Robotica (2018) volume 36, pp. 1571–1589. © Cambridge University Press 2018
doi:10.1017/S0263574718000577

Vision-only egomotion estimation in 6DOF using a sky compass

Tasarinan Jouir†*, Reuben Strydom†‡,
Thomas M. Stace§ and Mandyam V. Srinivasan†‡

†Queensland Brain Institute, QBI Building 79, University of Queensland, St Lucia, Brisbane, Queensland, Australia, 4072
‡School of Information Technology and Electrical Engineering, General Purpose South Building 78, University of Queensland, St Lucia, Brisbane, Queensland, Australia, 4072
E-mails: reuben.strydom@uqconnect.edu.au, m.srinivasan@uq.edu.au
§School of Mathematics and Physics, Parnell Building 07, University of Queensland, St Lucia, Brisbane, Queensland, Australia, 4072. E-mail: stace@physics.uq.edu.au

(Accepted June 28, 2018. First published online: July 25, 2018)

SUMMARY
A novel pure-vision egomotion estimation algorithm is presented, with extensions to Unmanned Aerial Systems (UAS) navigation through visual odometry. Our proposed method computes egomotion in two stages using panoramic images segmented into sky and ground regions. Rotations (in 3DOF) are estimated by using a customised algorithm to measure the motion of the sky image, which is affected only by the rotation of the aircraft, and not by its translation. The rotation estimate is then used to derotate the optic flow field generated by the ground, from which the translation of the aircraft (in 3DOF) is estimated by another customised, iterative algorithm. Segmentation of the rotation and translation estimations allows for a partial relaxation of the planar ground assumption, inherently increasing the robustness of the approach. The translation vectors are scaled using stereo-based height to compute the current UAS position through path integration for closed-loop navigation. Outdoor field tests of our approach in a small quadrotor UAS suggest that the technique is comparable to the performance of existing state-of-the-art vision-based navigation algorithms, whilst also removing all dependence on additional sensors, such as an IMU or global positioning system (GPS).

KEYWORDS: Navigation; Egomotion; Path Integration; Optic Flow; Sky Compass.

1. Introduction

Unmanned aerial systems (UAS) are being increasingly deployed in applications such as search and rescue, mapping and structural inspection. Such applications fundamentally require precise knowledge and control of the platform's position in space, and therefore an accurate, efficient and robust method for determining the motion of the aircraft in six degrees of freedom (6DOF).

1.1. Conventional methods of measuring egomotion

The conventional method of determining the 6DOF motion of a UAS is to use data from an inertial measurement unit (IMU) in combination with a position estimate derived from global positioning system (GPS) data. An IMU fuses information from a combination of accelerometers, gyroscopes and magnetometers. However, IMUs are prone to error, as both accelerometers and gyroscopes rely on integration over time that leads to noise-induced drift.[1] Additionally, magnetometers are insensitive when the rotational axis of the aircraft is parallel to the direction of the local magnetic field.[2] Ultimately when considering UAS navigation based on IMU data, any sensor drift will result in significant path error. The position estimate of a UAS can be estimated with high accuracy using a GPS, however,

* Corresponding author. E-mail: tas.jouir@uq.net.au

https://doi.org/10.1017/S0263574718000577 Published online by Cambridge University Press

this information is often unreliable at low altitudes or in cluttered environments. The combination of the attitude and position estimates provides the aircraft's current pose in 6DOF.

An alternative, conventional approach for determining the 6DOF motion of a UAS is to compute the optic flow field generated by the aircraft's motion using an onboard vision system. The egomotion is typically estimated by fitting a set of six egomotion parameters (three translational components and three rotational components) to the optical flow field computed between two consecutive camera images, or directly from the image sequence using techniques, such as the Fourier–Mellin transform.[3]

Techniques for computing optic flow have undergone considerable advances in the field of computer vision, with many applications to robotics. A classical approach is the pyramidal Lukas–Kanade algorithm,[4] which computes optic flow using the ratio of the local temporal derivative to the local spatial intensity gradient, and incorporating a pyramidal structure to improve the accuracy and reduce the overall complexity of the computation. Other approaches to computing optic flow include block-matching – which uses the sum of absolute pixel-wise intensity differences (SAD) to obtain the image-coordinate position change of an image patch between consecutive frames, and can also be implemented in a pyramidal structure – and an image interpolation algorithm (I^2A) that creates a set of reference images corresponding to small, known image displacements along each axis, and measures the optic flow between successive images by a process of interpolation.[5]

The measured optic flow field can be used to compute the egomotion of an aircraft in several ways. Grabe, *et al.*[6] compare three methods of computing the egomotion of a small quadrotor at 17 Hz using optic flow from a single downward facing camera. The optic flow for all three approaches is computed using the Lukas–Kanade algorithm. The first algorithm simultaneously estimates the six motion parameters and the orientation of an assumed ground plane, resulting in an 8DOF least-squares search over all tracked points. The second approach simplifies this process by first derotating the optic flow vectors using rotational information from an onboard IMU. This reduces the search to 5DOF, three involving the aircraft's translation and two describing the orientation of the ground plane relative to the aircraft. The third approach builds on this idea by fusing IMU data to obtain the orientation of the ground plane. That study compared the accuracy of each method through a comparison of the aircraft's position as recorded by an external tracking system. The authors concluded that the second method displayed the best performance, as the first approach produced noisy rotation estimates, while the third variant required more restrictive assumptions.

A hybrid stereoscopic vision system is presented by Eynard *et al.*[7] that uses a three-step approach to compute egomotion. Attitude is found from either vanishing points (in urban environments) or the horizon line (in natural environments), while altitude is obtained using a plane sweeping approach that matches the current stereo image of the ground to a reference image (requiring the vector normal to the ground plane – supplied by the onboard IMU). Finally, the translational motion is detected by tracking points using an adaptation of the Lukas–Kanade algorithm. The large scope of pose information available from optic flow is highlighted by Zuffery *et al.*,[8] who describe an optic flow based control strategy for a fixed wing UAS. That study assumes that the optic flow induced by translation is always aligned with the main axis of the aircraft during level flight, to greatly simplify computation. The optic flow signal is derotated using pitch and yaw estimations from MEMS gyros in combination with the aircraft speed obtained from an anemometer, to compute proximity data that can be used for collision avoidance.

A number of the conventional egomotion estimation techniques outlined above as well as those described in Section 1.3 are listed in Table I to illustrate the variety of technologies used by each process. Note that to obtain the full path integrated navigation solution, altitude (or more specifically height above ground level), is a fundamental requirement which is typically obtained using stereo. As such, the use of stereo and monocular visual information is not mutually exclusive.

1.2. Biologically inspired approaches

Insects, despite carrying limited computational capacity, have evolved simple and elegant vision-based methods for sensing their egomotion and guiding navigation.[9,10] It is noteworthy that these strategies are entirely self-contained and not reliant on information provided by external sources, such as GPS. The compound eyes of insects endow them with near-panoramic vision, which many walking insects (such as desert ants) use to sense and use the azimuthal bearing of the sun as a compass to track their heading.[11] In addition, the polarisation sensitivity of insect photoreceptors enables them to sense the

https://doi.org/10.1017/S0263574718000577 Published online by Cambridge University Press

Table I. Comparison of the egomotion estimation methodologies described in Sections 1.1 and 1.3.

	Technique				
Method	Optic fFlow	Stereo	Monocular	IMU	Vision only
Grabe *et al.*[6]	✓	✗	✓	✓	✗
Eynard *et al.*[7]	✓	✓	✓	✓	✗
Zuffery *et al.*[8]	✓	✗	✓	✓	✗
Thurrowgood *et al.*[23]	✓	✓	✓	✓	✗
Strydom *et al.*[24]	✓	✓	✓	✓	✗
Denuelle and Srinivasan[25]	✓	✓	✓	✓	✗
Proposed method	✓	✓	✓	✓	✓

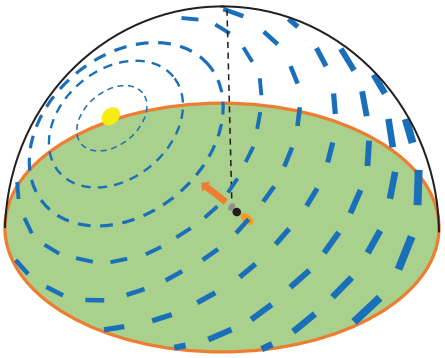


Fig. 1. Illustration of the polarised-light compass. When the sun (yellow circle) is visible in the sky, an insect uses the azimuthal position of the sun as a reference to maintain a constant heading direction, and to measure and control the rate of yaw. In this case, the sun's azimuth is 90° to the left of the insect's flight direction. The sun also creates a pattern of polarised light in the sky, as shown in blue. Each blue bar indicates the direction and degree of polarisation of the light from that particular region of the sky. In each sky region, the light is polarised along a direction that is perpendicular to the solar axis, and the pattern of polarisation in the sky is circularly symmetrical about the solar axis. If the sun is hidden by a cloud, the insect can maintain a constant heading by monitoring and holding constant the orientation of the polarised skylight directly overhead. In this case, light from the overhead sky is polarised in a direction that coincides with the flight direction. The sun cannot provide any compass information when it is directly overhead. Neither can the polarised light pattern, as it is always circularly symmetrical with respect to the solar axis.

pattern of polarised light that is produced by the sun in the sky, and to use this pattern as a 'polarised-light compass' in lieu of the sun when the sun is obscured by a cloud,[11] as illustrated in Fig. 1.

Examples of the application of the polarised light compass for robot navigation are described in.[12–16] In principle, the sun and its associated pattern of polarised light can also be used to compute yaw rates. However, this system breaks down or performs poorly when the sun is directly or nearly overhead, because of the rotational symmetry of the pattern of intensity or polarisation of the skylight about the sun's axis.[11] Measuring rotations in 3-D during flight is more complex. The sun or the polarisation pattern of the sky cannot on their own provide reliable rotational cues, because any component of rotation about the sun's axis will not be sensed. When flying high above flat terrain, the panoramic profile of the horizon can be used to estimate roll and pitch. Indeed, there is substantial evidence to indicate that flying insects use specialised light-sensitive organs – the ocelli – to sense the horizon profile, and use this information to control attitude.[17,18] This principle has been used in a number of robotic applications (e.g., refs. 14, 19, 20). However, the horizon profile does not change during yaw, and therefore cannot be used to sense this component of rotation. One way to estimate the yaw component visually in a general 3-D rotation would be to measure the roll and pitch by using the horizon profile, and then derotate the panoramic sky image to remove the roll and pitch components of rotation. The yaw component is then the only remaining rotational component in the sky image, which can be measured from the azimuthal rotation of the polarisation pattern in the sky,[14] or of the spatial intensity pattern created by the sun and the clouds in the sky.[21] However, these approaches are

https://doi.org/10.1017/S0263574718000577 Published online by Cambridge University Press

constrained by the inherent requirement that the horizon be within the field of view (FOV), which may not be the case for small UAS platforms operating in urban environments at low altitudes. Therefore, there is an important need for determining 3DOF attitude robustly in the absence of a visible horizon.

1.3. Our method of egomotion measurement

This paper forms the capstone to a recent series of developments in vision-based navigation, the goal of which is autonomous, bio-inspired flight, partly inspired by the navigational prowess of insects. The overarching problem statement for this body of work was to develop a vision-only navigation system, a task that has been developed incrementally over several studies, and for the first time completed in this paper onboard a rotorcraft UAS. Note that IMU or other sensor data could, in principle be fused through the use of a Kalman Filter.[22] However, such ventures fall outside the scope of this study, because the rationale for our endeavour is to investigate whether a competent navigational capacity can be implemented in a quadrotor (a) using only insect-based sensing, and (b) relying only on vision.

Thurrowgood *et al.*[23] describe a vision-based landing system for a fixed-wing UAS. This study utilises optic flow, computed by using a pyramidal block-matching method over a grid of 20×12 points, with sub-pixel refinement achieved using equiangular fitting. It assumes that the ground can be approximated by a plane, and simultaneously derives the translation and rotation vectors using an iterative non-linear derivative-free gradient descent method over the 6DOF search space. The translation vectors are expressed in terms of height units as that study only required a ratio of altitudes between consecutive frames to land the aircraft. An extension to the above approach is described by Strydom *et al.*[24] and implemented on a quadrotor UAS. Stereo height information – also obtained through the vision system described in Section 2 – is used to scale the translation estimates, which are then used in a navigation scheme that implements robust 3-D visual odometry. That study used an IMU to determine the 'up' direction relative to the world coordinate frame, resulting in data useable for navigation. A further adaptation of this concept was developed by Denuelle and Srinivasan,[25] which overcame the inherent integration error associated with frame-to-frame optic flow based models by using a sparse series of reference frames for navigating through a series of waypoints by successively matching a sequence of 'snapshots'. While both of these methods produced significant results, they relied on IMU data – leaving the vision-only navigation problem unsolved.

We propose a novel, vision-only approach that first utilises a sky compass to determine the 3DOF rotation of the aircraft[26] and then applies this information to derotate the optic flow generated by the ground. Once derotated, the residual ground-based optic flow vectors provide purely translational information in 3DOF, which is computed separately. The rotation and translation results are then combined to obtain the full 6DOF motion of the UAS (scaled to the height above the ground). Stereo height information is then incorporated to scale the translation vectors, thus achieving odometry and ultimately, path integration, using purely visual information.

The method proposed in this paper provides a solution to the goal of autonomous bio-inspired flight through the use of computer vision, and furthermore derives and demonstrates the significant final step to achieving a vision only navigation system by completely removing the IMU dependency. Throughout the remainder of this paper, results are compared directly to those of the preceding algorithm (i.e., the work by Strydom *et al.*[24]) due to unfettered access to their algorithm. We also provide a comparison of the performance of our system with those of some other state of the art techniques.

2. Flight Platform

The UAS used for flight-testing in this study is a custom built quadrotor (see Fig. 2A). An Intel NUC computer (with a 2.6 GHz dual-core processor) is used to perform all image processing, navigation and control onboard the flight platform. The quadrotor also features a MicroKopter flight controller, speed controller and motors for rotor control. Onboard sensors include the bio-inspired vision system shown in Fig. 2B, a MicroStrain 3DM-GX3-25 attitude and heading reference system IMU, and a Piksi GPS receiver.

The vision system operates at a 25 Hz frame rate, and consists of two fisheye cameras (Point Grey Firefly MV cameras with Sunex DSL216 lenses). Each camera has a FOV of approximately 190°, and they are configured back-to-back while also tilted 10° towards the front in order to achieve a stereo overlap of approximately 30° by 130° with a baseline of 10 cm. The cameras are calibrated using a

https://doi.org/10.1017/S0263574718000577 Published online by Cambridge University Press

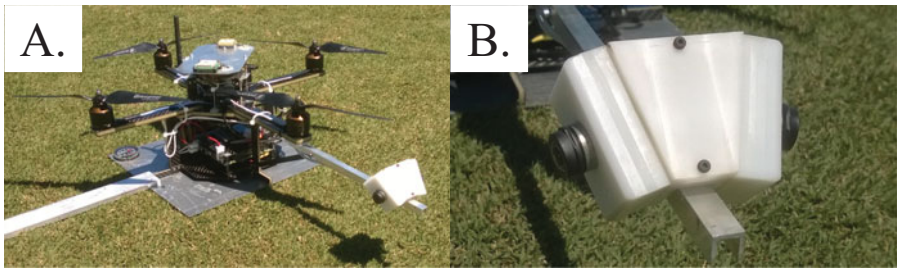


Fig. 2. (A) The quadrotor used in the study; (B) The bio-inspired vision system.

method similar to that of Kannala and Brandt.[27] The images from the two cameras are stitched together, resulting in a panoramic image with a FOV of approximately 360° azimuth and 150° elevation. The resolution of the stitched images is 360 by 220 pixels. A mask is also generated during this process to conceal regions of the aircraft in the image that occlude vision of the surrounding environment.

3. Egomotion Estimation

Panoramic images of outdoor scenes provide an attractive opportunity for naturally segmenting egomotion into its rotational and translational subcomponents. Since the sky can be assumed to be infinitely far away for low altitude UAS applications, its motion in the camera image is affected only by the rotations of the aircraft, and not by its translations.[28] The sky image thus offers a convenient means of estimating the rotation of the aircraft in 3DOF, uncontaminated by the aircraft's translation.[26]

In contrast, the motion of the ground image contains information about the rotation as well as the translation of the aircraft. In principle, the entire 6DOF motion of the aircraft can be recovered by computing iteratively the 6DOF egomotion parameters that best account for the optic flow measured between successive image frames, as described, for example, in Thurrowgood *et al.*[23] However, segmentation of the panoramic image into sky and ground regions simplifies the egomotion computation and enables independent and robust estimation of rotation and translation. Specifically, this is achieved by splitting the computation into two parts, using the sky image to compute the 3DOF rotation, and the ground image to compute the 3DOF translation.

An additional benefit of segmenting the computation into rotational and translational components is that it provides a partial relaxation of the planar ground assumption (as rotations estimated using only the sky portion of the panoramic image instead require that the scene is sufficiently far away). Notably this results in a decoupled estimate of rotation and translation, where any error associated with the planar assumption only affects the 3DOF translation estimation, rather than the full 6DOF egomotion computation.

Campbell *et al.*[29] explored a simplified version of this approach on a ground-based robot, to compute yaw and 1-D translation using a two-stage optic flow based algorithm. Our technique builds on this concept by first computing the 3DOF rotation using images of the sky, and then using this rotational information in combination with images of the ground to find the 3DOF translation of the aircraft. This process can be split into the following stages:

1. Image pre-processing.
2. Sky-based rotation estimation in 3DOF, by minimising the SAD between the derotated current image and a reference image acquired at the start of the egomotion computation.
3. Computation of the rates of rotation about the three principal axes (x, y, z) using a central difference formula to implement smoothed numerical differentiation.
4. Updating the reference sky image for subsequent rotation estimates.
5. Computation of the frame-to-frame optic flow field generated by the ground, using a pyramidal block-matching method.
6. Derotation of the optic flow generated by the ground, using the information from (3).
7. Estimation of translation by using an iterative optimisation to find the 3DOF translational vector that best accounts for the derotated optic flow field generated by the ground.

https://doi.org/10.1017/S0263574718000577 Published online by Cambridge University Press

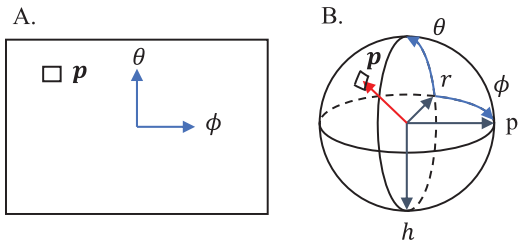


Fig. 3. (A) Panoramic image space pixel representation, $\boldsymbol{p} = f(\phi, \theta)$, where ϕ is the azimuth and θ the elevation. (B) World sphere pixel representation, $\boldsymbol{p} = g(r, p, h)$, with coordinates roll, pitch and heading.[26]

The outputs of the proposed algorithm are the rates of angular and translatory egomotion. These are combined with a stereo estimate of height to enable to-scale visual odometry [8].

3.1. Sky-based rotation estimation

The sky-based rotation estimation algorithm used in this paper estimates the roll, pitch and heading in any frame with respect to the aircraft's original orientation, by monitoring the motion of the spatial intensity pattern in the sky, which comprises the sun, the clouds and other intensity gradients in clear parts of the sky. The computed rotation is converted to the world frame through the use of a one-time initialisation using data from the onboard IMU at the start of the flight. This process is described in detail below, however, for a deeper insight including the results of both a validation procedure and flight-testing, see our previous paper.[26]

3.1.1. Pre-processing. The panoramic RGB images generated by the vision system are converted to grayscale, with a 2-D Gaussian spatial low-pass filter applied to each image using a 5×5 kernel. The first frame is then stored as the base reference image.

3.1.2. Absolute rotation estimation. Panoramic images can be accurately represented in three dimensions using a 'world sphere'. The images are mapped onto the unit sphere in such a way that each point on the sphere represents the pixel intensity in the corresponding viewing direction. This results in a 3-D vector representation of each pixel location, allowing for image rotations to be represented through the application of a rotation matrix (see Fig. 3).

The sky-compass algorithm utilises the equivalence of these two representations to link the intensity gradients and geometry captured by the panoramic images to the physical orientation of the aircraft. The conversion between the panoramic image space and the world sphere is encapsulated within the camera model calibration. An estimate of the aircraft orientation is obtained by applying a predicted rotation to the current image, then comparing the result with the stored reference frame using a SAD. The match is improved iteratively in panoramic image space by adjusting the rotation applied to the current image when represented on the world sphere.

To summarise the absolute rotation estimation process, first consider the relationship between the current image and the image generated using the rotation estimate (see Fig. 4). Note that in this paper, a single dash is used to indicate that a quantity is an estimate, while subscripted square brackets emphasise the reference frame.

Formally, the derotated current image (which will be iteratively adjusted to match the stored reference image) is obtained by applying an initial arbitrary rotation, $\boldsymbol{\Omega}'_t$, to each rotation-valid pixel, $\boldsymbol{p}_{i.j}$, within the current image. Rotation-valid pixels are defined as pixels located within the intersection of a 150° globally vertically oriented cone and the surface of the world sphere (as illustrated in Fig. 5), as well as being outside the image masks. The rotation is computed using the Rodrigues algorithm, defined as $R(\boldsymbol{\zeta})$, which produces the 3×3 rotation matrix corresponding to a given rotation vector, $\boldsymbol{\zeta}$. However, due to the rotation process resulting in non-integer image coordinates when applied directly, a two-step process is used to compute pixel intensity values in the derotated current image: (1) The location of each pixel in the derotated frame, $[\boldsymbol{p}_{i,j}]'_R$, is obtained in the current frame, $[\boldsymbol{p}_{x,y}]_C$:

$$[\boldsymbol{p}_{x,y}]_C = R(-\boldsymbol{\Omega}'_t) \cdot [\boldsymbol{p}_{i,j}]'_R. \tag{1}$$

https://doi.org/10.1017/S0263574718000577 Published online by Cambridge University Press

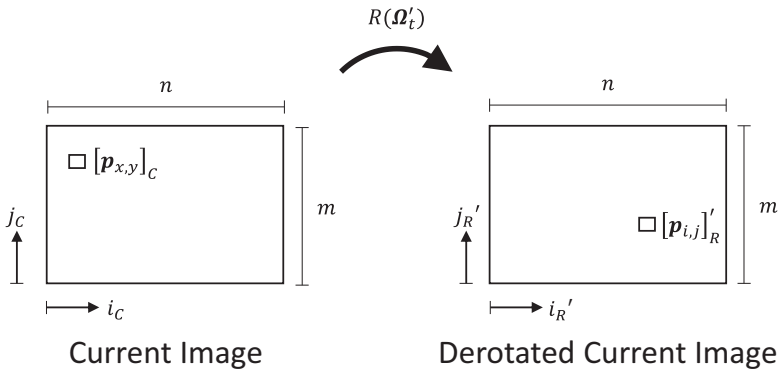


Fig. 4. Image transformation describing the rotation from the current frame to the current estimate of the reference frame.

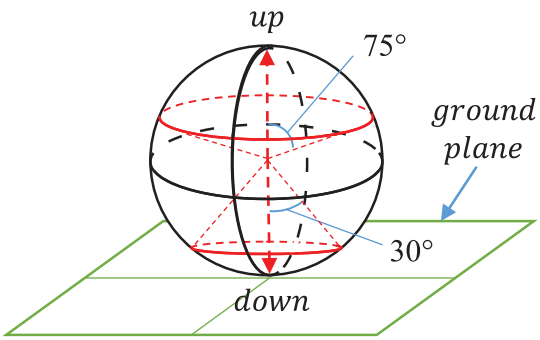


Fig. 5. The sky-cone, ground-cone and ground plane depicted relative to the world sphere, with the quadrotor at the centre. The reduced field of view of the sky (75°) and ground (30°) are not critical but they are chosen to ensure that the sky cone only sees the sky and the ground cone only the ground despite changes in roll/pitch. A more robust extension would be to segment the field of view based on an independently detected horizon line, however, this was not the focus of this research.

(2) The intensity, $I(\boldsymbol{p}, t)$, of the pixels in the derotated frame is calculated using bilinear interpolation based on neighbouring pixel intensities in the current frame:

$$I([\boldsymbol{p}_{i,j}]'_R, t) = (\lceil y \rceil - y)\Big((\lceil x \rceil - x)I([\boldsymbol{p}_{\lfloor x \rfloor, \lfloor y \rfloor}]_C, t) + (x - \lfloor x \rfloor)I([\boldsymbol{p}_{\lceil x \rceil, \lfloor y \rfloor}]_C, t)\Big) + (y - \lfloor y \rfloor)\Big((\lceil x \rceil - x)I([\boldsymbol{p}_{\lfloor x \rfloor, \lceil y \rceil}]_C, t) + (x - \lfloor x \rfloor)I([\boldsymbol{p}_{\lceil x \rceil, \lceil y \rceil}]_C, t)\Big), \tag{2}$$

where the notations $\lfloor x \rfloor$ and $\lceil x \rceil$, respectively, denote the integer floor and ceiling of the decimal pixel positions resulting from the derotation. The error between the derotated current image and the stored reference image is then quantified by computing the SAD between the two images and normalising by the total number of image pixels ($n \times m$):

$$C_R = \frac{\sum_i^n \sum_j^m (|I([\boldsymbol{p}_{i,j}]'_R) - I([\boldsymbol{p}_{i,j}]_R)|)}{n \times m}. \tag{3}$$

Finally, the 3DOF rotation of the aircraft, $\boldsymbol{\Omega}_t$, is estimated by iteratively adjusting the prediction, $\boldsymbol{\Omega}'_t$, to minimise the normalised SAD between the derotated current image and the stored reference image (see Eq. (4)). The optimisation is performed using an algorithm from the NLopt library,[30] BOBYQA,[31] which is a derivative-free gradient descent method.

$$\boldsymbol{\Omega}_t = \arg\min_{\boldsymbol{\Omega}'_t} C_R(\boldsymbol{\Omega}'_t). \tag{4}$$

https://doi.org/10.1017/S0263574718000577 Published online by Cambridge University Press

The initial guess and search bounds for each image are the rotation estimates (roll, pitch and heading) from the previous image, and those values $\pm 10^{\circ}$, respectively.

3.1.3. Rotational rates. Rotational rates are calculated from the 3DOF orientation estimates using a central difference formula (Eq. (5)) to facilitate numerical differentiation, which introduces a single frame time delay.

$$\boldsymbol{\omega}_{t-1} = \frac{d\boldsymbol{\Omega}_{t-1}}{dt} \approx \frac{\boldsymbol{\Omega}_t - \boldsymbol{\Omega}_{t-2}}{2}. \tag{5}$$

Although higher order approximations reduce noise in the derivative signal, they inherently introduce additional delay into the rotation estimation system, resulting in errors when estimating translation.

3.1.4. Accumulative reference image. As the sun and clouds can move across the sky, it is necessary to address the problem of defining a reference image for a dynamic scene. Inspired by the work of Moore *et al.*,[21] the stored reference image and associated mask are iteratively updated in order to (a) fill in regions of the sky image that were previously occluded by the aircraft and (b) track changes in the sky image. This increases the robustness of the scheme when encountering changes in the brightness of sky regions, or in the shape or position of any cloud formations. The updating process is defined as

$$I([\boldsymbol{p}_{i,j}]_R, t) = \alpha I([\boldsymbol{p}_{i,j}]_C, t) + (1-\alpha) I([\boldsymbol{p}_{i,j}]_R, t-1), \tag{6}$$

where α is the accumulation rate, quantifying the proportion of the instantaneous reference image estimation incorporated into the saved reference image per time-step. For this study, $\alpha = 0.01$, corresponding to an update time of $t_u \approx 4$s at 25 Hz as suggested by Moore *et al.*[21] Note that to avoid contaminating the reference image, only rotation-valid pixels are used in the updating process. A rotation-valid pixel is defined as a pixel located within the sky-cone but not concealed by the current mask.

3.2. Ground-based translation estimation

The proposed translation estimation algorithm utilises the sky-based rotation estimate to reduce the 6DOF egomotion calculation to a calculation of translation in 3DOF. To compute translation in 3DOF, optic flow vectors are calculated between two successive images using a pyramidal block matching method, and then derotated using the rotational rates from the sky-based computation. After derotation, the residual optic flow vectors represent the flow due to pure translation, which can be computed by iteratively adjusting the three components of translational motion to minimise the error between the predicted translatory optic flow and the translational flow that is actually experienced. Prediction of the translational flow is simplified by making the assumption that the ground forms a plane tangential to the world sphere at the down location (as shown in Fig. 5). The orientation of the down vector (and by definition, the ground plane) is computed using the sky-based rotation estimate. Note that during the translation estimation process, the translation ($\boldsymbol{\tau}_t$) per frame is measured in multiples of the instantaneous height above ground (referred to as either a variable, h_t, or as a unit, $\mathrm{u_h}$) with units $\mathrm{u_h}$/frame. The translation can be converted to any other distance unit (e.g., meters, m) by scaling the translation estimate $\boldsymbol{\tau}_t$ (in $\mathrm{u_h}$/frame) by the height above ground in that unit (i.e., h_t in m/$\mathrm{u_h}$). Similarly, the temporal unit can be converted to per second when the translation estimation is scaled by the frame-rate f in units of frames/s. Together, the conversion is expressed by

$$\boldsymbol{\tau}_{t,act} = h_t f \boldsymbol{\tau}_t. \tag{7}$$

This relationship is used for path integration (Section 3.3) and data analysis.

3.2.1. Optic flow. As stated above, the optic flow from the ground is computed using a pyramidal block matching method. Each image is divided into 25×12 pixel subregions (using a 360×180 pixel section of the 360×220 pixel stitched image), forming a 14×14 block grid. A point is chosen at random within each block, and the pixels in a 7×7 patch surrounding that point are used to obtain a match by searching a 7×7 area in the consecutive image. Only image patches located within the intersection of a 60° globally downward cone and the world sphere are used, as these provide the best

https://doi.org/10.1017/S0263574718000577 Published online by Cambridge University Press

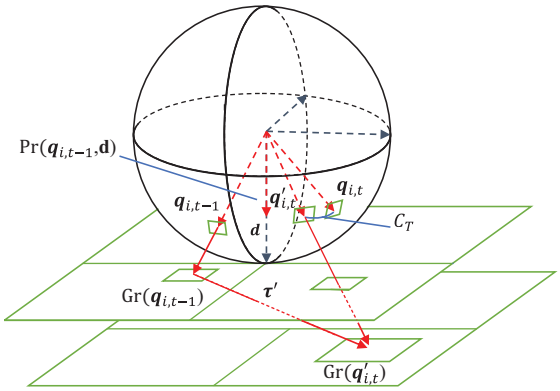


Fig. 6. The translation estimation process, with the quadrotor positioned at the centre of the sphere. $\boldsymbol{q}_{i,t-1}$ is the derotated image patch location; $\boldsymbol{d}$ is the unit down vector; $Pr(\boldsymbol{q}_{i,t-1}, \boldsymbol{d})$ is the projection of $\boldsymbol{q}_{i,t-1}$ in the direction of $\boldsymbol{d}$; $Gr(\boldsymbol{q}_{i,t-1})$ is $\boldsymbol{q}_{i,t-1}$ when scaled from the world sphere to the surface of the ground plane; $\boldsymbol{\tau}'_t$ is the assumed translation between frames; $Gr(\boldsymbol{q}_{i,t})$ is the estimated current location on the ground plane of the image patch; $\boldsymbol{q}'_{i,t}$ is the estimated current location on the world sphere of the image patch; $\boldsymbol{q}_{i,t}$ is the image patch current location as computed by the optic flow algorithm; C_T is the direct distance between $\boldsymbol{q}'_{i,t}$ and $\boldsymbol{q}_{i,t}$, which approximates the distance travelled along the surface of the world sphere.

signal to noise ratio for translation (see Fig. 5). The best match is computed by minimising the SAD, with sub-pixel refinement achieved using equiangular fitting.[32]

3.2.2. Translation estimation. The translation estimation process is illustrated in Fig. 6. Consider a single image patch, located on the world sphere by the vector $\boldsymbol{q}_{i,t-1}$ during the previous time-step. The patch location is first derotated from the previous camera frame, $]_P$, into the current camera frame, $]_C$, using the rotation rate estimate from the sky-based computation, $\boldsymbol{\omega}_t$. The rotation is computed using the Rodrigues algorithm, which produces the 3×3 rotation matrix corresponding to the rotation vector ($\boldsymbol{\omega}_t$).

$$[\boldsymbol{q}_{i,t-1}]_C = R(-\boldsymbol{\omega}_t) \cdot [\boldsymbol{q}_{i,t-1}]_P. \tag{8}$$

Dropping the reference frame bracket notation as the remainder of this section deals solely with the current frame, $]_C$, the projection of $[\boldsymbol{q}_{i,t-1}]_C$ in the direction of the unit down vector $\boldsymbol{d}$ is given by

$$Pr(\boldsymbol{q}_{i,t-1}, \boldsymbol{d}) = \frac{(\boldsymbol{d}^T \cdot \boldsymbol{q}_{t-1})}{|\boldsymbol{d}|^2}\boldsymbol{d} = (\boldsymbol{d}^T \cdot \boldsymbol{q}_{i,t-1})\boldsymbol{d}. \tag{9}$$

$Pr(\boldsymbol{q}_{i,t-1}, \boldsymbol{d})$ is then used to scale $\boldsymbol{q}_{i,t-1}$ such that it intersects the ground plane by taking the vector element-wise quotient:

$$Gr(\boldsymbol{q}_{i,t-1}) = \left[\frac{(\boldsymbol{q}_{i,t-1})_1}{(Pr(\boldsymbol{q}_{i,t-1}, \boldsymbol{d}))_1}, \frac{(\boldsymbol{q}_{i,t-1})_2}{(Pr(\boldsymbol{q}_{i,t-1}, \boldsymbol{d}))_2}, \frac{(\boldsymbol{q}_{i,t-1})_3}{(Pr(\boldsymbol{q}_{i,t-1}, \boldsymbol{d}))_3}\right]^T. \tag{10}$$

The translation vector estimate $\boldsymbol{\tau}'_t$ is then added, and the result normalised to obtain an estimate of the location $\boldsymbol{q}_{i,t}$ (the current location of $\boldsymbol{q}_i$ as computed by optic flow), denoted by $\boldsymbol{q}'_{i,t}$

$$\boldsymbol{q}'_{i,t} = \frac{Gr(\boldsymbol{q}_{i,t-1}) + \boldsymbol{\tau}'_t}{\|Gr(\boldsymbol{q}_{i,t-1}) + \boldsymbol{\tau}'_t\|}. \tag{11}$$

Finally, the squared distance between $\boldsymbol{q}_{i,t}$ and $\boldsymbol{q}'_{i,t}$ is computed as a measure of the translation estimation error, which approximates the distance travelled along the surface of the world sphere:

$$C_T = \|\boldsymbol{q}_{i,t} - \boldsymbol{q}'_{i,t}\|^2. \tag{12}$$

https://doi.org/10.1017/S0263574718000577 Published online by Cambridge University Press

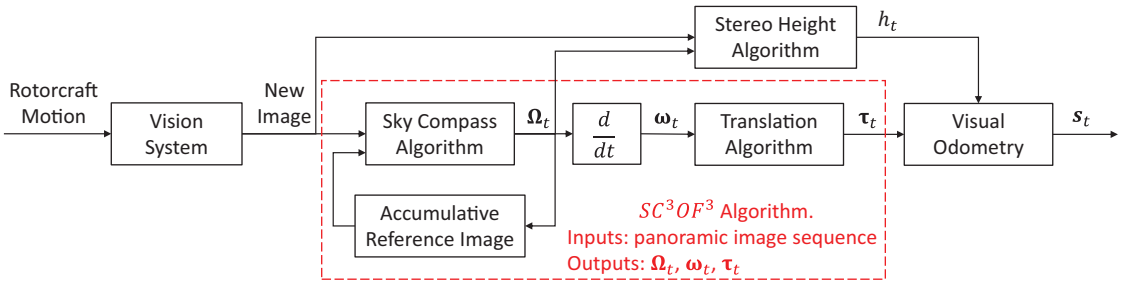


Fig. 7. Overview of the SC^3OF^3 algorithm (as detailed in Sections 3.1 and 3.2), highlighting the interface with the vision system and visual odometry process as well as the role of the stereo height computation. Height above ground is computed using stereo vision and the instantaneous attitude of the aircraft, $\boldsymbol{\Omega}_t$, as explained in Section 3.3.

The error C_T is minimised using an iterative optimisation algorithm, with the translation vector at the minimum ($\boldsymbol{\tau}_t$) corresponding to the translation of the UAS. Similar to the sky-based rotation estimation process, this minimisation is achieved using the BOBYQA[31] derivative free gradient descent method from the NLopt library.[30] Formally, the objective function used for the minimisation is as follows:

$$\boldsymbol{\tau}_t = \arg\min_{\boldsymbol{\tau}_t'} \sum_{i=1}^{N} C_T(\boldsymbol{\tau}_t'). \tag{13}$$

Here, N represents the number of image patches used for the calculation process, in this study $N = 400$. This concludes the 6DOF egomotion estimation, with the optimised orientation, rotational and translation rates ($\boldsymbol{\Omega}_t$, $\boldsymbol{\omega}_t$ and $\boldsymbol{\tau}_t$, respectively) forming the outputs of the algorithm. Figure 7 provides a graphical summary of the egomotion estimation process, henceforth, referred to as the SC^3OF^3 algorithm.

3.3. Visual odometry

The translation vectors computed by the process outlined above are in units scaled by the aircraft's height in the first frame of the image pair (see Eq. (7), however, leaving the temporal unit as per frame). The scale factor h_t is obtained using the stereo height computed by the modified optic flow algorithm as described in earlier studies.[23,24] The range to each image patch within the stereo overlap region is estimated using vector triangulation. The range is then converted to the height over ground using roll and pitch data from the sky-based rotation estimation. A histogram of all matched image patches is used to develop a confidence measure for the height, improving the accuracy of the final height estimate that is used to scale the translation vectors to obtain the true translations. Stereo measurements using the vision system described in Section II are valid over a [0.2, 15]m range, with an error obtained during hover at 3 m of -0.225 ± 0.299 m as per[33] (3 m is the height used during flight tests in this study). The current 3-D position of the aircraft, relative to a starting reference location, is computed by summing all of the frame-to-frame translational vectors that have been measured up to the present time. This iterative process, termed path integration, can be described as follows:

$$\boldsymbol{s}_t = \boldsymbol{s}_{t-1} + h_t \boldsymbol{\tau}_t, \tag{14}$$

where $\boldsymbol{s}_t$ and $\boldsymbol{s}_{t-1}$ are the current and previous 3-D positions relative to the starting point, respectively, h_t is the scalar height above ground as computed using stereo and $\boldsymbol{\tau}_t$ is the inter-frame translation measured in units of height above the ground. $\boldsymbol{\tau}_t$ is multiplied by the current height above ground (h_t) to obtain the inter-frame translation in metres.

3.4. Closed-loop control of aircraft motion

Navigation of the quadrotor is achieved by prescribing a series of 3-D position setpoints that define the desired trajectory. Associated with each setpoint are both spatial (distance to the point) and temporal

https://doi.org/10.1017/S0263574718000577 Published online by Cambridge University Press

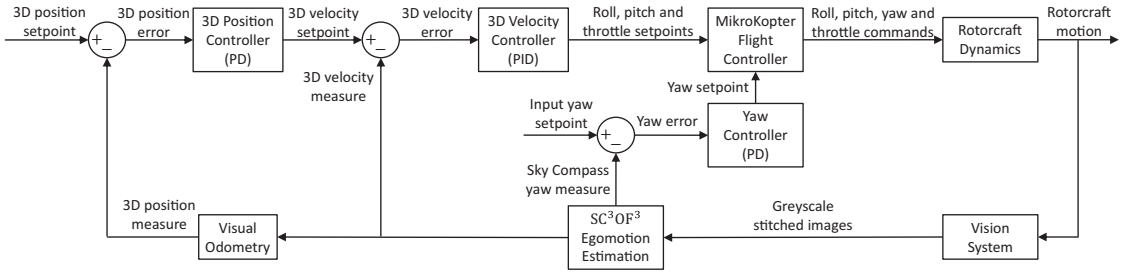


Fig. 8. Onboard control structure used by the flight platform.

(time within the specified setpoint region) tolerances, allowing the high level control architecture to robustly determine when the 3-D position setpoint has been reached.

Onboard control of the aircraft operates at the frame rate of the cameras (25 Hz), and is achieved by regulating the quadrotor's 3-D position and velocity using three sets of proportional-derivative (PD) and proportional-integral-derivative (PID) controllers, respectively. The objective is to minimise the error between the currently targeted 3-D position setpoint and currently estimated position of the quadrotor. Figure 8 is a graphical representation of this process. The form of the controllers, noting that the gains are vehicle specific, is given by the following:

$$C_{\mathrm{PD}}(s) = K_P + sK_D \tag{15}$$

$$C_{\mathrm{PID}}(s) = K_P + sK_D + \frac{1}{s}K_I \tag{16}$$

The 3-D position controller uses the position error to generate a velocity command, which is then compared to the vision-based velocity estimate to produce a velocity error. A PID controller then works to minimise the velocity error by adjusting the roll, pitch and throttle setpoints. Independently, another PD controller acts to control the yaw of the platform, with the resultant yaw set point combined with the velocity controller output and subsequently fed into the Mikrokopter flight controller. The Mikrokopter flight controller is effectively a black box, tasked with converting the setpoints into the required rotor commands. The functional block labelled 'SC^3OF^3 Egomotion Estimation' represents the algorithm described in Sections 3.1 and 3.2, while the block labelled 'Visual Odometry' indicates the process outlined in Section 3.3.

4. Results

A series of flight tests were conducted to validate and evaluate the proposed algorithm for the measurement of instantaneous translation in 3DOF (Section 4A), and also as a part of a visual odometry system (Section 4B). Navigation for each trial was completed in closed loop, utilising open loop translation measurements, with all processing occurring in real-time onboard the aircraft. The specified trajectories consisted of square flight paths using six waypoints (each with a 0.5 m spatial and 5 frame temporal tolerance) of varying size (5 × 5 m, 10 × 10 m, 15 × 15m and 20 × 20 m) at an altitude of 3 m (the influence of height on the accuracy of the odometry algorithm was investigated by Strydom *et al.*,[24] reporting negligible effects at low altitudes). Each test was conducted five times, for a total of 20 flights. The flights were conducted in an outdoor environment (Fig. 9), under a variety of conditions over several days. The final stage of landing (specifically, when the aircraft's altitude was less than 0.5 m during descent) was completed in open loop, with all control specifications aside from rotor speed relaxed.

The proposed algorithm was run in parallel with an implementation of the method outlined by Strydom *et al.*[24] to provide a quantitative comparison. Differential GPS (DGPS) data was used as the ground truth for all analysis. The Piksi DGPS operated at 5 Hz, while both vision-based methods ran at the camera frame-rate of 25 Hz. To account for this, in the post-flight analysis all vision data was sub-sampled at 5 Hz.

https://doi.org/10.1017/S0263574718000577 Published online by Cambridge University Press

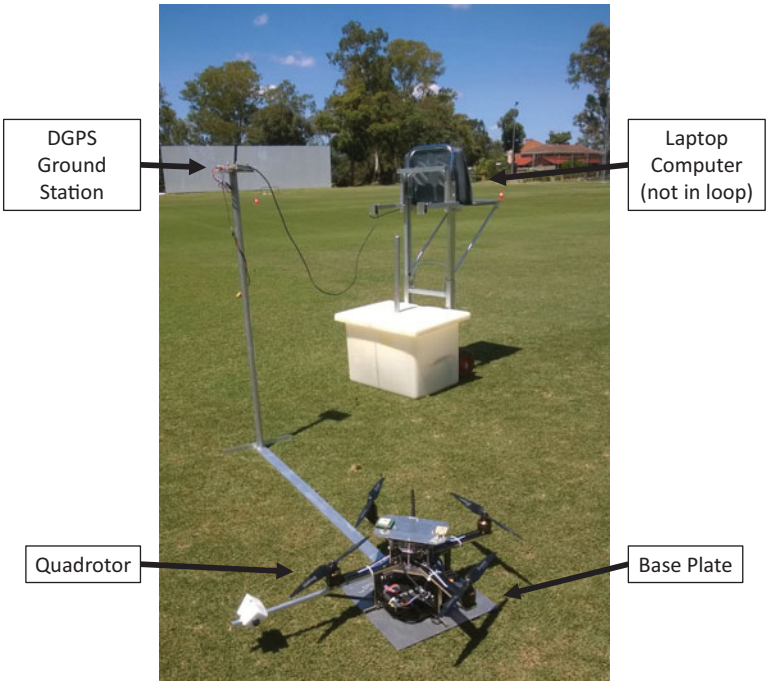


Fig. 9. The flight platform in the test environment. The quadrotor is sitting on a specially designed base plate to ensure high repeatability of launch location and attitude. The base plate fixes the initial position of the DGPS ground station relative to the quadrotor. Also shown is the laptop computer used for offline data analysis.

Throughout these sections, we denote our proposed algorithm the SC^3OF^3 method, and the egomotion estimation algorithm proposed by Strydom *et al.*[24] the OF^6 method.

4.1. Algorithm evaluation – translation estimation

The results illustrating the stand-alone performance of the algorithm are described in full in Table 4.1, which lists the mean and standard deviation of the error (in u_h/frame $\times 10^{-3}$) with respect to the ground truth (DGPS) for both the SC^3OF^3 and OF^6 methods – where the mean and standard deviation are computed for each set of five tests for 5 m, 10 m, 15 m and 20 m, respectively. Translation error is defined as the difference between the vision-based estimate and the ground truth (DGPS) measurement of frame-to-frame translation when sampled at 5 Hz. Figure 10 provides a graphical representation of this comparison, with plots describing the translation (in u_h/frame) against time (in seconds). The SC^3OF^3 method is shown in blue, the OF^6 approach in red and the ground truth is described by the shaded grey region corresponding to the known uncertainty of the DGPS (X, Z: ± 0.025 m/frame, Y: ± 0.075 m/frame). The translation is analysed in height units so that any error associated with the stereo computation of absolute height does not distort the output of the egomotion estimation algorithm. In order to accommodate the comparison, the DGPS data is scaled by the assumed height (3 m).

4.2. System evaluation – loop closure

The second experiment undertaken during this study tested the proposed closed-loop navigation scheme by prescribing a sequence of six way points, and using vision-based odometry and path integration to track and control the aircraft's trajectory. Loop closure is used as a metric to evaluate the performance of the algorithm when integrated into a visual odometry scheme. The true flight paths of all flights as recorded by the ground truth are overlaid for reference in Fig. 11A. Note that the 'scalloped' appearance of the flight paths observed in Fig. 11A is likely a result of the coarsely tuned flight controller, as these features also occur in both of the vision-based position estimates, which agree with the DGPS ground truth estimates, suggesting that it is not an algorithmic error (see Fig. 11C). Figure 11B shows the loop closure positions for individual landing points (shown in blue), the

https://doi.org/10.1017/S0263574718000577 Published online by Cambridge University Press

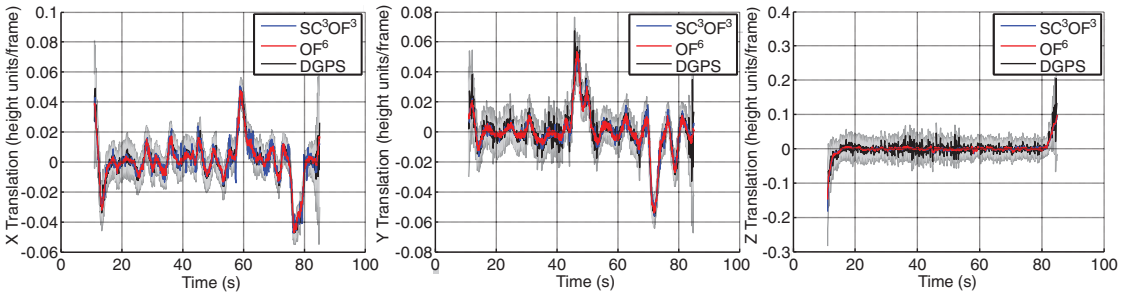


Fig. 10. Instantaneous translation (u_h/frame) vs. time (s) for a typical flight comparing the results for SC^3OF^3 (blue, detailed in Sections 3.1 and 3.2), OF^6 (red, originally described by Strydom *et al.*[24]) and differential GPS (black). The grey shading represents the known uncertainty of the ground truth data. Large spikes correspond to movement along the edges of the prescribed square trajectory, while smaller oscillations represent the UAS hovering at a position setpoint. Note that the Z direction is defined as the direction of the down vector, and the large initial and final values represent take-off and landing, respectively.

Table II. Translation error (u_h/frame $\times 10^{-3}$) compared to the ground truth (DGPS) data for SC^3OF^3 (detailed in Sections 3.1 and 3.2) and OF^6 (originally described by Strydom *et al.*[24]). Equation (7) can be used to determine the instantaneous translation in m/s, using the assumed height ($h_t = 3$ m) and known sample rate ($f = 5$ Hz). Rows denoted 'All' and 'Total' represent the mean and standard deviation found using all of the raw data for that particular path size and for all path sizes, respectively.

		SC^3OF^3 Error ($\mu \pm \sigma$, u_h/frame $\times 10^{-3}$)			OF^6 Error ($\mu \pm \sigma$, u_h/frame $\times 10^{-3}$)		
Flight		X	Y	Z	X	Y	Z
5×5	1	-0.22 ± 6.70	0.05 ± 6.11	0.56 ± 8.65	-0.25 ± 4.18	-0.10 ± 4.87	0.35 ± 8.32
	2	-0.15 ± 5.76	0.04 ± 6.64	-0.40 ± 9.99	-0.22 ± 4.55	-0.05 ± 5.50	-0.55 ± 9.28
	3	-0.09 ± 6.49	-0.01 ± 6.87	-0.93 ± 13.46	-0.10 ± 5.59	-0.26 ± 6.76	-0.90 ± 12.85
	4	-0.25 ± 6.65	0.11 ± 6.90	-0.45 ± 16.64	-0.22 ± 4.84	-0.05 ± 6.48	-0.64 ± 16.48
	5	2.03 ± 14.67	0.27 ± 11.10	0.92 ± 21.07	2.16 ± 14.69	-0.26 ± 10.87	0.73 ± 20.01
	All	0.01 ± 7.50	0.07 ± 7.09	-0.12 ± 13.06	-0.00 ± 6.30	-0.12 ± 6.40	-0.27 ± 12.55
10×10	1	0.07 ± 5.82	-0.19 ± 6.78	-0.57 ± 10.45	-0.01 ± 5.31	-0.51 ± 6.65	-0.84 ± 9.09
	2	-0.26 ± 7.25	0.21 ± 8.15	-0.89 ± 9.80	-0.11 ± 6.34	-0.23 ± 7.51	-1.08 ± 8.68
	3	-0.18 ± 7.55	-0.06 ± 7.28	-0.34 ± 7.65	-0.10 ± 6.17	-0.44 ± 6.42	-0.58 ± 7.21
	4	-0.47 ± 10.54	-0.04 ± 10.82	-1.17 ± 17.72	-0.16 ± 9.75	-0.57 ± 9.56	-1.12 ± 17.62
	5	-0.07 ± 7.30	0.00 ± 7.32	-0.29 ± 7.89	0.12 ± 6.07	-0.21 ± 6.24	-0.42 ± 7.40
	All	-0.14 ± 7.49	-0.04 ± 7.87	-0.59 ± 10.68	-0.03 ± 6.55	-0.39 ± 7.11	-0.77 ± 9.99
15×15	1	-0.38 ± 7.85	0.13 ± 7.86	-0.70 ± 11.07	-0.02 ± 7.17	-0.49 ± 7.76	-0.89 ± 10.39
	2	-0.09 ± 7.13	0.06 ± 6.84	-0.13 ± 8.71	-0.02 ± 6.06	-0.32 ± 6.17	-0.43 ± 8.28
	3	-0.57 ± 11.16	-0.41 ± 9.57	-1.09 ± 12.48	-0.23 ± 9.51	-0.62 ± 8.33	-1.08 ± 11.80
	4	0.06 ± 8.03	-0.17 ± 6.92	-0.35 ± 9.54	-0.07 ± 6.70	-0.54 ± 6.41	-0.80 ± 8.98
	5	-0.14 ± 6.50	-0.33 ± 6.77	-0.29 ± 8.65	-0.01 ± 5.38	-0.61 ± 5.94	-0.53 ± 8.26
	All	-0.21 ± 8.18	-0.16 ± 7.58	-0.48 ± 10.04	-0.07 ± 6.98	-0.52 ± 6.86	-0.72 ± 9.49
20×20	1	-0.50 ± 9.80	0.37 ± 9.69	-0.43 ± 8.62	-0.08 ± 7.96	-0.21 ± 7.80	-0.67 ± 7.79
	2	-0.38 ± 8.30	0.18 ± 9.20	-0.19 ± 7.18	-0.02 ± 7.13	-0.15 ± 8.12	-0.51 ± 6.77
	3	-0.63 ± 8.95	0.22 ± 9.46	-0.24 ± 8.94	-0.23 ± 7.53	-0.20 ± 8.67	-0.57 ± 8.35
	4	-0.81 ± 8.74	0.39 ± 8.17	-0.50 ± 8.36	-0.29 ± 7.33	-0.14 ± 7.28	-0.70 ± 7.55
	5	-0.39 ± 7.72	0.19 ± 7.72	-0.08 ± 9.29	-0.09 ± 6.86	-0.24 ± 6.80	-0.53 ± 8.74
	All	-0.54 ± 8.69	0.26 ± 8.84	-0.28 ± 8.56	-0.14 ± 7.35	-0.19 ± 7.74	-0.60 ± 7.93
Total		-0.24 ± 8.05	0.04 ± 7.94	-0.37 ± 10.50	-0.07 ± 6.86	-0.31 ± 7.10	-0.60 ± 9.92

https://doi.org/10.1017/S0263574718000577 Published online by Cambridge University Press

Table III. Mean and standard deviation of closure error (coordinate, direct distance and direct distance/total path length) for each set of five flight tests.

	Closure error ($\mu \pm \sigma$)			
Path size	X(m)	Y(m)	Direct (m)	Err/Len (%)
5×5	0.37 ± 0.45	-0.19 ± 0.41	0.63 ± 0.29	1.21 ± 0.37
10×10	0.50 ± 0.39	0.12 ± 0.13	0.55 ± 0.34	0.93 ± 0.62
15×15	0.94 ± 0.32	0.36 ± 0.50	1.11 ± 0.27	1.31 ± 0.33
20×20	1.71 ± 0.57	-0.08 ± 0.30	1.73 ± 0.56	1.65 ± 0.50
All	0.88 ± 0.67	0.05 ± 0.39	1.01 ± 0.60	1.28 ± 0.50

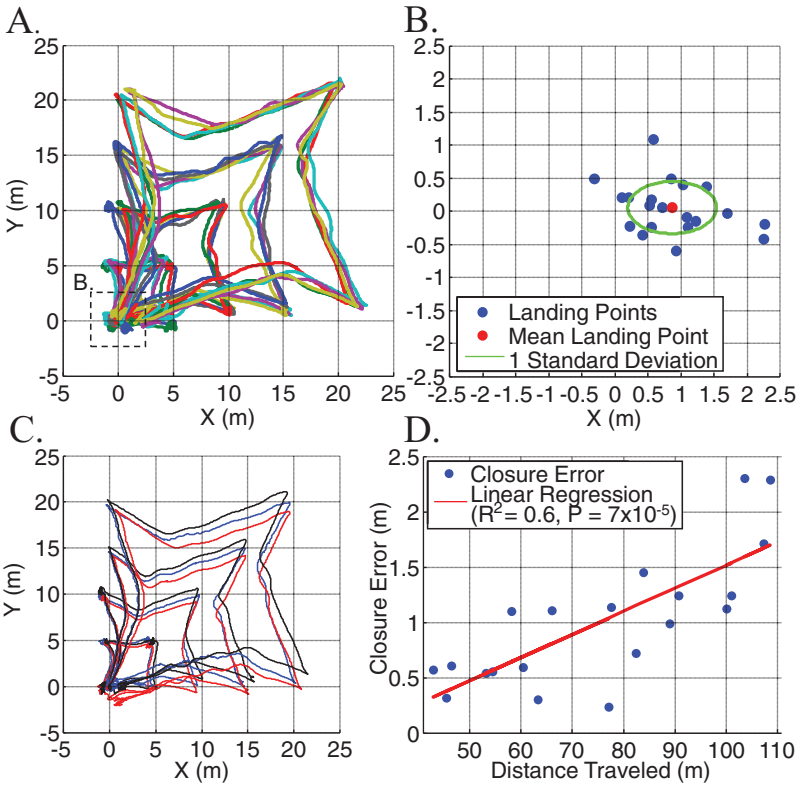


Fig. 11. (A) True flight path overlay; (B) Loop closure; (C) Estimate overlay (Black: DGPS, Blue: SC^3OF^3 (detailed in Sections 3.1 and 3.2), Red: OF^6 (originally described by Strydom *et al.*[24]); (D) Closure error vs total path length, including a linear regression.

mean landing point (shown in red) and an ellipse describing one standard deviation (drawn in green). The mean and standard deviation closure error for each set of five tests according to the ground truth data is recorded in Table III, which considers both the coordinate error and the direct distance error, as well as the percentage error when the direct distance is normalised by the total path length. The effect of path length is explored further in Fig. 11D, which illustrates that the direct distance closure error increases monotonically with the total path length. A linear regression performed on this data results in a fit ($m = 0.0209$, $b = -0.5744$) with an R^2 value of 0.6 and a P value of 7×10^{-5}. While it is difficult to theoretically predict exactly how the loop closure error should increase with travel distance without making assumptions about noise, etc. the linear regression provides an approximate estimate amounting to 2% of travel distance.

https://doi.org/10.1017/S0263574718000577 Published online by Cambridge University Press

5. Discussion

In this section, we present our analysis of the results obtained during flight-testing. We identify the limitations of the proposed algorithm and list some potential applications.

5.1. Analysis of results

The translation results listed in Table 4.1 indicate that the SC^{3}OF^3 method on average produces better estimates of translation for the Y (the vertical axis when considering Fig. 11A) and Z (down) directions, however, it is slightly inferior when considering the X direction (the horizontal axis of Fig. 11A). Another observation of note is that, for both methods, the error almost always corresponded to an underestimation of the translation in all directions. Finally, these values suggest that, while being quite accurate, the raw translatory outputs of both methods feature high levels of noise. This may be due to the small magnitudes of the optic flow vectors measured between successive frames at 25 Hz, which is an issue that can be removed without decreasing the frame rate through the use of a snapshot-based scheme, such as the system proposed by Denuelle and Srinivasan.[25] Note that Eq. (7) can be used to obtain a fair approximation of the instantaneous translation in m/s, using the assumed height during flight ($h_t = 3$ m) and known sample rate ($f = 5$ Hz). Using these figures, the order of magnitude of the translation error in meters per second is estimated to be approximately 0.001 ± 0.1 m/s.

While these points present interesting ramifications when considered in a navigational context, it is important to note that one standard deviation of the translation estimate error for both methods is within the known uncertainty of the ground truth (X, Z: ± 8.33, Y: ± 25 u_h/frame when scaled using an ideal height of 3 m). Thus, it is not reasonable to draw any definitive conclusions from the frame-to-frame data alone, other than that on a frame-to-frame scale, both methods are of a similar accuracy.

Analysis of the loop closure error suggests our method is superior (see Table III and Fig. 11B) with a $1.28 \pm 0.5\%$ error, compared to the mean error of 1.7% cited by Strydom *et al.*,[24] when considering the loop closure normalised by the total path length for all flight tests. Furthermore, this error is likely increased by the large spatial waypoint tolerance used during flight testing (0.5 m, as opposed to the 0.2 m used by Strydom *et al.*[24]). An additional comparison is provided in Table IV (modified from[34]), which ranks the average closure error of the two methods alongside existing visual odometry techniques.

The loop closure data described in Table III indicate a bias in the X direction (X: 0.88 ± 0.67 m vs. Y: 0.05 ± 0.39 m), suggesting that the flight platform repeatedly began its descent too early. This is likely caused by the integration error resulting from the systematic underestimation identified during the translation estimation analysis. In contrast, the small closure error in the Y direction further accentuates the high accuracy of the Sky Compass rotation estimation algorithm. The effect of integration error is explored further with Fig. 11D, which illustrates the relationship between closure error and distance travelled. A linear regression was performed on the data, producing an R^2 value of 0.6 and a P value of 7×10^{-5}, indicating that while the fit does not explain all of the variance, there is still a significant linear correlation. The effect of a positive linear relationship between closure error and distance travelled is that navigation over large distances becomes impractical without corrective measures, due to accumulation of the errors in the estimates of the frame-to-frame translations over time. This limitation is expected as it is a common feature of path integration. However, a potential solution is proposed below.

5.2. Limitations

As the proposed algorithm utilises the sky-based rotation estimation scheme outlined in our previous work,[26] it inherits all of the limitations described by that paper. The fundamental limitation outlined there is that the method requires an uninterrupted view of the sky. It was also identified that the presence of lens flares introduced significant error into the image-matching scheme. Another limitation observed in some tests was that rapid uniform movement of clouds (with cloud cover of more than $\sim$75%) produced an artificial rotation, however, extrapolation of the sky image based on cloud velocities was shown to provide a solution to this issue.

The key limitation of the work presented here is that the estimation of aircraft translation is frame-to-frame, resulting in the need for path integration. This is an issue when considering navigation over large distances, as it causes to accumulate over time. One solution would be to use a hybrid frame-

https://doi.org/10.1017/S0263574718000577 Published online by Cambridge University Press

Table IV. Comparison of loop closure error between existing visual odometry methods (white rows), OF^6 (red row, originally described by Strydom *et al.*[24]) and SC^3OF^3 (blue row, detailed in Sections 3.1 and 3.2).

Method	Platform	Path length	Number of tests	Average error (%)
SC^3OF^3	UAS	40 – 110 m	20	1.28±0.5
Feature-based stereo matching[35]	Ground vehicle	20 m	1	1.3
Feature-based monocular/ stereo matching[36]	Ground vehicle	186 m; 266 m; 365 m	3	1.4
OF^6 (24)	UAS	46 m; 50 m	15 of each path length	1.7
Maimone[37]	Ground vehicle	24 m; 29 m	2	2.0
Feature-based monocular/ stereo matching[38]	Ground vehicle	186 m; 266 m; 365 m	3	2.5
Monocular visual SLAM using SIFT features[39]	Ground vehicle	784 m; 2434 m	2	2.7
Optic flow[40]	UAS	94 m; 83 m	2	3.2
Template matching[41]	Ground vehicle	10 m; 20 m; 50 m; 100 m	8 of each path length	3.3
Stereo/monocular interest point matching SLAM[42]	UAS/Ground vehicle	100 m	2	3.5

to-frame and snapshot-based translation estimation scheme similar to that proposed by Denuelle and Srinivasan,[25] which computes egomotion using the optic flow field between the current frame and a particular reference frame (chosen from a series of reference snapshots). The advantage of snapshot-based navigation is that the drift caused by path integration is removed entirely.

Another potential limitation is the effect of large objects passing through the ground-cone. This would break the planar ground assumption, resulting in incorrect translation estimates. However, this will also affect any method that uses the ground plane assumption (e.g., ref. 24). While there is a robust outlier rejection scheme in place to deal with this contingency, it may be worth further segmenting the ground-cone to act as a safeguard by defining an additional confidence measure. It is important to note that this would only affect the velocity of the quadrotor temporarily (specifically causing the platform to slow down in response to measuring larger optic flow vectors while directly above the object), as the stereo height measurement uses a larger section of the image.

5.3. Accuracy, computational complexity and computational effort

Our approach has significant advantages over traditional algorithms that compute egomotion by estimating all 6DOF simultaneously. The advantages are: (1) Greater accuracy: The three rotation parameters are determined by using only the image of the sky – which is effectively at an infinite distance away – to obtain accurate results that are independent of, and uncontaminated by translation. The de-rotated image sequence, which now comprises only translational flow that is uncontaminated by rotational flow, is then analysed to obtain accurate estimates of the three translation parameters. (2) Reduced dimensionality: Reducing the 6DOF parameter estimation problem to two successive 3DOF parameter estimations mitigates against the 'curse of dimensionality'[43] by reducing the complexity of the requisite optimisations, and decreasing the likelihood of obtaining false solutions that do not correspond to the global optimum. Although we have not compared the computational costs of the two approaches quantitatively, published studies of the performance of the BOBYQA optimisation algorithm – which we have used in our study – indicate that doubling the dimensionality of the search space approximately doubles the number of function evaluations, increases the computational time by a factor of approximately 10, and increases the memory requirements by a factor of approximately

https://doi.org/10.1017/S0263574718000577 Published online by Cambridge University Press

3.5 (Table 1:[31]). It is therefore very likely that a 6DOF egomotion computation will be considerably slower and more resource-demanding than two 3DOF computations.

An alternative approach to measure egomotion in 6DOF would be to use rate gyro information to measure 3DOF rotation, and combine this with concurrently measured panoramic optic flow. The measured 3DOF rotation is then used to derotate the optic flow and obtain a residual pattern of optic flow that is purely due to the translational component of motion. This residual component of optic flow is then analysed to compute the translation in 3DOF.[8,44] Our approach adopts a similar structure, except that the rotation is measured visually by using the sky image as a reference. The advantage of measuring rotations visually is that they can always be compared to a standard reference image corresponding to a known orientation. Thus, our method avoids the cumulative errors that arise when integrating rate gyro information to compute attitude.

5.4. Advantages

One of the primary aims of this work was to investigate the validity of navigating using only the principles of visual processing that are adopted by insects. Our demonstration that a vision-only based solution can be used for egomotion computation validates existing theories of insect navigation, whereby it has been hypothesised that some insects use the sky for extracting compass information and their immediate surroundings for measuring translational motion[45] (our engineering approach uses an adaptation by limiting the immediately surrounding view to just the ground). The advantages of our system over existing vision-based navigation systems are:

1. Separating the computation of rotation and translation reduces computational complexity and increases estimation accuracy over methods that compute the two components of motion simultaneously.
2. The fact that the proposed navigation system relies purely on visual information means that it provides an independent backup for traditional systems that rely on GPS and IMU data.

Furthermore, our system retains the inherent additional advantages of vision-based navigation, conferring the capacity for surveillance, target detection, pursuit and obstacle avoidance.

A prevalent high-level navigation strategy for robotic systems that has not been thoroughly discussed here is simultaneous localisation and mapping (SLAM). It is important to emphasise that, unlike SLAM, we are (a) not tracking landmarks to monitor our progress, and (b) we are not trying to build a map. Reiterating, the work presented here addresses the fundamental problem of vision-only odometry and path integration. However, our technique for computing egomotion could be used as part of a system that performs SLAM.

As autonomous vehicles are required to deal with increasingly complex real world scenarios, it seems clear that vision systems such as ours will replace IMU, GPS and other sensors, or complement them to enhance navigational performance.

6. Conclusion

We have presented a pure vision-based approach to estimate egomotion, with extensions to visual odometry. The proposed algorithm first performs a rotation estimation using images of the sky. This is done by iteratively minimising the SAD between the current image and a reference image. The rotation corresponding to the minimum SAD is the rotation of the UAS. The optic flow field generated by the ground is then computed, and derotated using the sky-based rotation estimate to obtain residual flow vectors that represent the optic flow induced by the translatory component of UAS motion. An estimate of the UAS translation is then made by iteratively adjusting the three components of translation to minimise the error between the predicted translatory optic flow and the measured flow. The aircraft's translation vector, which is computed by assuming unit height above the ground, is then scaled to absolute units (meters per frame) using a stereo height measurement. The resulting frame-to-frame translational velocities are then used as a measure of visual odometry, enabling closed-loop navigation using path integration.

A series of flight tests were completed to evaluate and compare the performance of the new scheme with an existing visual odometry scheme, with respect to the estimation of instantaneous translational velocity, path integration, and closed-loop navigation. Square trajectories of differing sizes (5 × 5 m, 10 × 10 m, 15 × 15 m and 20 × 20 m) were traversed in closed loop at an altitude of 3 m, with each

https://doi.org/10.1017/S0263574718000577 Published online by Cambridge University Press

test repeated five times, resulting in twenty total flights. The translation estimate error was found to be in the order of 0.001 ± 0.1 m/s for both methods, when scaled by an assumed 3 m altitude (removing the effect of any error in height estimation during analysis). Odometric performance was evaluated using loop closure, recording an average of $1.28 \pm 0.5\%$ closure error when normalised by the total path length. These results suggest that the performance of the proposed scheme is comparable with or better than existing methods for measuring egomotion in 6DOF.

Future work will include incorporating a hybrid approach through the use of snapshots during the translation estimation process, to remove the drift associated with path integration.

Acknowledgements

The research described here was supported partly by Boeing Defence Australia Grant SMPBRT-11-044, ARC Linkage Grant LP130100483, ARC Discovery Grant DP140100896, and an ARC Distinguished Outstanding Researcher Award (DP140100914).

References

1. J. Rohac, "Accelerometers and an aircraft attitude evaluation," *Proceedings of the IEEE Sensors Conference*, Irvine, CA (2005) pp. 784–789.
2. S. Merhav, "Inertial rotation sensors," **In**: *Aerospace Sensor Systems and Applications* (Springer, New York, 1996) pp. 186–271.
3. R. Goecke, A. Asthana, N. Pettersson and L. Petersson, "Visual Vehicle Egomotion Estimation Using the Fourier–Mellin Transform," *Proceedings of the IEEE Intelligent Vehicles Symposium* (2007) pp. 450–455.
4. B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *Proceedings of the International Joint Conference on Artificial Intelligence* (1981) pp. 674–679.
5. M. V. Srinivasan, "An image-interpolation technique for the computation of optic flow and egomotion," *Biol. Cybern.* **71**(5), 401–415 (1994).
6. 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," *Int. J. Robot. Res.* **34**(8), 1114–1135 (2015).
7. D. Eynard, P. Vasseur, C. Demonceaux and V. Frémont, "Real time UAV altitude, attitude and motion estimation from hybrid stereovision," *Autonom. Robots* **33**, 157–172 (2012).
8. J. C. Zufferey, A. Beyeler and D. Floreano, "Optic flow to steer and avoid collisions in 3D," **In**: *Flying Insects and Robots* (Springer, Berlin, Heidelberg, 2009) pp. 73–86.
9. M. V. Srinivasan, R. J. D. Moore, S. Thurrowgood, D. Soccol and D. Bland, "From biology to engineering: Insect vision and applications to robotics," **In**: *Frontiers in Sensing* (Springer, Vienna, 2012) pp. 19–39.
10. S. Viollet and J. Zeil "Feed-forward and visual feedback control of head roll orientation in wasps (Polistes humilis, Vespidae, Hymenoptera)," *J. Exp. Biol.* **216**(7), 1280–1291 (2013).
11. R. Wehner, "Desert ant navigation: How miniature brains solve complex tasks," *J. Comp. Physiol.* **189**, 579–588 (2003).
12. D. Lambrinos, H. Kobayashi, R. Pfeifer, M. Maris and T. Labhart, "An adaptive agent navigating with a polarised light compass," *Adapt. Behav.* **6**, 131–161 (1997).
13. D. Lambrinos, R. Moller, T. Labhart, R. Pfeifer and R. Wehner "A mobile robot employing insect strategies for navigation," *Robot. Auton. Syst.* **30**, 39–64 (2000).
14. J. Chahl and A. Mizutani "Biomimetic attitude and orientation sensors," *IEEE Sens. J.* **12**(2), 289–297 (2012).
15. J. Dupeyroux, J. Diperi, M. Boyron, S. Viollet and J. Serres, "A Bio-Inspired Celestial Compass Applied to An Ant-Inspired Robot for Autonomous Navigation," *Proceedings of the European Conference on Mobile Robotics (ECMR)* (2017).
16. Z. Yang, J. Zhou, H. Huang, Y. Liu and Z. Li "Measuring solar vector with polarization sensors based on polarization pattern," *Optik* **141**, 147–156 (2017).
17. M. Wilson, "Functional organization of locust ocelli," *J. Comp. Physiol.* **124**, 297–316 (1978).
18. R. Berry, J. van Kleef and G. Stange, "The mapping of visual space by dragonfly lateral ocelli," *J. Comp. Physiol.* **193**, 495–513 (2007).
19. S. Thurrowgood, R. J. D. Moore, D. Bland, D. Soccol and M. V. Srinivasan, "UAV Attitude Control Using the Visual Horizon," *Proceedings of the Australasian Conference on Robotics and Automation*, Brisbane (2010).
20. R. J. D. Moore, S. Thurrowgood, D. Bland, D. Soccol and M. Srinivasan, "A fast and adaptive method for estimating UAV attitude from the visual horizon," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2011) pp. 4935–4940.
21. R. J. Moore, S. Thurrowgood, D. Soccol, D. Bland and M. V. Srinivasan, "A Method for the Visual Estimation and Control of 3–DOF Attitude for UAVs," *Proceedings of the Australasian Conference on Robotics and Automation*, Melbourne (2011).

https://doi.org/10.1017/S0263574718000577 Published online by Cambridge University Press

22. G. Loianno, M. Watterson and V. Kumar, "Visual inertial odometry for quadrotors on SE (3)," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (2016) pp. 1544–1551.
23. S. Thurrowgood, R. J. Moore, D. Soccol, M. Knight and M. V. Srinivasan, "A biologically inspired, vision based guidance system for automatic landing of a fixed–wing aircraft," *J. Field Robot.* **31**(4), 699–727 (2014).
24. R. Strydom, S. Thurrowgood and M. Srinivasan, "Visual Odometry: Autonomous UAV Navigation using Optic Flow and Stereo," *Proceedings of the Australasian Conference on Robotics and Automation*, Melbourne (2014).
25. A. Denuelle and M. V. Srinivasan, "Snapshot–Based Navigation for the Guidance of uas," *Proceedings of the Australasian Conference on Robotics and Automation*, Canberra (2015).
26. T. Jouir, R. Strydom and M. V. Srinivasan, "A 3d Sky Compass to Achieve Robust Estimation of UAV Attitude," *Proceedings of the Australasian Conference on Robotics and Automation*, Canberra (2015).
27. J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(8), 1335–1340 (2006).
28. J. Zeil, M. I. Hofmann and J. S. Chahl, "Catchment areas of panoramic snapshots in outdoor scenes," *J. Opt. Soc. Am.* **20**(3), 450–469 (2003).
29. J. Campbell, R. Sukthankar, I. Nourbakhsh and A. Pahwa, "A Robust Visual Odometry and Precipice Detection System using Consumer–Grade Monocular Vision," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (2005) pp. 3421–3427
30. S. G. Johnson, "The NLopt nonlinear–optimization package," (2014).
31. M. J. Powell, "The BOBYQA algorithm for bound constrained optimization without derivatives," *Cambridge NA Report NA2009/06*, (University of Cambridge, Cambridge, 2009) pp. 26–46.
32. M. Shimizu and M. Okutomi, "Significance and attributes of subpixel estimation on area–based matching," *Syst. Comput. Japan* **34**(12), 1–10 (2003).
33. A. Denuelle, R. Strydom and M. V. Srinivasan, "Snapshot-Based Control of UAS Hover in Outdoor Environments," *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO)* (2015) pp. 1278–1284.
34. R. Strydom, A. Denuelle and M. V. Srinivasan, "Bio-inspired principles applied to the guidance, navigation and control of UAS," *Aerospace* **3**(3), 21 (2016).
35. C. F. Olson, L. H. Matthies, M. Schoppers and M. W. Maimone, "Robust Stereo Ego-Motion for Long Distance Navigation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Hilton Head Island, SC, USA, vol. 2 (2000) pp. 453–458
36. D. Nistér, O. Naroditsky and J. Bergen, "Visual Odometry," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, vol. 1 (2004) pp. 652–659
37. M. Maimone, Y. Cheng and L. Matthies, "Two years of visual odometry on the mars exploration rovers," *J. Field Robot.* **24**, 169–186 (2007).
38. D. Nistér, O. Naroditsky and J. Bergen, "Visual odometry for ground vehicle applications," *J. Field Robot.* **23**, 3–20 (2006).
39. J. P. Tardif, Y. Pavlidis and K. Daniilidis, "Monocular Visual Odometry in Urban Environments Using an Omnidirectional Camera," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France (2008) pp. 2531–2538
40. L. Heng, D. Honegger, G. H. Lee, L. Meier, P. Tanskanen, F. Fraundorfer and M. Pollefeys, "Autonomous visual mapping and exploration with a micro aerial vehicle," *J. Field Robot.* **31**, 654–675 (2014).
41. N. Nourani-Vatani and P. V. K. Borges, "Correlation–based visual odometry for ground vehicles," *J. Field Robot.* **28**, 742–768 (2011).
42. T. Lemaire, C. Berger, I. K. Jung and S. Lacroix, "Vision-based slam: Stereo and monocular approaches," *Int. J. Comput. Vis.* **74**, 343–364 (2007).
43. R. Bellman, "*Adaptive Control Processes: A Guided Tour*" (Princeton University Press, Princeton, NJ, 2015).
44. M. V. Srinivasan, S. Thurrowgood and D. Soccol, "From Visual Guidance in Flying Insects to Autonomous Aerial Vehicles," *Flying Insects and Robots* (Springer, Berlin, Heidelberg, 2009) pp. 15–28.
45. M. V. Srinivasan, "Honeybees as a model for the study of visually guided flight, navigation, and biologically inspired robotics," *Physiol. Rev.* **91**, 389–411 (2011).

https://doi.org/10.1017/S0263574718000577 Published online by Cambridge University Press