An Efficient Iterated EKF-based Direct Visual-Inertial Odometry for MAVs Using a Single Plane Primitive

Shangkun Zhong and Pakpong Chirarattananon

***Abstract*—This letter proposes an efficient visual-inertial estimator for aerial robots. The main contribution lies in the direct use of intensity measurements of the latest image frames and key frame via both continuous and regular homographic relations under an assumption of a single planar scene. The filter-based method provides comprehensive estimates of position and attitude with notable efficiency and robustness. The flight evaluation indicates that the incorporation of keyframes significantly minimizes the estimation drift while retaining accurate estimates of flight velocity. Compared to state-of-the-art methods, the proposed work provides rivalling performance at substantially lower computational cost by taking the advantage of the assumed single planar structure.**

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

I. Introduction

AUTONOMOUS flight of Micro Aerial Vehicles (MAVs) demands high-bandwidth and accurate motion estimates. Visual-Inertial Systems (VINS) and Simultaneous Localization and Mapping (SLAM) frameworks for recovering a camera pose and 3D structures have gathered immense attention from the community owing to their scalability, accuracy and cost efficiency. The formulation of VINS and SLAM is generally based on either a batch optimization [1]–[4] or an Extended Kalman Filter (EKF) [5]–[7]. While optimization-based approaches often outperform filter-based methods in terms of accuracy [8], systems employing EKF frameworks [5]–[7] possess superior efficiency. This is because for EKF-based methods, the estimates of past camera poses are continuously marginalized based on the information propagated over time and only the latest state remains updated. In contrast, optimization-based methods consider the batch-optimization problem across a temporal and spatial window of the camera poses. Albeit the incorporation of keyframes, the high dimension of estimated variables leads to costly computation. This impedes real-time applications of these systems with small robots with limited processing power.

The VINS and SLAM can be alternatively categorized according to the image processing procedure as *direct* (or semi-direct) [9]–[11] and *feature-based* [1], [2], [4] methods. The *direct* operation on pixel illuminations to estimate the camera pose and geometries [10], [11] in the popular VINS or SLAM has achieved better efficiency than *feature-based* methods [1], [2], [4]. Photometric considerations render the sophisticated feature descriptor [2], [4] and the feature tracking process unnecessary, significantly saving the computational demand. Besides, *direct* methods offer superior robustness to environments with scarce salient features thanks to the added capability to process edgelets [12], [13].

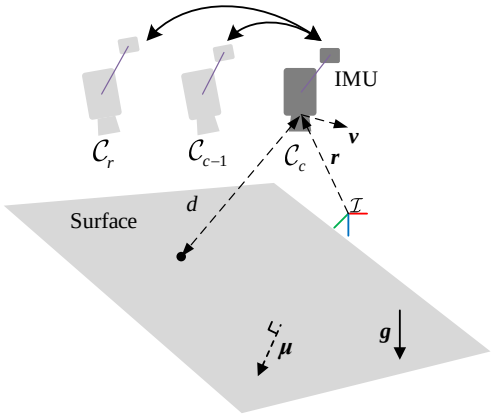


Fig. 1. Diagram of an IMU-camera rig movement. The moving IMU-camera setup observes a single non-horizontal plane with a unit normal $\boldsymbol{\mu}$ and orthogonal distance to the camera d. The current camera frame $\mathcal{C}_c$ has a linear velocity $\boldsymbol{v}$, angular rate $\boldsymbol{\omega}$, and a translation $\boldsymbol{r}$ with respect to the inertial frame $\mathcal{I}$. The motion of the current camera $\mathcal{C}_c$ and the plane primitive is recovered based on photometric comparison between images from the current frame, a previous frame $\mathcal{C}_{c-1}$, and a distant keyframe $\mathcal{C}_r$.

To markedly lower the complexity and bring down the computational cost, several works considered a down-looking camera operating in structured environments [14]–[18]. Therein, the authors assumed a monocular camera overlooking planar scenes for egomotion estimation via optic flow. Grabe *et al.* proposed a nonlinear observer to estimate the robot's altitude and velocity based on the plane's normal given by a separate IMU attitude estimator [15]. Hua *et al.* assumed the planar target to be horizontal to eliminate the dependence on an external attitude estimator [16]. We previously relaxed the assumption on the plane's orientation by decoupling the formulations of the gravity direction and the plane's normal [14]. Furthermore, using the direct EKF-based method, instead of the feature-based implementations as found in [15], [16], the estimator in [14] displays exceptional efficiency and robustness. Nevertheless, the simultaneous estimation of the 6D

Manuscript received: August 17, 2020; Revised November 6, 2020; Accepted December, 7, 2020. This paper was recommended for publication by Editor Eric Marchand upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the Research Grants Council of the Hong Kong Special Administrative Region of China (grant number CityU-11215117).

The authors are with the Department of Biomedical Engineering, City University of Hong Kong, Hong Kong SAR, China (emails: shanzhong4-c@my.cityu.edu.hk, pakpong.c@cityu.edu.hk).

camera pose (position and attitude) and plane primitive has not been achieved by these proposals [14]–[16].

This letter presents an efficient direct Visual-Inertial Odometry (VIO) for MAVs overlooking a planar target. Instead of only estimating the velocity, altitude, and gravity direction as in [14] (or [15], [16] but without the gravity direction), the proposed VIO provides the full six-degree-of-freedom pose estimation with comparable accuracy to two state-of-the-art VINS [2], [11] with unparalleled efficiency. Moreover, unlike most feature-based VIOs, the proposed filter-based estimator performs reliably when provided with low-resolution images (72×72 px) from an onboard camera of a consumer-grade drone. This potentially increases the image sample rate, resulting in high-bandwidth estimates.

Motivated by the semi-direct filter-based VIO [11], to develop an efficient estimator, photometric feedback of entire low-resolution images is fused with IMU measurements via the iterated extended Kalman filter (IEKF) using both continuous and regular homography [19] mappings. The achieved efficiency is attributed to three key factors: (i) the choice of the Kalman-based filtering implementation as opposed to elaborate nonlinear optimization techniques [1]–[3], (ii) the direct use of dense photometric errors that eliminates the feature extraction and tracking process and associated computational complexity [9]–[11], and (iii) a restriction on the observation of a single planar scene. This condition, also present in [14]–[16], tends to be comfortably met by a flying robot with a downward-facing camera. Leveraging this, the environment is modeled as a single planar object to be inferred rather than a collection of 3D unstructured points [2], [4], [11]. The planar geometry leads to a low-dimensional state vector (18) in contrast to hundreds of variables in the point-based VIO.

To attain satisfactory accuracy and robustness, particularly when benchmarked against state-of-the-art VINS, the photometric errors between two latest images are compared based on the continuous homography constraint to refine the state estimation as first shown in [14]. This largely prompts the estimates of plane's normal and velocity to converge. The inherent estimation drifts along four unobservable states (position and heading's angle) [20] are further suppressed through the introduction of keyframes from distant past and the associated keyframe management mechanism. The regular homography constraint is applied between two distant camera views to supplement the measurement update as shown in Fig. 1. In the meantime, the use of dense photometric feedback from entire images in both steps renders the estimator highly robust against measurement noises.

Compared to our previous strategy [14], the proposed VIO gains an ability to estimate full 6-DOF pose, including the position and yaw state. This is achieved with the introduction of the regular homography model and keyframe management to the measurement model. The efficiency of the estimator is retained thanks to the direct implementation and the assumption of a single planar scene that was also used in [14].

The proposed method still possesses limitations. The superior efficiency is highly dependent on the single plane assumption and thereby the framework is inapplicable to complex scenarios, in contrast to most existing VINS. However, the proposed work potentially provides a solution for computationally restrained platforms, such as small and insect-scale flying robots [21], [22], thanks to its superior efficiency and comparable accuracy with other VINS.

The rest of this paper is structured as follows. Section II provides preliminaries on the continuous and regular homographic relations. Section III presents the VIO formulation in IEKF framework with the single plane primitive. In Section IV, extensive flight experiments were performed to validate, assess, and compare the performance of the proposed method with respect to two state-of-the-art VINS [2], [11]. Lastly, a conclusion is provided.

II. Continuous Homography and Homography Mapping

In this section, the continuous homography constraint and homography mapping are provided as background. Vectors are expressed with respect to the current (latest) camera's frame unless stated otherwise. The camera coordinate system $\mathcal{C}$ and the inertial frame $\mathcal{I}$ are introduced as seen in Fig. 1. The continuous homography associates the camera's linear velocity $\boldsymbol{v}$ and angular rate $\boldsymbol{\omega}$ with respect to a point on a stationary plane to the optical flow $\dot{\boldsymbol{p}}$

$$\dot{\boldsymbol{p}} = -(\mathbf{1} - \boldsymbol{p}\boldsymbol{e}_z^T)\tilde{\boldsymbol{H}}\boldsymbol{p}, \tag{1}$$

where $\boldsymbol{p} = [u, v, 1]^T$ is the homogeneous point projection in image plane at $z = 1$, $\boldsymbol{e}_z = [0, 0, 1]^T$, and $\mathbf{1}$ is an identity matrix. $\tilde{\boldsymbol{H}} \in \mathbb{R}^{3\times3}$ is the continuous homography matrix [19]:

$$\tilde{\boldsymbol{H}} = \boldsymbol{M}([\boldsymbol{\omega}]_\times + \frac{1}{d}\boldsymbol{v}\boldsymbol{\mu}^T)\boldsymbol{M}^{-1}, \tag{2}$$

where $[\boldsymbol{\omega}]_\times \in \mathbb{R}^{3\times3}$ represents the skew-symmetric matrix associated with $\boldsymbol{\omega}$, $\boldsymbol{\mu} \in \mathbb{S}^2$ denotes the unit normal to the plane, d is the orthogonal distance from the camera center to the plane as illustrated in Fig. 1, and $\boldsymbol{M} \in \mathbb{R}^{3\times3}$ is the intrinsic matrix of a pinhole camera.

Unlike the continuous homography, the homography mapping relates the projections of points on a plane between two images up to a scale factor γ

$$\boldsymbol{p}_{\mathcal{C}_j} = \gamma\boldsymbol{H}\boldsymbol{p}_{\mathcal{C}_i}, \tag{3}$$

with $\boldsymbol{p}_{\mathcal{C}_j}$ and $\boldsymbol{p}_{\mathcal{C}_i}$ denoting a pair of coplanar points matched on two views $\mathcal{C}_i$ and $\mathcal{C}_j$. $\boldsymbol{H}$ is the homography matrix [19] that associates the two views' relative pose with respect to a plane to the image correspondence

$$\boldsymbol{H} = \boldsymbol{M}\left(\boldsymbol{R} + \frac{1}{d}\boldsymbol{r}\boldsymbol{\mu}^T\right)\boldsymbol{M}^{-1}, \tag{4}$$

where the matrix $\boldsymbol{R} \in SO(3)$ and $\boldsymbol{r} \in \mathbb{R}^3$ represent the relative orientation and translation between the two views.

III. IEKF Estimation Framework

Here, we detail the IEKF-based formulation to estimate the camera motion, the planar landmark, and the IMU's biases. The overall pipeline is illustrated in Fig. 2. The state and covariance propagation is carried out once the IMU data arrives whereas main state update step leverages photometric

intensities from entire current and keyframe images. After the primary update, a keyframe management routine is called to determine whether to replace the current keyframe and perform a second update on the estimated position and yaw angle.

A. State Definition

The state vector is composed of the following elements:

$$\boldsymbol{x} := (\boldsymbol{r}, \boldsymbol{v}, \boldsymbol{q}, \boldsymbol{\mu}_s, \alpha, \boldsymbol{b}_a, \boldsymbol{b}_\omega)\,, \tag{5}$$

where $\boldsymbol{r} \in \mathbb{R}^3$ and $\boldsymbol{v} \in \mathbb{R}^3$ are the robocentric position and velocity of the camera expressed in the current camera frame. The quaternion $\boldsymbol{q} \in SO(3)$ from the camera frame $\mathcal{C}$ with respect to the inertial frame $\mathcal{I}$ is parameterized with the exponential maps following the definition in [11], α is the inverse orthogonal distance ($\alpha = d^{-1}$) from the camera center to the surface. The unit normal $\boldsymbol{\mu} \in \mathbb{S}^2$ of the plane can be obtained by rotating the basis vector $\boldsymbol{e}_z$ via the rotation $\boldsymbol{\mu}_s \in SO(3)$ such that $\boldsymbol{\mu} = \boldsymbol{\mu}_s(\boldsymbol{e}_z)$. This implementation resolves the singularity issue and brings about relatively simple differentials. The IMU biases $\boldsymbol{b}_i$'s are defined below. The state definition differs from that of our prior work [14] in the inclusion of the position and the relative heading's angle, and the use of the velocity instead of the ratio velocity.

B. State Prediction

The state prediction process follows closely the procedure for a standard Kalman filter as described in [14]. The propagation of the state and uncertainties are based on discretized dynamics. The dynamics of the state depends on the specific acceleration $\hat{\boldsymbol{a}}$ and angular rate $\hat{\boldsymbol{\omega}}$ of the camera frame, which can be obtained from the IMU measurements ($\boldsymbol{a}_m$, $\boldsymbol{\omega}_m$) after subtractions of biases $\boldsymbol{b}$'s and white noises $\boldsymbol{w}$'s:

$$\hat{\boldsymbol{a}} = \boldsymbol{a}_m - \boldsymbol{b}_a - \boldsymbol{w}_a, \quad \hat{\boldsymbol{\omega}} = \boldsymbol{\omega}_m - \boldsymbol{b}_\omega - \boldsymbol{w}_\omega. \tag{6}$$

Consequently, the continuous-time state dynamics ($\dot{\boldsymbol{x}}$) are:

$$\dot{\boldsymbol{r}} = -\left[\hat{\boldsymbol{\omega}}\right]_\times \boldsymbol{r} + \boldsymbol{v} + \boldsymbol{w}_r, \tag{7}$$

$$\dot{\boldsymbol{v}} = -[\hat{\boldsymbol{\omega}}]_\times \boldsymbol{v} + \hat{\boldsymbol{a}} - \boldsymbol{q}^{-1}(\boldsymbol{g}_0) + \boldsymbol{w}_v, \tag{8}$$

$$\dot{\boldsymbol{q}} = -\boldsymbol{q}(\hat{\boldsymbol{\omega}}) \tag{9}$$

$$\dot{\boldsymbol{\mu}}_s = \boldsymbol{N}(\boldsymbol{\mu}_s)^T \hat{\boldsymbol{\omega}} + \boldsymbol{w}_\mu, \tag{10}$$

$$\dot{\alpha} = \alpha^2 \boldsymbol{\mu}^T \boldsymbol{v} + w_\alpha, \tag{11}$$

$$\dot{\boldsymbol{b}}_a = \boldsymbol{w}_{b_a}, \quad \dot{\boldsymbol{b}}_\omega = \boldsymbol{w}_{b_\omega}, \tag{12}$$

where $\boldsymbol{g}_0 = -g\boldsymbol{e}_z$ is the free fall acceleration. The mapping $\boldsymbol{q}(\bullet) : \mathbb{R}^3 \to \mathbb{R}^3$ rotates a vector from the camera frame to the inertial frame. The terms $\boldsymbol{w}_i$'s are zero-mean Gaussian white noise. The operator $\boldsymbol{N}(\bullet)^T$ linearly projects a 3D vector into the 2D tangent space of a unit vector in $\mathbb{R}^2$ such that $\boldsymbol{N}(\boldsymbol{\mu}_s) = \left[\boldsymbol{\mu}_s(\boldsymbol{e}_x), \boldsymbol{\mu}_s(\boldsymbol{e}_y)\right]$, where $\boldsymbol{e}_x = \left[1,0,0\right]^T$ and $\boldsymbol{e}_y = \left[0,1,0\right]^T$ so $\boldsymbol{\mu}_s(\boldsymbol{e}_i)$'s become bases of the coordinate system [11].

In the discrete-time domain at timestamp t_{k-1}, once the IMU data is available, the a-posteriori state $\boldsymbol{x}_{k-1}^+$ and covariance $\boldsymbol{\Sigma}_{k-1}^+$ at the time instance t_{k-1} are propagated to the a-priori state $\boldsymbol{x}_k^-$ and covariance $\boldsymbol{\Sigma}_k^-$ according to the IEKF prediction routine detailed in [14].

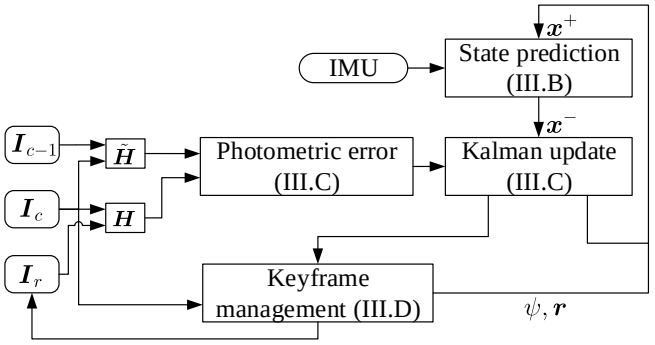


Fig. 2. A flowchart outlining the proposed IEKF Estimation method. The state prediction is executed once an IMU data frame is updated. Pixel intensities from entire images (current I_c, previous I_{c-1}, and a keyframe) are directly used for state update through continuous $\tilde{\boldsymbol{H}}$ and regular $\boldsymbol{H}$ homography constraints. After the primary Kalman update, a keyframe management system verifies whether to update the keyframe and perform a secondary update on the estimates of the camera's translation $\boldsymbol{r}$ and heading angle ψ.

C. State Update with Photometric Measurements

The proposed method is characterized by the use of image illumination from whole images for the state update instead of patches of pixels around features on keyframes [11] to improve robustness. To radically reduce the required computation, we formulate the plane primitive as the scene structure rather than 3D point landmarks [1], [2], [4], [11]. Similar to [11], the plane tracking in the primary Kalman update step eliminates the complexity in feature association projected from the plane by exploiting the single plane assumption.

To achieve accurate estimation of the position and yaw angle, during the state update, the regular homography is incorporated to the measurements in addition to the use of continuous homography model as present in our previous work [14]. The continuous homography model makes use of the instantaneous linear and angular velocities to dynamically track the state with fast convergence rate. While the model enables robust and precise tracking of the velocity and robot's inclination, four unobservable states (position and yaw angle) severely suffers from integration drifts [20]. The adoption of the regular homography mapping and keyframes essentially suppresses the integration drifts by correlating the relative rotation and distance between two distant images. After this update, a routine is brought in to maintain a pool of keyframes.

1) Continuous Homography Measurement Model: To use photometric measurements in the state update, the continuous homography equation (2) is applied to link photometric measurements from two consecutive frames, relating the observed image motion to the current camera state.

At instance t_{c-1} a point on the surface is projected through the homogeneous transformation onto the image plane (as defined with Eq. (1)) at $\boldsymbol{p}_{c-1}$. Let $\bar{\boldsymbol{p}}_{c-1} = \boldsymbol{\pi}(\boldsymbol{p}_{c-1}) \in \mathbb{R}^2$ be $\boldsymbol{p}_{c-1}$ with its last element truncated, the corresponding pixel intensity is $\boldsymbol{I}_{c-1}(\bar{\boldsymbol{p}}_{c-1})$ with $\boldsymbol{I} \in \mathbb{R}^{m \times n}$ denoting a 2D image. After one camera-based time period δT, the point displaces to a new location $\boldsymbol{p}_c$ on the current (new) image plane $\boldsymbol{I}_c$ according to the motion prescribed by the current state $\boldsymbol{x}_k$. This displacement can be characterized by the projective transformation $\tilde{\mathcal{H}}_k(\boldsymbol{p}_{c-1}|\boldsymbol{x}_k)$. Under the constant brightness

assumption, the pixel intensity of point $\boldsymbol{p}_i$ remains unchanged:

$$\boldsymbol{I}_c(\boldsymbol{\pi}(\tilde{\mathcal{H}}_k(\boldsymbol{p}_{c-1}|\boldsymbol{x}_k))) = \boldsymbol{I}_{c-1}(\bar{\boldsymbol{p}}_{c-1}), \tag{13}$$

where the mapping $\tilde{\mathcal{H}}_k(\boldsymbol{p}_{c-1}|\boldsymbol{x}_k)$ is derived from Eq. (1)

$$\tilde{\mathcal{H}}_k(\boldsymbol{p}_{c-1}|\boldsymbol{x}_k) \approx \boldsymbol{p}_{c-1} - \delta T(\mathbf{1} - \boldsymbol{p}_{c-1}\boldsymbol{e}_z^T)\tilde{\boldsymbol{H}}_k(\boldsymbol{x}_k)\boldsymbol{p}_{c-1}. \tag{14}$$

Eq. (13) allows an entire image $\boldsymbol{I}_c$ to be part of the measurement vector as explained below. The accuracy of the strategy depends on the camera's frame rate $(\delta T)^{-1}$.

2) Regular Homography Measurement Model: Unlike the continuous homography that accounts for marginal changes between two consecutive frames, the homography tranformation is applicable to images observing the same planar scene with overlaps. Here, a keyframe from the distant past $\mathcal{C}_r$ is introduced. A pair of image correspondence $\boldsymbol{p}_r$ and $\boldsymbol{p}_c$ on the keyframe $\mathcal{C}_r$ and current frame $\mathcal{C}_c$ are reflected onto the measured intensities of the keyframe $\boldsymbol{I}_r$ and $\boldsymbol{I}_c$ through the regular homography constraint. However, due to a possible illumination variation between distant images, the constant brightness assumption no longer holds. The affine intensity model consisting of two parameters κ and β is applied to link the keyframe $\boldsymbol{I}_r$ and current images $\boldsymbol{I}_c$ at time t_k as

$$\boldsymbol{I}_c(\bar{\boldsymbol{p}}_c) = \kappa \boldsymbol{I}_r\left(\boldsymbol{\pi}\left(\mathcal{H}(\boldsymbol{p}_c|\boldsymbol{x}_k)\right)\right) + \beta. \tag{15}$$

Both κ and β are to be marginalized out during the state update as carried out in [1], [11]. The mapping $\mathcal{H}(\bullet)$ depicts the connection between the projected points $\boldsymbol{p}_r$ and $\boldsymbol{p}_c$ following the definitions given by Eqs. (3)-(4)

$$\mathcal{H}(\boldsymbol{p}_c|\boldsymbol{x}_k) = \boldsymbol{p}_r = \boldsymbol{H}_k(\boldsymbol{x}_k, \boldsymbol{q}_r, \boldsymbol{r}_r)\boldsymbol{p}_c, \tag{16}$$

$$= \boldsymbol{M}\left(\boldsymbol{R}_{rk} + \alpha_k \boldsymbol{r}_{rk}\boldsymbol{\mu}_k^T\right)\boldsymbol{M}^{-1}\boldsymbol{p}_c, \tag{17}$$

with $\boldsymbol{q}_r$ and $\boldsymbol{r}_r$ taken from the state vector associated with the keyframe $\boldsymbol{x}_r$. The rotation $\boldsymbol{R}_{rk}$ and displacement $\boldsymbol{r}_{rk}$ describing the camera movement between frames $\mathcal{C}_c$ and $\mathcal{C}_r$ are calculated from $\boldsymbol{x}_k$ and $\boldsymbol{x}_r$. As a result, Eq. (15) enables $\boldsymbol{I}_r$ to be used as part of the estimator's measurements.

Based on the results of applying both homography constraints, an observation vector is obtained by stacking all elements belonging to two entire images $\boldsymbol{I}_c$ and $\boldsymbol{I}_r$. Subsequently, we follow the IEKF state update framework to produce the a-posteriori estimate $\boldsymbol{x}_k^+$ [11], [14].

D. Keyframe Management

A system is devised to maintain a collection of keyframe candidates and select an active keyframe to be used as a reference for comparison with the current image frame through the regular homography model as presented above. This routine is executed after the main IEKF update. First, it evaluates whether to archive the current keyframe, if so, whether to retreive a previously used keyframe from the pool or spawn a new keyframe from the latest acquired image.

1) Keyframe Archiving: A strategy is developed to decide when to discard the current keyframe while attempting to retain the keyframe for tracking as distantly as possible. Two determining factors are the size of the overlapped region between the current image and the reference and the overall magnitude of the image gradient in the overlapped region. The image gradient is considered to ensure the overlapped area contains sufficient texture for meaningful comparison through the homography model. The intersection over union (IoU) [23] between two images is used to quantify the co-visibility. The current keyframe is kept if the IoU is above the threshold ρ_1 and the mean of the illumination gradient magnitude in the overlapped area is higher than ρ_2. In such cases, the management routine for that time step is terminated. Otherwise the keyframe is discarded and stored in the pool, and the algorithm proceeds to replace the keyframe by either bringing back a past keyframe or generating a new keyframe.

Compared to the keyframe switching strategy for feature-based or semi-direct VINSs in which the number of tracked features in the current frame is employed as a key indicator for the quality of the keyframe [2], [4], [11], the adopted photometric approach may benefit from the use of global information embedded in whole images as opposed to the sparse salient features or landmarks.

2) Keyframe Retrieval: After archiving a keyframe, the pool is looked into for a keyframe candidate in case the camera returns to a previous spot. Reusing keyframes from distant past potentially further reduces the drift in the estimated position and heading angle. To achieve this, we first compare discarded frames with the current image, and list frames with IoU and photometric difference (computed with Eq. (18) below) above and below thresholds ρ_3 and ρ_4. From the list, the frame with a highest IoU is renewed as an active keyframe. If no image meets the criteria, a new keyframe is to be generated.

Motivated by the technique used during the loop closure in [2], in scenarios where a past keyframe is reactivated, a secondary partial update on $\boldsymbol{x}_k$ is performed by minimizing the following cost function using the Gassian-Newton method

$$\min_{\boldsymbol{r}_k, \psi_k, \kappa, \beta} \left\| \boldsymbol{I}_c(\bar{\boldsymbol{p}}_c) - \kappa \boldsymbol{I}_r(\boldsymbol{\pi}\left(\mathcal{H}(\boldsymbol{p}_c|\boldsymbol{x}_k)\right)) - \beta \right\|^2, \tag{18}$$

where ψ_k, the yaw angle, mathematically defined as the rotation angle around the gravity direction obtained by decomposing the rotation $\boldsymbol{q}_k$ into two rotations $\boldsymbol{q} = \boldsymbol{q}_\psi \boldsymbol{q}_g$. The rotation $\boldsymbol{q}_g$ is to align z-axis of the camera frame to the gravity direction and the rotation $\boldsymbol{q}_\psi$ is about the gravity direction. This minor update on $\boldsymbol{r}_k$ and ψ_k is only performed when a past keyframe is retrieved from the pool and affects only the four unobservable states. The estimator then moves forward to the next time step after this partial state update.

3) Keyframe Spawning: When required, the current image is designated as a new keyframe. This occurs when the image conditions change or the camera travels to a new scene.

IV. Experimental Evaluation

Various real-world flight experiments were conducted to assess our approach in terms of accuracy and computational cost. Root Mean Squared Errors (RMSE) of the estimated states with respect to the ground-truth are used for evaluation via the open-source tool*. In section IV-C, we first validate the effectiveness of the proposed measurement model in Eq.

*https://github.com/ethz-asl/trajectory_toolkit

(15) and compare it against the traditional feature-based or LK method [24]. A comparison between the proposed method and two benchmark VINS is provided in section IV-D. Lastly, the recovery performance of the plane primitive is evaluated under the flights over planes of the different incline angles.

A. Experimental Setup

Flight datasets were collected with the onboard IMU and camera of a Parrot Bebop 2 running the open-source Paparazzi software[†] as shown in Fig. 3(a). A motion tracking system (NaturalPoint, OptiTrack) was used to provide the ground-truth position and orientation, allowing the true state to be evaluated.

The built-in visual-inertial sensor of Bebop 2 features an MPU 6050 from InvenSense and a MT9V117 camera from ON Semiconductor. The IMU outputs specific accelerations and angular rates at 500 Hz. To attenuate the disturbance, a low-pass filter was employed. The IMU data were then downsampled to 100 Hz for the state prediction. Grayscale images of size 240×240-px were acquired at 50 Hz. Note that dropped frames occur occasionally (up to 200 ms) due to the restrained computation (Cortex A9 CPU). IMU readings and images were post-processed on a laptop (Intel Core i5-8250U CPU at 1.6GHz). The data collection allows several estimation strategies to be compared using the same datasets. To verify the proposed estimation strategy, the algorithm was implemented in C++[‡]. All estimates were obtained with the same set of parameters after tuning for the best results and compared to the ground-truth approximately 5 s after taking off. For all cases, the initial estimates α_0 and velocity $\boldsymbol{v}_0$ were set to 10.0 m^{-1} and 0.0 ms^{-1}. The initial normal $\boldsymbol{\mu}_0$ was $[0, 0, 1]^T$ and the rotation $\boldsymbol{q}_0$ was initialised according to the accelerometer readings. The IEKF termination condition is when the iteration step reaches three or the norm of state correction is lower than the set threshold.

B. Flight Data Collection

For validation, we performed eight flights over horizontal ground covered by a pattern shown in 3(c) and recorded the measurements. The repetitive texture was selected as it features salient corners and edges. Among eight flights, the robot was remotely controlled to follow two types of trajectories for over 120 s: four with arbitrary trajectories covering an approximate $1.2 \times 1.2 \times 1$-m volume and another four with a circular path in the horizontal direction and sinusoidal path along the vertical direction. These flight regimes were tested to inspect the performance of different methods in various scenarios. Among them, flight ④ was recorded when the camera was observing the non-planar ground as shown in Fig. 3(b).

C. Effectiveness of the Proposed Measurement Model

In addition to correlating consecutive images via the continuous homography model [14], the use of the photometric difference between the current frame and keyframe as the IEKF innovation term is one key feature of the proposed method. To verify the benefit contributed by the integration of the homography meassurement model, we first compared the results of the proposed method (DKF) against the estimation without the incorporation of the keyframe information in the state update stage (notated as DVIO). Furthermore, to highlight the robustness of the featureless approach (as it is not susceptible to feature tracking errors), the proposed direct method is compared with an alternative version that employs tracked features from the LK algorithm in place of photometric measurements from entire images.

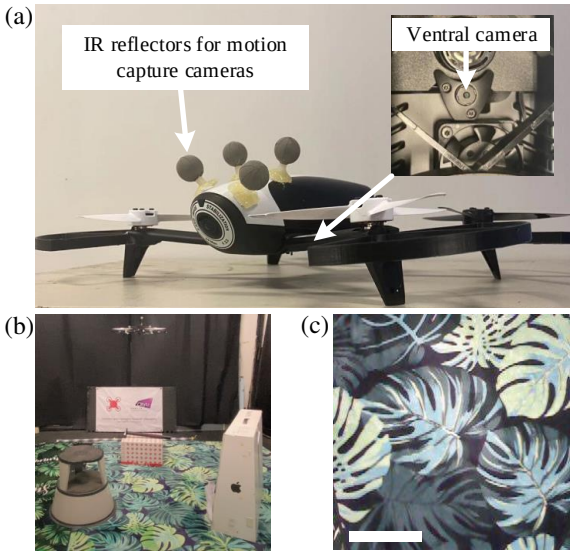


Fig. 3. (a) A Bebop 2 quadrotor and the built-in downward-facing low-resolution camera on Bebop 2. (b) The scene with non-planar objects. (c) The texture captured by the built-in camera of Bebop 2 used for validation flights. The motion capture system was employed for ground-truth measurements and position control of the robot. The scale bar in (c) indicates 0.1 m.

For the DKF and DVIO, images were downsampled from 240×240 px to 72×72 px. The only difference between these two methods is the presence of the measurement model in Eq. (15) and the subsequent keyframe management for the DKF. For the LK-based estimator, the feature tracking pipeline takes after [2]. That is, 50 Harris corners [25] were extracted from 240×240 px images. These corners were then tracked by the pyramidal LK method over consecutive images with 20×20 patch size and three image levels. The keyframe was substituted by the latest frame until the amount of the tracked features was less than a criterion $\rho_5 = 10$. The keyframe retrieval procedure was excluded for LK version due to the difficulty stemmed from the lack of a sophisticated feature descriptor and precise motion recovery. Similar to [14], Huber loss function was introduced to the measurement to improve robustness against feature tracking errors.

Fig. 4 depicts the estimation results from flight ① in detail. It can be seen that the estimates of observable quantities: attitude (roll and pitch angles) and velocity, from all versions do not display noticeable difference. However, the estimates of the position and yaw angle from the full algorithm outperform the other two implementations.

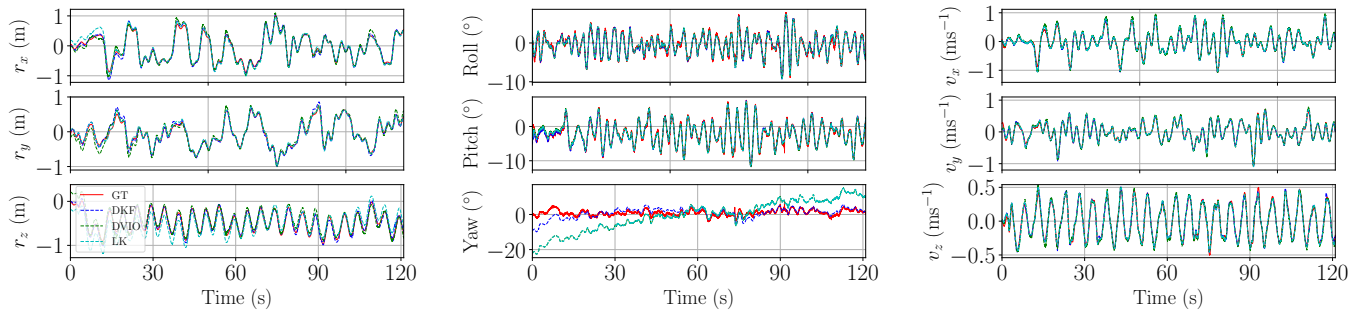
Table I shows estimation results from eight fligths in terms of the RMSEs of the estimated position (Pos), linear velocity (Vel), vehicle's inclination angle (Inc) and yaw angle. Similar to the finding from flight ①, estimation errors of

[†]https://wiki.paparazziuav.org/wiki/Main_Page
[‡]https://github.com/ris-lab/dvio-homo

TABLE I
COMPARISON OF THE ESTIMATION RESULTS FROM THE DKF, DVIO AND LK METHODS.

Flight	Trajectory	$\|\boldsymbol{v}\|_a$,$\|\boldsymbol{v}\|_m$ † (cm/s)	Pos RMSE (cm)			Vel RMSE (cm/s)			Inc RMSE (°)			Yaw RMSE (°)		
			DKF	DVIO	LK	DKF	DVIO	LK	DKF	DVIO	LK	DKF	DVIO	LK
①	Arbitrary	51,127	**11.1**	27.3	39.4	6.6	7.1	**5.9**	**0.5**	1.0	1.0	**4.1**	7.3	7.9
②		65,137	**5.8**	63.3	30.3	**7.3**	8.1	7.9	1.0	**0.5**	1.5	**1.4**	58.1	36.1
③		77,158	**12.7**	56.9	51.9	**8.6**	9.2	9.2	1.3	**0.6**	1.7	**0.8**	26.5	9.0
④		56,133	**25.0**	51.6	116.6	10.2	11.8	**8.4**	**0.8**	0.8	0.6	**0.8**	52.9	42.2
⑤	Circular	100,130	**13.2**	145.1	45.4	6.8	**8.2**	8.0	**1.4**	1.6	1.6	**0.7**	99.8	24.3
⑥		138,179	**25.9**	189.2	194.4	**13.7**	14.8	13.9	**1.3**	1.0	1.3	**0.4**	94.2	55.8
⑦		138,179	**11.5**	165.0	161.4	**13.6**	15.0	14.1	**0.9**	1.1	1.3	**1.2**	92.2	56.9
⑧		137,180	**14.6**	168.2	180.6	15.1	15.8	**14.7**	**2.4**	1.9	2.4	**0.7**	98.0	64.3

† $\|\boldsymbol{v}\|_a$ and $\|\boldsymbol{v}\|_m$ are the root mean squared velocity and maximum speed computed from the motion capture feedback. They qualitatively describe the flight characteristics.



(a) (b) (c)

Fig. 4. Comparison of the estimates from the proposed and benchmak methods from dataset ①. The estimates of (a) Position, (b) Euler angles, and (c) Velocity from the three approaches are plotted against the ground-truth values (GT).

velocity and the inclination angle from all methods exhibit marginal difference, confirming the inherent observability of these states when the flight acceleration is sufficiently excited. Nonetheless, the full implementation outperforms the DVIO and LK variants in terms of the accuracy of the position and yaw angle estimation. This is because for the DVIO, the estimation of the position and yaw angle relies exclusively on the integration of the linear and angular velocity. Whereas in the DKF, the use of keyframes allows a comparison between two distant views to be made, significantly mitigating the drift. For the LK implementation, the inferior performance is likely due to the omission of the second state update associated with the keyframe retrieval step and the feature tracking errors. As found in [14], even the adoption of the Huber loss function cannot completely eliminate outliers from the feature tracking process. This could be further improved with an outlier rejection strategy, such as an application of the epipolar constraint between image correspondences [2]. On the other hand, in the direct methods, the homography projective constraint is inherently robust from the use of the large number of pixels from entire images via Eqs. (14) and (16). Overall, the results demonstrate the contribution of the homography constraint through the use of keyframes and the inherent robustness of the direct approach. An inspection of flight ④ indicates that violation of the single planar condition somewhat reduces the estimation quality as anticipated, but the DKF still produces estimates with acceptable accuracy.

In regard to computational demand, the average time consumption per frame from all three implementations over all sequences are DKF: 2.2 ms, DVIO: 1.2 ms and LK: 3.3 ms. This implies that all three variants are lightweight whereas DKF and DVIO are slightly more efficient than LK thanks to the direct manipulation of image intensities and downsampling. The incorporation of the homography mapping through keyframes is benefitical to position and yaw angle estimates without severely affecting the efficiency.

D. Comparison of the Proposed Direct Method with Two State-of-the-art VINS

We further compare the DKF method against two state-of-the-art VINS: VINS-Mono [2] and ROVIO [11] using their published C++ codes. The two regimes as well as the proposed method provide 6D pose estimates.

VINS-Mono is a variant of a visual-inertial SLAM system rather than a front-end. It features an accurate joint optimization of visual inertial information, loop closure, and map merging and reuse [2]. For comparison, both the pose estimates from the sliding window estimator (VINS) and loop closure (VINSL) were logged out. In contrast to VINS-Mono, ROVIO is characterized as a robust and fast visual-inertial front-end. It leverages an IEKF framework by tightly integrating patch-based photometric feedback as its Kalman innovation term. For comparison with the DKF estimates, we used the default ROVIO parameter configuration, which has been well-tuned to achieve a balanced trade-off between accuracy and efficiency. The number of tracked features per frame is set to 25 and the patch size to 6×6. The second and third levels are employed for tracking the multiple level features.

Besides the data from Bebop 2 in Sec. IV-C, seven datasets from flights over horizontal ground collected in [14] are used to supplement the assessment as flights ⑨ to ⑮. The extra datasets were previously collected by an IMU-camera

TABLE II
COMPARISON OF THE ESTIMATION RESULTS FROM THE DKF, ROVIO, VINS AND VINSL METHODS.

Flight	$\|\boldsymbol{v}\|_a$,$\|\boldsymbol{v}\|_m$ (cm/s)	Pos RMSE (cm)				Inc RMSE (°)				Yaw RMSE (°)			
		DKF	ROVIO	VINS	VINSL	DKF	ROVIO	VINS	VINSL	DKF	ROVIO	VINS	VINSL
①	51,127	**11.1**	34.2	24.5	32.4	**0.5**	3.6	5.6	3.6	4.1	3.5	5.7	**0.5**
②	65,137	**5.8**	91.2	47.9	20.6	1.0	**0.9**	2.6	2.0	1.4	11.9	11.3	**0.4**
③	77,158	**12.7**	137.5	44.1	20.8	1.5	**0.7**	1.8	1.2	**0.8**	12.0	8.5	**0.8**
④	56,133	25.0	99.9	26.0	**5.6**	**0.8**	5.5	2.4	1.8	0.8	0.9	1.3	**0.2**
⑤	100,130	**13.2**	448.6	157.5	14.8	1.4	1.1	**1.0**	1.4	**0.7**	106.6	48.6	2.8
⑥	138,179	25.9	447.1	196.2	**19.2**	1.3	1.0	**0.8**	**1.4**	**0.4**	116.0	89.6	7.4
⑦	137,179	**11.5**	90.7	335.6	29.3	**0.9**	**0.9**	15.8	4.8	**1.2**	30.2	28.0	**1.2**
⑧	137,180	**14.6**	*[1]	175.4	14.7	2.4	*	1.9	**1.4**	**0.7**	*	45.3	**0.7**
⑨	41,83	6.0	2.7	6.8	**2.1**	**0.4**	0.3	1.1	0.5	1.6	**0.2**	0.7	**0.2**
⑩	36,113	2.6	5.9	4.7	**2.2**	0.4	0.4	1.4	**0.3**	0.2	1.0	0.7	**0.1**
⑪	41,94	3.9	2.7	3.9	**2.5**	0.8	**0.2**	0.3	**0.2**	**0.2**	0.8	**0.2**	**0.2**
⑫	36,64	**2.6**	6.3	25.1	3.7	0.7	0.6	3.5	**0.3**	0.3	1.8	0.2	**0.2**
⑬	38,86	**2.0**	26.9	3.1	2.9	**0.4**	0.5	0.7	0.7	**0.2**	1.0	**0.2**	0.3
⑭	76,194	9.0	*	16.7	**3.2**	**0.5**	*	1.3	0.9	3.1	*	1.2	**0.2**
⑮	98,275	19.1	*	10.9	**8.0**	**1.1**	*	1.5	1.8	2.5	*	7.8	**1.2**

[*] ROVIO failed to initialize whan applied to these datasets. This is because the estimator requires near-zero acceleration to compute the initial attitude. These datasets, however, were recorded with an initially large non-zero acceleration.

rig (MYNT AI, MYNT EYE) mounted on a flying robot. The camera generated 752×480-px images at 30 Hz with synchronized IMU data at 100 Hz. For estimations, images were downsampled to 90×58 px for the proposed DKF whereas full-size images were used for two benchmark VINS.

Table II lists the RMSEs of the estimates obtained from four implementations. First, focusing on the results from the Bebop data, the inclination errors for all approaches are invariably small. However, the results indicate relatively noticeable RMSEs in the position and yaw angle estimates from ROVIO and VINS compared to the other two methods. This is attributed to the efficient use of keyframes by DKF and the robust loop closure technique to suppress accumulated errors along the four unobservable degrees of freedom. A closer inspection suggests that the proposed DKF consistently produced position estimates with smaller errors except the results from flight ④. We hypothesize that the benchmark methods suffer from the use of poor quality images (240×240 px) provided by the onboard camera of Bebop 2 whereas the direct approach of DKF is relatively robust as it does not rely on image features. Regarding the better performance of flight ④ for VINSL, the distinct non-plannar objects (when compared to repetitive textures) improves the chance of positive loop closure.

For the second set of results from the standalone MYNT camera, the RMSEs from all methods are generally lower compared to those of the built-in Bebop 2 camera. This is possibly due to superior image quality (higher resolution and global shutter) and the precise camera-IMU synchronization. It should be highlighted that the RMSEs from DKF are similar to those from the other three approaches, though VINSL performed best overall. Still, it can be concluded that the accuracy of the pose estimation from the proposed estimator rivals those of two state-of-the-art regimes.

In terms of computation, the time costs per frame averaged from all 15 sequences from three schemes are shown in Fig. 5(a). Note that for VINS-Mono, three threads operate in parallel and only the time cost of the sliding optimization is counted. The plots show that, at less than 5 ms per frame on average, the proposed estimator is approximately 5-20 times faster than both VINS-Mono and ROVIO. The exceptional efficiency is a consequence of imposing the requirement of a single flat surface in view. In addition, the DKF benefits from the aggressive downsamping of original images. To elucidate how the downsampling affects the estimation accuracy and time cost, Fig. 5(b) presents the average time per frame and RMSE of the position estimates using the flight sequence ③ at various downsampled image sizes. In producing these results, the weights between the prediction and measurement model were re-tuned to account for the change in image sizes. It can be seen from Fig. 5(b) that the time cost grows almost quadratically when the image size increases. Interestingly, the RMSEs are not visibly influenced by the image size. This is possibly attributed to the less predictable nature of the keyframe retrieval step and the original image quality.

E. Plane Primitive Estimation and Initial Distance Value

The radical improvement in efficiency compared with point-based VIOs originated from the formulation of the scene structure as a single plane. This substantially reduces the state dimension from the order of hundreds in feature-based VIOs to tens in the proposed approach. In this section, the accuracies of the estimated plane normal and the orthogonal distance from the camera center to the plane are assessed.

Additional four flights with Bebop 2 following an arbitrary trajectory were carried out over surfaces with the angles of inclination from 0°to 30°. The estimator's parameters remained unchanged from the previous section. The estimated angle between the estimated vertical and the plane normal over time is shown in Fig. 5. The figure shows that the estimated angles start with large uncertainty and thereby vary evidently in the first five seconds. Subsequently the angles converge close to the groundtruth (within a few degrees). While it is possible to recover the actual initial normal using homography decomposition [4], it requires extra feature extraction and tracking process unsupported by the direct implementation. In these flights, the RMSEs of the estimated distances are 0°: 4.8 cm, 10°: 7.2 cm, 21°: 8.7 cm and 30°: 7.3 cm. The results verify that the proposed estimator is able to deal with planes at different inclination angles.

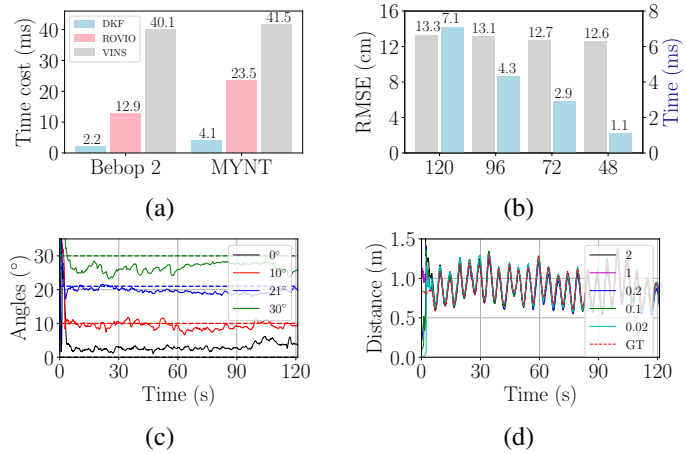


Fig. 5. (a) Time cost per frame of three estimation schemes averaged from all sequences obtained from both Bebop 2 and MYNT cameras. b) The position RMSE (gray) and time cost per frame (blue) using the sequence ③ with the various downsampled image sizes: 120×120, 96×96, 72×72 and 48×48. (c) The angle between the estimated normal vector and gravity vector on different incline planes. The dash lines are the ground-truth angles. (d) The distance estimates using different initial distances to the plane: 2 m, 1 m, 0.2 m, 0.1 m and 0.02 m. For the all previous experiments, the initial guess of the distance was set to 0.1 m ($\alpha = 10$ m^{-1}).

To further assess the reliability of the scale recovery, experiments were conducted using different initial guesses for the orthogonal distance from the camera center to the plane, from 0.02 m to 2.0 m with the plane of zero inclination. As illustrated in Fig. 5 (d), the estimated altitudes using the different initial values converge to the ground-truth within around 5 s. The RMSEs of the estimates are 2 m: 7.6 cm, 1 m: 4.7 cm, 0.2 m: 6.5 cm, 0.1 m: 4.8 cm and 0.02 m: 5.1 cm respectively. These results corroborate the robustness in the scale recovery ability of the estimator.

V. Conclusion

In this paper, we have proposed a computationally efficient VIO to recover the camera motion and a single plane landmark. The contribution lies in the direct use of photometric feedback over entire low-resolution images, encoded by both continuous and regular homography models, in the Kalman innovation term. Extensive flight experiments were carried out to assess the performance. The results prove that the correlation between the current frame and keyframe effectively suppresses the estimation's drift and offers better accuracy and efficiency than an indirect implementation. Further analysis reveals that the proposed scheme compares favorably against two state-of-the-art VINS when it comes to the accuracy of the pose estimates. It should be highlighted that the single plane assumption in the proposed estimator permits it to be ≈15-30 times faster than the two benchmark VINS. Finally, additional flights were performed to showcase the estimator's ability to infer non-horizontal planes' parameters. Overall, this work offers an attractive lightweight navigation solution for small aerial robots with limited computational power.

References

[1] 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, no. 3, pp. 314–334, 2015.

[2] 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.

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

[4] R. Mur-Artal and J. D. Tardos, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[5] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 3565–3572.

[6] Z. Huai and G. Huang, "Robocentric visual–inertial odometry," *The International Journal of Robotics Research*, 2019.

[7] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proceedings Ninth IEEE International Conference on Computer Vision*, 2003, pp. 1403–1410 vol.2.

[8] H. Strasdat, J. M. Montiel, and A. J. Davison, "Visual slam: why filter?" *Image and Vision Computing*, vol. 30, no. 2, pp. 65–77, 2012.

[9] H. Jin, P. Favaro, and S. Soatto, "A semi-direct approach to structure from motion," *The Visual Computer*, vol. 19, no. 6, pp. 377–394, 2003.

[10] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "Svo: Semidirect visual odometry for monocular and multicamera systems," *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249–265, 2016.

[11] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended kalman filter based visual-inertial odometry using direct photometric feedback," *The International Journal of Robotics Research*, vol. 36, no. 10, pp. 1053–1072, 2017.

[12] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European conference on computer vision*. Springer, 2014, pp. 834–849.

[13] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2017.

[14] S. Zhong and P. Chirarattananon, "Direct visual-inertial ego-motion estimation via iterated extended kalman filter," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1476–1483, 2020.

[15] V. Grabe, H. H. Bülthoff, D. Scaramuzza, and P. R. Giordano, "Nonlinear ego-motion estimation from optical flow for online control of a quadrotor uav," *International Journal of Robotics Research*, vol. 34, no. 8, pp. 1114–1135, 2015.

[16] M.-D. Hua, N. Manerikar, T. Hamel, and C. Samson, "Attitude, linear velocity and depth estimation of a camera observing a planar target using continuous homography and inertial data," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1429–1435.

[17] B. Guan, P. Vasseur, C. Demonceaux, and F. Fraundorfer, "Visual odometry using a homography formulation with decoupled rotation and translation estimation using minimal solutions," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2320–2327.

[18] R. Mebarki, V. Lippiello, and B. Siciliano, "Vision-based and imu-aided scale factor-free linear velocity estimator," *Autonomous Robots*, vol. 41, no. 4, pp. 903–917, 2017.

[19] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An invitation to 3-d vision: from images to geometric models*. Springer Science & Business Media, 2012, vol. 26.

[20] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Camera-imu-based localization: Observability analysis and consistency improvement," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 182–201, 2014.

[21] J. Shu and P. Chirarattananon, "A quadrotor with an origami-inspired protective mechanism," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3820–3827, 2019.

[22] Y. Chen, H. Zhao, J. Mao, P. Chirarattananon, E. F. Helbling, N.-s. P. Hyun, D. R. Clarke, and R. J. Wood, "Controlled flight of a microrobot powered by soft artificial muscles," *Nature*, vol. 575, no. 7782, pp. 324–329, 2019.

[23] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[24] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004.

[25] J. Shi and C. Tomasi, "Good features to track," in *IEEE Conference on Computer Vision & Pattern Recognition*, 2002.