A Lightweight and Accurate Localization Algorithm Using Multiple Inertial Measurement Units

Ming Zhang, Xiangyu Xu, Yiming Chen, and Mingyang Li

***Abstract*—This paper proposes a novel inertial-aided localization approach by fusing information from multiple inertial measurement units (IMUs) and exteroceptive sensors. IMU is a low-cost motion sensor which provides measurements on angular velocity and gravity compensated linear acceleration of a moving platform, and widely used in modern localization systems. To date, most existing inertial-aided localization methods exploit only one single IMU. While the single-IMU localization yields acceptable accuracy and robustness for different use cases, the overall performance can be further improved by using *multiple* IMUs. To this end, we propose a lightweight and accurate algorithm for fusing measurements from multiple IMUs and exteroceptive sensors, which is able to obtain noticeable performance gain without incurring additional computational cost. To achieve this, we first probabilistically map measurements from *all* IMUs onto a *virtual* IMU. This step is performed by stochastic estimation with least-square estimators and probabilistic marginalization of inter-IMU rotational accelerations. Subsequently, the propagation model for both state and error state of the virtual IMU is also derived, which enables the use of the classical filter-based or optimization-based sensor fusion algorithms for localization. Finally, results from both simulation and real-world tests are provided, which demonstrate that the proposed algorithm outperforms competing algorithms by noticeable margins.**

***Index Terms*—Sensor Fusion, Localization, SLAM.**

I. Introduction

In RECENT years, commercial products which exploit inertial measurement units (IMUs) have been under fast development. This popular motion sensor can be found in robotics, personal electronic devices, wearable devices, and so on [1]. On one hand, the maturity of MEMS manufacturing process significantly reduces the size, price, and power consumption of the IMU hardware. On the other hand, significant progress has also been made in both algorithm and software design for IMUs, ranging from sensor characterization and calibration [2]–[5], measurement integration [6]–[8], sensor fusion [9]–[13], and so on.

In this work, we focus on the inertial-aided localization, which is to estimate the 6D poses (3D position and 3D orientation) of a moving platform. Since localization with only IMU inevitably suffers from pose drift, measurements from other sensors (i.e. aiding), e.g., RGB cameras, depth cameras, or LiDARs (Light Detection And Ranging sensors), are typically used in combination with IMUs to provide long-term performance guarantees [12], [14], [15]. To perform accurate pose estimation, the majority of existing works use measurements from IMU for pose prediction, which is followed by probabilistic refinement using measurements from other sensors [7]–[12].

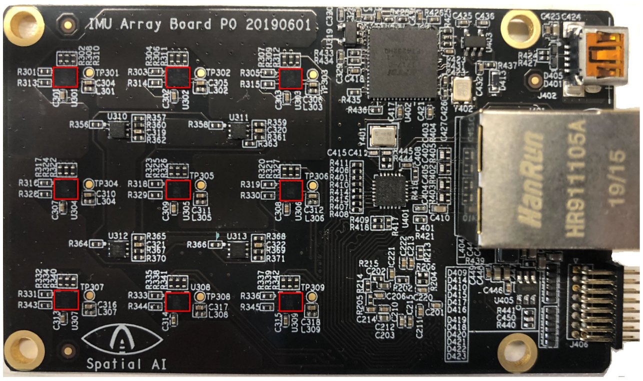

Fig. 1. The IMU array board used in this work, which contains nine ST LSM6DSOX IMUs marked by red rectangles and a processor interface to connect cameras. The IMUs are synchronized by an embedded processor.

To date, most algorithms on inertial-aided localization are designed based on a *single* IMU [7]–[13]. Although these algorithms are successfully deployed in different applications, using additional IMU sensors creates new possibilities for further improving the system accuracy and robustness. Compared to other popular sensors for localization (e.g., cameras or LiDARs), IMUs especially the off-the-shelf MEMS ones are priced only hundredths or thousandths, and of smaller size as well as lower power consumption. In addition, as a reliable proprioceptive sensor, IMU also poses less restrictions on operating environments and hardware configurations (in contrast, e.g. stereo cameras require enough spatial baseline to achieve performance gain [16], [17], which might not be feasible on various applications including mobile devices).

Most existing methods on using multiple IMUs focus on processing IMU measurements only or integration with global navigation satellite systems (GNSS) [10], [18], [19]. Fusing measurements from multiple IMUs with exteroceptive sensors for localization is a less-explored topic. To the best of the authors' knowledge, the only work in recent years in this domain is [20], which proposed an approach for vision-aided inertial navigation using measurements from multiple IMUs. However, the proposed algorithm is of significantly increased computation, which makes its real-world deployment on low-cost platforms infeasible.

Manuscript received September 10, 2019; accepted January 13, 2020. Date of publication January 24, 2020; date of current version February 4, 2020. This letter was recommended for publication by Associate Editor Dr. S. Julier and Editor Prof. E. Marchand upon evaluation of the reviewers' comments. *(Corresponding author: Mingyang Li.)*

The authors are with the A.I. Lab, Alibaba Group, Hangzhou, 310000 China (e-mail: mingzhang@alibaba-inc.com; xiangyuxu@alibaba-inc.com; yimingchen@alibaba-inc.com; mingyangli@alibaba-inc.com).

Digital Object Identifier 10.1109/LRA.2020.2969146

In this paper, a systematic framework of using multiple IMUs is proposed for localizing a moving platform, in combination with exteroceptive sensors. Compared to methods of using a single IMU, the proposed method is able to achieve better localization accuracy without incurring noticeable extra computation. The key contribution of this work is the design of multiple-IMU propagation equation, which can be integrated into both filter-based [11], [20] and optimization-based sensor fusion algorithms [15], [21] for localization. To achieve this, we first probabilistically map measurements from all IMUs onto a *virtual* IMU. This step is performed by stochastic estimation with least-square estimators and probabilistic marginalization of inter-IMU rotational accelerations. Subsequently, the propagation models for both state and error state of the virtual IMU are derived, with closed-form representation of state-transition and noise Jacobian matrices. By integrating them into a representative vision-aided inertial localization system, we show that the proposed method outperforms competing methods by a wide margin. We also point out that, in this work, experimental analysis on localization accuracy with different number of IMUs is also provided, based on our customized sensor platform consisting of nine IMUs (see Fig. 1).

II. Related Work

We here group the related work by methods for inertial-aided localization and methods using multiple IMUs.

A. Inertial Aided Localization

IMU typically generates measurements at high frequencies and provides direct 6DoF (degrees of freedom) motion estimates of a moving platform. These properties make IMU an important complementary sensor for designing localization systems [22].

Camera is another widely used sensor in various applications. However, when used for localization, the methods only relying on cameras have theoretical drawbacks. They are unable to uniquely determine the roll and pitch angles against gravity as well as the metric scale (in monocular camera setup). This limits the localization accuracy and robustness. However, when an IMU is used together with a camera, these problems can be theoretically resolved, and the performance is largely improved. As a result, there are a variety of algorithms proposed in recently years on visual-inertial localization, ranging from estimator design [9], [21], camera cost functions [23], [24], efficiency and accuracy optimization [7], [8], sensor calibration [3], [4], and so on.

IMU is also commonly used in combination with wheel encoders and/or laser range finders (LRFs) [25]. When combined with wheel encoders, IMU can help improve the dead reckoning and general localization precision [14], as well as detect wheel slippery [26]. On the other hand, LRF measures intensity and range from each laser beam at typically lower frequencies. Thus, the high-FPS measurements from an IMU can provide motion estimates between LRF scans and then derive more robust localization systems [15].

B. Methods of Using Multiple IMUs

The methods of using multiple IMUs have a long history from inertial navigation community [27]. However, the scope of early works is confined to fault detection and isolation (FDI), rather than localization accuracy until quite recently. An extensive literature review on multi-IMU FDI can be found in [28]. In what follows, we focus on the methods improving localization accuracy. Early works on processing measurements from multiple IMUs aim on measurement noise reduction via least-square estimation [29], [30]. However, the crucial correlation terms between accelerometer measurements and rotational accelerations are ignored, and then the estimation becomes suboptimal. A method is proposed in [28] to estimate acceleration/angular velocity based on a constrained optimization. These algorithms are later improved by [10], which presents a two-stage method by firstly computing error-reduced rotational velocities and subsequently obtaining the rotational accelerations and specific force vectors. [16] focuses on improving calibration accuracy, designing an anti-parallel multi-IMU array and using IMUs of mixed measurement ranges to improve localization performance. Note that the virtual IMU is generated by simply averaging the data of multiple IMUs. Another category of methods of using multiple IMUs is to design centralized estimators with a ‘stacked’ state vector, concatenating the state of each IMU [31], [32]. This type of methods does not require approximations of sensors’ measurements, and is typically of higher accuracy. However, the size of the concatenated state vector grows along with the number of IMU, and consequently so does the computation. The third type of methods is to design federated filters, using both local and master filters [33], [34]. Shared states are estimated in both the local and master filter, while other states such as biases are only estimated in the local filters.

Note that all previously mentioned methods focus on IMU signal processing or GNSS localization only. To the best of our knowledge, in existing literature, only [20] is dedicated to general localization with multiple IMUs and exteroceptive sensors. In [20], poses of each IMU are concatenated into the state vector, and relative pose constraints are derived between different IMUs. This approach is shown to generate results with higher accuracy, while at a cost of significantly increased computational complexity. In addition, although the general sensor fusion accuracy can be improved by [20], the improvement will only take effect after each update. In other words, for applications that require high frequency pose estimates from IMU forward integration (e.g., real-time control for drones or low-latency rendering in virtual reality headsets), [20] has limitations. By contrast, our method is able to improve both overall localization and forward integration accuracy without incurring additional computations.

The organization of the rest of this paper is as follows. Section III revisits the single-IMU localization method. Section IV presents the virtual IMU model generated from multiple IMUs, which is used for deriving propagation equations in Section V. Finally, Section VI presents experimental results.

III. Single IMU Method Revisited

This section revisits a standard method of using measurements from a single IMU for localization, which will be extended to the multiple-IMU case in the following sections.

A. IMU Measurement Equations

Assuming an IMU, $\{I\}$, moves with respect to a global frame $\{G\}$, the IMU measurements can be described by:

$$\boldsymbol{\omega}_m = {}^I\boldsymbol{\omega} + \boldsymbol{b}_g + \boldsymbol{n}_g, \ \ \boldsymbol{n}_g \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\sigma}_g^2 \mathbf{I}_3\right), \tag{1}$$

$$\boldsymbol{a}_m = {}^I\mathbf{R}_G({}^G\boldsymbol{a} - {}^G\boldsymbol{g}) + \boldsymbol{b}_a + \boldsymbol{n}_a, \ \ \boldsymbol{n}_a \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\sigma}_a^2 \mathbf{I}_3\right) \tag{2}$$

where $\boldsymbol{\omega}_m$ and $\boldsymbol{\alpha}_m \in \mathbb{R}^3$ are the gyroscope and accelerometer measurements, ${}^I\boldsymbol{\omega}$ and ${}^G\boldsymbol{a}$ the angular velocity and linear acceleration of the IMU expressed in frames $\{I\}$ and $\{G\}$ respectively, ${}^I\mathbf{R}_G$ the rotation from $\{G\}$ to $\{I\}$, $\boldsymbol{n}_g$ and $\boldsymbol{n}_a$ the white Gaussian noises, $\boldsymbol{b}_g$ and $\boldsymbol{b}_a$ the measurement biases modeled as random walk processes, and ${}^G\boldsymbol{g}$ the known gravity vector in global frame.

B. IMU State Vector and Error State

Conforming to [6], [7], [12], the IMU state is defined as:

$$\boldsymbol{x} = \left[{}^G_I\bar{\boldsymbol{q}}^\intercal,\, \boldsymbol{b}_g^\intercal,\, {}^G\boldsymbol{v}_I{}^\intercal,\, \boldsymbol{b}_a^\intercal,\, {}^G\boldsymbol{p}_I{}^\intercal\right]^\intercal \in \mathbb{R}^{16} \tag{3}$$

where ${}^G_I\bar{\boldsymbol{q}} \in \mathbb{R}^4$ represents rotation from IMU frame to global frame (i.e. ${}^G\mathbf{R}_I$) in quaternion [6], and ${}^G\boldsymbol{p}_I$ and ${}^G\boldsymbol{v}_I$ stand for the IMU position and velocity. The continuous-time motion dynamics of the IMU are described as:

$$\dot{\boldsymbol{x}} = f\left(\boldsymbol{x},\, \boldsymbol{\omega}_m, \boldsymbol{a}_m\right) \tag{4}$$

whose detailed form is

$${}^G\dot{\boldsymbol{p}}_I(t) = {}^G\boldsymbol{v}_I(t),\;\; \dot{\boldsymbol{b}}_g(t) = \boldsymbol{n}_{wg}(t),\;\; \dot{\boldsymbol{b}}_a(t) = \boldsymbol{n}_{wa}(t), \tag{5}$$

$${}^G\dot{\boldsymbol{v}}_I(t) = {}^G\boldsymbol{a}_I(t) = {}^I\mathbf{R}_G^T\left(\boldsymbol{a}_m - \boldsymbol{b}_a - \boldsymbol{n}_a\right) + {}^G\boldsymbol{g} \tag{6}$$

$${}^G_I\dot{\bar{\boldsymbol{q}}}(t) = \frac{1}{2}\Omega({}^I\boldsymbol{\omega}(t)){}^G_I\bar{\boldsymbol{q}}(t) = \frac{1}{2}\Omega(\boldsymbol{\omega}_m - \boldsymbol{b}_g - \boldsymbol{n}_g){}^G_I\bar{\boldsymbol{q}}(t) \tag{7}$$

where $\boldsymbol{n}_{wg} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_{wg}^2\mathbf{I}_3)$, $\boldsymbol{n}_{wa} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_{wa}^2\mathbf{I}_3)$,

$$\Omega({}^I\boldsymbol{\omega}) = \begin{bmatrix} 0 & -{}^I\boldsymbol{\omega}^\intercal \\ {}^I\boldsymbol{\omega} & -\lfloor{}^I\boldsymbol{\omega}\rfloor \end{bmatrix}, \tag{8}$$

and $\lfloor\cdot\rfloor$ represents the skew-symmetric matrix. Similar to [6], [7], [12], the IMU error state is defined as:

$$\tilde{\boldsymbol{x}} = \left[{}^I\tilde{\boldsymbol{\theta}}^\intercal,\, \tilde{\boldsymbol{b}}_g^\intercal,\, {}^G\tilde{\boldsymbol{v}}_I{}^\intercal,\, \tilde{\boldsymbol{b}}_a^\intercal,\, {}^G\tilde{\boldsymbol{p}}_I{}^\intercal\right]^\intercal \in \mathbb{R}^{15} \tag{9}$$

in which the standard additive error definition is used for the position, velocity, and biases. The quaternion local error ${}^I\tilde{\boldsymbol{\theta}}$ is defined as:

$${}^I\mathbf{R}_G \simeq (\mathbf{I}_3 - \lfloor{}^I\tilde{\boldsymbol{\theta}}\rfloor){}^I\hat{\mathbf{R}}_G \tag{10}$$

where ${}^I\hat{\mathbf{R}}_G$ represents the estimate of ${}^I\mathbf{R}_G$.

C. IMU Propagation Equations

To implement probabilistic estimators, a general method is to derive a nonlinear equation for predicting the state estimate $\hat{\boldsymbol{x}}$ and a linearized one for characterizing the error state $\tilde{\boldsymbol{x}}$:

$$\dot{\hat{\boldsymbol{x}}} = f\left(\hat{\boldsymbol{x}},\, \boldsymbol{\omega}_m, \boldsymbol{a}_m\right) \tag{11}$$

$$\dot{\tilde{\boldsymbol{x}}} = \mathbf{F}\tilde{\boldsymbol{x}} + \mathbf{G}\mathbf{n}_c \tag{12}$$

where $\mathbf{F}$ and $\mathbf{G}$ represent the continuous-time linearized state-transition matrix and noise Jacobian matrix, and $\mathbf{n}_c$ is the process noise. Once Eq. 11 and 12 are defined, IMU measurements can be straightforwardly integrated into different types of localization estimators for motion estimation [3], [6]–[9], [11], [12], [14], [15].

IV. Virtual IMU Generation

Our high-level design principle of formulating multiple-IMU algorithm is to derive IMU propagation equations conforming to the structure of Eq. 11 and 12, without non-probabilistic assumptions. The main advantage of this design is that the proposed algorithm can be directly integrated into a variety of existing inertial-aided localization systems without any complicated software re-design. To achieve this, we rely on information from multiple IMUs to generate measurements for a 'virtual' IMU, which are used to replace $\boldsymbol{\omega}_m$ and $\boldsymbol{a}_m$ in Eq. 11. The entire process is described in this section. In the next section, we derive the detailed equations for $\mathbf{F}$ and $\mathbf{G}$ in Eq. 12, after the virtual IMU substitution. We also note that, for the rest of this section, we discuss the representative case when two IMUs are involved, which can be straightforwardly extended to cases of more IMUs.

We begin by introducing our method of generating a 'virtual' IMU from multiple real IMUs.

Virtual IMU Measurement: Suppose that synchronized measurements $\boldsymbol{\omega}_{mA}$, $\boldsymbol{\omega}_{mB}$, $\boldsymbol{a}_{mA}$ and $\boldsymbol{a}_{mB}$ from IMU A and B are given, along with their extrinsic calibration $({}^A\mathbf{R}_B, {}^B\boldsymbol{p}_A)$. By arbitrarily picking the extrinsics $({}^A\mathbf{R}_V, {}^V\boldsymbol{p}_A)$ for the virtual IMU (reference frame V) relative to IMU A, the corresponding virtual gyroscope and accelerometer measurements can be generated as:

$$\boldsymbol{\omega}_{mV} = \mathbf{N}^{+}\boldsymbol{\omega}_m \tag{13}$$

$$\boldsymbol{a}_{mV} = \mathbf{T}\left(\boldsymbol{a}_m - \mathbf{S}(\boldsymbol{\omega}_{mV})\right) \tag{14}$$

where

$$\mathbf{N} = \begin{bmatrix} {}^A\mathbf{R}_V \\ {}^B\mathbf{R}_V \end{bmatrix}, \boldsymbol{Y} = \begin{bmatrix} {}^A\mathbf{R}_V\lfloor{}^V\boldsymbol{p}_A\rfloor \\ {}^B\mathbf{R}_V\lfloor{}^V\boldsymbol{p}_B\rfloor \end{bmatrix}, \mathbf{T} = \left(\mathbf{Z}^\intercal\mathbf{N}\right)^{+}\mathbf{Z}^\intercal, \tag{15}$$

$$\boldsymbol{\omega}_m = \begin{bmatrix} \boldsymbol{\omega}_{mA} \\ \boldsymbol{\omega}_{mB} \end{bmatrix}, \boldsymbol{a}_m = \begin{bmatrix} \boldsymbol{a}_{mA} \\ \boldsymbol{a}_{mB} \end{bmatrix}, \mathbf{S}(\cdot) = \begin{bmatrix} {}^A\mathbf{R}_V\lfloor\cdot\rfloor^{2V}\boldsymbol{p}_A \\ {}^B\mathbf{R}_V\lfloor\cdot\rfloor^{2V}\boldsymbol{p}_B \end{bmatrix}, \tag{16}$$

$\mathbf{Z}^\intercal$ is left nullspace projection of $\boldsymbol{Y}$, and $\mathbf{A}^{+}$ is the Moore-Penrose inverse of a real matrix $\mathbf{A}$, defined as:

$$\mathbf{A}^{+} = \left(\mathbf{A}^\intercal\mathbf{A}\right)^{-1}\mathbf{A}^\intercal. \tag{17}$$

In practice, for general multiple IMU setup with $\mathbf{N} = [{}^1\mathbf{R}_V^\intercal, {}^2\mathbf{R}_V^\intercal, \ldots, {}^{n_I}\mathbf{R}_V^\intercal]^\intercal$, the evaluation of $\mathbf{N}^{+}$ can be simplified as $\mathbf{N}^{+} = \mathbf{N}^\intercal/n_I$. With the generated virtual IMU measurements (Eq. 13 and 14), the prediction equation of the IMU state estimate (Eq. 11) can be used without any modification. The following two subsections discuss the methodology and derivations on generating the virtual IMU measurements.

A. Virtual Gyroscope Model

We first derive the measurement model for the virtual gyroscope. To start with, we use the following identities:

$${}^A\boldsymbol{\omega} = {}^A\mathbf{R}_V{}^V\boldsymbol{\omega},\; {}^B\boldsymbol{\omega} = {}^B\mathbf{R}_V{}^V\boldsymbol{\omega}. \tag{18}$$

Combination of Eq. 1 and 18 leads to:

$$\begin{bmatrix} \boldsymbol{\omega}_{mA} \\ \boldsymbol{\omega}_{mB} \end{bmatrix} = \begin{bmatrix} {}^A\mathbf{R}_V \\ {}^B\mathbf{R}_V \end{bmatrix}{}^V\boldsymbol{\omega} + \begin{bmatrix} \boldsymbol{b}_{gA} \\ \boldsymbol{b}_{gB} \end{bmatrix} + \begin{bmatrix} \boldsymbol{n}_{gA} \\ \boldsymbol{n}_{gB} \end{bmatrix} \tag{19}$$

Based on the measurements from IMUs A and B, the optimal estimate of the virtual IMU angular velocity is given by:

$$ {}^{V}\hat{\boldsymbol{\omega}} = \arg\min_{\boldsymbol{\omega}} \left\| \begin{bmatrix} \frac{\boldsymbol{\omega}_{mA}}{\sigma_{gA}} \\ \frac{\boldsymbol{\omega}_{mB}}{\sigma_{gB}} \end{bmatrix} - \begin{bmatrix} \frac{1}{\sigma_{gA}} {}^{A}\mathbf{R}_V \\ \frac{1}{\sigma_{gB}} {}^{B}\mathbf{R}_V \end{bmatrix} {}^{V}\boldsymbol{\omega} - \begin{bmatrix} \frac{\boldsymbol{b}_{gA}}{\sigma_{gA}} \\ \frac{\boldsymbol{b}_{gB}}{\sigma_{gB}} \end{bmatrix} \right\|_2^2 . \tag{20} $$

To simplify the analysis, we here assume $\sigma_{gA} = \sigma_{gB}$ for our derivations throughout the paper. Similar analysis can be developed for $\sigma_{gA} \neq \sigma_{gB}$. We also note that, if an IMU array is designed using the same type of IMUs, the $\sigma_{gA} = \sigma_{gB}$ assumption becomes exact. Solving for Eq. 20 leads to the linear least-square estimate:

$$ {}^{V}\hat{\boldsymbol{\omega}} = \mathbf{N}^{+} \begin{bmatrix} \boldsymbol{\omega}_{mA} \\ \boldsymbol{\omega}_{mB} \end{bmatrix} - \mathbf{N}^{+} \begin{bmatrix} \boldsymbol{b}_{gA} \\ \boldsymbol{b}_{gB} \end{bmatrix} \tag{21} $$

where $\mathbf{N}^{+}$ is defined by Eq. 15 and 17. In addition, by substituting Eq. 19 into Eq. 21, we obtain:

$$ {}^{V}\boldsymbol{\omega} - {}^{V}\hat{\boldsymbol{\omega}} = -\mathbf{N}^{+} \begin{bmatrix} \boldsymbol{n}_{gA} \\ \boldsymbol{n}_{gB} \end{bmatrix} . \tag{22} $$

With Eq. 21 and 22 being defined, we are able to formulate the virtual gyroscope measurements, similar to the the original IMU measurements (i.e. Eq. 1), as:

$$ \boldsymbol{\omega}_{mV} = {}^{V}\boldsymbol{\omega} + \boldsymbol{b}_{gV} + \boldsymbol{n}_{gV}, \;\; \boldsymbol{n}_{gV} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{Q}_{gV}\right) \tag{23} $$

where $\boldsymbol{\omega}_{mV}$, $\boldsymbol{b}_{gV}$, and $\boldsymbol{n}_{gV}$ are the measurement, bias vector, and noise for the virtual IMU respectively, given by:

$$ \boldsymbol{\omega}_{mV} \triangleq \mathbf{N}^{+} \begin{bmatrix} \boldsymbol{\omega}_{mA} \\ \boldsymbol{\omega}_{mB} \end{bmatrix}, \; \boldsymbol{b}_{gV} \triangleq \mathbf{N}^{+} \begin{bmatrix} \boldsymbol{b}_{gA} \\ \boldsymbol{b}_{gB} \end{bmatrix}, \; \boldsymbol{n}_{gV} \triangleq \mathbf{N}^{+} \begin{bmatrix} \boldsymbol{n}_{gA} \\ \boldsymbol{n}_{gB} \end{bmatrix}. \tag{24} $$

We also note that, taking *expectation* on both sides of Eq. 23 results Eq. 21, and further calculating the error terms results in Eq. 22. The noise covariance matrix $\mathbf{Q}_{gV}$ in Eq. 23 can be computed as:

$$ \mathbf{Q}_{gV} = \sigma_{gA}^2 \mathbf{N}^{+} \mathbf{N}^{+T} = \sigma_{gA}^2 \left(\begin{bmatrix} {}^{A}\mathbf{R}_V \\ {}^{B}\mathbf{R}_V \end{bmatrix}^T \begin{bmatrix} {}^{A}\mathbf{R}_V \\ {}^{B}\mathbf{R}_V \end{bmatrix} \right)^{-1} = \frac{\sigma_{gA}^2}{2} \mathbf{I}_3 . \tag{25} $$

Eq. 25 clearly indicates that, by using multiple IMUs, the overall IMU measurement noises can be reduced, and the localization accuracy should be improved. When $\sigma_{gA} \neq \sigma_{gB}$, the expression in Eq. 25 becomes slightly different, but the conclusion on noise reduction still holds.

Additionally, based on Eq. 24, the continuous time dynamics of $\boldsymbol{b}_{gV}$ can be described by:

$$ \dot{\boldsymbol{b}}_{gV}(t) = \boldsymbol{n}_{wgV}(t), \;\; \boldsymbol{n}_{wgV} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{Q}_{wgV}\right) \tag{26} $$

and the above noises are modeled as

$$ \mathbf{Q}_{wgV} \triangleq \mathbf{N}^{+} \begin{bmatrix} \sigma_{wgA}^2 \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \sigma_{wgB}^2 \mathbf{I}_3 \end{bmatrix} \left(\mathbf{N}^{+}\right)^{\intercal} , \tag{27} $$

Similar to Eq. 25, Eq. 27 also indicates that by generating the virtual measurements, the corresponding gyroscope bias drifts slower.

B. Virtual Accelerometer Model

Similar to the virtual gyroscope, to derive the model for virtual accelerometer, we start with the identity:

$$ {}^{G}\boldsymbol{p}_A = {}^{G}\boldsymbol{p}_V + {}^{G}\mathbf{R}_V {}^{V}\boldsymbol{p}_A . \tag{28} $$

Taking the first and second derivative of Eq. 28 leads to:

$$ \begin{aligned} {}^{G}\boldsymbol{v}_A &= {}^{G}\boldsymbol{v}_V + {}^{G}\mathbf{R}_V \lfloor {}^{V}\boldsymbol{\omega} \rfloor {}^{V}\boldsymbol{p}_A , \\ {}^{G}\boldsymbol{a}_A &= {}^{G}\boldsymbol{a}_V + {}^{G}\mathbf{R}_V \lfloor {}^{V}\boldsymbol{\omega} \rfloor^{2\,V}\boldsymbol{p}_A + {}^{G}\mathbf{R}_V \lfloor {}^{V}\boldsymbol{\phi} \rfloor {}^{V}\boldsymbol{p}_A \end{aligned} \tag{29} $$

where ${}^{V}\boldsymbol{\omega}, {}^{V}\boldsymbol{\phi} \in \mathbb{R}^3$ are the angular velocity and the angular acceleration of virtual IMU frame in global frame.

Substituting Eq. 2 into Eq. 29 leads to

$$ \boldsymbol{a}_{mA} - \mathbf{b}_{aA} - \boldsymbol{n}_{aA} = {}^{A}\mathbf{R}_V \left(\boldsymbol{s}_V + \lfloor {}^{V}\boldsymbol{\omega} \rfloor^{2\,V}\boldsymbol{p}_A - \lfloor {}^{V}\boldsymbol{p}_A \rfloor {}^{V}\boldsymbol{\phi} \right) \tag{30} $$

where the identity $\boldsymbol{a}\lfloor \boldsymbol{b} \rfloor = -\boldsymbol{b}\lfloor \boldsymbol{a} \rfloor$ is applied, and

$$ \boldsymbol{s}_V \triangleq {}^{V}\mathbf{R}_G ({}^{G}\boldsymbol{a}_V - {}^{G}\boldsymbol{g}) \tag{31} $$

represents the specific force vector [22].

Similar to computing the optimal estimate of $\boldsymbol{\omega}$ in Eq. 20, stacking accelerometer measurements from both IMUs via Eq. 30 leads to:

$$ \begin{aligned} \hat{\boldsymbol{s}}_V = \arg\min_{\boldsymbol{s}_V} \Bigg\| & \begin{bmatrix} \boldsymbol{a}_{mA} \\ \boldsymbol{a}_{mB} \end{bmatrix} - \begin{bmatrix} {}^{A}\mathbf{R}_V \\ {}^{B}\mathbf{R}_V \end{bmatrix} \boldsymbol{s}_V + \mathbf{Y} {}^{V}\boldsymbol{\phi} - \mathbf{S}({}^{V}\boldsymbol{\omega}) \\ & - \begin{bmatrix} \boldsymbol{b}_{aA} \\ \boldsymbol{b}_{aB} \end{bmatrix} \Bigg\|_2^2 \end{aligned} \tag{32} $$

where matrix $\mathbf{Y}$ and operator $\mathbf{S}(\cdot)$ are defined in Eq. 15 and 16 respectively. In Eq. 32, the rotational acceleration ${}^{V}\boldsymbol{\phi}$ is *unknown*, and also *not* directly measured by the IMU sensor. If ${}^{V}\boldsymbol{\phi}$ is not properly handled, $\boldsymbol{s}_V$ can not be accurately represented. In this work, we adopt a method inspired by [7], in which *unknown* visual features are probabilistically marginalized. By denoting $\mathbf{Z^T}$ the left nullspace projection.[1] of the matrix $\mathbf{Y}$, Eq. 32 is equivalent to:

$$ \begin{aligned} \hat{\boldsymbol{s}}_V &= \arg\min_{\boldsymbol{s}_V} \\ &\times \left\| \mathbf{Z^T} \left(\begin{bmatrix} \boldsymbol{a}_{mA} \\ \boldsymbol{a}_{mB} \end{bmatrix} - \begin{bmatrix} {}^{A}\mathbf{R}_V \\ {}^{B}\mathbf{R}_V \end{bmatrix} \boldsymbol{s}_{\mathbf{V}} - \mathbf{S}\left({}^{\mathbf{V}}\boldsymbol{\omega}\right) - \begin{bmatrix} \boldsymbol{b}_{aA} \\ \boldsymbol{b}_{aB} \end{bmatrix} \right) \right\|_{\mathbf{2}}^{\mathbf{2}} \end{aligned} \tag{33} $$

Solving Eq. 33, it follows that

$$ \hat{\boldsymbol{s}}_V = \mathbf{T} \left(\begin{bmatrix} \boldsymbol{a}_{mA} \\ \boldsymbol{a}_{mB} \end{bmatrix} - \mathbf{S}({}^{V}\boldsymbol{\omega}) - \begin{bmatrix} \boldsymbol{b}_{aA} \\ \boldsymbol{b}_{aB} \end{bmatrix} \right) , \tag{34} $$

with $\mathbf{T}$ also defined in Eq. 15. Eq. 34 allows us to define the virtual accelerometer measurement as Eq. 14. Combining Eq. 14, 30 and 34, we are able to write:

$$ \boldsymbol{a}_{mV} = \boldsymbol{s}_V + \boldsymbol{b}_{aV} + \boldsymbol{n}_{aV} - \mathbf{T} \cdot \mathbf{S}_{\boldsymbol{a}} \tag{35} $$

[1] $\mathbf{Z^T}$ can be computed through QR decomposition [7].

with

$$\boldsymbol{b}_{aV} \triangleq \mathbf{T}\begin{bmatrix}\boldsymbol{b}_{aA}\\ \boldsymbol{b}_{aB}\end{bmatrix},\ \boldsymbol{n}_{aV} \triangleq \mathbf{T}\begin{bmatrix}\boldsymbol{n}_{aA}\\ \boldsymbol{n}_{aB}\end{bmatrix} \tag{36}$$

and

$$\mathbf{S}_{\boldsymbol{a}} = \mathbf{S}(\boldsymbol{\omega}_{mV} - {}^{V}\boldsymbol{\omega}) = \begin{bmatrix}{}^{A}\mathbf{R}_{V}\boldsymbol{\zeta}{}^{V}\boldsymbol{p}_{A}\\ {}^{B}\mathbf{R}_{V}\boldsymbol{\zeta}{}^{V}\boldsymbol{p}_{B}\end{bmatrix},$$

$$\boldsymbol{\zeta} = \lfloor\boldsymbol{\omega}_{mV}\rfloor^{2} - \lfloor{}^{V}\boldsymbol{\omega}\rfloor^{2} \tag{37}$$

where $\boldsymbol{\zeta}$ can be expanded as

$$\boldsymbol{\zeta} = \lfloor\boldsymbol{\omega}_{mV}\rfloor^{2} - \lfloor\boldsymbol{\omega}_{mV} - \boldsymbol{b}_{gV} - \boldsymbol{n}_{gV}\rfloor^{2} \tag{38}$$

$$\begin{aligned}&\simeq -\lfloor\boldsymbol{b}_{gV}\rfloor^{2} + \lfloor\boldsymbol{\omega}_{mV}\rfloor\lfloor\boldsymbol{b}_{gV}\rfloor + \lfloor\boldsymbol{n}_{gV}\rfloor\lfloor\boldsymbol{\omega}_{mV}\rfloor - \lfloor\boldsymbol{b}_{gV}\rfloor\lfloor\,\boldsymbol{n}_{gV}\rfloor\\ &\quad + \lfloor\boldsymbol{b}_{gV}\rfloor\lfloor\boldsymbol{\omega}_{mV}\rfloor + \lfloor\boldsymbol{\omega}_{mV}\rfloor\lfloor\boldsymbol{n}_{gV}\rfloor - \lfloor\,\boldsymbol{n}_{gV}\rfloor\lfloor\boldsymbol{b}_{gV}\rfloor.\end{aligned} \tag{39}$$

In Eq. 39, we have used the quadratic error approximation $\lfloor\boldsymbol{n}_{gV}\rfloor^{2} \simeq \mathbf{0}_{3}$.

Eq. 35–39 clearly demonstrate that, the virtual accelerometer measurements are affected by gyroscope biases, accelerometer biases, gyroscope noises, and accelerometer noises from original IMUs. This is unlike the original IMU measurement equations, i.e., Eq. 1 and 2, in which measurement noises are *not* correlated and biases are independent.

V. Virtual IMU Propagation

In this section, a method is presented to integrate virtual IMU measurements, i.e. Eq. 13 and 14, for both the IMU state and error state. Similar to the case of single IMU, i.e. Eq. 3 and 9, we define the state and error state of the virtual IMU as:

$$\boldsymbol{x}_{V} = \left[{}^{G}_{V}\bar{\boldsymbol{q}}^{\top},\, \boldsymbol{b}_{gV}^{\top},\, {}^{G}\boldsymbol{v}_{V}{}^{\top},\, \boldsymbol{b}_{aV}^{\top},\, {}^{G}\boldsymbol{p}_{V}{}^{\top}\right]^{\top} \in \mathbb{R}^{16} \tag{40}$$

$$\tilde{\boldsymbol{x}}_{V} = \left[{}^{V}\tilde{\boldsymbol{\theta}}^{\top},\, \tilde{\boldsymbol{b}}_{gV}^{\top},\, {}^{G}\tilde{\boldsymbol{v}}_{V}{}^{\top},\, \tilde{\boldsymbol{b}}_{aV}^{\top},\, {}^{G}\tilde{\boldsymbol{p}}_{V}{}^{\top}\right]^{\top} \in \mathbb{R}^{15}. \tag{41}$$

whose continuous-time propagation equations are given by

$$\dot{\hat{\boldsymbol{x}}}_{V} = f\left(\hat{\boldsymbol{x}},\, \boldsymbol{\omega}_{mV}, \boldsymbol{a}_{mV}\right) \tag{42}$$

$$\dot{\tilde{\boldsymbol{x}}}_{V} = \mathbf{F}_{V}\tilde{\boldsymbol{x}}_{V} + \mathbf{G}_{V}\mathbf{n}_{V} \tag{43}$$

with

$$\mathbf{n}_{V} = [\boldsymbol{n}_{gV}{}^{\top},\, \boldsymbol{n}_{wgV}{}^{\top},\, \boldsymbol{n}_{aV}{}^{\top},\, \boldsymbol{n}_{waV}{}^{\top}]^{\top} \in \mathbb{R}^{12}. \tag{44}$$

Eq. 42 and 43 provide core equations for performing state and error state propagation based on the virtual IMU. This propagation can be straightforwardly integrated into different types of tightly-coupled localization algorithms, identical to the process of using a single IMU. To implement Eq. 42, we first generate the virtual IMU measurements from all IMUs via Eq. 13 and 14, and the remaining step of pose integration is identical to that of Eq. 11. By denoting

$$\hat{\boldsymbol{\omega}} = \boldsymbol{\omega}_{Vm} - \hat{\mathbf{b}}_{gV},\ \hat{\boldsymbol{a}}_{V} = \boldsymbol{a}_{Vm} - \hat{\mathbf{b}}_{aV} + \mathbf{T}\hat{\mathbf{S}}_{a} \tag{45}$$

the state-transition matrix and noise Jacobian matrix in Eq. 43 can be computed, and the detailed expression is given by Eq. 46.

$$\mathbf{F_V} = \begin{bmatrix} -\lfloor\hat{\boldsymbol{\omega}}\rfloor & -\mathbf{I_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3}\\ \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3}\\ -{}^{V}\hat{\mathbf{R}}_{G}^{\top}\lfloor\hat{\mathbf{a}}_{V}\rfloor & -{}^{V}\hat{\mathbf{R}}_{G}^{\top}\mathbf{T}\boldsymbol{\Psi} & \mathbf{0_3} & -{}^{V}\hat{\mathbf{R}}_{G}^{\top} & \mathbf{0_3}\\ \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3}\\ \mathbf{0_3} & \mathbf{0_3} & \mathbf{I_3} & \mathbf{0_3} & \mathbf{0_3} \end{bmatrix},$$

$$\mathbf{G_V} = \begin{bmatrix} -\mathbf{I_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3}\\ \mathbf{0_3} & \mathbf{I_3} & \mathbf{0_3} & \mathbf{0_3}\\ \mathbf{0_3} & -{}^{V}\hat{\mathbf{R}}_{G}^{\top}\mathbf{T}\boldsymbol{\Xi} & -{}^{V}\hat{\mathbf{R}}_{G}^{\top} & \mathbf{0_3}\\ \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{I_3}\\ \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} \end{bmatrix} \tag{46}$$

Compared to the single IMU case, $\mathbf{F}_{V}$ and $\mathbf{G}_{V}$ contain the following additional non-zero terms:

$$\frac{\partial{}^{G}\dot{\tilde{\boldsymbol{v}}}_{V}}{\partial\tilde{\boldsymbol{b}}_{gV}} = -{}^{V}\hat{\mathbf{R}}_{G}^{\top}\mathbf{T}\boldsymbol{\Psi},\ \frac{\partial{}^{G}\dot{\tilde{\boldsymbol{v}}}_{V}}{\partial\boldsymbol{n}_{gV}} = -{}^{V}\hat{\mathbf{R}}_{G}^{\top}\mathbf{T}\boldsymbol{\Xi} \tag{47}$$

where

$$\boldsymbol{\Psi} = \boldsymbol{\Xi} = \begin{bmatrix}{}^{A}\mathbf{R}_{V}\left(-\lfloor{}^{V}\hat{\boldsymbol{\omega}}\rfloor\lfloor{}^{V}\boldsymbol{p}_{A}\rfloor - \lfloor\lfloor{}^{V}\hat{\boldsymbol{\omega}}\rfloor{}^{V}\boldsymbol{p}_{A}\rfloor\right)\\ {}^{B}\mathbf{R}_{V}\left(-\lfloor{}^{V}\hat{\boldsymbol{\omega}}\rfloor\lfloor{}^{V}\boldsymbol{p}_{B}\rfloor - \lfloor\lfloor{}^{V}\hat{\boldsymbol{\omega}}\rfloor{}^{V}\boldsymbol{p}_{B}\rfloor\right)\end{bmatrix}. \tag{48}$$

The detailed derivations of $\boldsymbol{\Psi}$ and $\boldsymbol{\Xi}$ are shown in Appendix. In addition to those two terms, the expressions of all other components of $\mathbf{F}_{V}$ and $\mathbf{G}_{V}$ are identical to those in the single IMU case. Once detailed expression of both matrices in Eq. 43 is given, a computer programmable discrete-time estimator can be derived accordingly, similar to existing work on IMU integration [6], [8], [12].

It is also important to note that compared to processing measurements of a single IMU, the additional computation introduced by the proposed algorithm can be neglected. The major additional computational operations include i) computing nullspace projection of matrix $\mathbf{Y}$ and obtaining $\mathbf{T}$ in Eq. 15 and ii) computing discrete time error state propagation by handling extra fill-in in $\mathbf{F}_{V}$ and $\mathbf{G}_{V}$. Note that the computational complexity of the first task is *linear* in the number of IMUs while can be done offline given the IMUs extrinsics. The added computational cost of the second task is constant. By contrast, the computational cost of the method in [20] is cubic in the total number of camera poses and IMUs, and thus adding IMUs incurs significant extra computing time. This is not the case in our proposed method.

VI. Experiments

A. Methodology

To demonstrate the performance of our proposed localization algorithm with multiple IMUs, we integrate it into a visual-inertial odometry (VIO) approach [14].[2] The performance of VIO is tested against different number of IMUs. When there is only one IMU used, the VIO algorithm is identical to [14]. When more IMUs are used, the proposed method of generating virtual IMU measurements and performing virtual IMU propagation replaces the single-IMU pipeline. In addition, the method of [20] is implemented in our tests, as a competing algorithm to compare against.

B. Simulation Tests

We first show results from simulation tests. To minimize the reality gap of simulation, we generated synthetic poses and sensor measurements based on real-world data, similar to [35]. In the tests, we used 9 IMUs and a monocular camera, with perfectly known sensor intrinsic and extrinsic parameters. During data generation, each IMU was sampled at 200 Hz and the

[2] In our implementation of [14] only measurements from the IMUs and cameras are used.

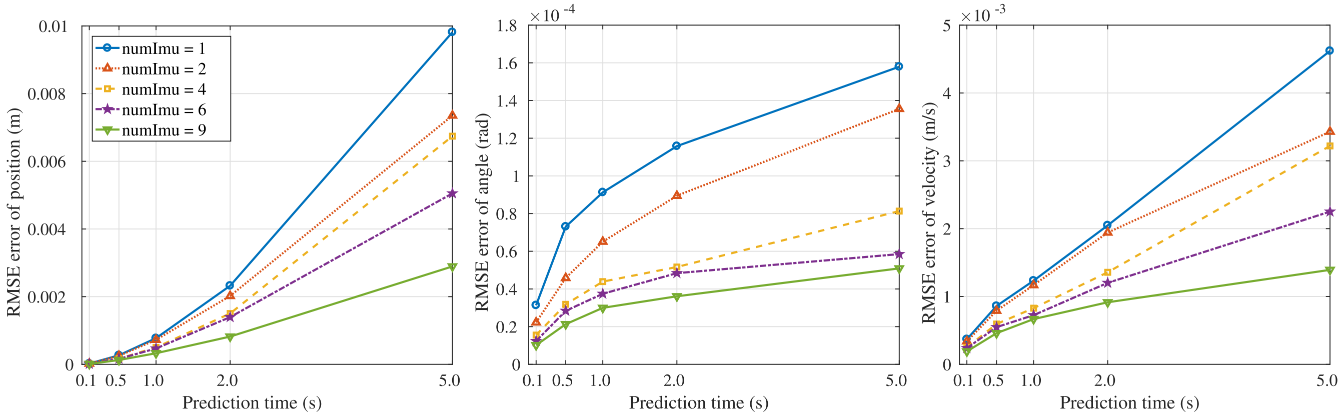

Fig. 2. IMU integration errors when different number of IMUs used for varying time durations. Left: Errors in predicted position; Middle: Errors in predicted orientation; Right: Errors in predicted velocity.

TABLE I
Pose RMS Errors as a Function of Number of IMUs

Num. of IMUs	1	2	4	6	9
Environment 1					
Pos. err. (m)	0.1985	0.1853	0.1572	0.1524	**0.1455**
Rot. err. (rad.)	0.0041	0.0040	0.0036	0.0035	**0.0031**
Environment 2					
Pos. err. (m)	0.3917	0.3721	0.3117	0.3105	**0.2955**
Rot. err. (rad.)	0.0054	0.0052	0.0048	0.0048	**0.0041**

camera captured measurements at 10 Hz. During the simulation, the layout of the multiple IMU array is identical to that of Fig. 1, and all IMUs were synchronized.

1) Pose Prediction Error: The first test is to demonstrate the pose integration accuracy by using different number of IMUs for varying time durations. Specifically, we started this test with *perfectly* known IMU pose, i.e., position, orientation, velocity, biases, and used sensor measurements to predict poses in future timestamps. For all methods, we computed the root-mean-square (RMS) errors for 3D position, rotation and velocity over 50000 simulation tests. Specifically, we generated 50 data sequences, and sampled 1000 different IMU poses from each sequence for performing prediction. We also note that [20] is identical to the single IMU case in this test, since the formulation in [20] is not able to improve the IMU prediction accuracy.

The testing results are shown in Fig. 2, which clearly demonstrate that by using more IMUs for pose integration the accuracy can be largely improved. This validates our theoretical analysis, and our design motivation of using more IMUs. In addition, we emphasize that, in all cases, using multiple IMUs for pose prediction is consistently better than the original method of using a single IMU and that of [20].

2) VIO Localization Error: The second test is to demonstrate the localization accuracy when the proposed method is integrated into VIO. Similar to the previous tests, we conducted statistical comparison between the cases when different number of IMUs were involved. For all methods, we computed the RMS errors for both 3D position and rotation, under two sets of representative simulation environments.

Table I shows the results when 1, 2, 4, 6, and 9 IMUs are used. The most important conclusion from Table I is that when multiple IMUs are used via the proposed method the localization accuracy is consistently higher than that of the single-IMU based method. In addition, we also observe that when more IMUs are used, the accuracy can be further improved, in terms of both rotational and positional estimates.

C. Real World Experiments

1) Testing Platforms and Localization Environment: To evaluate the performance of the proposed method, we also conducted experiments by using data sets from our customized sensor platform.Fig. 4 shows that our sensor platform consists of a stereo camera system with ON AR0144 imaging sensors, an array of 9 ST LSM6DSOX IMUs (see Fig. 1) and a RTK-GPS system. In our experiment, we chose the virtual IMU frame identical to the central IMU in the IMU array.

In our experiment, the multiple IMUs are synchronized by hardware design using IMUs' interrupt signals [36], and performed extrinsic calibration between IMUs offline via [4]. The camera intrinsics and camera to IMU extrinsics were also calibrated offline via [4]. Note that although both temporal and spatial parameters between sensors can be estimated online [37], [38], reducing these uncertainties beforehand is beneficial for the overall performance.

To test the proposed method in different environments, we collected three datasets in indoor environments and three data sets on urban streets. During the data collection, images were recorded at 10 Hz with 640 × 400 and 1280 × 800 pixel resolutions in indoor and outdoor environments respectively. Measurements from all IMUs were at 200 Hz. For indoor data sets, the trajectory lengths are around 50 to 100 meters. For the urban street data sets, the lengths are about 3.5 km. For the tests on urban streets, we also recorded the poses from the RTK-GPS, which are used as ground truth for computing RMS errors for different algorithms. In indoor environments, due to lack of precise ground truth, we started and stopped the motion of our sensor platform at exactly the same location, to enable computing the final drift as the error metric. Similar to the simulation tests, we here also implemented VIO algorithms by using different number of IMUs as well as the competing algorithm [20].

2) Qualitative Results: We here report the qualitative results of all methods tested in both indoor and urban street tests, which are shown in Fig. 3. For all datasets, the proposed method consistently outperforms the original VIO algorithm with a

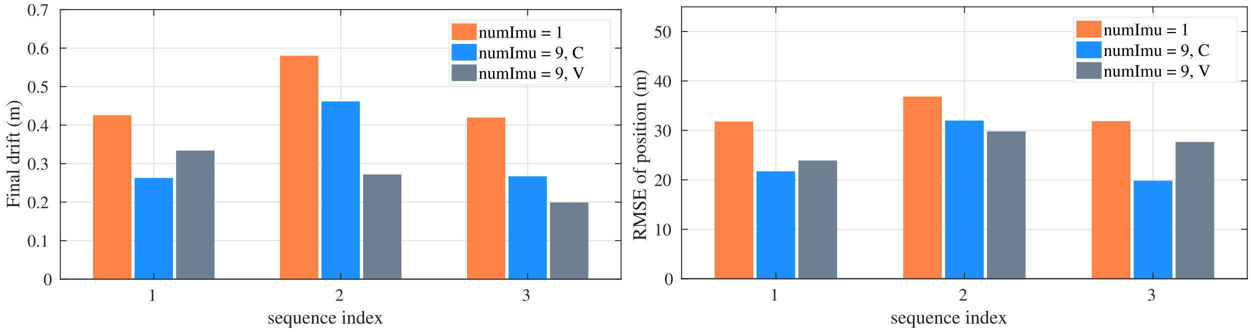

Fig. 3. Localization errors in real-world experiments. Three methods are tested: The original VIO algorithm using a single IMU, the modified VIO algorithm using the proposed method with 9 IMUs (method V, V = Virtual), and [20] with 9 IMUs (method C, C = Centralized). The left plot represents the final drifts in indoor tests, and the right plot denotes the positional RMS errors in urban street tests.

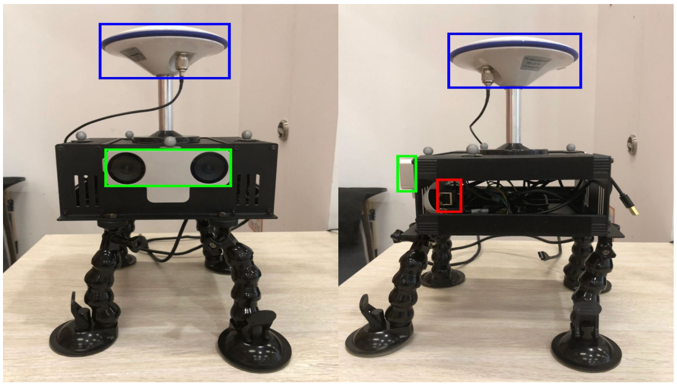

Fig. 4. Sensor platform in our tests. GPS antenna is marked by a blue rectangle, the stereo camera system by a green one and the IMU array board by a red one. Left figure: The front view. Right figure: The side view.

TABLE II
AVERAGE ACCURACY IMPROVEMENT PER COMPUTATIONAL COST, COMPUTED BY EQ. 49

Num. Seq	1	2	3
Eckenhoff et al. [20]	0.2545	0.1186	0.2865
Proposed	0.6894	0.3795	0.8198

TABLE III
TIME (MSEC) PER UPDATE FOR COMPETING ALGORITHMS

Num. Seq	1	2	3
Single-IMU (msec)	9.9380	9.4280	10.6717
Eckenhoff et al. [20] (msec)	39.5155	40.9543	42.0586
Proposed (msec)	10.0381	9.8014	11.6740

single IMU. In fact, the average error reduction is about 31%, which is significant and can be used as guidance when designing hardware sensor platform and software package for performing high-precision localization. In addition, when compared against [20], the proposed method yields similar localization accuracy, since both methods utilize measurement information from multiple IMUs without non-probabilistic approximations.

Moreover, the computational efficiency of the proposed method and [20] is evaluated with the following metric:

$$\kappa = \frac{\text{Err. in one-IMU method} - \text{Err. in testing method}}{\text{Time used per update cycle in testing method}} \tag{49}$$

In fact, κ represents the gain in localization accuracy per computational cost, and the results are shown in Table II. The results demonstrate that, the proposed method is significantly more efficient when seeking for accuracy gain. In addition, the average time per update for tested algorithms is also given by Table III, which shows that the computational cost of the proposed method is almost identical to that of the single-IMU method and that of [20] is noticeably higher. This verifies our theoretical analysis that since [20] models all IMUs' poses in the state vector of an estimator, the computational cost grows significantly along with the number of IMU. By contrast, in our method, since we generate data for a virtual IMU before performing update (with linear computational cost in the number of IMU, and constant cost in the number of VIO keyframes), the computational cost almost keeps unchanged.

It is also interesting to point out that the accuracy gain achieved in our test is similar to that of [21], which uses an 11-cm baseline stereo vision system. Compared to stereo solution, our method is of lower sensor cost and computational cost. Moreover, the performance of stereo based algorithms will degrade once stereo baseline becomes shorter [16]. The feasible baseline for various applications, e.g., mobile devices, will be shorter than 11 cm, and the proposed method will become even more preferred in those cases. Moreover, compared to using a single IMU with higher precision, the proposed method has comparable localization accuracy yet lower sensor cost.

VII. CONCLUSIONS

In this paper, we present a lightweight and accurate localization algorithm by using multiple inertial measurement units and exteroceptive sensors. Specifically, we propose a method of optimally generating a virtual IMU from the measurements of multiple IMUs, which is followed by closed-form derivation of virtual IMU propagation equations for sensor fusion. By integrating the proposed method into a VIO algorithm, we show that it significantly improves the localization accuracy and outperforms competing algorithms by wide margins.

APPENDIX

We here provide detailed derivation for $\boldsymbol{\Psi}$ and $\boldsymbol{\Xi}$:

$$\boldsymbol{\Psi} = \frac{\partial \mathbf{S}(\boldsymbol{\omega}_{mV} - {}^{V}\boldsymbol{\omega})}{\partial \boldsymbol{b}_{gV}}, \quad \boldsymbol{\Xi} = \frac{\partial \mathbf{S}(\boldsymbol{\omega}_{mV} - {}^{V}\boldsymbol{\omega})}{\partial \boldsymbol{n}_{gV}} \tag{50}$$

To start with, we write $\hat{\boldsymbol{\zeta}}$ from Eq. 39 as:

$$\hat{\boldsymbol{\zeta}} \simeq -\lfloor \hat{\boldsymbol{b}}_{gV} \rfloor^2 + \lfloor \boldsymbol{\omega}_{mV} \rfloor \lfloor \hat{\boldsymbol{b}}_{gV} \rfloor + \lfloor \hat{\boldsymbol{b}}_{gV} \rfloor \lfloor \boldsymbol{\omega}_{mV} \rfloor \tag{51}$$

Subtracting Eq. 39 from Eq. 51 leads to

$$\begin{aligned}\tilde{\boldsymbol{\zeta}} = \boldsymbol{\zeta} - \hat{\boldsymbol{\zeta}} &\simeq -\lfloor \hat{\boldsymbol{b}}_{gV} \rfloor \lfloor \tilde{\boldsymbol{b}}_{gV} \rfloor - \lfloor \tilde{\boldsymbol{b}}_{gV} \rfloor \lfloor \hat{\boldsymbol{b}}_{gV} \rfloor + \lfloor \boldsymbol{\omega}_{mV} \rfloor \lfloor \tilde{\boldsymbol{b}}_{gV} \rfloor \\ &\quad + \lfloor \boldsymbol{n}_{gV} \rfloor \lfloor \boldsymbol{\omega}_{mV} \rfloor - \lfloor \hat{\boldsymbol{b}}_{gV} \rfloor \lfloor \boldsymbol{n}_{gV} \rfloor + \lfloor \tilde{\boldsymbol{b}}_{gV} \rfloor \lfloor \boldsymbol{\omega}_{mV} \rfloor \\ &\quad + \lfloor \boldsymbol{\omega}_{mV} \rfloor \lfloor \boldsymbol{n}_{gV} \rfloor - \lfloor \boldsymbol{n}_{gV} \rfloor \lfloor \hat{\boldsymbol{b}}_{gV} \rfloor \\ &= \lfloor {}^V\hat{\boldsymbol{\omega}} \rfloor \lfloor \tilde{\boldsymbol{b}}_{gV} \rfloor + \lfloor \tilde{\boldsymbol{b}}_{gV} \rfloor \lfloor {}^V\hat{\boldsymbol{\omega}} \rfloor + \lfloor {}^V\hat{\boldsymbol{\omega}} \rfloor \lfloor \boldsymbol{n}_{gV} \rfloor + \lfloor \boldsymbol{n}_{gV} \rfloor \lfloor {}^V\hat{\boldsymbol{\omega}} \rfloor\end{aligned}$$

Therefore, for any vector $\boldsymbol{y}$, we have

$$\begin{aligned}\tilde{\boldsymbol{\zeta}}\boldsymbol{y} &= -\left(\lfloor {}^V\hat{\boldsymbol{\omega}} \rfloor \lfloor \boldsymbol{y} \rfloor + \lfloor \lfloor {}^V\hat{\boldsymbol{\omega}} \rfloor \boldsymbol{y} \rfloor\right) \tilde{\boldsymbol{b}}_{gV} \\ &\quad - \left(\lfloor {}^V\hat{\boldsymbol{\omega}} \rfloor \lfloor \boldsymbol{y} \rfloor + \lfloor \lfloor {}^V\hat{\boldsymbol{\omega}} \rfloor \boldsymbol{y} \rfloor\right) \boldsymbol{n}_{gV}.\end{aligned}$$

As a result,

$$\begin{aligned}\mathbf{S}(\boldsymbol{\omega}_{mV} - {}^V\boldsymbol{\omega}) &= \begin{bmatrix} {}^A\mathbf{R}_V \boldsymbol{\zeta} {}^V\boldsymbol{p}_A \\ {}^B\mathbf{R}_V \boldsymbol{\zeta} {}^V\boldsymbol{p}_B \end{bmatrix} = \begin{bmatrix} {}^A\mathbf{R}_V (\hat{\boldsymbol{\zeta}} + \tilde{\boldsymbol{\zeta}}) {}^V\boldsymbol{p}_A \\ {}^B\mathbf{R}_V (\hat{\boldsymbol{\zeta}} + \tilde{\boldsymbol{\zeta}}) {}^V\boldsymbol{p}_B \end{bmatrix} \\ &= \begin{bmatrix} {}^A\mathbf{R}_V \hat{\boldsymbol{\zeta}} {}^V\boldsymbol{p}_A \\ {}^B\mathbf{R}_V \hat{\boldsymbol{\zeta}} {}^V\boldsymbol{p}_B \end{bmatrix} + \boldsymbol{\Psi} \tilde{\boldsymbol{b}}_{gV} + \boldsymbol{\Xi} \boldsymbol{n}_{gV}\end{aligned} \tag{52}$$

where

$$\boldsymbol{\Psi} = \boldsymbol{\Xi} = \begin{bmatrix} {}^A\mathbf{R}_V \left(-\lfloor {}^V\hat{\boldsymbol{\omega}} \rfloor \lfloor {}^V\boldsymbol{p}_A \rfloor - \lfloor \lfloor {}^V\hat{\boldsymbol{\omega}} \rfloor {}^V\boldsymbol{p}_A \rfloor\right) \\ {}^B\mathbf{R}_V \left(-\lfloor {}^V\hat{\boldsymbol{\omega}} \rfloor \lfloor {}^V\boldsymbol{p}_B \rfloor - \lfloor \lfloor {}^V\hat{\boldsymbol{\omega}} \rfloor {}^V\boldsymbol{p}_B \rfloor\right). \end{bmatrix} \tag{53}$$

This completes our derivation.

REFERENCES

[1] B. B. Koçer, V. E. Ömürlü, E. Akdoğan, and C. S. Tüfekçi, “Development of a MEMS-based IMU unit,” in *Proc. 6th Int. Conf. Recent Adv. Space Technologies*, 2013, pp. 389–393.

[2] J. Wall and D. Bevly, “Characterization of various imu error sources and the effect on navigation performance,” in *Proc. ION GNSS*, 2014, pp. 967–978.

[3] M. Li, H. Yu, X. Zheng, and A. I. Mourikis, “High-fidelity sensor modeling and self-calibration in vision-aided inertial navigation,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 409–416.

[4] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, “Extending kalibr: Calibrating the extrinsics of multiple imus and of individual axes,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 4304–4311.

[5] T. Schneider, M. Li, M. Burri, J. Nieto, R. Siegwart, and I. Gilitschenski, “Visual-inertial self-calibration on informative motion segments,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 6487–6494.

[6] N. Trawny and S. I. Roumeliotis, “Indirect kalman filter for 3d pose estimation,” University of Minnesota, Minneapolis, MN, USA, Dept. Comp. Sci. & Eng., Tech. Rep., 2005, vol. 2.

[7] M. Li and A. I. Mourikis, “High-precision, consistent EKF-based visual-inertial odometry,” *Int. J. Robot. Res.*, vol. 32, no. 6, pp. 690–711, 2013.

[8] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, “On-manifold preintegration for real-time visual–inertial odometry,” *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 1–21, Feb. 2017.

[9] T. Qin, P. Li, and S. Shen, “VINS-mono: A robust and versatile monocular visual-inertial state estimator,” *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[10] I. Skog, J.-O. Nilsson, P. Händel, and A. Nehorai, “Inertial sensor arrays, maximum likelihood, and cramér–rao bound,” *IEEE Trans. Signal Process.*, vol. 64, no. 16, pp. 4218-4227, 15 Aug. 15, 2016.

[11] M. Li, B. H. Kim, and A. I. Mourikis, “Real-time motion tracking on a cellphone using inertial sensing and a rolling-shutter camera,” in *Proc. Int. Conf. Robot. Autom.*, 2013, pp. 4712–4719.

[12] T. Schneider *et al.*, “maplab: An open framework for research in visual-inertial mapping and localization,” *IEEE IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1418–1425, Jul. 2018.

[13] Z. Huai and G. Huang, “Robocentric visual-inertial odometry,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 6319–6326.

[14] M. Zhang, X. Zuo, Y. Chen, and M. Li, “Localization for ground robots: On manifold representation, integration, re-parameterization, and optimization,” 2019, *arXiv:1909.03423v2*.

[15] H. Ye, Y. Chen, and M. Liu, “Tightly coupled 3d lidar inertial odometry and mapping,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2019, pp. 3144–3150.

[16] D. C. Herath, K. Kodagoda, and G. Dissanayake, “Modeling errors in small baseline stereo for slam,” in *Proc. 9th Int. Conf. Control, Autom., Robot. Vision*, 2006, pp. 1–6.

[17] H. Herath, S. Kodagoda, and G. Dissanayake, *Stereo Vision based SLAM: Issues and solutions*. Rijeka, Croatia: ITECH, 2007.

[18] M. Jafari, “Optimal redundant sensor configuration for accuracy increasing in space inertial navigation system,” *Aerosp. Sci. Technol.*, vol. 47, pp. 467–472, Dec. 2015.

[19] A. Shahri and R. Rasoulzadeh, “Implementation of a low-cost multi-imu by using information form of a steady state kalman filter,” *AUT J. Elect. Eng.*, vol. 49, no. 2, pp. 195–204, 2017.

[20] K. Eckenhoff, P. Geneva, and G. Huang, “Sensor-failure-resilient multi-imu visual-inertial navigation,” in *Proc. Int. Conf. Robot. Autom.*, 2019, pp. 3542–3548.

[21] 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, Mar. 2015.

[22] J. A. Farrell, *Aided Navigation: GPS With High Rate Sensors*. NY, USA: McGraw Hill, 2008.

[23] X. Zheng, Z. Moratto, M. Li, and A. I. Mourikis, “Photometric patch-based visual-inertial odometry,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 3264–3271.

[24] L. Von Stumberg, V. Usenko, and D. Cremers, “Direct sparse visual-inertial odometry using dynamic marginalization,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 2510–2517.

[25] Y. Chen, M. Zhang, D. Hong, C. Deng, and M. Li, “Perception system design for low-cost commercial ground robots: Sensor configurations, calibration, localization, and mapping,” in *Proc. Int. Conf. Intell. Robots Syst.*, 2019.

[26] J. Yi, J. Zhang, D. Song, and S. Jayasuriya, “Imu-based localization and slip estimation for skid-steered mobile robots,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2007, pp. 2845–2850.

[27] F. A. Evans and J. C. Wilcox, “Experimental strapdown redundant sensor inertial navigation system,” *J. Spacecraft Rockets*, vol. 7, no. 9, pp. 1070–1074, 1970.

[28] Y. Yuksel and N. El-Sheimy, “An optimal sensor fusion method for skew-redundant inertial measurement units,” *JAG*, vol. 5, no. 2, pp. 99–115, 2011.

[29] I. Colomina, M. Giménez, J. Rosales, M. Wis, A. Gómez, and P. Miguelsanz, “Redundant imus for precise trajectory determination,” in *Proc. 20th ISPRS Congr.*, 2004, p. 7.

[30] A. Waegli, S. Guerrier, and J. Skaloud, “Redundant mems-imu integrated with gps for performance assessment in sports,” in *Proc. IEEE/ION PLANS*, 2008, pp. 1260–1268.

[31] J. B. Bancroft, G. Lachapelle, M. E. Cannon, and M. G. Petovello, “Twin imu-hsgps integration for pedestrian navigation,” in *Proc. ION GNSS*, 2008, pp. 16–20.

[32] J. B. Bancroft, “Multiple imu integration for vehicular navigation,” in *Proc. ION GNSS*, vol. 1, 2009, pp. 1–13.

[33] N. A. Carlson, “Federated filter for distributed navigation and tracking applications,” in *Proc. Annu. Meeting ION*, vol. 41, pp. 297–322, 2001.

[34] J. B. Bancroft and G. Lachapelle, “Data fusion algorithms for multiple inertial measurement units,” *Sensors*, vol. 11, no. 7, pp. 6771–6798, 2011.

[35] M. Li and A. I. Mourikis, “Vision-aided inertial navigation with rolling-shutter cameras,” *Int. J. Robot. Res.*, vol. 33, no. 11, pp. 1490–1507, 2014.

[36] B. Li “A cost effective synchronization system for multisensor integration,” in *Proc. ION GNSS*, 2004, pp. 1627–1635.

[37] M. Li and A. I. Mourikis, “Online temporal calibration for camera–imu systems: Theory and algorithms,” *Int. J. Robot. Res.*, vol. 33, no. 7, pp. 947–964, 2014.

[38] J. Maye, H. Sommer, G. Agamennoni, R. Siegwart, and P. Furgale, “Online self-calibration for robotic systems,” *Int. J. Robot. Res.*, vol. 35, no. 4, pp. 357–380, 2016.