Modeling Varying Camera-IMU Time Offset in Optimization-Based Visual-Inertial Odometry

Yonggen Ling, Linchao Bao, Zequn Jie, Fengming Zhu, Ziyang Li,
Shanmin Tang, Yongsheng Liu, Wei Liu, and Tong Zhang

Tencent AI Lab, China
ylingaa@connect.ust.hk,{linchaobao,zequn.nus,fridazhu}@gmail.com,
{tzeyangli,mickeytang,kakarliu}@tencent.com, wl2223@columbia.edu,
tongzhang@tongzhang-ml.org

Abstract. Combining cameras and inertial measurement units (IMUs) has been proven effective in motion tracking, as these two sensing modalities offer complementary characteristics that are suitable for fusion. While most works focus on global-shutter cameras and synchronized sensor measurements, consumer-grade devices are mostly equipped with rolling-shutter cameras and suffer from imperfect sensor synchronization. In this work, we propose a nonlinear optimization-based monocular visual inertial odometry (VIO) with varying camera-IMU time offset modeled as an unknown variable. Our approach is able to handle the rolling-shutter effects and imperfect sensor synchronization in a unified way. Additionally, we introduce an efficient algorithm based on dynamic programming and red-black tree to speed up IMU integration over variable-length time intervals during the optimization. An uncertainty-aware initialization is also presented to launch the VIO robustly. Comparisons with state-of-the-art methods on the Euroc dataset and mobile phone data are shown to validate the effectiveness of our approach.

Keywords: Visual-Inertial Odometry, Online Temporal Camera-IMU Calibration, Rolling Shutter Cameras.

1 Introduction

Online, robust, and accurate localization is the foremost important component for many applications, such as autonomous navigation of mobile robots, online augmented reality, and real-time localization-based service. A monocular VIO that consists of one IMU and one camera is particularly suitable for this task as these two sensors are cheap, ubiquitous, and complementary. However, a VIO works only if both visual and inertial measurements are aligned spatially and temporally. This requires that both sensor measurements are synchronized and sensor extrinsics between sensors are known. While online sensor extrinsic calibration has gained lots of discussions in recent works, VIO with inperfect synchronization is less explored. Historically, some works [12, 14] calibrate the sensor time offsets offline, and assume that these parameters are not changed

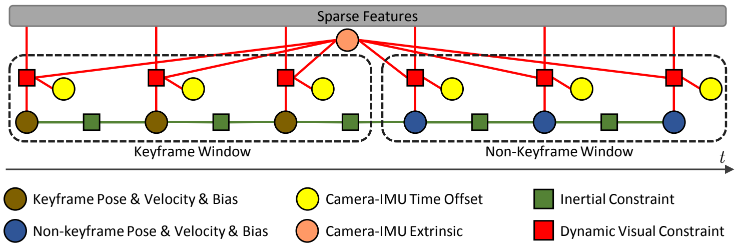


Fig. 1: The graph representation of our model. All variables (circles) and constraints (squares) in both the keyframe window and non-keyframe window are involved in the optimization. Note that modeling the camera-IMU time offset for each frame raises computational difficulties during the optimization, since the computation of visual constraints depends on the estimated time offset. In other words, the visual constraints in our model are "dynamic" due to the varying camera-IMU time offset.

in next runs. In real cases, time offsets change over time due to the variation of the system processing payload and sensor jitter. Other works [18, 10, 21] calibrate sensor time offsets online in an extended Kalman filter (EKF) framework. However, these methods suffer from the inherent drawbacks of filtering based approaches. They require a good prior about the initial system state (such as poses, velocities, biases, the camera-IMU extrinsic/time offsets) such that the estimation at each update step converges close to the global minimum. In contrast to filtering based approaches, nonlinear optimization-based methods [13, 17, 40, 4] iteratively re-linearize all nonlinear error costs from visual and inertial constraints to better treat the underlying nonlinearity, leading to increased tracking robustness and accuracy. However, introducing time offsets in a nonlinear optimization framework is non-trivial since the visual constraints are varying as they depend on the estimated time offsets that are varied between iterations. Another critical problem of VIO is the use of rolling-shutter cameras. Unlike global-shutter cameras that capture all rows of pixels at one time instant, rolling-shutter cameras capture each row of pixels at a different time instant. The rolling-shutter effect on captured images causes a significant geometry distortion if the system movement is fast. Without taking the rolling-shutter effect into account, the estimation performance degrades rapidly. Unfortunately, most consumer-grade cameras (such as cameras on mobile phones) are rolling-shutter cameras. If we optimize camera poses at every readout time of rolling-shutter cameras, the computational complexity will be intractable.

To the best of our knowledge, we are the first to propose a nonlinear optimization-based VIO to overcome the difficulties mentioned above. The graph representation of our model is shown in Fig. 1. Different from prior VIO algorithms based on nonlinear optimization, we incorporate an unknown, dynamically changing time offset for each camera image (shown as yellow circle in Fig.

1). The time offset is jointly optimized together with other variables like poses, velocities, biases, and camera-IMU extrinsics. We show that by modeling the time offset as a time-varying variable, imperfect camera-IMU synchronization and rolling-shutter effects can be handled in a unified formulation (Sect. 4.1 and Sect. 4.2). We derive the Jacobians involved in the optimization after introducing the new variable (Sect. 4.3), and show that the varying time offset brings computational challenges of pose computation over variable-length time intervals. An efficient algorithm based on dynamic programming and red-black tree is proposed to ease these difficulties (Sect. 4.4). Finally, since the nonlinear optimization is based on linearization, an initial guess is required for optimization bootstrap. A poor initialization may lead to a decrease of VIO robustness and accuracy. To improve the robustness of the system bootstrap, we present an initialization scheme, which takes the uncertainty of sensor measurements into account and better models the underlying sensor noises. Main contributions of this paper are as follows:

- We propose a nonlinear optimization-based VIO with varying camera-IMU time offset modeled as an unknown variable, to deal with the rolling-shutter effects and online temporal camera-IMU calibration in a unified framework.
- We design an efficient algorithm based on dynamic programming and red-black tree to speed up the IMU integration over variable-length time intervals, which is needed during optimization.
- We introduce an uncertainty-aware initialization scheme to improve the robustness of the VIO bootstrap.

Qualitative and quantitative results on the Euroc dataset with simulated camera-IMU time offsets and real-world mobile phone data are shown to demonstrate the effectiveness of the proposed method.

2 Related Work

The idea of VIO goes back at least to the work [35] proposed by Roumeliotis *et al.* based on filtering and the work [13] proposed by Jung and Taylor based on nonlinear optimization. Subsequently, lots of work have been published based on an exemplar implementation of filtering based approaches, called EKF [33, 19, 11, 27]. EKFs predict latest motions using IMU measurements and perform updates according to the reprojection errors from visual measurements. To bound the algorithmic complexity, many works follow a loosely coupled fashion [38, 26, 33, 24, 25, 23, **?**]. Relative poses are firstly estimated by IMU propagations as well as vision-only structure from motion algorithms separately. They are then fused together for motion tracking. Alternatively, approaches in a tightly coupled fashion optimize for estimator states using raw measurements from IMUs and cameras [27, 19]. They take the relations among internal states of different sensors into account, thus achieving higher estimation accuracy than loosely coupled methods at the cost of a higher computational complexity. Additionally, to benefit from increased accuracy offered by relinearization, nonlinear optimization-based methods iteratively minimize errors from both inertial measurements and visual measurements [17, 40, 4]. The main drawback of nonlinear optimization-based methods is their high computational complexity due to repeated linearizations, which can be lessened by limiting the variables to optimize and utilizing structural sparsity of the visual-inertial problem [17].

Recent approaches on VIOs consider the problem of spatial or temporal camera-IMU calibration. The camera-IMU relative transformation is calibrated offline [6] using batch optimization, or online by including it into the system state for optimization [19, 38, 39]. The temporal calibration between the camera and the IMU is a less-explored topic [12, 14, 18, 10]. Jacovitti *et al.* [12] estimate the time-offset by searching the peak that maximizes the correlation of different sensor measurements. Kelly *et al.* [14] firstly estimated rotations from different sensors independently, and then temporally aligned these rotations in the rotation space. However, both [12, 14] cannot estimate time-varying time offsets. Li *et al.* [18] adopted a different approach. They assume constant velocities around local trajectories. Time offsets are included in the estimator state vector, and optimized together with other state variables within an EKF framework. Instead of explicitly optimizing the time offset, Guo *et al.* [10] proposed an interpolation model to account for the pose displacements caused by time offsets.

While most works on VIOs use global-shutter cameras, deployments on consumer devices drive the need for using rolling-shutter cameras. A straightforward way to deal with rolling-shutter effects is to rectify images as if they are captured by global-shutter cameras, such as the work [15] proposed by Klein and Murray that assumes a constant velocity model and corrects distorted image measurements in an independent thread. For more accurate modeling, some approaches extend the camera projection function to take rolling-shutter effects into account. They represent local trajectories using zero order parameterizations [18, 20] or higher order parameterizations [36]. Instead of modeling the trajectories, [22] predicts the trajectories using IMU propagation and models the prediction errors of the estimated trajectories. These errors are represented as a weighted sum of temporal basis functions.

3 Preliminaries

In this section we briefly review the preliminaries of the nonlinear optimization framework used in our model. For detailed derivations, please refer to [32, 17].

We begin by giving notations. We consider $(\cdot)^w$ as the earth's inertial frame, and $(\cdot)^{b_k}$ and $(\cdot)^{c_k}$ as the IMU body frame and camera frame while taking the k^{th} image, respectively. We use $\mathbf{p}_Y^X$, $\mathbf{v}_Y^X$, and $\mathbf{R}_Y^X$ to denote the 3D position, velocity, and rotation of frame Y w.r.t frame X, respectively. The corresponding quaternion ($\mathbf{q}_Y^X = [q_x, q_y, q_z, q_w]$) for rotation is in Hamilton notation in our formulation. We assume that the intrinsic of the monocular camera is calibrated beforehand with known focal length and principle point. The relative translation and rotation between the monocular camera and the IMU are $\mathbf{p}_b^c$ and $\mathbf{q}_b^c$. The system-recorded time instant for taking the k^{th} image is t_k, while the image is actually captured at $\tilde{t}_k = t_k + \Delta t^o_k$, with an unknown time offset Δt^o_k due to inaccurate timestamps. Note that the time offset Δt^o_k is generally treated as a known constant in other optimization-based VIO algorithms, whereas it is modeled as an unknown variable for each image in our model.

In a sliding-window nonlinear optimization framework, the full state is usually encoded as $\mathcal{X} = [\mathbf{x}_{b_0} \ ... \ \mathbf{x}_{b_k} \ ... \ \mathbf{x}_{b_n} \ \mathbf{f}^w_0 \ ... \ \mathbf{f}^w_j \ ... \ \mathbf{f}^w_m \ \mathbf{p}^b_c \ \mathbf{q}^b_c]$, where a sub-state $\mathbf{x}_{b_k} = [\mathbf{p}^w_{b_k}, \mathbf{v}^w_{b_k}, \mathbf{q}^w_{b_k}, \mathbf{b}^{b_k}_a, \mathbf{b}^{b_k}_\omega]$ consists of the position, velocity, rotation, linear acceleration bias, and angular velocity bias at t_k, $\mathbf{f}^w_j$ is the 3D Euclidean position of feature j in the world coordinate, and $\mathbf{p}^b_c$ as well as $\mathbf{q}^b_c$ are the camera-IMU extrinsics. Finding the MAP estimate of state parameters is equivalent to minimizing the sum of the Mahalanobis norm of all measurement errors:

$$\min_{\mathcal{X}} \ ||\mathbf{b}_p - \mathbf{H}_p\mathcal{X}||^2 + \sum_{\hat{\mathbf{z}}^k_{k+1}\in S_i} ||r_i(\hat{\mathbf{z}}^k_{k+1}, \mathcal{X})||^2_{\mathbf{\Sigma}^k_{k+1}} + \sum_{\hat{\mathbf{z}}_{ik}\in S_c} ||r_c(\hat{\mathbf{z}}_{ik}, \mathcal{X})||^2_{\mathbf{\Sigma}_c}, \tag{1}$$

where $\mathbf{b}_p$ and $\mathbf{H}_p$ are priors obtained via marginalization [17], S_i and S_c are the sets of IMU and camera measurements, with the corresponding inertial and visual constraints modeled by residual functions $r_i(\hat{\mathbf{z}}^k_{k+1}, \mathcal{X})$ and $r_c(\hat{\mathbf{z}}_{ik}, \mathcal{X})$, respectively. The corresponding covariance matrices are denoted as $\mathbf{\Sigma}^k_{k+1}$ and $\mathbf{\Sigma}_c$.

To derive the inertial residual term $r_i(\hat{\mathbf{z}}^k_{k+1}, \mathcal{X})$ in Eq. (1), the IMU propagation model needs to be derived from the kinematics equation first, that is

$$\begin{aligned}
\mathbf{p}^w_{b_{k+1}} &= \mathbf{p}^w_{b_k} + \mathbf{v}^w_{b_k}\Delta t_k - \frac{1}{2}\mathbf{g}^w \Delta t^2_k + \mathbf{R}^w_{b_k}\hat{\boldsymbol{\alpha}}^k_{k+1}, \\
\mathbf{v}^w_{b_{k+1}} &= \mathbf{v}^w_{b_k} - \mathbf{g}^w \Delta t_k + \mathbf{R}^w_{b_k}\hat{\boldsymbol{\beta}}^k_{k+1}, \\
\mathbf{q}^w_{k+1} &= \mathbf{q}^w_k \otimes \hat{\mathbf{q}}^k_{k+1},
\end{aligned} \tag{2}$$

where $\Delta t_k = t_{k+1} - t_k$, $\mathbf{g}^w = [0, 0, 9.8]^T$ is the gravity vector in the earth's inertial frame, and $\hat{\mathbf{z}}^k_{k+1} = \{\hat{\boldsymbol{\alpha}}^k_{k+1}, \hat{\boldsymbol{\beta}}^k_{k+1}, \hat{\mathbf{q}}^k_{k+1}\}$ as well as its covariance $\mathbf{\Sigma}^k_{k+1}$ can be obtained by integrating linear accelerations $\mathbf{a}^{b_t}$ and angular velocities $\boldsymbol{\omega}^{b_t}$ [5]. Then the inertial residual term can be derived as:

$$r_i(\hat{\mathbf{z}}^k_{k+1}, \mathcal{X}) = \begin{bmatrix} \mathbf{R}^{b_k}_w(\mathbf{p}^w_{b_{k+1}} - \mathbf{p}^w_{b_k} - \mathbf{v}^w_{b_k}\Delta t_k + \frac{1}{2}\mathbf{g}^w \Delta t^2_k) - \hat{\boldsymbol{\alpha}}^k_{k+1} \\ \mathbf{R}^{b_k}_w(\mathbf{v}^w_{b_{k+1}} - \mathbf{v}^w_{b_k} + \mathbf{g}^w \Delta t_k) - \hat{\boldsymbol{\beta}}^k_{k+1} \\ (\hat{\mathbf{q}}^k_{k+1})^{-1}(\mathbf{q}^w_{b_k})^{-1}\mathbf{q}^w_{b_{k+1}} \end{bmatrix}. \tag{3}$$

The visual residual term $r_c(\hat{\mathbf{z}}_{ik}, \mathcal{X})$ in Eq. (1) is defined by the projection errors of tracked sparse features, which can be obtained using ST corner detector [34] and tracked across sequential images using sparse optical flow [1]. Note that, to handle the rolling-shutter effect, the generalized epipolar geometry [3] can be adopted as the fitted model in the RANSAC outlier removal procedure during correspondences establishment. Suppose a feature $\mathbf{f}^w_i$ in the world coordinate, following the pinhole model, its projection $\mathbf{u}^k_i$ on the k frame is:

$$\mathbf{u}^k_i = \begin{bmatrix} x^{c_k}_i / z^{c_k}_i \\ y^{c_k}_i / z^{c_k}_i \end{bmatrix}, \text{ where } \mathbf{f}^{c_k}_i = \begin{bmatrix} x^{c_k}_i \\ y^{c_k}_i \\ z^{c_k}_i \end{bmatrix} = \mathbf{R}^c_b(\mathbf{R}^{b_k}_w(\mathbf{f}^w_i - \mathbf{p}^w_{b_k}) - \mathbf{p}^b_c). \tag{4}$$

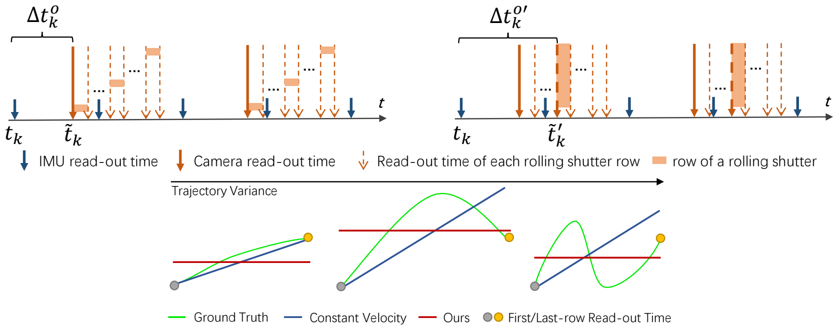


Fig. 2: The illustration of a camera-IMU sensor suite with a rolling-shutter camera and imperfect camera-IMU synchronization. We 'average' the rolling shutter readout time instants and approximate the rolling-shutter images (top-left) with global-shutter images captured at the 'middle-row' readout time of the rolling-shutter cameras (top-right). The position of this 'middle-row' is optimized to be the expected position of the local ground truth trajectory (bottom).

Then the projection error is $r_c(\hat{\mathbf{z}}_{ik}, \mathcal{X}) = \mathbf{u}_i^k - \hat{\mathbf{u}}_i^k$, where $\hat{\mathbf{u}}_i^k$ is the tracked feature location. The covariance matrix $\mathbf{\Sigma}_c$ is set according to the tracking accuracy of the feature tracker. By linearizing the cost function in Eq. (1) at the current best estimation $\hat{\mathcal{X}}$ with respect to error state $\delta\mathcal{X}$, the nonlinear optimization problem is solved via iteratively minimizing the following linear system over $\delta\mathcal{X}$ and updating the state estimation $\hat{\mathcal{X}}$ by $\hat{\mathcal{X}} \leftarrow \hat{\mathcal{X}} + \delta\mathcal{X}$ until convergence:

$$\begin{aligned} \min_{\delta\mathcal{X}} ||\mathbf{b}_p - \mathbf{H}_p(\hat{\mathcal{X}} + \delta\mathcal{X})||^2 + \sum_{\hat{\mathbf{z}}_{k+1}^k \in S_i} ||r_i(\hat{\mathbf{z}}_{k+1}^k, \hat{\mathcal{X}}) + \mathbf{H}_k \delta\mathcal{X}||^2_{\mathbf{\Sigma}_{k+1}^k} \\ + \sum_{\hat{\mathbf{z}}_{ik} \in S_c} ||r_c(\hat{\mathbf{z}}_{ik}, \hat{\mathcal{X}}) + \mathbf{H}_k^i \delta\mathcal{X}||^2_{\mathbf{\Sigma}_c}, \end{aligned} \tag{5}$$

where $\mathbf{H}_k$ and $\mathbf{H}_k^i$ are Jacobians of the inertial and visual residual functions.

4 Modeling Varying Camera-IMU Time Offset

In this section, we first show that rolling-shutter effects can be approximately compensated by modeling a camera-IMU time offset (Sect. 4.1). Then we present our time-varying model for the offset (Sect. 4.2) and the derivation of the Jacobian for optimization after introducing the time offset variable (Sect. 4.3). Finally, an efficient algorithm is described to accelerate the IMU integration over variable-length time intervals (Sect. 4.4), required by the optimization.

4.1 Approximate Compensation for Rolling-Shutter Effects

Fig. 2.(a) shows a camera-IMU suite with a rolling-shutter camera and imperfect camera-IMU synchronization. The system-recorded time instant for taking the

k^{th} image is t_k, which serves as our time reference for the retrieval of IMU data. Due to imperfect sensor synchronization (or inaccurate timestamps), the image is actually captured at $\tilde{t}_k = t_k + \Delta t_k^o$, with an unknown time offset Δt_k^o. With a rolling-shutter camera, this means that $\tilde{t}_k$ is the time instant when the camera starts to read out pixels row by row. Instead of modeling local trajectories using constant velocities [10, 15, 20], we model them with constant poses, which are expected poses of local trajectories (Fig. 2). With this approximation, a rolling-shutter image captured at $\tilde{t}_k$ with time offset Δt_k^o can be viewed as an global-shutter image captured at $\tilde{t}_k'$ with time offset $\Delta t_k^{o\prime}$. In the following, we slightly abuse notation by replacing $\Delta t_k^{o\prime}$ with Δt_k^o and $\tilde{t}_k'$ with $\tilde{t}_k$. We also replace $\mathbf{p}_{b_k}^w$ and $\mathbf{R}_w^{b_k}$ in Eq. (4) by $\tilde{\mathbf{p}}_{b_k}^w$ and $\tilde{\mathbf{R}}_w^{b_k}$ as they are now evaluated at time instant $\tilde{t}_k$. To calculate the pose at $\tilde{t}_k$, we use IMU propagation from the pose at t_k:

$$\begin{aligned}
\tilde{\mathbf{p}}_{b_k}^w &= \mathbf{p}_{b_k}^w + \mathbf{v}_{b_k}^w \Delta t_k^o - \frac{1}{2}\mathbf{g}^w (\Delta t_k^o)^2 + \mathbf{R}_{b_k}^w \hat{\boldsymbol{\alpha}}_{c_k}^{b_k}, \\
\tilde{\mathbf{q}}_{b_k}^w &= \mathbf{q}_{b_k}^w \otimes \hat{\mathbf{q}}_{c_k}^{b_k},
\end{aligned} \tag{6}$$

$$\begin{aligned}
\hat{\boldsymbol{\alpha}}_{c_k}^{b_k} &= \iint_{t\in[t_k,\tilde{t}_k]} \mathbf{R}_t^{b_k}(\mathbf{a}^{b_t} - \mathbf{b}_a^{b_k})dt^2, \\
\hat{\mathbf{q}}_{c_k}^{b_k} &= \int_{t\in[t_k,t_{c_k}]} \frac{1}{2} \begin{bmatrix} -\lfloor \boldsymbol{\omega}^{b_t} - \mathbf{b}_{\boldsymbol{\omega}}^{b_k} \rfloor_\times & \boldsymbol{\omega}^{b_t} - \mathbf{b}_{\boldsymbol{\omega}}^{b_k} \\ -(\boldsymbol{\omega}^{b_t} - \mathbf{b}_{\boldsymbol{\omega}}^{b_k})^T & 0 \end{bmatrix} \mathbf{q}_{b_t}^{b_k} dt
\end{aligned} \tag{7}$$

where $\mathbf{a}^{b_t}/\boldsymbol{\omega}^{b_t}$ is the instant linear acceleration/angular velocity. Since only discrete IMU measurements are available on IMUs, $\hat{\boldsymbol{\alpha}}_{c_k}^{b_k}$ and $\hat{\mathbf{q}}_{c_k}^{b_k}$ in (7) are approximately computed using numerical integration (i.e. mid-point integration).

The benefit of our constant-pose approximation is that additional variables, i.e. velocities and the rolling-shutter row time, are not needed for estimation, which leads to a large reduction of computational complexity.

4.2 Modeling Camera-IMU Time Offset

From the previous subsection, we see that the time offset Δt_k^o is the addition of two parts. The first part is the camera-IMU time offset, which varies smoothly because of the system payload variation and sensor jitter. The second part is the compensated time offset caused by the rolling-shutter effect approximation, which varies smoothly according to the change of local trajectories. We see Δt_k^o as a slowly time-varying quantity and model it as a Gaussian random walk: $\dot{\Delta t}_k^o = \mathbf{n}_o$, where $\mathbf{n}_o$ is zero-mean Gaussian noise with covariance $\boldsymbol{\Sigma}_o$. Since time offsets we optimize are at discrete time instants, we integrate this noise over the time interval between two consecutive frames in the sliding window $[t_k,\ t_{k+1}]$: $\Delta t_{k+1}^o = \Delta t_k^o + \mathbf{n}_k^o, \boldsymbol{\Sigma}_k^o = \Delta t_k \boldsymbol{\Sigma}_o$, where $\mathbf{n}_k^o$ and $\boldsymbol{\Sigma}_k^o$ are discrete noise and covariance, respectively. Thus we add $\|\Delta t_{k+1}^o - \Delta t_k^o\|_{\boldsymbol{\Sigma}_k^o}$ into Eq. (1) for all consecutive frames. By including constraints between consecutive time offsets, we avoid "offset jumping" between consecutive frames.

4.3 Optimization with Unknown Camera-IMU Time Offset

Our state vector at time instant t_k reads as $\mathbf{x}_{b_k} = [\mathbf{p}^w_{b_k}, \mathbf{v}^w_{b_k}, \mathbf{q}^w_{b_k}, \mathbf{b}^{b_k}_a, \mathbf{b}^{b_k}_\omega, \Delta t^o_k]$, where Δt^o_k is the dynamically changing camera-IMU time offset modeling both the approximate compensation of rolling-shutter effects and imperfect sensor synchronization. With Δt^o_k, the error state $\delta\mathcal{X}$ for linearization becomes $\delta\mathcal{X} = [\delta\mathbf{p}^w_{b_k}\ \delta\mathbf{v}^w_{b_k}\ \delta\boldsymbol{\theta}^w_{b_k}\ \delta\mathbf{b}^{b_k}_a\ \delta\mathbf{b}^{b_k}_\omega\ \delta\mathbf{p}^c_b\ \delta\boldsymbol{\theta}^c_b\ \delta\mathbf{f}^w_i\ \delta\Delta t^o_k]$, where we adopt a minimal error representation for rotations ($\delta\boldsymbol{\theta}_{b_k}, \delta\boldsymbol{\theta}^c_b \in \mathbb{R}^3$): $\mathbf{q}^w_{b_k} = \hat{\mathbf{q}}^w_{b_k} \otimes \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta}^w_{b_k} \\ 1 \end{bmatrix}$, $\mathbf{q}^c_b = \hat{\mathbf{q}}^c_b \otimes \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta}^c_b \\ 1 \end{bmatrix}$. Other error-state variables $\delta\mathbf{p}^w_{b_k}$, $\delta\mathbf{v}^w_{b_k}$, $\delta\mathbf{b}^{b_k}_a$, $\delta\mathbf{b}^{b_k}_\omega$, $\delta\mathbf{p}^c_b$, $\delta\mathbf{f}^w_i$, and $\delta\Delta t^o_k$ are standard additive errors. After introducing $\delta\Delta t^o_k$, the Jacobian $\mathbf{H}_k$ of the inertial residual function in Eq. (5) remains the same as before, while the Jacobian $\mathbf{H}^i_k$ of the visual residual function needs to be reformulated. Denoting $\hat{(\cdot)}$ the states obtained from the last iteration of the nonlinear optimization, the Jacobian $\mathbf{H}^i_k$ can be written as

$$\mathbf{H}^i_k = \frac{\partial r_c}{\partial \delta\mathcal{X}} = \frac{\partial r_c}{\partial \mathbf{f}^{c_k}_i}\frac{\partial \mathbf{f}^{c_k}_i}{\partial \delta\mathcal{X}} = \begin{bmatrix} \frac{1}{\hat{z}^{c_k}_i} & 0 & -\frac{\hat{x}^{c_k}_i}{\hat{z}^{c_k}_i} \\ 0 & \frac{1}{\hat{z}^{c_k}_i} & -\frac{\hat{y}^{c_k}_i}{\hat{z}^{c_k}_i} \end{bmatrix} \mathbf{J},$$

$$\mathbf{J} = [-\hat{\mathbf{R}}^c_b\tilde{\mathbf{R}}^{b_k}_w \ \ -\hat{\mathbf{R}}^c_b\tilde{\mathbf{R}}^{b_k}_w\Delta\hat{t}^o_k \ \ \hat{\mathbf{R}}^c_b \left\lfloor \tilde{\mathbf{R}}^{b_k}_w(\hat{\mathbf{f}}^w_i - \tilde{\mathbf{p}}^w_{b_k}) \right\rfloor_\times \ \mathbf{0}_{3\times 6} \ \ \hat{\mathbf{R}}^c_b \ \left\lfloor \hat{\mathbf{f}}^{c_k}_i \right\rfloor_\times \ \hat{\mathbf{R}}^c_b\tilde{\mathbf{R}}^{b_k}_w \ \mathbf{J}_{\delta\Delta t}],$$

$$\mathbf{J}_{\delta\Delta t} = \hat{\mathbf{R}}^c_b(\left\lfloor \boldsymbol{\omega}^{b_k} \right\rfloor_\times \tilde{\mathbf{R}}^{b_k}_w(\hat{\mathbf{f}}^w_i - \tilde{\mathbf{p}}^w_{b_k}) - \tilde{\mathbf{R}}^{b_k}_w\mathbf{v}^w_{b_k} + \mathbf{g}^w\Delta\hat{t}^o_k),$$

where $\lfloor\cdot\rfloor_\times$ denotes the skew symmetric matrix of a vector. Recall that the position $\tilde{\mathbf{p}}^w_{b_k}$ and rotation $\tilde{\mathbf{R}}^{b_k}_w$ in the Jacobian is computed at time instant $\tilde{t}_k$ instead of t_k, which depends on variable $\Delta\hat{t}^o_k$ and varies in each iteration of the optimization. Besides, the computation of the feature position $\hat{\mathbf{f}}^{c_k}_i$ projected on the k-th frame depends on $\tilde{\mathbf{p}}^w_{b_k}$ and $\tilde{\mathbf{R}}^{b_k}_w$ as shown in Eq. (4), which also needs to be recomputed when $\Delta\hat{t}^o_k$ changes. We in the next section present a novel algorithm based on dynamic programming and red-black tree to efficiently compute $\tilde{\mathbf{p}}^w_{b_k}$ and $\tilde{\mathbf{R}}^{b_k}_w$ as the estimated time offset $\Delta\hat{t}^o_k$ varies during the iterations.

4.4 Efficient IMU Integration over Variable-Length Time Intervals

With a naive implementation, the computation of $\hat{\boldsymbol{\alpha}}^{b_k}_{c_k}$ and $\hat{\mathbf{q}}^{b_k}_{c_k}$ in Eq. (7) between t_k and $\tilde{t}_k$ needs to be recomputed each time the offset Δt^o_k changes during the optimization. In order to reuse the intermediate integration results, we decompose the integration into two steps (Fig. 3): firstly, compute the integration between t_k and t_i; secondly, compute the integration between t_i and $\tilde{t}_k$. Here, without loss of generality, we assume t_i is the closest IMU read-out time instant before $\tilde{t}_k$. The decomposition makes the results in the first step reusable since the integration is computed over variable yet regular time intervals. We design

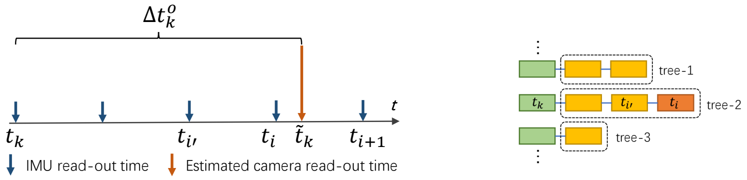


Fig. 3: Computing the pose at $\tilde{t}_k$ from the pose at t_k is decomposed into two steps (left): firstly, compute the pose at t_i, where t_i is the closest IMU measurement time instant to $\tilde{t}_k$ and smaller than $\tilde{t}_k$; secondly, compute the pose at $\tilde{t}_k$ based on the pose at t_i. We design an algorithm based on dynamic programming and red-black tree (right) for efficient indexing to accelerate the first step.

an algorithm based on discrete dynamic programming [16] and red-black tree [16] to perform efficient integration over the variable-length time intervals.

Specifically, we build a table (implemented as a red-black tree [16]) for each time instant t_k to store the intermediate results of IMU integration starting from t_k (right part of Fig. 3). Each node in the tree stores a key-value pair, where the key is an IMU read-out time t_i and the value is the integration from t_k to t_i computed using Eq. (7). Note that this integration is independent of the pose and velocity at t_k, so the stored results do not need to be updated when the pose and velocity in Eq. (6) as t_k change. During the optimization, each time when we need to compute the integration from t_k to t_i (recall that t_i is variable according to $\tilde{t}_k$), we first try to search in the tree at t_k to query the integration results for t_i. If the query failed, we then instead search if there exists a record for another IMU read-out time $t_{i'}$ such that $t_{i'} < t_i$. If multiple records exist, we select the maximum $t_{i'}$ (which is closest to t_i), and compute the integration from t_k to t_i based on the retrieved record at $t_{i'}$ and IMU measurements from $t_{i'}$ to t_i. During the process, each time a new integration is computed, the result is stored into the tree for future retrieval.

5 Uncertainty-Aware Initialization

We initialize our VIO system by first running a vision-only bundle adjustment on K consecutive frames. Suppose that the K consecutive rotations and positions obtained from vision-only bundle adjustment are $[\mathbf{p}_{c_0}^{c_0}\ \mathbf{R}_{c_0}^{c_0}\ \ldots\ \mathbf{p}_{c_{K-1}}^{c_0}\ \mathbf{R}_{c_{K-1}}^{c_0}]$. The goal of the initialization is to solve initial velocities $[\mathbf{v}_{b_k}^{b_0}\ \ldots\ \mathbf{v}_{b_{K-1}}^{b_0}]$, gravity vector $\mathbf{g}^{c_0}$, and metric scale s. The initializations in [31, 28] solve for the unknowns by minimizing the least squares of the L_2 errors between the poses obtained from vision-only bundle adjustment and the poses obtained from IMU integration. These methods do not take into account the uncertainties introduced by IMU noise during the IMU integration, which can cause failures of the initialization when the camera-IMU time offset is unknown (see Sect. 6.2).

We employ a different method to perform uncertainty-aware initialization by incorporating the covariances obtained during the IMU integration. The IMU propagation model for the K initialization images can be written as

$$\begin{aligned}\mathbf{p}^{c_0}_{b_{k+1}} &= \mathbf{p}^{c_0}_{b_k} + \mathbf{R}^{c_0}_{b_k}\mathbf{v}^{b_k}_{b_k}\Delta t_k - \frac{1}{2}\mathbf{g}^{c_0}\Delta t_k^2 + \mathbf{R}^{c_0}_{b_k}\hat{\boldsymbol{\alpha}}^k_{k+1}, \\ \mathbf{R}^{c_0}_{b_{k+1}}\mathbf{v}^{b_{k+1}}_{b_{k+1}} &= \mathbf{R}^{c_0}_{b_k}\mathbf{v}^{b_k}_{b_k} - \mathbf{g}^{c_0}\Delta t_k + \mathbf{R}^{c_0}_{b_k}\hat{\boldsymbol{\beta}}^k_{k+1},\end{aligned} \tag{8}$$

where $\mathbf{p}^{c_0}_{b_k} = s\mathbf{p}^{c_0}_{c_k} - \mathbf{R}^{c_0}_{b_k}\mathbf{p}^b_c$ and $\mathbf{R}^{c_0}_{b_k} = \mathbf{R}^{c_0}_{c_k}\mathbf{R}^c_b$. We set the camera-IMU extrinsics $\mathbf{p}^b_c$ and $\mathbf{R}^b_c$ to a zero vector and an identity matrix, respectively. Instead of minimizing the L_2 norm of the measurement errors, we minimize the Mahalanobis norm of the measurement errors as:

$$\min_{\mathcal{X}} \sum_{\hat{\mathbf{z}}^k_{k+1} \in S_i} \left\| \begin{bmatrix} \mathbf{R}^{b_k}_{c_0}(s\mathbf{p}^{c_0}_{c_{k+1}} - \mathbf{R}^{c_0}_{b_{k+1}}\mathbf{p}^b_c - s\mathbf{p}^{c_0}_{c_k} + \mathbf{R}^{c_0}_{b_k}\mathbf{p}^b_c + \frac{1}{2}\mathbf{g}^{c_0}\Delta t_k^2) - \mathbf{v}^{b_k}_{b_k}\Delta t_k - \hat{\boldsymbol{\alpha}}^k_{k+1} \\ \mathbf{R}^{b_k}_{c_0}(\mathbf{R}^{c_0}_{b_{k+1}}\mathbf{v}^{b_{k+1}}_{b_{k+1}} + \mathbf{g}^{c_0}\Delta t_k) - \mathbf{v}^{b_k}_{b_k} - \hat{\boldsymbol{\beta}}^k_{k+1} \end{bmatrix} \right\|^2_{\boldsymbol{\Sigma}^k_{k+1}},$$

where $\hat{\mathbf{z}}^k_{k+1} = \{\hat{\boldsymbol{\alpha}}^k_{k+1}, \hat{\boldsymbol{\beta}}^k_{k+1}\}$ and $\boldsymbol{\Sigma}^k_{k+1}$ is the covariance matrix computed during IMU integration [5]. Note that the covariance $\boldsymbol{\Sigma}^k_{k+1}$ in our formulation models the uncertainty of the IMU measurements. The resulting problem is a weighted least squares problem and can be solved efficiently. After solving the problem, we project the solved variables into the world coordinate. Other variables like accelerator biases, gyrocope biases, and time offsets are initialized as zeros.

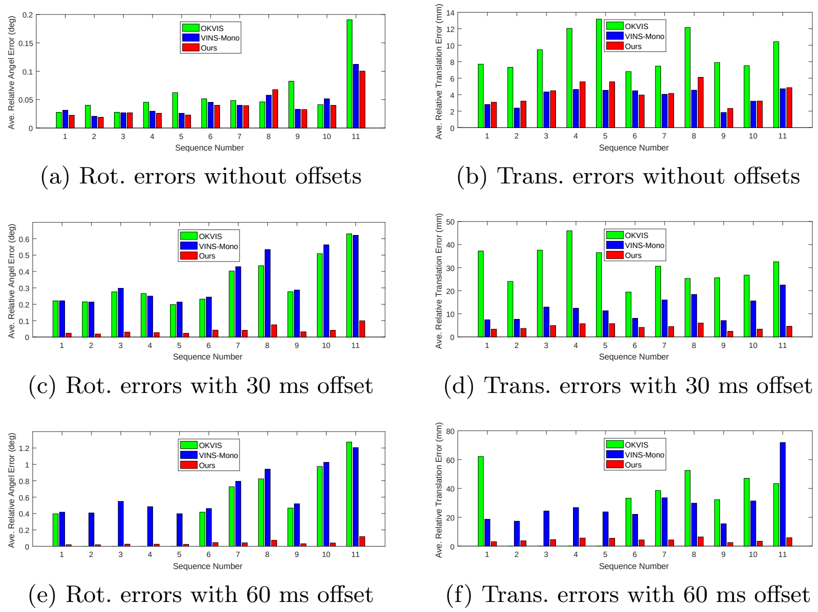
6 Experiments

We compare our approach to state-of-the-art monocular VIO systems: OKVIS [17] and VINS-Mono [30]. Loop closure in VINS-Mono is disabled for a fair comparison. The number of keyframes and non-keyframes in the optimization window of OKVIS and our approach are set to be 8 and 3 respectively. The sliding window size in VINS-Mono is set to be 11.

6.1 Performance on The Euroc Dataset

Sequences of the Euroc dataset consist of synchronized global-shutter stereo images (only mono/left images are used) and IMU measurements. The complexity of these sequences varies regarding trajectory length, flight dynamics, and illumination conditions. Ground truth poses are obtained by Vicon motion capture system. For presentation, we use numbers to denote sequence names: 1 ∼ 5 for MH_01_easy ∼ MH_05_difficult, 6 ∼ 8 for V1_01_easy ∼ V1_03_difficult, 9 ∼ 11 for V2_01_easy ∼ V2_03_difficult.

Significance of Online Temporal Camera-IMU Calibration Two error metrics are used for tracking accuracy evaluation: the average relative rotation error (deg) and the average relative translation error (m) [8]. Approaches are compared under different simulated time offsets between visual and inertial



(a) Rot. errors without offsets

(b) Trans. errors without offsets

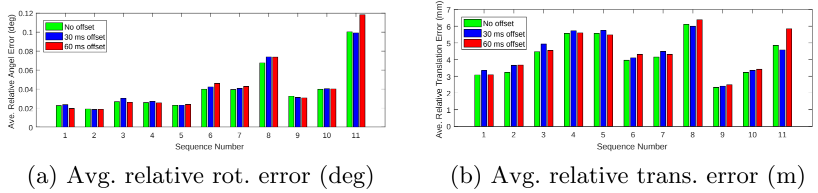
(c) Rot. errors with 30 ms offset

(d) Trans. errors with 30 ms offset

(e) Rot. errors with 60 ms offset

(f) Trans. errors with 60 ms offset

Fig. 4: The relative rot. and trans. errors with different time offsets on Euroc.



(a) Avg. relative rot. error (deg)

(b) Avg. relative trans. error (m)

Fig. 5: The average relative rot. and trans. errors of our approach with different time offsets on Euroc.

measurements: no time offsets; 30 ms offset; 60 ms offset. Fig. 4 shows the comparison results. When visual measurements and inertial measurements are synchronized, the average relative rotation/translation error between VINS-Mono and our approach are similar, while OKVIS performs the worst. As the time offset increases, VINS-Mono and OKVIS perform worse and worse due to the lack of camera-IMU time offset modeling. Our estimator achieves the smallest tracking error when there exists a time offset. OKVIS fails to track in sequence 2 $\sim$ 5 when the time offset is set to be 60 ms. We have tested different approaches under larger time offsets, such as 90 ms. However, neither OKVIS or VINS-Mono provides reasonable estimates. We thus omit comparisons for larger time offsets. We also illustrate the tracking performance of our approach under different time offsets in Fig. 5. We see that, by modeling the time offset properly, there is no significant performance decrease as time offset increases. Fig. 6 gives a visual

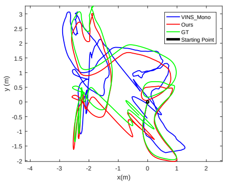


Fig. 6: Comparison on the Euroc V1_03_difficult with a 60 ms time offset.

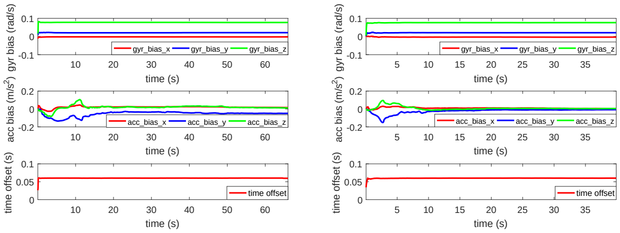


Fig. 7: The convergence study of variables in our estimator on the Euroc MH_03_medium (left) and V1_02_medium (right) with a 60 ms time offset.

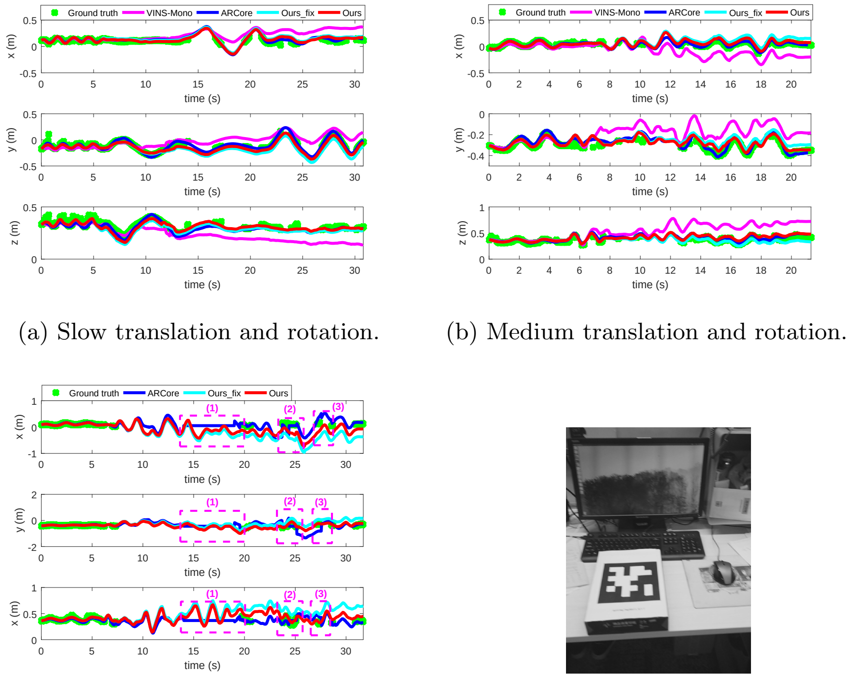
comparison on trajectories estimated by our approach and VINS-Mono on the Euroc sequence V1_03_difficult with a 60 ms time offset. Noticeable ‘jaggies’ are found on the trajectory estimated by VINS-Mono.

Parameter Convergence Study Another aspect we pay attention to is whether estimator parameters converge or not. We analyze our results on the sequence MH_03_medium and sequence V1_02_medium with a simulated 60 ms time offset. Estimated biases and time offsets w.r.t. time are plotted in Fig. 7. Both gyroscope biases and time offsets converge quickly. On the one hand, relative rotations are well constrained by visual measurements. They are not related to metric scale estimation. On the other hand, relative rotations are the first-order integration of angular velocities and gyroscope biases, thus they are easy to estimate. Another interesting thing we found is that, the convergence of time offsets is the same as the convergence of gyroscope biases. This means that, time offsets are calibrated mainly by aligning relative rotations from visual constraints and from gyroscope integrations, which is consistent with ideas of offline time offset calibration algorithms [12, 14]. Compared to gyroscope biases, acceleration biases converge much more slowly. They are hard to estimate as positions are the second-order integration of accelerations and acceleration biases. Additionally, acceleration measurements obtained from IMUs are coupled with gravity measurements. Abundant rotations collected across time are required to distinguish actual accelerations and gravity vector from measurements.

6.2 Performance on Mobile Phone Data

One of the applications of VIO is the motion tracking on mobile phones. We collect visual and inertial measurements using Samsung Galaxy S8 with a rolling-shutter camera and unsynchronized sensor measurements. Google developed ARCore [9] is also used for comparison. After recording the data, we run our system on the Samsung Galaxy S8. It takes 70 ± 10ms for each nonlinear optimization.

Qualitative Comparison of Tracking Accuracy We use AprilTag [37] to obtain time-synchronized and drift-free ground truths (Fig. 8 (d)). We detect tags



(a) Slow translation and rotation.

(b) Medium translation and rotation.

(c) Fast translation and rotation.

(d) Image

Fig. 8: The qualitative comparison of tracking accuracy. VINS-Mono fails to track on the fast translation and rotation sequence.

on captured images. If tags are detected, we calculate ground truth camera poses via P3P [7] using tag corners. Since VINS-Mono performs better than OKVIS, we only include tracking results from VINS-Mono for the following comparison. We test our approach with two conditions: 1)'Ours-fix': the variable time-delay estimation is switched off, i.e. is replaced with a single time offset, to account for the camera-IMU synchronization; 2)'Ours': the variable time-delay is enabled to account for both the camera-IMU synchronization and the rolling-shutter effect approximation. Three sequences are recorded around an office desk. These datasets are increasingly difficult to process in terms of motion dynamics: slow, medium, and fast translation and rotation (Fig. 8). To align coordinates of different approaches for comparison, we regularly move the mobile phone at the beginning of recorded sequences (see 0-5s in (a), 0-4s in (b), and 0-6s in (c)). VINS-Mono performs worst among all datasets, as it does not model time offsets between visual and inertial measurements. Our approach with condition 'Ours-fix' performs worse than the one with condition 'Ours'. The tracking accuracy of our approach is comparable to that of ARCore in small and medium motion settings. While for the fast translation and rotation sequence, where part of captured images are blurred, our approach exhibits better tracking robustness compared to ARCore. ARCore loses tracks in time periods within dashed boxes (1)(2)(3). This is because ARCore is based on EKF [11], which requires good initial guesses about predicted states. If pose displacements are large and feature correspondences are not abundant because of image blur, ARCore may fail. Conversely, our nonlinear optimization approach iteratively minimizes errors from sensor measurements, which treats the underlying non-linearity better and are less sensitive to initial guesses. Note that, there is a loop closing module in ARCore for pose recovery and drift correction. The final position drift of ARCore is thus smallest. However, loop closing is not the focus of this work.

Qualitative Comparison of Estimator Initialization To evaluate the significance of our initialization method, we compare it with the state-of-the-art visual-inertial initialization method [28]. We record 20 testing sequences on Samsung Galaxy S8 with medium translation and rotation. Each testing sequence lasts for about 30 seconds. We use the first 2-second sensor measurements to do the initialization. After the initialization is done, we use the visual-inertial estimator proposed in this work estimate camera poses. We then calculate an optimal scale factor by aligning the estimated trajectory with the trajectory reported by ARCore by a similarity transformation [2]. Although there is inevitably pose drift in ARCore estimates, this drift is not significant compared to the scale error caused by improper initializations. If the estimator fails to track in any time intervals of the testing sequences or the calculated scale error is larger than 5%, we declare the initialization as failures. For a fair comparison, we use the same relative poses obtained by visual sfm [29] as the initialization input of [28] and that of our approach. We find that 14 successful trials out of 20 (70%) using initialization proposed in [28], while 18 successful trials out of 20 (90%) using our initialization method. The successful initialization rate using our approach is higher than using initialization in [28]. We study the two testing sequences where our initialization fails. We find that time offsets between visual and inertial measurements in these two sequences are larger than 100 ms (this can be obtained by enumerating offsets and testing whether the following visual-inertial estimator outputs are close to that of ARCore or not after initialization), causing a big inconsistency when aligning the visual and inertial measurements. Since the main focus of this work is on the estimator, we are going to handle this failure case thoroughly in the future work.

7 Conclusions

In this work, we proposed a nonlinear optimization-based VIO that can deal with rolling-shutter effects and imperfect camera-IMU synchronization. We modeled the camera-IMU time offset as a time-varying variable to be estimated. An efficient algorithm for IMU integration over variable-length time intervals, which is required during the optimization, was also introduced. The VIO can be robustly launched with our uncertainty-aware initialization scheme. The experiments demonstrated the effectiveness of the proposed approach.

References

1. Baker, S., Matthews, I.: Lucas-Kanade 20 years on: A unifying framework. International Journal of Computer Vision **56**(3), 221–255 (2004)
2. Berthold K. P. Horn: Closed-form solution of absolute orientation using unit quaternions. Optical Society of America **4** (1987)
3. Dai, Y., Li, H., Kneip, L.: Rolling shutter camera relative pose: Generalized epipolar geometry. In: Proc. of the IEEE Intl. Conf. on Comput. Vis. and Pattern Recognition (2016)
4. Dong-Si, T., Mourikis, A.I.: Estimator initialization in vision-aided inertial navigation with unknown camera-imu calibration. In: Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst. (2012)
5. Forster, C., Carlone, L., Dellaert, F., Scaramuzza, D.: IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. In: Proc. of Robot.: Sci. and Syst. (2015)
6. Furgale, P., Rehder, J., Siegwart, R.: Unified temporal and spatial calibration for multi-sensor systems. In: Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst. (2013)
7. Gao, X.S., Hou, X.R., Tang, J., Cheng, H.F.: Complete solution classification for the perspective-three-point problem. IEEE Transactions on Pattern Analysis and Machine Intelligence (2003)
8. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
9. Google: ARCore: https://developers.google.com/ar/
10. Guo, C., Kottas, D., DuToit, R., Ahmed, A., Li, R., Roumeliotis, S.: Efficient visual-inertial navigation using a rolling-shutter camera with inaccurate timestamps. In: Proceedings of Robotics: Science and Systems (2014)
11. Hesch, J.A., Kottas, D.G., Bowman, S.L., Roumeliotis, S.I.: Consistency analysis and improvement of vision-aided inertial navigation. IEEE Trans. Robot. **30**(1), 158–176 (Feb 2014)
12. Jacovitti, G., Scarano, G.: Discrete time techniques for time delay estimation. IEEE Transactions on Signal Processing **41** (1993)
13. Jung, S.H., Taylor, C.J.: Camera trajectory estimation using inertial sensor measurements and structure from motion results. In: Proc. of the IEEE Intl. Conf. on Comput. Vis. and Pattern Recognition (2001)
14. Kelly, J., Sukhatme, G.: A general framework for temporal calibration of multiple proprioceptive and exteroceptive sensors. In: Proc. of the Intl. Sym. on Exp. Robot. (2010)
15. Klein G., Murray D.: Parallel tracking and mapping on a camera phone. In: Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (2009)
16. Knuth, D.: The Art of Computer Programming, vol. 1-3. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1998)
17. Leutenegger, S., Furgale, P., Rabaud, V., Chli, M., Konolige, K., Siegwart, R.: Keyframe-based visual-inertial SLAM using nonlinear optimization. In: Proc. of Robot.: Sci. and Syst. (2013)
18. Li, M., Mourikis, A.I.: 3D motion estimation and online temporal calibration for camera- IMU systems. In: Proc. of the IEEE Intl. Conf. on Robot. and Autom. (2013)

19. Li, M., Mourikis, A.I.: High-precision, consistent EKF-based visual-inertial odometry. Intl. J. Robot. Research **32**(6), 690–711 (May 2013)
20. Li, M., Mourikis, A.I.: Real-time motion tracking on a cellphone using inertial sensing and a rolling shutter camera. In: Proc. of the IEEE Intl. Conf. on Robot. and Autom. (2013)
21. Li, M., Mourikis, A.I.: Online temporal calibration for camera-imu systems: Theory and algorithms. Intl. J. Robot. Research **33** (2014)
22. Li, M., Mourikis, A.I.: Vision-aided Inertial Navigation with Rolling-Shutter Cameras. Intl. J. Robot. Research **33** (2014)
23. Ling, Y., Kuse, M., Shen, S.: Edge alignment-based visual-inertial fusion for tracking of aggressive motions. Autonomous Robots (2017)
24. Ling, Y., Shen, S.: Dense visual-inertial odometry for tracking of aggressive motions. In: Proc. of the IEEE Intl. Conf. on Robot. and Bio. (2015)
25. Ling, Y., Shen, S.: Aggressive quadrotor flight using dense visual-inertial fusion. In: Proc. of the IEEE Intl. Conf. on Robot. and Autom. (2016)
26. Lynen, S., Achtelik, M., Weiss, S., Chli, M., Siegwart, R.: A robust and modular multi-sensor fusion approach applied to MAV navigation. In: Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst. (2013)
27. Mourikis, A.I., Roumeliotis, S.I.: A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation. In: Proc. of the IEEE Intl. Conf. on Robot. and Autom. (2007)
28. Mur-Artal, R., D.Tardós, J.: Visual-inertial monocular slam with map reuse. IEEE Robotics and Automation Letters **2** (2017)
29. Mur-Artal, R., Montiel, J.M.M., Tardós, J.D.: ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Transactions on Robotics **31**(5), 1147–1163 (2015). https://doi.org/10.1109/TRO.2015.2463671
30. Qin, T., Li, P., Shen, S.: VINS-Mono: A robust and versatile monocular visual-inertial state estimator. arXiv preprint arXiv:1708.03852 (2017)
31. Qin, T., Shen, S.: Robust initialization of monocular visual-inertial estimation on aerial robots. In: Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst. (2017)
32. Shen, S., Michael, N., Kumar, V.: Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft MAVs. In: Proc. of the IEEE Intl. Conf. on Robot. and Autom. Seattle, WA (May 2015)
33. Shen, S., Mulgaonkar, Y., Michael, N., Kumar, V.: Vision-based state estimation for autonomous rotorcraft MAVs in complex environments. In: Proc. of the IEEE Intl. Conf. on Robot. and Autom. (2013)
34. Shi, J., Tomasi, C.: Good features to track. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (1994)
35. S.I. Roumeliotis, A.E. Johnson, J.F. Montgomery: Augmenting inertial navigation with image-based motion estimation. In: Proc. of the IEEE Intl. Conf. on Robot. and Autom. (2002)
36. Steven, L., Alonso, P.P., Gabe, S.: Spline fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras. In: British Machine Vision Conference (2013)
37. Wang, J., Olson, E.: AprilTag 2: Efficient and robust fiducial detection. In: Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst. (2016)
38. Weiss, S., Achtelik, M.W., Lynen, S., Chi, M., Siegwart, R.: Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments. In: Proc. of the IEEE Intl. Conf. on Robot. and Autom. (2012)

39. Yang, Z., Shen, S.: Monocular visual-inertial fusion with online initialization and camera-IMU calibration. In: Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst. (2015)
40. Yang, Z., Shen, S.: Tightly-coupled visual-inertial sensor fusion based on IMU pre-integration. Tech. rep., Hong Kong University of Science and Technology (2016), URL: http://www.ece.ust.hk/ eeshaojie/vins2016zhenfei.pdf
41. Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.: Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images. In: European Conference on Computer Vision (2018)