2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Madrid, Spain, October 1-5, 2018

Online Temporal Calibration for Monocular Visual-Inertial Systems

Tong Qin and Shaojie Shen

***Abstract*— Accurate state estimation is a fundamental module for various intelligent applications, such as robot navigation, autonomous driving, virtual and augmented reality. Visual and inertial fusion is a popular technology for 6-DOF state estimation in recent years. Time instants at which different sensors' measurements are recorded are of crucial importance to the system's robustness and accuracy. In practice, timestamps of each sensor typically suffer from triggering and transmission delays, leading to temporal misalignment (time offsets) among different sensors. Such temporal offset dramatically influences the performance of sensor fusion. To this end, we propose an online approach for calibrating temporal offset between visual and inertial measurements. Our approach achieves temporal offset calibration by jointly optimizing time offset, camera and IMU states, as well as feature locations in a SLAM system. Furthermore, the approach is a general model, which can be easily employed in several feature-based optimization frameworks. Simulation and experimental results demonstrate the high accuracy of our calibration approach even compared with other state-of-art offline tools. The VIO comparison against other methods proves that the online temporal calibration significantly benefits visual-inertial systems. The source code of temporal calibration is integrated into our public project, VINS-Mono[1].**

I. Introduction

State estimation has been a fundamental research topic in robotics and computer vision communities over the last decades. Various applications, such as robot navigation, autonomous driving, virtual reality (VR) and augmented reality (AR), highly rely on accurate state estimation. We are particularly interested in state estimation solutions that involve only one camera, due to its small size, low power consumption, and simple mechanical configuration. There have been excellent results in monocular visual-only techniques [1]–[7], which computed accurate camera motion and up-to-scale environmental structure. To solve the well-known scale ambiguity, multi-sensor fusion approaches attract more and more attention. Many researches [8]–[17] assisted camera with IMU (Inertial Measurement Unit), which achieved impressive performance in 6-DOF SLAM (simultaneous localization and mapping). On the one hand, inertial measurements render pitch and roll angle, as well as scale, observable. On the other hand, inertial measurements improve motion tracking performance by bridging the gap when visual tracking fails.

To fuse data from different sensors, time instants at which measurements are recorded must be precisely known. In practice, the timestamps of each sensor typically suffer from triggering and transmission delays, leading to a temporal misalignment (time offset) between different sensor streams. Consequently, the time synchronization of sensors may cause a crucial issue to a multi-sensor system. For the visual-inertial system, the time offset between the camera and IMU dramatically affects robustness and accuracy. Most visual-inertial methods [13, 14, 16, 17] assumed measurements' timestamps are precise under a single clock. Therefore, these methods work well with a few strictly hardware-synchronized sensors. For most low-cost and self-assembled sensor sets, hardware synchronization is not available. Due to triggering and transmission delays, there always exists a temporal misalignment (time offset) between camera and IMU. The time offset usually ranges from several milliseconds to hundreds of milliseconds. Dozens of milliseconds will lead to IMU sequences totally misaligning with image stream, thus dramatically influencing the performance of a visual-inertial system.

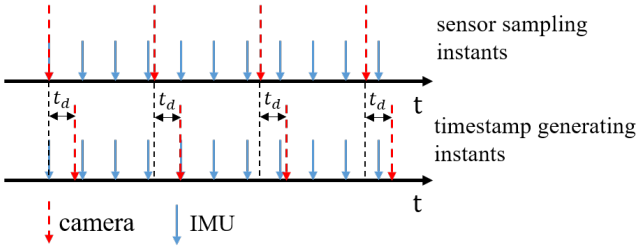

Fig. 1. An illustration of temporal misalignment (time offset) between camera and IMU streams. The upper plot represents sampling instants. The lower plot shows timestamping instants. The generated timestamp is not equal to the actual sampling time due to triggering delay, transmission delay, and unsynchronized clocks, leading to a temporal misalignment between camera and IMU. The time offset t_d is the amount of time by which we should shift the camera timestamps so that the camera and IMU data streams became temporally consistent.

To this end, we propose a method to online calibrate temporal offset for a visual-inertial system. We assume time offset is a constant but unknown variable. We calibrate it by estimating it online along with camera and IMU states, as well as feature locations in a SLAM system. Our calibration approach is a general factor, which can be easily employed in other feature-based visual-inertial optimization frameworks. Although we use the monocular sensor suite to showcase our method, the proposed approach can be easily applied to multi-camera visual-inertial systems. We highlight our contribution as follows:

- We propose an online approach to calibrate temporal

All authors are with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, China. tqinab@connect.ust.hk, eeshaojie@ust.hk. This work was supported by the Hong Kong Research Grants Council Early Career Scheme under project no. 26201616.

[1]https://github.com/HKUST-Aerial-Robotics/VINS-Mono

offset between camera and IMU in the visual-inertial system.
- We showcase the significance of online temporal calibration through both simulation and real-world experiments.
- Open-source code integrated into the public project.

The rest of the paper is structured as follows. In Sect. II, we discuss the relevant literature. The algorithm is introduced in detail in Sect. III. Implementation details and experimental evaluations are presented in Sect. IV. Finally, the paper is concluded in Sect. V.

II. Related Work

Over the past few decades, there have been tremendous researches in visual-inertial odometry techniques, which aimed to compute camera motion and environment structure with high accuracy. The popular techniques are either filter-based framework [9]–[12, 17], or batch optimization [13]–[16, 18]. Most of visual-inertial algorithms process image by extracting robust sparse features instead of operating on the dense image. Among these works, [9, 10, 18] used structure-less vision factor, which eliminated features by projecting visual residual onto null space. They focus more on estimating camera or IMU motion instead of feature positions. [13, 14, 16] selectively kept keyframes and features in a bundle, which optimized camera motion and feature together. All of these methods assumed IMU and camera are precisely synchronized without temporal misalignment.

The temporal misalignment between IMU and camera is a typical issue in low-cost and self-assembled devices. The measurement's timestamp is misaligned with actual sampling time instant due to unsynchronized clocks, triggering delay and transmission delay. This time offset is unknown and needs to be calibrated. Several pieces of research have focused on calibrating it. Mair [19] proposed an initialization approach for temporal and spatial calibration, which used either cross-correlation or phase congruency. This approach formulated calibration procedure in a novel and special view. It separated calibrated variables from other unknown variables (poses, feature positions). Therefore, it can provide a good prior without influence from other variables. Further on, methods modeled time offset in a more precise formulation. Kelly [20] aligned rotation curves of camera and IMU to calibrate time offset. It leveraged a variant of ICP (iterative closest point) method to gradually match two rotation curves. Kalibr, which came from Furgale [21], estimated time offset, camera motion, as well as extrinsic parameters between camera and IMU in the continuous batch optimization procedure. Kalibr achieved impressive performance and became a popular toolbox. However, these two methods operated offline with a fixed planar pattern (such as a chessboard). The calibration pattern provided them with robust feature tracking and association, as well as accurate 3D position. Moreover, Li proposed a motion estimation method with online temporal calibration for the camera-IMU system in [22]. The time offset was calibrated in a multi-state constrained EKF framework. His method had a significant advantage in computation complexity, which can be used on portable mobile devices. Compared with his method, our optimization-based algorithm outperforms in term of accuracy, since we can iteratively optimize a lot of variables in a big bundle instead of fixing linearization error early.

III. Algorithm

In this section, we model temporal offset in a vision factor, and online calculate it along with features, IMU and camera states in an optimization-based VIO framework.

We briefly denote frame and notation as follows. $(\cdot)^w$ denotes global frame. $(\cdot)^c$ denotes local camera frame. ($\mathbf{R}_c^w$, $\mathbf{p}_c^w$) is camera pose in the global frame, which can transform 3D feature from camera frame to global frame.

A. Temporal Offset

For low-cost and self-assembled visual-inertial sensor sets, camera and IMU are put together without strict time synchronization. The generated timestamp is not equal to the time instant at which the measurement is sampled due to triggering delay, transmission delay and unsynchronized clocks. Hence, there usually exists temporal offset between different measurements. In general cases, the time offset between sensors is a constant but unknown value. In some worse cases, sensors are collected with different clocks and the time offset drifts along with the time. This kind of sensors is unqualified for sensor fusion.

In this paper, we consider the general case, where time offset t_d is a constant but unknown value. One picture illustrating time offset is depicted in Fig. 1. In the picture, the upper plot represents sampling instants. The lower plot shows timestamping instants. The generated timestamp is not equal to the actual sampling time due to triggering delay, transmission delay and unsynchronized clocks, leading to a temporal misalignment between camera and IMU. Specifically, we define t_d as,

$$t_{IMU} = t_{cam} + t_d. \tag{1}$$

The time offset t_d is the amount of time by which we should shift the camera timestamps, so that the camera and IMU data streams became temporally consistent. t_d may be a positive or negative value. If the camera sequence has a longer latency than the IMU sequence, t_d is a negative value. Otherwise, t_d is a positive value.

B. Feature Velocity on Image Plane

To make camera and IMU data streams temporally consistent, the camera sequence should be shifted forward or backward according to t_d. Instead of shifting whole camera or IMU sequence, we specifically shift features' observations in the timeline. To this end, we introduce feature velocity for modeling and compensating the temporal misalignment.

In a very short time period (several milliseconds), the camera's movement can be treated as constant speed motion. Hence, a feature moves at an approximately constant velocity on the image plane in short time period. Based on this

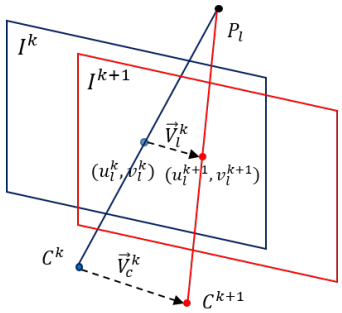

Fig. 2. An illustration of feature's velocity on image plane. I^k and I^{k+1} are two consecutive image frames. $[u_l^k, v_l^k]$ and $[u_l^{k+1}, v_l^{k+1}]$ are feature's 2D observations on the image planes I^k and I^{k+1} respectively. Camera is assumed to move at a constant speed from C^k to C^{k+1} in short time period $[t_k, t_{k+1}]$. Hence, we approximately think that feature l also moves at a constant speed $\mathbf{V}_l^k$ on the image plane in short time period.

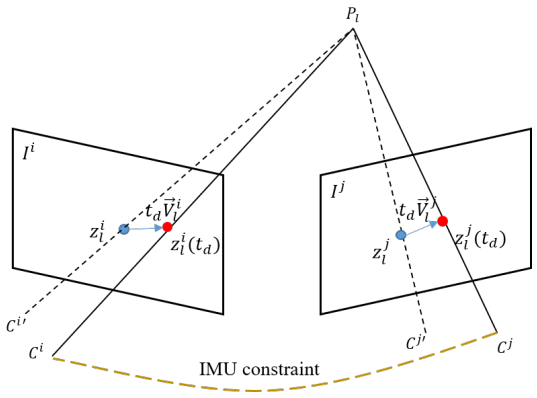

Fig. 3. An illustration of reprojection process. The dashed line presents traditional reprojection procedure without time offset modeling. The solid line presents proposed reprojection which takes time offset into consideration. The yellow line presents IMU constraint. The IMU constraint is inconsistent with traditional reprojection constraint. By optimizing t_d, we can find the optimal camera pose and feature's observation in time domain which matches IMU constraint.

assumption, we compute the feature's velocity on the image plane.

As depicted in Fig. 2, I^k and I^{k+1} are two consecutive image frames. The camera is assumed to move at a constant speed from C^k to C^{k+1} in the short time period $[t_k, t_{k+1}]$. Hence, we approximately think that feature l also moves at a constant speed $\mathbf{V}_l^k$ on the image plane in this short time period. The velocity $\mathbf{V}_l^k$ is calculated as follows:

$$\mathbf{V}_l^k = (\begin{bmatrix} u_l^{k+1} \\ v_l^{k+1} \end{bmatrix} - \begin{bmatrix} u_l^k \\ v_l^k \end{bmatrix})/(t_{k+1} - t_k) \tag{2}$$

where $[u_l^k, v_l^k]$ and $[u_l^{k+1}, v_l^{k+1}]$ are feature's 2D observations on the image planes I^k and I^{k+1} respectively.

C. Vision Factor with Time Offset

In classical sparse visual SLAM algorithms, visual measurements are formulated as (re)projection error in cost function. We refector the classical (re)projection error by adding a new variable, time offset. There are two typical parameterizations of a feature. Some algorithms parameterize feature as its 3D position in the global frame, while other algorithms parameterize feature as depth or inverse depth with respect to a certain image frame. In the following, we respectively model time offset into vision factors with these two kinds of parameterizations.

1) 3D Position Parameterization: The feature is parameterized as 3D position ($\mathbf{P}_l = [x_l, y_l, z_l]^T$) in the global frame. In traditional, the visual measurement is formulated as the projection error,

$$\begin{aligned} \mathbf{e}_l^k &= \mathbf{z}_l^k - \pi(\mathbf{R}_{c_k}^{w^T}(\mathbf{P}_l - \mathbf{p}_{c_k}^w)) \\ \mathbf{z}_l^k &= [u_l^k \ v_l^k]^T. \end{aligned} \tag{3}$$

$\mathbf{z}_l^k$ is the observation of feature l in frame k. $(\mathbf{R}_{c_k}^w, \mathbf{p}_{c_k}^w)$ is the camera pose, which transform feature $\mathbf{P}_l$ from global frame to local camera frame. $\pi(\cdot)$ denotes the camera projection model, which projects 3D feature into image plane with distortion.

The camera pose $(\mathbf{R}_{c_k}^w, \mathbf{p}_{c_k}^w)$ is constrained by visual measurements in the above-mentioned formulation. It is also constrained by IMU measurements. In practice, if there exists time misalignment between IMU and camera, the IMU constraint is inconsistent with vision constraint in the time domain. In other words, we should shift camera sequence forward or backward, so that the camera and IMU data streams become temporally consistent. Instead of shifting whole camera or IMU sequence, we specifically shift feature' observations in the timeline. The new formulation is written as follows,

$$\begin{aligned} \mathbf{e}_l^k &= \mathbf{z}_l^k(t_d) - \pi(\mathbf{R}_{c_k}^{w^T}(\mathbf{P}_l - \mathbf{p}_{c_k}^w)) \\ \mathbf{z}_l^k(t_d) &= [u_l^k \ v_l^k]^T + t_d \mathbf{V}_l^k. \end{aligned} \tag{4}$$

$\mathbf{V}_l^k$ is feature's speed on the image plane, got from eq. 2. t_d is the unknown variable of time offset, which shifts feature's observation in time domain. By optimizing t_d, we can find the optimal camera pose and feature's observation in the time domain which matches IMU constraints.

2) Depth Parameterization: The feature can be also parameterized as depth or inverse depth with respect to an image frame. We take depth λ_i in image i as the example. The traditional reprojection error from image i to image j is written as,

$$\begin{aligned} \mathbf{e}_l^j &= \mathbf{z}_l^j - \pi(\mathbf{R}_{c_j}^{w^T}(\mathbf{R}_{c_i}^w \lambda_i \begin{bmatrix} \mathbf{z}_l^i \\ 1 \end{bmatrix} + \mathbf{p}_{c_i}^w - \mathbf{p}_{c_j}^w)) \\ \mathbf{z}_l^i &= [u_l^i \ v_l^i]^T, \quad \mathbf{z}_l^j = [u_l^j \ v_l^j]^T. \end{aligned} \tag{5}$$

The feature l is first projected into global frame, then back projected onto the image plane in local camera frame j. The residual is the displacement between observation and back projection location.

Similarly with eq. 4, we take the time offset variable t_d

into account,

$$\begin{aligned}\mathbf{e}_l^j &= \mathbf{z}_l^j(t_d) - \pi(\mathbf{R}_{c_i}^{w^T}(\mathbf{R}_{c_j}^w \lambda_i \begin{bmatrix}\mathbf{z}_l^i(t_d)\\ 1\end{bmatrix} + \mathbf{p}_{c_i}^w - \mathbf{p}_{c_j}^w)) \\ \mathbf{z}_l^i &= [u_l^i \ v_l^i]^T + t_d \mathbf{V}_l^i, \quad \mathbf{z}_l^j = [u_l^j \ v_l^j]^T + t_d \mathbf{V}_l^j.\end{aligned} \tag{6}$$

Fig. 3 depicts the reprojection process. The dashed line represents traditional reprojection procedure without time offset modeling. The solid line represents proposed reprojection which takes time offset into consideration. The yellow line denotes IMU constraint. The IMU constraint is inconsistent with traditional reprojection constraint. By optimizing t_d, we can find the optimal camera pose and feature's observation in the time domain which matches IMU constraints.

D. Optimization with Time Offset

By leveraging the above-mentioned vision factor, we can easily add the temporal calibration function into typical visual-inertial optimization-based frameworks, such as [13, 16, 23]. In these frameworks, Visual-inertial localization and mapping is formulated as a nonlinear optimization problem that tightly couples visual and inertial measurements. As depicted in Fig. 4, several camera frames and IMU measurements are kept in a bundle. The bundle size usually is limited to bound computational complexity. A local bundle adjustment (BA) jointly optimizes camera and IMU states, as well as feature locations.

We can easily add the proposed visual factor (III-C) into this kind of framework. To be specific, the whole state variables are augmented with time offset, which are defined as:

$$\begin{aligned}\mathcal{X} &= [\mathbf{x}_0, \mathbf{x}_1, \cdots \mathbf{x}_n, \mathbf{P}_0, \mathbf{P}_1, \cdots \mathbf{P}_l, t_d] \\ \mathbf{x}_k &= [\mathbf{p}_k^w, \mathbf{v}_k^w, \mathbf{R}_k^w, \mathbf{b}_a, \mathbf{b}_g], k \in [0, n].\end{aligned} \tag{7}$$

where the k-th IMU state consists of the position $\mathbf{p}_k^w$, velocity $\mathbf{v}_k^w$, orientation $\mathbf{R}_k^w$ in the global frame, and IMU bias $\mathbf{b}_a$, $\mathbf{b}_g$ in the local body frame. The feature $\mathbf{P}_l$ is parameterized by either 3D position in the global frame or depth with respect to a certain image frame.

The whole problem is formulated as one cost function containing IMU propagation factor, reprojection factor, as well as a certain prior factor. Hereby, we use the proposed vision (III-C) factor to achieve time offset calibration,

$$\begin{aligned}\min_{\mathcal{X}} \Bigg\{ &\underbrace{\|\mathbf{e}_p - \mathbf{H}_p \mathcal{X}\|^2}_{\text{prior factor}} + \underbrace{\sum_{k \in \mathcal{B}} \left\|\mathbf{e}_{\mathcal{B}}(\mathbf{z}_{k+1}^k, \mathcal{X})\right\|_{\mathbf{P}_{k+1}^k}^2}_{\text{IMU propagation factor}} \\ &+ \underbrace{\sum_{(l,j) \in \mathcal{C}} \left\|\mathbf{e}_{\mathcal{C}}(\mathbf{z}_l^j, \mathcal{X})\right\|_{\mathbf{P}_l^j}^2}_{\text{proposed vision factor}} \Bigg\}.\end{aligned} \tag{8}$$

$\mathbf{e}_{\mathcal{B}}(\mathbf{z}_{k+1}^k, \mathcal{X})$ is the error term from IMU propagation. $\mathcal{B}$ is the set of all IMU measurements. $\mathbf{e}_{\mathcal{C}}(\mathbf{z}_l^j, \mathcal{X})$ is the proposed visual (re)projection error, which includes the time offset variable. $\mathcal{C}$ is the set of features which have been observed at least twice in the image frames. The errors are weighted by their inverse covariance $\mathbf{P}$. $\{\mathbf{e}_p, \mathbf{H}_p\}$ is the prior information from prior knowledge and marginalization. Only a small amount of measurements and states are kept in the optimization bundle, while others are marginalized out and converted into prior. The non-linear least squares cost function can be efficiently optimized using Gauss-Newton methods.

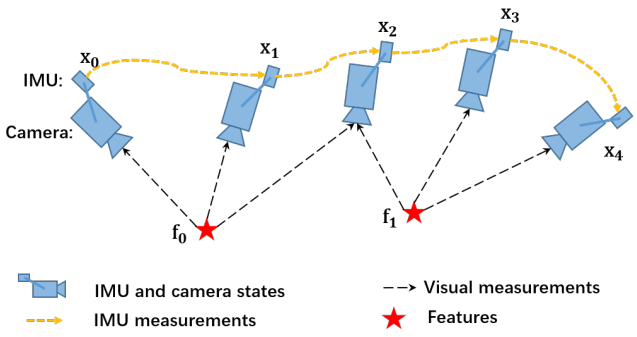

Fig. 4. An illustration of visual-inertial localization and mapping problem. We maintain several camera frames and IMU measurements in a bundle. The bundle size usually is limited to reduce computation complexity. A local bundle adjustment (BA) jointly optimizing camera and IMU states, as well as feature locations.

E. Compensation of Time Offset

After each optimization, we compensate time offset by shifting timestamps of subsequent visual streams, as $t'_{cam} = t_{cam} + t_d$. Then the system estimates δt_d between compensated visual measurement and inertial measurement in the following. δt_d will be iteratively optimized in subsequent data streams, which will converge to zero. With the decrease of time interval δt_d, our basic assumption (feature moves at a constant speed on the image plane in a short time interval) is more and more reasonable. Even if there is a huge time offset (e.g. hundreds of milliseconds) at the beginning, the process will compensate it from coarse to fine gradually.

IV. Experiment Results

In this section, we first demonstrate the accuracy of temporal calibration, then we show the overall VIO performance improved by temporal calibration. The calibration experiments are presented with simulated data and real sensor set. The overall VIO performance is shown with public dataset and real-world experiment. In each part, we compare the proposed algorithm against other popular methods.

A. Implement Details

We adopt the visual-inertial optimization framework proposed in [23]. We only add the time offset into the state vector and use the proposed vision factor (Sect. III-C). Features are detected by Shi-Tomasi Corner Detector [24] and tracked by KLT tracker [25], while IMU measurements are locally integrated. Poses, velocities, IMU bias of several keyframes,

TABLE I

SIMULATION CALIBRATION RESULTS

Sequence[ms]	Mean[ms]	RMSE[ms]	NEES
I. 5	5.12	0.36	7.2%
II. 15	15.06	0.61	4.1%
III. 30	30.17	0.68	2.3%

The calibration results of simulated data with 5ms, 15ms and 30ms time offset. RMSE is the root mean square error. NEES is normalized estimation error squared.

Fig. 5. Intel Realsense camera ZR300, which contains a fisheye global shutter camera with $100^\circ \times 133^\circ$ FOV and IMU (Gyro & Acc).

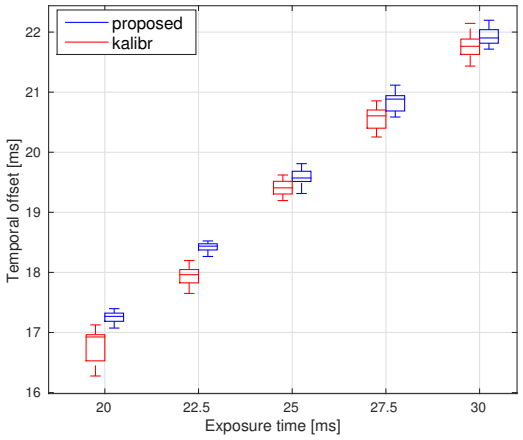

Fig. 6. The estimated time offset from the proposed method and Kalibr with respect to different exposure times.

as well as feature position, are optimized in a local bundle adjustment. Only keyframes, which contain sufficient feature parallax with their neighbors, are temporarily kept in the local window. Previous keyframes are marginalized out of the window in order to bound computation complexity. Ceres Solver [26] is used for solving this nonlinear problem. The whole system runs in real-time with Intel i7-3770 CPU.

B. Temporal Calibration Results

1) Simulation: We randomly generate 500 feature points in the 60m x 60m x 60m space. Features locations are unknown. Visible features are projected to the virtual camera subjected to zero-mean Gaussian noise with a standard deviation of 0.5 pixels. Inertial measurements are also subjected to zero-mean Gaussian noise with standard deviation of $0.01m/s^2$ and $0.001rad/s$ in accelerometer and gyroscope respectively without bias. The inertial measurement rate is 100 Hz and the camera frame rate is 10 Hz. The camera and IMU move together with sufficient accelerating and rotating motion. The whole trajectory lasts 30 seconds. We set time offsets as 5ms, 15ms, and 30ms. For each time offset, 100 simulated trials are conducted. The calibration results are shown in Table. I. Our algorithm can successfully calibrate time offset in simulated data with low RMSE (Root Mean Square Error).

2) Real Sensor: We used Intel Realsense camera ZR300[2], which contains a fisheye global shutter camera with $100^\circ \times 133^\circ$ FOV and IMU (Gyro & Acc), as shown in Fig. 5. The manufacturer claimed that sensors are well synchronized. In practice, we find out there is an apparent time offset between the fisheye camera and IMU got from default SDK, which is affected by exposure time. Since ground truth is unavailable, we take the state-of-art temporal calibration toolbox, Kalibr [21] for comparison. Kalibr calibrated time offset in an offline batch optimization framework, which needs addition calibration pattern (chessboard).

We set five exposure times from 20ms to 30ms. For each exposure time, we collected fifteen datasets by moving the sensor in front of a calibration chessboard for 40 seconds. Actually, Kalibr relies on a calibration pattern, while our algorithm does not. To make the calibration fully observable, we ensured sufficient rotational and accelerated motion over all datasets.

The results of temporal calibration are depicted in Fig. 6. We can see that the temporal offset evolves linearly with the exposure time with a slope around 0.5. That is because the middle of the exposure time is treated as the optimal point to timestamp an image. Therefore, the temporal offset consists of fixed communication and triggering delays plus half exposure time. Both the proposed method and Kalibr satisfy this situation. Since ground truth is unavailable, we take Kalibr's results as reference. Our results are quite close to Kalibr's results. As for standard derivation, the proposed method achieved [0.095, 0.071, 0.14, 0.16, 0.15], which is smaller than Kalibr's standard derivation [0.27, 0.16, 0.13, 0.18, 0.20]. The proposed method outperform Kalibr in terms of consistency. Note that Kalibr is an offline batch optimization, which consumes dozens of times more than the proposed method. Furthermore, Kalibr relies on calibration pattern. Hence, the proposed method also outperforms Kalibr in terms of efficiency and practicability.

C. Overall VIO Performance

1) Dataset: We evaluate the proposed method using EuRoC MAV Visual-Inertial Datasets [27]. Datasets are collected onboard a micro aerial vehicle, which contains stereo images (Aptina MT9V034 global shutter, WVGA monochrome, 20 FPS), synchronized IMU measurements (ADIS16448, 200 Hz), and ground truth states (VICON and Leica MS50). We only use images from the left camera. It is

[2]https://software.intel.com/en-us/realsense/zr300

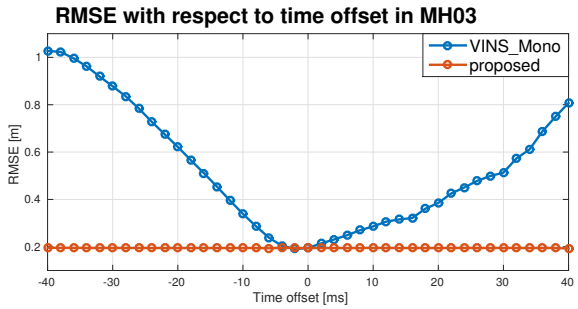

Fig. 7. RMSE with respect to the time offset in MH03 sequence. The x-axis shows the predefined time offset, and the y-axis shows the RMSE (Root Mean Square Error) [28]. The blue line represents results of VINS-Mono [23], which is the base framework proposed method build on. The red line represents results of proposed method, which has capability of time offset calibration.

well known that images and IMU measurements are strictly synchronized in this dataset. To demonstrate the capability of temporal calibration, we set the time offset by manually shifting IMU timestamps. To be specific, we add a fixed millisecond value to IMU timestamps, such that there is a fixed time offset between IMU and camera. We made time-shifted sequences and used them to test the proposed algorithm and other methods.

At first, we studied the influence of time offset on visual-inertial odometry. We set time offsets from -40 to 40ms, and test these time-biased sequences with VINS-Mono [23] and the proposed method respectively. VINS-Mono is the base framework which we build our system on. VINS-Mono does not have time offset calibration capability, thus it significantly suffers from temporal misalignment. The result is depicted in Fig. 7. The x-axis shows the predefined time offset, and the y-axis shows RMSE (Root Mean Square Error), as proposed in [28]. The test data is the MH03 sequence, whose IMU timestamp is shifted. The blue line represents results of VINS-Mono. We can see that the RMSE evolves along a parabolic curve with respect to time offsets for VINS-Mono. The performance deteriorates dramatically when the time offset increases. The tolerance interval is only within 6 milliseconds. That demonstrates that it is necessary to do time offset calibration. The red line represents results of the proposed method, which calibrates the time offset. It can be seen that the RMSEs are same under different time offsets, which proves that our calibration procedure is very effective.

In the following, we compare against OKVIS [16], which is another state-of-art visual-inertial odometry algorithm without temporal calibration ability. We use time-biased sequences to test proposed method and OKVIS. The results are shown in TABLE. II. The trajectory is also evaluated by RMSE. For OKVIS, with the increase of time offset, the performance degrades (RMSE become larger and larger). In some sequences (i.e. MH_03, V1_03), RMSE dramatically increases when the time offset is up to 30ms. Such time offset makes the system diverge. For proposed method, however, the performance is not affected by the time offset. The RMSEs are almost same in one sequence under different time

TABLE II

RMSE IN EUROC DATASET.

Sequences	t_d[ms]	Proposed		OKVIS
		Esti. t_d [ms]	RMSE[m]	RMSE[m]
MH_01	5	4.87	0.155	0.318
	15	14.87	0.158	0.382
	30	29.87	0.156	0.544
MH_03	5	4.99	0.194	0.284
	15	15.02	0.194	0.451
	30	29.99	0.195	2.805
MH_05	5	5.10	0.303	0.432
	15	15.30	0.326	0.577
	30	30.08	0.303	0.652
V1_01	5	5.16	0.088	0.100
	15	15.16	0.088	0.202
	30	30.21	0.089	0.257
V1_03	5	4.87	0.185	0.349
	15	14.88	0.187	1.008
	30	29.90	0.189	1.817
V2_02	5	4.92	0.159	0.378
	15	14.93	0.161	0.520
	30	29.92	0.162	1.010

RMSE is root mean square error, as proposed in [28].

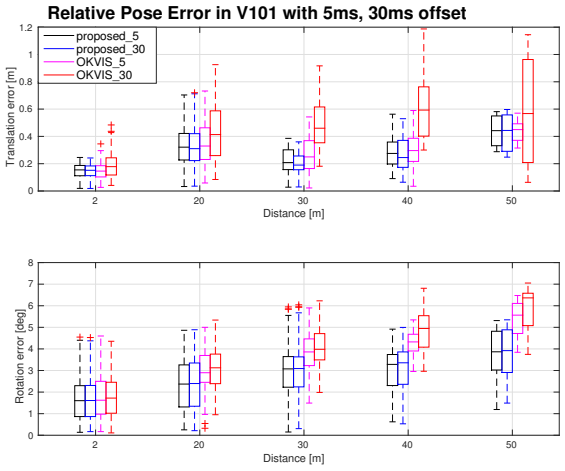

Fig. 8. Relative pose error [29] comparison between proposed method and OKVIS in V101 sequence with 5ms and 30ms temporal offset. The relative pose errors of proposed method are almost same under two different temporal offsets (black and blue plots). However, the relative pose error of OKVIS increase a lot when temporal offset increases (pink and red plots).

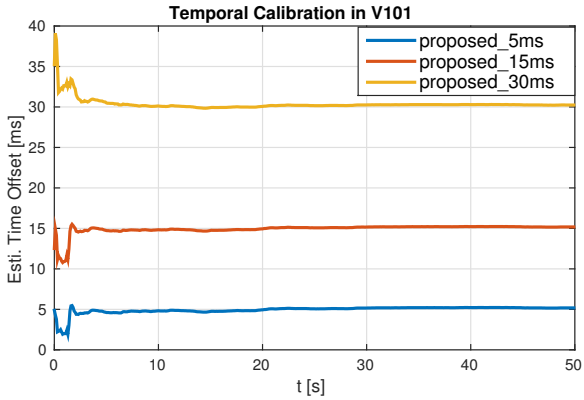

Fig. 9. Temporal offset estimation in V101 sequence with 5ms, 15ms and 30ms temporal offset. Estimated offsets converge to the stable value quickly within a few seconds.

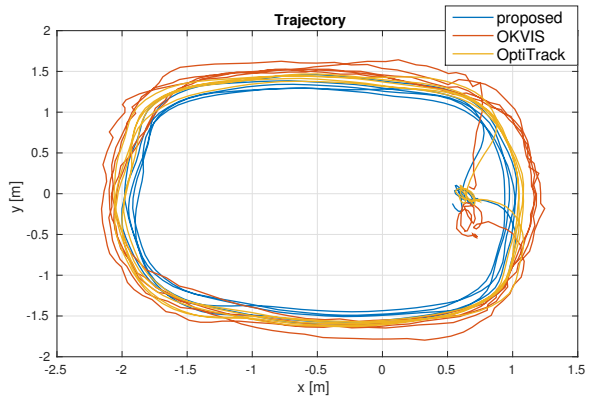

Fig. 10. Trajectory of real world experiment. We hold the sensor and walked five circles. Proposed method compares against OKVIS and OptiTrack.

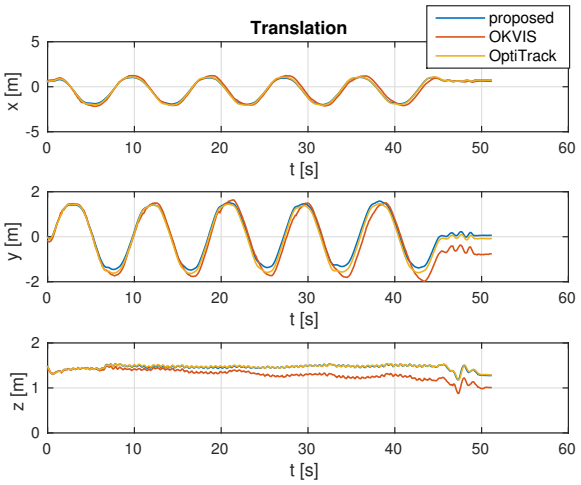

Fig. 11. Translation of real world experiment in x, y, and z axis. Proposed method compares against OKVIS and OptiTrack.

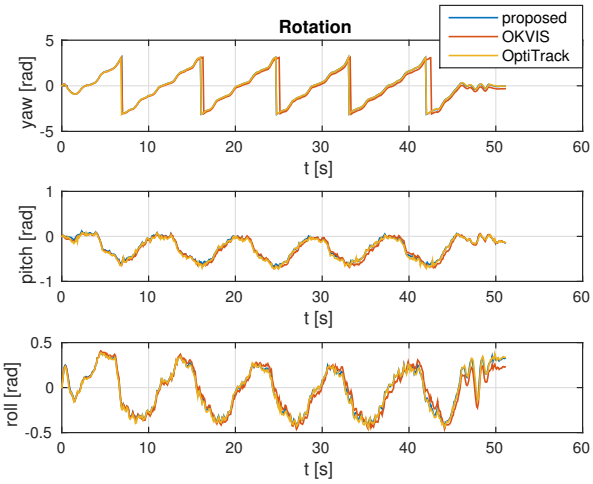

Fig. 12. Rotation of real world experiment in yaw, pitch and roll. Proposed method compares against OKVIS and OptiTrack.

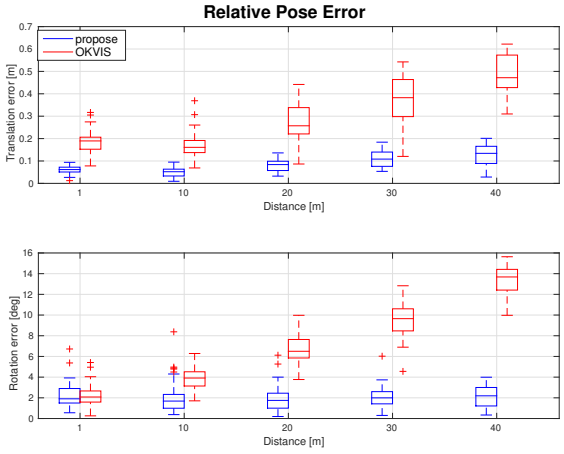

Fig. 13. Relative pose error [29] comparison between proposed method and OKVIS in real-world experiment.

offset, because the proposed method can calibrate time offset correctly. The calibration results are also listed in the Table. The proposed method can accurately calibrate predefined time offset. The Proposed method obviously outperforms OKVIS when time offset is larger than 10ms.

Specifically, relative pose error [29] comparison between proposed method and OKVIS is shown in Fig. 8. The figure is conducted on V101 sequence under 5ms and 30ms temporal offset. We can see that the relative pose errors of the proposed method are almost same under two different temporal offsets (black and blue plots). However, the relative pose error of OKVIS increases a lot when temporal offset increases (pink and red plots).

The process of temporal offset estimation is shown in Fig. 9. It can be seen that the estimated offset converges to the stable value quickly only within a few seconds. Online temporal calibration significantly benefits overall performance.

2) Real-world Experiment: We carried out a real-world experiment to validate the proposed system. The sensor set is same as in Sec. IV-B.2, Intel Realsense camera, as shown in Fig. 5. The image rate is 30Hz, and the inertial measurement's rate is 350Hz. We held the sensor suite by hand and walk circularly at a normal pace in a room. We compare our results against OKVIS [16]. Meanwhile, the results from OptiTrack [3] are treated as ground truth.

We held the sensor and walked five circles. The trajectory is depicted in Fig. 10. The detailed translation in x, y and z-axis is shown in Fig. 11. The detailed rotation in yaw, pitch, and roll is shown in Fig. 12. In translation and rotation comparison, we can see that OKVIS's results drift noticeably along with the time. Relative pose error [29] comparison between the proposed method and OKVIS is shown in Fig.

[3]http://optitrack.com/

13. The relative pose error of OKVIS is larger than the proposed method. Moreover, the relative pose error increases at a faster speed than the proposed method. Obviously, the proposed method outperforms OKVIS in both translation and rotation due to online temporal calibration. The temporal offset calibrated from the proposed method is 12.74ms, which will significantly affect VIO performances in a long run without effective calibration and compensation.

V. Conclusion

In this paper, we have presented an online approach to calibrate time offset between IMU and camera. Our approach is a general model, which can be easily adopted in optimization-based visual-inertial frameworks. The time offset is jointly optimized along with IMU and camera states, as well as features. Our simulation and experimental results indicate the proposed approach can achieve high accuracy in both time offset calibration and system's motion estimation, even compared with other state-of-art offline methods. Although we use the monocular sensor suite to showcase our method in this paper, the proposed method can be easily generalized to multi-camera visual-inertial systems.

References

[1] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[2] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *Mixed and Augmented Reality, 2007. IEEE and ACM International Symposium on*, 2007, pp. 225–234.

[3] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Proc. of the IEEE Int. Conf. on Robot. and Autom.*, Hong Kong, China, May 2014.

[4] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European Conference on Computer Vision*. Springer International Publishing, 2014, pp. 834–849.

[5] R. Mur-Artal, J. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, 2015.

[6] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "isam2: Incremental smoothing and mapping using the bayes tree," *Int. J. Robot. Research*, vol. 31, no. 2, pp. 216–235, 2012.

[7] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[8] P. Corke, J. Lobo, and J. Dias, "An introduction to inertial and visual sensing," *Int. J. Robot. Research*, vol. 26, no. 6, pp. 519–535, 2007.

[9] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. of the IEEE Int. Conf. on Robot. and Autom.*, Roma, Italy, Apr. 2007, pp. 3565–3572.

[10] M. Li and A. Mourikis, "High-precision, consistent EKF-based visual-inertial odometry," *Int. J. Robot. Research*, vol. 32, no. 6, pp. 690–711, May 2013.

[11] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments," in *Proc. of the IEEE Int. Conf. on Robot. and Autom.*, 2012, pp. 957–964.

[12] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to mav navigation," in *Proc. of the IEEE/RSJ Int. Conf. on Intell. Robots and Syst.* IEEE, 2013, pp. 3923–3929.

[13] S. Shen, N. Michael, and V. Kumar, "Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft MAVs," in *Proc. of the IEEE Int. Conf. on Robot. and Autom.*, Seattle, WA, May 2015.

[14] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular slam with map reuse," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.

[15] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2017.

[16] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," *Int. J. Robot. Research*, vol. 34, no. 3, pp. 314–334, Mar. 2014.

[17] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct ekf-based approach," in *Proc. of the IEEE/RSJ Int. Conf. on Intell. Robots and Syst.* IEEE, 2015, pp. 298–304.

[18] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation," in *Proc. of Robot.: Sci. and Syst.*, Rome, Italy, Jul. 2015.

[19] E. Mair, M. Fleps, M. Suppa, and D. Burschka, "Spatio-temporal initialization for imu to camera registration," in *IEEE International Conference on Robotics and Biomimetics*, 2011, pp. 557–564.

[20] J. Kelly and G. S. Sukhatme, "A general framework for temporal calibration of multiple proprioceptive and exteroceptive sensors," *Springer Tracts in Advanced Robotics*, vol. 79, pp. 195–209, 2010.

[21] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *Proc. of the IEEE/RSJ Int. Conf. on Intell. Robots and Syst.* IEEE, 2013, pp. 1280–1286.

[22] M. Li and A. I. Mourikis, "3-d motion estimation and online temporal calibration for camera-imu systems," in *Proc. of the IEEE Int. Conf. on Robot. and Autom.* IEEE, 2013.

[23] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *arXiv preprint arXiv:1708.03852*, 2017.

[24] J. Shi and C. Tomasi, "Good features to track," in *Proc. of the IEEE Int. Conf. on Pattern Recognition*, Seattle, WA, Jun. 1994, pp. 593–600.

[25] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. of the Intl. Joint Conf. on Artificial Intelligence*, Vancouver, Canada, Aug. 1981, pp. 24–28.

[26] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.

[27] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *Int. J. Robot. Research*, 2016.

[28] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proc. of the IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, 2012, pp. 573–580.

[29] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proc. of the IEEE Int. Conf. on Pattern Recognition*, 2012, pp. 3354–3361.