2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

An Observability Constrained Downward-Facing Optical-Flow-Aided Visual-Inertial Odometry

Dandi Liu[1], Jiahao Mei[2], Jin Zhou[1], and Shuo Li[1]

***Abstract*— Visual-Inertial Odometry (VIO) has been widely used by autonomous drones as an onboard navigation method. However, it suffers from drifts especially in scenarios where the environments have few texture features such as an empty room with solid color walls. Optical flow sensors are another type of onboard sensor used by drones that face downward and measure the velocity by detecting changes in pixels between consecutive images, which don't introduce accumulative error. In this work, we present an efficient tight-coupled estimator to improve the accuracy of VIO by fusing the measurements of a downward-facing optical flow sensor into the VIO framework consistently. We further analyze the observability of the estimators and prove that there are four unobservable directions in the ideal case and then we utilize OC-EKF to maintain the consistency of the estimator. Furthermore, we extend an adaptive weighting algorithm to the proposed method, which can better adapt to the scenes where feature tracking is less accurate. Finally, both simulation and real-world experiments demonstrate the feasibility of the proposed method.**

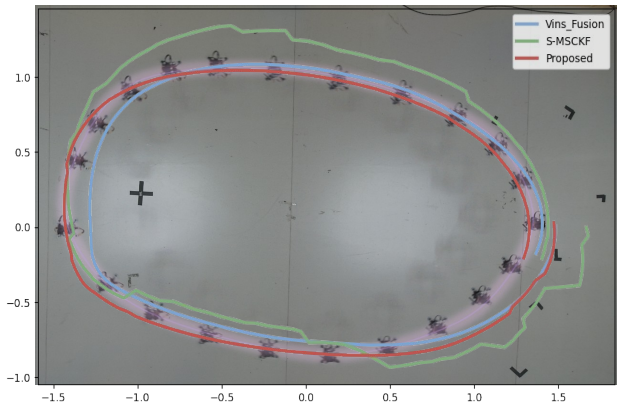


Fig. 1. The estimated trajectories by the proposed method and the benchmarks.

I. INTRODUCTION

Localization techniques are one of the key elements for drones to fly autonomously. In outdoor environments, they can use the Global Positioning System (GPS) [1] to localize themselves. With the development of the hardware, the weight and size of the drones keep decreasing and micro aerial vehicles (MAVs) start to show their potential application value, especially in indoor environments such as search and rescue and warehouse inspection [2], [3].

The GPS-denied indoor environments make it difficult for drones to navigate themselves. External equipment such as motion capture systems [4] or Ultra Wide-Band (UWB) Indoor Positioning System [5] can be used to provide position measurements for drones. However, the complex installation/calibration procedure or the high expense make them unsuitable to be widely used by autonomous flights. The simultaneous localization and mapping (SLAM) technique is widely used as the main localization method by different types of robots [6]. Nevertheless, it is so computationally expensive that the MAVs can hardly provide enough onboard computational resources. A commonly used approach by autonomous drones is visual-inertial odometry (VIO) which utilizes onboard cameras to track features and Inertial Measurement Units (IMUs) to integrate inertial information to estimate the drones' states. Compared to the SLAM, VIO does not do the loop closure detection which makes them relatively computationally affordable. However, these methods suffer from drifts, especially in long-time flights. Thus, in this work, we develop a computationally efficient method for a lightweight quadrotor that only uses onboard sensors and computing resources to localize itself in GPS-denied environments.

Current VIO methods can be divided into two categories. The first one is optimization-based methods including OKVIS [7], VINS-Mono [8], and VINS-Fusion [9], which typically employ the keyframe selection and marginalization strategy to maintain a bounded-sized optimization window and obtain the optimal estimate poses by solving the nonlinear optimization problems. Some strategies such as delayed marginalization are applied to improve the precision of scale estimation further [10]. However, due to the requirement of solving the complex nonlinear optimization problem, optimization-based methods usually require heavy onboard computing resources. The other category is the filtering-based methods which estimate the states by constructing the visual measurement model and performing the EKF updates. For example, the Multi-State Constraint Kalman Filter (MSCKF) was first proposed by Mourikis et al. [11], which solves the problem of dimensionality explosion and can be run onboard the quadrotors. This method was then extended to onboard stereo cameras by Sun et al. (S-MSCKF) [12] and can successfully run with limited computational resources by restricting stereo-matching direction [13]. Nevertheless, the filter-based methods have relatively large linearization errors [14] and a higher reliance on the accuracy of 3D

[1]Authors are with the College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China shuo.li@zju.edu.cn
[2]Jiahao Mei is with the Department of Automation, Zhejiang University of Technology, Hangzhou 310023, China.
This work was supported in part by NSFC under Grants 62203385, 62088101.

feature points estimation, which may lead to the drift on long trajectories. More importantly, when in low-texture environments like an empty room with solid color walls, the forward-facing cameras may fail to extract enough features, which may further affect the estimation accuracy. Thus, another lightweight and energy-efficient onboard sensor is needed as a complement to help the quadrotor localize itself.

Optical flow sensors are lightweight sensors that are usually mounted on the bottom of the quadrotors to measure the velocity of the quadrotors. Instead of facing forward, they extract the features on the ground and estimate the velocities by calculating pixel changes in consecutive images. They are widely used by quadrotors in different applications. For example, de Croon et al. use the optical flow to extract attitude information to control a quadrotor [15]. An optical flow deck is used in [2] to close the velocity control loop and then the quadrotors can execute a search and rescue mission [2]. They can also be used to help the quadrotors land on the ground [16], [17] or predict the velocities and positions of surrounding obstacles [18]. In addition, Li et al. analyzed the measurement model of the optical flow sensor and presented a cubature transform-based data fusion scheme to augment the UAV's position estimation [19]. However, in the applications mentioned above, the optical flow provides the velocity information or attitude information instead of the position information. It is difficult for the quadrotor to localize itself only using the optical flow as integrating the velocity from the optical flow can lead to heavy drift in position estimation. Thus, we propose a method that extends S-MSCKF [12] fusing the optical flow sensor data, to improve the performance of the VIO system. In this work, we

1) propose a filter-based tight-coupled estimator to integrate optical flow sensor measurements into the VIO framework.
2) analyze the observability of the proposed estimators and employ a new observation matrix to maintain the consistency of the estimator.
3) develop an online calibration method to estimate the extrinsic between the optical flow sensor and IMU.
4) introduce an adaptive weighting method to the proposed estimator to adapt to scenarios where feature tracking is challenging.
5) validate and analyze the estimator through both simulation and real-world experiments.

II. METHODOLOGY

In this section, we provide a detailed description of the design of the proposed estimator, which fuses the measurements of the IMU, stereo camera, and optical flow sensor. We first define the coordinate frames in our system, which include VIO frame {V}, the frame of the reference of the optical flow measurements {F}, optical flow sensor frame {S}, IMU frame {I}, and camera sensor frame {C} (Fig. 2 (b)). In addition, we define quaternion ${}^b_a\mathbf{q}$ and rotation matrix ${}^b_a\mathbf{R}$ to be the transformation from frame a to b while $C(\cdot)$ converts the quaternion ${}^b_a\mathbf{q}$ to the corresponding rotation matrix ${}^b_a\mathbf{R}$. The notion $\hat{\mathbf{x}}$ represents the estimate of $\mathbf{x}$ and the error of $\hat{\mathbf{x}}$ is defined as $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$.

A. Design of Estimator

The state vector of the estimator at timestamp k can be presented as

$$ {}^V\mathbf{x_k} = \begin{bmatrix} {}^V\mathbf{x}^\top_{IMU,k} & {}^V\mathbf{x}^\top_{CAM,k} \end{bmatrix}^\top \tag{1} $$

which is composed of two components: the IMU state vector $\mathbf{x}_{IMU,k}$, the historical camera pose state vector $\mathbf{x}_{CAM,k}$. More specifically,

$$ {}^V\mathbf{x}_{IMU,k} = \begin{bmatrix} {}^{I_k}_V\mathbf{q} & \mathbf{b}_g & {}^V\mathbf{v}_{I_k} & \mathbf{b}_a & {}^V\mathbf{p}_{I_k} & {}^I_C\mathbf{q} & {}^I\mathbf{p}_C & {}^F_V\mathbf{q} \end{bmatrix}^\top $$
$$ {}^V\mathbf{x}_{CAM,k} = \begin{bmatrix} {}^{C_{k-1}}_V\mathbf{q} & {}^V\mathbf{p}_{C_{k-1}} & \cdots & {}^{C_{k-N}}_V\mathbf{q} & {}^V\mathbf{p}_{C_{k-N}} \end{bmatrix}^\top $$

where ${}^{I_k}_V\mathbf{q}$ is the rotation from the VIO frame to the IMU frame at time-step k; ${}^V\mathbf{p}_{I_k}$ and ${}^V\mathbf{v}_{I_k}$ represent the position and the velocity of the IMU in the VIO frame respectively; $\mathbf{b}_g$ and $\mathbf{b}_a$ denote the gyroscope and accelerator biases; $\left[{}^V_{C_m}\mathbf{q}, {}^V\mathbf{p}_{C_m}\right] (m = k-1, \cdots, k-N)$ are N historical camera poses in a sliding window; ${}^I\mathbf{p}_C$ and ${}^I_C\mathbf{q}$ are camera extrinsic with respect to the IMU frame; Different to the MSCKF, we add ${}^F_V\mathbf{q}$ to our states which are the transform from VIO frame to optical flow sensor frame. Similarly, the prediction model can be written as

$$ \begin{aligned} {}^{I_k}_V\dot{\hat{\mathbf{q}}} &= \mathbf{\Omega}\left(\boldsymbol{\omega}_m - \hat{\mathbf{b}}_g\right)/2 \otimes {}^{I_k}_V\hat{\mathbf{q}} \\ {}^V\dot{\hat{\mathbf{v}}}_{I_k} &= C({}^{I_k}_V\hat{\mathbf{q}})^\top\left(\mathbf{a}_m - \hat{\mathbf{b}}_a\right) + \mathbf{g} \\ \dot{\hat{\mathbf{b}}}_g &= \mathbf{0}_3, \quad \dot{\hat{\mathbf{b}}}_a = \mathbf{0}_3 \\ {}^V\dot{\hat{\mathbf{p}}}_{I_k} &= {}^V\hat{\mathbf{v}}_{I_k}, \quad {}^I_C\dot{\hat{\mathbf{q}}} = \mathbf{0}_3 \\ {}^I\dot{\hat{\mathbf{p}}}_C &= \mathbf{0}_3, \quad {}^F_V\dot{\hat{\mathbf{q}}} = \mathbf{0}_3 \end{aligned} \tag{2} $$

where $\boldsymbol{\omega}_m$ and $\mathbf{a}_m$ are the measurements of accelerometer and gyroscope, $\mathbf{\Omega}$ represents the transform from $\mathbb{R}^3$ to a pure imaginary quaternion. Thus, the linearized model for predicting the error-state of IMU $\tilde{\mathbf{x}}_{IMU}$ can be written as:

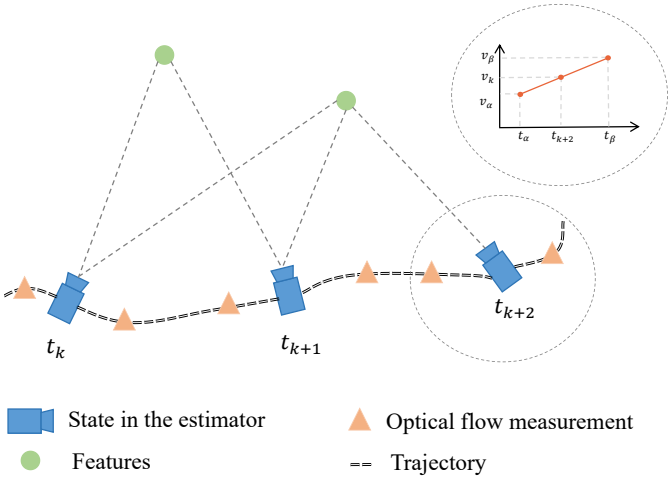
$$ \dot{\tilde{\mathbf{x}}}_{IMU} = \mathbf{F}\tilde{\mathbf{x}}_{IMU} + \mathbf{G}\mathbf{n}_{IMU} \tag{3} $$

where

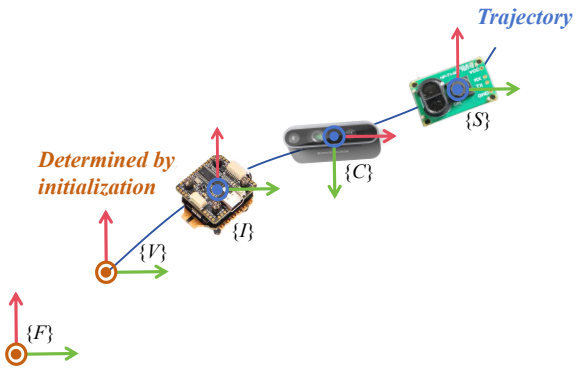
$$ \mathbf{F} = \begin{bmatrix} & & \mathbf{0}_3 \\ & \mathbf{F}_{msckf} & \vdots \\ \mathbf{0}_3 & \cdots & \mathbf{0}_3 \end{bmatrix}_{24\times 24} \tag{4} $$

$$ \mathbf{G} = \begin{bmatrix} & & \\ & \mathbf{G}_{msckf} & \\ \mathbf{0}_3 & \cdots & \mathbf{0}_3 \end{bmatrix}_{24\times 12} \tag{5} $$

and $\mathbf{n}_{IMU} = \begin{bmatrix} \mathbf{n}_g & \mathbf{n}_{\omega_g} & \mathbf{n}_a & \mathbf{n}_{\omega_a} \end{bmatrix}$. $\mathbf{n}_g, \mathbf{n}_a$ are white gaussian noises for IMU measurements, and $\mathbf{n}_{b_a}, \mathbf{n}_{b_g}$ are bias random walk of accelerometer and gyroscope. Since states ${}^F_V\mathbf{q}$ are added to (2), the corresponding columns and rows are added to matrix $\mathbf{F}$ and $\mathbf{G}$ based on the original matrices $\mathbf{F}_{msckf}$ and $\mathbf{G}_{msckf}$ in [11]. Then we can calculate



(a) Illustration of the design of the estimator.



(b) The used sensors and the definition of the frames.

Fig. 2. Proposed optical-flow-aided VIO system and the basic definition mentioned in this paper.

the transition matrix $\mathbf{\Phi}_k$ and the covariance matrix $\mathbf{P}_{k+1|k}$ as in a standard EKF predict procedure.

For updating the states by the visual measurements from the onboard stereo camera, the procedure is the same as in [12]. For the readers' convenience, we put the key elements here and the readers are referred to [12] for more details. The measurement model with the j-th feature is constructed and the residual is defined as

$$\mathbf{r}_c = \mathbf{H}_{V,x}\tilde{\mathbf{x}} + \mathbf{H}_{V,f}{}^V\tilde{\mathbf{p}}_f + \mathbf{n}_f \tag{6}$$

where $\mathbf{n}_f$ is the zero-mean white Gaussian measurement noise of visual measurements, and ${}^V\tilde{\mathbf{p}}_f$ represents the error state of tracked feature position in the VIO frame. Since ${}^V\tilde{\mathbf{p}}_f$ is correlated with $\tilde{\mathbf{x}}$, the feature position in the measurement model is linearly marginalized by left null space $\mathbf{V}^T$ of the Jacobian $\mathbf{H}_{V,f}$ and the residual model is rewritten as

$$\begin{aligned}\mathbf{r}_o &= \mathbf{V}^\top\mathbf{r}_c = \mathbf{V}^\top\mathbf{H}_{V,x}\tilde{\mathbf{x}} + \mathbf{V}^\top\mathbf{n}_f \\ &= \mathbf{H}_{V,x,o}\tilde{\mathbf{x}} + \mathbf{n}_o\end{aligned} \tag{7}$$

The visual update occurs when the number of states in the sliding window reaches the threshold or there exists a feature that is no longer tracked.

As mentioned before, the forward-facing stereo camera may suffer from low-texture environments like an empty room with solid color walls. Thus, a downward-facing optical flow sensor is used to provide the velocity measurements as a complement which are fused to the MSCKF. However, the stereo camera and the optical flow sensor have different frequencies. Thus, we need to first synchronize the measurements of the two resources and when a new image is captured by the stereo camera, we do the visual and optical flow update. To synchronize the two sensors, we employ linear interpolation to estimate the optical flow measurement at time step t_k [20]. Assuming that we receive stereo images at times t_k, and the optical flow measurements are obtained at t_α and t_β, where $t_\alpha < t_k < t_\beta$ (Fig. 2 (a)), the aligned optical flow measurement at time t_k is calculated by:

$${}^F\mathbf{v}_k = \lambda{}^F\mathbf{v}_\beta + (1-\lambda){}^F\mathbf{v}_\alpha \tag{8}$$

where

$$\lambda = \frac{t_k - t_\alpha + \Delta t_F}{t_\beta - t_\alpha}$$

and Δt_F is the time offset between the optical flow sensor and IMU, which will be estimated during the procedure, the details of which can be found in II-B. The optical flow residual model depends on the IMU states ${}^V\mathbf{x}_{IMU,k}$, and the extrinsic ${}^F_V\mathbf{R}$ between the IMU and optical flow sensor.

$$\begin{aligned}\tilde{\mathbf{z}}_{F,k} &= \mathbf{z}_{F,k} - \hat{\mathbf{z}}_{F,k} \\ &= {}^F\mathbf{v}_k - {}^F_V\hat{\mathbf{R}}{}^V\hat{\mathbf{v}}_{I_k} + \mathbf{n}_F\end{aligned} \tag{9}$$

where $\mathbf{n}_F$ is the noise of optical flow sensor's measurements. The Jacobian matrix is computed by

$$\begin{aligned}\frac{\partial\tilde{\mathbf{z}}_{F,k}}{\partial{}^V\tilde{\mathbf{v}}_k} &= {}^F_V\hat{\mathbf{R}} \\ \frac{\partial\tilde{\mathbf{z}}_{F,k}}{\partial{}^F_V\tilde{\theta}} &= \lfloor {}^F_V\hat{\mathbf{R}}{}^V\hat{\mathbf{v}}_k\times\rfloor\end{aligned} \tag{10}$$

and the observation matrix can be written as

$$\hat{\mathbf{H}}_{F,k} = \begin{bmatrix}\mathbf{0}_{3\times6} & {}^F_V\hat{\mathbf{R}} & \mathbf{0}_{3\times9} & \lfloor {}^F_V\hat{\mathbf{R}}{}^V\hat{\mathbf{v}}_{I_k}\times\rfloor\end{bmatrix} \tag{11}$$

where $\lfloor\cdot\times\rfloor$ is the skew symmetric matrix. With the propagation model (2) the observation matrices (7) and (9), the extended Kalman Filtered can be run to estimate the states of the quadrotor.

However, it is proved that in a standard visual-inertial odometry system, there are four unobservable directions: three-d.o.f. global translations and one-d.o.f. global rotation about the gravity vector [21]. But in the real world, one direction is always observable due to the use of the linearized states at different stages when calculating the Jacobian matrix $\hat{\mathbf{H}}_{F,k}$ and the transition matrix $\hat{\mathbf{\Phi}}_k$, leading to less accurate estimation. Since we add the optical flow sensor to the

estimation, we also need to first analyze the observability of the new estimation system to see if this observation problem also exists. To simplify the analysis, we reformulate the state vector, which includes the IMU states, and the j-th feature.

$$\mathbf{x}_k = \begin{bmatrix} {}^{I_k}_{V}\mathbf{q} & {}^{V}\mathbf{p}_{I_k} & {}^{V}\mathbf{v}_{I_k} & {}^{V}\mathbf{p}_{f_j} & {}^{F}_{V}\mathbf{q} \end{bmatrix}^\top \tag{12}$$

The state transition matrix from timestamp t_1 to t_k can be calculated in the ideal case

$$\mathbf{\Phi}_{k,1} = \begin{bmatrix} \mathbf{\Phi}_1 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_{3\times 6} \\ \mathbf{\Phi}_2 & \mathbf{I}_3 & \Delta t\mathbf{I}_3 & \mathbf{0}_{3\times 6} \\ \mathbf{\Phi}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_{3\times 6} \\ \mathbf{0}_{6\times 3} & \mathbf{0}_{6\times 3} & \mathbf{0}_{6\times 3} & \mathbf{I}_6 \end{bmatrix} \tag{13}$$

where

$$\begin{aligned} \mathbf{\Phi}_1 &= {}^{I_k}_{V}\mathbf{R}\ {}^{I_1}_{V}\mathbf{R}^\top \\ \mathbf{\Phi}_2 &= -\left\lfloor {}^{V}\mathbf{p}_{I_k} - {}^{V}\mathbf{p}_{I_1} - {}^{V}\mathbf{v}_{I_k}\Delta t + \frac{1}{2}\mathbf{g}\Delta t^2 \times \right\rfloor {}^{I_1}_{V}\mathbf{R}^\top \\ \mathbf{\Phi}_3 &= -\left\lfloor {}^{V}\mathbf{v}_{I_k} - {}^{V}\mathbf{v}_{I_1} + \mathbf{g}\Delta t \times \right\rfloor {}^{I_1}_{V}\mathbf{R}^\top \\ \Delta t &= t_k - t_1 \end{aligned}$$

We rewrite the optical flow measurement jacobians in the ideal case:

$$\mathbf{H}_{F,k} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & {}^{F}_{V}\mathbf{R} & \mathbf{0}_3 & \lfloor {}^{F}_{V}\mathbf{R}\,{}^{V}\mathbf{v}_{I_k} \times \rfloor \end{bmatrix} \tag{14}$$

We can construct the observability matrix $\mathbf{M}$ and matrix $\mathbf{N}$, which is the right nullspace of the observability matrix.

$$\mathbf{M} = \begin{bmatrix} \vdots \\ \mathbf{H}_{F,k}\mathbf{\Phi}_{k,1} \\ \vdots \end{bmatrix} \tag{15}$$

$$\mathbf{N}_1 = \mathbf{null}\left(\mathbf{M}\right) = \begin{bmatrix} \mathbf{0}_3 & -{}^{I_1}_{V}\mathbf{R}\mathbf{g} \\ \mathbf{I}_3 & \lfloor {}^{V}\mathbf{p}_{I_1} \times \rfloor \mathbf{g} \\ \mathbf{0}_3 & \lfloor {}^{V}\mathbf{v}_{I_1} \times \rfloor \mathbf{g} \\ \mathbf{I}_3 & \lfloor {}^{V}\mathbf{p}_{f_j} \times \rfloor \mathbf{g} \\ \mathbf{0}_3 & -{}^{F}_{V}\mathbf{R}\mathbf{g} \end{bmatrix} = \begin{bmatrix} \mathbf{N}_{1,t} & \mathbf{N}_{1,r} \end{bmatrix} \tag{16}$$

where $\mathbf{N}_{1,t}$ and $\mathbf{N}_{1,r}$ are the global translations and rotation. It can be demonstrated that $\mathbf{H}_{F,k}\mathbf{\Phi}_{k,1}\mathbf{N}_1 = \mathbf{0}$, which illustrates that the proposed estimator also has four unobservable directions after fusing the optical flow measurements in the ideal case. However, in the real-world implementation where optical flow measurement matrix $\hat{\mathbf{H}}_{F,k}$ are calculated with the estimated states $\hat{\mathbf{x}}_{k+1|k}$, the error $\hat{\mathbf{v}}_{I_{k|k}} - \hat{\mathbf{v}}_{I_{k+1|k}}$ can make $\hat{\mathbf{H}}_{F,k}\hat{\mathbf{\Phi}}_{k,1}\hat{\mathbf{N}}_{1,r} \neq \mathbf{0}$ and the rank of observability matrix $\hat{\mathbf{M}}_k$ is as a result increased by one. Thus, it is necessary to maintain unobservability in four directions, avoiding the introduction of erroneous information. We choose OC-EKF [21] to maintain the observability of the system by modifying the measurement jacobians $\hat{\mathbf{H}}_{F,k}$ to satisfy the constraints of the ideal situation. During each optical flow update step, we solve the corresponding KKT optimality conditions and obtain the modified $\mathbf{H}^*_{F,k}$ by:

$$\mathbf{H}^*_{F,k} = \hat{\mathbf{H}}_{F,k} - \hat{\mathbf{H}}_{F,k}\hat{\mathbf{N}}_k\left(\hat{\mathbf{N}}_k^\top\hat{\mathbf{N}}_k\right)^{-1}\hat{\mathbf{N}}_k^\top \tag{17}$$

In this way, the unobservable space of the estimated model is enforced to align with that of the ideal case. This avoids the gain of spurious information and inconsistency in the estimation process is reduced.

B. Calibration of Estimator

As mentioned above, to perform the Extended Kalman Filter (EKF) update with optical flow measurements, it is necessary to estimate the transformation ${}^{F}_{V}\mathbf{q}$ and the time offset Δt_F. To achieve this, we introduce a sliding window that records historical velocity measurements from the optical flow sensor and the corresponding velocities computed by the estimator. And then solve the nonlinear optimization problem

$$\min_{{}^{F}_{V}\mathbf{q},\Delta t_F} \left\{ \sum_{k=0}^{n} \| {}^{F}\mathbf{v}_{k,m} - {}^{F}_{V}\hat{\mathbf{R}}\,{}^{V}\hat{\mathbf{v}}_k \|^2 \right\} \tag{18}$$

where n is the number of states in the sliding window, ${}^{F}\hat{\mathbf{v}}_k$ is the velocity at the k-th frame and ${}^{V}\mathbf{v}_{k,m}$ is the corresponding velocity measurement of optical flow sensor and ${}^{F}\mathbf{v}_{k,m} = {}^{F}_{V}\hat{\mathbf{R}}\,{}^{V}\hat{\mathbf{v}}_k$.

In the real-world experiment, we find that during the static initialization process of the quadrotor, the estimation of the accelerometer bias is inaccurate. To address this, we begin recording historical states when the estimator starts to remove the oldest frame. We solve the optimization problem once the number of states in the sliding window reaches a predetermined threshold. This approach allows us to obtain a more accurate estimation of rotation ${}^{F}_{V}\mathbf{q}$ and time offset Δt_F.

C. Adaptive Weighting

In the standard MSCKF framework, the noise parameters of the feature and the IMU are assumed to be constant and manually tuned. However, in scenarios with few texture features, the number of detected features will significantly decrease, resulting in a reduction in the tracking quality of the feature points. Trusting all feature points equally may lead to poor performance. Therefore, it becomes crucial to evaluate the confidence level of each feature point. To evaluate the confidence level of each feature, we extend the method proposed in [22] that is used by optimization-based VIO algorithm to our method.

The error in the propagation of image measurements to pose parameters is calculated by the covariance matrix below:

$$\begin{aligned} \mathbf{\Sigma}_\theta &= \left(\mathbf{J}^T\mathbf{J}\right)^{-1}\mathbf{J}^T\mathbf{\Sigma}_I\mathbf{J}\left(\mathbf{J}^T\mathbf{J}\right)^{-1} \\ &= \sigma_I^2\left(\sum_{h=1}^{q}\mathbf{J}_h^\top\mathbf{J}_h\right)^{-1} \end{aligned} \tag{19}$$

where q is the number of feature points utilized in the update, $\mathbf{J}_h$ is the partial derivative of the visual reprojection error to the pose, $\mathbf{\Sigma_I}$ is the covariance matrix of the image measurements, and $\mathbf{\Sigma}_\theta$ represents the confidence ellipsoid of the pose space. The average radius of the confidence ellipsoid is commonly used to evaluate the quality of the

pose covariance matrix [23], which is calculated by $\eta = (\alpha)^{1/2} (det\mathbf{\Sigma}_\theta)^{1/12}$, where α is a constant. The smaller the average radius is, the higher the confidence level is.

In the standard MSCKF [11], the covariance matrix $\mathbf{R}_n$ of the noise vector is calculated by:

$$\mathbf{R}_n = \mathbf{Q}_1^T \mathbf{R}_o \mathbf{Q}_1 = \sigma_{\text{im}}^2 \mathbf{I}_r \tag{20}$$

where $\mathbf{I}_r$ is an Identity matrix and $\mathbf{R}_n$ is modeled as zero-mean, white, and uncorrelated Gaussian noise. So, we can adjust the value of σ_{im} to change the confidence of each feature. With the higher noise parameters, we trust this feature less. Hence, the $\mathbf{R}_n$ matrix can be written as:

$$\mathbf{R}_n = \begin{bmatrix} \omega_{h_1}\mathbf{I}_{h_1} & & & \\ & \omega_{h_2}\mathbf{I}_{h_2} & & \\ & & \ddots & \\ & & & \omega_{h_q}\mathbf{I}_{h_q} \end{bmatrix} \tag{21}$$

$$\omega_j = \left(\det \mathbf{\Sigma}_\theta^j\right)^{1/12} \Lambda \tag{22}$$

where Λ is a parameter to be tuned based on the quality of the camera, and ω_j is positively related to the average radius of the confidence ellipsoid [23]. A large ω indicates a low confidence level of the feature. Every time the feature tracking is lost, we compute the pose covariance matrix $\mathbf{\Sigma}_\theta$ and set different weights for each feature to prevent over-reliance on error messages. If the QR decomposition is employed in the update step, we will construct the square matrix using the first n rows of $\mathbf{R}_n$ to ensure the dimensional consistency required for matrix multiplication.

III. Simulation Result and Analysis

In this section, we test the proposed estimator with an open-source dataset, the EuRoC dataset [24], which provides the ground truth data, the synchronized stereo images with the frequency of 20Hz, and the IMU data with the frequency of 200Hz. We will compare its performance with that of the benchmark S-MSCKF [12]. The index of evaluating the performance is the root-mean-square error (RMSE) of the absolute trajectory error (ATE) using the RPG Trajectory Evaluation toolbox [25].

$$\begin{aligned} \text{RMSE}_{\text{ATE,rot}} &= \left(\frac{1}{N} \sum_{i=0}^{N} \left\| \angle \Delta\mathbf{R}_i \right\|^2 \right)^{\frac{1}{2}} \\ \text{RMSE}_{\text{ATE,tran}} &= \left(\frac{1}{N} \sum_{i=0}^{N} \left\| \Delta\mathbf{p}_i \right\|^2 \right)^{\frac{1}{2}} \end{aligned} \tag{23}$$

Eq. 23 represents the error of rotation and translation over the whole trajectory respectively. Readers are referred to [25] for the details of the index.

As the EuRoC dataset does not have the optical flow sensor measurements, we will generate the optical flow sensor data $\mathbf{v}_{k,m}$ with measurement noise $\Delta\mathbf{v}_k$ based on the ground truth data by

$$\begin{aligned} \mathbf{v}_{k,m} &= \mathbf{v}_k^{gt} + \Delta\mathbf{v}_k \\ \Delta\mathbf{v}_k &\sim \mathbf{N}(\mathbf{0}, \mathbf{S}_k) \\ \mathbf{S}_k &= \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix} \end{aligned} \tag{24}$$

where $\mathbf{v}_{k,m} = [v_{k,m}^x, v_{k,m}^y, v_{k,m}^z]$ is the generated optical flow sensor's measurements, $\mathbf{v}_k^{gt}$ is the ground truth velocity in the dataset and σ^2 is used to model the noise level of the optical flow sensor. Besides adding the noise to the ground truth as in (24), we also take into account adding the rotation ${}_V^F\mathbf{R}$ between frame V and frame F and the optical flow sensor's frequency f_{of}, the time offset Δt_F to the ground truth data to simulate the real-world optical flow measurements.

A. Simulation of Calibration and Different Sensor Noise

We first conduct an experiment to test the proposed estimator and the calibration method described in II-B using dataset V1_03 which has fast maneuvers and challenging motion blurs. In this experiment, we set three Euler angles between frame V and frame F by $\phi = 30°$, $\theta = 40°$, $\psi = 50°$. The time offset Δt_F is set as $0.05s$ and the frequency of the optical flow sensor f_{of} is 10Hz. The initial values of the optimization problem (18) are all set to 0.

The non-linear optimization problem is solved using the Ceres-Solver, an open-source C++ library specifically designed for modeling and solving different types of optimization problems [26]. Fig. 3 shows the optimization/calibration result of the estimated Euler angles between frame V and frame F and the time offset Δt_F with different σ^2. After $2.0s$, the quadrotor begins to maneuver, and the calibration process starts to run periodically to calculate ${}_V^F\mathbf{q}$ and Δt_f. We can find that when $\sigma^2 < 0.5m/s$, the solver can find the correct poses between two frames.

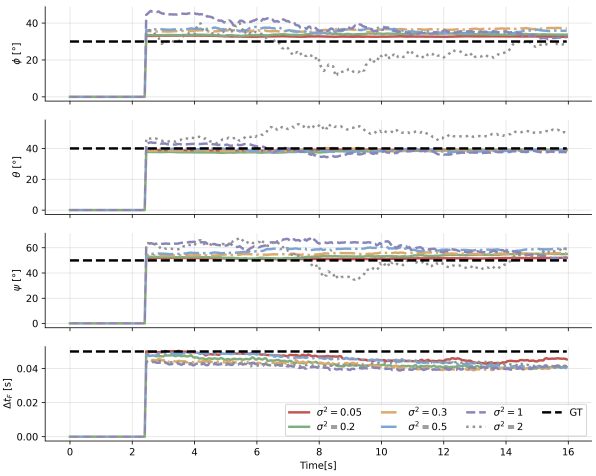


Fig. 3. The simulation result of the calibration method.

TABLE I

RMSEs WITH DIFFERENT OPTICAL FLOW SENSOR NOISE σ

σ^2[m/s]	RMSE.trans.[m]	RMSE.rot.[°]
0.05	0.111	1.84
0.2	0.096	2.64
0.3	0.117	2.23
0.5	0.158	3.80
1	0.184	2.90
2	0.270	2.91
S-MSCKF	0.259	3.88

The RMSEs with different σ^2 are listed in Table I. The errors become larger with the increasing σ^2. When σ^2 reaches $2.0m/s$, the RMSE is larger than that of the S-MSCKF method which means that the optical flow sensor is no longer useful in improving the estimation accuracy of VIO. Fortunately, in real-world flight, the optical flow sensors in most cases don't have such large noises. For example, in our experiments, when the flight speed is $1.0m/s$, the σ^2 of the optical flow sensor is approximately $0.1m/s$.

B. Simulation of Different Optical Flow Sensor Frequencies

The output frequency of the optical flow sensor f_{of} is another crucial factor affecting the accuracy of the proposed method. This simulation aims to evaluate the performance of the proposed method with different f_{of}. We use four challenging datasets from the EuRoC which are V1_03, MH_02, MH_03, and MH_04 to do the test as they include scenarios with fast maneuvers, dark scenes, or motion blur, which poses significant challenges for feature extraction and tracking tasks. Fig. 4 shows the results of the estimation error of the proposed method with different datasets and different f_{of}. It should be noted that 0Hz represents the standard S-MSCKF method, which does not fuse any measurements from the optical flow sensor. It can be observed that with the optical flow measurements, the proposed method outperforms the original S-MSCKF on all four datasets. Especially, the estimation accuracy improves with the increase of f_{of}. We can also find that when f_{of} reaches 100Hz, the estimation accuracy has no significant improvement.

C. Validation of Observability Constraint

This simulation experiment is designed to validate the method of solving inconsistency described in II-A. We set the noise level $\sigma^2 = 0.1$ and the frequency to $f_{of} = 40$Hz as with the real-world experiments and compare the performance of the S-MSCKF, the proposed method with and without the observability constraint (17) on 11 datasets. The results are shown in Table III.

TABLE II

RMSE[M] OF ATE IN EUROC DATASET FOR DIFFERENT ALGORITHMS.

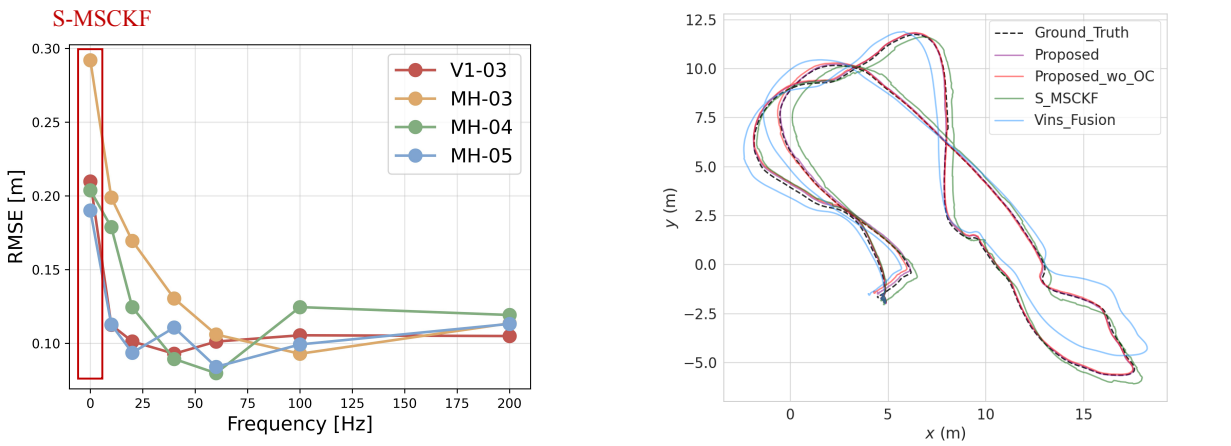
Sequence	S-MSCKF	Vins_Fusion	Proposed.wo.OC	Proposed
V1_01	0.08	0.10	0.06	**0.05**
V1_02	0.12	0.10	0.09	**0.09**
V1_03	0.21	0.11	0.10	**0.08**
V2_01	0.10	0.12	0.06	**0.04**
V2_02	0.15	0.10	0.10	**0.07**
V2_03	1.11	0.27	0.11	**0.10**
MH_01	0.11	0.24	0.08	**0.08**
MH_02	0.18	0.18	0.13	**0.08**
MH_03	0.29	0.23	0.09	**0.10**
MH_04	0.20	0.39	0.12	**0.10**
MH_05	0.19	0.19	0.16	**0.13**

We can see from Table III that the standard optical flow update using the observation matrix (11) leads to larger RMSEs compared to the ones using the observability constraints (17), which demonstrates that the method we employed can effectively address the issue of excessive reliance on the z-axis rotation.

IV. EXPERIMENT SETUP AND RESULT

A. Experiment Setup

As the datasets do not have real optical flow sensor measurements, we can only generate the sensor data based on the ground truth data, which is less convincing. To further test



(a) The RMSEs with different f_{of}.

(b) An example of the estimated trajectories using different methods.

Fig. 4. The RMSEs with different optical flow sensor frequency f_{of}.

TABLE III

RMSEs [m] of ATE in EuRoC dataset by different algorithms

Method	Proposed.wo.OC	Proposed
V1_01	0.06	**0.05**
V1_02	0.09	**0.09**
V1_03	0.10	**0.08**
V2_01	0.06	**0.04**
V2_02	0.10	**0.07**
V2_03	0.11	**0.10**
MH_01	0.08	**0.08**
MH_02	0.13	**0.08**
MH_03	0.09	**0.10**
MH_04	0.12	**0.10**
MH_05	0.16	**0.13**

the feasibility of the proposed method in the real world, we design a quadrotor as our experiment flying platform (Fig. 5). The flying platform is a self-made $420g$ quadrotor equipped with a Cool Pi board as the upboard computer, which has an 8-core CPU with a clock speed of 2.4 Ghz. The quadrotor has an Intel RealSense depth camera D435i running onboard with a frequency of 15Hz and the resolution is 640×480. An optical flow module UP-T1-100-PLUS is mounted at the bottom of the quadrotor running at 50Hz.

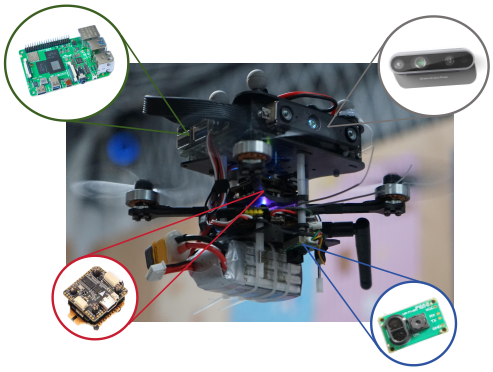
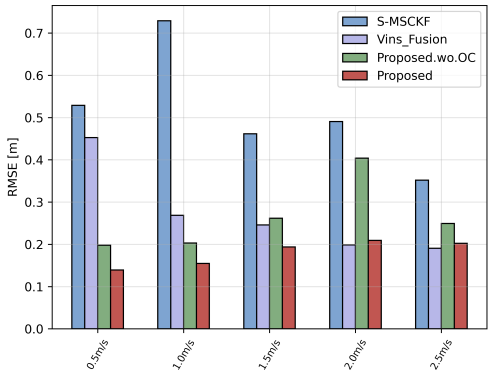


Fig. 5. The self-made flying platform used in the experiment which has a Cool Pi upboard computer, a stereo camera and an optical flow sensor onboard.

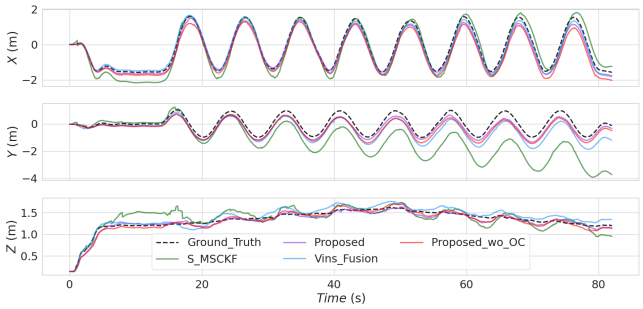
The optical flow sensors can directly estimate the velocity $v_{x,m}$ and $v_{y,m}$, as well as the height $h_{z,m}$ from the ground using a laser sensor. $v_{z,m}$ is obtained by interpolating the height $h_{z,m}$. Due to the inherent high-frequency vibration and the maneuvers of the quadrotor, the onboard optical flow sensor suffers from the inevitable noise. We employ a complementary filter to fuse the IMU measurements and the optical flow velocity to smooth the velocity measurements [27]. The Optitrack motion capture system is utilized to provide the ground truth measurements of the quadrotor's positions and attitudes, the cover size of which is $4.6m \times 2.6m \times 2.5m$.

B. Experiment Results

Due to the limited size of the flight area and the Optitrack system's coverage, we design 5 spiral trajectories with different flying speeds ranging from $0.5m/s$ to $2.5m/s$ for the quadrotor maneuvers as shown in Fig. 1. To test the performance of the proposed method in low-texture environments, during the flight, the stereo camera keeps facing toward a white wall with few texture features, which poses challenges for estimators while the optical flow sensor faces a solid color floor as shown in Fig. 1. After collecting all the flying data, we run the S-MSCKF, Vins-Fusion and the proposed method with/without observability constraints separately to show their performance.



(a) The RMSEs of the four methods under different speeds.



(b) An example of the estimated trajectory of different methods at $1m/s$.

Fig. 6. The experiment result of the estimation with different estimation methods.

TABLE IV

CPU Load of Different Methods

	S-MSCKF	Vins_Fusion	Proposed
CPU Load [%]	50%-70%	110%-150%	50%-70%

Fig. 6 shows the results of the experiment, where the upper subgraph is the statistical result of different methods at different flying speeds and the lower subgraph is the estimated trajectories by different methods together with the ground truth trajectories. In Fig. 6 (a), not surprisingly, when the flying speed is less than $1.5m/s$ where the optical flow sensor has less measurement noise σ^2, the proposed method outperforms others including the optimization-based method, Vins-fusion. However, when the quadrotor's speed exceeds $1.5m/s$, the Vins-fusion's performance surpasses our method with the expense of a heavier CPU load as shown in Table IV. However, it should be noted that the proposed method still outperforms the standard S-MSCKF significantly even at

the speed of $2.5m/s$, which indicates the advantages of the proposed method especially in high-speed flight with very limited onboard computing resources.

V. CONCLUSIONS

In this work, we introduce an efficient and robust VIO system that fuses optical flow sensor, IMU, and stereo camera measurements in a tight-coupled estimator. We first add the optical flow measurements to the S-MSCKF framework and provide an observation constraint to improve the estimation accuracy based on the observability analysis. Then a calibration method is proposed to estimate the relative rotation relationship between the optical flow sensor and the VIO frame. Furthermore, we extend an adaptive weighting algorithm to the proposed method, which can effectively adapt the inaccurate information from the tracking failures. Both simulation and real-world experiments are conducted to verify the feasibility of the proposed method. For future work, we are interested in extending this work to the localization of the quadrotor in rather more aggressive flights.

REFERENCES

[1] W. Lee, P. Geneva, Y. Yang, and G. Huang, "Tightly-coupled GNSS-aided Visual-Inertial Localization," in *2022 International Conference on Robotics and Automation (ICRA)*, (Philadelphia, PA, USA), pp. 9484–9491, IEEE, May 2022.

[2] K. McGuire, C. De Wagter, K. Tuyls, H. Kappen, and G. C. de Croon, "Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment," *Science Robotics*, vol. 4, no. 35, p. eaaw9710, 2019.

[3] W. Kwon, J. H. Park, M. Lee, J. Her, S.-H. Kim, and J.-W. Seo, "Robust autonomous navigation of unmanned aerial vehicles (uavs) for warehouses' inventory application," *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 243–249, 2019.

[4] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE international conference on robotics and automation*, pp. 2520–2525, IEEE, 2011.

[5] M. Hamer and R. D'Andrea, "Self-calibrating ultra-wideband network supporting multi-robot localization," *Ieee Access*, vol. 6, pp. 22292–22304, 2018.

[6] S. Weiss, D. Scaramuzza, and R. Siegwart, "Monocular-slam–based navigation for autonomous micro helicopters in gps-denied environments," *Journal of Field Robotics*, vol. 28, no. 6, pp. 854–874, 2011.

[7] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, 02 2014.

[8] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[9] T. Qin, J. Pan, S. Cao, and S. Shen, "A general optimization-based framework for local odometry estimation with multiple sensors," *arXiv preprint arXiv:1901.03638*, 2019.

[10] L. v. Stumberg and D. Cremers, "Dm-vio: Delayed marginalization visual-inertial odometry," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1408–1415, 2022.

[11] A. I. Mourikis and S. I. Roumeliotis, "A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, (Rome, Italy), pp. 3565–3572, IEEE, Apr. 2007.

[12] K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, and V. Kumar, "Robust Stereo Visual Inertial Odometry for Fast Autonomous Flight," Jan. 2018.

[13] S. Bahnam, S. Pfeiffer, and G. Croon, "Stereo visual inertial odometry for robots with limited computational resources*," pp. 9154–9159, 09 2021.

[14] G. Huang, "Visual-inertial navigation: A concise review," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 9572–9582, 2019.

[15] G. C. De Croon, J. J. Dupeyroux, C. De Wagter, A. Chatterjee, D. A. Olejnik, and F. Ruffier, "Accommodating unobservability to control flight attitude with optic flow," *Nature*, vol. 610, no. 7932, pp. 485–490, 2022.

[16] K. Scheper and G. Croon, "Evolution of robust high speed optical-flow-based landing for autonomous mavs," *Robotics and Autonomous Systems*, vol. 124, 11 2019.

[17] H. W. Ho, G. C. de Croon, E. Van Kampen, Q. Chu, and M. Mulder, "Adaptive gain control strategy for constant optical flow divergence landing," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 508–516, 2018.

[18] J. Liang, Y.-L. Qiao, T. Guan, and D. Manocha, "Of-vo: Efficient navigation among pedestrians using commodity sensors," *IEEE Robotics and Automation Letters*, vol. 6, pp. 6148–6155, 10 2021.

[19] X. Li, X. Qing, Y. Tang, C. Hu, J. Niu, and C. Xu, "Unmanned aerial vehicle position estimation augmentation using optical flow sensor," *IEEE Sensors Journal*, 2023.

[20] M. Li, "Visual-inertial odometry on resource-constrained systems," *Dissertations Theses - Gradworks*, 2014.

[21] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Consistency Analysis and Improvement of Vision-aided Inertial Navigation," *IEEE Transactions on Robotics*, vol. 30, pp. 158–176, Feb. 2014.

[22] Z. Song, X. Zhang, T. Li, S. Zhang, Y. Wang, and J. Yuan, "IR-VIO: Illumination-Robust Visual-Inertial Odometry Based on Adaptive Weighting Algorithm With Two-Layer Confidence Maximization," *IEEE/ASME Transactions on Mechatronics*, vol. 28, pp. 1920–1929, Aug. 2023.

[23] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 451–462, 2009.

[24] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[25] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2018.

[26] S. Agarwal, K. Mierle, and T. C. S. Team, "Ceres Solver," 10 2023.

[27] R. Mahony, T. Hamel, and J.-M. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1203–1218, 2008.