RANSAC to determine the relative orientation between *a pair* of images, from two feature correspondences, under the assumption that the distance to the scene is very large as compared to the camera's translation. This relative (two view) attitude estimate is then refined by performing least squares minimization [10], involving all inlier feature pair matches. Given the global orientation of the first image in this pair, this relative orientation is then integrated to obtain the global orientation of the second image. This incremental approach is employed for all consecutive image pairs in order to estimate their global orientation. The main advantage of this approach is that by employing the simplifying assumption that any (small) translation between consecutive image pairs is a (small) disturbance in the relative orientation estimate, it offers an efficient formulation for computing their relative orientation. It suffers, however, from large drift in global orientation, due to the fast accumulation of relative rotational errors.

To address this issue, [8] employs the 5pt RANSAC [21] to determine the (up-to-scale) relative translation of image pairs along with their relative orientation to reduce the 3-DOF drift. Furthermore, [8] estimates an extra set of relative orientations between a number of selected image pairs (e.g., non-consecutive) that have a substantial overlapping field of view. This extra information is used as additional constraints to ensure consistency among all global attitude estimates. Even though the method of [8] reduces the drift in attitude estimates, its computation is quite high for real-time resource-constrained applications (per [8] it runs at 5 Hz on a 3.2 GHz core i7 Intel desktop).

To perform real-time (at 20Hz), accurate 3-DOF attitude tracking on low-cost gyro-less devices, we propose a method that: (i) maintains a *sliding window* of poses along with a local map of point features and estimates the camera's local orientation within this map, then (ii) employs an extended Kalman filter (EKF) to first integrate this local (w.r.t. the sliding-window map) estimate with the global one and then correct its roll, pitch, and accelerometer biases, using gravity-direction measurements from an accelerometer. Initially, and similarly to PTAM, our sliding-window map is constructed

[1] Although many of these devices may be equipped with magnetometers, we do not consider them since they are susceptible to disturbances and bias from local and global magnetic distortion.

from a set of few keyframes; in contrast to PTAM, however, we do not need to maintain a consistent scale with respect to the first two keyframes. Subsequently, PTAM's extra thread for global mapping is not needed in our method, hence making it light, both in terms of processing and memory. To further reduce the cost of the map construction, our sliding-window comprises only 3 keyframes and we employ an efficient and robust 3-view bundle adjustment solver (see [6]). The main reasons we can realize these processing savings (as compared to PTAM and ORB-SLAM) are: (i) the global position is not required for our applications, hence the global map is not necessary and (ii) the global roll and pitch can be corrected every time gravity measurements are available.

Finally, we should note that due to the cost of image processing (extracting and tracking features), our method is slower than alternative ones that rely only on an inertial measurement unit (IMU) where the gyroscope measurements (instead of the camera's relative attitude) are used to propagate the device's global attitude. Their estimates though, as shown in our experiments, drift faster than those of our approach, and most importantly they cannot be employed on gyro-less mobile devices which are the main focus of this work. In summary, this paper makes the following contributions:

- An EKF-based accelerometer-aided visual-gyroscope estimator for jointly estimating the device's 3-DOF global orientation and the accelerometer's bias;
- A nonlinear observability analysis of our system proving that the roll, pitch and the accelerometer's biases are observable, while there exists only one unobservable direction: the rotation around global gravity (yaw);
- An extensive experimental evaluation of the accuracy and robustness of our method against those that use only a camera [11], [8], or an IMU [15], [16].

2 Technical Approach

2.1 Overview

Our objective is to efficiently estimate (at 20-30 Hz) the quaternion of orientation, ${}^{G}\mathbf{q}_{C_k}$, of every image frame $\{C_k\}$ w.r.t. an inertial global frame $\{G\}$, based on: (i) a sequence of camera images and (ii) accelerometer measurements. In particular, using only images, we create visual feature tracks to construct a local map from a set of at most 3 keyframes, while simultaneously localizing the current image within it. Specifically, our local map contains the following information:

- $\{\mathrm{M}\}$ - the local map frame of reference, which is subject to change over time, and its orientation w.r.t. $\{\mathrm{G}\}$, ${}^{G}\mathbf{q}_{M}$;
- A set of keyframes (KFs) and the visual tracks of interest points across them;
- The pose of each KF i w.r.t. the map, which is defined as a 4×4 homogeneous transformation matrix, ${}^{M}\mathbf{T}_{C_{KF_i}}$, as in [4];
- The 3D position w.r.t. the local map of each triangulated track feature j, ${}^{M}\mathbf{f}_{j}$;
- The status of the map $\{$`Initialized, Uninitialized`$\}$.

The details of initializing and maintaining the local map are presented in Sect. 2.5. Once the current image is localized w.r.t. the map, we employ an EKF to fuse this relative orientation with the previous image's global orientation, as well as the accelerometer's

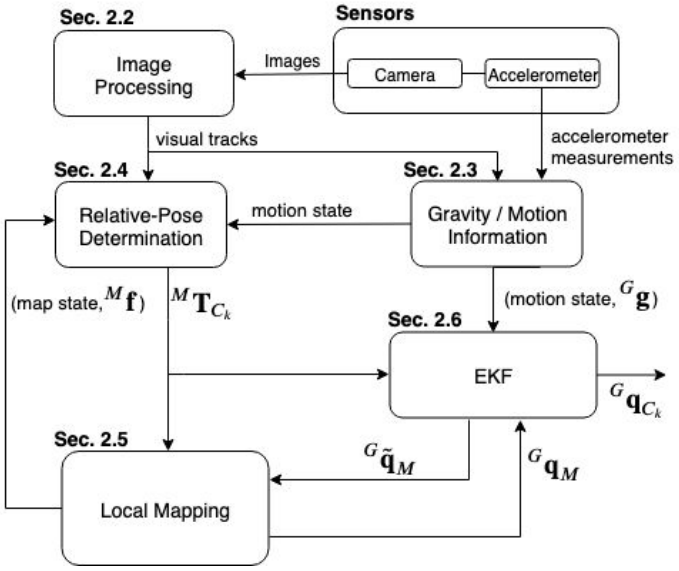

Fig. 1: Schematic block diagram of the proposed algorithm.

measurements, so as to obtain an updated attitude estimate. Fig. 1 depicts a schematic block diagram of our algorithm, which comprises the following 4 main modules:

- *Gravity/Motion Information* (Sect. 2.3) takes as input measurements from the accelerometer and the visual tracks in order to (i) determine whether the device's motion state is `static`, `semi-static`, or `moving`, (ii) extract the current gravity measurement ${}^{G}\mathbf{g}$ (when the device is not `moving`).
- *Relative Pose Determination* (Sect. 2.4) takes as input (i) the motion state (from *Gravity/Motion Information*), (ii) the local map state and triangulated points ${}^{M}\mathbf{f}$ (from *Local Mapping*), and (iii) their visual tracks (from *Image Processing*) in the current image C_k in order to determine its relative pose, ${}^{M}\mathbf{T}_{C_k}$ (comprising its position ${}^{M}\mathbf{p}_{C_k}$ and orientation ${}^{M}\mathbf{q}_{C_k}$), w.r.t. the local map $\{M\}$, using a combination of the 2pt, P3P [12], and 5pt RANSACs depending on the motion and map states. This differs from using only the 2pt RANSAC (as in [11]), or only the 5pt RANSAC (as in [8]).
- *Local Mapping* (Sect. 2.5) takes its input from the *Relative Pose Determination* and maintains the local map (defined previously) that shares a sufficient number of feature correspondences with the next image. To do so, it maintains a window of at most 3 keyframes and the corresponding 3D points viewed by them.
- *EKF Estimator* (Sect. 2.6), comprising the propagation and update submodules, recursively estimates the global orientation of the current image, ${}^{G}\mathbf{q}_{C_k}$, and the accelerometer's bias, $\mathbf{b}_{a_k}$. The propagator integrates the relative orientation estimate, ${}^{M}\mathbf{q}_{C_k}$, with the prior for the map's global orientation, ${}^{G}\mathbf{q}_{M}$ (from the *Local Mapping* module), to obtain a prior estimate for ${}^{G}\mathbf{q}_{C}$. The update submodule, then,

uses the gravity measurement from the *Gravity/Motion Information* module to update the accelerometer bias, $\mathbf{b}_{a_k}$, and the roll and pitch parts of ${}^{G}\mathbf{q}_{C_k}$ and ${}^{G}\mathbf{q}_{M}$.

In what follows, we describe each module in detail and highlight their relations.

2.2 Image Processing

Once a new image C_k is received, previously tracked features in C_{k-1}, along with newly extracted FAST [19] corners in C_{k-1} (to sum up to a fixed number of features, e.g., 500), are tracked into C_k using the Lucas-Kanade algorithm [2].[2] From these visual tracks, we extract $2D_{KF} - 2D_{C_k}$ correspondences, i.e., feature matches between a keyframe image and the current image. Furthermore, if a feature is tracked in more than 2 keyframes, it is triangulated to obtain a 3D position w.r.t. the map, which yields $3D_M - 2D_{C_k}$ correspondences. The $2D_{KF} - 2D_{C_k}$ and $3D_M - 2D_{C_k}$ correspondences between the current image and *any* keyframe image or *only* mapped points are then used in the *Relative Pose Determination* module (see Sect. 2.4).

2.3 Gravity / Motion Information

The objective of this module is to classify the system's motion into `static`, `semi-static`, or `moving`, and then (if the system is not `moving`) to extract gravity from the accelerometer's measurement. To determine the state of the system, we check the following two conditions on: (i) the accelerometer's measurements $\mathbf{a}_{m_{1,\ldots,n}}$ acquired between C_{k-1} and C_k and (ii) the rotation angle between C_{k-1} and C_k, ${}^{C_{k-1}}\theta_{C_k}$, estimated from the 2pt RANSAC:[3]

$$
\begin{aligned}
(\mathcal{C}_1): &\quad -c_s\sigma_a < ||\mathbf{a}_{m_i} - \hat{\mathbf{b}}_{a_{k-1}}|| - ||\mathbf{g}|| < c_s\sigma_a, \quad i = 1, \ldots, n \\
(\mathcal{C}_2): &\quad {}^{C_{k-1}}\theta_{C_k} < \theta_s
\end{aligned}
$$

where $(+/-)\, c_s\sigma_a$ represent the upper/lower bound of the acceleration's magnitude centered around the gravity's norm, $||\mathbf{g}||$; θ_s is the static motion threshold based on visual information,[4] and

$$
\mathbf{a}_{m_i} = \mathbf{R}({}^{G}\mathbf{q}_{C_i})^T({}^{G}\mathbf{a}_i - {}^{G}\mathbf{g}) + \mathbf{b}_{a_i} + \mathbf{n}_{a_i} \tag{1}
$$

where $\mathbf{n}_{a_i}$ is zero mean, white Gaussian noise with covariance $\sigma_a^2\mathbf{I}_3$; $\mathbf{b}_{a_i}$ is the accelerometer's bias modeled as a Brownian motion process with its incremental step following $\mathcal{N}(\mathbf{0}, \sigma_{ba}^2 \Delta t \mathbf{I}_3)$; and Δt is the time interval between two consecutive accelerometer measurements.[5]

[2] In special cases, e.g., blurry images or dramatic changes in illumination that cause all feature tracks to drift, we extract ORB features on both images and perform brute-force matching [18].

[3] If the inlier size returned by the 2pt RANSAC is less than 80% of the input size, the static condition on the rotation angle is not considered satisfied.

[4] We employ the visual constraint (with $\theta_s = 0.05$ deg) due to the fact that the accelerometer's measurements are contaminated by noise and bias, which can make the static detection unreliable.

[5] For simplicity, we consider the accelerometer and camera frames to coincide. In practice, we determine the relative orientation, ${}^{A}\mathbf{q}_{C}$, offline.

If both conditions, $(\mathcal{C}_1)$ and $(\mathcal{C}_2)$, hold, we classify the motion as `static`, i.e., ${}^{G}\mathbf{a}_i = \mathbf{0}$, and rewrite (1) as:

$$\mathbf{a}_{m_i} = -\mathbf{R}({}^{G}\mathbf{q}_{C_i})^{T}\,{}^{G}\mathbf{g} + \mathbf{b}_{a_i} + \mathbf{n}_{a_i}$$

then, we extract the gravity measurement as the average of all the accelerometer measurements between two images, after subtracting the bias estimate, i.e.,

$$\bar{\mathbf{g}}_{m_k} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{a}_{m_i} - \hat{\mathbf{b}}_{a_{k-1}}) \tag{2}$$

Note that from (2), and under the assumption that the gravity direction is constant between C_{k-1} and C_k, it is expressed as a Gaussian random variable:

$$\bar{\mathbf{g}}_{m_k} \sim \mathcal{N}\Big(-\mathbf{R}({}^{G}\mathbf{q}_{C_k})^{T}\,{}^{G}\mathbf{g}, \underbrace{\mathbf{P}(\widetilde{\mathbf{b}}_{a_{k-1}}) + (\frac{(n+1)(2n+1)}{6n}\sigma_{b_a}^2\Delta t + \frac{1}{n}\sigma_a^2)\mathbf{I}_3}_{\mathbf{\Psi}_m}\Big) \tag{3}$$

where the measurement covariance, $\mathbf{\Psi}_m$, comprises the previous bias' uncertainty, $\mathbf{P}(\widetilde{\mathbf{b}}_{a_{k-1}})$, with additional terms stemming from the bias' cross-correlations over time.

If only the first condition $(\mathcal{C}_1)$ is satisfied, we declare the state as `semi-static` and extract the gravity measurement as in (2), but with a higher noise covariance $\mathbf{\Psi'}_m = c_k\mathbf{\Psi}_m$, where $c_k = \exp\Big(\frac{1}{n}\sum_{i=1}^{n}\text{abs}(||\mathbf{a}_{m_i} - \hat{\mathbf{b}}_{a_{k-1}}|| - ||\mathbf{g}||)\Big)$. Finally, when neither condition is satisfied, we declare the system motion state as `moving`.

The gravity measurement, $\bar{\mathbf{g}}_{m_k}$, along with its covariance $\mathbf{\Psi}_m$ (or $\mathbf{\Psi'}_m$) is then provided to the *EKF-Update* module (Sect. 2.6) to correct the global roll and pitch, as well as the accelerometer bias. Additionally, the system's motion state is passed to *Relative-Pose Determination* described in the next section.

2.4 Relative-Pose Determination

The objective of this module is to compute ${}^{M}\mathbf{T}_{C_k}$, i.e., the relative pose of the current image, $\{C_k\}$, w.r.t. the current local map, $\{M\}$. To localize $\{C_k\}$ within $\{M\}$, we use a combination of the 2pt [5], P3P [12] and 5pt [21] RANSACs to compute the initial estimate followed by a 2- or 3-view bundle adjustment (BA) for refining it. To determine which RANSAC and BA to use, we employ the motion state and the local map's state, provided by the *Gravity/Motion Information*, and the *Local Mapping* (Sect. 2.5) modules, respectively. Specifically, we consider the following cases of (motion state, map state):

- Case 1 (`Static, any map state`): ${}^{M}\mathbf{T}_{C_k} = {}^{M}\mathbf{T}_{C_{k-1}}$, where we have used the 2pt RANSAC to detect this state, regardless of the map's state.
- Case 2 (`Moving, Uninitialized`): ${}^{M}\mathbf{T}_{C_k}$ = output of the 5pt followed by the 2-view BA($2D_M$-$2D_{C_k}$). In this case, the map points have not been triangulated yet; hence we use the 2D-2D pixel correspondences (see Sect. 2.2) between the oldest keyframe (the map's frame $\{M\}$ always coincides with the oldest keyframe - see Sect. 2.5) and the current frame, as input to the 5pt RANSAC to estimate the up-to-scale ${}^{M}\mathbf{T}_{C_k}$. This estimate is then refined by the 2-view (the oldest keyframe and the current frame) motion-only BA [5] using only the 5pt RANSAC's inliers.

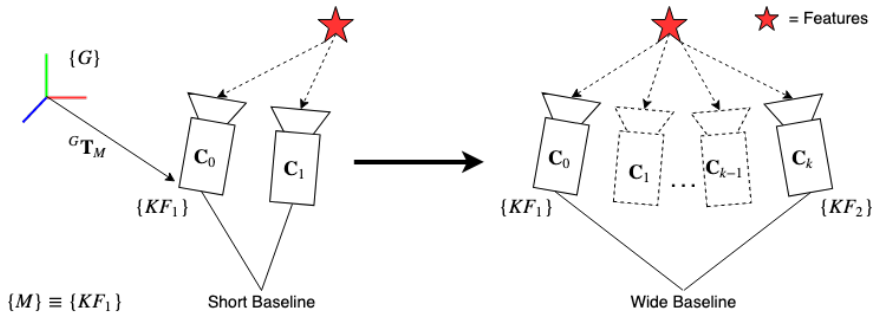

Fig. 2: (Left): First, C_0 is chosen to be KF_1. If there is a short baseline between KF_1 and subsequent frames, e.g., C_1, the map remains `Uninitialized`. (Right): When a frame, C_k, is found to have a wide baseline w.r.t. KF_1, it is selected as KF_2 and map points are triangulated between KF_1 and KF_2.

– Case 3 (`Moving, Initialized`): ${}^{M}\mathbf{T}_{C_k}$ = output of the P3P followed by the 3-view BA($3D_M$-$2D_{KF}$-$2D_{C_k}$). In this case, we extract 3D-2D correspondences between mapped points and 2D features in the current image as input to the P3P RANSAC to estimate ${}^{M}\mathbf{T}_{C_k}$. This estimate is then refined by the 3-view (the oldest keyframe, the second oldest keyframe, and the current frame) motion-only BA using the P3P RANSAC's inliers [6].

When the device is in motion, Case 3 is preferred over Case 2 due to the speed of the P3P solver (160x faster) as compared to the 5pt solver. The refined ${}^{M}\mathbf{T}_{C_k}$ is provided to the (i) *EKF-Propagation* to integrate the global orientation and (ii) *Local Map* to update the local map for the next image.

2.5 Local Mapping

The *Local Mapping* module provides 3D map points, ${}^{M}\mathbf{f}$, to *Relative-Pose Determination*, necessary for employing the P3P RANSAC. Specifically, a local map is built based on a sliding window of up to three keyframes (KF_j, KF_{j+1}, and KF_{j+2}). Our key objective here is to keep the oldest keyframe in the window for as long as possible in order to minimize the drift in the unobservable global yaw. To achieve this, the window slides only when the oldest keyframe no longer has a sufficient number of feature correspondences with C_k.[6] In this process, we define the frame of the local map, $\{M\}$, as that of the oldest keyframe. The mapping module takes as input the visual tracks (to determine if sliding the window is necessary) and the current image's pose, ${}^{M}\mathbf{T}_{C_k}$, w.r.t. the map, and outputs a set of triangulated 3D point features, ${}^{M}\mathbf{f}$. Additionally, it updates the state of the map ($\{$`Initialized`, `Uninitialized`$\}$) and provides the orientation of the map, ${}^{G}\mathbf{q}_{M}$, to the *EKF* to be used for propagation.

[6] In contrast to other sliding-window algorithms, in which the oldest keyframe is removed whenever a new keyframe is available, our method keeps accumulating keyframes and only uses the oldest 3 to create the structure.

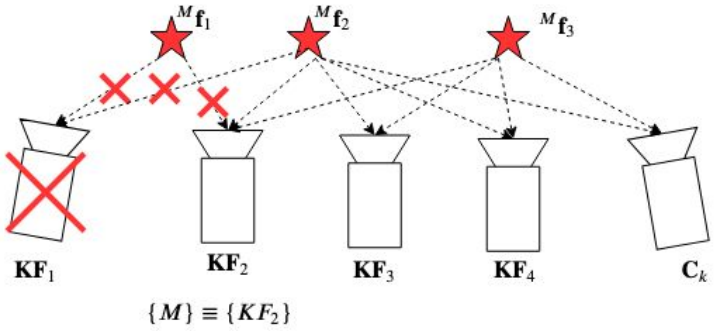

Fig. 3: Map sliding: When KF_1 no longer has enough correspondences with C_k, we slide the window to include KF_4 and remove KF_1. The map's reference frame will now be be that of KF_2, i.e., $\{M\} \equiv \{KF_2\}$. In addition, we remove all features that are no longer visible by C_k (e.g., $^M\mathbf{f}_1$) while updating the coordinates of those that are still tracked in C_k (e.g., $^M\mathbf{f}_2$, $^M\mathbf{f}_3$).

Map Initialization This process, depicted in Fig. 2, requires selecting two keyframes, KF_1 (typically C_0) and KF_2, with sufficient baseline and triangulating (using the 2-view BA) their common features, which become the map points, $^M\mathbf{f}$.

Keyframe Selection Additional keyframes (KF_{j+3}, KF_{j+4}, ..., KF_n), and their pose estimates w.r.t. the map, are kept, although not optimized in the 3-views BA, so as to facilitate the map window sliding process described next. In particular, an image C_k, is selected to be the next keyframe (i.e., KF_{n+1}) if it has a wide baseline w.r.t. to KF_n.

Map Window Sliding The map window needs to slide in order to maintain a sufficient number of 3D feature points within the view of C_k. Fig. 3 illustrates an example of the sliding process, in which the first keyframe in the window (e.g., KF_1) that no longer has enough correspondences with C_k, is removed and map frame $\{\mathrm{M}\}$ becomes the second oldest keyframe (e.g., KF_2). Removing a keyframe from the window leads into three cases, depending on the number of remaining keyframes that we need to consider: (i) one keyframe – we simply set the map's state to `Uninitialized`, (ii) two or three keyframes – we employ the 2- or 3-view BA to refine their poses and the 3D point features, and (iii) more than 3 keyframes – we employ 3-view BA involving only the 3 oldest keyframes [as in (ii)] and apply the proper transformation to express the rest of the keyframes w.r.t. the new local map frame $\{\mathrm{M}\}$.

The orientation of the new map, $^G\mathbf{q}_M$, is then provided to the *EKF* to be used for propagation as described in the next section.

2.6 EKF

In this section, we describe the process for estimating the state vector:

$$\mathbf{y}_{k+1} = \begin{bmatrix} {}^{G}\mathbf{q}_{M}^{T} & {}^{G}\mathbf{q}_{C_{k+1}}^{T} & \mathbf{b}_{a_{k+1}}^{T} \end{bmatrix}^{T} \tag{4}$$

and its covariance at every time step $k+1$. For any camera frame $\{C_\ell\}$, we follow [22] and employ the following relation between the true quaternion of orientation ${}^{G}\mathbf{q}_{C_\ell}$, its estimate (or measurement) ${}^{G}\hat{\mathbf{q}}_{C_\ell}$, and the error $\delta\mathbf{q}_{C_\ell}$:

$${}^{G}\mathbf{q}_{C_\ell} = \delta\mathbf{q}_{C_\ell} \otimes {}^{G}\hat{\mathbf{q}}_{C_\ell} \tag{5}$$

Propagation We start by expressing the orientation measurement of $\{C_{k+1}\}$ w.r.t. the map $\{M\}$, ${}^{M}\mathbf{q}_{C_{k+1}}^{(m)}$, provided by the *Relative-Pose Determination* module, as:

$${}^{M}\mathbf{q}_{C_{k+1}}^{(m)} = {}^{M}\mathbf{q}_{C_{k+1}} \otimes \delta\mathbf{q}_{n} \tag{6}$$

where $\delta\mathbf{q}_n \simeq \begin{bmatrix} \frac{1}{2}\mathbf{n}_\theta^T & 1 \end{bmatrix}^T$ is the quaternion representation of the measurement noise $\mathbf{n}_\theta$, with $\mathbf{n}_\theta \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_\theta)$, and then propagate the prior $\hat{\mathbf{y}}_{k|k}$ as:

$$\hat{\mathbf{y}}_{k+1|k} = \begin{bmatrix} {}^{G}\hat{\mathbf{q}}_{M_{k+1|k}} \\ {}^{G}\hat{\mathbf{q}}_{C_{k+1|k}} \\ \hat{\mathbf{b}}_{a_{k+1|k}} \end{bmatrix} = \begin{bmatrix} {}^{G}\hat{\mathbf{q}}_{M_{k|k}} \\ {}^{G}\hat{\mathbf{q}}_{M_{k|k}} \otimes {}^{M}\mathbf{q}_{C_{k+1}}^{(m)} \\ \hat{\mathbf{b}}_{a_{k|k}} \end{bmatrix} \tag{7}$$

The error state $\widetilde{\mathbf{y}}_{k|k} = \begin{bmatrix} {}^{G}\widetilde{\boldsymbol{\theta}}_{M_{k|k}}^{T} & {}^{G}\widetilde{\boldsymbol{\theta}}_{C_{k|k}}^{T} & \widetilde{\mathbf{b}}_{a_{k|k}}^{T} \end{bmatrix}^{T}$ is propagated as:

$$\widetilde{\mathbf{y}}_{k+1|k} = \begin{bmatrix} {}^{G}\widetilde{\boldsymbol{\theta}}_{M_{k+1|k}} \\ {}^{G}\widetilde{\boldsymbol{\theta}}_{C_{k+1|k}} \\ \widetilde{\mathbf{b}}_{a_{k+1|k}} \end{bmatrix} = \begin{bmatrix} {}^{G}\widetilde{\boldsymbol{\theta}}_{M_{k|k}} \\ {}^{G}\widetilde{\boldsymbol{\theta}}_{M_{k|k}} - \mathbf{R}({}^{G}\hat{\mathbf{q}}_{C_{k+1|k}})\mathbf{n}_\theta \\ \widetilde{\mathbf{b}}_{a_{k|k}} + \mathbf{n}_{b_a} \end{bmatrix} = \widetilde{\mathbf{y}}_{k|k} + \begin{bmatrix} \mathbf{0} \\ -\mathbf{R}({}^{G}\hat{\mathbf{q}}_{C_{k+1|k}})\mathbf{n}_\theta \\ \mathbf{n}_{b_a} \end{bmatrix}$$

where ${}^{G}\widetilde{\boldsymbol{\theta}}_{C_{k+1}}$ is propagated as in [22] and $\mathbf{n}_{b_a} \sim \mathcal{N}(\mathbf{0}, \sigma_{b_a}^2 \Delta t \mathbf{I}_3)$ as in Sect. 2.3. Then, the propagated covariance matrix is:

$$\mathbf{P}_{k+1|k} = \mathbf{P}_{k|k} + \mathbf{Q}_{k+1}, \quad \text{where } \mathbf{Q}_{k+1} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & {}^{G}\mathbf{R}_{C_{k+1}}\mathbf{Q}_\theta {}^{G}\mathbf{R}_{C_{k+1}}^{T} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \sigma_{b_a}^2 \Delta t \mathbf{I}_3 \end{bmatrix} \tag{8}$$

Update Given a measurement of the gravity, $\bar{\mathbf{g}}_{m_k}$ [see (2)], we employ the small-angle approximation on ${}^{G}\widetilde{\boldsymbol{\theta}}_{C_{k+1}}$ to obtain the following linear relationship between the measurement residual, $\mathbf{r}$, the measurement Jacobian, $\mathbf{H}$, and the error state, $\widetilde{\mathbf{y}}_{k+1|k+1}$:

$$\underbrace{\bar{\mathbf{g}}_{m_{k+1}} + \mathbf{R}({}^{G}\hat{\mathbf{q}}_{C_{k+1}})^{T}{}^{G}\mathbf{g} - \hat{\mathbf{b}}_{a_{k+1}}}_{\mathbf{r}} = \underbrace{\begin{bmatrix} \mathbf{0}_{3\times 3} & \mathbf{R}({}^{G}\hat{\mathbf{q}}_{C_{k+1}})^{T}\lfloor {}^{G}\mathbf{g} \times \rfloor & \mathbf{I}_3 \end{bmatrix}}_{\mathbf{H}} \widetilde{\mathbf{y}}_{k+1|k+1} + \mathbf{n}_g$$

where $\mathbf{n}_g \sim \mathcal{N}(\mathbf{0}, \mathbf{\Psi}_m \text{ or } \mathbf{\Psi'}_m)$ is the gravity measurement's noise, as defined in Sect. 2.3. We then update the posterior error state as:

$$\tilde{\mathbf{y}}_{k+1|k+1} = \begin{bmatrix} {}^G\widetilde{\boldsymbol{\theta}}_{M_{k+1|k+1}} \\ {}^G\widetilde{\boldsymbol{\theta}}_{C_{k+1|k+1}} \\ \widetilde{\mathbf{b}}_{a_{k+1|k+1}} \end{bmatrix} = \underbrace{\mathbf{P}_{k+1|k}\mathbf{H}^T\mathbf{S}^{-1}}_{\mathbf{K}\text{ : Kalman gain}}\mathbf{r}, \text{ where } \mathbf{S} \triangleq \mathbf{H}\mathbf{P}_{k+1|k}\mathbf{H}^T + \mathbf{\Psi}_m \quad (9)$$

Lastly, the posterior state and covariance are updated as:

$$\hat{\mathbf{y}}_{k+1|k+1} = \begin{bmatrix} {}^G\hat{\mathbf{q}}_{M_{k+1|k+1}} \\ {}^G\hat{\mathbf{q}}_{C_{k+1|k+1}} \\ \hat{\mathbf{b}}_{a_{k+1|k+1}} \end{bmatrix} = \begin{bmatrix} \delta\mathbf{q}({}^G\widetilde{\boldsymbol{\theta}}_{M_{k+1|k+1}}) \otimes {}^G\hat{\mathbf{q}}_{M_{k+1|k}} \\ \delta\mathbf{q}({}^G\widetilde{\boldsymbol{\theta}}_{C_{k+1|k+1}}) \otimes {}^G\hat{\mathbf{q}}_{C_{k+1|k}} \\ \hat{\mathbf{b}}_{a_{k+1|k}} + \widetilde{\mathbf{b}}_{a_{k+1|k+1}} \end{bmatrix} \quad (10)$$

$$\mathbf{P}_{k+1|k+1} = (\mathbf{I}_9 - \mathbf{K}\mathbf{H})\mathbf{P}_{k+1|k}(\mathbf{I}_9 - \mathbf{K}\mathbf{H})^T + \mathbf{K}\mathbf{\Psi}_m\mathbf{K}^T \quad (11)$$

Note that, if the motion state is `semi-static` (see Sect. 2.3) and in order to improve robustness to the unmodeled body acceleration contaminating (2), we choose not to update $\mathbf{b}_{a_{k+1}}$. Therefore, instead of performing full state update, we employ the Schmidt Kalman filter [20] to only update the map's and current frame's orientations. In this case, the state update equation becomes:

$$\hat{\mathbf{y}}_{k+1|k+1} = \begin{bmatrix} {}^G\hat{\mathbf{q}}_{M_{k+1|k+1}} \\ {}^G\hat{\mathbf{q}}_{C_{k+1|k+1}} \\ \hat{\mathbf{b}}_{a_{k+1|k+1}} \end{bmatrix} = \begin{bmatrix} \delta\mathbf{q}({}^G\widetilde{\boldsymbol{\theta}}_{M_{k+1|k+1}}) \otimes {}^G\hat{\mathbf{q}}_{M_{k+1|k}} \\ \delta\mathbf{q}({}^G\widetilde{\boldsymbol{\theta}}_{C_{k+1|k+1}}) \otimes {}^G\hat{\mathbf{q}}_{C_{k+1|k}} \\ \hat{\mathbf{b}}_{a_{k+1|k}} \end{bmatrix} \quad (12)$$

Note that $\hat{\mathbf{b}}_{a_{k+1|k+1}}$ remains the same, while the covariance update is still given by (11) but $\mathbf{K}$ is replaced by $\mathbf{K}' = \begin{bmatrix}\mathbf{K}_1^T & \mathbf{K}_2^T & \mathbf{0}_{3\times3}\end{bmatrix}^T$, where $\mathbf{K}_1$ and $\mathbf{K}_2$ are the submatrices of the Kalman gain corresponding to ${}^G\widetilde{\boldsymbol{\theta}}_{M_{k+1|k+1}}$ and ${}^G\widetilde{\boldsymbol{\theta}}_{C_{k+1|k+1}}$, respectively.

At this point, we should note that, the current keyframe, to which the map frame of reference is assigned to, may be removed in the next time step (see Sect. 2.5). Therefore, to be able to perform consistent EKF updates after the oldest keyframe is replaced, we need to keep the covariance of the second oldest (denoted as $M+1$) keyframe's orientation and its cross correlation with $\mathbf{b}_{a_{k+1}}$. This is approximated as follows:

$$\begin{aligned} {}^G\hat{\mathbf{q}}_{M+1_{k+1|k+1}} &= {}^G\hat{\mathbf{q}}_{M_{k+1|k+1}} \otimes {}^M\hat{\mathbf{q}}^{(m)}_{M+1} \\ \mathbf{P}_{k+1|k+1}({}^G\tilde{\boldsymbol{\theta}}_{M+1}, {}^G\tilde{\boldsymbol{\theta}}_{M+1}) &= \mathbf{P}_{k+1|k+1}({}^G\tilde{\boldsymbol{\theta}}_{M}, {}^G\tilde{\boldsymbol{\theta}}_{M}) + {}^G\mathbf{R}_{M+1}\mathbf{Q}_{\theta_{M,M+1}}{}^G\mathbf{R}^T_{M+1} \\ \mathbf{P}_{k+1|k+1}({}^G\tilde{\boldsymbol{\theta}}_{M+1}, \widetilde{\mathbf{b}}_{a_{k+1}}) &= \mathbf{P}_{k+1|k+1}({}^G\tilde{\boldsymbol{\theta}}_{M}, \widetilde{\mathbf{b}}_{a_{k+1}}) \end{aligned}$$

This simplification (as compared to the inclusion of ${}^G\mathbf{q}_{M+1}$ into the state vector) reduces slightly the complexity of the filter while making the implementation easier since we do not need to consider multiple cases based on the local map's state (i.e., whether to include the second oldest keyframe into the update or not).

3 Observability of the Accelerometer-aided Visual Gyroscope

In this section, we analyze the observability of the proposed system. Specifically, we define the state as: $\mathbf{x} = \begin{bmatrix} {}^C\mathbf{s}_G^T & \mathbf{b}_a^T \end{bmatrix}^T$, where ${}^C\mathbf{s}_G$ is the Cayley-Gibbs-Rodrigues[7] parameter describing the local orientation C w.r.t. the global G [4] and $\mathbf{b}_a$ is the accelerometer's bias. Then, we employ the following non-linear system with input $\boldsymbol{\omega}$ and output the gravity measurement:

$$\begin{cases} \dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{s}} \\ \dot{\mathbf{b}}_a \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\partial \mathbf{s}}{\partial \boldsymbol{\theta}} \\ \mathbf{0} \end{bmatrix}}_{\mathbf{f_1}} \boldsymbol{\omega} \\ \mathbf{z} = \mathbf{h}(\mathbf{x}) = \mathbf{b}_a + \mathbf{R}(\mathbf{s})^G\mathbf{g} \end{cases} \tag{13}$$

Next, applying the following change of variables $\boldsymbol{\beta}_1 = \mathbf{h}(\mathbf{x}) = \mathbf{b}_a + \mathbf{R}(s)^G\mathbf{g}$ and $\boldsymbol{\beta}_2 = \mathbf{b}_a$ results in the following nonlinear system:

$$\begin{cases} \dot{\boldsymbol{\beta}} = \begin{bmatrix} \dot{\boldsymbol{\beta}}_1 \\ \dot{\boldsymbol{\beta}}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \lfloor \boldsymbol{\beta}_1 - \boldsymbol{\beta}_2 \times \rfloor \\ \mathbf{0} \end{bmatrix}}_{\boldsymbol{\xi}} \boldsymbol{\omega} \\ \mathbf{h}(\boldsymbol{\beta}) = \boldsymbol{\beta}_1 \end{cases} \tag{14}$$

Next, following [9], we compute the span of the Lie derivatives of the resulting system up to the first order:

$$\begin{aligned} \mathcal{L}^0\mathbf{h} = \boldsymbol{\beta}_1 \Rightarrow \nabla\mathcal{L}^0\mathbf{h} &= \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3\times3} \end{bmatrix} \\ \mathcal{L}^1_{\boldsymbol{\xi}}\mathbf{h} = \langle \nabla\mathcal{L}^0\mathbf{h}, \boldsymbol{\xi} \rangle = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3\times3} \end{bmatrix} &\begin{bmatrix} \lfloor \boldsymbol{\beta}_1 - \boldsymbol{\beta}_2 \times \rfloor \\ \mathbf{0}_{3\times3} \end{bmatrix} = \lfloor \boldsymbol{\beta}_1 \times \rfloor - \lfloor \boldsymbol{\beta}_2 \times \rfloor \\ \Rightarrow \nabla\mathcal{L}^1_{\boldsymbol{\xi}}\mathbf{h} = \begin{bmatrix} -\lfloor \mathbf{e}_3 \times \rfloor & \lfloor \mathbf{e}_3 \times \rfloor \\ -\lfloor \mathbf{e}_2 \times \rfloor & \lfloor \mathbf{e}_2 \times \rfloor \\ -\lfloor \mathbf{e}_1 \times \rfloor & \lfloor \mathbf{e}_1 \times \rfloor \end{bmatrix}&, \quad \text{where } \mathbf{I}_3 = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \end{bmatrix} \end{aligned}$$

Using Gaussian elimination, we can show that the upper part $\begin{bmatrix} \nabla\mathcal{L}^0\mathbf{h} \\ \nabla\mathcal{L}^1_{\boldsymbol{\xi}}\mathbf{h} \end{bmatrix}$ of the observability matrix of system (14) is of full rank (rank 6), and hence this system is fully observable. Now, employing Theorem 4.1 in [9], it is straightforward to prove that the unobservable direction of the original system [described in (13)] is the null space $\mathbf{v}$ of the transformation matrix $\frac{\partial \boldsymbol{\beta}}{\partial \mathbf{x}}$:

$$\frac{\partial \boldsymbol{\beta}}{\partial \mathbf{x}} = \begin{bmatrix} \lfloor \mathbf{R}(s)^G\mathbf{g} \times \rfloor \frac{\partial \boldsymbol{\theta}}{\partial \mathbf{s}} & \mathbf{I}_3 \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 \end{bmatrix} \quad \text{with } \mathbf{v} \triangleq \begin{bmatrix} \frac{\partial \mathbf{s}}{\partial \boldsymbol{\theta}} \left(\mathbf{R}(s)^G\mathbf{g}\right) \\ \mathbf{0}_{3\times3} \end{bmatrix} \tag{15}$$

Note that since $\frac{\partial \boldsymbol{\theta}}{\partial \mathbf{s}}$ is of full rank, the matrix $\frac{\partial \boldsymbol{\beta}}{\partial \mathbf{x}}$ has rank 5 and its only null direction is $\mathbf{v}$, which is the rotation around the global gravity (i.e., the yaw).

[7] Even though our actual state comprises the global-in-local attitude, we chose to perform the analysis using the local-in-global attitude due to its simpler form of time derivatives. Note that this is without loss of generality, since there exists a bijective mapping ${}^G\mathbf{s}_C = -{}^C\mathbf{s}_G$.

4 Experimental Results

In this section, we experimentally evaluate the performance of our proposed accelerometer-aided visual-gyroscope algorithm and compare it with other state-of-the-art methods for 3-DOF orientation estimation, including a simplified version of the 5pt RANSAC visual-gyroscope *5pt* of [8][8] and the IMU-based complementary filter *IMU* of [16], on the MAV EuRoC datasets [3]. Note that in its simpler and most efficient form, the 5pt method integrates into a global orientation the relative orientation estimates (found from 5pt RANSAC followed by 2-view BA) of consecutive image pairs. In [8], however, a more complex variant of this approach is presented where at every time step, the relative orientations of an extra set of selected images needs to be computed. Although this results in better global orientation estimates, its very high computational requirements prohibit its use on low-end mobile devices. For this reason, in what follows, we compare our algorithm against the simpler form of the 5pt. Lastly, we time our implementation on a low-cost gyro-less cell phone; the Huawei 7A.

4.1 Performance Evaluation

The accuracy of the global and relative orientation estimates is assessed using as ground truth the ones provided in the MAV EuRoC datasets [3]. Specifically, the datasets comprise 11 sequences captured from a global shutter stereo (Aptina MT9V034, WVGA monochrome) with images available at 20 Hz; the stereo images are time synchronized with an onboard IMU (ADIS16448, 200 Hz), and ground-truth is given by VICON and Leica MS50. We only use the left camera images in our evaluations.

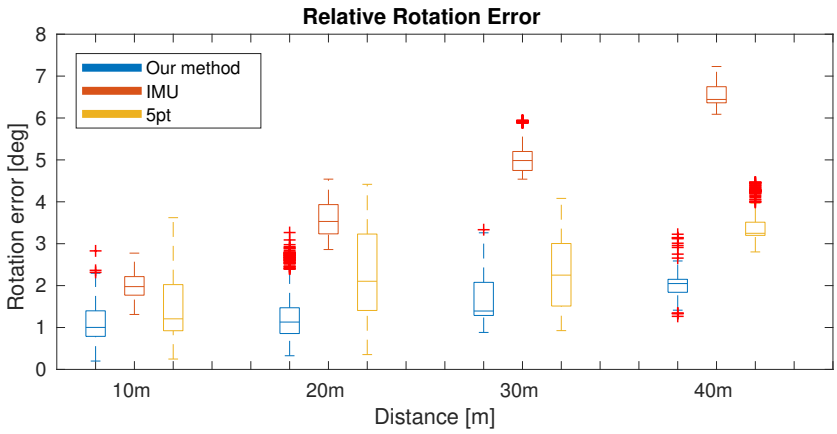

Fig. 4: Relative rotation error as in [7] for dataset MH_02 (easy).

[8]Note that we also compared our algorithm against the *2pt* method of [11]. Its median error, however, even for the MH_02 dataset (easy) was over 75 deg; hence, we did not consider it further.

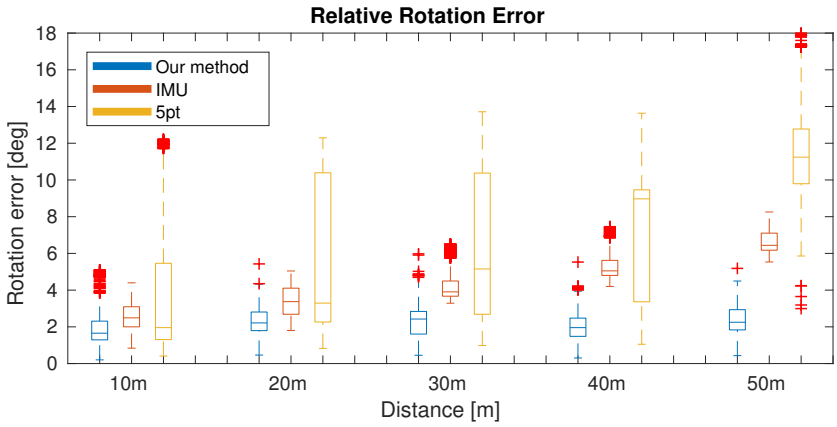

Fig. 5: Relative rotation error as in [7] for dataset MH_05 (difficult).

We compute the relative rotation error as described in [7] and compare our method with *5pt* and *IMU* over multiple trajectory lengths for the sequences MH_02 and MH_05. The relative rotation error for the dataset MH_02 (easy) is shown in Fig. 4. As evident, our method outperforms both the *IMU* and *5pt* methods. The *5pt* approach achieves its best results on MH_02, but it still has larger rotational drift than our method over longer trajectories. Similar results are shown for MH_05 (difficult) in Fig. 5, where the reduction in error when using our system is even more pronounced. The root mean

Sequence	Proposed	5pt	IMU
MH_01	**2.0552**	5.5615	2.0826
MH_02	**1.0696**	1.7978	5.0049
MH_03	**2.3241**	2.8940	7.5638
MH_04	**3.1869**	10.2277	6.9717
MH_05	**1.6105**	11.5828	8.6051
V1_01	**4.9397**	17.7179	5.1158
V1_02	**5.3341**	21.3663	7.0060
V1_03	9.0713	94.8403	**7.3946**
V2_01	3.6318	22.1934	**1.5978**
V2_02	**4.3306**	27.0637	8.0651
V2_03	14.7684	83.3954	**4.7618**
Mean	**4.7566**	27.1492	5.8336
Median	**3.6318**	17.7179	6.9717

Table 1: Absolute (global) orientation RMSE (deg) for our proposed algorithm, *5pt*, and *IMU*.

square error (RMSE) of the global roll, pitch, and yaw of the methods compared for all sequences of the EuRoC datasets are shown in Table 1. Our algorithm outperforms the *5pt* and the *IMU* most of the time, except for the three sequences V1_03, V2_01, and V2_03. In these 3 datasets, the scene changes drastically and the illumination varies

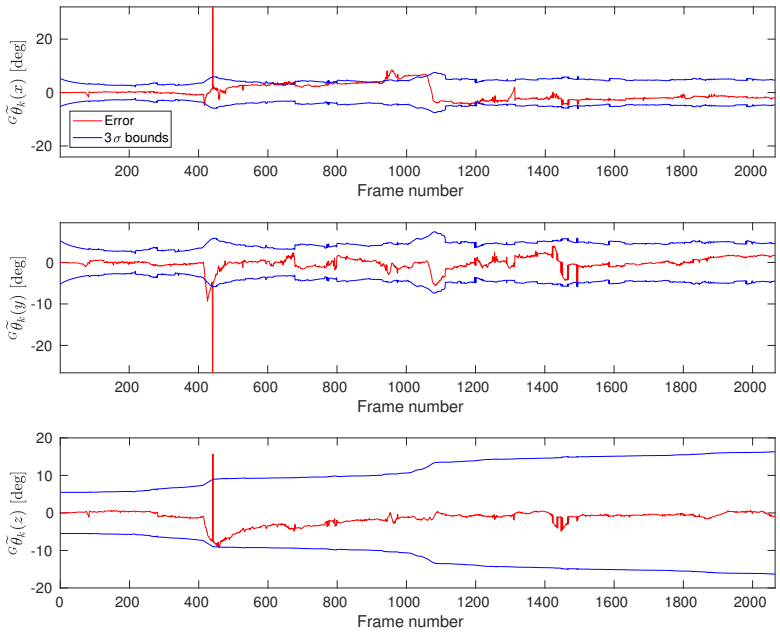

Fig. 6: Proposed algorithm: Orientation error and 3σ uncertainty bounds for each axis at each image frame $\{C_k\}$ w.r.t. the global frame $\{G\}$ for sequence V1_02. The spikes in the error are caused by abrupt illuminations changes.

abruptly, corrupting the visual tracks and causing a large error spike in the estimates (see Fig. 6). Through the use of the accelerometer, however, and the observability of the roll and pitch, we achieve a bounded error in the global roll and pitch, as shown in Fig. 6.

4.2 Timing

In order to assess the suitability of our algorithm for real-time applications, we used the MH sequences of EuRoC (due to its moderate changes in illumination) to time it on the Huawei 7A, which contains a Qualcomm Snapdragon 430 SoC designed for low/mid tier phones with 4 x 1.4 GHz ARM Cortex A53 processors and 4 x 1.1 GHz ARM Cortex A53 processors. All tests are performed using a single core. First, we time the 5pt and P3P solvers to justify the preference of our algorithm towards the use of the P3P RANSAC. On average, the P3P solver only takes 2.5 μs while the 5pt requires 400 μs (i.e., approximately 160x more). Since this timing is only for a single iteration of the solver, which typically runs $50 - 100$ times within RANSAC, it comes as no surprise that the full version of [8], which requires finding the relative orientation of multiple pairs of images at each time, cannot run in real time on low-end mobile processors.

Our attitude estimation algorithm runs in real-time at 20 Hz on the Huawei 7A phone with an average *Image Processing* time of 30 ms, while (on average) the combination of all other modules, i.e., *Relative-Pose Determination*, *Local Mapping* and *EKF*, takes 15 ms. One of the main reasons we are able to achieve real-time performance is our *Local Mapping* module, which relies on P3P RANSAC $\sim 80\%$ of the time for computing the relative orientation estimate, while the 5pt is only employed for the remaining 20%; this yields a maximum theoretical speedup of 5x over the simplified *5pt* of [8]. In practice though, and due to the overhead of the *Local Mapping* and 3-view BA, our algorithm realizes only a 2x speed up over the simplified 5pt, but achieves significantly better accuracy (see Table 1).

5 Conclusions and Future Work

In this paper, we presented an accelerometer-aided visual-gyroscope algorithm for tracking the 3D attitude of low-cost gyro-less devices. To achieve this objective, our algorithm jointly estimates the global orientation as well as the accelerometer's bias, using as inputs visual feature tracks (extracted from the cameras images) and gravity measurements (estimated from the accelerometer). In particular, the visual tracks are used to construct a sliding window map of 3 keyframes necessary for tracking the camera's relative (local) 3DOF attitude. This relative orientation is then fused with gravity measurements within an extended Kalman filter (EKF) to estimate the device's attitude, as well as the accelerometer biases. Furthermore, we proved that this system has only one unobservable degree of freedom, rotations around gravity (i.e., yaw), while the roll, pitch and the accelerometer biases can be accurately estimated as they are observable and hence their errors remain bounded. The proposed algorithm was compared against alternative methods that rely on visual or IMU only data using 11 sequences of the publicly available EuRoC dataset. In these experiments, we demonstrated that our algorithm typically achieves higher accuracy at a significantly lower processing cost. Finally, timings on a low-cost gyro-less cell phone, the Huawei 7A, verified that our algorithm is able to provide precise 3D attitude estimates at 20 Hz.

As part of our future work, we plan to improve our algorithm's accuracy and robustness by employing a relocalization module using bag-of-word features. Moreover, we will extend our algorithm to 6-DOF tracking and mapping in metric-scale.

Acknowledgements This work was supported by the University of Minnesota and the National Science Foundation (IIS-1328722).

References

1. J.-C. Bazin, Y. Seo, C. Demonceaux, P. Vasseur, K. Ikeuchi, I. Kweon, and M. Pollefeys. "Globally optimal line clustering and vanishing point estimation in manhattan world". In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 638–645, Providence, RI, Jun. 16–21 2012.
2. J.-Y. Bouguet et al. "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm". *Intel Corporation*, 2000.

3. M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart. "The euroc micro aerial vehicle datasets". *The International Journal of Robotics Research*, 35(10):1157–1163, Jan. 2016.
4. J. J. Craig. *"Introduction to robotics: mechanics and control"*. Pearson Prentice-Hall, Upper Saddle River, NJ, USA, 2005.
5. T. Do, L. C. Carrillo-Arce, and S. I. Roumeliotis. "High-speed autonomous quadrotor navigation through visual and inertial paths". *The International Journal of Robotics Research*, 38(4):486–504, Aug. 2019.
6. T. Do, L. Neira, and S. I. Roumeliotis. "Attitude tracking from a camera and an accelerometer on gyro-less devices". https://mars.cs.umn.edu/tr/attitude_gyroless.pdf, 2019.
7. A. Geiger, P. Lenz, and R. Urtasun. "Are we ready for autonomous driving? the kitti vision benchmark suite". In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, Providence, RI, Jun. 16–21 2012.
8. W. Hartmann, M. Havlena, and K. Schindler. "Visual gyroscope for accurate orientation estimation". In *Proc. of the IEEE Winter Conference on Applications of Computer Vision*, pages 286–293, Waikoloa, HI, Jan. 5–9 2015.
9. J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis. "Camera-imu-based localization: Observability analysis and consistency improvement". *The International Journal of Robotics Research*, 33(1):182–201, 2014.
10. B. Horn. "Closed-form solution of absolute orientation using unit quaternions". *Journal of the Optical Society of America A*, 4(4):629–642, Apr. 1987.
11. D. Kamran, M. Karimian, A. Nazemipour, and M. T. Manzuri. "Online visual gyroscope for autonomous cars". In *Proc. of the 24th Iranian Conference on Electrical Engineering (ICEE)*, pages 113–118, Shiraz, Iran, May 10–12 2016.
12. T. Ke and S. Roumeliotis. "An efficient algebraic solution to the perspective-three-point problem". In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7225–7233, Honolulu, HI, Jul. 21–26 2017.
13. G. Klein and D. Murray. "Parallel tracking and mapping for small ar workspaces". In *Proc. of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, ISMAR '07, pages 1–10, Nara, Japan, Nov. 13–16 2007.
14. J.-K. Lee and K.-J. Yoon. "Real-time joint estimation of camera orientation and vanishing points". In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1866–1874, Boston, MA, Jun. 7–12 2015.
15. E. J. Lefferts, F. L. Markley, and M. D. Shuster. "Kalman filtering for spacecraft attitude estimation". *Journal of Guidance, Control, and Dynamics*, 5(5):417–429, 1982.
16. R. Mahony, T. Hamel, and J. Pflimlin. "Nonlinear complementary filters on the special orthogonal group". *IEEE Transactions on Automatic Control*, 53(5):1203–1218, Jun. 2008.
17. F. M. Mirzaei and S. I. Roumeliotis. "Optimal estimation of vanishing points in a manhattan world". In *Proc. of the 2011 International Conference on Computer Vision*, pages 2454–2461, Barcelona, Spain, Nov. 6–13 2011.
18. R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. "Orb-slam: a versatile and accurate monocular slam system". *IEEE Transactions on Robotics*, 31(5):1147–1163, Oct. 2015.
19. E. Rosten and T. Drummond. "Machine learning for high-speed corner detection". In *European conference on computer vision*, pages 430–443. Springer, 2006.
20. S. F. Schmidt. "Application of state-space methods to navigation problems". In *Advances in control systems*, volume 3, pages 293–340. Elsevier, 1966.
21. H. Stewenius, C. Engels, and D. Nistér. "Recent developments on direct relative orientation". *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(4):284–294, Jun. 2006.
22. N. Trawny and S. I. Roumeliotis. "Indirect kalman filter for 3d attitude estimation". Technical report, Jan. 2005.