Robotics and Autonomous Systems 76 (2016) 1–14

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Metric sensing and control of a quadrotor using a homography-based visual inertial fusion method

Ping Li *, Matthew Garratt, Andrew Lambert, Shanggang Lin

School of Engineering and Information Technology, The University of New South Wales, Australia

HIGHLIGHTS

- The 1-SVD method is shown to be superior over the traditional 2-SVD approach.
- Robustness of the LK algorithm is improved using a transformed binary image.
- A visual inertial fusion method is proposed to estimate metric speed and distance.
- Closed-loop flight proves our approach is suitable for general flight of a MAV.

ARTICLE INFO

Article history:
Received 26 May 2015
Received in revised form
11 November 2015
Accepted 27 November 2015
Available online 4 December 2015

Keywords:
Visual inertial fusion
Optic flow
Homography
Micro Aerial Vehicles

ABSTRACT

The combination of a camera and an Inertial Measurement Unit (IMU) has received much attention for state estimation of Micro Aerial Vehicles (MAVs). In contrast to many map based solutions, this paper focuses on optic flow (OF) based approaches which are much more computationally efficient. The robustness of a popular OF algorithm is improved using a transformed binary image from the intensity image. Aided by the on-board IMU, a homography model is developed in which it is proposed to directly obtain the speed up to an unknown scale factor (the ratio of speed to distance) from the homography matrix without performing Singular Value Decomposition (SVD) afterwards. The RANSAC algorithm is employed for outlier detection. Real images and IMU data recorded from our quadrotor platform show the superiority of the proposed method over traditional approaches that decompose the homography matrix for motion estimation, especially over poorly-textured scenes. Visual outputs are then fused with the inertial measurements using an Extended Kalman Filter (EKF) to estimate metric speed, distance to the scene and also acceleration biases. Flight experiments prove the visual inertial fusion approach is adequate for the closed-loop control of a MAV.

1. Introduction

Rotary-wing Micro Aerial Vehicles (RMAVs), with the capability to hover, take off and land vertically, can operate effectively in low-altitude and cluttered environments. Coupled with their great agility, RMAVs have an advantage over ground robots and fixed-wing MAVs for tasks like bridge or vessel inspection [1] and search and rescue in a partially collapsed building. A comprehensive review for the navigation and control of RMAVs is presented in [2]. Fast, accurate and robust pose estimation is crucial for successful applications of RMAVs. GPS signals are not reliable in a confined space, where RMAVs are often deployed. This paper describes state estimation based on a monocular camera and an Inertial Measurement Unit (IMU). These two sensors are widely available and can be made very compact, energy-efficient and light-weight (can be a few grams [3]).

Many algorithms have been proposed for motion estimation (ME) with cameras, using either feature based approaches or direct methods. The former tries to establish feature (corners, lines or blobs) correspondences between two or more images before computing the camera motion and possibly scene geometry based on a motion model. The latter, as the name suggests, directly solve for the camera motion by minimizing a cost function containing the motion parameters [4]. Feature-based methods, although they only use part of the information in an image, are dominant in the literature because they are easier to implement and more computationally efficient. The two methods are combined to achieve a real-time and high-accuracy visual odometry system in [5].

* Corresponding author.
E-mail addresses: ping.li@student.adfa.edu.au (P. Li), M.Garratt@adfa.edu.au (M. Garratt), A.Lambert@adfa.edu.au (A. Lambert), Shanggang.Lin@student.adfa.edu.au (S. Lin).

http://dx.doi.org/10.1016/j.robot.2015.11.011
0921-8890/

The movement of features in two images is often termed as optic flow (OF) and some insects are believed to rely on OF for navigation [6]. Insect-mimicking behaviours have been achieved using OF on robotic platforms, like wall-centring [7], visual homing [8], and obstacle avoidance [9]. OF has also been adopted for hover control [10] and landing [11]. Because they only consider frame-to-frame motion, OF-based techniques are usually subject to long-term positional drift [12]. A snapshot image is captured and acts as a reference for the following images to prevent drifting in hover [13,14]. This method is only effective within the local area where the snapshot image is taken. The same problem exists for methods using artificial patterns [15]. The simultaneous localization and mapping (SLAM) algorithms track features over multiple frames, build a map of these features while at the same time estimating the camera pose. The filter-based SLAM [16] estimates camera pose and feature positions together using an Extended Kalman Filter (EKF) while the keyframe-based SLAM [17] optimizes the feature map using bundle adjustment and recovers camera pose from tracking features of known positions in the map. Both SLAM algorithms have been implemented on RMAVs in [18,19]. A bio-inspired SLAM system, termed as RatSLAM, is also designed for robot navigation [20]. Although attractive, SLAM requires much computation for carefully maintaining a map, detecting loop-closure and performing a large-scale optimization task. In fact, because the complexity of SLAM increases with the scale of the environment, it often reduces to a sliding window odometry approach for constant-time processing [3]. This paper focuses on OF based methods since they can be very lightweight and easy to implement. They can help automatic initialization of a prior map for the monocular SLAM method, and act as a fallback when the latter fails [21]. Specifically, in this paper, a homography model is adopted. The epipolar geometry is independent of the scene structure, however is degenerate over planar scenes or when the camera is purely rotating around the principal axis. Although a homography model assumes planarity of the scene, it is still applicable to scenes where a dominant plane or multiple planes [22] are present. Such scenes are easily found, especially in the man-made world. Using the idea of 'Virtual Parallax, the homography model can be extended to handle non-planar scenes [23]. Normally, after the homography matrix is computed, it is further decomposed to estimate surface normal, rotation and translation (up to scale) with two possible solutions [24]. In this paper, we explain how to compute the homography matrix from the Jacobian motion model [25]. With a new parametrization, it is shown that the unscaled velocity is directly known from the homography matrix, avoiding the need to perform Singular Value Decomposition (SVD) after the homography matrix is calculated. Using real images, we show that our method is more accurate and robust than the method involving the decomposition of homography matrix.

A monocular camera suffers from scale ambiguity. The Parrot AR. Drone scales OF estimation using a downward-facing sonar sensor [26], limiting the operation of the vehicle close to the ground. Sonar sensors may also fail over soft or uneven ground. A scanning laser range finder consumes much power and is cumbersome for a MAV. Whilst the scale can be resolved with another visual sensor, this method requires the two sensors to be placed apart by a proper distance [27,28], making the platform less compact than a monocular system. The method in [28] becomes degenerate for static motion and establishing feature correspondence for a stereo pair is not trivial [27]. In our previous work [29], the scale ambiguity is resolved by fusing the homography based speed estimation (up to a scale) with the acceleration measurements in an EKF framework. The proposed Visual Inertial Fusion (VIF) method is able to estimate metric speed, distance to the terrain and acceleration biases. This paper further examines the OF algorithm on public datasets, provides more details and discussion on the homography model, considers outlier rejection using the popular Random sample consensus (RANSAC) algorithm [30], and evaluates the accuracy of the ME over different textures. Flight tests have also been carried out to prove its effectiveness for closed-loop control of a MAV.

2. Related work

In real-world applications, feature correspondences often contain outliers which could seriously corrupt the solution and thus should be detected and excluded from the estimation. A standard method is the RANSAC algorithm, which computes hypotheses from a randomly generated subset of samples in the dataset and then evaluates the reliability of the results using the rest of the data [30]. The correct solution is chosen as the hypotheses supported by the maximum number of samples. The computational complexity of RANSAC increases dramatically with the minimal number (m) of correspondences required to solve the motion model:

$$itera = \frac{\log(1 - p_s)}{\log(1 - (1 - r_{out})^m)} \tag{1}$$

where *itera* is the number of required iterations to guarantee a probability of p_s (0.99) for successful estimations; r_{out} means the ratio of outliers in the feature correspondences. To reduce *itera*, a simplified motion model can be chosen so that fewer points are needed (m is smaller). In [31], a 1-point algorithm is developed by taking advantage of the non-holonomic constraints of wheeled vehicles. The restriction of planar motion has led to the development of a 3-point algorithm [32], and even a 1-point RANSAC approach further aided by the relative rotation estimation from the inertial sensors [33]. A 4-point method is proposed, also using relative rotation angle measurements from another sensor [34]. The novelty is that no inter-sensor calibration is required. The Manhattan world assumption is employed to simplify the homography model [35]. Most of these approaches make strict assumptions about the vehicle motion or scene structures and may not be applied to general cases. Another way of decreasing *itera* is to reduce the outlier ratio by improving the OF algorithm, which is the focus of this paper.

Most OF techniques can be classified as gradient based methods [36] or template matching (TM) based approaches [37]. The former is further divided into local and global schemes [38]. In contrast to the local counterparts, global methods regularize OF through a smoothness term in addition to a data term within an optimization framework. They are very accurate but also computationally demanding. TM has the potential to handle large OF but lacks sub-pixel accuracy and its computational cost scales quadratically with the search range. The Lucas Kanade (LK) algorithm [36] is a local gradient based method and is still widely used since it gives good performance in terms of accuracy, density of motion field and low computational burden [39]. However, like most other OF algorithms, the LK method requires constant brightness of the scene, which is often not the case in reality. To improve the robustness of the intensity based LK, it is proposed in this paper to pre-process the intensity image for extracting information less sensitive to lighting changes. In this way, the main body of the OF algorithm stays unchanged, in contrast to methods that model illumination changes explicitly and needs to solve more parameters [40]. Specifically, the technique in [41] is used that obtains a binary image by comparing the intensities of neighbouring pixels, which has been proven to greatly enhance the robustness of the TM algorithm. This binarization method only requires one comparison per pixel and thus is preferred over other pre-processing approaches that obtain gradient orientation [42] or

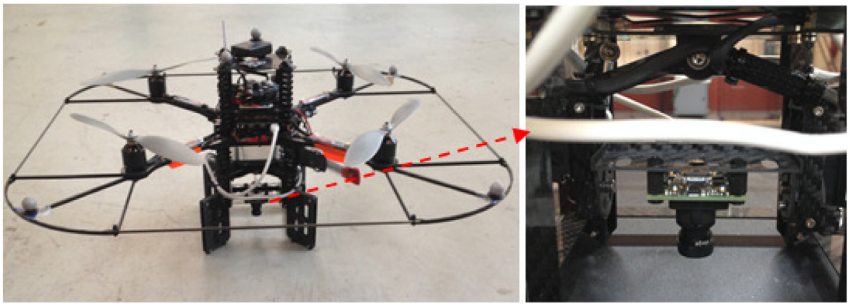

Fig. 1. The AscTec Pelican (left) and the camera installed on the pan–tilt unit (right).

image moments [43]. Experimentation using public datasets and real sensory data from our platform proves that, as the input for LK, the transformed binary image produces a higher quality OF field than the intensity image over different textures. This translates to less iterations for the RANSAC algorithm.

The number of observable states in the VIF system is decided by the types of visual algorithms and the motion patterns of the vehicle. Generally, it is possible to estimate the visual scale factor, the camera-IMU transformation, the IMU biases and the vehicle pose. However, the vehicle has to rotate about at least two axes and has non-zero acceleration along at least two axes to achieve the full system observability [44]. If the frame-to-frame motion is considered, global position and yaw angle are not observable [21]. However, if the world frame is anchored in the local terrain plane, yaw angle can be computed exploiting the surface inclination [45]. This work is expanded in [46] to achieve a faster initialization and enhanced robustness against temporal misalignment between the camera and IMU. To make yaw and position observable in the global frame, features in the current image need to be registered against a global map (e.g. SLAM) [21]. The observability of the coupled system is improved by taking into account the dynamic model of the quadrotor [47]. The model-aided fusion approach is able to recover the visual scale factor in all motion patterns, contrary to general VIF methods that require persistent excitation of the vehicle. However, the method relies on an accurate modelling of the vehicle.

For VIF, the loosely coupled and tightly coupled methods are discriminated by whether the visual estimation is performed independently of the inertial readings. In [3], the SLAM algorithm is treated as a black-box that processes a sequence of images for six-degree-of-freedom pose estimation. Then, the visual output is fused with inertial data with a separate EKF. This belongs to the loosely coupled approaches. For tightly coupled approaches, inertial and visual information are combined in a single filter [48, 49], which are more accurate but also more complex than the loosely coupled methods [48]. For the visual part, several VIF methods make use of the two-view geometry [50,51] or three-view geometry [52]. The observation of a single feature on a static plane is directly treated as the system measurement for an Unscented Kalman Filter (UKF). However, this method is only tested in simulation [53]. In [54], the continuous and discrete forms of the homography model are combined in their VIF scheme. The frame-to-frame motion is represented by the continuous form while the discrete form is used to describe motion between non-adjacent frames. A novel VIF scheme is proposed that includes a history of vehicle poses in an EKF, termed as the multi-state-constrained Kalman Filter (MSC-KF) [55]. This framework can optimally utilize the information of tracked features and is demonstrated to be more consistent than augmenting the EKF states with feature positions, as done in the EKF based SLAM algorithm [16]. In [56], it is shown that the time offset between an IMU and a camera can be estimated, in addition to the visual scale factor, IMU sensor biases, camera pose and inter-sensor transformation. The idea is to build a throw-and-go system [46], however, including more states in a filter results in a larger system that needs to propagate a larger covariance matrix at the IMU sampling frequency. The camera-IMU transformation can be estimated beforehand so that it does not need to be updated online, reducing the size of the VIF system, where the IMU sensor biases are also ignored [51].

In this paper, a semi-tightly coupled system is built by involving inertial measurements in the development of the homography model. Our work may be most similar to [51]. The novelty of our work is that, firstly, the robustness of the LK OF algorithm is improved using the binary image [41] as input. Secondly, the homography model is developed in a different way and it is proposed to directly obtain the unscaled speed from the homography matrix without performing SVD. Thirdly, rather than putting all the states in a single filter, the three axes are treated separately so that smaller subsystems are handled. An EKF is employed for the estimation in the Z axis. For the X and Y axes, a linear Kalman Filter is used. Closed-loop flight tests have also been carried out based on the VIF estimated states. We do not claim that our method is competitive in accuracy when comparing with the tightly-coupled and self-calibrating VIF systems [46,56,49]. Instead, we aim at constructing a simple and robust VIF system that is easy to implement. The proposed method is evaluated over different textures and is shown to even work reliably over the highly repetitive and poorly-textured ground.

3. System overview

The platform we are using is the AscTec Pelican quadrotor with a maximum payload of 650 g (Fig. 1). One of the two mainboards hosted by the vehicle comprises the AscTec Autopilot, which has an Inertial Measurement Unit (IMU) and two 60 MHz ARM 7 microcontrollers. One ARM7 processor is designated the Low Level Processor (LLP) and the other is designated the High Level Processor (HLP). The LLP samples and processes inertial measurements for attitude (pitch and roll) estimation and attitude (inner-loop) control. The inertial information is also available to the HLP where speed and position control (outer-loop) can be implemented. The other mainboard features an Intel 1.6 GHz Atom processor, 1 GB RAM and a MicroSD card slot for the on-board Ubuntu operating system. A 6100 mhA LiPo battery is used to power the on-board Atom processor and the Autopilot.

An UI-1221LE-M-GL monochrome and global shutter camera is used for our work which has a maximum resolution of 752 $\times$ 480 pixels and a field of view of 58°. The downward-looking camera is installed on the pan–tilt unit (Fig. 1) which counteracts the effects of vehicle pitching and rolling, aligning the principal axis of the camera with the gravity vector. The captured image has large distortion, so we first calibrated the camera and undistorted the image using a look-up table.

Fig. 2 provides an overview of our whole MAV system. The Vicon system, ground station (a laptop) and the vehicle are all

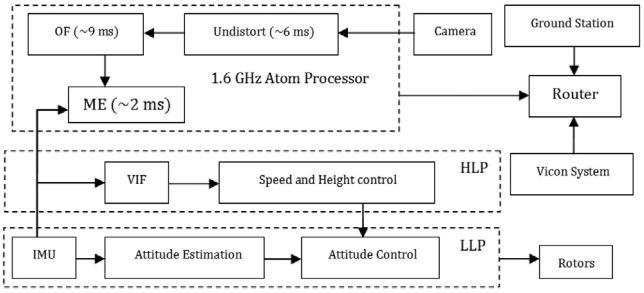

Fig. 2. System overview: visual processing, sensor fusion, control and data communication.

connected to a wireless router and exchange data through the router. The implementation of our algorithm is based on the Robotic Operating System (ROS) [57] and the OpenCV (Open Source Computer Vision) library. The computational time is also shown in Fig. 2 and the overall time for the visual processing is around 20 ms using the on-board Atom processor. That means a 50 Hz update for visual estimation can be achieved. However, we only capture images at 20 Hz and leave some computational power for accommodating other algorithms in our future work. In order to control the vehicle from the ground station, a key-handling function is written that maps keys to height and speed setpoints. For example, the quadrotor takes off to 1 m with the key 't' pressed and climbs 0.4 m higher after pressing 'a'. The details for image processing, motion estimation and sensor fusion are discussed in the following sections.

4. Optic flow (OF)

The Lucas Kanade (LK) algorithm assumes that pixels in a neighbourhood (Ω) have the same image motion and calculates the OF vector (u, v) by minimizing the weighted least square error function:

$$\xi = \sum_{(x,y)\in\Omega} w^2(x,y)(I_x u + I_y v + I_t)^2 \tag{2}$$

where I_x, I_y are the image spatial derivatives, I_t is the image temporal derivative and $w(x, y)$ is the weighting matrix that adjusts the influence of pixels in the proximity.

The LK algorithm is developed with the assumption of constant brightness of pixel intensities, which is often violated in practice due to image noise and varying lighting conditions. Compared to intensities themselves, the relative ordering of pixel intensities is less likely to change under these situations. Based on this idea, the intensity image is transformed into a binary image using the following method [41]:

$$I_b(x,y) = \begin{cases} \max & \text{if } I(x+m, y+n) > I(x,y) \\ \min & \text{otherwise} \end{cases} \tag{3}$$

where I_b denotes the binary image. Setting $m = 1,\ n = 0,\ \min = 0,\ \max = 1$, the transformed binary image is used for template matching in [41]. Pre-processing images using Eq. (3) only takes one comparison per pixel. In this work, setting $\min = 0, \max = 255$, the transformed binary image is used instead of the intensity image in Eq. (2) to calculate OF. The Middlebury datasets [58] are used to evaluate the accuracy of LK with respect to the choice of m. In Table 1, 'Int' means intensity, 'AEE' refers to Average Endpoint Error and 'P05' means percentage of points whose AEE is less than 0.5 pixel [58]. For the binary transformation (BT), it is noted that too small a distance is not desirable most of the time but too large a distance deteriorates the accuracy as well. More low-frequency components are preserved with a larger m value but with a loss of the details in an image (Fig. 3). A good option may be setting m at 2 or 3, and the BT based LK shows comparable or even better accuracy than the intensity based LK on these synthetic datasets without apparent illumination change.

Table 1
The accuracy of the intensity and binary image based LK on the Middlebury Datasets.

	AEE	P05	AEE	P05	AEE	P05	AEE	P05
	Dimetrodon		Grove2		Grove3		Hydrangea	
Int	0.30	12.7	0.32	11.8	1.20	33.7	0.71	21.5
$m = 1$	0.98	26.9	1.56	28.7	1.91	37.7	0.73	18.7
$m = 2$	0.40	14.7	0.59	17.4	1.66	33.5	0.65	19.1
$m = 3$	0.39	18.4	0.49	16.3	1.50	34.0	0.59	19.7
$m = 4$	0.41	20.2	0.44	17.1	1.50	35.3	0.61	21.0
$m = 5$	0.38	18.0	0.46	17.9	1.52	36.7	0.68	22.4
	Rubberwhale		Urban2		Urban3		Venus	
Int	0.32	11.4	2.35	22.7	2.11	31.9	0.85	19.5
$m = 1$	0.29	7.40	1.32	35.9	2.63	54.0	0.76	26.8
$m = 2$	0.29	7.40	1.10	30.2	2.16	47.3	0.57	21.5
$m = 3$	0.30	8.30	1.23	31.0	2.14	47.3	0.58	22.1
$m = 4$	0.32	9.60	1.42	33.4	2.15	47.3	0.58	23.1
$m = 5$	0.33	10.9	1.44	34.4	2.13	47.4	0.60	23.9

5. Motion estimation

5.1. Homography model

The number subscript of a vector or a matrix (e.g. $a_i, i = 1, 2, 3\ldots$) denotes the ith element of the vector or ith column of the matrix, unless specified otherwise. Let us first define an operation that reshapes a 9-element vector h_c into a 3×3 matrix H_c as:

$$M(h_c) = H_c = \begin{bmatrix} h_{c1} & h_{c2} & h_{c3} \\ h_{c4} & h_{c5} & h_{c6} \\ h_{c7} & h_{c8} & h_{c9} \end{bmatrix}. \tag{4}$$

Also, the symbol $[]_\times$ denotes a skew-symmetric matrix formed by a vector $W = [W_x, W_y, W_z]^T$:

$$[W]_\times = \begin{bmatrix} 0 & -W_z & W_y \\ W_z & 0 & -W_x \\ -W_y & W_x & 0 \end{bmatrix}. \tag{5}$$

Consider the following motion model that relates camera motion with OF [25]:

$$\begin{bmatrix} u \\ v \end{bmatrix} * FPS = \begin{bmatrix} \frac{f_x}{Z} & 0 & \frac{-x}{Z} & \frac{-xy}{f_x} & \frac{f_x^2 + x^2}{f_x} & -y \\ 0 & \frac{f_y}{Z} & \frac{-y}{Z} & \frac{-f_y^2 - y^2}{f_y} & \frac{xy}{f_y} & x \end{bmatrix} \begin{bmatrix} V \\ W \end{bmatrix} \tag{6}$$

where FPS is the frame rate; x, y are the image coordinates of the tracked feature points; u, v represent the optic flow; f_x, f_y are the focal lengths; Z is the feature depth; $V = [V_x, V_y, V_z]^T$ is the linear velocity and $W = [W_x, W_y, W_z]^T$ is the angular velocity. Assuming $f_x = f_y = f$, Eq. (6) equals:

$$\begin{bmatrix} u \\ v \end{bmatrix} * \overline{FPS} = \begin{bmatrix} \frac{1}{Z} & 0 & \frac{-\bar{x}}{Z} & -\bar{x}\bar{y} & 1+\bar{x}^2 & -\bar{y} \\ 0 & \frac{1}{Z} & \frac{-\bar{y}}{Z} & -1-\bar{y}^2 & \bar{x}\bar{y} & \bar{x} \end{bmatrix} \begin{bmatrix} V \\ W \end{bmatrix} \tag{7}$$

where $\bar{x} = x/f,\ \bar{y} = y/f$ and $\overline{FPS} = FPS/f$. Suppose a scene point on a plane has a coordinate of (X, Y, Z) in the camera coordinate frame, the following relation holds:

$$N_x X + N_y Y + N_z Z = d \tag{8}$$

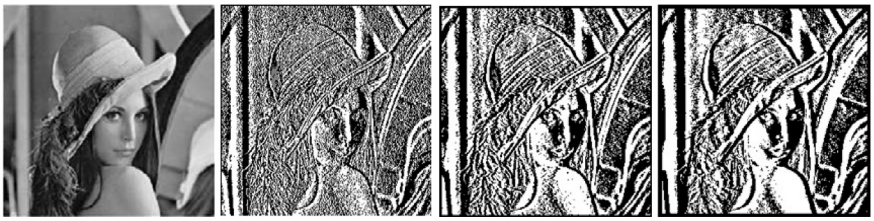

Fig. 3. Three transformed binary images from the Lena image based on Eq. (3) when $m = 1,\ n = 0$; $m = 3,\ n = 0$ and $m = 5,\ n = 0$ respectively.

where $N = [N_x, N_y, N_z]^T$ denotes the surface normal and d is the distance from the camera optical centre to the plane (Fig. 4). With a pinhole camera model, that is $X/Z = x/f = \bar{x},\ Y/Z = y/f = \bar{y}$, we have:

$$(N_x\bar{x} + N_y\bar{y} + N_z)/d = \frac{1}{Z}. \tag{9}$$

Substituting Eq. (9) back into Eq. (7) gives:

$$\begin{bmatrix} u \\ v \end{bmatrix} * \overline{FPS} = \begin{bmatrix} \bar{x} & \bar{y} & 1 & 0 & 0 & 0 & -\bar{x}^2 & -\bar{x}\bar{y} & -\bar{x} \\ 0 & 0 & 0 & \bar{x} & \bar{y} & 1 & -\bar{x}\bar{y} & -\bar{y}^2 & -\bar{y} \end{bmatrix} h_c \tag{10}$$

where $M(h_c) = H_c = [W]_\times + \bar{V}N^T$ and $\bar{V} = [V_x/d, V_y/d, V_z/d]^T$. We define $H_c' = \bar{V}N^T$ and h_c' is the corresponding vector. The rank of H_c' is 1 since the determinant of any 2×2 submatrix is 0.

The vector h_c has 9 elements but has 8 independent parameters ($\|N\| = 1$), which means that at least four point correspondences are needed to solve h_c since every correspondence gives two equations. Stacking Eq. (10) from m (more than 4) feature matches results in an over-determined linear system: $Ah_c = b,\ A \in R^{2m\times 9},\ b \in R^{2m\times 1}$, which has the same solutions as $A^TAh_c = A^Tb$. The solution for $A^TAh_c = A^Tb$ is: $h_c = Se + ks$, where $A^TA = UDS^T$ (SVD); $s = S_9$; $e_i = b_i'/d_i$, d_i is the ith diagonal entry of D; $b' = U^TA^Tb$ and k is an unknown parameter. $A^TA \in R^{9\times 9}$, so it is simpler to decompose A^TA than A if more than 4 features are used.

It is found that $M(s)$ is a diagonal matrix with the three diagonal elements being equal. It is also observed that Eq. (10) still holds by adding a constant to h_{c1}, h_{c5}, h_{c9}. That means the solution of h_c can be rewritten as: $h_c = Se + kI$, where I is the identity matrix. Let us denote the three eigenvalues of the matrix $(M(Se) + M(Se)^T)$ as $\lambda_1, \lambda_2, \lambda_3$ ($\lambda_1 \geq \lambda_2 \geq \lambda_3$), then we have $k = -\lambda_2/2$.

Proof. Since $M(h_c) = H_c = [W]_\times + H_c' = M(Se) + kI$, then: $H_c + H_c^T = H_c' + H_c'^T = M(Se) + M(Se)^T + 2kI$; one can prove that the second eigenvalue of the matrix $(H_c' + H_c'^T)$ is 0 and the second eigenvalue of $(M(Se) + M(Se)^T + 2kI)$ is $\lambda_2 + 2k$, thus we have $k = -\lambda_2/2$.

After h_c (H_c) is calculated, it can be further decomposed to obtain $\bar{V}$, N and W. However, two possible solutions exist [24]. It will be shown that if the angular rates in two of the axes are given, $\bar{V}$, N and the angular rate in another axis can be computed without any ambiguity. Without loss of generality, let us assume $W_x = 0, W_y = 0$, then:

$$H_c = \begin{bmatrix} h_{c1}' & h_{c2}' - W_z & h_{c3}' \\ h_{c4}' + W_z & h_{c5}' & h_{c6}' \\ h_{c7}' & h_{c8}' & h_{c9}' \end{bmatrix}. \tag{11}$$

The rank of the matrix H_c' being 1 is exploited to solve W_z. The determinant of H_c' being 0 yields a 2nd order polynomial in W_z that has two roots. Matrix H_c' has nine 2×2 submatrices. The absolute value of the determinant of these submatrices is added and the correct root is chosen as the one that achieves the minimum sum. After that, H_c' is known and can be decomposed to solve for $\bar{V}$, N: $N = S_1'$; $\bar{V} = H_c'N$, where $H_c' = U'D'S'^T$ (SVD). If all the three angular rates are known from inertial sensors, they can be subtracted so that $H_c = H_c'$ can be directly decomposed to compute $\bar{V}$ and N. In fact, in this case, the parameter k can be calculated using the constraint that the rank of the matrix H_c' is 1, similar to the computation of W_z, except that a 3rd order polynomial in k needs to be solved [29]. Note that only three point matches are needed to solve the motion model if angular rates in at least two of the axes are provided by the on-board IMU.

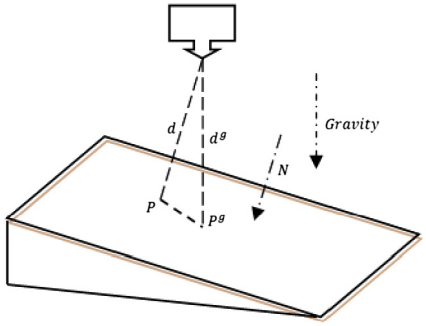

Fig. 4. A downward-looking camera observing a plane.

For our quadrotor platform, the ventral camera is installed on a pan–tilt unit that automatically positions the camera to point in the direction of the gravity vector. Accordingly, the angular rate W_x, W_y are set to be 0 in Eq. (10). In Fig. 4, a line is drawn parallel to the gravity vector from the camera centre and joins the plane at P^g. In the camera coordinate frame, point P^g has a coordinate of $(0, 0, d^g)$. From Eq. (8), we have $d^gN_z = d$. The last column of matrix H_c' is $[N_z\bar{V}_x, N_z\bar{V}_y, N_z\bar{V}_z]^T$, which equals: $[V_x/d^g, V_y/d^g, V_z/d^g]^T$, denoted as $\bar{V}^g = [\bar{V}_x^g, \bar{V}_y^g, \bar{V}_z^g]^T$. Computing unscaled velocity in this way does not require performing SVD after H_c' is obtained, and is termed as **1-SVD**. The method that decomposes H_c' to obtain unscaled linear velocity is called **2-SVD**. In theory, over a ground plane, $\bar{V} = \bar{V}^g$. If one only requires the estimation of horizontal motion, the parameter k also does not need to be solved because it only affects the diagonal entry. When the camera is fixed with respect to the vehicle body, the last column of matrix H_c' normalizes the velocity V by the distance from the camera optic centre to the intersection between the plane and the camera principal axis. This could be an advantage over using $\bar{V}$ if an altimeter such as a sonar sensor is installed at the bottom of the vehicle to provide scale. To compute $\bar{V}^g$ on platforms without a pan–tilt unit, one may virtually rotate the camera parallel to the ground plane using attitude estimation provided by the on-board IMU [33].

The accuracy of IMU attitude estimation has an effect on our proposed method. Suppose the camera has a pitch angle of θ ($N_z = \cos(\theta)$, roll angle is 0) relative to a plane and the IMU has an error of $\Delta\theta$, then the relative error for speed estimation is $\cos(\theta + \Delta\theta) - \cos(\theta)$. With the same $\Delta\theta$, the relative error increases with θ, which is around 4.5% when $\Delta\theta = 3°,\ \theta = 60°$ and 0.1% when $\Delta\theta = 3°,\ \theta = 0°$. For our platform, the accuracy of the horizontal speed estimation is also affected by the error (ΔW_x, ΔW_y) of the pan–tilt unit in cancelling the pitching and rolling

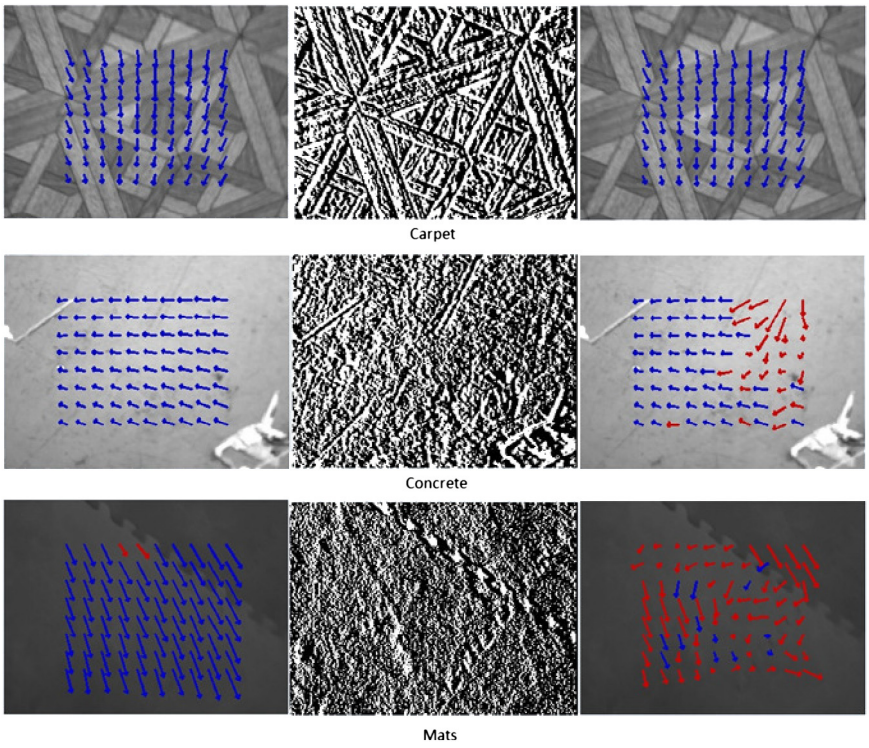

Fig. 5. The computed OF on the 'Carpet', 'Concrete' and 'Mats' datasets, with one example given for each dataset. From left to right, the first column shows the flow field from the BT based LK, the second row gives the transformed binary image using Eq. (3), and the third row is the OF from the intensity based LK. For the outlier detection using RANSAC, 10 iterations are used for the BT based approach and 30 iterations are used for intensity based method. Red arrows represent the detected outliers and the inliers are shown in blue arrows. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

motion. From Eq. (11), the estimated speed becomes $h'_{c3} + \Delta W_y$ and $h'_{c6} + \Delta W_x$.

5.2. Outlier rejection

In this paper, OF is calculated for predefined and equally spaced image points rather than the feature points detected by some feature detectors, which take extra processing time and there could be few features in low-texture environments. The RANSAC algorithm is used to reject outliers in the flow field. A few subsets of 3 features (not co-linear) are randomly chosen from all the tracked features to compute h_c. Then all the features in the image are examined to compute the reprojection error: $|A_{2i}^T h_c - b_{2i}| + |A_{2i+1}^T h_c - b_{2i+1}|$, where i is the feature index. If the error is less than a threshold, the feature is counted as an inlier. In fact, the computation of h_c is not necessary at this point. As known from the previous discussion for the computation of h_c, we have $|A_{2i}^T Se - b_{2i}| + |A_{2i+1}^T Se - b_{2i+1}| = |A_{2i}^T h_c - b_{2i}| + |A_{2i+1}^T h_c - b_{2i+1}|$. That means only Se needs to be solved for the detection of outliers. The subset that has the maximum number of inliers is picked, and if number of inliers is greater than a threshold, the homography matrix is computed by stacking Eq. (10) for all the inliers. Otherwise, the current estimation is equal to the estimation in the previous frame.

5.3. Evaluation on real sensory data

Some images were logged from the ventral camera of our Pelican quadrotor by manually moving the vehicle within the Vicon area to test the performance of our algorithm in fast and jerky motion. The vehicle was lifted above the ground, freely moved in the three axes and at last landed on the ground. The pitch angle and roll angle of the quadrotor was up to 20° during the recording. Images were captured at 20 Hz and IMU data was collected at 100 Hz. Three datasets were recorded by moving the quadrotor for around 90 s (around 1800 images) over three different kinds of textures, as seen in Fig. 5. They are called the 'Carpet', 'Concrete' and 'Mats' datasets hereafter.

A total of 80 OF vectors (8×10, Fig. 5) are calculated using LK. The RANSAC algorithm is employed to detect outliers in the flow field. Let us denote the total number of recorded images as n_t and the number of frames where the detected inliers are more than m as n_m. In Fig. 6, 'Carpet_R60' refers to the ratio of n_{60} to n_t on the 'Carpet' dataset. The ratio of detected inliers using the BT based LK algorithm remains high and stable over the three datasets while the intensity image gives much lower inlier ratios over more poorly-textured scenes. Although the outlier ratio drops as the number of iterations for RANSAC increases, the improvement is small. Still more iterations translate to a higher probability of finding the correct solution. Using 50 iterations for RANSAC, the BT based LK gives an average of 88% inliers on the poorly-textured 'Mats' dataset. From Eq. (1), this inlier ratio only requires 5 iterations for robust estimation using RANSAC. On the 'Mats' dataset, the average inlier ratio for the intensity based LK is 56%, which means 25 iterations are needed. For robustness, 10 iterations are used for the BT based approach and 30 iterations are used for intensity based method in the following experiments. Fig. 5 shows some examples of the computed flow field with the detected outliers (red arrows) on the three datasets. It is seen that BT based LK behaves consistently well over different textures while the intensity based LK relies on a well-textured scene for a good performance. When the outlier ratio is high, the RANSAC algorithm is not able to guarantee a correct solution. It is counter-intuitive that BT would be a better input than intensity for LK. Over low-textured scenes, the signal-to-noise ratio (SNR) of the intensity image is low, which could affect the OF computation. The BT is robust against image noise and illumination change, as proven in [14,41], which explains its

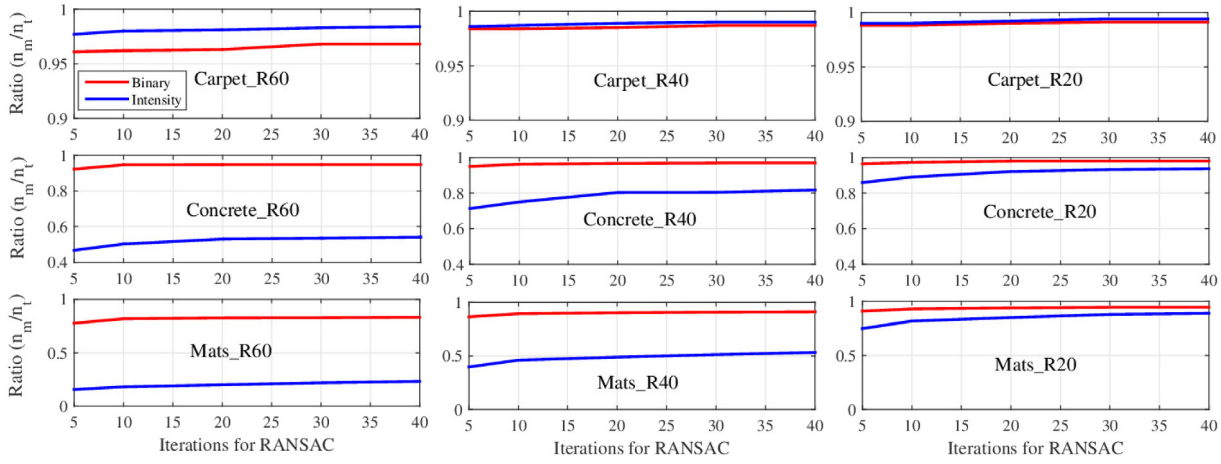

Fig. 6. The ratio of frames over all the images, where the number of detected inliers is over a certain threshold.

improved performance especially in low-textured environments. A video showing the computed OF on the three datasets can be accessed at https://www.youtube.com/watch?v=ugKrCgtZ2q0. In the video, we also show OF calculation on a dataset recorded in an office environment where apparent illumination change occurs. When the image quality becomes very poor (e.g. severe saturation), the BT based LK fails too. Under these circumstances, any visual algorithms may hardly work and other sources of information (in this case, the IMU data) fills in the gap.

The vision-estimated speed is scaled by the height from the Vicon system and then compared quantitatively with the speed from the Vicon using the absolute error (AE):

$$errv_x = |V_x - V_{xvcn}| \tag{12}$$

where V_{xvcn} is the speed from the Vicon system in the X direction. Errors in the other two axes are defined in the same way. Fig. 7 shows the normalized histogram of the speed estimation error on the three datasets using the combinations of BT, intensity image, **1-SVD** and **2-SVD** methods, where 'P01' refers to the percentage of estimations whose absolute error is less than 0.1 m/s and $errv_{xy} = (errv_x + errv_y)/2$. Note that we only calculate the error when the number of inliers is more than 10. On the well-textured 'Carpet' dataset, the BT and intensity based methods give almost the same accuracy. The BT based approach performs consistently well on the other two datasets while the accuracy of the intensity based method drops considerably. The **1-SVD** method is also superior over the **2-SVD** method because it avoids SVD after the homography matrix is obtained. The estimation error in the Z axis changes more substantially than the errors in the other two axes when different methods are used. With a downward-looking camera, vertical motion is inferred from the image expansion and contraction (loom), which is the differentiation of the flow field. The horizontal motion may be regarded as the summation of the flow vectors. Therefore, the estimation of the vertical motion is more affected by noise, either in OF computation or the SVD.

It is noted that when the 'BT + 1SVD' method is used or when the scene is well-textured, estimation in the Z axis is more accurate than the average results of the horizontal directions. The reason may be that the pan–tilt unit does not cancel out the pitching or rolling motion completely, which has more influence on the estimation of horizontal motion. It is also observed from Fig. 8 that the estimation of V_x and V_y is noisier than that of V_z. Fig. 9 shows the estimated surface normal from the homography decomposition (**2-SVD**). Because all the datasets were logged over a ground plane, the surface normal should ideally be [0, 0, 1]. In reality, the estimation accuracy depends on the motion and the quality of OF. The benefit of the BT over intensity for the LK algorithm is once again noted in Fig. 9. On the 'Mats' dataset, the estimation of N becomes much noisier than that on the other two datasets, which explains for the poor speed estimation using the **2-SVD** method. The estimation of W_z on the three datasets is given in Fig. 10, where a few sharp jumps are noted for the vision based method, and the same phenomenon is observed in Fig. 8 for speed estimation. This means that 10 inliers is not sufficient and we choose to update the visual estimation when the inlier ratio is more than 0.5. Although some correct solutions may also be rejected, the majority (more than 90% on the 'Mats' dataset) of the estimations are kept thanks to the robust BT based LK algorithm.

6. Sensor fusion

Acceleration measurements can be transformed into O^g thanks to the on-board attitude estimation from the IMU. Consider the following system:

$$\begin{cases} \dot{d}^g = V_z^g \\ \dot{V}_z^g = A_z^g + B_z^g \\ \dot{B}_z^g = n_{bz} \end{cases} \tag{13}$$

where d^g, V_z^g, A_z^g, B_z^g stand for position, velocity, acceleration and its bias in the Z direction of O^g, n_{bz} is the system noise. We define $[d^g, V_z^g, B_z^g]^T$ as the system states, A_z^g as the system input. Then $\bar{V}_z^g = V_z^g/d^g$ is treated as the system measurement. Although the process model in Eq. (13) is linear, the measurement model $\bar{V}_z^g = V_z^g/d^g$ is nonlinear. All the system states can be estimated using an EKF, following a standard prediction and correction routine and the details are omitted here. The state transition matrix (F_z) and the observation matrix (H_z) are:

$$F_z = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}, \qquad H_z = \begin{bmatrix} \frac{-V_z^g}{d^{g2}} & \frac{1}{d^g} & 0 \end{bmatrix}$$

where Δt is the time step for the IMU data which is 0.01 s in our case. A synthetic data is generated in MATLAB to evaluate the performance of the EKF, where $d^g = 2 + AM\sin(\pi t)$ and AM is the amplitude of the sine wave. V_z^g, A_z^g can be obtained from the differentiation of d^g. Random noise is then added to A_z^g and $\bar{V}_z^g$ using the MATLAB function *awgn*. The simulated acceleration measurement A_z^g is sampled at 100 Hz and the $\bar{V}_z^g$ is sampled at 20 Hz. The SNR ranges from 15 to 60 in the simulation. For each SNR value, 20 tests are performed and each test lasts 80 s. Initial states for the filter are set at [2, 0, 0]. Mean Absolute Error (MAE) is employed to quantify the accuracy of the filter for height estimation. Because the filter requires some time to converge, we

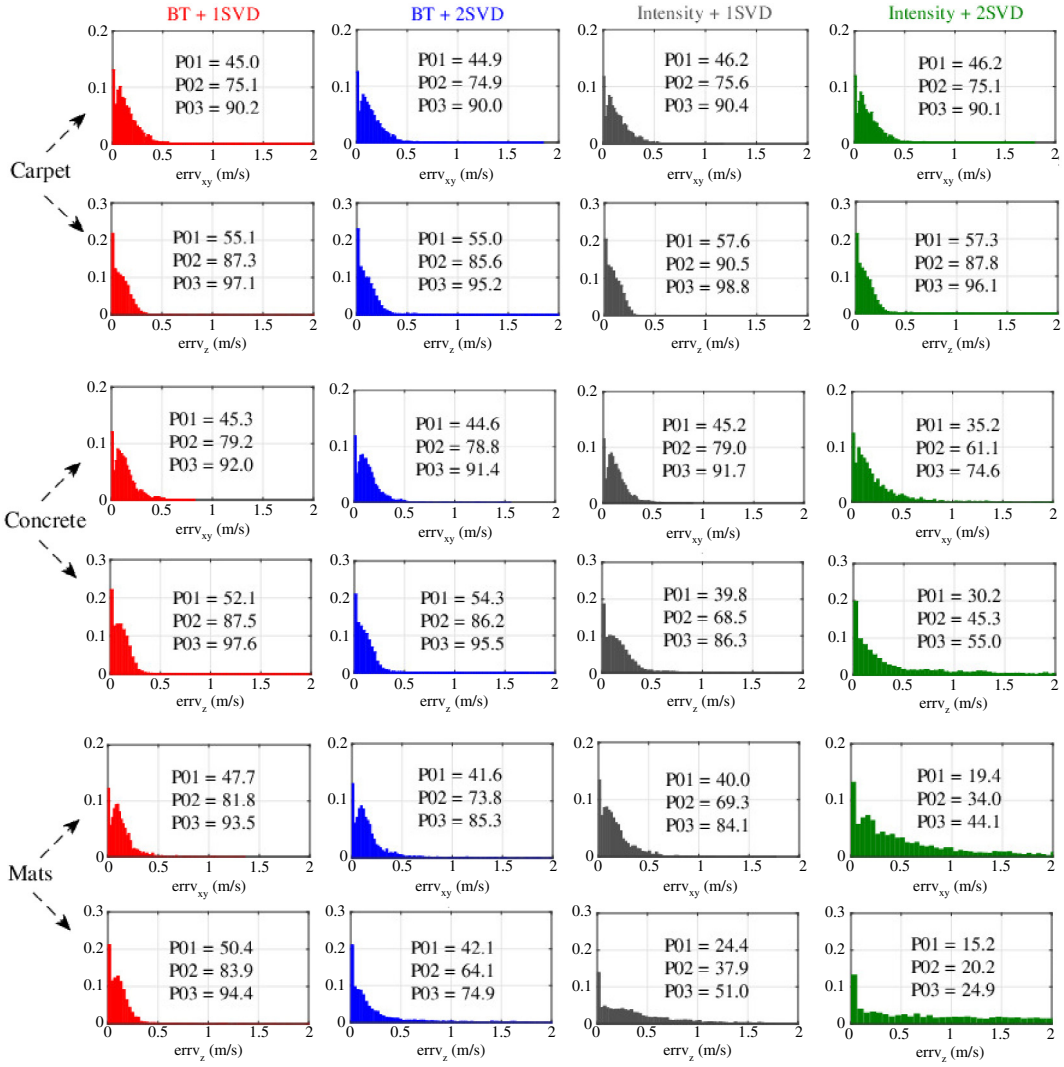

Fig. 7. The normalized histogram of the speed estimation error on the 'Carpet', 'Concrete' and 'Mats' datasets using the combinations of BT, intensity image, **1-SVD** and **2-SVD** methods. 'P01' means the percentage of estimations whose absolute error is less than 0.1 m/s.

only calculate the error for the last 70 s. Fig. 11 shows the accuracy of EKF with different noise levels and motion amplitudes (*AM*). It is observed that the error drops with increasing SNR and *AM* values. When the SNR is large enough, the influence of *AM* is almost negligible. Having low SNR values, a larger motion is required for more reliable estimation. The convergence rate (transient property) of the filter is also evaluated. Setting $SNR = 30$, the initial value for height $d^g(0)$ and *AM* are varied. Still, 20 experiments are conducted for each setting. Fig. 12 shows that the convergence is faster with smaller initial errors and larger motion amplitudes.

The estimated d^g is used to scale the visual estimation $(\bar{V}_x^g, \bar{V}_y^g)$ to obtain the raw speed in the horizontal directions, which is then fused with the acceleration measurements A_x^g, A_y^g based on the following system model using a Kalman Filter:

$$\begin{cases} \dot{V}_x^g = A_x^g + B_x^g \\ \dot{B}_x^g = n_{bx} \end{cases} \tag{14}$$

where the state transition matrix (F_x) and the observation matrix (H_x) are:

$$F_x = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \qquad H_x = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

where both the process and measurement models are linear. The proposed sensor fusion framework treats the three axes separately and is easy to implement. Parameter tuning for the filter is simpler and smaller covariance matrices need to be updated at the sampling frequency of IMU data. Note that the estimated d^g is relative to the observed terrain, and in theory, it requires persistent vertical motion in order for all the system states to be observable. The VIF method is compared with a purely vision based method for height estimation. For that purpose, the height is recursively estimated by vision:

$$\begin{cases} V_{zvis}^- = \bar{V}_z^{g-} \cdot d_{zvis}^- \\ d_{zvis}^+ = d_{zvis}^- + \bar{v}_{zvis}^- \cdot \Delta t_{vis} \end{cases} \tag{15}$$

where Δt_{vis} is the time interval between the capture of two images; the superscripts $+, -$ refer to the previous and current estimation; d_{zvis}, V_{zvis} are the height and speed estimation using the purely vision based method.

For our method, only the relative yaw angle between the IMU and camera needs to be known. After the acceleration measurements are transformed to O^g, their directions are parallel to the camera axes if the relative yaw angle is known and corrected. The relative translation between the two sensors is not important for

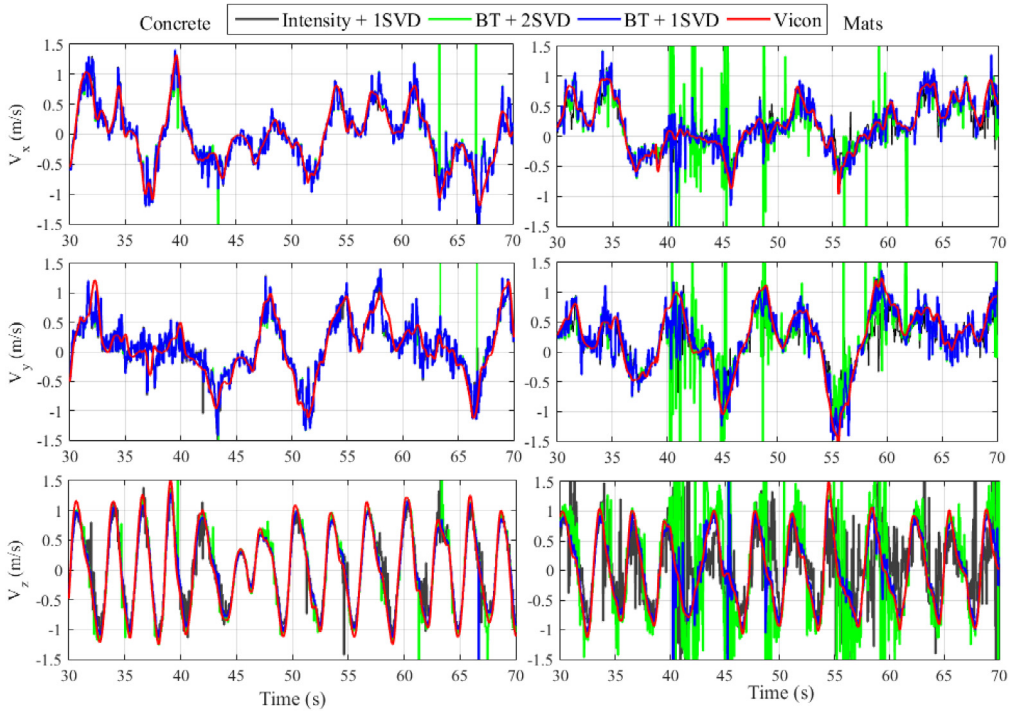

Fig. 8. The raw speed estimation on the 'Concrete' and 'Mats' datasets. The scale is provided by the Vicon system.

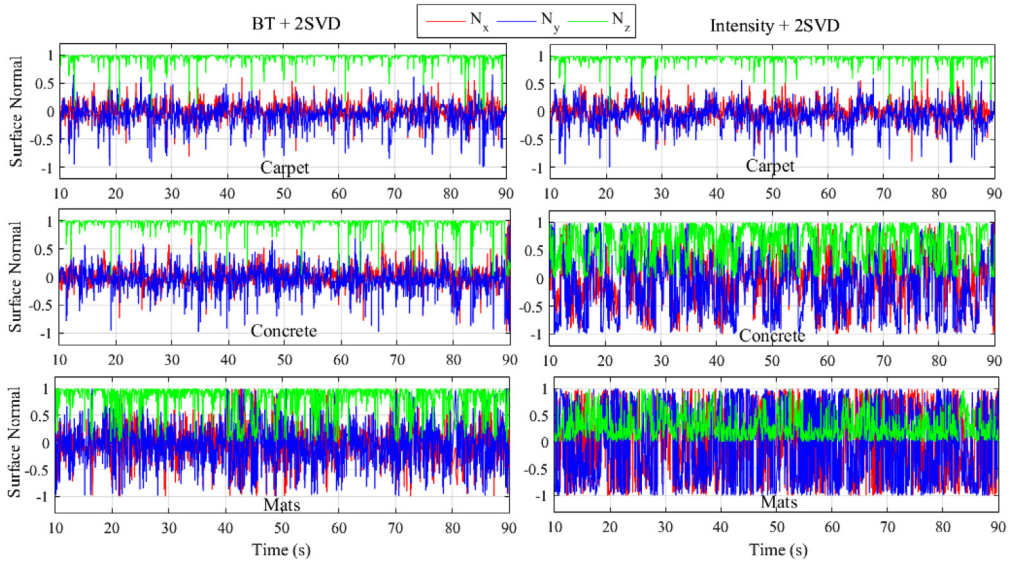

Fig. 9. From top to bottom, the three rows show the estimated surface normal from homography decomposition on the three datasets.

our approach. For general IMU-camera calibration, please refer to the toolbox.[1]

7. Experimentation

7.1. Recorded datasets

On the three recorded datasets, in this section, the scale is provided by the BT based VIF approach. Initial values for the EKF states are: $d^g = 0.3$ m, $V_z^g = 0.0$ m/s, $B_z^g = 0.0$ m/s^2. The accuracy of the state estimation is evaluated for different number of OF vectors, namely 8×10, 6×8, 5×7, 4×6. Note that for each setting, the features are distributed within the same region of an image. That means the spacing between features is larger for a smaller number of tracked features. Raw visual estimation is only updated when the inlier ratio is no less than 0.5. Table 2 gives the mean absolute error for horizontal speed ($errv_{xy}$ in m/s) and height estimation ($errd$ in m) using different number of features. It is found that the number of tracked features has little effect on the accuracy of the speed estimation. The accuracy of height estimation drops evidently using 4×6 features. For the following experiments, OF for 5×7 features is computed. Fig. 13 shows the estimated height and speed from Vicon system and the VIF method. State estimation from the VIF method corresponds well with the estimation of the Vicon system. Height estimation from the purely

[1] https://github.com/ethz-asl/kalibr/wiki/camera-imu-calibration.

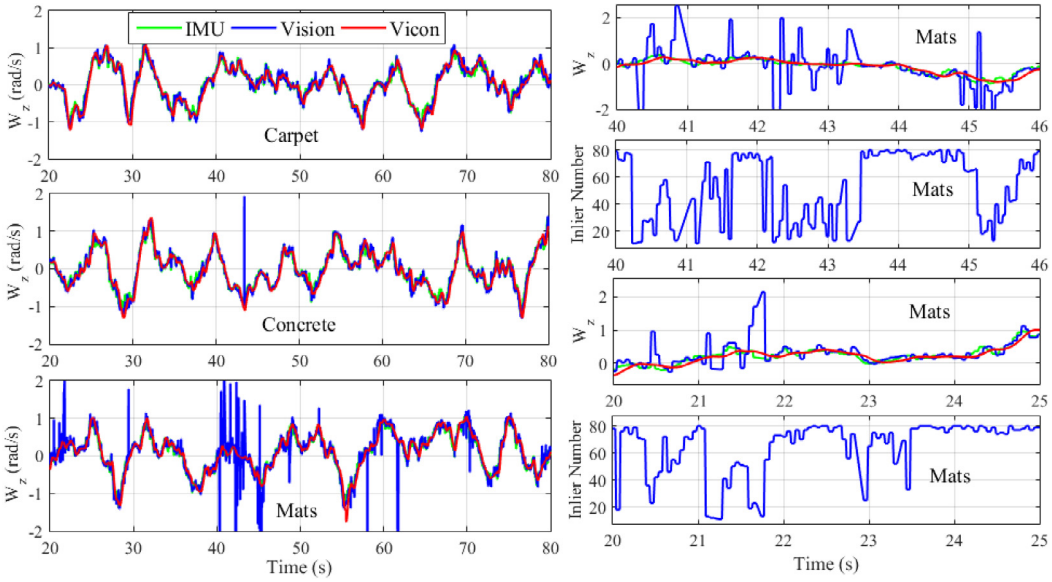

Fig. 10. The left column shows the estimation of W_z using the BT based method as compared to that of IMU and Vicon, and the right column is the zoomed-in plots for the number of detected inliers and the estimation of W_z on the 'Mats' dataset.

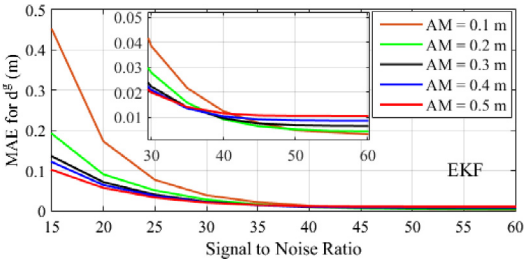

Fig. 11. The accuracy of EKF for height estimation under different noise levels and motion amplitudes.

vision based approach (Eq. (15)) is drifting and not consistent, where the initial value for d_{zvis} is 0.3 m.

7.2. Flight tests

A PID controller is used to control the height and the horizontal speed based on our VIF method. The vision-estimated speed is integrated for horizontal position estimation, which is fed into the horizontal speed controller in hover mode. Fig. 14 shows the height and horizontal position estimation from our VIF method and the Vicon system in one of the flight tests. The vehicle takes off, follows a sine trajectory vertically and then hovers for around 60 s from the moment of 30 s. During hover, the VIF height estimation drifts slowly due to a lack of vertical excitation, a phenomenon also observed in [53]. The drift is corrected quickly after a sine motion. A zero setpoint for horizontal position means that the vehicle is in non-hover mode, where only the speed is regulated. For landing, a setpoint of 0.2 m is given for height, and when the altitude

Table 2
The accuracy of the VIF method for speed and height estimation.

Features	Carpet		Concrete		Mats	
	errd	$errv_{xy}$	*errd*	$errv_{xy}$	*errd*	$errv_{xy}$
8 × 10	0.026	0.107	0.035	0.108	0.058	0.104
6 × 8	0.024	0.107	0.036	0.109	0.058	0.105
5 × 7	0.026	0.106	0.034	0.109	0.059	0.105
4 × 6	0.029	0.107	0.042	0.109	0.073	0.104

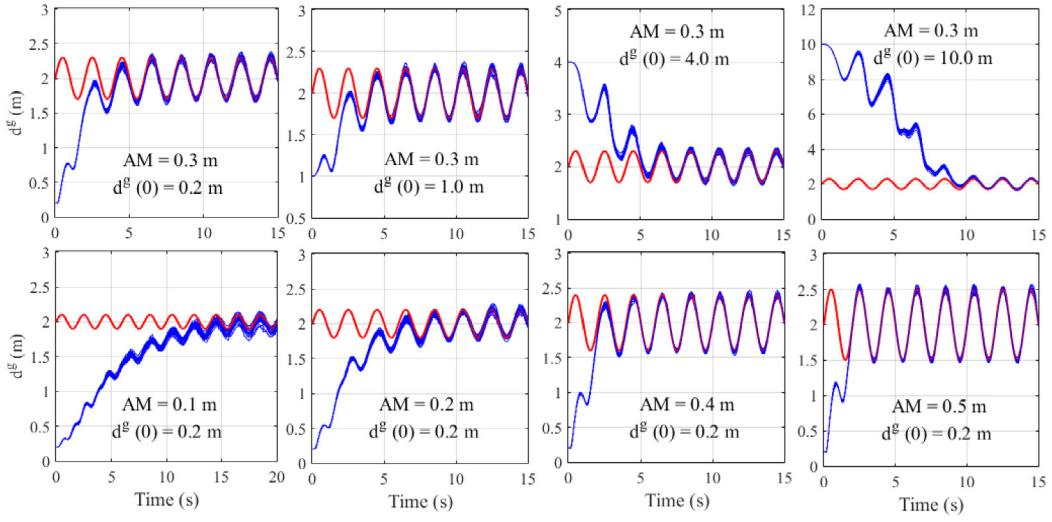

Fig. 12. The convergence rate of EKF with different initial states and motion amplitudes.

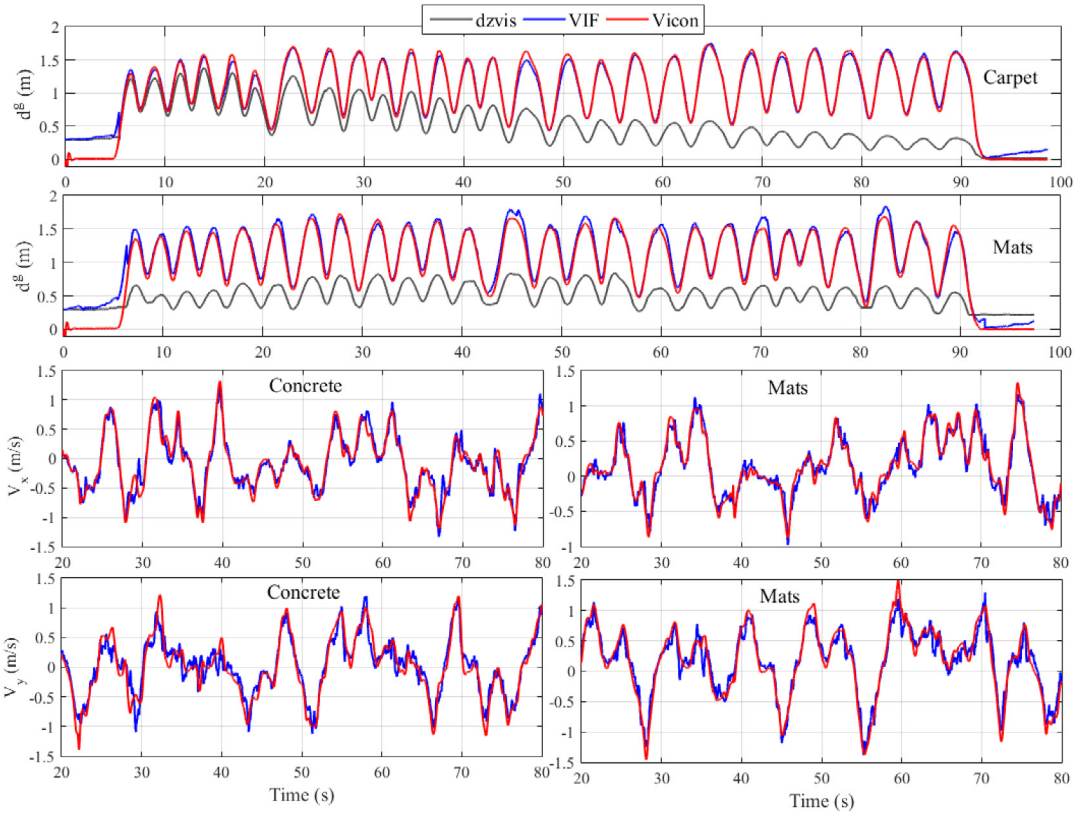

Fig. 13. Height and speed estimation from the Vicon system and the VIF method on the recorded datasets. Also shown is the height estimation (d_{zvis}) using the purely vision based method (Eq. (15)).

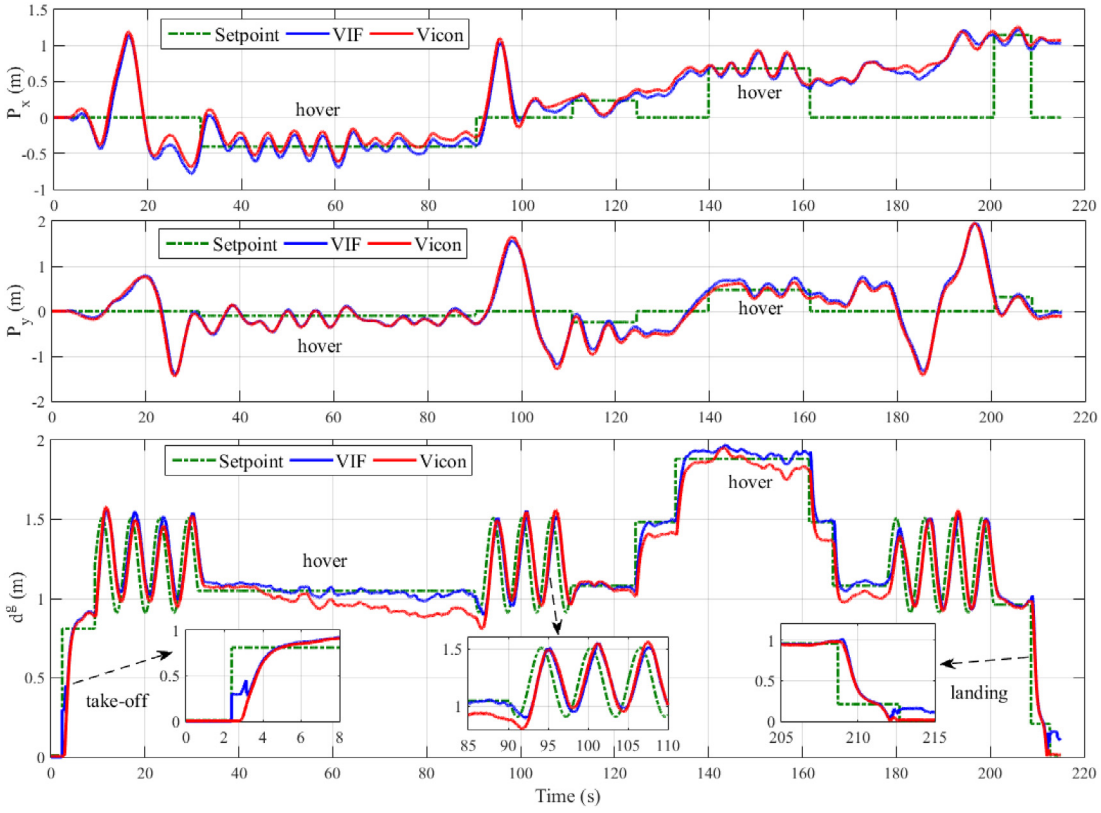

Fig. 14. The height and horizontal position estimation in one of the flight tests.

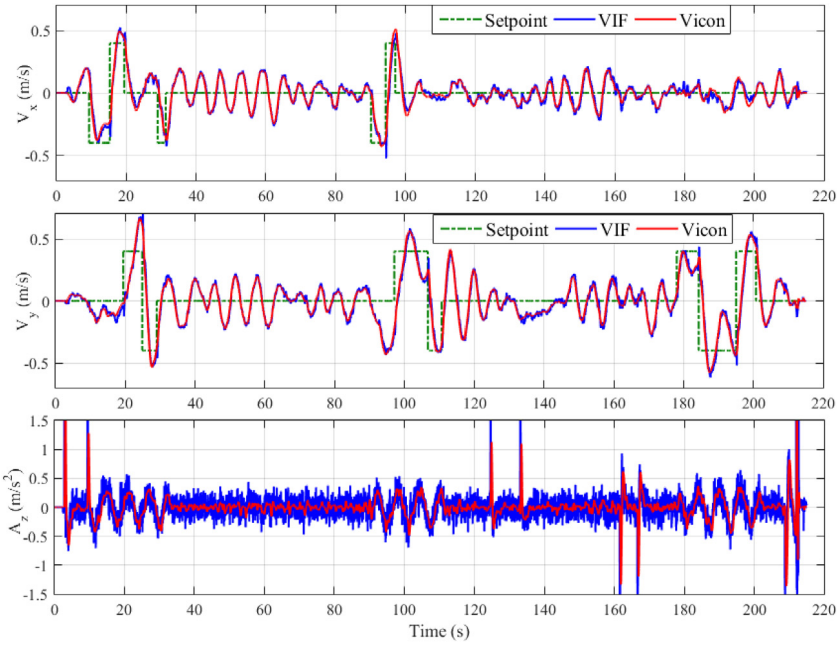

Fig. 15. The estimation of horizontal speed and acceleration in the vertical axis in one of the flight tests.

drops below 0.3 m, the thrust is reduced linearly to 0 to achieve a soft touchdown. The speed and acceleration measurements are shown in Fig. 15. The height and speed command is only roughly tracked in our experiments because we did not try to optimize the controller parameters. A few sharp jumps are noted in the vertical acceleration measurement due to ascending or descending of the vehicle when a new height setpoint is given. The mean absolute error for the height and position in the X and Y directions are respectively 0.057 m, 0.050 m, 0.052 m. The error for the speed estimation is 0.018 m/s, 0.020 m/s and 0.018 m/s in the three axes, which is much smaller than that on the recorded datasets because of smaller and smoother motion in the flight test. To contain the positional drift in hover, the snapshot idea may be implemented [13,14], where a snapshot image is captured and acts as the absolute reference for the following images.

8. Conclusion

A robust and lightweight VIF algorithm has been proposed in this paper for the state estimation and control of a RMAV. Compared with intensity images, the simple binary transformation produces a higher quality optic flow field using the LK algorithm over different textures. For outlier detection using the RANSAC algorithm, the binary image based ME requires fewer iterations, yet performs better than the intensity based approach with more iterations. This may indicate that it is more important to improve the OF estimation than using more iterations for RANSAC. Other ME algorithms may also benefit from the binary image based method. The unscaled translation is parametrized in a different way to avoid SVD after the homography matrix is calculated, improving the accuracy of ME over the traditional approach. The VIF scheme has been proven in closed-loop flight tests to be suitable for general control of a RMAV. The SLAM algorithms give global yaw angle and position estimation and is more accurate than OF based approaches; however, it is fragile in low-textured environments. Our method may be combined with the SLAM algorithms to build a more robust and accurate system than using either method alone.

For future work, a more advanced controller structure will be designed to enhance the control performance. We are also considering fitting a forward-looking camera to the quadrotor to enable obstacle avoidance in forward flight using optic flow. Now, purely forward motion would induce small OF for objects along the principal axis of the front camera and thus has a difficulty sensing such obstacles. Hence, we imagine a situation where the vehicle is performing sinusoidal motion vertically while moving forward. The vertical motion contributes to a fully observable VIF system, and helps the forward-looking camera with the detection of impending obstacles. Interestingly, locusts are believed to rely on 'peering' (side-to-side translational head motion) to infer the distance to target [59], which is similar to the role of the vertical motion in our approach.

Acknowledgement

We would like to thank the Australian Defence Science and Technology Organisation for provision of the Vicon Motion Tracking System which has supported this work (Grant Number: 2013/1169699/1).

References

[1] A. Ortiz, F. Bonnin-Pascual, E. Garcia-Fidalgol, Vessel inspection: A micro-aerial vehicle-based approach, J. Intell. Robot. Syst. 76 (1) (2014) 151–167.
[2] F. Kendoul, Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems, J. Field Robot. 29 (2) (2012) 315–378.
[3] S. Weiss, M.W. Achtelik, S. Lynen, M.C. Achtelik, L. Kneip, M. Chli, R. Siegwart, Monocular vision for long-term micro aerial vehicle state estimation: A compendium, J. Field Robot. 30 (5) (2013) 803–831.
[4] M.N. Haque, M. Biswas, M.R. Pickering, M.R. Frater, A low-complexity image registration algorithm for global motion estimation, IEEE Trans. Circuits Syst. Video Technol. 22 (3) (2012) 426–433.
[5] C. Forster, M. Pizzoli, D. Scaramuzza, SVO: Fast semi-direct monocular visual odometry, in: IEEE International Conference on Robotics and Automation, Hong Kong, China, 2014.
[6] M. Srinivasan, M. Lehrer, W. Kirchner, S.W. Zhang, Range perception through apparent image speed in freely flying honeybees, Visual Neurosci. 6 (5) (1991) 519–535.
[7] J. Serres, D. Dray, F. Ruffier, N. Franceschini, A vision-based autopilot for a miniature air vehicle: Joint speed control and lateral obstacle avoidance, Auton. Robots 25 (1) (2008) 103–122.
[8] A. Vardy, R. Moller, Biologically plausible visual homing methods based on optical flow techniques, Connect. Sci. 17 (1) (2005) 47–89.
[9] J. Zufferey, D. Floreano, Fly-inspired visual steering of an ultralight indoor aircraft, IEEE Trans. Robot. 22 (1) (2006) 137–146.

[10] M. Garratt, J. Chahl, An optic flow damped hover controller for an autonomous helicopter, in: International UAV Systems Conference, Bristol, 2007.
[11] B. Hérissé, F. Russotto, T. Hamel, R. Mahony, Hovering flight and vertical landing control of a VTOL unmanned aerial vehicle using optical flow, in: IEEE/RJS International Conference on Intelligent Robot Systems, Nice, France, 2008.
[12] F. Kendoula, I. Fantoni, K. Nonamib, Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles, Robot. Auton. Syst. 57 (6–7) (2009) 591–602.
[13] M. Garratt, A. Lambert, H. Teimoori, Design of a 3D snapshot based visual flight control system using a single camera in hover, Auton. Robots 34 (1) (2012) 19–34.
[14] Ping Li, M. Garratt, A. Lambert, Monocular snapshot-based sensing and control of hover, takeoff, and landing for a low-cost quadrotor, J. Field Robot. 32 (7) (2015) 984–1003.
[15] S. Yang, S.A. Scherer, A. Zell, An onboard monocular vision system for autonomous takeoff, hovering and landing of a micro aerial vehicle, J. Intell. Robot. Syst. 69 (1) (2013) 499–515.
[16] A.J. Davison, I.D. Reid, N.D. Molton, O. Stasse, MonoSLAM: Real-time single camera SLAM, IEEE Trans. Pattern Anal. Mach. Intell. 29 (6) (2007) 1052–1067.
[17] G. Klein, D.W. Murray, Parallel tracking and mapping for small AR workspaces, in: International Symposium on Mixed and Augmented Reality, ISMAR, 2007, pp. 225–234.
[18] G. Chowdhary, E.N. Johnson, D. Magree, A. Wu, GPS-denied indoor and outdoor monocular vision aided navigation and control of unmanned aircraft, J. Field Robot. 30 (3) (2013) 415–438.
[19] J. Engel, J. Sturm, D. Cremers, Scale-aware navigation of a low-cost quadrocopter with a monocular camera, Robot. Auton. Syst. 62 (11) (2014) 1646–1656.
[20] D. Ball, S. Heath, J. Wiles, G. Wyeth, P. Corke, M. Milford, OpenRatSLAM: an open source brain-based SLAM system, Auton. Robots 34 (3) (2013) 149–176.
[21] S. Weiss, M.W. Achtelik, S. Lynen, M. Chli, R. Siegwart, Real-time onboard visual inertial state estimation and self-calibration of MAVs in unknown environments, in: IEEE International Conference on Robotics and Automation, St Paul, USA, 2012.
[22] D.S. Kumar, C.V. Jawahar, Robust homography-based control for camera positioning in piecewise planar environments, in: Indian Conference on Computer Vision, Graphics and Image Processing, 2006.
[23] E. Malis, F. Chaumette, 2 1/2 D visual servoing with respect to unknown objects through a new estimation scheme of camera displacement, Int. J. Comput. Vis. 37 (1) (2000) 79–97.
[24] Y. Ma, S. Soatto, J. Košecká, S.S. Sastry, An Invitation to 3-D Vision, in: Interdisciplinary Applied Mathematics, vol. 26, Springer, 2004.
[25] P. Corke, J. Lobo, J. Dias, An introduction to inertial and visual sensing, Int. J. Robot. Res. 26 (6) (2007) 519–535.
[26] P.J. Bristeau, F. Callou, D. Vissière, N. Petit, The navigation and control technology inside the AR. Drone micro-UAV, in: IFAC World Congress, Milano, Italy, vol. 18, no. 1, 2011.
[27] P. Corke, An inertial and visual sensing system for a small autonomous helicopter, J. Robot. Syst. 21 (2) (2004) 43–51.
[28] J. Chahl, K. Rosser, A. Mizutani, Axially displaced optical flow sensors for feasuring and controlling the landing height of a UAV, in: Australian International Aerospace Congress, Melbourne, 2013, pp. 221–227.
[29] P. Li, M. Garratt, A. Lambert, A homography-based visual inertial fusion method for robust sensing of a micro aerial vehicle, in: IEEE International Conference on Mechatronics and Automation, Beijing, China, 2015.
[30] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (6) (1981) 381–395.
[31] D. Scaramuzza, F. Fraundorfer, R. Siegwart, Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC, in: IEEE International Conference on Robotics and Automation, Kobe, Japan, 2009.
[32] G.H. Lee, F. Fraundorfer, M. Pollefeys, Motion estimation for self-driving cars with a generalized camera, in: IEEE International Conference on Computer Vision and Pattern Recognition, 2013.
[33] C. Troiani, A. Martinelli, C. Laugier, D. Scaramuzza, Low computational-complexity algorithms for vision-aided inertial navigation of micro aerial vehicles, Robot. Auton. Syst. 69 (2014) 80–97.
[34] B. Li, L. Heng, G.H. Lee, M. Pollefeys, A 4-point algorithm for relative pose estimation of a calibrated camera with a known relative rotation angle, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 2013.
[35] F. Fraundorfer, O. Saurer, M. Pollefeys, Homography based visual odometry with known vertical direction and weak manhattan world assumption, in: IEEE/IROS Workshop on Visual Control of Mobile Robots, ViCoMoR, 2012.
[36] B.D. Lucas, T. Kanadei, An Iterative image registration technique with an application to stereo vision, in: DARPA Image Understanding Workshop, 1981, pp. 121–130.
[37] P. Anandan, A computational framework and an algorithm for the measurement of visual motion, Int. J. Comput. Vis. (1989) 283–310.
[38] A. Bruhn, J. Weickert, C. Schnörr, Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods, Int. J. Comput. Vis. 61 (3) (2005) 211–231.
[39] J.L. Barron, D.J. Fleet, S.S. Beauchemin, Performance of optical-flow techniques, Int. J. Comput. Vis. 12 (1) (1994) 43–77.
[40] Y.H. Kim, A.M. Martinez, A.C. Kak, Robust motion estimation under varying illumination, J. Image Vis. Comput. 23 (4) (2004) 365–375.
[41] S. Kaneko, I. Muraseb, S. Igarashia, Robust image registration by increment sign correlation, Pattern Recognit. 35 (2002) 2223–2234.
[42] P. Boonsieng, T. Kondo, W. Kongprawechnon, Unit gradient vectors based motion estimation Techniques, ECTI Trans. Electr. Eng. Electron. Commun. 9 (2) (2011).
[43] M. Kharbat, N. Aouf, A. Tsourdos, B.A. White, Robust brightness description for computing optical flow, in: British Machine Vision Conference, 2008.
[44] J. Kelly, G.S. Sukhatme, Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration, Int. J. Robot. Res. 30 (1) (2011) 56–79.
[45] S. Weiss, R. Brockers, L. Matthies, 4DoF drift free navigation using inertial cues and optical flow, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 2013.
[46] S. Weiss, R. Brockers, S. Albrektsen, L. Matthies, Inertial optical flow for throw-and-go micro air vehicles, in: IEEE Winter Conference on Applications of Computer Vision, 2015.
[47] D. Abeywardena, Z. Wang, S. Kodagoda, G. Dissanayake, Visual-inertial fusion for quadrotor micro air vehicles with improved scale observability, in: IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 2013.
[48] D. Abeywardena, G. Dissanayake, Tightly-coupled model aided visual-inertial fusion for quadrotor micro air vehicles, Field Serv. Robot. 105 (2013) 153–166.
[49] S. Shen, N. Michael, V. Kumar, Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft MAVs, in: IEEE International Conference on Robotics and Automation, Washington, USA, 2015.
[50] S. Zhao, F. Liny, K. Pengy, B.M. Chen, T.H. Lee, Homography-based vision-aided inertial navigation of UAVs in unknown environments, in: AIAA Guidance, Navigation, and Control Conference, 2012.
[51] V. Grabe, H.H. Bülthoff, P.R. Giordano, A comparison of scale estimation schemes for a quadrotor UAV based on optical flow and IMU measurements, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 2013.
[52] S. Wang, L. Chen, D. Gu, H. Hu, Vision-aided inertial navigation using three-view geometry, in: World Congress on Intelligent Control and Automation, Shenyang, China, 2014.
[53] S. Omari, G. Ducard, Metric visual-inertial navigation system using single optical flow feature, in: European Control Conference, Zürich, Switzerland, 2013.
[54] D. Tick, A.C. Satici, J. Shen, N. Gans, Tracking control of mobile robots localized via chained fusion of discrete and continuous epipolar geometry, IMU and odometry, IEEE Trans. Cybern. 43 (4) (2013) 1237–1250.
[55] M.Y. Li, A.I. Mourikis, High-precision, consistent EKF-based visual-inertial odometry, Int. J. Robot. Res. 32 (6) (2013) 690–711.
[56] M.Y. Li, A.I. Mourikis, 3-D motion estimation and online temporal calibration for camera-IMU systems, in: IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 2013.
[57] M. Quigley, K. Conley, B.P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A.Y. Ng, ROS: an open-source robot operating system, in: ICRA Workshop on Open Source Software, 2009.
[58] S. Baker, D. Scharstein, J. Lewis, A database and evaluation methodology for optical flow, Int. J. Comput. Vis. 92 (1) (2011) 1–31.
[59] E.C. Sobel, The locust's use of motion parallax to measure distance, J. Comp. Physiol. A 167 (5) (1990) 579–588.

Ping Li received his bachelor degree in Mechanical Engineering from the Nanjing University of Science and Technology in 2009 and his master degree in Control Theory and the Application from the same university in 2012. He is now a Ph.D. student at the University of New South Wales, working on sensing and control of Micro Aerial Vehicles using a monocular camera and an Inertial Measurement Unit.

Matthew Garratt worked for over a decade as an aeronautical engineer in the Australian Defence Force. In 1999, he joined the Australian National University (ANU) where he worked on biologically inspired vision and control for an autonomous helicopter. He completed a Ph.D. in Robotics at ANU in 2007. Since 2001, he has been with the University of New South Wales, Canberra, Australia, as a lecturer in the School of Engineering and Information Technology. His main research areas are flight dynamics and sensing and control for autonomous systems. He currently teaches helicopter dynamics and fundamentals of flight.

Andrew Lambert received the B.Sc. degree in Physics (with honors) from the University of Otago, Dunedin, New Zealand, in 1984 and the Ph.D. degree in Electrical Engineering from The University of New South Wales, Canberra, Australia, in 1997. He has been a Member of the lecturing staff with UNSW Canberra since 1988. His research interests include optical and digital image and signal processing, high speed processing hardware and electrooptics for UAV guidance and adaptive optics for imaging through turbulence applications in astronomy, surveillance, and ophthalmology, and holographic video display.

Shanggang Lin received the B.Sc. degree from South China University of Technology in 2009. He is currently a Ph.D. student at the University of New South Wales (UNSW), Canberra, Australia. He is currently working on vision based UAV navigation.