Spline-Based Initialization of Monocular Visual–Inertial State Estimators at High Altitude

Tianbo Liu and Shaojie Shen

Abstract—**Due to the lack of direct distance measurements, robust and accurate state estimation at high altitude but GPS-denied environments is a challenging task. A possible solution is monocular visual–inertial estimators, in which visual and inertial measurements are properly fused to recover the metric estimates. However, these estimators suffer from initialization under poor numerical conditioning or even degeneration, due to difficulties in retrieving observations of visual features with sufficient parallax, and the excessive period of inertial measurement integration. In this letter, we introduce the joint formulation into the spline-based high altitude estimator initialization method for monocular visual–inertial navigation system, in which the fitting of the spline and the alignment of visual measurements and inertial measurements are jointly optimized to recover metric estimates. The method ensures that sufficient excitation is contained in the inertial measurements when initialized, which eliminates the numerical issues. Compared with the work of Liu and Shen, the joint formulation makes our initialization method insensitive to the choice of spline parameters. Thus, the adaptivity to various environments and motions is obtained, as well as higher accuracy. The method is applicable for both loosely coupled and tightly coupled visual–inertial estimators. Extensive experiments are conducted to validate our approach.**

Index Terms—**Aerial systems: perception and autonomy, localization, sensor fusion.**

I. Introduction

THERE is an increasing demand for autonomous aerial robots in research, commercial and industrial applications, considering their advantages of mobility and agility, especially the ability to switch between hovering and fast maneuvering. A large number of applications are at high altitude, for example, surveillance, delivery and inspection. Reliable GPS and inertial measurement unit (IMU) are commonly fused to estimate states (e.g., position, attitude, velocity). However, high-quality GPS reception is not guaranteed at high altitude. For example, while operating between high-rise buildings in urban areas or in the middle of deep canyons (Fig. 1), the sky view is often obstructed, which prevents GPS receivers from locating the aerial robot precisely. Thus it is essential to develop a solution for reliable state estimation at high altitude in GPS-denied environments.

Fig. 1. Illustrations for environments where GPS signals can be poor even at high altitude, such as in canyons[1] or “city canyons”[2].

It is common to use active ranging sensors for state estimation, however they are not suitable at high altitude. For Time-of-Flight ranging sensors and RGB-D sensors, nearest objects are too far away to be detected, and radars or LiDARs with sufficient emission power are too heavy to be carried on aerial robots. Due to large scene depth, it is difficult for common passive distance measurement, such as stereo cameras, to have large and rigid baselines, which degenerate to the monocular case. Therefore, monocular visual-inertial navigation system (VINS), consists of only a camera and an IMU, becomes an attractive and reasonable choice. In order to obtain closed-loop control, the metric scale should be recovered by means of fusing visual and IMU measurements.

A good initialization is essential for every monocular VINS to be bootstrapped, and a similar procedure is also applicable to the failure recovery of the system. Parameters to be initialized are commonly velocity and attitude of the aerial robot, as well as scene depth or feature depths for certain monocular VINS. There should be appropriate motions to render the unknown parameters observable. Interestingly, for monocular VINS, visual measurements and inertial measurements have different requirements for motion.

Accurate pose estimation needs precise 3D positions of the visual features. In order to achieve good observability when acquiring 3D position through feature triangulation, there ought to be sufficient parallax between frames. It takes a long movement to satisfy the parallax requirement at high altitude, which is equivalent to long intervals between poses in a sliding window. Vision-based algorithms also favor less agile motions to benefit feature tracking for accuracy.

In contrast, IMU is good at measuring agile motions in a short period of time. We denote the IMU measurements acquired during agile motions as “*informative*” measurements. IMU measurements will degenerate and output nearly zero readings besides the gravity when the robot is hovering or moving

Manuscript received February 15, 2017; accepted July 3, 2017. Date of publication July 11, 2017; date of current version July 19, 2017. This letter was recommended for publication by Associate Editor F. Kendoul and Editor J. Roberts upon evaluation of the reviewers” comments. *(Corresponding author : Tianbo Liu.)*

The authors are with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong (e-mail: tliuam@connect.ust.hk; eeshaojie@ust.hk).

This paper has supplemental material available at http://ieeexplore.ieee.org. Video can be accessed at https://youtu.be/9gb8XLlt7YA

Digital Object Identifier 10.1109/LRA.2017.2724770

[1] http://www.0717nt.com/User/pic_tongbu/1/20140714223551.jpg

[2] http://wallpapersafari.com/w/xXm8FW/

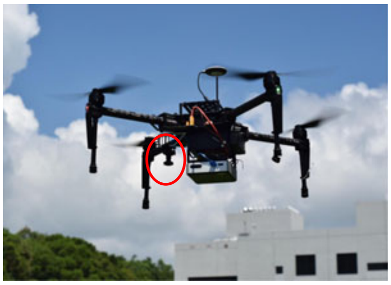


Fig. 2. The quadrotor testbed equipped with a monocular fisheye camera (marked in red), an onboard IMU, and the onboard computational device.

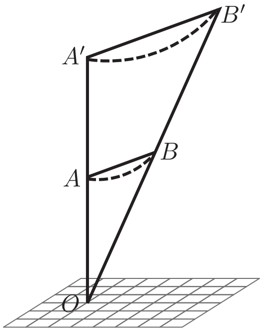


Fig. 3. Illustration for triangular relationship of visual and inertial measurements at high altitude.

at constant velocity. It is impossible to recover metric scale from these measurements, and they are named "*uninformative*" measurements.

Actually, there is a contradiction between these two motion requirements at high altitude.

As illustrated in Fig. 3, dashed curves between AB and $A'B'$ represent the window of camera poses, and solid straight lines between them are the baselines for visual algorithms. At low altitude or in indoor cases, the triangle OAB provides enough visual parallax AB w.r.t. scene depth OA, and IMU measurements inside AB are within 1 or 2 seconds. At high altitude, a similar triangle $OA'B'$ which satisfies the same parallax requirement will have a longer side $A'B'$ w.r.t. scene depth OA'. Typically, OA' will be $10 \sim 20$ times longer than OA, which implies that it takes several seconds for the platform to cover $A'B'$. On one hand, long baselines are preferred for visual algorithms, which contains an excessive number of uninformative measurements. On the other hand, to reduce the number of uninformative measurements, agile motions are required, which has negative effects on the feature tracking algorithm used in the monocular visual odometry module. The contradictory leads to the difficulty of initialization at high altitude.

While several methods are proposed for monocular VINS initialization [2]–[8], none of them focus on the high altitude case, which makes them unreliable in such circumstances. [2]–[5] do initialization based on comparing visual measurements and integrated IMU measurements, while [6]–[8] extract differentiated visual measurements to avoid double integration of IMU measurements.

The concept of spline-based initialization method is originally proposed in our previous work [1] to tackle the aforementioned issues at high altitude. Existing methods suffer from long-term double integration of noisy IMU or noisy results of differentiated visual measurements. With the help of B-spline, we take derivatives instead of integration to reduce the impact of IMU noise, and the visual measurements are naturally smoothed by B-spline fitting. Our formulation relies on an accurate monocular visual odometry (VO) to provide up-to-scale pose estimation. A B-spline is fitted with the provided pose estimates, from which the continuous up-to-scale velocity and acceleration can be extracted by taking derivatives. To solve the metric scale of VO and the direction of gravity, we align the accelerometer measurements with the corresponding accelerations extracted from the B-spline, which can be summarized as *visual-inertial alignment*.

This letter improves our previous work [1] by utilizing a joint formulation of spline fitting and visual-inertial alignment for optimization. Different from the two-step optimization in [1], the inertial measurements also contribute to the formation of the spline in the joint formulation, which makes it insensitive to the choice of spline parameters. Hence, the method is more adaptable to various environments and motions. Furthermore, the accuracy of initialization is improved, and we also address the observability of the bias and extrinsic parameters in this letter.

To verify our initialization process, recovered quantities are provided for bootstrapping either tightly-coupled or loosely-coupled visual-inertial estimators according to their demands. An extended Kalman filter (EKF)-based estimator and a multi-state constraint Kalman filter (MSCKF) are chosen to represent these two types of estimators. All the quantities required by the initialization of a certain VINS estimator are recovered and used to bootstrap the estimator. We also validate our method in a real-time scenario. A closed-loop experiment is conducted on our testbed (Fig. 2).

In Section II, we discuss the relevant literature. After the system overview in Section III, we briefly review our monocular VO in Section IV. Details and analyses for our spline-based high altitude estimator initialization are presented in Section V. The interfaces between our initialization and VINS estimators are presented in Section VI, and the implementation details and experimental results are discussed in Section VII. The letter is concluded in Section VIII.

II. Related Work

Loosely-coupled monocular visual-inertial estimators are popular among aerial robots [9]–[12]. A classical framework PTAM is proposed by [13], while a thorough discussion of monocular VO is introduced in [14]. When it comes to visual-inertial fusion, valuable explorations are conducted by [10] and [11]. Their solutions are based on the EKF framework, in which the metric scale is estimated as a state. It is mentioned in [10] that divergence can happen if the metric scale in the filter is not properly initialized.

Recently, researchers have favored a tightly-coupled formulation for monocular VINS, because the accuracy is greatly improved by directly fusing raw visual measurements. [15] fuses separate visual measurements into the EKF framework, while graph optimization-based solutions are discussed in [16]–[19]. [17] introduces an initialization scheme where velocity, attitude and bias are solved by optimization using information in a sliding window of sensor measurements. This method works under the assumption of both sufficient parallax for visual cue and sufficient IMU excitation in the window of initialization.

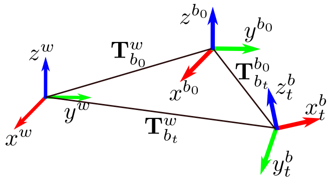


Fig. 4. Frame definition.

Several initialization techniques are explored for monocular VINS in [2]–[4], [6]–[8]. [2] proposes a closed-form formulation for solving metric scale, attitude, velocity, and bias. [3] contributes the initialization of extrinsic calibration parameters, and the advantage of initializing gyroscope bias is fully investigated in [4]. However, these methods are based on integrated inertial measurements and visual cue in the same period of time for initialization. At high altitude, large drift from long-term integration can not be avoided owing to the request for sufficient visual parallax. Thus the drift and uninformative IMU measurements in the long-term window make it difficult to achieve good performance at high altitude. In [7], a closed-form method that uses delta-velocity from visual measurements is proposed, and [8] improves the algorithm by employing a sliding least-square technique, which avoids ill-conditioning and simplifies parameter tuning. Similar to our work, the visual algorithm individually provides motion estimates up to a scale factor, which are then compared with inertial measurements to compute the scale and gravity direction. However, numerical differentiation of estimated poses induces noisy velocity measurements. This contrasts with our spline-based method, as our global spline fitting significantly reduces the impact of noise from VO.

Our work is inspired by [20], which describes a general continuous-time framework for visual-inertial simultaneous localization and mapping and calibration. We model the position trajectory of the aerial robot as a B-spline in our formulation.

III. System Overview

We first define notations and frames. $\mathbf{T}^A_B \in \mathbf{SE}(3)$ denotes a transformation from A to B. Vector $\mathbf{v}$ in B can be transformed to A through $\mathbf{T}^A_B$, expressed as $\mathbf{v}^A = \mathbf{T}^A_B \mathbf{v}^B$. A transformation is formed from rotation $\mathbf{R}$ and translation $\mathbf{p}$: $\mathbf{T} = \begin{pmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{p} & 1 \end{pmatrix}$. Roll, pitch and yaw are extracted from rotation around the x-axis, y-axis and z-axis in a $z-y-x$ sequence. World frame w is constructed by aligning gravity along the z-axis with an arbitrary yaw direction. $\mathbf{T}^w_{b_t}$ represents transformation from world w to robot body frame b at time t. We assume that the camera and the IMU are precalibrated and coincide with the body frame, such that the z-axis of the IMU is aligned with the camera optical axis. Fig. 4 shows the frame definition.

Seven modules serve the system as illustrated in Fig. 5. The autopilot provides IMU data and receives low-level control commands. The raw image sequence from the fisheye camera is pre-stabilized through gyroscope-assisted electronic image stabilization (EIS), as described in [1], to reduce rotational components.

A monocular visual odometry takes stabilized images as input and provides up-to-scale pose estimation of the camera.

The output of monocular VO is used for high altitude initialization. This module fits the up-to-scale position from monocular VO as a smooth B-spline, meanwhile aligns the second-order derivative of the B-spline with metric acceleration. By minimizing the fitting and alignment error, we can recover the metric scale and the gravity vector. The byproduct is the recovery of the metric velocity and scene depth, even positions of feature points.

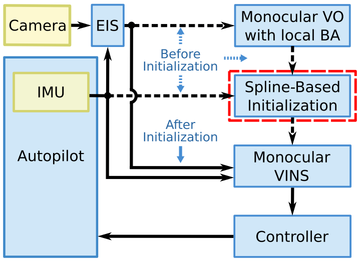


Fig. 5. System architecture. Red box highlights our main contribution on initialization.

Recovered quantities from initialization are provided for an EKF-based loosely-coupled estimator and an MSCKF to help bootstrap them. Estimation results from MSCKF are utilized to close the control loop for high altitude autonomous navigation.

IV. Monocular Visual Odometry With Local Bundle Adjustment

Our spline-based initialization approach (Section V) relies on a monocular VO providing high-accuracy up-to-scale pose estimates. In high altitude scenes, only features that are long-term tracked can be triangulated precisely. As such, we compensate the rotational components by applying EIS to all fisheye images. We adopt a VO pipeline similar to PTAM [13], where the camera poses are obtained through 2D-3D pose estimation, and 3D features are triangulated conditioned on available pose estimates. A local sliding window bundle adjustment is incorporated to refine the results. With sufficient parallax between the first two key-frames, the monocular VO is initialized by either five-point algorithm or homography. The interested reader is referred to our previous work [1] for the details on monocular visual odometry and EIS.

V. Spline-Based High Altitude Estimator Initialization

A novel high altitude initialization method for monocular VINS is detailed in this section. Failure recovery can be solved in the same way, as it is just a re-initialization of the system on-the-fly. The following states should be initialized: the scale parameter that turns the poses from monocular VO to metric, gravity-aligned attitude and velocity. The accelerometer bias term is modeled but not implemented in practice, which is explained in Section V-E

Two key attributes lie in our initialization procedure: First, we use a spline-based formulation to propagate informative accelerometer measurements through the whole initialization process. This prevents the uninformative measurements from bringing the system to a poorer numerical conditioning. Second, the smoothness property of spline helps to reduce the impact of noise brought by the sensors and monocular visual odometry estimates.

The initialization process can be formulated into two aspects: a) fitting the up-to-scale position from VO into a B-spline, and

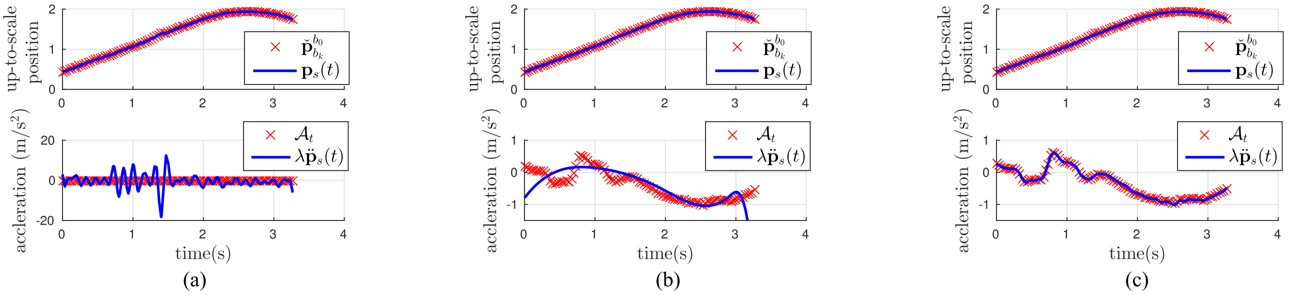


Fig. 6. Illustration for comparison between two-step and joint formulation (Section V-C). $\check{\mathbf{p}}^{b_0}_{b_k}$ and $\mathcal{A}_t$ are the data terms from VO and IMU. $\mathbf{p}_s(t)$ and $\lambda\ddot{\mathbf{p}}_s(t)$ are the fitted spline and its rescaled second-order derivative. Only the x-axis is plotted. (a) is the result of the two-step formulation, with knots interval $\Delta t = 0.1$s, all 51 knots. (b) is the result of the two-step formulation, with knots interval $\Delta t = 1.0$s, all 5 knots. (c) is the result of the joint formulation, with knots interval $\Delta t = 0.1$s, all 51 knots. The joint formulation recovers acceleration more accurately and, meanwhile, avoids over-fitting, which is proved to be superior to the two-step formulation. (a) Two-step formulation, over-fitting. (b) Two-step formulation, under-fitting. (c) Joint formulation, properly aligned.

b) aligning accelerometer readings with respect to the second-order derivative of the B-spline to solve for the scale and attitude.

These two aspects can be optimized in a two-step way as described in Section V-A, which makes the result sensitive to the choice of the intervals between the knots of the B-spline. To eliminate the sensitivity, the joint optimization for spline fitting and visual-inertial alignment is introduced in Section V-B, followed by the analyses and implementation details.

A. Fitting and Alignment in a Two-Step Formulation

Here we briefly review the two-step formulation of the initialization process proposed in [1]. We model robot position as a 6-order B-spline $\mathbf{p}_s(t) \in \mathbb{R}^3$ according to the dynamic of the aerial robot:

$$\mathbf{p}_s(t) = \sum_{i=0}^{n} \boldsymbol{\alpha}_i B_{i,d}(t), \tag{1}$$

where $t \in [t_0, t_n]$ represents the time in the period of initialization. $t_i \in [t_0, t_n]$ is the time for the ith knot of the spline. $\boldsymbol{\alpha}_i \in \mathbb{R}^3$ are the control points for the ith knot, and $B_{i,d(t)}$ are the dth-order basis functions. A least square problem is solved to fit the B-spline:

$$\min_{\boldsymbol{\alpha}_i} \frac{1}{2} \left\| \check{\mathbf{p}}^{b_0}_{t_j} - \mathbf{p}_s(t_j) \right\|_2, \tag{2}$$

where up-to-scale position estimates from the VO are denoted as $\check{\mathbf{p}}^{b_0}_{t_j}$. b_0 is the reference frame for the VO. $t_j \in [t_0, t_n]$ is the time for the jth pose, and $\|\cdot\|_2$ is the L^2 norm.

Let the linear acceleration in the platform body frame be $\mathbf{a}_t$, and $\mathbf{b}_a$ is the slow varying bias term. The noisy accelerometer measurement is $\check{\mathbf{a}}_t = \mathbf{a}_t + \mathbf{b}_a + \mathbf{n}_{a_t}$, where $t \in [t_0, t_n]$ is the time for the measurement, and $\mathbf{n}_{a_t}$ is the additive Gaussian noise. Ignoring the bias term $\mathbf{b}_a$, and aligning the acceleration w.r.t. the second-order derivative of the fitted spline, we solve the following least-square problem,

$$\min_{\lambda, \mathbf{g}^{b_0}} \sum_{t_k} \left\| \lambda \ddot{\mathbf{p}}_s(t_k) + \mathbf{g}^{b_0} - \check{\mathbf{R}}^{b_0}_{b_k} \check{\mathbf{a}}_{t_k} \right\|_2, \tag{3}$$

to recover the unknown scale parameter λ and gravity vector represented in b_0-frame $\mathbf{g}^{b_0}$. $\check{\mathbf{R}}^{b_0}_{b_k}$ is interpolated from VO poses, and t_k is the time for the kth data entry.

This two-step formulation is sensitive to the choice of intervals between knots, which motivates us to develop the joint formulation.

B. Fitting and Alignment in a Joint Formulation

In this section, we will explain how to turn the two-step formulation into a joint one, and include the estimation of IMU bias.

Due to the property of B-spline, once the knots are determined, basis functions and their derivatives are able to be evaluated at any point in the domain. Thus, the fitting problem (2) can be reformulated as

$$\min_{\boldsymbol{\alpha}} \frac{1}{2} \left\| \mathbf{A}\boldsymbol{\alpha} - \begin{bmatrix} \cdots & \check{\mathbf{p}}^{b_0 T}_{t_j} & \cdots \end{bmatrix}^T \right\|_2. \tag{4}$$

where $\mathbf{A}$ is a M by N matrix with elements being evaluation of the basis functions at all VO times t_j. M is the number of fitted data entries, and N is the total number of coefficients in all the basis functions.

Similarly, we stack the evaluation of the second-order derivatives of all the basis functions at all IMU times t_k as a K by N matrix $\ddot{\mathbf{A}}$ where K is the number of alignment data entries. Rewrite (3) in a stacked version similar to (4), and substitute $\ddot{\mathbf{p}}_s(t_k)$ with $\ddot{\mathbf{A}}\boldsymbol{\alpha}$ in it, then we can get

$$\min_{\lambda, \mathbf{g}^{b_0}, \boldsymbol{\alpha}, \mathbf{b}_a} \frac{1}{2} \left\| \ddot{\mathbf{A}}\boldsymbol{\alpha} + \begin{bmatrix} \vdots & \vdots & \vdots \\ \lambda \mathbf{g}^{b_0} & \lambda \check{\mathbf{R}}^{b_0}_{b_k} \mathbf{b}_a & -\lambda \check{\mathbf{R}}^{b_0}_{b_k} \check{\mathbf{a}}_{t_k} \\ \vdots & \vdots & \vdots \end{bmatrix} \right\|_2. \tag{5}$$

Note that, compared with (4), we change where the scaling factor λ multiplies, and incorporate the bias term into the formulation.

Combining (4) and (5) together, we get a joint nonlinear optimization problem:

$$\begin{aligned} \min_{\lambda, \mathbf{g}^{b_0}, \boldsymbol{\alpha}, \mathbf{b}_a} & \frac{1}{2} \left\| \mathbf{A}\boldsymbol{\alpha} - \begin{bmatrix} \cdots & \check{\mathbf{p}}^{b_0 T}_{t_j} & \cdots \end{bmatrix}^T \right\|_2 \\ & + w \cdot \frac{1}{2} \left\| \ddot{\mathbf{A}}\boldsymbol{\alpha} + \begin{bmatrix} \vdots & \vdots & \vdots \\ \lambda \mathbf{g}^{b_0} & \lambda \check{\mathbf{R}}^{b_0}_{b_k} \mathbf{b}_a & \lambda \check{\mathbf{R}}^{b_0}_{b_k} \check{\mathbf{a}}_{t_k} \\ \vdots & \vdots & \vdots \end{bmatrix} \right\|_2 \end{aligned} \tag{6}$$

where w denotes the trade-off parameter between fitting and alignment. Details for solving this problem is presented in Section V-D.

C. Comparison Between the Two Formulations

To be concise, we denote $\mathcal{A}_t = \mathbf{R}_{b_t}^{b_0}(\check{\mathbf{a}}_t - \mathbf{b}_a) - \mathbf{g}^{b_0}$ in the following descriptions.

In the two-step formulation described in Section V-A, we find that the result is sensitive to the interval between knots. The density of knots directly controls the degree of freedom (DOF) of the spline, i.e., the ability to fit and align. With dense knots, the spline is properly fitted, but the second-order derivative of the fitted spline tends to over-fit. In contrast, a sparse knots configuration leads to under-fitting. These two situations are illustrated in Fig. 6.

We eliminate this trade-off by utilizing the joint optimization detailed in Section V-B. IMU measurements directly constrain the spline from over-fitting through the $\ddot{\mathbf{A}}\boldsymbol{\alpha}$ component in (6), which is different from only position constraints in (2). Thanks to the inclusion of IMU constraints, a dense configuration of knots no longer makes the spline over-fit, which improves the precision of both fitting and alignment.

Fig. 6 illustrates the comparison. For the same period of data, the two-step formulation cannot balance between under-fitting and over-fitting. Fig. 6(a) shows over-fitting on the acceleration with a dense knots configuration. Fig. 6(b) shows under-fitting on the acceleration due to the lack of DOF with a sparse knots configuration. In contrast, the joint formulation is capable of suppressing over-fitting brought by a dense knots configuration, as shown in Fig. 6(c). The accuracy of the alignment is drastically improved in the joint formulation.

Owing to the improvement brought by the joint optimization, we can fix the interval for the uniformly distributed knots to 0.1s in the joint formulation, and there is no need to tune this parameter.

D. Implementation Details for Initialization

The bias of gyroscope and the extrinsics between camera and IMU can be pre-calibrated. The norm of gravity G_m is set according to the local gravity and the accelerometer scale factor.

Data for at least 2 seconds is required to bootstrap the initialization. We define $\|\check{\mathbf{a}}_t^{b_0} - \check{\mathbf{a}}_{\text{avg}}^{b_0}\|_2 >= \epsilon_{\mathbf{g}}$ as the criteria for informative measurements, where $\epsilon_{\mathbf{g}}$ is a tunable threshold. $\check{\mathbf{a}}_t^{b_0}$ is acceleration reading projected to b_0 frame, and $\check{\mathbf{a}}_{\text{avg}}^{b_0}$ is the average in $[t_0, t_n]$. We set $\epsilon_{\mathbf{g}} = 0.2$ m/s^2. Only when informative measurements exceed a certain number (typically 200 for a 100 Hz IMU), is a trail for initialization conducted.

During the solving process, we first solve the linear system (2) in a dense knot setting to obtain initial guess of control points. Initial values for other parameters are set as $\mathbf{g}^{b_0} = G_m[0\ 0\ 1]^T$, $\lambda = 0.01$. As illustrated in Fig. 7, the chosen initial value will not affect the final results. The nonlinear system (6) is then solved using Gauss-Newton method. Note that the initial guess from (2) may subject to over-fitting, but the inclusion of IMU measurements in (6) effectively regularizes the spline to avoid over-fitting. We use a similar technique in [21] to optimize $\mathbf{g}^{b_0}$ on its spherical manifold, remaining its norm unchanged. The weight parameter w can be set around 1.0, and we are able to obtain good results.

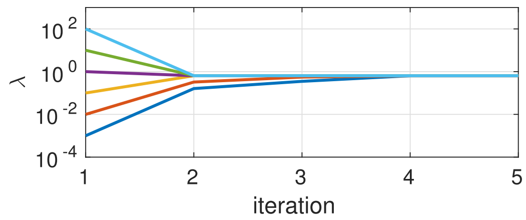


Fig. 7. Consistent results of scale factor λ are achieved from different initial values. Inappropriate initial value only costs more iterations, and does not affect the final result. The test is conducted on the same dataset as Section VII-B.

Thanks to the sparsity of B-spline, (6) can be formulated in a sparse way, which bounds the complexity of our algorithm to be nearly linear. On our experimental platform described in Section VII, it consumes several milliseconds for 2 seconds of data, and only 35 ms for 10 seconds of data.

Once the solution for (6) is available, we verify it by calculating the alignment error, and define $\eta = \|\ddot{\mathbf{p}}_s(k) - \lambda\hat{\mathcal{A}}_t\|_2 / \|\lambda\hat{\mathcal{A}}_t\|_2$ as the error percentage for one entry. A solution is dropped if the averaged error percentage is larger than 20%. New trials can immediately begin when a new VO estimate comes, due to the low time consumption in each trial.

Attitude and velocity are able to be recovered with the verified solution, which is detailed in [1].

Till now, we have initialized the attitude $\hat{\mathbf{R}}_{b_0}^{w}$, velocity $\hat{\mathbf{v}}_{\text{init}}^{w}$ and scale $\hat{\lambda}$ for monocular VO. Additionally, the depths of the feature points from monocular VO can also be recovered by simply applying the scale $\hat{\lambda}$.

E. Discussion on Extrinsics and Bias

We first discuss the extrinsics of the sensor suite. $\mathbf{R}_c^b$, the rotation between camera and IMU, can be extracted according to the method proposed in [17].

From the hardware configuration, we can get initial guess for $\mathbf{p}_c^b$ under 1 cm precision. Assume that we use a camera with 640 pixels covering 90° field of view. With this camera from 50 meters high, 1 cm baseline only achieves 0.064 disparity in pixel, and 1 pixel disparity requires 15 cm baseline. Thus, centimeter level error of $\mathbf{p}_c^b$ can hardly be observed from the visual measurements. As a result, we decide to ignore the extrinsics for our high altitude initialization method.

Thus without extrinsics, we effectively minimize the alignment error during optimization.

$$\mathbf{e}_a = \ddot{\mathbf{p}}_s(k) - \lambda(\check{\mathbf{R}}_{b_k}^{b_0}(\check{\mathbf{a}}_k - \mathbf{b}_a) - \mathbf{g}^{b_0}). \tag{7}$$

Then we discuss the bias of IMU. Since the bias of gyroscope can be easily determined in a stationary condition, and we do not use the gyroscope in our formulation, we focus on the observability of the accelerometer bias $\mathbf{b}_a$. From (7), there is only $\check{\mathbf{R}}_{b_k}^{b_0}$ between unknowns $\mathbf{b}_a$ and $\mathbf{g}^{b_0}$, meaning that sufficient rotation excitation is required to distinguish between them. We process our initialization on two datasets to analyze the observability of the bias term. The first dataset contains fast motion and large rotation, with roll and pitch angle varying between ±35°, while the second one has slow motion and small rotation, with roll and pitch angle varying between ±10°. The yaw angles for both datasets remain unchanged. Solving (6) leads to an equation in the form of $\mathbf{J}^T\mathbf{J}\delta\mathbf{x} = \mathbf{J}^T\mathbf{r}$, and we extract the Jacobian matrices $\mathbf{J}$ from initialization on these two datasets. Note that we only

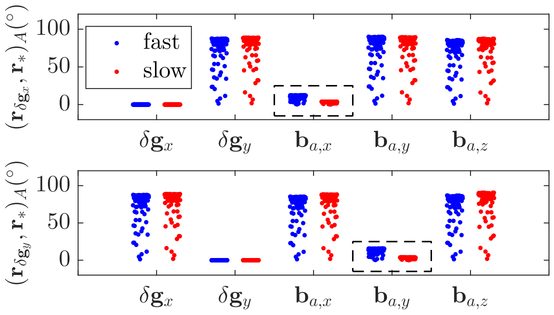


Fig. 8. Observability illustration for bias term. As highlighted in black dashed boxes, the angles between residual vectors perturbed by $\delta\mathbf{g}_x$ and $\mathbf{b}_{a,x}$ (upper) or $\delta\mathbf{g}_y$ and $\mathbf{b}_{a,y}$ (lower) are small, which means difficulty distinguishing between them. Blue and red dots represent results from fast and slow motion dataset respectively. The detailed discussion is in Section V-E.

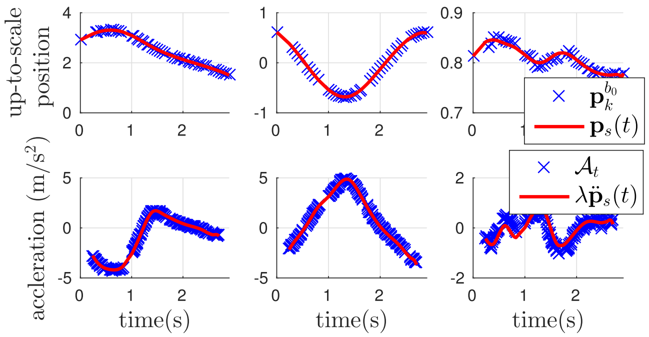
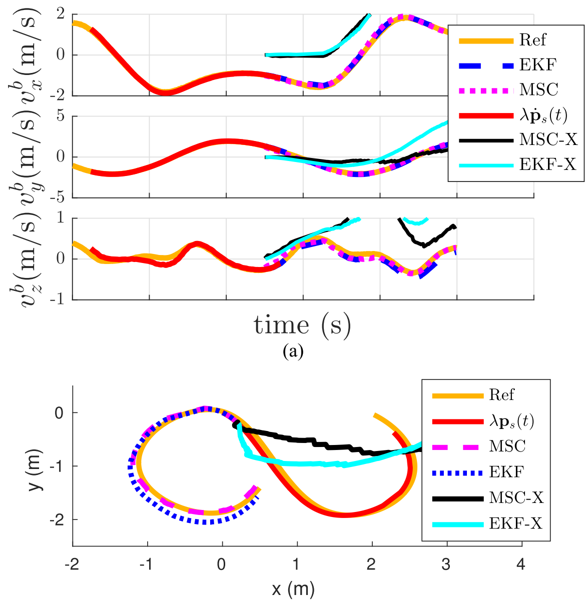


Fig. 9. Result of fitting (top) and alignment (bottom) for the indoor environment. Both position and acceleration are fitted and aligned well in the experiment.

optimize on the manifold of the gravity as described in [21]; i.e., the parameters are error states $\delta\mathbf{g}_x$ and $\delta\mathbf{g}_y$ on its yaw-aligned tangent space.

We get the perturbed residual vectors $\mathbf{r}_p$ by first assigning a small perturbation to the elements in $\delta\mathbf{x}$ which corresponds to $\delta\mathbf{g}_x$, $\delta\mathbf{g}_y$, $\mathbf{b}_{a,x}$, $\mathbf{b}_{a,y}$ and $\mathbf{b}_{a,z}$ respectively to get $\delta\mathbf{x}_p$, and then applying $\mathbf{r}_p = \mathbf{J}\delta\mathbf{x}_p$. Ideally, we should get several perturbed residual vectors which are "far away" from each other because different unknowns should make different impacts. To measure the *similarity* between two vectors $\mathbf{r}_1$ and $\mathbf{r}_2$, we define: $(\mathbf{r}_1, \mathbf{r}_2)_A = \mathrm{acos}(\frac{\mathbf{r}_1^T}{\|\mathbf{r}_1\|_2} \cdot \frac{\mathbf{r}_2}{\|\mathbf{r}_2\|_2})$, which is intuitively the angle between the two vectors in the high-dimensional space. A value near 90° means that the two vectors are very different, and 0° means nearly the same.

The following descriptions show the difficulty of estimating the bias. Applying random perturbations on both fast and slow motion dataset for 100 times, the angles between the residual vector perturbed by $\delta\mathbf{g}_x$ and that perturbed by the other terms are accumulatively plotted in the first row of Fig. 8. Similar plotting for $\delta\mathbf{g}_y$ is given in the second row. As illustrated, perturbation applied on $\delta\mathbf{g}_x$ and other terms besides $\mathbf{b}_{a,x}$ lead to different residual vectors, since the angles between them are mostly 90°. They make different impacts on the system. However, for the $\mathbf{b}_{a,x}$ case, the angle is only 15° for fast motion dataset (in blue), and even worse in slow motion dataset (in red). It is clear that $\delta\mathbf{g}_x$ and $\mathbf{b}_{a,x}$ make similar impacts on the system. This coincides with our previous analysis that only when there is enough rotation excitation can the bias and gravity be distinguished. The $\delta\mathbf{g}_y$ case is similar as shown in the second row of Fig. 8. Solving for parameters that are so close as such might cause numerical issues or ambiguity.



(a)

(b)

Fig. 10. (a) Velocity estimates and (b) position estimates from various algorithms. Estimators bootstrapped by our initialization (MSCKF, EKF) and the initialization itself (red) fit the reference (orange), while naively initialized estimators (EKF-X, MSC-X) fail to converge. The position estimates are manually aligned at the point of bootstrapping for plotting. Refer to Section VII-B for details.

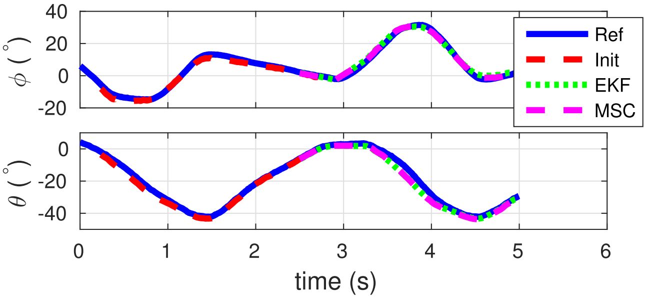


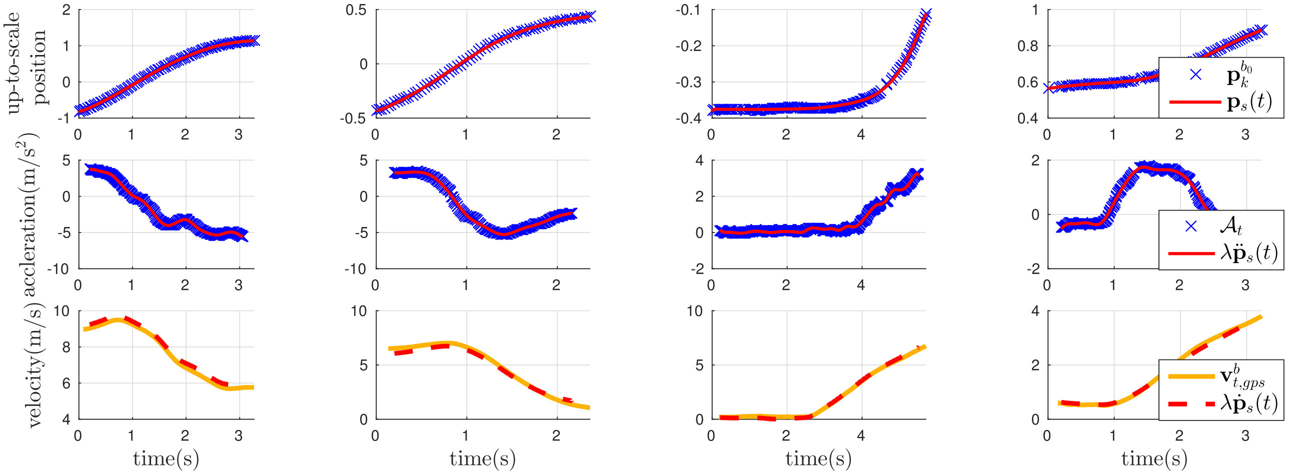
Fig. 11. Roll (θ) and pitch (ϕ) angles from various algorithms. All of them match the reference (in blue) well.

The results convince us that, in practice, we should not incorporate the bias term during initialization, since rotation excitation is not guaranteed and has negative effects on vision-based algorithms. Moreover, accelerometers are usually biased by less than 0.5 m/s^2, which is much smaller than the gravity. So the bias term has minor effect on the result.

At high altitude, the effect of bias and extrinsics can hardly be observed in the short-term initialization phase, and also, these quantities will not affect the bootstrapping of monocular VINS. Therefore we do not incorporate them into our formulation. If necessary, these quantities can be recovered by long-term estimating from monocular VINS.

VI. Bootstrapping Visual-Inertial Estimators

Initialized states obtained from the initialization module described in Section V can be used to bootstrap both



(a) (b) (c) (d)

Fig. 12. Results of the initialization method at high altitude. We use fused GPS velocity as the reference. (a)–(d) are the results of four trials with randomly chosen start time. Only one of the three axes that contains the major motion is plotted for each trial. They all successfully recover the required quantities and bootstrap the successive estimators. Refer to Section VII-C for details. (a) 1st trial at 50 m, y-axis. (b) 2 nd trial at 50 m, y-axis. (c) 3rd trial at 90 m, x-axis. (d) 4th trial at 90 m, y-axis.

tightly-coupled or loosely-coupled VINS estimators. To verify our initialization, we implement both an EKF-based loosely-coupled visual-inertial estimator described in [1] and an MSCKF proposed in [15].

The initial attitude $\hat{\mathbf{R}}_{b_0}^w$, scale $\hat{\lambda}$ and velocity $\hat{\mathbf{v}}_{\text{init}}^w$ are used for bootstrapping the loosely-coupled estimator as the initial values for the filter states. The MSCKF and our monocular VO share the same front-end. $\hat{\mathbf{R}}_{b_0}^w$, $\hat{\mathbf{v}}_{\text{init}}^w$ and metric depths for features tracked by the front-end serve as the initial values for the MSCKF algorithm.

These filters could easily diverge if they were to be bootstrapped from incorrect initial values. Our initialization method is verified through them since they are both bootstrapped successfully in all test cases, including slow and fast motion in both indoor and outdoor cases, which are shown in Section VII.

VII. Experimental Results

A. System Implementation Details

A DJI M100[3] is employed for real-time closed-loop control experiment. All algorithms run onboard an Intel NUC computer (i5-4250U, 16GB RAM). The visual sensor is an mvBlueFOX-MLC200wG equipped with a 210° fisheye lens. The IMU is acquired from the autopilot on M100 through the Onboard-SDK.

All modules are written in C++, using ROS[4] as the communication middleware. B-spline module in GSL[5] is used. The image sequence is at 30 Hz, and IMU measurements from the DJI autopilot are at 100 Hz.

B. Quantitative Performance of the Initialization Algorithm

In high altitude environments, the ground-truth is difficult to obtain, and the GPS is not accurate enough to serve as the ground-truth; thus we conduct the quantitative evaluation indoor. The ground-truth (labeled as "Ref" in the figures) comes from the OptiTrack motion capture system. We provide results from the estimators, which are bootstrapped by a naive method in which the initial values for body velocities are set as zero and initial attitude is directly extracted from treating a short-term averaging of the accelerator readings as the gravity direction.

Fig. 9 shows the result of fitting and alignment. The fitting error is defined as $\epsilon_p = \check{\mathbf{p}}_t^{b_0} - \mathbf{p}_s(t)$ from (2). Standard deviation (STD) of the fitting error is 0.0024,0.0023,0.0061 m respectively for each axis. Note that these numbers are under a scaling factor of $\lambda = 1.63$. Alignment error is defined similarly as $\epsilon_a = \ddot{\mathbf{p}}_s(k) - \lambda\hat{\mathcal{A}}_t$, of which STDs are 0.09, 0.09, 0.22 m/s^2, and the error percentage η is 10.6%.

Velocities from different sources are plotted in Fig. 10(a). Define velocity error in body frame as $\epsilon_v = \mathbf{R}_{t_k}^b \lambda\dot{\mathbf{p}}_s(k) - \mathbf{v}_{k,\text{ref}}^b$, where $\mathbf{v}_{k,\text{ref}}^b$ are extracted from the the OptiTrack system. The STDs of ϵ_v for initialization part are 0.04, 0.02, 0.08 m/s.

Meanwhile, we discover from Fig. 10(a) that, after being bootstrapped by our initialization method, both EKF and MSCKF recover the body velocity well w.r.t. the ground-truth. In contrast, without our initialization, neither of them converges. The same situation can be observed in Fig. 10(b), in which the position estimates from different algorithms are plotted.

The evaluation for attitude is plotted in Fig. 11. Pitch and roll angles are recovered under the precision of 5°, which enables bootstrapping the estimators from an appropriate initial state.

C. Performance of Initialization at High Altitude

In order to show the "launch anywhere" capability of our algorithm, we run our initialization at different heights and in various scenarios with the same parameter settings. Fig. 12 illustrates the results of several trials at high altitude. For each trial, we only plot one of the three axes that contains the major motion due to space limitation. Fig. 12(a) and (b) are two trials at about 50 meters high. Fig. 12(c) and (d) are another two trials at about 90 meters.

[3] https://developer.dji.com/matrice-100/
[4] http://www.ros.org/
[5] http://www.gnu.org/software/gsl/

In all of them, the scale between the monocular VO and metric measurements are able to be recovered. The first-order derivative multiplied by the scale is transformed to body frame for comparison with the metric body velocities retrieved from the fusion of IMU and GPS (labeled as "Ref" in the figures). The plots show that the velocities match well, which illustrates that the initialization is able to recover the scale at high altitude.

We would like to highlight that in Fig. 12(c), our method takes a longer period of time to initialize than in other cases, because the aerial robot stays nearly stationary for a period of 3 seconds, which provides no informative measurements for our algorithm. Upon there being enough informative measurements after the 4th second, the initialization process finishes successfully.

Results of these 4 trials are included in the attached video as well as other 9 trials, with the success rate 100% through all 13 trials. Additionally we can observe in the video that naively bootstrapped estimators easily diverge, while estimators bootstrapped by our method quickly converge.

D. Closed-Loop Control Experiment

We conduct a closed-loop control experiment outdoor using fused odometry from MSCKF as the feedback. All modules run onboard and the initialization is at about 60 meters high. The experiment verifies our method in realistic real-time systems. Due to the lack of ground truth benchmarking equipment, we only show empirical results in the attached video.

VIII. Conclusion

In this letter, we present a spline-based initialization method for monocular visual-inertial state estimators at high altitude. Compared with our preliminary work [1], the joint formulation makes our initialization method insensitive to the choice of spline parameters. Thus the adaptivity to various environments and motions is obtained, as well as higher accuracy. Properties and performance of the initialization method are also analyzed in detail, which helps to give a thorough comprehension of it. Experimental results justify the practicability and usefulness of our method.

Acknowledgment

The authors would like to thank the studentship and equipment support from DJI.

References

[1] T. Liu and S. Shen, "High altitude monocular visual-inertial state estimation: Initialization and sensor fusion," in *Proc. IEEE Int. Conf. Robot. Autom.*, Jun. 2017. [Online]. Available: https://drive.google.com/open?id=0By08HNZRtOAYdy15M3BMN3pkQ0U

[2] A. Martinelli, "Closed-form solution of visual-inertial structure from motion," *Int. J. Comput. Vis.*, vol. 106, no. 2, pp. 138–152, 2014.

[3] T. C. Dong-Si and A. I. Mourikis, "Estimator initialization in vision-aided inertial navigation with unknown camera-IMU calibration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 1064–1071.

[4] J. Kaiser, A. Martinelli, F. Fontana, and D. Scaramuzza, "Simultaneous state initialization and gyroscope bias calibration in visual inertial aided navigation," *IEEE Robot. Autom. Lett.*, vol. 2, no. 1, pp. 18–25, Jan. 2017.

[5] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Initialization-free monocular visual-inertial estimation with application to autonomous MAVs," in *Proc. Int. Symp. Exp. Robot.*, 2016, pp. 211–227, doi:10.1007/978-3-319-23778-7_15.

[6] L. Kneip, A. Martinelli, S. Weiss, D. Scaramuzza, and R. Siegwart, "Closed-form solution for absolute scale velocity determination combining inertial measurements and a single feature correspondence," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 4546–4553.

[7] L. Kneip, S. Weiss, and R. Siegwart, "Deterministic initialization of metric state estimation filters for loosely-coupled monocular vision-inertial systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2011, pp. 2235–2241.

[8] V. Lippiello and R. Mebarki, "Closed-form solution for absolute scale velocity estimation using visual and inertial data with a sliding least-squares estimation," in *Proc. 21st Mediterranean Conf. Control Autom.*, Jun. 2013, pp. 1261–1266.

[9] K. Jonghyuk and S. Salah, "Real-time implementation of airborne inertial-SLAM," *Robot. Auton. Syst.*, vol. 55, no. 1, pp. 62–71, 2007.

[10] S. Weiss and R. Siegwart, "Real-time metric state estimation for modular vision-inertial systems," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 4531–4537.

[11] 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. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 3923–3929.

[12] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Multi-sensor fusion for robust autonomous flight in indoor and outdoor environments with a rotorcraft MAV," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2014, pp. 4974–4981.

[13] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality*, Nov. 2007, pp. 225–234.

[14] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *IEEE Robot. Autom. Mag.*, vol. 18, no. 4, pp. 80–92, Dec. 2011.

[15] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 10–14, 2007, pp. 3565–3572.

[16] V. Indelman, S. Williams, M. Kaess, and F. Dellaert, "Information fusion in navigation systems via factor graph based incremental smoothing," *Robot. Auton. Syst.*, vol. 61, no. 8, pp. 721–738, 2013.

[17] Z. Yang and S. Shen, "Monocular visual-inertial state estimation with online initialization and camera-IMU extrinsic calibration," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 39–51, Jan. 2017.

[18] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 314–334, 2015.

[19] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation," in *Proc. Robot., Sci. Syst.*, 2015 [Online]. Available: http://www.roboticsproceedings.org/rss11/p06.html

[20] S. Lovegrove, A. Patron-Perez, and G. Sibley, "Spline fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras," in *Proc. Brit. Mach. Vis. Conf.*, pp. 93.1–93.12, 2013. [Online]. Available: link: http://www.bmva.org/bmvc/2013/Papers/paper0093/

[21] Y. Ling and S. Shen, "High-precision online markerless stereo extrinsic calibration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2016, pp. 1771–1778.