Efficient Visual-Inertial Navigation using a Rolling-Shutter Camera with Inaccurate Timestamps

Chao X. Guo, Dimitrios G. Kottas, Ryan C. DuToit, Ahmed Ahmed, Ruipeng Li, and Stergios I. Roumeliotis

***Abstract*—In order to develop Vision-aided Inertial Navigation Systems (VINS) on mobile devices, such as cell phones and tablets, one needs to consider two important issues, both due to the commercial-grade underlying hardware: (i) The unknown and varying time offset between the camera and IMU clocks (ii) The rolling-shutter effect caused by CMOS sensors. Without appropriately modelling their effect and compensating for them online, the navigation accuracy will significantly degrade. In this work, we introduce a linear-complexity algorithm for fusing inertial measurements with time-misaligned, rolling-shutter images using a highly efficient and precise linear interpolation model. As a result, our algorithm achieves a better accuracy and improved speed compared to existing methods. Finally, we validate the superiority of the proposed algorithm through simulations and real-time, online experiments on a cell phone.**

I. Introduction and Related Work

Among the methods employed for tracking the six-degrees-of-freedom (d.o.f.) position and orientation (pose) of a sensing platform within GPS-denied environments, vision-aided inertial navigation is one of the most established, primarily due to its high precision and low cost. During the past decade, VINS have been successfully applied to spacecraft [20], automotive [17], and personal localization [9], demonstrating real-time performance.

The increasing range of sensing capabilities offered by modern cell phones, as well as their increasing computational resources make them ideal for applying VINS. Fusing visual and inertial measurements on a cell phone, however, requires addressing two key problems, both of which are related to the low-cost, commercial-grade hardware used. First, the camera and inertial measurement unit (IMU) have separate clocks, which are not synchronized. Hence, visual and inertial measurements which may correspond to the same time instant, will be reported with a time difference between them. Furthermore, this time offset may change over time due to inaccuracies in the sensors' clocks, or clock jitters from CPU overloading. Therefore, high-accuracy navigation on a cell phone requires modelling and online estimating such time parameters. Second, commercial-grade CMOS sensors suffer from the rolling-shutter effect; that is each pixel row of the imager is read at a different time instant, resulting in an ensemble distorted image. Thus, an image captured by a rolling-shutter camera under motion will contain bearing measurements to features which are recorded at different camera poses. Achieving high-accuracy navigation requires properly modelling and compensating for this phenomenon.

Both the time synchronization and rolling-shutter effect correspond to a time offset between visual and inertial measurements. Previous works have demonstrated offline methods for calibrating a *constant* time offset between a camera and an IMU [13, 4], or the readout time of a rolling-shutter camera [21, 2]. However, the equipment required for offline calibration is not always available. Furthermore, since the time offset between the two clocks may jitter, the result of an offline calibration process may be of limited use in practice.

This motivates us to introduce a new measurement model for fusing rolling-shutter images that have a time offset with inertial measurements. Ideally, modelling the rolling-shutter effect would require estimating the pose corresponding to each pixel row. However, this would lead to an intractable computational cost. An efficient, real-time sensor fusion algorithm taking into account the rolling shutter and time misalignment effects requires an alternative, approximate, camera measurement model.

By exploiting the underlying kinematic motion model, one can employ the estimated linear and rotational velocity for relating camera measurements with IMU poses corresponding to different time instants. Following such an approach, in [1], the authors proposed a vision-only structure from motion algorithm, tailored to rolling-shutter images. In [12], an EKF is employed for estimating the rotational velocity of a rolling-shutter camera as an aid for video rectification. Recently, Li et al. adapted this idea to the case of the Multi-State Constrained Kalman Filter (MSC-KF) to account for the rolling shutter and time synchronization effects [15, 16]. Such an approach, however, has two main drawbacks: (i) Increased computational cost due to the extra parameters (linear and rotational velocities) that need to be estimated for each processed camera image. As we will show later on, this introduces a significant computational burden to the MSC-KF. (ii) The requirement to switch between two propagation models, which increases the algorithm's implementation complexity.

In this work, we propose an interpolation-based camera measurement model, targeting vision-aided inertial navigation using low-grade rolling-shutter cameras. In particular, the main contributions of this paper are:

- We introduce an interpolation model for expressing the camera pose of each visual measurement, as a function of adjacent IMU poses, that are included in the estimator's optimization window.
- A significant speedup compared to [15, 16] for fusing

This work was supported by the National Science Foundation (IIS-0835637), the Air Force Office of Scientific Research (FA9550-10-1-0567), and the University of Minnesota through the Digital Technology Center (DTC) and the ADC Fellowship.

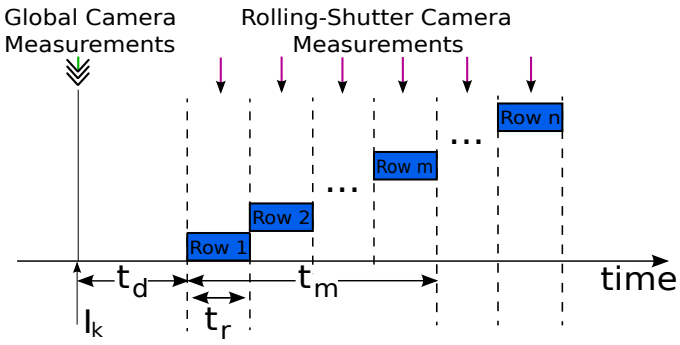


Fig. 1. Time synchronization and rolling-shutter effect.

visual and inertial measurements while compensating for varying time offset and rolling shutter.

- We determine the system's unobservable directions when applying our interpolation measurement model, and use them to improve the VINS consistency and accuracy by employing an Observability-Constrained Extended Kalman filter (OC-EKF).
- We validate the proposed algorithm in simulation, as well as through *real-time*, *online* and *offline* experiments using a cell phone.

The rest of the paper is structured as follows. In Section II, we explain how the camera-IMU lack of synchronization and the rolling-shutter effect cause a time offset between camera and IMU measurements. In Section III, we describe our interpolation-based camera measurement model employed for compensating for the time synchronization and rolling-shutter effects. In Section IV, we present the modified MSCKF-based algorithm for six d.o.f. sensor motion estimation. Furthermore, we compare its computational complexity to that of [15, 16]. In Section V, we present our OC-EKF implementation that uses an interpolation-based camera measurement model. In Section VI, we evaluate the performance of our proposed algorithm, both in simulation and experimentally. Finally, in Section VII, we provide concluding remarks.

II. Time Misalignment due to Time Synchronization and Rolling-Shutter Effects

Most prior work on VINS assumes a global shutter camera perfectly synchronized with the IMU. In such a model, all pixel measurements of an image are recorded at the same time instant as a particular IMU measurement. However, this is unrealistic for most consumer devices mainly for two reasons: (i) The camera and IMU clocks may not be synchronized. That is, when measuring the same event, the time stamp reported by the camera and IMU will differ. (ii) The camera and IMU may sample at a different frequency and phase, meaning that measurements do not necessarily occur at the same time instant. Thus, a varying time delay, t_d, between the corresponding camera and IMU measurements exists, which needs to be appropriately modelled.

In addition, if a rolling-shutter camera is used, an extra time offset introduced by the rolling-shutter effect, should be

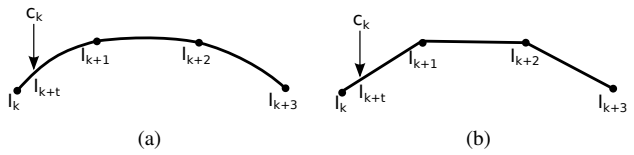


Fig. 2. The cell phone's trajectory between poses I_k and I_{k+3}. The camera measurement, C_k, is recorded at the time instant $k+t$ between poses I_k and I_{k+1}. (a) The real cell phone trajectory (b) The cell phone trajectory with linear approximation.

accounted for. Specifically, the rolling-shutter camera reads the imager row by row, so the time delay for a pixel measurement in row m with image readout time t_m can be computed as $t_m = mt_r$, where t_r is the read time of a single row.

As depicted in Fig. 1, both the time delay of the camera, as well as the rolling-shutter effect can be represented by a single time offset, corresponding to each row of pixels. For a pixel measurement in the m-th row of the image, the time difference can be written as: $t = t_d + t_m$.

Ignoring such time delays can lead to significant performance degradation (see Sec. VI-B). To address this problem, we introduce a measurement model that approximates the pose corresponding to a particular camera measurement as a linear interpolation (or extrapolation, if necessary) of the closest (in time) IMU poses, among the ones that comprise the estimator's optimization window (see Fig. 2).

III. Camera Model for Time Misaligned Measurements

In this section, we present the proposed interpolation-based measurement model for expressing the pose, $\mathbf{I}_{k+t}$, corresponding to image C_k (see Fig. 2(a)), as a function of the poses comprising the estimator's optimization window. Several methods exist for approximating a 3D trajectory as a polynomial function of time, such as the *Spline* method [23]. Rather than using a high-order polynomial, we choose to employ a linear interpolation model. Such a choice is motivated by the short time period between two consecutive poses, $\mathbf{I}_k$ and $\mathbf{I}_{k+1}$, that are adjacent to the pose $\mathbf{I}_{k+t}$, which corresponds to the recorded camera image.

Specifically, defining $\{G\}$ as the global frame of reference and an interpolation ratio $\lambda_k \in [0,\ 1]$ (in this case, λ_k is the distance between $\mathbf{I}_k$ and $\mathbf{I}_{k+t}$ over the distance between $\mathbf{I}_k$ and $\mathbf{I}_{k+1}$), the translation interpolation ${}^G\mathbf{p}_{I_{k+t}}$ between two IMU positions ${}^G\mathbf{p}_{I_k}$ and ${}^G\mathbf{p}_{I_{k+1}}$ expressed in $\{G\}$, can be easily approximated as:

$$ {}^G\mathbf{p}_{I_{k+t}} = (1-\lambda_k){}^G\mathbf{p}_{I_k} + \lambda_k {}^G\mathbf{p}_{I_{k+1}} \tag{1} $$

In contrast, the interpolation of the frames' orientations is more complicated, due to the nonlinear representation of rotations. In [23], Shoemaker proposed the SLERP model for rotation interpolation, which is designed to apply interpolation on the arc defined by two quaternions on the unit sphere. Although it is geometrically elegant, such a model leads to cumbersome expressions. Instead, in our case one can take advantage of two characteristics of the problem at hand for

designing a simpler model: (i) The IMU pose is cloned [1] at around 5 Hz (the same frequency as processing image measurements), thus the rotation between consecutive poses, $\mathbf{I}_k$ and $\mathbf{I}_{k+1}$, is small during regular motion. (ii) We can always clone the IMU pose at the time instant closest to the image's recording time, thus the interpolated pose $\mathbf{I}_{k+t}$ is very close to the pose $\mathbf{I}_k$ and the rotation between them is very small.

Exploiting (i), the rotation between the consecutive IMU orientations, described by the rotation matrices ${}^{G}_{I_k}\mathbf{C}$ and ${}^{G}_{I_{k+1}}\mathbf{C}$, respectively, expressed in $\{G\}$, can be written as:

$$\begin{aligned} {}^{I_{k+1}}_{G}\mathbf{C}\ {}^{G}_{I_k}\mathbf{C} &= \cos\alpha\mathbf{I} - \sin\alpha\lfloor\boldsymbol{\theta}\rfloor + (1-\cos\alpha)\boldsymbol{\theta}\boldsymbol{\theta}^T \\ &\simeq \mathbf{I} - \alpha\lfloor\boldsymbol{\theta}\rfloor \end{aligned} \tag{2}$$

where we have employed the small-angle approximation, $\lfloor\boldsymbol{\theta}\rfloor$ denotes the skew-symmetric matrix of the 3×1 rotation axis, $\boldsymbol{\theta}$, and α is the rotation angle. Similarly, according to (ii) the rotation interpolation ${}^{I_{k+t}}_{I_k}\mathbf{C}$ between ${}^{G}_{I_k}\mathbf{C}$ and ${}^{G}_{I_{k+1}}\mathbf{C}$ can be written as:

$$\begin{aligned} {}^{I_{k+t}}_{I_k}\mathbf{C} &= \cos(\lambda_k\alpha)\mathbf{I} - \sin(\lambda_k\alpha)\lfloor\boldsymbol{\theta}\rfloor + (1-\cos(\lambda_k\alpha))\boldsymbol{\theta}\boldsymbol{\theta}^T \\ &\simeq \mathbf{I} - \lambda_k\alpha\lfloor\boldsymbol{\theta}\rfloor \end{aligned} \tag{3}$$

If we substitute $\alpha\lfloor\boldsymbol{\theta}\rfloor$ from (2) into (3), ${}^{I_{k+t}}_{I_k}\mathbf{C}$ can be expressed in terms of two consecutive rotations:

$${}^{I_{k+t}}_{I_k}\mathbf{C} \simeq (1-\lambda_k)\mathbf{I} + \lambda_k {}^{I_k}_{G}\mathbf{C}\ {}^{G}_{I_{k+1}}\mathbf{C} \tag{4}$$

This interpolation model is exact at the two end points ($\lambda_k = 0$ *or* 1), and less accurate for points in the middle of the interpolation interval (i.e., the resulting rotation matrix does not belong to $SO(3)$). Since we can always choose to place the cloned IMU poses as close as possible to the reported time of the image, such a model can fit the purposes of our application.

IV. Estimation Algorithm Description

In this section, we will introduce the proposed VINS that utilizes a rolling-shutter camera with a varying time offset. Our goal is to estimate the 3D position and orientation of a device equipped with an IMU and a rolling-shutter camera. The measurement frequencies of both sensors are assumed known, while there exists an unknown time offset between the IMU and the camera timestamps. Our algorithm derives from the MSC-KF [20], which is a linear-complexity (in the number of features tracked) visual-inertial odometry algorithm, initially designed for inertial and global shutter camera measurements that are perfectly time synchronized. Rather than maintaining a map of the environment, the MSC-KF marginalizes all observed features, exploiting all available information for estimating a sliding window of past camera poses. In what follows, we will first present the state vector, and system propagation using inertial measurements. Then, we will introduce the proposed measurement model and the corresponding EKF measurement update.

[1] We refer to stochastic cloning [22], for maintaining past IMU poses in the sliding window of the estimator.

A. System State

The state vector we estimate is:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_I & \mathbf{x}_{I_{k+n-1}} & \cdots & \mathbf{x}_{I_k} \end{bmatrix} \tag{5}$$

where $\mathbf{x}_I$ denotes the current robot pose, and $\mathbf{x}_{I_i}$, for $i = k+n-1,\ldots,k$, are the cloned IMU poses in the sliding window, corresponding to the time instants of the last n camera measurements. Specifically, the current robot pose is defined as: [2]

$$\mathbf{x}_I = \begin{bmatrix} {}^{I}\mathbf{q}_G^T & {}^{G}\mathbf{v}_I^T & {}^{G}\mathbf{p}_I^T & \mathbf{b}_a^T & \mathbf{b}_g^T & \lambda_d & \lambda_r \end{bmatrix}^T$$

where ${}^{I}\mathbf{q}_G$ is the quaternion representation of the orientation of $\{G\}$ in the IMU's frame of reference $\{I\}$, ${}^{G}\mathbf{v}_I$ and ${}^{G}\mathbf{p}_I$ are the velocity and position of $\{I\}$ in $\{G\}$ respectively, while $\mathbf{b}_a$ and $\mathbf{b}_g$ correspond to the gyroscope and accelerometer biases. The interpolation ratio can be divided into a time-variant part, λ_d, and a time-invariant part, λ_r. In our case, λ_d corresponds to the IMU-camera time offset, t_d, while λ_r corresponds to the readout time of an image-row, t_r. Specifically,

$$\lambda_d = \frac{t_d}{t_{intvl}} \qquad \lambda_r = \frac{t_r}{t_{intvl}} \tag{6}$$

where t_{intvl} is the time interval between two consecutive IMU poses (known). Then, the interpolation ratio for a pixel measurement in the m-th row of the image is written as:

$$\lambda = \lambda_d + m\lambda_r \tag{7}$$

When a new image measurement arrives, we clone the IMU pose at the time instant closest to the image recording time. The cloned IMU poses $\mathbf{x}_{I_i}$ are defined as:

$$\mathbf{x}_{I_i} = \begin{bmatrix} {}^{I_i}\mathbf{q}_G^T & {}^{G}\mathbf{p}_{I_i}^T & \lambda_{d_i} \end{bmatrix}^T$$

where ${}^{I_i}\mathbf{q}_G$, ${}^{G}\mathbf{p}_{I_i}^T$, λ_{d_i} are cloned at the time instant that the i-th image was recorded. Note, that λ_{d_i} is also cloned because the time offset between the IMU and camera may change over time.

According to (5), for a system with a fixed number of cloned IMU poses, the size of the system's state vector depends on the dimension of each cloned IMU pose. In contrast to Li et al.'s approach [15, 16], which requires to also clone the linear and rotational velocities, our interpolation-based measurement model reduces the dimension of the cloned state from 13 to 7. As we will show later on, this smaller clone state size significant minimizes the algorithm's computational complexity.

B. Propagation

When a new inertial measurement arrives, we use it to propagate the EKF state and covariance. In this section, we will present the state and covariance propagation of the current robot pose and the cloned IMU poses.

[2] For clarity of presentation, we assume that the IMU and camera frames spatially coincide, while in practice they can be extrinsically calibrated following the approach of [14].

(a) Current pose propagation: The continuous-time system model describing the time evolution of the states is:[3]

$$
\begin{aligned}
{}^{I}\dot{\mathbf{q}}_G(t) &= \frac{1}{2}\mathbf{\Omega}(\boldsymbol{\omega}_m(t)-\mathbf{b}_g(t)-\mathbf{n}_g(t)){}^{I}\mathbf{q}_G(t)\\
{}^{G}\dot{\mathbf{v}}_I(t) &= \mathbf{C}({}^{I}\mathbf{q}_G(t))^T(\mathbf{a}_m(t)-\mathbf{b}_a(t)-\mathbf{n}_a(t))+{}^{G}\mathbf{g}\\
{}^{G}\dot{\mathbf{p}}_I(t) &= {}^{G}\mathbf{v}_I(t) \qquad \dot{\mathbf{b}}_a(t)=\mathbf{n}_{wa} \qquad \dot{\mathbf{b}}_g(t)=\mathbf{n}_{wg}\\
\dot{\lambda}_d(t) &= n_{td} \quad \dot{\lambda}_r(t)=0
\end{aligned}
\tag{8}
$$

where $\mathbf{C}({}^{I}\mathbf{q}_G(t))$ denotes the rotation matrix corresponding to ${}^{I}\mathbf{q}_G(t)$, $\boldsymbol{\omega}_m(t)$ and $\mathbf{a}_m(t)$ are the rotational velocity and linear acceleration measurements provided by the IMU, while $\mathbf{n}_g$ and $\mathbf{n}_a$ are the corresponding white Gaussian measurement noise components. ${}^{G}\mathbf{g}$ denotes the gravitational acceleration in $\{G\}$, while $\mathbf{n}_{wa}$ and $\mathbf{n}_{wg}$ are zero-mean white Gaussian noise processes driving the gyroscope and accelerometer biases $\mathbf{b}_g$ and $\mathbf{b}_a$. Finally, n_{td} is a zero-mean white Gaussian noise process modelling the random walk of λ_d (corresponding to the time offset between the IMU and camera). For state propagation, we linearize around the current state estimate and apply the expectation operator to (8), as shown in [20]. For propagating the covariance, we first define the error-state vector of the current robot pose as: [4]

$$
\widetilde{\mathbf{x}} = \begin{bmatrix} {}^{I}\delta\boldsymbol{\theta}_G^T & {}^{G}\widetilde{\mathbf{v}}_I^T & {}^{G}\widetilde{\mathbf{p}}_I^T & {}^{G}\widetilde{\mathbf{p}}_f^T & \widetilde{\mathbf{b}}_a^T & \widetilde{\mathbf{b}}_g^T & \widetilde{\lambda}_d & \widetilde{\lambda}_r \end{bmatrix}^T \tag{9}
$$

Then, as shown in [20], the linearized continuous-time error-state equation can be written as:

$$
\dot{\widetilde{\mathbf{x}}} = \mathbf{F}_E\widetilde{\mathbf{x}} + \mathbf{G}_E\,\mathbf{w} \tag{10}
$$

where $\mathbf{w} = \begin{bmatrix} \mathbf{n}_g^T & \mathbf{n}_{wg}^T & \mathbf{n}_a^T & \mathbf{n}_{wa}^T & n_{td} \end{bmatrix}^T$ is modelled as a zero-mean white Gaussian process with auto-correlation $\mathbb{E}[\mathbf{w}(t)\mathbf{w}^T(\tau)] = \mathbf{Q}_E\delta(t-\tau)$, and $\mathbf{F}_E$, $\mathbf{G}_E$ are the continuous-time error-state transition and input noise matrices, respectively. Following [19], the discrete-time state transition matrix $\mathbf{\Phi}_{k+1.k}$ and the system covariance matrix $\mathbf{Q}_k$ from time t_k to t_{k+1} can be computed as:

$$
\begin{aligned}
\mathbf{\Phi}_{k+1,k} &= \mathbf{\Phi}(t_{k+1},t_k) = \exp\left(\int_{t_k}^{t_{k+1}}\mathbf{F}_E(\tau)\mathrm{d}\tau\right) \qquad (11)\\
\mathbf{Q}_k &= \int_{t_k}^{t_{k+1}}\mathbf{\Phi}(t_{k+1},\tau)\mathbf{G}_E\mathbf{Q}_E\mathbf{G}_E^T\mathbf{\Phi}^T(t_{k+1},\tau)\mathrm{d}\tau
\end{aligned}
$$

If we define the covariance corresponding to the current pose as $\mathbf{P}_{EE_{k|k}}$, the propagated covariance $\mathbf{P}_{EE_{k+1|k}}$ can be determined as[5]

$$
\mathbf{P}_{EE_{k+1|k}} = \Phi_{k+1,k}\mathbf{P}_{EE_{k|k}}\mathbf{\Phi}_{k+1,k}^T + \mathbf{Q}_k \tag{12}
$$

(b) System propagation: During propagation, the state and covariance estimates of the cloned robot poses do not change, however their cross-correlations with the current IMU pose need to be propagated. If we define $\mathbf{P}$ as the covariance matrix of the whole state $\mathbf{x}$, $\mathbf{P}_{CC_{k|k}}$ as the covariance matrix of the cloned poses, and $\mathbf{P}_{EC_{k|k}}$ as the correlation matrix between the errors in the current pose and cloned poses, the system covariance matrix is propagated as:

$$
\mathbf{P}_{k+1|k} = \begin{bmatrix} \mathbf{P}_{EE_{k+1|k}} & \mathbf{\Phi}_{k+1,k}\mathbf{P}_{EC_{k|k}} \\ \mathbf{P}_{EC_{k|k}}^T\mathbf{\Phi}_{k+1,k}^T & \mathbf{P}_{CC_{k|k}} \end{bmatrix} \tag{13}
$$

with $\Phi_{k+1,k}$ defined in (11).

C. MSC-KF Measurement Model For Rolling-Shutter & Time Synchronization

Each time the camera records an image, a stochastic clone [22] comprising the IMU pose, ${}^{I}\mathbf{q}_G$, ${}^{G}\mathbf{p}_I$, and the interpolation ratio, λ_d, describing its time offset from the image, is created. This process enables the MSC-KF to utilize delayed image measurements; in particular, it allows all observations of a given feature $\mathbf{f}_j$ to be processed during a single update step (when the first pose that observed $\mathbf{f}_j$ is about to be marginalized), while avoiding to maintain estimates of this feature, in the state vector.

For a feature $\mathbf{f}_j$ observed in the m-th row of the image associated with the IMU pose I_k, the interpolation ratio can be expressed as $\lambda_k = \lambda_{d_k} + m\lambda_r$, where λ_{d_k} is the interpolation ratio corresponding to the time offset between the clocks of the two sensors at time step k, and $m\lambda_r$ is the contribution from the rolling-shutter effect. The corresponding measurement model is given by:

$$
\mathbf{z}_k^{(j)} = \mathbf{h}({}^{I_{k+t}}\mathbf{p}_{f_j}) + \mathbf{n}_k^{(j)},\ \mathbf{n}_k^{(j)} \sim N(\mathbf{0},\mathbf{R}_{k,j}) \tag{14}
$$

where ${}^{I_{k+t}}\mathbf{p}_{f_j}$ is the feature position expressed in the camera frame of reference at the exact time instant that the m-th image-row was read. Without loss of generality, we assume that the camera is intrinsically calibrated with the camera perspective measurement model, $\mathbf{h}$, described by:

$$
\mathbf{h}({}^{I_{k+t}}\mathbf{p}_{f_j}) = \begin{bmatrix} \frac{{}^{I_{k+t}}\mathbf{p}_{f_j}(1)}{{}^{I_{k+t}}\mathbf{p}_{f_j}(3)} \\ \frac{{}^{I_{k+t}}\mathbf{p}_{f_j}(2)}{{}^{I_{k+t}}\mathbf{p}_{f_j}(3)} \end{bmatrix} \tag{15}
$$

where ${}^{I_{k+t}}\mathbf{p}_{f_j}(i),\ i=1,2,3$ represents the i-th element of ${}^{I_{k+t}}\mathbf{p}_{f_j}$. Expressing ${}^{I_{k+t}}\mathbf{p}_f$ as a function of the states that we estimate, results in:

$$
\begin{aligned}
{}^{I_{k+t}}\mathbf{p}_{f_j} &= {}^{I_{k+t}}_{G}\mathbf{C}({}^{G}\mathbf{p}_{f_j} - {}^{G}\mathbf{p}_{I_{k+t}}) \qquad (16)\\
&= {}^{I_{k+t}}_{I_k}\mathbf{C}\ {}^{I_k}_{G}\mathbf{C}({}^{G}\mathbf{p}_{f_j} - {}^{G}\mathbf{p}_{I_{k+t}})
\end{aligned}
$$

Substituting ${}^{I_{k+t}}_{I_k}\mathbf{C}$ and ${}^{G}\mathbf{p}_{I_{k+t}}$ from (4) and (1), (16) can be rewritten as:

$$
\begin{aligned}
{}^{I_{k+t}}\mathbf{p}_{f_j} = &\left((1-\lambda_k)\mathbf{I} + \lambda_k\ {}^{I_k}_{G}\mathbf{C}\ {}^{G}_{I_{k+1}}\mathbf{C}\right){}^{I_k}_{G}\mathbf{C} \\
&\left({}^{G}\mathbf{p}_{f_j} - ((1-\lambda_k){}^{G}\mathbf{p}_{I_k} + \lambda_k\,{}^{G}\mathbf{p}_{I_{k+1}})\right) \qquad (17)
\end{aligned}
$$

Linearizing the measurement model about the filter estimates, the residual corresponding to this measurement can be com-

[3] $\mathbf{\Omega}(\boldsymbol{\omega})$ is defined as: $\mathbf{\Omega}(\boldsymbol{\omega}) \triangleq \begin{bmatrix} -\lfloor\boldsymbol{\omega}\rfloor & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & \mathbf{0} \end{bmatrix}$

[4] For quaternion $\mathbf{q}$ we employ a multiplicative error model $\delta\bar{q} = \bar{q}\otimes\hat{\bar{q}}^{-1} \simeq \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta}^T & 1 \end{bmatrix}^T$, where $\delta\boldsymbol{\theta}$ is a minimal representation of the attitude error.

[5] $\mathbf{x}_{k|\ell}$ denotes the estimate of $\mathbf{x}$ at time step k using measurements up to time step ℓ.

puted as:

$$\begin{aligned}\mathbf{r}_k^{(j)} &= \mathbf{z}_k^{(j)} - \mathbf{h}({}^{I_{k+t}}\hat{\mathbf{p}}_{f_j}) \\ &\simeq \mathbf{H}_{\mathbf{x}_{I_k}}^{(j)}\widetilde{\mathbf{x}}_{I_k} + \mathbf{H}_{\mathbf{x}_{I_{k+1}}}^{(j)}\widetilde{\mathbf{x}}_{I_{k+1}} + \mathbf{H}_{f_k}^{(j)}{}^{G}\widetilde{\mathbf{p}}_{f_j} + \mathbf{H}_{\lambda_{r_k}}^{(j)}\widetilde{\lambda}_r + \mathbf{n}_k^{(j)} \end{aligned} \quad (18)$$

where $\mathbf{H}_{\mathbf{x}_{I_k}}^{(j)}$, $\mathbf{H}_{\mathbf{x}_{I_{k+1}}}^{(j)}$, $\mathbf{H}_{f_k}^{(j)}$ and $\mathbf{H}_{\lambda_{r_k}}^{(j)}$ are the Jacobians with respect to the cloned poses $\mathbf{x}_{I_k}$, $\mathbf{x}_{I_{k+1}}$, the feature position ${}^{G}\mathbf{p}_{f_j}$, and the interpolation ratio corresponding to the image-row readout time, λ_r, respectively.

By stacking the measurement residuals corresponding to the same point feature, $\mathbf{f}_j$, we arrive at:

$$\mathbf{r}^{(j)} = \begin{bmatrix} \mathbf{r}_k^{(j)} \\ \vdots \\ \mathbf{r}_{k+n-1}^{(j)} \end{bmatrix} \simeq \mathbf{H}_{\mathbf{x}_{clone}}^{(j)}\widetilde{\mathbf{x}}_{clone} + \mathbf{H}_f^{(j)}{}^{G}\widetilde{\mathbf{p}}_{f_j} + \mathbf{H}_{\lambda_r}^{(j)}\widetilde{\lambda}_r + \mathbf{n}^{(j)} \quad (19)$$

where $\widetilde{\mathbf{x}}_{clone} = \begin{bmatrix} \widetilde{\mathbf{x}}_{I_{k+n-1}}^T & \cdots & \widetilde{\mathbf{x}}_{I_k}^T \end{bmatrix}^T$ is the error in the cloned pose estimates, while $\mathbf{H}_{\mathbf{x}_{clone}}^{(j)}$ is the corresponding Jacobian matrix. Furthermore, $\mathbf{H}_f^{(j)}$ and $\mathbf{H}_{\lambda_r}^{(j)}$ are the Jacobians corresponding to the feature and interpolation ratio contributed by the readout time error, respectively.

To avoid including feature $\mathbf{f}_j$ in the state vector, we marginalize its error term, ${}^{G}\widetilde{\mathbf{p}}_{f_j}$ by multiplying both sides of (19) with the left nullspace, $\mathbf{V}$, of the feature's Jacobian matrix $\mathbf{H}_f^{(j)}$, i.e.,

$$\begin{aligned}\mathbf{r}_o^{(j)} &\simeq \mathbf{V}^T\mathbf{H}_{\mathbf{x}_{clone}}^{(j)}\widetilde{\mathbf{x}}_{clone} + \mathbf{V}^T\mathbf{H}_f^{(j)}{}^{G}\widetilde{\mathbf{p}}_{f_j} + \mathbf{V}^T\mathbf{H}_{\lambda_r}^{(j)}\widetilde{\lambda}_r + \mathbf{V}^T\mathbf{n}^{(j)} \\ &\triangleq \mathbf{H}_o^{(j)}\widetilde{\mathbf{x}} + \mathbf{n}_o^{(j)} \end{aligned} \quad (20)$$

where $\mathbf{r}_o^{(j)} \triangleq \mathbf{V}^T\mathbf{r}^{(j)}$. Note that we do not have to compute $\mathbf{V}$ explicitly. Instead, this operation can be applied efficiently using in-place Givens rotations [5].

D. Filter Updates

In the previous section, we formulated the measurement model for each individual feature. Specifically, we compensated for the time-misaligned camera measurements with the interpolation ratio corresponding to both the time offset between sensors and the rolling shutter effect. Additionally, we removed the dependence of our measurement model on the feature positions. Hereafter, we will describe the EKF updates using all the available measurements from L features.

Stacking measurements of the form (20), originating from all features, $\mathbf{f}_j,\ j = 1,\ldots,L$, yields the residual vector:

$$\mathbf{r} \simeq \mathbf{H}\widetilde{\mathbf{x}} + \mathbf{n} \quad (21)$$

where $\mathbf{H}$ is a matrix with block rows the Jacobians $\mathbf{H}_o^{(j)}$, while $\mathbf{r}$ and $\mathbf{n}$ are the corresponding residual and noise vectors, respectively.

In practice, $\mathbf{H}$ is a tall matrix. Following [20], we can reduce the computational cost by employing the QR decomposition of $\mathbf{H}$ denoted as:

$$\mathbf{H} = \begin{bmatrix}\mathbf{Q}_1 & \mathbf{Q}_2\end{bmatrix}\begin{bmatrix}\mathbf{R}_H \\ \mathbf{0}\end{bmatrix} \quad (22)$$

where $\begin{bmatrix}\mathbf{Q}_1 & \mathbf{Q}_2\end{bmatrix}$ is an orthonormal matrix, and $\mathbf{R}_H$ is an upper triangular matrix. Then, if we multiply the transpose of $\begin{bmatrix}\mathbf{Q}_1 & \mathbf{Q}_2\end{bmatrix}$ to both sides of (21), we arrive at:

$$\begin{bmatrix}\mathbf{Q}_1^T\mathbf{r} \\ \mathbf{Q}_2^T\mathbf{r}\end{bmatrix} = \begin{bmatrix}\mathbf{R}_H \\ \mathbf{0}\end{bmatrix}\widetilde{\mathbf{x}} + \begin{bmatrix}\mathbf{Q}_1^T\mathbf{n} \\ \mathbf{Q}_2^T\mathbf{n}\end{bmatrix} \quad (23)$$

It is clear that all information related to the error in the state estimate is included in the first block row, while the residual in the second block row corresponds to noise and can be completely discarded. Therefore, we only need to keep the first block row of (23) as residual for the EKF update:

$$\mathbf{r}_n = \mathbf{Q}_1^T\mathbf{r} = \mathbf{R}_H\widetilde{\mathbf{x}} + \mathbf{Q}_1^T\mathbf{n} \quad (24)$$

The Kalman gain is computed as:

$$\mathbf{K} = \mathbf{P}\mathbf{R}_H^T(\mathbf{R}_H\mathbf{P}\mathbf{R}_H^T + \mathbf{R})^{-1} \quad (25)$$

where $\mathbf{R}$ is the measurement noise. If we define the covariance of the noise $\mathbf{n}$ as $\sigma^2\mathbf{I}$, then $\mathbf{R} = \sigma^2\mathbf{Q}_1^T\mathbf{Q}_1 = \sigma^2\mathbf{I}$. Finally, the state and covariance updates are determined as:

$$\mathbf{x}_{k+1|k+1} = \mathbf{x}_{k+1|k} + \mathbf{K}\mathbf{r}_n \quad (26)$$

$$\mathbf{P}_{k+1|k+1} = \mathbf{P} - \mathbf{P}\mathbf{R}_H^T(\mathbf{R}_H\mathbf{P}\mathbf{R}_H^T + \mathbf{R})^{-1}\mathbf{R}_H\mathbf{P} \quad (27)$$

E. Computational Complexity Comparison

Defining the dimension of $\mathbf{H}$ to be $m \times n$, the computational complexity for the measurement compression QR in (22) will be $O(2mn^2 - \frac{2}{3}n^3)$, and roughly $O(n^3)$ for matrix multiplications or inversions in (25) and (26). Since $\mathbf{H}$ is a very tall matrix, and m is, typically, much larger than n, the main computational cost of the MSC-KF corresponds to the *measurement compression QR* [20]. It is important to note that the number of columns n depends not only on the number of cloned poses, but also on the dimension of each clone. For the proposed approach this would correspond to 7 states per clone (i.e., 6 for the camera pose, and a scalar parameter representing the time-synchronization). In contrast, the method employed in [16] requires 13 states per clone (i.e., 6 for the camera pose, 6 for its corresponding rotational and linear velocities, and a scalar parameter representing the time-synchronization). As we demonstrate experimentally in Sec. VI-C, this key difference results in a **3-fold** computational speedup compared to [16], for this particular step of an MSC-KF update. Furthermore, since the dimension of the system is reduced to almost half through the proposed interpolation model, all the operations in the EKF update will also gain a significant speedup.

V. Observability-Constrained EKF

In [11], it is shown that the linearization error causes the EKF to be inconsistent, thus also adversely affecting the estimation accuracy. In this section, we will show the methodology to address this issue by employing the OC-EKF proposed in [10].

As shown in [19], a system's unobservable directions, $\mathbf{N}$,

span the nullspace of the system's observability matrix $\mathbf{M}$:

$$\mathbf{M}\mathbf{N} = \mathbf{0} \tag{28}$$

where by defining $\mathbf{\Phi}_{k,1} \triangleq \mathbf{\Phi}_{k,k-1}\cdots\mathbf{\Phi}_{2,1}$ as the state transition matrix from time step 1 to k, and $\mathbf{H}_k$ as the measurement Jacobian at time step k, $\mathbf{M}$ can be expressed as:

$$\mathbf{M} = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2\mathbf{\Phi}_{2,1} \\ \vdots \\ \mathbf{H}_k\mathbf{\Phi}_{k,1} \end{bmatrix} \tag{29}$$

However, when the system is linearized using the current estimate, (28), in general, does not hold [11]. This means the estimator gains spurious information along unobservable directions and becomes inconsistent. To address this problem, the OC-EKF [10] enforces (28) by modifying the state transition and measurement Jacobian matrices according to the following two observability constraints:

$$\mathbf{N}_{k+1} = \mathbf{\Phi}_{k+1,k}\mathbf{N}_k \tag{30}$$

$$\mathbf{H}_k\mathbf{N}_k = \mathbf{0}, \quad \forall k > 0 \tag{31}$$

where $\mathbf{N}_k$ and $\mathbf{N}_{k+1}$ are the system's unobservable directions evaluated at time-steps k and $k+1$. Hereafter, we will describe how this method is applied to our system to appropriately modify $\mathbf{\Phi}_{k+1,k}$, as defined in (11), and $\mathbf{H}_k$, and thus retain the system's observability properties.

(a) System Unobservable Directions: In [10], it is shown that the inertial navigation system aided by time-aligned global-shutter camera has four unobservable directions: one corresponding to rotations about the gravity vector, and three to a global translations. Specifically, the system's unobservable directions with respect to the IMU pose and feature position, $\begin{bmatrix} {}^{I}\mathbf{q}_G^T & \mathbf{b}_g^T & {}^{G}\mathbf{v}_I^T & \mathbf{b}_a^T & {}^{G}\mathbf{p}_I^T & {}^{G}\mathbf{p}_f^T \end{bmatrix}^T$, can be written as:

$$\mathbf{N} \triangleq \left[\begin{array}{cc} {}^{I}_{G}\mathbf{C}\mathbf{g} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} \\ -\lfloor {}^{G}\mathbf{v}\rfloor\mathbf{g} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} \\ -\lfloor {}^{G}\mathbf{p}\rfloor\mathbf{g} & \mathbf{I}_{3\times3} \\ \hline -\lfloor {}^{G}\mathbf{p}_f\rfloor\mathbf{g} & \mathbf{I}_{3\times3} \end{array}\right] = \left[\begin{array}{c} \mathbf{N}_r \\ \hline \mathbf{N}_f \end{array}\right] \tag{32}$$

(b) Modification of the State Transition Matrix $\mathbf{\Phi}_{k+1,k}$: Once we have determined the system's unobservable directions, we start by modifying the state transition matrix, $\mathbf{\Phi}_{k+1,k}$, according to the observability constraint (30)

$$\mathbf{N}_{r_{k+1}} = \mathbf{\Phi}_{k+1,k}\mathbf{N}_{r_k} \tag{33}$$

where $\mathbf{\Phi}_{k+1,k}$ has the following structure:

$$\begin{bmatrix} \mathbf{\Phi}_{11} & \mathbf{\Phi}_{12} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{\Phi}_{31} & \mathbf{\Phi}_{32} & \mathbf{I}_3 & \mathbf{\Phi}_{34} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{\Phi}_{51} & \mathbf{\Phi}_{52} & \delta t\mathbf{I}_3 & \mathbf{\Phi}_{54} & \mathbf{I}_3 \end{bmatrix} \tag{34}$$

In [6], it is shown that (33) is equivalent to the following three constraints:

$$\mathbf{\Phi}_{11}\ {}^{I_k}_{G}\mathbf{C}\mathbf{g} = {}^{I_{k+1}}_{G}\mathbf{C}\mathbf{g} \tag{35}$$

$$\mathbf{\Phi}_{31}\ {}^{I_k}_{G}\mathbf{C}\mathbf{g} = \lfloor {}^{G}\mathbf{v}_{I_k}\rfloor\mathbf{g} - \lfloor {}^{G}\mathbf{v}_{I_{k+1}}\rfloor\mathbf{g} \tag{36}$$

$$\mathbf{\Phi}_{51}\ {}^{I_k}_{G}\mathbf{C}\mathbf{g} = \delta t\lfloor {}^{G}\mathbf{v}_{I_k}\rfloor\mathbf{g} + \lfloor {}^{G}\mathbf{p}_{I_k}\rfloor\mathbf{g} - \lfloor {}^{G}\mathbf{p}_{I_{k+1}}\rfloor\mathbf{g} \tag{37}$$

in which (35) can be easily satisfied by modifying $\mathbf{\Phi}_{11}^* = {}^{I_{k+1}}_{G}\mathbf{C}\ {}^{I_k}_{G}\mathbf{C}^T$.

Both (36) and (37) are in the form $\mathbf{A}\mathbf{u} = \mathbf{w}$, where $\mathbf{u}$ and $\mathbf{w}$ are fixed. We seek to select another matrix $\mathbf{A}^*$ that is closest to the $\mathbf{A}$ in the Frobenius norm sense, while satisfying constraints (36) and (37). To do so, we can formulate the following optimization problem:

$$\begin{aligned} \mathbf{A}^* = \underset{\mathbf{A}^*}{\operatorname{argmin}} & ||\mathbf{A}^* - \mathbf{A}||_{\mathscr{F}}^2 \\ \text{s. t.} \quad & \mathbf{A}^*\mathbf{u} = \mathbf{w} \end{aligned} \tag{38}$$

where $||\cdot||_{\mathscr{F}}$ denotes the Frobenius matrix norm. The optimal $\mathbf{A}^*$, as shown in [9], can be determined by solving its KKT optimality condition [3], whose solution is:

$$\mathbf{A}^* = \mathbf{A} - (\mathbf{A}\mathbf{u} - \mathbf{w})(\mathbf{u}^T\mathbf{u})^{-1}\mathbf{u}^T \tag{39}$$

(c) Modification of the Measurement Jacobian $\mathbf{H}_k$: During the update at time step k, the nonzero elements of the measurement Jacobian $\mathbf{H}_k$, as shown in (18), are $\begin{bmatrix} \mathbf{H}_{I_k\mathbf{q}_G} & \mathbf{H}_{G\mathbf{p}_{I_k}} & \mathbf{H}_{I_{k+1}\mathbf{q}_G} & \mathbf{H}_{G\mathbf{p}_{I_{k+1}}} & \mathbf{H}_{G\mathbf{p}_f} & \mathbf{H}_{\lambda_{d_k}} & \mathbf{H}_{\lambda_r} \end{bmatrix}$, corresponding to the elements of the state vector involved in the measurement model (as expressed by the subscript).

Since two IMU poses are involved in the interpolation-based measurement model, the system's unobservable directions at time step k, as shown in [7], are:

$$\mathbf{N}_k' \triangleq \begin{bmatrix} \mathbf{N}_{r_k}^T & \mathbf{N}_{r_{k+1}}^T & \mathbf{N}_{f_k}^T & \mathbf{0} \end{bmatrix}^T \tag{40}$$

where $\mathbf{N}_{r_i},\ i = k,\ k+1$, and $\mathbf{N}_{f_k}$ are defined in (32), while the zero corresponds to the interpolation ratio. If we define $\mathbf{N}_k' \triangleq \begin{bmatrix} \mathbf{N}_k^g & \mathbf{N}_k^p \end{bmatrix}$, where $\mathbf{N}_k^g$ is the first column of $\mathbf{N}_k'$ corresponding to the rotation about the gravity, and $\mathbf{N}_k^p$ is the other three columns corresponding to global translations, then according to (31), we seek to modify $\mathbf{H}_k$ so as to fulfill the following two constraints:

$$\mathbf{H}_k\mathbf{N}_k^p = \mathbf{0} \ \Leftrightarrow\ \mathbf{H}_{G\mathbf{p}_{I_k}} + \mathbf{H}_{G\mathbf{p}_{I_{k+1}}} + \mathbf{H}_{G\mathbf{p}_f} = \mathbf{0} \tag{41}$$

$$\mathbf{H}_k\mathbf{N}_k^g = \mathbf{0} \ \Leftrightarrow\ \begin{bmatrix} \mathbf{H}_{I_k\mathbf{q}_G} & \mathbf{H}_{G\mathbf{p}_{I_k}} & \mathbf{H}_{I_{k+1}\mathbf{q}_G} & \mathbf{H}_{G\mathbf{p}_{I_{k+1}}} & \mathbf{H}_{G\mathbf{p}_f} \end{bmatrix} \begin{bmatrix} {}^{I_k}_{G}\mathbf{C}\mathbf{g} \\ -\lfloor {}^{G}\mathbf{p}_{I_k}\rfloor\mathbf{g} \\ {}^{I_{k+1}}_{G}\mathbf{C}\mathbf{g} \\ -\lfloor {}^{G}\mathbf{p}_{I_{k+1}}\rfloor\mathbf{g} \\ -\lfloor {}^{G}\mathbf{p}_f\rfloor\mathbf{g} \end{bmatrix} = \mathbf{0} \tag{42}$$

Substituting $\mathbf{H}_{G\mathbf{p}_f}$ from (41) into (42), the observability constraint for the measurement Jacobian matrix is written as:

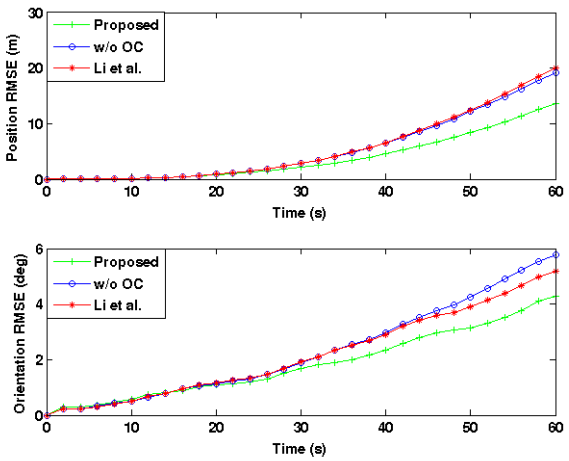


Fig. 3. Monte-Carlo simulations comparing: (a) Position RMSE (b) Orientation RMSE, over 20 runs.

$$\begin{bmatrix} \mathbf{H}_{I_k \mathbf{q}_G} & \mathbf{H}_{G\mathbf{p}_{I_k}} & \mathbf{H}_{I_{k+1}\mathbf{q}_G} & \mathbf{H}_{G\mathbf{p}_{I_{k+1}}} \end{bmatrix} \begin{bmatrix} {}^{I_k}_{G}\mathbf{C}\mathbf{g} \\ ({\lfloor}^{G}\mathbf{p}_f{\rfloor} - {\lfloor}^{G}\mathbf{p}_{I_k}{\rfloor})\mathbf{g} \\ {}^{I_{k+1}}_{G}\mathbf{C}\mathbf{g} \\ ({\lfloor}^{G}\mathbf{p}_f{\rfloor} - {\lfloor}^{G}\mathbf{p}_{I_{k+1}}{\rfloor})\mathbf{g} \end{bmatrix} = \mathbf{0} \tag{43}$$

which is of the form $\mathbf{A}\mathbf{u} = \mathbf{0}$. Therefore, we can analytically determine $\mathbf{H}^*_{I_k \mathbf{q}_G}$, $\mathbf{H}^*_{G\mathbf{p}_{I_k}}$, $\mathbf{H}^*_{I_{k+1}\mathbf{q}_G}$ and $\mathbf{H}^*_{G\mathbf{p}_{I_{k+1}}}$ using (38) and (39), for the special case when $\mathbf{w} = \mathbf{0}$. Finally, according to (41), we have $\mathbf{H}^*_{G\mathbf{p}_f} = -\mathbf{H}^*_{G\mathbf{p}_{I_k}} - \mathbf{H}^*_{G\mathbf{p}_{I_{k+1}}}$.

VI. Simulations and Experiments

A. Monte-Carlo Simulations

Our simulations involved a MEMS-quality IMU, as well as a rolling-shutter camera with a readout time of 30 msec. The time offset between the camera and the IMU clock was modelled as a random walk with mean 3.0 msec and standard deviation 1.0 msec. The IMU provided measurements at a frequency of 100 Hz, while the camera ran at 10 Hz. The sliding-window state contained 6 cloned IMU poses, while 20 features were processed during each EKF update.

We compared the following variants of the MSC-KF, over 20 Monte-Carlo runs:

- *Proposed*: The proposed OC-MSC-KF, employing an interpolation-based measurement model.
- *w/o OC*: The proposed interpolation-based MSC-KF without using OC-EKF.
- *Li et al.*: The algorithm proposed by Li et al. in [15, 16], which uses a constant velocity model, thus also clones the corresponding linear and rotational velocities, besides the cell phone pose, in the state vector.

The estimated position and orientation root-mean square errors (RMSE) are plotted in Fig. 3. By comparing *Proposed* and *w/o OC*, it is evident that employing the OC-EKF improves the position and orientation estimates. Furthermore, the proposed algorithm achieves lower RMSE compared to that of Li et al., at a significantly lower computational cost. (see Section.VI-B)

TABLE I
Loop Closure Errors

Estimation Algorithm	Final Error (m)	Pct. (%)
Proposed	1.64	0.59
w/o OC	2.16	0.79
w/o Time Sync	2.46	0.91
w/o Rolling Shutter	5.02	1.88

B. Real-World Experiments

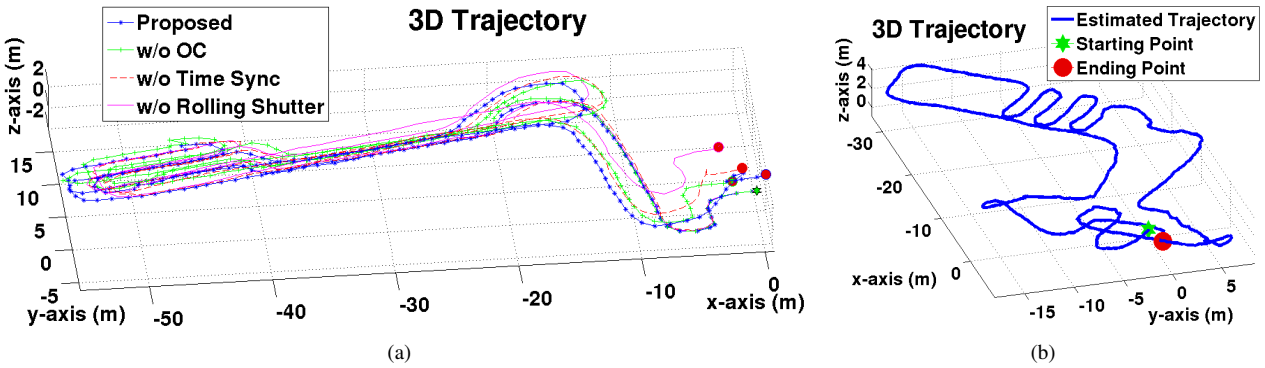
In addition to simulations, we further validated the performance of the proposed algorithm using a Samsung S4 mobile phone. The S4 is equipped with 3-axial gyroscopes and accelerometers, a rolling-shutter camera, and a 1.6 GHz quad-core Cortex-A15 ARM CPU. Camera measurements were acquired at a frequency of 15 Hz, while point features were tracked across different images via the Lucas Kanade algorithm [18]. For every 230 ms or 20 cm of displacement, new Harris corners [8] were extracted while the corresponding IMU pose was inserted in the sliding window of 10 poses, maintained by the filter. The readout time for an image is about 30 ms, and the time offset between the IMU and camera clocks is approximately 10 ms. All image-processing algorithms were optimized using ARM NEON assembly. The system we developed requires no initial calibration of the IMU biases, rolling-shutter time, or camera-IMU clock offset, as these parameters are estimated online using the approach described in Section IV. Since no high-precision ground truth is available, in the end of our experiments, we bring the cell phone back to the initial position and this allow us to examine the final position error.

We performed two experiments. The first, Fig. 4(a) serves the purpose of demonstrating the impact of not employing the OC-EKF or ignoring the time synchronization and rolling shutter effects, while the second, Fig. 4(b), demonstrates the performance of the developed system, during an online experiment.

The first experiment comprises a loop of 277 meters, with an average velocity of 1.5 m/sec. The final position errors of *Proposed*, *w/o OC*, and the following two algorithms are examined:

- *w/o Time Sync*: The proposed interpolation-based OC-MSC-KF considering only the rolling shutter, but not the time synchronization.
- *w/o Rolling Shutter*: The proposed interpolation-based OC-MSC-KF considering only the time synchronization, but not the rolling shutter.

The 3D trajectories of the cell phone estimated by the above algorithms are plotted in Fig. 4(a), and their final position errors are reported in Table. I. Several key remarks can be made. First, by utilizing the OC-EKF, the position estimation error decreases significantly (from 0.79% to 0.59%). Second, even a (relatively small) unmodeled time offset of 10 msec between the IMU and the camera clocks, results in an increase of the loop closure error from 0.59% to 0.91%. In practice, we have seen that with about 50 msec of an unmodelled



(a) (b)

Fig. 4. Experimental Results: (a) Experiment 1: The trajectory of the cell phone estimated by the algorithms under consideration. (b) Experiment 2: The trajectory of the cell phone estimated online.

time offset, the filter will diverge immediately. Third, by ignoring the rolling shutter effect, the estimation accuracy drops dramatically, since during the readout time of an image (about 30 msec), the cell phone can move even 4.5 cm, which for a scene at 3 meters from the camera, corresponds to a 2 pixel measurement noise. Finally, we have also attempted to ignore both the rolling shutter and the time synchronization, in which case the filter diverged immediately.

In the second experiment, estimation is performed online. During the trial, our cell phone traversed a path of 231 meters across two floors of a building, with an average velocity of 1.2 m/sec. This trajectory included both crowded areas and featureless scenes. The final position error was 1.8 meters, corresponding to 0.8% of the total distance travelled (see Fig. 4(b)).

C. Computational Efficiency

In order to experimentally validate the computational gains of the proposed method versus existing approaches for online time synchronization and rolling-shutter calibration [15, 16], which require augmenting the state vector with the velocities of each clone, we compared the QR decomposition of the measurement compression step in the MSC-KF for the two measurement models. Care was taken to create a representative comparison. We used the QR decomposition algorithm provided by the C++ linear algebra library Eigen, on Samsung S4. The time to perform this QR decomposition was recorded for various numbers of cloned poses, M, observing measurements of 50 features.

Similar to both algorithms, we considered a Jacobian matrix with $50(2M-3)$ rows. However, the number of columns differs significantly between the two methods. As expected, based on the computational cost of the QR factorization, $O(mn^2)$ for a matrix of size $m \times n$, our method leads to significant computational gains. As demonstrated in Fig. 5, our algorithm requires a QR factorization that is 3 times faster compared to [16]. Furthermore, since the dimension of the system is reduced to almost half through the proposed interpolation model, all the operations in the EKF update will also gain a significant speedup (i.e., a factor of 4 for the covariance update, and a factor of 2 for the number of

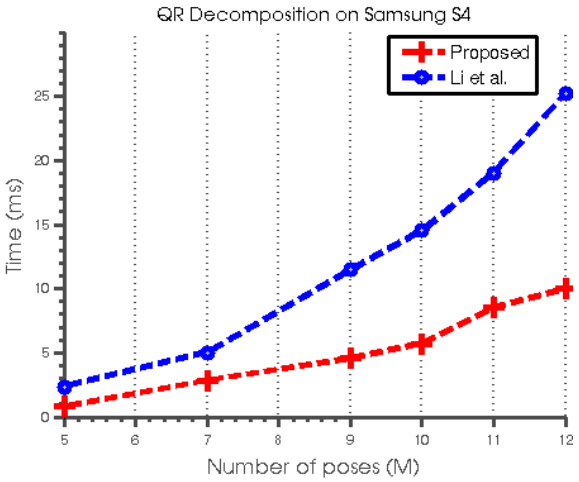


Fig. 5. Experiment: Time comparison for the measurement compression QR, employed in the MSC-KF between the proposed measurement model and the method of [15, 16].

Jacobians evaluated). Such speedup on a cell phone, which has very limited processing resources and battery, provides additional benefits, because it both allows other applications to run concurrently, and extends the phone's operating time substantially.

VII. Conclusion

In this work, we have presented a linear-complexity inertial navigation system for processing rolling-shutter camera measurements. To model the time offset of each camera row between the IMU measurements, we have proposed an interpolation-based measurement model that considers both the time synchronization effect and the image read-out time. Furthermore, we have employed the OC-EKF for improving the estimation consistency and accuracy, based on the system's observability properties. Compared to alternative methods, we have shown that the proposed approach achieves similar or better accuracy, while obtaining a significant speedup. Finally, we have demonstrated the high accuracy of the proposed algorithm through real-time, online experiments on a cell phone.

REFERENCES

[1] Omar Ait-Aider, Nicolas Andreff, Jean Marc Lavest, and Philippe Martinet. Simultaneous object pose and velocity computation using a single view from a rolling shutter camera. In *Proc. of the IEEE European Conference on Computer Vision*, pages 56–68, Graz, Austria, May 7–13 2006.

[2] Simon Baker, Eric Bennett, Sing Bing Kang, and Richard Szeliski. Removing rolling shutter wobble. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2392–2399, San Francisco, CA, June 13–18 2010.

[3] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[4] Paul Furgale, Joern Rehder, and Roland Siegwart. Unified temporal and spatial calibration for multi-sensor systems. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1280–1286, Tokyo, Japan, November 3–7 2013.

[5] Gene Golub and Charles Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[6] Chao X. Guo and Stergios I. Roumeliotis. IMU-RGBD camera 3d pose estimation and extrinsic calibration: Observability analysis and consistency improvement. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 2920–2927, Karlsruhe, Germany, May 6–10 2013.

[7] Chao X. Guo, Dimitrios G. Kottas, Ryan C. DuToit, Ahmed Ahmed, Ruipeng Li, and Stergios I. Roumeliotis. Efficient visual-inertial navigation using a rolling-shutter camera with inaccurate timestamps. Technical report, University of Minnesota, Dept. of Comp. Sci. & Eng., March 2014. URL http://www-users.cs.umn.edu/~chaguo/.

[8] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Proc. of the Alvey Vision Conference*, pages 147–151, Manchester, UK, August 31 – September 2 1988.

[9] Joel A. Hesch, Dimitrios G. Kottas, Sean L. Bowman, and Stergios I. Roumeliotis. Towards consistent vision-aided inertial navigation. In *Proc. of the 10th International Workshop on the Algorithmic Foundations of Robotics*, pages 559–574, Cambridge, Massachusetts, June 13–15 2012.

[10] Joel A. Hesch, Dimitrios G. Kottas, Sean L. Bowman, and Stergios I. Roumeliotis. Consistency analysis and improvement of vision-aided inertial navigation. *IEEE Trans. on Robotics*, 30(1):158–176, February 2014.

[11] Guoquan P. Huang, Anastasios I. Mourikis, and Stergios I. Roumeliotis. Observability-based rules for designing consistent ekf slam estimators. *International Journal of Robotics Research*, 29(5):502–528, April 2010.

[12] Chao Jia and Brian L. Evans. Probabilistic 3d motion estimation for rolling shutter video rectification from visual and inertial measurements. In *Proc. of the IEEE Internatiional Workshop on Multimedia Signal Processing*, pages 203–208, Sanff, Canada, September 2012.

[13] Jonathan Kelly and Gaurav S Sukhatme. A general framework for temporal calibration of multiple proprioceptive and exteroceptive sensors. In *Proc. of International Symposium on Experimental Robotics*, Delhi, India, December 18–21 2010.

[14] Jonathan Kelly and Gaurav S. Sukhatme. Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *International Journal of Robotics Research*, 30(1):56–79, January 2011.

[15] Mingyang Li and Anastasios I. Mourikis. 3-d motion estimation and online temporal calibration for camera-imu systems. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 5709–5716, Karlsruhe, Germany, May 6-10 2013.

[16] Mingyang Li, Byung Hyung Kim, and Anastasios I. Mourikis. Real-time motion tracking on a cellphone using inertial sensing and a rolling-shutter camera. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 4697–4704, Karlsruhe, Germany, May 6-10 2013.

[17] Bingbing Liu, Martin Adams, and Javier Ibanez-Guzman. Multi-aided inertial navigation for ground vehicles in outdoor uneven environments. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 4703 – 4708, Barcelona, Spain, April 18–22 2005.

[18] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of the International Joint Conference on Artificaial Intelligence*, pages 674–679, Vancouver, British Columbia, August 24–28 1981.

[19] Peter S. Maybeck. *Stochastic models, estimation and control*, volume 1. Academic Press, New York, NY, 1979.

[20] Anastasios I. Mourikis, Nikolas Trawny, Stergios I. Roumeliotis, Andrew E. Johson, Adnan Ansar, and Larry Matthies. Vision-aided inertial navigation for spacecraft entry, descent, and landing. *IEEE Trans. on Robotics*, 25 (2):264–280, April 2009.

[21] Luc Oth, Paul Furgale, Laurent Kneip, and Roland Siegwart. Rolling shutter camera calibration. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1360–1367, Portland, OR, June 23–28 2013.

[22] Stergios I. Roumeliotis and Joel W. Burdick. Stochastic cloning: A generalized framework for processing relative state measurements. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 1788–1795, Washington D.C., May 11-15, 2002.

[23] Ken Shoemake. Animating rotation with quaternion curves. *ACM SIGGRAPH computer graphics*, 19(3):245–254, 1985.