A General Optimization-based Framework for Local Odometry Estimation with Multiple Sensors

Tong Qin, Jie Pan, Shaozu Cao, and Shaojie Shen

***Abstract*— Nowadays, more and more sensors are equipped on robots to increase robustness and autonomous ability. We have seen various sensor suites equipped on different platforms, such as stereo cameras on ground vehicles, a monocular camera with an IMU (Inertial Measurement Unit) on mobile phones, and stereo cameras with an IMU on aerial robots. Although many algorithms for state estimation have been proposed in the past, they are usually applied to a single sensor or a specific sensor suite. Few of them can be employed with multiple sensor choices. In this paper, we proposed a general optimization-based framework for odometry estimation, which supports multiple sensor sets. Every sensor is treated as a general factor in our framework. Factors which share common state variables are summed together to build the optimization problem. We further demonstrate the generality with visual and inertial sensors, which form three sensor suites (stereo cameras, a monocular camera with an IMU, and stereo cameras with an IMU). We validate the performance of our system on public datasets and through real-world experiments with multiple sensors. Results are compared against other state-of-the-art algorithms. We highlight that our system is a general framework, which can easily fuse various sensors in a pose graph optimization. Our implementations are open source[1].**

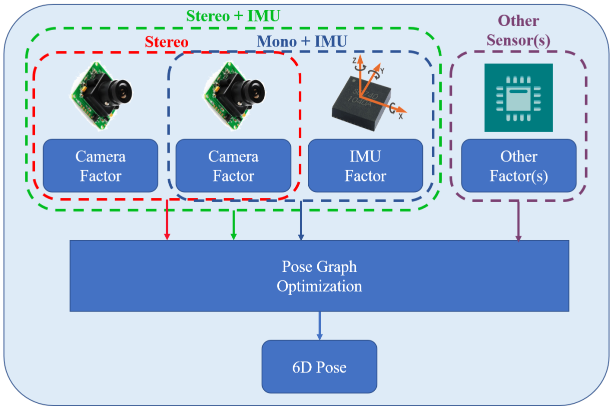

Fig. 1. An illustration of the proposed framework for state estimation, which supports multiple sensor choices, such as stereo cameras, a monocular camera with an IMU, and stereo cameras with an IMU. Each sensor is treated as a general factor. Factors which share common state variables are summed together to build the optimization problem.

I. Introduction

Real-time 6-DoF (Degrees of Freedom) state estimation is a fundamental technology for robotics. Accurate state estimation plays an important role in various intelligent applications, such as robot exploration, autonomous driving, VR (Virtual Reality) and AR (Augmented Reality). The most common sensors we use in these applications are cameras. A large number of impressive vision-based algorithms for pose estimation has been proposed over the last decades, such as [1]–[5]. Besides cameras, the IMU is another popular option for state estimation. The IMU can measure acceleration and angular velocity at a high frequency, which is necessary for low-latency pose feedback in real-time applications. Hence, there are numerous research works fusing vision and IMU together, such as [6]–[12]. Another popular sensor used in state estimation is LiDAR. LiDAR-based approaches [13] achieve accurate pose estimation in a confined local environment. Although a lot of algorithms have been proposed in the past, they are usually applied to a single input sensor or a specific sensor suite.

Recently, we have seen platforms equipped with various sensor sets, such as stereo cameras on ground vehicles, a monocular camera with an IMU on mobile phones, stereo cameras with an IMU on aerial robots. However, as most traditional algorithms were designed for a single sensor or a specific sensor set, they cannot be ported to different platforms. Even for one platform, we need to choose different sensor combinations in different scenarios. Therefore, a general algorithm which supports different sensor suites is required. Another practical requirement is that in case of sensor failure, an inactive sensor should be removed and an alternative sensor should be added into the system quickly. Hence, a general algorithm which is compatible with multiple sensors is in need.

In this paper, we propose a general optimization-based framework for pose estimation, which supports multiple sensor combinations. We further demonstrate it with visual and inertial sensors, which form three sensor suites (stereo cameras, a monocular camera with an IMU, and stereo cameras with an IMU). We can easily switch between different sensor combinations. We highlight the contribution of this paper as follows:

- a general optimization-based framework for state estimation, which supports multiple sensors.
- a detailed demonstration of state estimation with visual and inertial sensors, which form different sensor suites (stereo cameras, a monocular camera + an IMU, and stereo cameras + an IMU).
- an evaluation of the proposed system on both public datasets and real experiments.

All authors are with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, China. {tong.qin, jie.pan, shaozu.cao}@connect.ust.hk, eeshaojie@ust.hk.

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

- open-source code for the community.

II. Related Work

State estimation has been a popular research topic over the last decades. A large number of algorithms focus on accurate 6-DoF pose estimation. We have seen many impressive approaches that work with one kind of sensor, such as visual-based methods [1]–[5], LiDAR-based methods [13], RGB-D based methods [14]. and event-based methods [15]. Approaches work with a monocular camera is hard to achieve 6-DoF pose estimation, since absolute scale cannot be recovered from a single camera. To increase the observability and robustness, multiple sensors which have complementary properties are fused together.

There are two trends of approaches for multi-sensor fusion. One is filter-based methods, the other is optimization-based methods. Filter-based methods are usually achieved by EKF (Extended Kalman Filter). Visual and inertial measurements are usually filtered together for 6-DoF state estimation. A high-rate inertial sensor is used for state propagation and visual measurements are used for the update in [9, 16]. MSCKF [6, 7] was a popular EKF-based VIO (Visual Inertial Odometry), which maintained several camera poses and leveraged multiple camera views to form the multi-constraint update. Filter-based methods usually linearize states earlier and suffer from error induced by inaccurate linear points. To overcome the inconsistency caused by linearized error, observability constrained EKF [17] was proposed to improve accuracy and consistency. An UKF (Unscented Kalman Filter) algorithm was proposed in [18], where visual, LiDAR and GPS measurements were fused together. UKF is an externsion of EKF without analytic Jacobians. Filter-based methods are sensitive to time synchronization. Any late-coming measurements will cause trouble since states cannot be propagated back in filter procedure. Hence, special ordering mechanism is required to make sure that all measurements from multiple sensors are in order.

Optimization-based methods maintain a lot of measurements and optimize multiple variables at once, which is also known as Bundle Adjustment (BA). Compared with filter-based method, optimization-based method have advantage in time synchronization. Because the big bundle serves as a nature buffer, it can easily handle the case when measurments from multiple sensors come in disorder. Optimization-based algorithms also outperform the filter-based algorithms in term of accuracy at the cost of computational complexity. Early optimization solvers, such as G2O [19], leveraged the Gauss-Newton and Levenberg-Marquardt approaches to solve the problem. Although the sparse structure was employed in optimization solvers, the complexity grown quadratically with the number of states and measurements. In order to achieve real-time performance, some algorithms have explored incremental solvers, while others bounded the size of the pose graph. iSAM2 [20] was an efficient incremental solver, which reused the previous optimization result to reduce computation when new measurements came. The optimization iteration only updated a small part of states instead of the whole pose graph. Afterward, an accelerated solver was proposed in [21], which improved efficiency by reconstructing dense structure into sparse blocks. Methods, that keep a fixed sized of pose graph, are called sliding-window approaches. Impressive optimization-based VIO approaches, such as [8, 10, 12], optimized variables over a bounded-size sliding window. The previous states were marginalized into a prior factor without loss of information in [8, 12]. In this paper, we adopt a sliding-window optimization-based framework for state estimation.

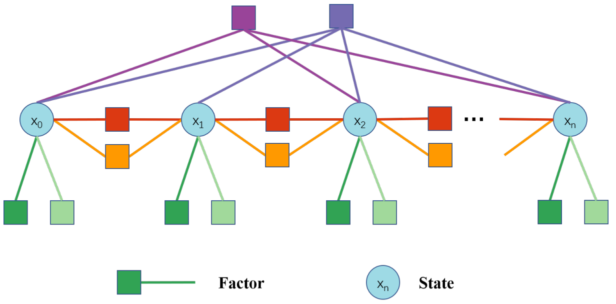

Fig. 2. A graphic illustration of the pose graph. Each node represents states (position, orientation, velocity and so on) at one moment. Each edge represents a factor, which is derived by one measurement. Edges constrain one state, two states or multiple states.

III. System Overview

The structure of proposed framework is shown in Fig. 1. Multiple kinds of sensors can be freely combined. The mesurement of each sensor is treated as a general factor. Factors and their related states form the pose graph. An illustration of pose graph is shown in Fig. 2. Each node represents states (position, orientation, velocity and so on) at one moment. Each edge represents a factor, which is derived by one measurement. Factors constrain one state, two states or multiple states. For IMU factor, it constrains two consecutive states by continuous motion restriction. For a visual landmark, its factor constrains multiple states since it is observed on multiple frames. Once the graph is built, optimizing it equals to finding the configuration of nodes that match all edges as much as possible.

In this paper, we specifically demonstrate the system with visual and inertial sensors. Visual and inertial sensors can form three combinations for 6-DoF state estimation, which are stereo cameras, a monocular camera with an IMU, and stereo cameras with an IMU. A graphic illustration of the proposed framework with visual and inertial sensors is shown in Fig. 3. Several camera poses, IMU measurements and visual measurements exist in the pose graph. The IMU and one of cameras are optional.

IV. Methodology

A. Problem Definition

1) States: Main states that we need to estimate includes 3D position and orientation of robot's center. In addition, we

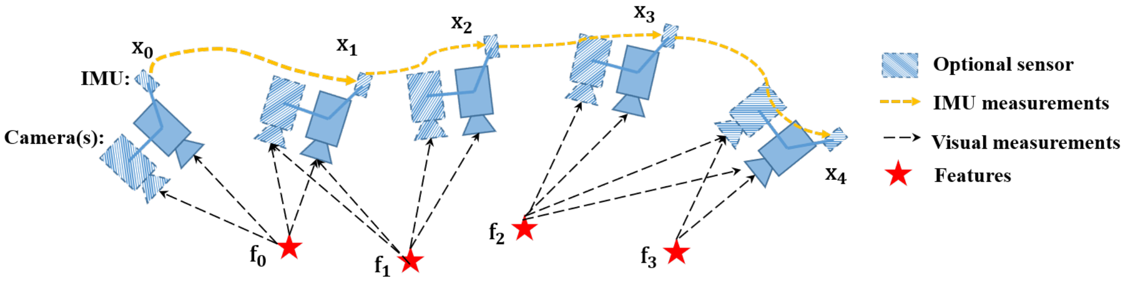

Fig. 3. A graphic illustration of the proposed framework with visual and inertial sensors. The IMU and one of cameras are optional. Therefore, it forms three types (stereo cameras, a monocular camera with an IMU, and stereo cameras with an IMU). Several camera poses, IMU measurements and visual measurements exist in the pose graph.

have other optional states, which are related to sensors. For cameras, depths or 3D locations of visual landmarks need to be estimated. For IMU, it produces another motion variable, velocity, Also, time-variant acceleration bias and gyroscope bias of the IMU are needed to be estimated. Hence, for visual and inertial sensors, whole states we need to estimate are defined as follows:

$$\begin{aligned} \mathcal{X} &= [\mathbf{p}_0, \mathbf{R}_0, \mathbf{p}_1, \mathbf{R}_1, ..., \mathbf{p}_n, \mathbf{R}_n, \mathbf{x}_{cam}, \mathbf{x}_{imu}] \\ \mathbf{x}_{cam} &= [\lambda_0, \lambda_1, ..., \lambda_l] \\ \mathbf{x}_{imu} &= [\mathbf{v}_0, \mathbf{b}_{a_0}, \mathbf{b}_{g_0}, \mathbf{v}_1, \mathbf{b}_{a_1}, \mathbf{b}_{g_1}, ..., \mathbf{v}_n, \mathbf{b}_{a_n}, \mathbf{b}_{g_n}], \end{aligned} \tag{1}$$

where $\mathbf{p}$ and $\mathbf{R}$ are basic system states, which correspond to position and orientation of body expressed in world frame. $\mathbf{x}_{cam}$ is camera-ralated state, which includes depth λ of each feature observed in the first frame. $\mathbf{x}_{imu}$ is IMU-ralated variable, which is composed of velocity $\mathbf{v}$, acceleration bias $\mathbf{b}_a$ and gyroscope bias $\mathbf{b}_g$. $\mathbf{x}_{imu}$ can be omitted if we only use stereo camera without an IMU. The translation from sensors' center to body's center are assumed to be known, which are calibrated offline. In order to simplify the notation, we denote the IMU as body's center (If the IMU is not used, we denote left camera as body's center).

2) Cost Function: The nature of state estimation is an MLE (Maximum Likelihood Estimation) problem. The MLE consists of the joint probability distribution of robot poses over a period of time. Under the assumption that all measurements are independent, the problem is typically derived as,

$$\mathcal{X}^* = \arg\max_{\mathcal{X}} \prod_{t=0}^{n} \prod_{k \in \mathbf{S}} p(\mathbf{z}_t^k | \mathcal{X}), \tag{2}$$

where $\mathbf{S}$ is the set of measurements, which come from cameras, IMU and other sensors. We assume the uncertainty of measurements is Gaussian distributed, $p(\mathbf{z}_t^k|\mathcal{X}) \sim \mathcal{N}(\bar{\mathbf{z}}_t^k, \Omega_t^k)$. Therefore, the negative log-likelihood of above-mentioned equation is written as,

$$\begin{aligned} \mathcal{X}^* &= \arg\max_{\mathcal{X}} \prod_{t=0}^{n} \prod_{k \in \mathbf{S}} exp(-\frac{1}{2} \left\| \mathbf{z}_t^k - h_t^k(\mathcal{X}) \right\|_{\mathbf{\Omega}_t^k}^2) \\ &= \arg\min_{\mathcal{X}} \sum_{t=0}^{n} \sum_{k \in \mathbf{S}} \left\| \mathbf{z}_t^k - h_t^k(\mathcal{X}) \right\|_{\mathbf{\Omega}_t^k}^2 . \end{aligned} \tag{3}$$

The Mahalanobis norm is defined as $\|\mathbf{r}\|_{\mathbf{\Omega}}^2 = \mathbf{r}^T \mathbf{\Omega}^{-1} \mathbf{r}$. $h(\cdot)$ is the sensor model, which is detailed in the following section. Then the state estimation is converted to a nonlinear least squares problem, which is also known as Bundle Adjustment (BA).

B. Sensor Factors

1) Camera Factor: The framework supports both monocular and stereo cameras. The intrinsic parameters of every camera and the extrinsic transformation between cameras are supposed to be known, which can be easily calibrated offline. For each camera frame, corner features [22] are detected. These features are tracked in previous frame by KLT tracker [23]. For the stereo setting, the tracker also matches features between the left image and right image. According to the feature associations, we construct the camera factor with per feature in each frame. The camera factor is the reprojection process, which projects a feature from its first observation into following frames.

Considering the feature l that is first observed in the image i, the residual for the observation in the following image t is defined as:

$$\begin{aligned} \mathbf{z}_t^l - h_t^l(\mathcal{X}) &= \mathbf{z}_t^l - h_t^l(\mathbf{R}_i, \mathbf{p}_i, \mathbf{R}_t, \mathbf{p}_t, \lambda_l) \\ &= \begin{bmatrix} u_t^l \\ v_t^l \end{bmatrix} - \pi_c(\mathbf{T}_c^{b\,-1} \, \mathbf{T}_t{}^{-1} \, \mathbf{T}_i \, \mathbf{T}_c^b \, \pi_c{}^{-1}(\lambda_l, \begin{bmatrix} u_i^l \\ v_i^l \end{bmatrix})), \end{aligned} \tag{4}$$

where $[u_i^l, v_i^l]$ is the first observation of the l feature that appears in the i image. $[u_t^l, v_t^l]$ is the observation of the same feature in the t image. π_c and π_c^{-1} are the projection and back-projection functions which depend on camera model (pinhole, omnidirectional or other models). $\mathbf{T}$ is the 4x4 homogeneous transformation, which is $\begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0} & 1 \end{bmatrix}$. We omit some homogeneous terms for concise expression. $\mathbf{T}_b^c$ is the extrinsic transformation from body center to camera center, which is calibrated offline. The covariance matrix $\mathbf{\Omega}_t^l$ of reprojection error is a constant value in pixel coordinate, which comes from the camera's intrinsic calibration results.

This factor is universal for both left camera and right camera. We can project a feature from the left image to the left image in temporal space, also we can project a feature from the left image to the right image in spatial space.

For different cameras, a different extrinsic transformation $\mathbf{T}_b^c$ should be used.

2) IMU Factor: We use the well-known IMU preintegration algorithm [11, 12] to construct the IMU factor. We assume that the additive noise in acceleration and gyroscope measurements are Gaussian white noise. The time-varying acceleration and gyroscope bias are modeled as a random walk process, whose derivative is Gaussian white noise. Since the IMU acquires data at a higher frequency than other sensors, there are usually multiple IMU measurements existing between two frames. Therefore, we pre-integrate IMU measurements on the manifold with covariance propagation. The detailed preintegration can be found at [12]. Within two time instants, $t-1$ and t, the preintegration produces relative position $\boldsymbol{\alpha}_t^{t-1}$, velocity $\boldsymbol{\beta}_t^{t-1}$ and rotation $\boldsymbol{\gamma}_t^{t-1}$. Also, the preintegration propagates the covariance of relative position, velocity, and rotation, as well as the covariance of bias. The IMU residual can be defined as:

$$\begin{gathered}\mathbf{z}_t^{imu} - h_t^{imu}(\mathcal{X}) = \\ \begin{bmatrix} \boldsymbol{\alpha}_t^{t-1} \\ \boldsymbol{\beta}_t^{t-1} \\ \boldsymbol{\gamma}_t^{t-1} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \ominus \begin{bmatrix} {\mathbf{R}_{t-1}}^{-1}(\mathbf{p}_t - \mathbf{p}_{t-1} + \frac{1}{2}\mathbf{g}dt^2 - \mathbf{v}_{t-1}dt) \\ {\mathbf{R}_{t-1}}^{-1}(\mathbf{v}_t - \mathbf{v}_{t-1} + \mathbf{g}dt) \\ {\mathbf{R}_{t-1}}^{-1}\mathbf{R}_t \\ \mathbf{b}_{a_t} - \mathbf{b}_{a_{t-1}} \\ \mathbf{b}_{g_t} - \mathbf{b}_{g_{t-1}} \end{bmatrix},\end{gathered} \tag{5}$$

where $\ominus$ is the minus operation on manifold, which is specially used for non-linear rotation. dt is the time interval between two time instants. g is the known gravity vector, whose norm is around 9.81. Every two adjacent frames construct one IMU factor in the cost function.

3) Other Factors: Though we only specify camera and IMU factors, our system is not limited to these two sensors. Other sensors, such as wheel speedometer, LiDAR and Radar, can be added into our system without much effort. The key is to model these measurements as general residual factors and add these residual factors into cost function.

C. Optimization

In traditional, the nonlinear least square problem of eq.3 is solved by Newton-Gaussian or Levenberg-Marquardt approaches. The cost function is linearized with respect to an initial guess of states, $\hat{\mathcal{X}}$. Then, the cost function is equals to:

$$\arg\min_{\delta\mathcal{X}} \sum_{t=0}^{n} \sum_{k\in\mathbf{S}} \left\| \mathbf{e}_t^k + \mathbf{J}_t^k \delta\mathcal{X} \right\|_{\boldsymbol{\Omega}_t^k}^2, \tag{6}$$

where $\mathbf{J}$ is the Jacobian matrix of each factor with respect to current states $\hat{\mathcal{X}}$. After linearization approximation, this cost function has closed-form solution of $\delta\mathcal{X}$. We take Newton-Gaussian as example, the solution is derived as follows,

$$\underbrace{\sum\sum {\mathbf{J}_t^k}^T {\boldsymbol{\Omega}_t^k}^{-1} \mathbf{J}_t^k}_{\mathbf{H}} \delta\mathcal{X} = \underbrace{-\sum\sum {\mathbf{J}_t^k}^T {\boldsymbol{\Omega}_t^k}^{-1} \mathbf{e}_t^k}_{\mathbf{b}}. \tag{7}$$

Finally, current state $\hat{\mathcal{X}}$ is updated with $\hat{\mathcal{X}} \oplus \delta\mathcal{X}$, where $\oplus$ is the plus operation on manifold for rotation. This procedure iterates several times until convergence. We adopt Ceres solver [24] to solve this problem, which utilizes advanced mathematical tools to get stable and optimal results efficiently.

D. Marginalization

Since the number of states increases along with time, the computational complexity will increase quadratically accordingly. In order to bound the computational complexity, marginalization is incorporated without loss of useful information. Marginalization procedure converts previous measurements into a prior term, which reserves past information. The set of states to be marginalized out is denoted as $\mathcal{X}_m$, and the set of remaining states is denoted as $\mathcal{X}_r$. By summing all marginalized factors (eq.7), we get a new $\mathbf{H}$ and $\mathbf{b}$. After rearrange states' order, we get the following relationship:

$$\begin{bmatrix} \mathbf{H}_{mm} & \mathbf{H}_{mr} \\ \mathbf{H}_{rm} & \mathbf{H}_{rr} \end{bmatrix} \begin{bmatrix} \delta\mathcal{X}_m \\ \delta\mathcal{X}_r \end{bmatrix} = \begin{bmatrix} \mathbf{b}_m \\ \mathbf{b}_r \end{bmatrix}. \tag{8}$$

The marginalization is carried out using the Schur complement [25] as follows:

$$\underbrace{(\mathbf{H}_{rr} - \mathbf{H}_{rm}{\mathbf{H}_{mm}}^{-1}\mathbf{H}_{mr})}_{\mathbf{H}_p} \delta\mathcal{X}_r = \underbrace{\mathbf{b}_r - \mathbf{H}_{rm}{\mathbf{H}_{mm}}^{-1}\mathbf{b}_m}_{\mathbf{b}_p}. \tag{9}$$

We get a new prior $\mathbf{H}_p, \mathbf{b}_p$ for the remaining states. The information about marginalized states is converted into prior term without any loss. To be specific, we keep ten spacial camera frames in our system. When a new keyframe comes, we marginalize out the visual and inertial factors, which are related with states of the first frame.

After we get the prior information about current states, with Bayes' rule, we could calculate the posterior as a product of likelihood and prior: $p(\mathcal{X}|\mathbf{z}) \propto p(\mathbf{z}|\mathcal{X})p(\mathcal{X})$. The state estimation then becomes a MAP (Maximum A Posteriori) problem. Denote that we keep states from instant m to instant n in the sliding window. The states before m are marginalized out and converted to a prior term. Therefore, the MAP problem is written as:

$$\begin{aligned} \mathcal{X}_{m:n}^* &= \arg\max_{\mathcal{X}_{m:n}} \prod_{t=m}^{n} \prod_{k\in\mathbf{S}} p(\mathbf{z}_t^k | \mathcal{X}_{m:n}) \;\; p(\mathcal{X}_{m:n}) \\ &= \arg\min_{\mathcal{X}_{m:n}} \sum_{t=m}^{n} \sum_{k\in\mathbf{S}} \left\| \mathbf{z}_t^k - h_t^k(\mathcal{X}_{m:n}) \right\|_{\boldsymbol{\Omega}_t^k}^2 \\ &\quad + (\mathbf{H}_p \delta\mathcal{X}_{m:n} - \mathbf{b}_p). \end{aligned} \tag{10}$$

Compared with eq.3, the above-mentioned equation only adds a prior term. It is solved as same as eq.3 by Ceres solver [24].

E. Discussion

The proposed system is a general framework. Various sensors can be easily added into our system, as long as it can be derived as a general residual factor. Since our system is not specially designed for a certain sensor, it is capable to handle sensor failure case. When sensor failure occurs, we just remove factors of the inactive sensor and add new factors from other alternative sensors.

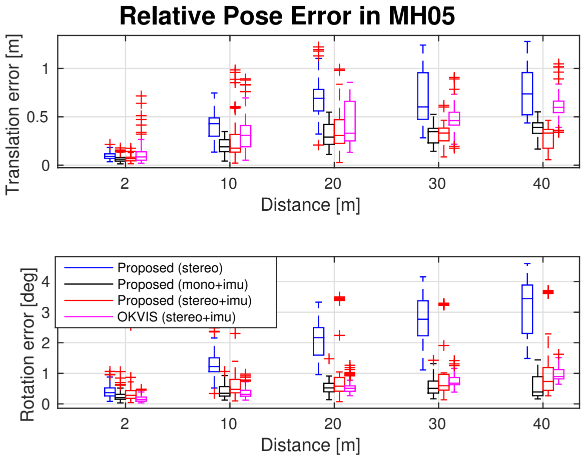

Fig. 4. Relative pose error [26] in MH_05_difficult. Two plots are relative errors in translation and rotation respectively.

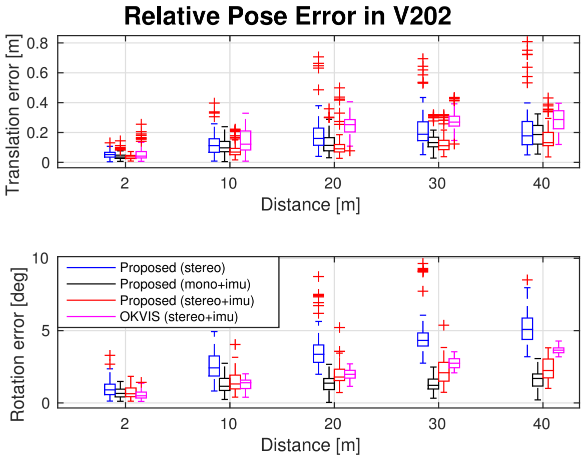

Fig. 5. Relative pose error [26] in V2_02_medium. Two plots are relative errors in translation and rotation respectively.

V. Experimental Results

We evaluate the proposed system with visual and inertial sensors both on datasets and with real-world experiments. In the first experiment, we compare the proposed algorithm with another state-of-the-art algorithm on public datasets. We then test our system in the large-scale outdoor environment. The numerical analysis is generated to show the accuracy of our system in detail.

A. Datasets

We evaluate our proposed system using the EuRoC MAV Visual-Inertial Datasets [27]. This datasets are collected onboard a micro aerial vehicle, which contain stereo images (Aptina MT9V034 global shutter, 752x480 monochrome, 20 FPS), synchronized IMU measurements (ADIS16448, 200 Hz), Also, the ground truth states are provided by VICON and Leica MS50. We run datasets with three different combinations of sensors, which are stereo cameras, a monocular camera with an IMU, stereo cameras with an IMU separately.

TABLE I

RMSE[m] in EuRoC dataset.

Sequence	Length	Proposed RMSE			OKVIS
		stereo	mono+imu	stereo+imu	RMSE
MH_01	79.84	0.54	0.18	0.24	**0.16**
MH_02	72.75	0.46	**0.09**	0.18	0.22
MH_03	130.58	0.33	**0.17**	0.23	0.24
MH_04	91.55	0.78	**0.21**	0.39	0.34
MH_05	97.32	0.50	0.25	**0.19**	0.47
V1_01	58.51	0.55	**0.06**	0.10	0.09
V1_02	75.72	0.23	**0.09**	0.10	0.20
V1_03	78.77	x	0.18	**0.11**	0.24
V2_01	36.34	0.23	**0.06**	0.12	0.13
V2_02	83.01	0.20	0.11	**0.10**	0.16
V2_03	85.23	x	**0.26**	0.27	0.29

In this experiment, we compare our results with OKVIS [8], a state-of-the-art VIO that works with stereo cameras and an IMU. OKVIS is another optimization-based sliding-window algorithm. OKVIS is specially designed for visual-inertial sensors, while our system is a more general framework, which supports multiple sensors combinations. We tested the proposed framework and OKVIS with all sequences in EuRoC datasets. We evaluated accuracy by RPE (Relative Pose Errors) and ATE (Absolute Trajectory Errors). The RPE is calculated by tools proposed in [26]. The RPE (Relative Pose Errors) plot of two sequences, MH_05_difficult and V2_02_medium, are shown in Fig. 4 and Fig. 5 respectively.

The RMSE (Root Mean Square Errors) of ATE for all sequences in EuRoC datasets is shown in Table. I. Estimated trajectories are aligned with the ground truth by Horn's method [28]. The stereo-only case fails in V1_03_difficult and V2_03_difficult sequences, where the movement is too aggressive for visual tracking to survive. Methods which involves the IMU work successfully in all sequences. It is a good case to show that the IMU can dramatically improve motion tracking performance by bridging the gap when visual tracks fail due to illumination change, textureless area, or motion blur.

From the relative pose error and absolute trajectory error, we can see that the stereo-only method performed worst in most sequences. Position and rotation drift obviously grown along with distance in stereo-only case. In other words, the IMU significantly benefited vision in states estimation. Since the IMU measures gravity vector, it can effectively suppress drifts in roll and pitch angles. Stereo cameras with an IMU didn't always perform best, because it requires more accurate calibration than the case of a monocular camera with an IMU. Inaccurate intrinsic and extrinsic calibration will introduce more noise into the system. In general, multiple sensor fusion increase the robustness of the system. Our results outperforms OKVIS in most sequences.

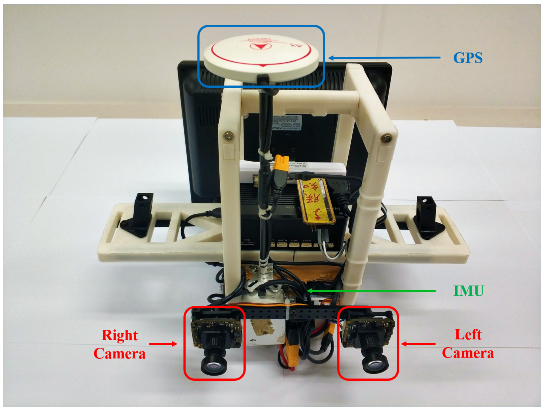

Fig. 6. The self-developed sensor suite used in the outdoor environment. It contains stereo cameras (mvBlueFOX-MLC200w, 20Hz) and DJI A3 controller, which include inbuilt IMU (200Hz) and GPS receiver.

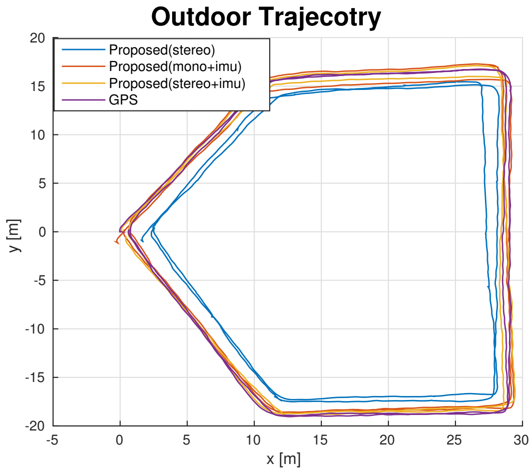

Fig. 7. Estimated trajectories in outdoor experiment.

B. Real-world experiment

In this experiment, we used a self-developed sensor suite to demonstrate our framework. The sensor suite is shown in Fig. 6. It contains stereo cameras (mvBlueFOX-MLC200w, 20Hz) and DJI A3 controller[2], which inculdes inbuilt IMU (200Hz) and GPS receiver. The GPS position is treated as ground truth. We hold the sensor suite by hand and walk around on the outdoor ground. We run states estimation with three different combinations, which are stereo cameras, a monocular camera with an IMU, and stereo cameras with an IMU.

[2] http://www.dji.com/a3

TABLE II
RMSE[M] IN OUTDOOR EXPERIMENT.

Sequence	Length	Proposed RMSE		
		stereo	mono+imu	stereo+imu
outdoor1	223.70	1.85	0.71	**0.52**
outdoor2	229.91	2.35	0.56	**0.43**
outdoor3	232.13	2.59	**0.65**	0.75

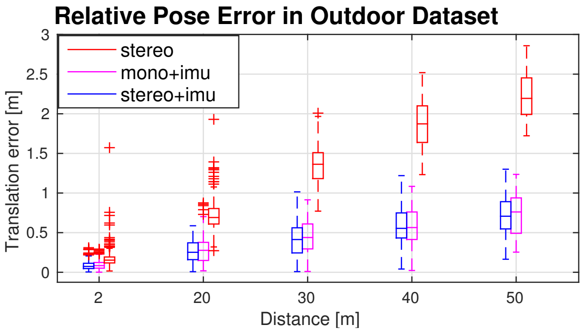

Fig. 8. Relative pose error [26] in outdoor experiment.

For accuracy comparison, we walked two circles on the ground and compared our estimation with GPS. The trajectory is shown in Fig. 7, and the RPE (Relative Pose Error) is shown in Fig. 8. As same as dataset experiment, noticeable position drifts occurred in the stereo-only scenario. With the assistance of the IMU, the accuracy improves a lot. The RMSE of more outdoor experiments is shown in Table. II. The method which involves the IMU always performs better than the stereo-only case.

VI. Conclusion

In this paper we have presented a general optimization-based framework for local pose estimation. The proposed framework can support multiple sensor combinations, which is desirable in aspect of robustness and practicability. We further demonstrate it with visual and inertial sensors, which form three sensor suites (stereo cameras, a monocular camera with an IMU, and stereo cameras with an IMU). Note that although we only show the factor formulations for the camera and IMU, our framework can be generalized to other sensors as well. We validate the performance of our system with multiple sensors on both public datasets and real-world experiments. The numerical result indicates that our framework is able to fuse sensor data with different settings.

In future work, we will extend our framework with global sensors (e.g. GPS) to achieve locally accurate and globally aware pose estimation.

References

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

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

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

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

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

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

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

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

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

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

[11] 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, 2017.

[12] 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, 2018.

[13] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems*, vol. 2, 2014, p. 9.

[14] C. Kerl, J. Sturm, and D. Cremers, "Dense visual slam for rgb-d cameras," in *Proc. of the IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*

[15] H. Rebecq, T. Horstschaefer, G. Gallego, and D. Scaramuzza, "Evo: A geometric approach to event-based 6-dof parallel tracking and mapping in real time," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 593–600, 2017.

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

[17] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Observability-based rules for designing consistent ekf slam estimators," *Int. J. Robot. Research*, vol. 29, no. 5, pp. 502–528, 2010.

[18] 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. of the IEEE Int. Conf. on Robot. and Autom.*, Hong Kong, China, May 2014, pp. 4974–4981.

[19] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Proc. of the IEEE Int. Conf. on Robot. and Autom.* IEEE, 2011, pp. 3607–3613.

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

[21] H. Liu, M. Chen, G. Zhang, H. Bao, and Y. Bao, "Ice-ba: Incremental, consistent and efficient bundle adjustment for visual-inertial slam," in *Proc. of the IEEE Int. Conf. on Pattern Recognition*, 2018, pp. 1974–1982.

[22] J. Shi and C. Tomasi, "Good features to track," in *Computer Vision and Pattern Recognition,1994. IEEE Computer Society Conference on*, 1994, pp. 593–600.

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

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

[25] G. Sibley, L. Matthies, and G. Sukhatme, "Sliding window filter with application to planetary landing," *J. Field Robot.*, vol. 27, no. 5, pp. 587–608, Sep. 2010.

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

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

[28] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *JOSA A*, vol. 4, no. 4, pp. 629–642, 1987.