A One-Step Visual-Inertial Ego-Motion Estimation using Photometric Feedback

Shixin Tan, Shangkun Zhong, and Pakpong Chirarattananon, Member, IEEE/ASME

Abstract—This article presents a robust brightness gradient-based estimation strategy for small aerial robots. The proposed nonlinear observer is capable of estimating the flight altitude and ego-motion by the fusion of monocular vision and inertial measurement unit feedback. The novelty primarily lies in the implementation of the gradient-based featureless approach and the direct use of photometric feedback in the state and output vectors. Under the single-plane assumption, the proposed framework permits the entire estimation process to be accomplished efficiently in a single iterative step without the need for feature detection and tracking or pre-computation of optic flow as commonly seen in conventional methods. The nonlinear and featureless implementation reduces the computational demand, enlarges the region of attraction, and markedly improves the robustness of the ego-motion estimation against scenes with scarce features when compared to Kalman-based estimators and feature-based methods. We conducted extensive flight experiments with different flying patterns and textures to evaluate the performance of the proposed observer. The results reveal that the root-mean-square error in the altitude estimates is approximately 10% for flights at \( \approx 40 \text{–} 130 \text{ cm} \) above the ground, comparable to feature-based estimators. Nevertheless, the devised observer does better than feature-based methods when deployed on low-textured scenes or with low-resolution blurry images.

Index Terms—optic-flow, direct gradient-based methods, inertial measurement unit, nonlinear observer

I. INTRODUCTION

Micro Aerial Vehicles (MAVs) have gained tremendous popularity in recent decades. The advancements are driven by numerous civilian and defense applications, including espionage, transportation, inspection, and agriculture. With the progress in miniaturization and increasing integration of robots in everyday life, MAVs are anticipated to operate in urban and confined environments [1]. To this end, technical challenges related to perception and navigation must be addressed to allow these small flying machines to safely negotiate unstructured and GPS-denied environments.

Unlike terrestrial platforms that prevalently rely on multiple sensors such as LiDAR and cameras to compute RGB-depth images [2], small aerial robots are severely limited in payload and power budget. Among MAVs, an integration of an onboard camera and inertial sensor or Visual-Inertial Systems (VINS) have emerged as a common framework for providing state estimation and localization [3]–[7]. The fusion of readings from an inertial measurement unit (IMU) with visual feedback allows retrieval of the scale factor, leading to accurate pose estimates through iterative refinements. Widely used strategies involve feature detection, tracking, map reconstruction.

Despite the capability of existing map-based VINS to provide rich and accurate information, there exist several shortcomings. The robustness of feature-based methods critically depends on the ability to track features, landmarks, and retain key frames over a prolonged period. In addition to the computational resource required [8], the feature detection and tracking process could be adversely affected by poor image quality, motion blur, and low-textured scenes [9]. Besides, the refinement over a large number of poses and landmarks is demanding. This renders the solution unsuitable for smaller MAVs with severely restricted payload and power [10]–[12].

An alternative lightweight vision-based strategy suitable for ego-motion estimation for small drones is reactive navigation [13]. These algorithms directly leverage apparent motion or optic flow from most recent images to infer ego-motion when other devices are absent. When incorporated with another sensor modality, such as a time-of-flight camera [14], ultrasonic sensor, [15] or IMU [16]–[18], true flight velocity and distance can be estimated.

A. Related works and contributions

This paper addresses the optic flow-based ego-motion estimation problem for small flying robots. The omission of mapping of reactive approaches not only reduces the computational complexity, but also improves the robustness since extended tracking of landmarks is no longer required. Most existing reactive ego-motion estimation algorithms operate in multiple stages. The first step is concerned with the feature identification [19], [20], followed by the computation of optic flow via feature tracking with the Lucas-Kanade algorithm [21]. The optic flow is then fused with IMU measurements through either an optimization-based [17], [22] or Extended Kalman Filter-based [16] algorithm to produce ego-motion estimates. Among these, the findings from [23], [24] suggest that the feature detection is the most costly step, even when the reconstruction of map points are taken into consideration.

To improve the robustness and efficiency of the ego-motion estimation, an inspiration can be drawn from the direct or featureless methods [25], [26] used by map-based VINS. In such cases, the motion is inferred from the dense photometric feedback, bypassing the feature identification and tracking. The image intensity, even from areas where gradients are
small, is directly used in the optimization or filtering step [27]. Since the entire image is exploited, it has been shown to outperform feature-based methods in terms of robustness to motion blur or low-textured images when implemented with map-based localization [28]. Nevertheless, the direct methods suffer from the elevated computational cost from the generation of dense map when applied in the context of localization. This is alleviated by uses of patches of photometric feedback. The patch-based or semi-direct approaches offer a balance between robustness and complexity by making use of small image patches around identified features [4].

To date, uses of the direct approach in lightweight estimators for MAVs are still limited. In [29], the authors devised a featureless method for controlling mobile robots based on time-to-contact. For aerial vehicles, the featureless approach was proposed to replace the optic flow measurements from the feature tracking [30]. By taking image gradients, motion information is extracted. When combined with readings from an IMU through an EKF, estimates of flight altitude were obtained. In these examples, the altitude is recovered in multiple steps as the flow divergence (ratio of velocities in x-y-z directions to the distance) are computed first, followed by the altitude estimation.

This work takes a consolidated approach to tackle the ego-motion estimation based on the featureless method. Unlike previous direct estimation methods, where filters or observers require the knowledge of pre-computed optic flow [17], [22] or flow divergence (ratio of flight velocity to the distance) [30] for each iteration, the entire ego-motion estimation for each iteration herein is accomplished in a single step. To achieve this, photometric feedback from an entire image is part of the state vector and directly used as measurements of the proposed estimator. The evolution of the state, output, and their predictions are tightly coupled with the ego-motion and corresponding image gradients. With IMU readings, the scheme efficiently estimates the inverse altitude, flow divergence, and the plane’s normal in a single step assuming there exists only a single plane under the camera’s view. The single plane assumption is prevalent among optic flow-based ego-motion estimation algorithms due to the exclusion of mapping [17], [18], [22], [29], [30]. Nevertheless, the proposed framework does not impose a restriction on the camera’s motion or the plane’s inclination as found in [17], [18], [30].

In the proposed one-step featureless approach, photometric feedback is modeled and included in the state and output. This results in a large output vector whose length is equal to the number of pixels observed. The size of the output renders an EKF-based estimator impractical as the computation of the Kalman gain involves an inverse operation on a matrix of which the dimension is dictated by the length of the output vector. This necessitates the development of a nonlinear observer (NLO) that produces the estimates of flight altitude, flow divergence, and plane’s normal. With a convergence proof, the NLO has certain advantages over EKF-based algorithms. It provides a quantifiable convergence rate (but lacking the estimates of the covariance) [22]. Without linearization, a nonlinear approach possesses a large region of attraction and therefore is less sensitive to disturbances or poor initial conditions [17], [22]. Overall, the use of dense photometric feedback provides improved robustness over feature-based methods and the one-step implementation, which cannot be achieved with indirect methods, radically streamlines the estimation. It is conceivable that the efficiency and robustness of the proposed ego-motion estimator potentially renders it a suitable estimator for a lightweight reactive navigation strategy to be employed by MAVs with acute computational constraints such as small or insect-sized flying robots [10]–[12].

This paper is structured as follows. Section II provides descriptions of the camera-centric odometry, optic flow, and formulation of the equations of motion. Section III presents the NLO and its convergence proof based on the state dynamics from Section II in the discrete-time domain. The proposed NLO is validated through a series of flight experiments in section IV, accompanied by the analysis of the results and the comparison to an EKF-based estimator, a feature-based approach, and a map-based state-of-the-art method. Lastly, in section V, a conclusion and future directions are discussed.

II. EGO-MOTION, OPTIC FLOW, AND DYNAMICS

A. Problem formulation

Consider a scenario where a moving camera observes points on a plane as depicted in Fig. 1. The ego-motion problem considered here begins with the formulation that relates the distance between the camera and the plane, image intensity values, and optic flow. In this work, the distance, relative orientation between the camera and the plane, and flow divergence [24] are regarded as unknowns to be estimated. Photometric feedback or pixel irradiance from entire images is directly treated as measurements. In addition, the proposed method requires the knowledge of the camera-centric angular velocity and linear acceleration which can be provided by an IMU (through an accelerometer and a gyroscope). Without loss of generality, the camera’s frame and IMU’s frame are assumed to coincide. The framework can be applied in the context of reactive navigation of mobile or aerial robots.

B. Continuous homography constraint

Fig. 1 shows a camera moving with respect to a stationary plane with the camera frame and inertia frame. The camera
frame is defined such that the Z axis aligns with the optical axis. The camera-centric linear and angular velocities with respect to the inertial frame are \( \mathbf{v} = [v_x, v_y, v_z]^T \) and \( \omega = [\omega_x, \omega_y, \omega_z]^T \). The apparent velocity of a point \( \mathbf{P} = [X, Y, Z]^T \) on the plane due to the camera motion is \( \mathbf{P} = [\omega]_x \mathbf{P} + \mathbf{v} \), where \([\omega]_x\) is the skew-symmetric matrix associated with \( \omega \) [31]. We define \( \mathbf{n} = [n_x, n_y, n_z]^T \) as a camera-centric unit vector normal to the plane. The orthogonal distance \( d \) between the camera and the plane satisfies \( \mathbf{n}^T \mathbf{P} / d = 1 \). Therefore,

\[
\mathbf{P} = [\omega]_x \mathbf{P} + \mathbf{v} / d \mathbf{n}^T \mathbf{P}.
\]  

(1)

Camera images provide the projection of \( \mathbf{P} \), or \( \mathbf{p} = [x, y, f]^T \), where \( f \) is the focal length with an unknown scaling parameter \( \lambda \) [31], such that \( \lambda \mathbf{p} = \mathbf{P} \). The velocity of \( \mathbf{p} \) on the image plane satisfies \( \dot{\mathbf{p}} = \lambda \dot{\mathbf{P}} + \dot{\lambda} \mathbf{P} \). With equation (1), this yields

\[
\dot{\mathbf{p}} = (\dot{\mathbf{P}} - \lambda \dot{\mathbf{p}}) / \lambda = \left( [\omega]_x + \frac{\mathbf{v}}{d} \mathbf{n}^T \right) \mathbf{p} - \lambda \dot{\mathbf{p}} / \lambda.
\]  

(2)

The quantity \( \dot{\mathbf{p}} = [\dot{p}_x, \dot{p}_y, 0]^T \), where \( \dot{p}_x \) and \( \dot{p}_y \) are the optic flow of the point \( \mathbf{p} \) [31], represents the apparent velocity on the image plane. To eliminate unknown inverse depth \( \lambda \), the equation above is pre-multiplied with the skew-symmetric matrix \([\omega]_x\) and re-written as:

\[
[\omega]_x (\dot{\mathbf{p}} - [\omega]_x \mathbf{p}) = [\omega]_x \frac{\mathbf{v}}{d} \mathbf{n}^T \mathbf{p}.
\]  

(3)

The resultant equation, also known as the continuous homography constraint [22], [31], is suitable for the estimation of \( d \), \( \mathbf{v} / d \), and \( \mathbf{n} \) via the one-step direct method below.

### C. Optic flow and direct method

Suppose a camera provides images with \( N \) pixels in the form of pixel irradiance, such that \( I_{xy} \) represents irradiance of the pixel at location \((x, y)\) at time \( t \). The brightness constancy constraint assumes \( I_{xy} \) does not change significantly during a short period of time [25] or:

\[
\frac{\partial I_{xy}}{\partial x} \dot{p}_x + \frac{\partial I_{xy}}{\partial y} \dot{p}_y + \frac{\partial I_{xy}}{\partial t} = 0,
\]  

(4)

in which \( \partial I_{xy}/\partial x \) and \( \partial I_{xy}/\partial y \) are the spatial irradiance gradients, and \( \partial I_{xy}/\partial t \) is the rate of the pixel irradiance. Here, \( \dot{p}_x \) and \( \dot{p}_y \) are optic flow as present by \( \dot{\mathbf{p}} \) in equation (2). We define a \( 3 \times 1 \) image gradient vector \( \nabla \mathbf{I} = [\partial I_{xy}/\partial x \ \partial I_{xy}/\partial y \ \partial I_{xy}/\partial t]^T \) and a basis vector \( \mathbf{e}_3 = [0 \ 0 \ 1]^T \), equation (4) can be re-written as:

\[
\frac{\partial I_{xy}}{\partial t} \mathbf{f} = -\dot{p}_x \frac{\partial I_{xy}}{\partial x} \mathbf{f} - \dot{p}_y \frac{\partial I_{xy}}{\partial y} \mathbf{f} = -e_3^T \nabla \mathbf{I} \times [\mathbf{p}]_x \dot{\mathbf{p}}.
\]  

(5)

Premultiplying equation (3) by \( e_3^T \nabla \mathbf{I} \times \) and combining it with equation (5) to eliminate the optic flow yields

\[
\frac{\partial I_{xy}}{\partial t} \mathbf{f} = -e_3^T \nabla \mathbf{I} \times [\mathbf{p}]_x \frac{\mathbf{v}}{d} (\mathbf{n}^T \mathbf{p} + [\omega]_x \mathbf{p}).
\]  

This provides a direct relation between the camera motion \((\mathbf{n}^T, \omega)\) and the image gradients \((\partial I_{xy}/\partial t, \nabla \mathbf{I})\). Consequently, no feature detection nor tracking of feature points is required for distance estimation. This markedly reduces the computational demand and enhances the robustness to low-textured scenes and blurred images.

### D. Equations of motion

To iteratively estimate \( d \) and other quantities of interest from equation (6), we define the flow divergence vector \( \vartheta = \mathbf{v} / d \) and the inverse distance \( \alpha = d^{-1} \). Instead of directly dealing with \( d \), the inverse distance parameterization is similar to the treatment in [4], [22] and known to produce better results. The camera provides measurements of the pixel irradiance as an \( N \times 1 \) vector obtained by stacking \( I_{xy} \)’s together:

\[
\mathbf{I} = \{I_{xy}\} \in \mathbb{R}^N.
\]

The state and output vectors are defined as

\[
\mathbf{X} = \begin{bmatrix} \alpha & \mathbf{n}^T & \mathbf{I}^T \end{bmatrix}^T
\]  

(7)

\[
\mathbf{Y} = \mathbf{C} \mathbf{X} = \mathbf{I},
\]  

(8)

where \( \mathbf{C} = [0_{N \times 7} \ 1_{N \times N}] \) (1 denotes an identity matrix). The time evolution of \( \mathbf{X} \) is described by the followings:

\[
\dot{\alpha} = \Phi(n) \alpha,
\]  

(9)

\[
\dot{\mathbf{n}} = \Phi(n) \mathbf{n},
\]  

(10)

where we have used the fact that \( \dot{\alpha} = \Phi(n) \alpha \) when \( \mathbf{n} \) is a camera-centric linear acceleration. Evaluation of \( \Phi(n) \) requires the specific acceleration from the IMU and the knowledge of the gravity vector in the camera frame (but not magnetometer readings for the camera-centric approach). This is achieved with the sensor fusion present in standard IMUs or flight controllers (in which IMU biases are also taken care of). The use of a unit vector \( \mathbf{n} \) to represent the orientation of the plane avoids issues related to the parameterization of a rotation.

Lastly, the dynamics of \( \mathbf{I} \) is computed from equation (6) as

\[
\frac{\partial I_{xy}}{\partial t} = - \left( 1 / f \right) \mathbf{e}_3^T \nabla \mathbf{I} \times [\mathbf{p}]_x (\partial \mathbf{n} \mathbf{p} + [\omega]_x \mathbf{p}).
\]  

(12)

Together, equations (9)-(12) capture the dynamics of \( \mathbf{X} \) as

\[
\dot{\mathbf{X}} = \mathcal{F}(\mathbf{X}, \omega, \alpha).
\]  

(13)

In this work, both \( \omega \) and \( \alpha \) are provided by an IMU and presumed known.

### III. NONLINEAR OBSERVER

#### A. Formulation of the NLO

In this section, we establish a discrete-time nonlinear observer for estimating the state vector \( \mathbf{X} \) from IMU measurements and image luminosity \( \mathbf{Y} \). To begin, let \( T \) denote a sample time. Following notations are employed:

- A subscript \((\cdot)_k\) indicates a quantity at the \( k \)th time step.
- A variable \( \bar{z}_{k|j} \), for example, describes the estimate of \( z \) at the \( k \)th time step given observations up to and including the \( j \)th time step. \( \bar{z}_k \) is a shorthand form of \( \bar{z}_{k|k} \).
- Given an unknown quantity \( z_k \) and its estimate \( \bar{z}_k \), the estimation error \( \hat{z}_k \) is defined as \( \hat{z}_k = z_k - \bar{z}_k \).
- For a vector \( \mathbf{z}_i \in \mathbb{R}^{3 \times 1} \) for \( i = 1, 2, \ldots, N \), an operator \( \{\mathbf{z}_i\}_N \) represents a horizontal stacking operation of \( z \) such that \( \{\mathbf{z}\}_N = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N\} \in \mathbb{R}^{3 \times N} \).

In the discrete-time domain, the dynamics described by equation (13) is approximated using the forward finite difference as the transition model

\[
\mathbf{X}_{k+1} = \mathbf{X}_k + \mathcal{F}(\mathbf{X}_k, \omega_k, \alpha_k) T.
\]  

(14)
Fig. 2. A block diagram describing the input-output stability of the NLO with $z^{-1}$ denoting the unit delay. Each equivalent subsystem $H_i(\cdot)$ is $\mathcal{L}$ stable, implying the convergence of the estimation errors.

The nonlinear observer serves as the observation model to iteratively predicts and updates the $\hat{X}_k$ according to

$$
\hat{X}_{k+1} = \hat{X}_k + F(\hat{X}_k, \omega_k, a_k)T
$$

$$
\hat{X}_{k+1} = \hat{X}_{k+1} + G_k(Y_{k+1} - C \hat{X}_{k+1}), \quad (15)
$$

where $G_k$ is

$$
G_k = \begin{bmatrix}
-2\Delta_n^T \alpha f a_k^- \{ p^T \hat{n}_k[p] \times [\nabla I_k]_x e_3 \}^N \\
-2\Delta_n^T \alpha f \{ p^T \hat{n}_k[p] \times [\nabla I_k]_x e_3 \}^N \\
-2\Delta_n^T \alpha f \{ p^T \hat{n}_k[p] \times [\nabla I_k]_x e_3 \}^N \\
1_{N \times N}
\end{bmatrix}, \quad (16)
$$

when $\Delta_i$’s and $\Gamma_i$’s are symmetric positive definite matrices that satisfy the conditions: (i) $\Delta_i \to \infty$; and (ii) $\Delta_i \Gamma_i T \ll 1$. Readers are referred to the Supplemental Materials for the origin of equation (16).

**B. Convergence of the NLO**

Stability of the NLO is shown in multiple steps. The proof shows that the estimate of each element in $X_k$ sequentially asymptotically converges in the absence of process and measurement noises. With disturbances, the estimation errors remain bounded. For simplicity, process noise and the error induced by the measurement noise in the term $\nabla I_k$ are neglected. Otherwise, they can be treated in a similar manner to the measurement noise. With these considerations, the convergences of $\hat{I}_k, \hat{n}_k, \hat{\theta}_k,$ and $\hat{\alpha}_k$ are shown in the respective order as schematically depicted in Fig. 2. That is, it can be shown that there exist equivalent subsystems $H(\cdot)$’s that are $\mathcal{L}$ stable. As a result, the entire system is asymptotically stable. We introduce two following Lemmas to assist with the convergence proof.

**Lemma 1.** A discrete-time system of $Z_k$ with an input $U_k$ and a sample time $T$:

$$
Z_{k+1} = Z_k - 2\Lambda T A_k^T A_k Z_k + D_k T + B_k U_k, \quad (17)
$$

where $A_k^T$ is full row rank, $D_k$ is uniformly bounded, and $\Lambda, \Gamma$ are constant symmetric positive definite matrices, has a globally asymptotically stable equilibrium point at $Z = 0$ when $\Lambda \to \infty$ and $\Delta T \ll 1$. In the presence of a bounded input $U_k$, if $B_k$ is finite-gain $\mathcal{L}_\rho$ stable for $p \in [1, \infty]$, then $Z_k$ is also finite-gain $\mathcal{L}_\rho$ stable.

*Proof.** See the Supplemental Materials.

**Lemma 2.** A discrete-time system

$$
Z_{k+1} = Z_k - A_k Z_{k-1} + D_k T + B_k U_k, \quad (18)
$$

in which $0 < B_k < \frac{2}{T}$, and $D_k$ is uniformly bounded as $|D_k| < D_+$, is Lyapunov stable at the equilibrium point $Z = 0$. Moreover, the system becomes asymptotically stable as $T \to 0$. In the presence of a bounded input $U_k$, if $B_k$ is finite-gain $\mathcal{L}_\rho$ stable for $p \in [1, \infty]$, then $Z_k$ is also finite-gain $\mathcal{L}_\rho$ stable.

*Proof.** See the Supplemental Materials.

1) **Convergence of $\hat{I}_k$:** Suppose $\delta I_k$ represents the measurement (image) noise such that $Y_k = I_k + \delta I_k$, it can be deduced from equations (15) and (16) that

$$
\hat{I}_{k+1} = \hat{I}_{k+1} + (Y_{k+1} - \hat{I}_{k+1}) = I_{k+1} + \delta I_{k+1}. \quad (19)
$$

In other words, $\hat{I}_k = \delta I_k$ or the estimate of $I$ immediately converges to the true value in the absence of the measurement noise regardless of the prediction $\hat{I}_{k+1}$. Correspondingly, $H_I$ in Fig. 2 is $1_{N \times N}$.

2) **Convergence of $\hat{n}_k$:** The dynamics of $\hat{n}_{k+1}$ through the update and correction steps from equations (11)-(16) are

$$
\hat{n}_{k+1} = \hat{n}_k + [\hat{n}_k]_x \omega_k T \quad \hat{n}_{k+1} = \hat{n}_{k+1} + \hat{\theta}_k T [\nabla I_k]_x e_3, \quad (20)
$$

where

$$
\hat{\theta}_k_T = \left[\begin{array}{c}
\hat{\theta}_k T [\nabla I_k]_x e_3
\end{array}\right]. \quad (21)
$$

The term $Y_{k+1} - \hat{I}_{k+1}$ in equation (20) can be written using the form provided by equation (12) and the fact that $e_3 T [\nabla I_k]_x \hat{\theta}_k T p = (p^T [\nabla I_k]_x e_3)^T e_3$, as a result,

$$
Y_{k+1} - \hat{I}_{k+1} = (I_{k+1} - I_k) - (\hat{I}_{k+1} - \hat{I}_k) + \Delta I_k
$$

$$
\quad = -T \frac{\partial^2 I_k}{\partial (x, y, f)^T} [\nabla I_k]_x e_3 T \hat{n}_k
$$

$$
\quad + T \frac{\partial^2 I_k}{\partial (x, y, f)^T} [\nabla I_k]_x e_3 T \hat{n}_k + \Delta I_k, \quad (22)
$$

where $\Delta I_k = \hat{I}_{k+1} - \hat{I}_k$. To deal with the unknown $\hat{\theta}_k$, we define a scalar quantity $\Psi_k(x, y)$ for each individual pixel at $p = [x, y, f]^T$ as $\Psi_k(x, y) = \hat{\theta}_k_T [\nabla I_k]_x e_3 / \hat{\theta}_k_T [\nabla I_k]_x e_3$. Subsequently,

$$
Y_{k+1} - \hat{I}_{k+1} = -T \frac{\partial^2 I_k}{\partial (x, y, f)^T} \left[\begin{array}{c}
\hat{\theta}_k_T [\nabla I_k]_x e_3 / \hat{\theta}_k_T [\nabla I_k]_x e_3
\end{array}\right] n_k + \Delta I_k, \quad (23)
$$

where $\nabla I_k$ is the product of $\Psi_k(x, y)$ for all $N$ pixels (the nominal value of $\Psi_k$ when $\hat{\theta}_k \to 0$ is unity). With some manipulation, substitution of equation (23) into (20) produces

$$
\hat{n}_{k+1} = \hat{n}_k + [\hat{n}_k]_x \omega_k T - 2\Delta_n \Gamma_n f A_{n,k} \Delta I_k
$$

$$
\quad + 2\Delta_n \Gamma_n T A_{n,k} A_{n,k}^T \left[\begin{array}{c}
\hat{\theta}_k T \left(\frac{\hat{\theta}_k T [\nabla I_k]_x e_3}{\hat{\theta}_k T [\nabla I_k]_x e_3} \right) n_k - \hat{n}_k
\end{array}\right]. \quad (24)
$$

Let $\eta_k$ denote $\hat{\eta}_k T A_{n,k} A_{n,k}^T \left(\frac{\hat{\theta}_k T [\nabla I_k]_x e_3}{\hat{\theta}_k T [\nabla I_k]_x e_3} \right) n_k + D_{n,k} \Delta I_k$.

$$
\eta_{k+1} = \eta_k - 2\Delta_n \Gamma_n T A_{n,k} A_{n,k}^T \eta_k + D_{n,k} \Delta I_k. \quad (25)
$$
where
\[
D_{n,k} = \Pi \Psi_k [n_k]_X \omega_k - [\hat{n}_k]_X \omega_k,
\]
\[
B_{n,k} = 2 \Lambda_n \Gamma_n f A_{n,k}.
\] (26)

According to Lemma 1, the asymptotic stability of \( \eta_k \) is attained in the absence of \( \Delta I_k \) with suitable gains (\( \Lambda_n \to \infty \) and \( \Lambda_n \Gamma_n T \ll 1 \)) by treating \( D_{n,k} \) as bounded disturbances. The condition requires \( A_{n,k} \) to be full row rank. This places some limitations on the camera movement (\( \hat{\vartheta}_k \)) and image gradients (\( \partial I_{xy}/\partial x, \partial I_{xy}/\partial y \neq 0 \)) as suggested by equation (21). The condition on camera motion is easily achieved on a flying robot in practice. Furthermore, this rank constraint is relaxed when the persistence of excitation is taken into account (see the Supplemental Materials and [32]). In addition, with the measurement noise \( \Delta I_k \), the system remains stable as long as \( B_{n,k} \) or \( \Lambda_n \Gamma_n A_{n,k} \) remains bounded. The \( \mathcal{L} \) gain of \( B_{n,k} \) influences the stability properties of the equivalent subsystem \( H_n \) in Fig. 2 as outlined in the Supplemental Materials. The stability of \( \eta_k \) warrants the convergence of \( \hat{n}_k \) up to the scale factor \( \Pi \Psi_k \). Then, \( \hat{n}_k \) is obtained through normalization after an update. In practice, it is adequate to perform the normalization every \( \sim 100 \) steps.

3) Convergence of \( \hat{\vartheta}_k \): The stability of \( \hat{\vartheta}_k \) is achieved through a similar framework to that of \( \hat{n}_k \) without the need for normalization. From equations (10) and (16), the estimation error evolves according to
\[
\hat{\vartheta}_{k+1} = \hat{\vartheta}_k + 2 \Lambda_\vartheta \Gamma_\vartheta f A_{\vartheta,k} \left( Y_{k+1} - \bar{I}_{k+1|k} \right) + D_{\vartheta,k} T,
\] (27)
where
\[
D_{\vartheta,k} = a_k \vartheta_k + \hat{\vartheta}_k^T n_k \vartheta_k + [\vartheta_k]_X \omega_k - a_k \hat{\vartheta}_k \hat{n}_k + \hat{\vartheta}_k - [\vartheta_k]_X \omega_k, \quad \text{and}
\]
\[
\hat{A}_{\vartheta,k} = \left\{ p_T^T \hat{n}_k [p]_X [\nabla I_{xy}] T e_3 \right\}_N.
\] (28)
The term \( Y_{k+1} - \bar{I}_{k+1|k} \) in equation (27) can be treated in a similar manner to equation (22), but with the fact that \( e_T^T [\nabla I]_X [p]_X \vartheta = (p_T^T [p]_X [\nabla I]_X e_3) T \vartheta \).

\[
Y_{k+1} - \bar{I}_{k+1|k} = -\frac{T}{f} \left\{ p_T^T n_k [p]_X [\nabla I_{xy}] T e_3 \right\}_N \vartheta_k + T \left\{ p_T^T \hat{n}_k [p]_X [\nabla I_{xy}] T e_3 \right\}_N \hat{\vartheta}_k + \Delta I_k
\]
\[
= -\frac{T}{f} \hat{A}_{\vartheta,k} \vartheta_k - T \hat{A}_{n,k} \hat{n}_k + \Delta I_k,
\] (29)
Combining equations (27) and (29) yields
\[
\hat{\vartheta}_{k+1} = \hat{\vartheta}_k - 2 \Lambda_\vartheta \Gamma_\vartheta T \hat{A}_{\vartheta,k} \hat{A}_{\vartheta,k}^T \hat{\vartheta}_k + D_{\vartheta,k} T
\]
\[
+ B_{\vartheta,k} \left[ \hat{n}_k^T \Delta I_k \right]^T,
\] (30)
where
\[
B_{\vartheta,k} = 2 \Lambda_\vartheta \Gamma_\vartheta \left[ -T \hat{A}_{\vartheta,k} A_{n,k}^T f \hat{A}_{\vartheta,k} \right].
\] (31)
Applying Lemma 1, the estimation error \( \hat{\vartheta}_k \) is reduced in the absence of disturbance (\( \Delta I_k, \hat{n}_k \to 0 \)) as long as \( A_{\vartheta,k} \) is full rank and the gain conditions are met. That is the convergence of \( \hat{\vartheta}_k \) is obtained under the similar conditions to that of \( \hat{n}_k \), but by treating both \( \Delta I_k \) and \( \hat{n}_k \) as inputs of the equivalent subsystem \( H_\vartheta \) as shown in Fig. 2 and the Supplemental Materials.

4) Convergence of \( \alpha_k \): Again, we start with the incremental update of the estimation error \( \alpha_k - \bar{\alpha}_k \) derived from equations (9), (15), and (29):
\[
\alpha_{k+1} = \alpha_k - 2 \Lambda_\alpha \Gamma_\alpha T a_{k-1}^T \hat{A}_{\alpha,k} \hat{A}_{\alpha,k}^T \bar{\alpha}_k + D_{\alpha,k} T
\]
\[
- 2 \Lambda_\alpha \Gamma_\alpha a_{k-1}^T \hat{A}_{\alpha,k} \left( T A_{n,k}^T \hat{n}_k - f \Delta I_k \right)
\] (32)
where
\[
D_{\alpha,k} = \hat{\vartheta}_k^T n_k \alpha_k - \hat{\vartheta}_k^T \hat{n}_k \alpha_k.
\] (33)
Exploiting equation (10), the term \( \hat{\vartheta}_k \) in equation (32) can be re-written:
\[
\hat{\vartheta}_k = (\hat{\vartheta}_k - \hat{\vartheta}_{k-1}) + (\hat{\vartheta}_{k-1} - \hat{\vartheta}_{k-2}) + \cdots + (\hat{\vartheta}_1 - \hat{\vartheta}_0)
\]
\[
= a_{k-1} \alpha_{k-1} + \hat{\vartheta}_{k-1} T + \hat{\vartheta}_{k-1} \hat{n}_k T
\]
\[
+ \left( \hat{\vartheta}_1^T \hat{n}_k - \left[ \omega_\vartheta \right]_X \right) \hat{\vartheta}_k T + \hat{\vartheta}_{k-1}
\] (34)
Substituting this to equation (32) yields
\[
\alpha_{k+1} = \alpha_k - 2 \Lambda_\alpha \Gamma_\alpha T a_{k-1}^T \hat{A}_{\alpha,k} \hat{A}_{\alpha,k}^T \bar{\alpha}_k - a_{k-1} \bar{\alpha}_{k-1} + D_{\alpha,k} T
\]
\[
+ B_{\alpha,k} \left[ \n_k^T \hat{\vartheta}_k^T \Delta I_k \right]^T,
\] (35)
where
\[
B_{\alpha,k} = -2 \Lambda_\alpha \Gamma_\alpha a_{k-1}^T \hat{A}_{\alpha,k}.
\] (36)
The stability of \( \bar{\alpha}_k \) is then attained according to Lemma 2 as long as \( 2 \Lambda_\alpha \Gamma_\alpha T^2 a_{k-1}^T \hat{A}_{\alpha,k} \hat{A}_{\alpha,k} a_{k-1} \in (0, \frac{1}{2}) \), and \( D_{\alpha,k} \) is bounded. The first condition requires the gains \( \Lambda_\alpha \Gamma_\alpha \) to be sufficiently small, \( A_{\alpha,k} \) to be full rank as in section III-B3, and the acceleration to be non-zero. The restriction on the motion is similar and related to the rank condition of \( A_{n,k} \) considered earlier. The boundedness of \( D_{\alpha,k} \) is readily achieved in physically feasible flights. Furthermore, as \( T \to 0 \), the system is asymptotically stable, or the system remains stable in the presence of non-zero \( \hat{n}_k, \hat{\vartheta}_k \), and \( \Delta I_k \) when they are regarded as bounded inputs to the subsystem \( H_\alpha \) as long as \( \Lambda_\alpha \Gamma_\alpha \) is finite.

In summary, the stability of the proposed observer is achieved for each quantity in the state vector in the order shown in Fig. 2. If present, estimation errors accumulate over each step but the output remains bounded. Under suitable gains, the system is globally asymptotically stable as \( T \to 0 \). This renders the NLO more robust against disturbances and initial errors compared to EKF-based strategies that suffer from inaccuracies introduced by the linearization and restricted region of attraction.
described by equations (7)-(13). Compared to the proposed
luminosity measurements according to the nonlinear dynamics
were uniformly downsampled to directly compared. In the implementation,
implementation allows results from different estimators to be
implemented in Python and applied to the collected images
contains 120 seconds of mid-air measurements suitable for
flight patterns, textures, plane inclinations, etc. Each flight
out to demonstrate the estimation performance for various
used for feedback. All data are time aligned.

IV. EXPERIMENTAL VERIFICATION

In this section, we perform indoor flight experiments to validate the proposed nonlinear observer and compare the performance to that of an EKF and a state-of-the-art map-based approach. In-depth investigations on different flight patterns, textures, computational requirements, and the impact of the planar scene, are provided.

A. Experiment setup and implementation

For flight experiments, we employed a single-board com-
puter (Raspberry Pi 3 Model B) and the Pi Camera v2 with
the field of view of $24.4^\circ \times 19.0^\circ$ to record $640 \times 480$-px images at 90 Hz. An IMU (MPU9250) was used to provide the angular velocity ($\omega$) and gravity integrated acceleration ($a$) at the rate of 100 Hz, synchronized with the images using Python scripts. After the camera calibration, the camera-IMU unit was attached to an AscTec Hummingbird robot (Ascending Technologies) with the downlooking camera as shown in Fig. IV-A(a). The experiments were carried out in a $3.0 \times 3.0 \times 2.5$ m arena equipped with six motion capture cameras (OptiTrack Prime 13w) for tracking the position and orientation of the quadrotor for the ground-truth measurements of $\alpha$, $\theta$, and $n$ and robot’s position control as the estimation results were not used for feedback. All data are time aligned.

To validate the proposed estimator, experiments were carried
out to demonstrate the estimation performance for various
flight patterns, textures, plane inclinations, etc. Each flight
contains 120 seconds of mid-air measurements suitable for
the estimator. After the experiments, the proposed NLO was
implemented in Python and applied to the collected images
and IMU measurements for estimation of the state vector
defined in equation (7) on the Raspberry Pi. The offline
implementation allows results from different estimators to be
directly compared. In the implementation, $640 \times 480$-px images
were uniformly downsampled to $160 \times 120$-px images after
the application of $5 \times 5$ averaging kernels. After that the spatial
gradients $\nabla f$ in equation (12) were then obtained by applying
an image convolution with normalized $3 \times 3$ Sobel kernels.

To provide benchmark comparisons, an EKF was employed
to provide estimations of the state vector from the image
luminosity measurements according to the nonlinear dynamics
described by equations (7)-(13). Compared to the proposed
NLO, EKF has two major drawbacks: (i) as a result of the
linearization, the stability is no longer guaranteed and
the convergence critically depends on the initial conditions;
(ii) the evaluation of Kalman gain involves an inversion of
an $N \times N$ matrix. This is impractically expensive even for
downsampling images, rendering the method unsuitable for
real-time operation, particular on platforms with limited power.
In contrast, it can be seen that the complexity of the proposed
method is dominated by simple matrix multiplications related
to matrices of size $7 \times N$ and $N \times 1$ only.

B. Flight Experiments

1) Flights on horizontal plane: The first set of flight
experiments were performed on a horizontal ground. We
executed three flying patterns over four types of textures. Three
flights were carried out for each combination of texture and
flight pattern, producing 36 flights in total. Three hovering
flights were performed at 0.4 m, 0.8 m, 1.2 m altitude for
each texture, whereas for the other two flight patterns, all three
flights for each texture are nominally identical.

As illustrated in Fig. 5, four textures chosen include
checkerboard (CHB), ramp (RMP), sinusoid (SIN) and arti-
ficial leaves (LEAF). The first three textures were selected
to emphasize the difference between feature-based and direct
methods. The checkerboard, prevalent in computer vision
owing to pronounced edges and corners for easy detection,
features $6 \times 6$-cm tiles. The ramp and sinusoidal textures
used are grayscale patterns with the spatial intensity varying
according to 2D sinusoidal and ramp functions. The absence
of apparent corners and edges from these textures is intended
for evaluating one key benefit of the proposed gradient-
based approach when compared to feature-based methods.
The periods of the first three textures are 12 cm. For the
leaf pattern, it mimics a more realistic scene to evaluate the
performance of the NLO with less structured patterns.

Three flight patterns investigated here are hovering, vertical
trajectory, and 3D circular trajectory. During hovering flights,
the robot is anticipated to exhibit minimal movement, rep-
resenting the scenario where the proposed estimator is near
the unobservable condition. This is to demonstrate that slight
oscillations and vibrations provide sufficient excitation for the
estimator. For the vertical trajectory, the robot was commanded
to follow a sinusoidal trajectory at the altitude between 0.45-
0.95 m at 0.2 Hz. The altitude variation verifies the observer’s
ability to deal with changes in altitude and flow divergence in
the absence of horizontal motion. For the 3D flight pattern,
the robot followed a cyclic circular trajectory with the radius
of 0.65 m at 0.2 Hz, spanning the approximate volume of
$0.5 \times 0.5 \times 0.4$ m. The pattern, similar to the trajectory used
in [22], verifies the effectiveness of the proposed method in
practical situations in which the robot traverses in 3D space.

2) Flights on inclined plane: Since the proposed NLO
is capable of estimating relative plane orientation ($n$), six
additional flights were performed on a $10^\circ$ incline affixed
with the CHB texture. The robot flew horizontally covering
the distance of 1.5m at 0.8 m altitude in a cyclic manner at
0.2 Hz to assess the quality of the distance and other estimates
when the robot traverses over a non-horizontal surface.
Fig. 4. Example estimation results corresponding to three flight patterns over SIN texture. Ground truth data are provided by the motion capture system. Both estimates from the NLO and EKF are plotted. The flight altitudes are in the range of 0.45–0.95 m, depending on the flight pattern. The darker region in each subplot indicates a 30-second period after the estimator initializes. The estimates are assumed to have converged thereafter. (Top) The plot shows the distance estimates and the ground truth. (Middle) The flow divergence in three directions. (Bottom) The estimate of the plane’s normal with respect to the camera frame. A closed-up plot detailing the last five seconds of the vertical flight can be found in the Supplemental Materials.
noise covariances. In contrast, the NLO displays robustness to estimate the distance despite our best effort to adjust the noise covariances. In 13 out of the 36 flights, the EKF estimates diverge, failing to converge. For the EKF, only 23 out of 36 datasets result in converged estimates that are eligible for the calculation of the RMS errors. The numbers of datasets with converged estimates for each combination of flight patterns and textures are labeled in brackets on top of each bars in the last row of the plots.

3) Flights over an occluded ground: The derivation of the NLO relies on the assumption of a planar scene. To assess the robustness and the performance of the NLO when such condition is violated, three additional flights were carried out over the CHB texture, partially occluded by a 25 × 28 × 30-cm cardboard box. The robot flew vertically in a cyclic manner at the altitude 0.45 to 0.95 m at the frequency of 0.2 Hz.

C. Estimation Results

Fig. 4 shows representative results from three flight patterns over the SIN texture above a horizontal ground. The estimated state vectors are compared to ground truth measurements provided by the motion capture system to calculate the root mean square (RMS) errors. It can be seen that the NLO consistently produces better results than the EKF. For the distance estimates, the EKF displays larger steady-state errors and slower convergence speed. For the flow divergence, the EKF shows significant fluctuation while the NLO yields noticeably smoother results. For the normal vector, the steady-state errors from the EKF are also visibly larger. Among different flight patterns, hovering flights estimation yields a much slower convergence rate for the distance estimation.

1) Estimation results from flights over horizontal ground: In 13 out of the 36 flights, the EKF estimates diverge, failing to estimate the distance despite our best effort to adjust the noise covariances. In contrast, the NLO displays robustness with no divergence. The estimation errors are given in Fig. 5. The root-mean-square (RMS) errors are calculated from 90-s periods, 30 s after taking off to permit the estimates to converge. For the EKF, the RMS errors are computed from 23 flights with converged estimates, while all 36 flights are taken into consideration for the NLO. Overall, the RMS distance errors from the EKF for three flight patterns are 16.7, 11.7, and 15.3 cm, while for the NLO, the errors are 12.1, 10.2 and 10.7 cm. All data combined, the RMS error from the EKF is 28% larger than that of from the NLO.

For all four textures, Fig. 5 reveals that the NLO provides reliable estimates of the distance and flow divergence. Altogether, the RMS error of the estimated distance is approximately 10% at the 0.7 m average flight altitude and the RMS error of the flow divergence is 0.16 s⁻¹. The performance is comparable to that of the state-of-the-art feature-based method which yields the RMS distance error of 9.2 cm for a flight at 1 m altitude and the mean speed error of 0.1 ms⁻¹ [22].

Among three flight patterns, the distance estimates from the NLO corresponding to vertical and circular flights show approximately 17% lower RMS errors than those from hovering flights. The observed trend is reversed for the estimates of the flow divergence, whereas the estimation errors of the normal vector from three flight patterns are nearly indistinguishable. This is likely because the quality of the distance...
estimates benefits from motion or excitation. In contrast, the flow divergence is tightly and directly coupled with the measurements. Therefore, the estimates of the flow divergence are detrimentally affected by motion-induced disturbances.

2) Estimation results from flights over an incline: The results obtained from six flights above a 10° incline are shown in Fig. 6. During the horizontal translation, the camera perceived an alteration in the altitude, resembling the change induced by the vertical motion above a horizontal plane. For this reason, we benchmark the horizontal flight results against the outcomes belonging to vertical flights with the CHB from Section IV-B1. The major difference in two cases is highlighted in Fig. 6(a), which illustrates the RMS angle between the camera’s optical axis and the true camera-centric plane’s normal (11.4° versus 3.4°). The estimation results shown in Fig. 6(b)-(d) reveals that the errors from both scenarios are highly comparable. In case of the inclined surface, the NLO demonstrate no significant change in the error of the plane normal estimates, whereas the EKF evidently suffered with the RMS error of nearly 20°. The results suggest that the NLO is more robust than the EKF when there exists an appreciable deviation between the camera axis and the plane’s normal.

3) Estimation results from flights with occlusion: Fig. 7 shows the results when the one plane assumption is violated in comparison with the benchmark experiments. The benchmark results were taken from the previous experimental set. Fig. 7(a) shows the altitude estimates from the NLO and EKF, with respect to the groundtruth. It can be seen that that the estimates always underpredict the altitude. This is likely because the observers detect the combined distance to the plane and the occluding object. Fig. 7(b) highlights the average unoccluded area in terms of the percentage of the camera’s field of view during three flights, demonstrating that > 25% of the view was obstructed by a box (refer to the Supplemental Materials for the method used for computing the segregation). The results in Fig. 7(c)-(e) reveal a noticeable degradation in the performance due to the provided reason. However, the proposed scheme was still able to provide converged estimates, demonstrating the robustness of the nonlinear approach.

D. Convergence rates among different flight patterns

According to section III-B, it has been concluded that the observability of the system lies in the non-zero values of $\theta$ and the acceleration $a$. In hovering flights, however, vibrations and fluctuations in velocity and acceleration turn out to provide sufficient excitation for the estimation. Among three flying patterns, both $a$ and $\theta$ associated with hovering flights are still smaller in magnitude than in circular and vertical flights. As a result, the distance converges at a slower rate as predicted by the persistency of excitation condition given in the Supplemental Materials.

E. Robustness to textures and image features

Unlike feature-based approaches that require prominent corners and edges, the NLO leverages the motion and non-zero image gradients, or $\nabla I$ (equation (5)), for the estimator to be observable. The NLO is anticipated to perform better with flights over RMP and SIN textures than over CHB texture owing to the absence of spatial gradients between edges in the CHB pattern. This is consistent with the results in Fig. 5, where the trend can be observed for all flight patterns.

The use of image gradients offers robustness when benchmarked against feature-based or semi-direct methods to scenes with scarce features. In evidence, we applied two popular feature detectors: FAST tracker [20] and Shi-Tomasi Corner Detector [19], on the obtained flight images. Both algorithms
the estimation errors is less pronounced as illustrated in Fig. 7 to the need for a nonlinear observer as a Kalman-based estimator 4-6, the average computing times per frame are 69 ms and the Processing times were recorded. As presented in Fig. 8(a), for data collected from nine circular flights from Section IV-B1, were implemented on a Raspberry Pi 3 Model B to process computational power is vital. Here, the time cost is inspected to assess the efficiency. Both Python scripts for NLO and EKF putational power is vital. Here, the time cost is inspected to inspect the efficiency. Both Python scripts for NLO and EKF were implemented on a Raspberry Pi 3 Model B to process data collected from nine circular flights from Section IV-B1. Processing times were recorded. As presented in Fig. 8(a), for the 160 × 120-px images used to produce the results in Fig. 4-6, the average computing times per frame are 69 ms and 6390 ms for the NLO and EKF. The substantial reduction in the cost of the proposed NLO with respect to the EKF attests the need for a nonlinear observer as a Kalman-based estimator becomes inefficient when entire images are used.

Furthermore, to understand the trade-off between computational demand and accuracy, we repeated the comparison using the same original images that were uniformly downsampled to 80 × 60 px, instead of to 160 × 120 px, after applying 7 × 7 averaging kernels. As shown in Fig. 8(a), with the 4-fold decrease in image pixels, the time costs for both NLO and EKF reduce by a factor of 3 to 4. Nevertheless, the impact on the estimation errors is less pronounced as illustrated in Fig. 8(b)-(d). The decrease in image resolution magnifies the RMS errors of the distance and flow divergence by ≈20% and ≈35% and the influence on the normal vector is almost negligible. Depending on platform requirements, the outcomes suggest that low resolution images may provide sufficiently accurate estimates while demanding considerably less resources.

G. Comparison to State-of-the-Art

To compare the performance of the proposed NLO against a state-of-the-art approach, we selected the flight data with the CJB pattern and a published dataset from [34] (with the top flight speed of 5 m/s). Both sets of flight data were pipelined to the NLO and the map-based VINS—VINS-mono [6]. Due to the nature of map-based VINS, the estimate of the flight altitude or distance to a flat terrain is not directly available for comparison. Instead, we resort to using the estimates of robo-centric flight velocity. For the NLO, flight velocity is a product of the flow divergence $\nabla$ and the distance $d$.

As presented in Fig.9, for flight data with the checkerboard pattern, the RMS in velocity errors from the NLO and VINS-mono are highly comparable: 0.07 m/s and 0.06 m/s. For the published datasets, the flight speed reaches up to 5 m/s, rendering errors of the estimated velocity to be visibly larger. The RMS errors from the NLO and VINS-mono remain similar at 0.18 m/s and 0.15 m/s. In addition, for the CHB pattern, the average computational times per frame are 3.7 ms and 19.3 ms for the NLO and VINS-mono (both implemented on a laptop with Intel i5-4440 processor). For the dataset from [34], the average computing times are 6.7 ms and 31.3 ms. It can be seen that the omission of mapping and the use of the planar scene assumption renders the NLO is highly efficient and suitable for platforms with limited computational power.

V. CONCLUSION AND FUTURE WORK

In this article, we have proposed a robust one-step ego-motion estimation strategy that integrate monocular vision and IMU measurements for feedback. When applied to aerial robots, the framework is capable of estimating the flight altitude and velocity for reactive navigation. The novelty lies in the use of image intensities as part of the state and output vectors. This eliminates the feature detection and tracking process, allowing the entire estimation to be achieved in a single step. As verified by several flight experiments, the featureless approach is robust to low-textured scenes and the proposed nonlinear observer offers superior efficiency when compared to an EKF-based implementation.
Similar to existing ego-motion estimation strategies for reactive navigation, the proposed framework is still limited for real-world deployment owing to the single-plane assumption. Possible extensions of this work include the consideration of multiple planes in the camera view. This would allow a flying robot to negotiate a corridor while maintaining safe distances to the ground and surrounding walls at the same time.

REFERENCES

Pakpong Chirarattananon (S’12-M’15) received the B.A. degree in Natural Sciences from the University of Cambridge, U.K., in 2009, and the Ph.D. degree in Engineering Sciences from Harvard University, Cambridge, MA, USA, in 2014.

He is currently an Associate Professor with the Department of Biomedical Engineering, City University of Hong Kong, Hong Kong, China. His research interests include bio-inspired robots, micro air vehicles, and the applications of control and dynamics in robotic systems.
The condition stated by equation (40) is equivalent to the existence of state models in [36], as the gain $A_k$ is, the system is finite-gain stable for each $p \in [1, \infty]$. The condition stated by equation (41) is globally asymptotically stable as $t \to \infty$. Moreover, in the limit $T \to 0$, the system described by equation (41) is globally asymptotically stable as $L_{k+1} - L_k < \lambda_E \|Z_k\|^2 + \frac{5}{3} D_+ \|Z_k\| T + D_+^2 T^2$

$$L_{k+1} - L_k < \lambda_E \|Z_k\|^2 + \frac{5}{3} D_+ \|Z_k\| T + D_+^2 T^2$$

$$< \lambda_E (\|Z_k\| + \frac{5}{6 \lambda_E} D_+ T)^2$$

$$+ \frac{1}{36 \lambda_E} (36 \lambda_E - 25) D_+^2 T^2.$$
B. Conventional Feature Detectors

Fig. 10 shows three different images taken during a flight over a SIN chosen to illustrate the shortcomings of the conventional feature detectors. The combination of poor image resolution, motion blur, and the lack of sharp edges make these images devoid of evident corners. In the majority of cases, both FAST [20] and Shi-Tomasi [19] detectors classify only point adjacent to the edges as features with highest qualities as demonstrated in Fig. 10(a). It can be seen that these detected features are unsuitable (not functional) for tracking purposes as they will likely be out of frame in subsequent images due to the motion. In addition, they do not correspond to actual features, rendering them susceptible to poor tracking performance when used with the Lucas-Kanade algorithm [33]. Fig. 10(b) shows a rare (< 5%) example in cases which FAST identifies four or more functional features. Similarly, Fig. 10(c) provides another example in which Shi-Tomasi detector returns four functional points located at some distance from edges. This is only achieved in less than 20% of images with SIN or RMP textures. These examples highlight possible advantages of the featureless methods. While the outcomes shown here do not necessarily reflect real-world scenarios, where there are likely more prominent features visible in images, these example images are reasonable representatives of images from low-cost cameras or images degraded by motion blur.

C. Closed-up Estimation Results

In order to clearly differentiate the estimation performance of the NLO and EKF in Fig. 4, Fig. 11 provides closed-up plots of the flight data in Fig. 4, focusing on the last 30 seconds. It can be seen that the distance estimation from the NLO is superior to that of EKF. Other plots suggest that that the NLO produces smoother estimates of the flow divergence with less chattering compared to the EKF.

![Figure 10](image-url)  
Fig. 10. Result of Shi-tomasi corner detector and FAST corner detector. (a) a representative case when both detectors find no functional feature away from the image edges. (b) another case that FAST corner detector detects feasible features however for Shi-tomasi corner detector, most of the features are on the edge of the image. (c) shows another case when FAST method only captures two features but Shi-tomasi corner detector works out fine. Each figure has a resolution of 160 × 120.
Fig. 11. Closed up plots of the estimation results (vertical flight) from Fig. 4.
D. Calculation of Occluded Region

To compute the percentage of an occluded area in the image, we post-processed the images by the application of the Hough Transform to the entire images. This results in both the boundaries of the boxes and the checkerboard pattern. To segment the box boundaries, a thresholding process was applied to only retain the shape of the box by considering the color of nearby pixels. Only the edges belonging to the edges of the brown cardboard box were kept. Thereafter, is straightforward to calculate the percentage of the occluded region in the image. Example results from each step are shown in Fig. 12.

E. Derivation of the observer gain $G_k$

Herein, we provide some explanations on the form of $G_k$ to provide readers some intuition on its origin. The outline here applies to the convergence of other estimates follow the same strategy in the gain design.

We focus on the gain used for updating $\tilde{n}_k$, the third row of $G_k$ or $G_{k,3}$. To begin, observe that the time evolution of the photometric values from equation (12) is

$$I_{k+1} - I_k = -(1/f) \{ (e_3^T [\nabla I_k] \times [p] \times \hat{\theta}_k n_k^T p)^T \}_{N}.$$  

Recall that the operator $\{z\}_{N}$ stands for a horizontal stacking operation of $z_i \in \mathbb{R}^3$ such that $\{z\}_{N} = [z_1, z_2, \ldots, z_N] \in \mathbb{R}^{3 \times N}$. It can be seen that $I_{k+1} - I_k$ is directly related to $n_k$. To make use of this, the portion $\{ (e_3^T [\nabla I_k] \times [p] \times \hat{\theta}_k n_k^T p)^T \}_{N}$ is re-arranged to $\{ p \hat{\theta}_k^T [p] \times [\nabla I_k] \times e_3 \}_{N} n_k$. Hence, the innovation term $\left( Y_{k+1} - C \tilde{X}_{k+1|k} \right)$ is approximately proportional to

$$\left( Y_{k+1} - C \tilde{X}_{k+1|k} \right) \approx \{ p \hat{\theta}_k^T [p] \times [\nabla I_k] \times e_3 \}_{N} n_k.$$

The intention is to ultimately obtain $\tilde{n}_{k+1} \propto -\tilde{n}_{k+1}$. This is approximately achieved by designing the third row of the observer’s gain $G_{k,3}$ to be $\approx - \{ p \hat{\theta}_k^T [p] \times [\nabla I_k] \times e_3 \}_{N}$, such that the condition

$$G_{k,3} \left( Y_{k+1} - C \tilde{X}_{k+1|k} \right) \approx - \{ p \hat{\theta}_k^T [p] \times [\nabla I_k] \times e_3 \}_{N} n_k$$

or $G_{k,3} \left( Y_{k+1} - C \tilde{X}_{k+1|k} \right) \propto -n_k$ is approximately met as $\{ p \hat{\theta}_k^T [p] \times [\nabla I_k] \times e_3 \}_{N} \{ p \hat{\theta}_k^T [p] \times [\nabla I_k] \times e_3 \}_{N}^T$ is positive definite. In the actual proof and implementation, there are extra terms emerging from the image noise and the fact that only the estimate of $\hat{\theta}_k$ is available. However, they can be taken care of as detailed in the proof and the input-output stability is achieved.