Robotica (2018) volume 36, pp. 1225–1243. © Cambridge University Press 2018
doi:10.1017/S0263574718000358

A survey on stereo vision-based autonomous navigation for multi-rotor MUAVs

Jose-Pablo Sanchez-Rodriguez* and Alejandro Aceves-Lopez

Tecnologico de Monterrey, Atizapán de Zaragoza, Estado de México, Z.C. 52926, México.
E-mail: aaceves@itesm.mx

(Accepted April 7, 2018. First published online: May 6, 2018)

SUMMARY

This paper presents an overview of the most recent vision-based multi-rotor micro unmanned aerial vehicles (MUAVs) intended for autonomous navigation using a stereoscopic camera. Drone operation is difficult because pilots need the expertise to fly the drones. Pilots have a limited field of view, and unfortunate situations, such as loss of line of sight or collision with objects such as wires and branches, can happen. Autonomous navigation is an even more difficult challenge than remote control navigation because the drones must make decisions on their own in real time and simultaneously build maps of their surroundings if none is available. Moreover, MUAVs are limited in terms of useful payload capability and energy consumption. Therefore, a drone must be equipped with small sensors, and it must carry low weight. In addition, a drone requires a sufficiently powerful onboard computer so that it can understand its surroundings and navigate accordingly to achieve its goal safely. A stereoscopic camera is considered a suitable sensor because of its three-dimensional (3D) capabilities. Hence, a drone can perform vision-based navigation through object recognition and self-localise inside a map if one is available; otherwise, its autonomous navigation creates a simultaneous localisation and mapping problem.

KEYWORDS: Autonomous navigation; Computer vision; Multi-rotor micro unmanned aerial vehicle (MUAV); Simultaneous localisation and mapping (SLAM); Stereoscopic vision.

1. Introduction

Recently, consumers and researchers have exhibited growing interest in multi-rotor micro unmanned aerial vehicles (MUAVs) such as quad-copters and hex-copters because of the many possible applications of MUAVs—from cinema and other entertainment to surveillance and search and rescue.[1–3] The attractiveness of these vehicles comes from their small size, relatively low price and capabilities for accessing zones unreachable to humans. A great deal of research has focussed on the capability of MUAVs to navigate autonomously and to perform mapping without any human-in-the-loop control while avoiding obstacles. Thus, the successful autonomous navigation systems will lead to more robust, reliable and safer flights of MUAVs in indoor-outdoor applications.

The motion of ready-to-fly drones relies mainly on the ability of a human-in-the-loop operator. For some applications, this kind of device is enough, but for more autonomous applications, the human-in-the-loop becomes a restrictive factor because of the loss of line of sight, which can lead to collisions with objects such as wires and branches. Therefore, the human-in-the-loop must be substituted with a computer to overcome these drawbacks and to enhance the systems and move them in the direction of autonomous MUAVs. To accomplish this, the questions 'Where am I?' and 'Where do I have to go?' must be solved repeatedly in real time. These questions are solved by interpreting the information retrieved by sensors like global positioning system (GPS),[4–6] lasers[7–12] and ultrasonic sensors.[13–16] Currently, one of the most suitable sensors is the stereo camera because of its small size and low

* Corresponding author. E-mail: pablo270991@gmail.com

https://doi.org/10.1017/S0263574718000358 Published online by Cambridge University Press

weight, as well as algorithms that permit three-dimensional (3D) scene interpretation that can run in real time on small-sized computers. The position of the MUAV in a map can be estimated from the information retrieved by a camera. This, in addition to navigation planning, provides the necessary information to produce the control actions that allow the MUAV to fly.

Currently, there exist many successful implementations of vision-based autonomous navigation for ground vehicles,[17–19] humanoid robots,[20–22] underwater robots[23,24] and small UAVs[25–27] in which payload size is not a big issue. Unfortunately, for MUAVs, useful payload and energy consumption are limited. Still, modern vision-based systems are suitable for MUAV navigation because of their small size, low weight and 3D perception capabilities.

We present a classification of vision-based systems according to the number of cameras used. The choice of the number of cameras presents important advantages and disadvantages. Single-camera systems are simple because only one camera is used. Because of their dimensions, they can be mounted on board MUAVs; nevertheless, these systems need a sequence of images to estimate the extrinsic parameters of each of the elements of the sequence. The extrinsic parameters are then used to localise features and to auto-localise the vehicle in the environment. Multi-camera systems achieve 3D perception more easily by taking multiple photographs simultaneously, and with known extrinsic parameters, the system can localise the features in the 3D space. These systems are more complex, however, they need confined places, and most of the systems are fixed on ground. Moreover, a high-performance computer is needed to process all the information in real time, but it cannot be mounted on board. The stereoscopic configuration has some of the benefits of single-and multi-camera systems. This configuration is almost as small and low weight as a single-camera system and permits 3D perception using only one pair of stereoscopic images. Currently, stereoscopic cameras are popular, and there are successful implementations. This explains why they are considered the most suitable devices for MUAVs.

Path-planning algorithms are suitable only if a map is always available and the goal state and the robot's current state are known. But when a map is not available, a simultaneous localisation and mapping (SLAM) algorithm is needed.[28,29] Its purpose is to build a map of an unknown environment using a mobile robot (e.g., MUAV) while navigating. Many vision-based SLAM approaches, such as extended Kalman filter (EKF) SLAM, graph SLAM, fast SLAM and sparse extended information filter SLAM,[30] use images features and relative projections to gather navigation information.

In this paper, we present the topics addressed by state-of-the-art MAUVs for which the objective is to navigate autonomously using a stereoscopic camera. We suggest referring to the cited works to gain a complete understanding of the topics. The text is organised into five sections. Section 2 provides an overview of MUAV navigation. Section 3 presents vision systems for MUAVs in terms of single-, multi- and stereoscopic cameras. In Section 4, stereoscopic vision-based approaches are addressed. A discussion on stereoscopic vision-based localisation systems is presented in Section 5.

2. Background on Robot Navigation

Navigation solves the problem of reaching a goal state, and to do this, we must consider robot motion. Motion models give information on how a robot pose changes as a function of control inputs and time. In general, for MUAVs (multi-rotor kind), there are two ways to address motion models: the complete dynamic model and constant velocity, constant angular velocity model. The complete dynamic model of the MUAV is given in Eqs. (1) and (2). For these, $U_t = (\mathcal{F}_i\, M_i)$ denotes the control at time t, where $\mathcal{F}_i$ is the thrust and M_i the momentum produced by each propeller and motor with $i = (1, 2, 3, 4)$ in the case of a quad-rotor.

$$\mathrm{m}\ddot{\mathbf{r}} = \begin{bmatrix} 0 \\ 0 \\ -\mathrm{m}g \end{bmatrix} + \mathrm{R} \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^{n=4} \mathcal{F}_i \end{bmatrix} \tag{1}$$

$$\mathbf{I} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} L(\mathcal{F}_2 - \mathcal{F}_4) \\ L(\mathcal{F}_3 - \mathcal{F}_1) \\ M_1 - M_2 + M_3 - M_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \mathbf{I} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2}$$

https://doi.org/10.1017/S0263574718000358 Published online by Cambridge University Press

Table I Robot navigation algorithms comparison.[33–36]

	Optimal	Complete	Query	Repeatable	Obstacles
Braitenberg	Yes[1]	No	Reactive	Yes	Any
Bugs	No	Yes	Reactive	Yes	Any
Distance Transform	No	Depends on resolution	Single	Yes	Grid
D*	No	Depends on resolution	Single	Adaptable	Grid
Voronoi Road map[2]	No	Yes	Multiple	Yes	Polygons
PRM	Probabilistically	Probabilistically	Multiple	No	Any
RRT	No	Probabilistically	Single	No	Any

[1]It is optimal in the sense that if the agent faces a concave obstacle, the algorithm will stop at that point (at a minimum).
[2]It is the most common method for obtaining a topological map.

In Eq. (1), $\ddot{\mathbf{r}}$ is the lineal acceleration vector, and R the rotation matrix from the MUAV body frame to the inertial frame. In Eq. (2), $\mathbf{I}$ is the inertia matrix, $[\ \dot{p}\ \dot{q}\ \dot{r}\ ]^{\mathrm{T}}$ the vector of angular accelerations, and L the length of the vehicle's arm.

On the other hand, the constant velocity, constant angular velocity model relies on an internal controller. This model is simpler and permits an omni-directional-like motion (assuming near-hover position). In Eq. (3), $U_t = (\mathcal{V}\ \omega)^{\mathrm{T}}$ denotes the control at time t, where $\mathcal{V}$ and ω are linear and angular velocities with assumed zero-mean and Gaussian distribution noise $\mathbf{n} = (\mathtt{v}\ \mathtt{w})^{\mathrm{T}}$.[31]

$$\begin{bmatrix} \chi_t \\ q_t \\ \mathcal{V}_t \\ \omega_t \end{bmatrix} = \begin{bmatrix} \chi_{t-1} + (\mathcal{V} + \mathtt{v})\Delta t \\ q_{t-1} \times q((\omega + \mathtt{w})\Delta t) \\ \mathcal{V} + \mathtt{v} \\ \omega + \mathtt{w} \end{bmatrix} \tag{3}$$

In the state vector, χ_t is position and q_t the rotation quaternion at time t.

Motion models permit the estimation of the pose of a robot for navigation. In general, there are two types of navigation for mobile robots (besides MUAVs): reactive and map-based.[32] Table I shows a comparison of some of the most widely used navigation algorithms for vehicles.

2.1. Reactive navigation

Reactive navigation robots are goal-oriented machines that sense and act only according to the environment conditions. There is a direct connection between the sensors and the motors. Examples are Braitenberg vehicles,[32] which operate without any knowledge of the environment. That is why they do not perform high-level tasks; they instead execute logical behaviours. Even though reactive navigation is a basic method in robotics, it is being used also for MUAV control and obstacle avoidance, as in refs. [37–39].

2.2. Map-based navigation

Map-based navigation is for achieving path-to-goal with *a priori* knowledge of the environment. Map-based navigation involves two phases: the planning phase (when the path is created) and the query phase (when a path to a goal is executed). Hence, the map must be provided to the robot, or the agent must explore the environment and build a map to make navigation possible for the robot.

One of the most used tools to build a map is the occupancy grid. This is a matrix that contains a discretised representation of the observed world classified into cells that are free, occupied (robot, obstacles) or unexplored. Usually, the cell state value is 'zero' if it is a drivable space, 'one' if there is an obstacle or another number to represent unknown zones. This representation considers the vehicle itself as an occupied space; subsequently, it is useful for multi-MUAV navigation because each MUAV would be aware of the others. The size of the grid must be big enough to capture the size of the real environment. The bigger the environment, the bigger would be the grid. Moreover, the dimension and the complexity of the grid increase when dealing with 3D spaces.

https://doi.org/10.1017/S0263574718000358 Published online by Cambridge University Press

Another way to represent the environment is through topological maps, which are more efficient than grid-only maps.[40] Topological maps depend on grid maps and are built within them. To create topological maps, the drivable space is divided into regions of space. These regions are represented by nodes in a graph, and the connection paths to other regions are represented as edges. Graph representation permits a more efficient implementation of classical navigation methods because the planning phase is done symbolically, and the query phase would follow the nodes on a stack until reaching the goal. Some examples related to MUAVs are the distance transform method, as in ref. [41], D*; the Voronoi roadmap method; the probabilistic roadmap method, as in ref. [42]; and RRT, as in ref. [32]. Nevertheless, one drawback of topological navigation is the difficulty of differentiating between similar places.[40]

A basic requirement for performing autonomous navigation is knowing the current position of the robot in the map. That is why robots need sensors for self-localisation. The process of gathering information through the sensor and estimating the position of the robot in the map is known as localisation.

2.3. Localisation

Localisation is defined as the problem of estimating the pose by finding correspondence between the environment and the robot. This task is not always easy because of uncertainties in the readings. Some of the readings may be equivalent in different zones of the map, thus, leading to incorrect estimations. Therefore, some inferences must be made from the sensor data. There are two types of localisation: robot localisation and features localisation. The first one consists of estimating the position of the robot, and the second one consists of localising features relative to the robot pose.

The easiest way to localise a robot in a map is by means of a GPS, but this method produces poor results indoors and near to buildings. An alternative way of localising a robot is to use inertial measurement units (IMU) to estimate the position of the vehicle by considering its dynamics.

Walls and corners are some of the easiest features to localise with sensors.To estimate the position of these features in the environment, we first need to detect lines and points in the image and infer the position of these features. Therefore, the following homogeneous transformation is needed:

$$\mathbf{x} = P\mathbf{X} \tag{4}$$

where

$$P = K[R|t]$$

$$K = \begin{bmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

and $\mathbf{x}$ is a 3×1 matrix ($(u, v, 1)^{\mathrm{T}} \in \mathcal{P}(\mathbb{R}^3)$) of the point in the image in homogeneous coordinates of the projective space, $\mathbf{X}$ is a 4×1 matrix ($(X, Y, Z, 1)^{\mathrm{T}} \in \mathbb{R}^3$) of the point in the world in homogeneous coordinates and P is a 3×4 matrix of the camera projection matrix that transforms from the real-world space (3D) to the projective space (2D). In (5), K is a 3×3 matrix of the intrinsic calibration parameters, where f_x and f_y are the focal lengths, s is the skew coefficient that models the lens distortion and x_0 and y_0 are the coordinates of the camera centre. R is a 3×3 rotation matrix $\in SO(3)$ and t is a 3×1 translation matrix ($(x, y, z)^{\mathrm{T}} \in \mathbb{R}^3$), which are also called the extrinsic calibration matrices.

We need to calibrate the cameras to use them for localisation. This means that we must estimate the focal length, optical centre and distortion coefficients in matrix K. The calibration also permits the removal of lens distortion so that the resultant image represents a realistic perspective. Extrinsic calibration is the process of estimating the rotation and translation of the camera frame from one shot to another.

In general, for vision-based systems, the process of using images for localisation is based on feature detection and pairing. Features are those salient and easily detectable points for which the position in the real world is useful to localise the robot and to build a map. Consecutive images contain many

https://doi.org/10.1017/S0263574718000358 Published online by Cambridge University Press

Table II. Typical vision-based systems for MUAVs.

	Single-camera	Multi-camera	Stereo-camera
Place	Onboard	Offboard	Onboard
No. cameras	1	≥3	2
Shots	Multiple	1 each camera	1 pair
Epipolar geometry	Continuous extrinsic calibration	Fixed extrinsic calibration	Continuous extrinsic calibration.[1]
Advantages	Low weight	3D mapping	Depth mapping
Processing	1 image @ time	many images @ time	2 images @ time

[1]The relative calibration between the left and right sensors of the stereo camera is fixed.

common features for performing a pairing algorithm. These algorithms detect the same features on consecutive images, and through this process, the rotation matrix and translation vector from the point of view of the camera can be estimated. Some feature extraction algorithms are Harris,[43] multi-scale-oriented patches (MOPS),[44] scale-invariant feature transform (SIFT)[45] and speeded-up robust features (SURF).[46] For the pairing task, some of the most common algorithms are random sample consensus (RANSAC),[47] iterative closest point,[48] robust point matching,[49] kernel correlation[50] and coherent point drift.[51] The association stage permits the localisation of these features in the map (and, therefore, the robot).

3. Vision for MUAVs

Vision-based navigation has been studied for decades and has been applied in many robots, including MUAVs. After a review of the most representative papers on vision-based navigation for MUAVs, we classified them into three groups (see Table II): single, multi- and stereo cameras. It is been evident that multi-camera systems (especially stereo) have been significantly more popular than single-camera approaches[52] (see Fig. 1). Therefore, in the following paragraphs, the characteristics of single cameras are presented first to subsequently highlight the benefits of multi-camera systems.

3.1. Single-camera

The first works on vision-based MUAVs used single-cameras. For pose estimation, these systems extract information from the scene using extrinsic calibration recursively. The extrinsic calibration uses a sequence of consecutive images from the same scene taken from multiple perspectives. After detecting and pairing the common features, they must satisfy the fundamental matrix equation

$$\xi'^{\mathrm{T}} F \xi = 0 \tag{6}$$

and the homography matrix

$$\xi' = H\xi \tag{7}$$

where F is a 3×3 rank 2 matrix called the fundamental matrix. It is the mathematical representation of the epipolar geometry constrained by the corresponding points lying on the same plane. H is a 3×3 full rank matrix called the homography transformation that describes the transformation of a set of projected points $\xi = \{\mathbf{x}_1, \mathbf{x}_2, \ldots\}$ from one view to the set $\xi' = \{\mathbf{x}'_1, \mathbf{x}'_2, \ldots\}$ in the other view, where $\mathbf{x}'_i$ and $\mathbf{x}_i$ are the projected points in two images. The method presupposes that the projected points have been previously paired and accordingly indexed. It has been proved that the number of points must be at least eight.[53]

As with F, the essential matrix E describes the correspondence between two normalised image coordinates

$$\hat{\xi}'^{\mathrm{T}} E \hat{\xi} = 0 \tag{8}$$

https://doi.org/10.1017/S0263574718000358 Published online by Cambridge University Press

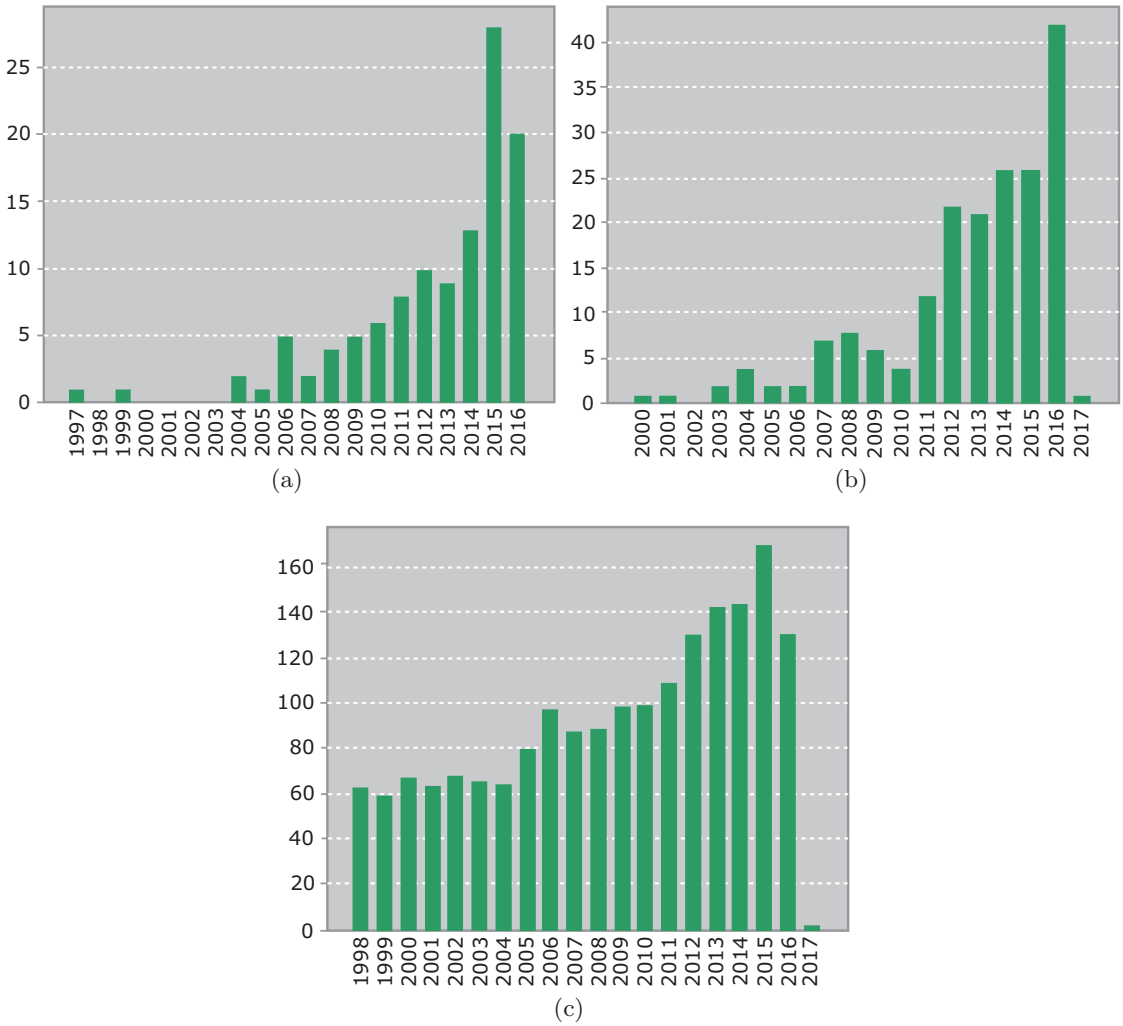


Fig. 1. Published items in each year. (a) Search Criterion: UAV (AND) monocular vision. (b) Search Criterion: UAV (AND) multi vision. (c) Search Criterion: UAV (AND) stereoscopic vision. From Web Of Science[52]

where

$$\hat{\xi} = K^{-1}\xi \tag{9}$$

This matrix also describes the rotation R and translation t from one camera to the other in the form

$$E = [t]_{\times}R \tag{10}$$

Moreover, there exists a direct relation between the F and E matrices

$$E = K^{\mathrm{T}}FK \tag{11}$$

From (10), $[t]_{\times}$ is the translation of the camera in skew-symmetric-matrix form, and R is its rotation from one pose to the next. With this information, the correspondence of the scene in 3D can be estimated. Some of the most widely used techniques in MUAVs are points triangulation,[54] perspective-n-point algorithm[55] and structure from motion.[56] Some representative examples are presented by Lee *et al.*,[57] Ferworn *et al.*[58] and Omari *et al.*[59]

https://doi.org/10.1017/S0263574718000358 Published online by Cambridge University Press

Perspective-n-point is a technique for finding the location of a point in the world space $\mathbf{X}_i$ given a point projection $\mathbf{x}_i^j$

$$\begin{bmatrix} [\mathbf{x}_i^1]_\times P^1 \\ \vdots \\ [\mathbf{x}_i^n]_\times P^n \end{bmatrix} \mathbf{X}_i = 0 \tag{12}$$

This must be solved for each of the projected points $\mathbf{x}_i^j$, where $i = (1, 2, \ldots)$ refers to the index of the detected feature, and $j = (1, 2, \ldots, n)$ to the index of the camera perspective. Projected points are provided by the MUAV vision system at different positions described in P^j. This equation must be solved for $\mathbf{X}_i$, for instance, via single value decomposition.

Some studies have used this technique applied to MUAVs. Lee *et al.*[57] presented a system to detect features combining MOPS (to detect corners) and SIFT (to detect central zones). Similarly, in Carloni's paper,[60] obstacles were detected and discretised on a 3D grid map giving information about which of the detected obstacles represented a threat for the mission.

Another well-known method used for estimating and enhancing the calibration parameters and the 3D points is bundle adjustment (BA). This method is desirable for dealing with the noise encountered in measurements. Therefore, Eq. (4) is not satisfied exactly. Hence, this method considers the re-projection of the estimated point $\mathbf{X}_i$ back to the corresponding images.

$$\mathbf{x}_i^j + \hat{\mathbf{v}}_i^j = \hat{\lambda}_i^j \hat{P}^j \hat{\mathbf{X}}_i \tag{13}$$

In (13), $\hat{\mathbf{v}}_i^j$ is the correction parameter, $\hat{\mathbf{X}}_i$ is the point i in 3D space, $\hat{P}^j$ is the projection matrix of the jth view and $\hat{\lambda}_i^j$ is a scaling parameter. This method adjusts these parameters such that the difference between re-projected and originally detected points in the image plane can be minimised

$$\min_{\hat{P}^j, \hat{\mathbf{X}}_i} \sum_{ij} d(\hat{P}^j \hat{\mathbf{X}}_i, \mathbf{x}_i^j)^2 \tag{14}$$

where d is the difference between detected and re-projected points. BA considers all calibration parameters, detected points and uncertainties (assuming Gaussian distributions).[54]

3.2. Multi-camera

The studies addressing multi-vision used cameras in different layouts. They could be fixed on board, fixed on ground or fixed on another vehicle (e.g., on-ground robot). They could also be a set of cameras, with each one fixed on a MUAV. Using a multi-vision system, the MUAV takes many pictures simultaneously from different perspectives. Equation (12) must be solved after feature detection in each picture.

Multi-camera systems are used to obtain reliable depth maps, such as in the work of Harmat and Sharf[61] who used a laser pattern projected into objects to produce easily detectable features and to estimate depth (active vision). Their system looked only for discontinuities of the laser pattern in images. They tested their method by using only one camera, then two cameras up to six cameras. Their results showed that the depth error decreases proportionally to the number of cameras used.

A commonly used system for MUAVs is motion capture (MoCAP). It uses many cameras fixed in a room to observe the MUAV movements precisely. These systems are intended for use in laboratories. In ref. [62], the author presented a team of quadrotors sharing the same space, avoiding collisions with one another and navigating according to a generated trajectory in a complex environment with obstacles. This was done with a precise tracking system at a high rate (e.g., 100 Hz), possible with high-performance tools (MoCAP and computer on ground) only. Similarly, the studies[59,63–65] used MoCAP for precise MUAV tracking. Another variant for localisation using MoCAP was presented by Kwon[64] and Charrow.[65] Both used MUAVs forming a stereoscopic rig with a variable baseline (distance between cameras' centres of the MUAVs (see Fig. 2)). The MUAVs generated a depth map to help an unmanned ground vehicle (UGV) avoid obstacles and reach its goal position.

Although MoCAP systems retrieve very precise tracking information of the MUAV, they are not suitable outdoors because the multi-camera system is fixed to a specific location. Hence, the

https://doi.org/10.1017/S0263574718000358 Published online by Cambridge University Press

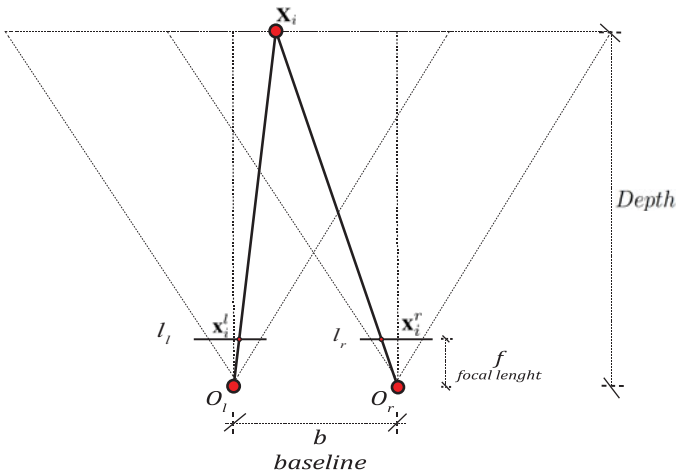


Fig. 2. Principle of stereo vision.

stereoscopic camera is proposed as a suitable sensor for estimating 3D data. This type of camera has become very popular and is increasingly used for MUAV navigation because of its small dimensions and 3D perception capabilities.

4. Stereoscopic Camera Systems

Table III presents a list of surveyed stereoscopic vision-based MUAVs. The aim of this table is to summarise the most recent approaches on autonomous navigation via stereo vision. It shows authors, stereo configuration, application, mapping, environment and method of autonomous navigation.

4.1. Fundamentals

A stereoscopic configuration has two parallel cameras (if they are parallel, they are coplanar) separated by a distance b (called baseline). Figure 2 shows the principle of stereo vision. The feature $\mathbf{X}_i$ in the world produces two points in the stereoscopic images $\mathbf{x}_i^l$ and $\mathbf{x}_i^r$ lying on the image planes l_l and l_r. The points labeled O_l and O_r are the camera centres, and f is the focal length. The relationship between the projected corresponding points in the left and right images $\mathbf{x}_i^l \longleftrightarrow \mathbf{x}_i^r$, and the world point in 3D $\mathbf{X}_i$ is

$$\begin{aligned} \mathbf{x}_i^l &= P^l \mathbf{X}_i \\ \mathbf{x}_i^r &= P^r \mathbf{X}_i \end{aligned} \tag{15}$$

where P^l and P^r are the cameras' matrices, the scripts l and r correspond to the left and right images accordingly and subscript $i = (1, 2, \ldots, n)$ to the detected point. As with single camera, $\mathbf{x}_i^l$ and $\mathbf{x}_i^r$ must satisfy (6) and (7) from the same point $\mathbf{X}_i$. They are also constrained by the epipoles located at infinity, parallel epipolar planes and correspondent points lying on the same epipolar line x $\longrightarrow$ l'. Therefore, the cameras' matrices are as follows:

$$\begin{aligned} P^l &= K^l[I|0] \\ P^r &= K^r[I|t] \end{aligned} \tag{16}$$

4.2. Depth estimation

A stereoscopic configuration is intended for depth perception like that of human vision.[18] Humans can move through environments and avoid collisions with obstacles (people, walls or any object) by interpreting the depth. In robotics, researchers have tried to replicate this capability by using a stereo camera for navigation. With a stereo camera, the features in the images are in the same epipolar plane

https://doi.org/10.1017/S0263574718000358 Published online by Cambridge University Press

Table III. Surveyed works.

Author	Stereo-camera +other sensors	Depth	Obstacle avoidance	Proposed solution	Autonomous navigation
Garcia Carrillo *et al.*[66] (2012)	Stereo-camera + IMU	Sparse	Feature tracking TLK[67]	Vision and intertial odometry fusion with Kalman filter. State estimation via strap down algorithm (SDA)	Yes
Schneider *et al.*[68] (2012)	1 Stereo- camera facing forwards 1 Stereo-camera facing backwards +IMU +Ultrasonic +LiDAR	Sparse	No	Keyframe-based visual SLAM with incremental bundle adjustment	No
Thamrin *et al.*[25] (2012)	Stereo-camera	Sparse	Network free space bubbles[69]	Real-time SLAM: Trees as landmarks. Row detection by virtual point projection (VPP)	No
Tomic *et al.*[70] (2012)	Stereo-camera +LiDAR	Dense	Waypoints	SLAM	Yes: Under controlled conditions
Griesbach, Baumbach and Zuev[71] (2014)	Stereo-camera + IMU	Dense	No	Egomotion estimation. Visual odometry and inertial measurements fused in Kalman filter	No
Kwon, Seo and Kim[64] (2014)	Multi-UAV Movable baseline + MoCAP	Dense	Multi-UAV obstacle detection	Object detection behind obstacles for UGVs from UAVs	Yes under lab. conditions
Schauwecker and Zell[72] (2014)	1 Stereo- camera facing forwards 1 Stereo-camera facing downwards	Sparse	Stereo matching	SLAM	Yes
Schmid *et al.*[73] (2014)	Stereo-camera + IMU	Dense	Via Dijkstra's re-planning	Dijkstra's algorithm for path planning Sensor fusion, vision and inertial	Yes
Cook[42] (2015)	Stereo- thermal camera + LiDAR	Dense	Roadmap	Voxel obstacle avoidance	Yes: Computer simulation
Fu, Carrillo and Campoy[74] (2015)	Stereo-camera	Dense	Detection via OctoMap	Visual odometry and mapping	No
Omari *et al.*[59] (2015)	Stereo-camera + MoCAP	Dense and Sparse	No	MoCAP robust tracking	No
Park, Kim, and Kim[41] (2015)	Stereo-camera	Dense	For future work	Distance transform	No

https://doi.org/10.1017/S0263574718000358 Published online by Cambridge University Press

but have different image coordinates. With computer vision, the depth is calculated from the image disparity, which is the displacement between the location of the same feature in the stereo-images.

Disparity is the difference between the projection coordinates of the same point from the real world

$$d = u_i^l - u_i^r, \tag{17}$$

where u_i^l and u_i^r are the horizontal locations of the feature in the left and right images. Along with the focal length and baseline, depth can be estimated.

$$\text{Depth} = f\frac{b}{d} \tag{18}$$

First, the stereo correspondence must be solved to compute the disparity by finding the projection coordinates. As described in (6), the stereoscopic configuration constrains the correspondent points to lie on the same epipolar line; hence, the search algorithms reduce their exploration zone over such a line. There are two types of stereo-matching algorithms: dense (area-based algorithms, full-depth) and sparse (feature-based, partial depth).[75]

Dense correspondence algorithms look for the matching of every pixel and its surroundings from the left image in the right image. This search process uses a squared mask window with odd dimensions and its centre placed over the pixel of interest. This mask window is then convoluted with same-sized blocks of pixels (templates) from the left image to evaluate the zone that surrounds the pixel of interest. Once a zone has been evaluated in the left image, the algorithm looks for the corresponding zone in the right image by convolving the mask window with candidate templates over the epipolar line. When searching for the stereo matching, an image-similarity measure is used to evaluate the candidate disparity and to determine whether both left and right blocks are correspondents. One of the most used image-similarity measures is the zero mean normalised cross-correlation.[76] It computes a correlation coefficient r that, compared to a threshold, indicates whether it is a match

$$r(d_u, d_v) = \frac{\Sigma_{(u,v)\in S}[A][B]}{\sqrt{\Sigma_{(u,v)\in S}[A]^2 \Sigma_{(u,v)\in S}[B]^2}} \tag{19}$$

where

$$\begin{aligned} A &= f^l(u, v) - \bar{f}^l \\ B &= f^r(u + d_u, v + d_v) - \bar{f}^r \end{aligned} \tag{20}$$

In Eqs. (19) and (20), S is the image space, f^l and f^r are a pair of blocks from the stereo images, $\bar{f}^l$ and $\bar{f}^r$ are the average pixel intensities of such candidate blocks, d_u and d_v are the disparities and u, v are the coordinates of the block centre on its corresponding image. Disparities are saved accord to the coordinates, where they were calculated into a matrix, the disparity map. Equation (18) is then applied to the complete map to transform it into a depth map.

For MUAV navigation, dense correspondence algorithms have been used to find obstacles or to build maps like those of studies by Tomic *et al.*,[70] Griesbach *et al.*,[71] Kwon *et al.*,[64] Schmid *et al.*,[73] Cook,[42] Fu *et al.*,[74] Omari *et al.*[59] and Park *et al.*[41] The advantage of these type of algorithms is that they transform the full scene into a depth map; nevertheless, they are susceptible to noise and are computationally expensive. An alternative method for acquiring dense maps is using a single camera, together with an RGB-D sensor (e.g., Kinect). This permits depth acquisition in a more direct way—in a single view like the stereo vision. Unfortunately, this type of sensor is prone to interference, even from moderate sunlight, because it depends on projecting and recognising an infrared pattern (i.e., active-vision). Therefore, using this sensor for outdoor aerial navigation would result in erroneous measurements. In contrast, passive-vision systems function well in a wider range of light but require more time for computation.[70] Nevertheless, RGB-D sensor was used for accomplishing interesting tasks under light-controlled environments such as safe landing, as in refs. [77, 78]. The researchers proposed a detector for flat surfaces using depth information to decide whether a site is safe for landing. Other recent studies used RGB-D sensors to detect and to avoid obstacles, as in refs.

https://doi.org/10.1017/S0263574718000358 Published online by Cambridge University Press

[79, 80]. Their system looks for objects inside the view. If the depth of the detected objects is lower than a certain threshold, a warning and an avoiding command are sent to the MUAV.

In contrast to dense methods, sparse correspondence methods extract interesting features such as points, lines or zones using feature-extraction algorithms to find the disparity. Some of the most used methods for key point extraction are MOPS, SURF and SIFT. Sparse methods retrieve key points accompanied by a descriptor related to scale and orientation that differentiate it from others. This means that if two descriptors are similar, they have correspondent features.

These methods search for those points (and surroundings) that stand out after changes in rotation, scale or distortion in the image. This is done by applying filters (e.g., Gaussian) to images at different scales, extracting salient points for each scale and considering only those points that prevailed over the different scales after filtering. Noisy and weak points are rejected. These sets of points form a key point, and their distribution and gradient determine their descriptor.

Descriptors are the identifiers of the features. Generally, the descriptor is formed by u, v, σ, θ and $\mathbf{f}$. These are the horizontal and vertical locations of the feature in the image, the scale, the orientation and the histogram vector feature, which contains elements such as the gradient orientations of the key-point-pixels neighbourhood, as with SIFT, or responses to filters, as with MOPS and SURF.

Sparse depth acquisition results in an interesting method for navigation that permits the stereo matching to build maps and to estimate the depth of interesting features that belong to objects (e.g., obstacles). This type of depth acquisition is used for MUAVs like those in refs. [25, 66, 72 and 59], where feature-based algorithms were used to find obstacles in the scene.

4.3. Probabilistic approaches

The understanding of depth is useful in obstacle avoidance tasks and mapping, as in ref. [64]. However, some systems use other sensors, in addition to the stereoscopic camera, to make the system redundant and robust. These include inertial sensors, as in refs. [66, 73 and 71]; specialised thermal cameras for fire detection, as in ref. [42]; and laser scanners, as in ref. [70].

The use of sensors implies uncertainty about the measurements, and if the uncertainty is not treated properly, the estimations can be wrong. Hence, probabilistic methods are used to reduce uncertainty when retrieving an estimate with a mean and a covariance describing the probability of the estimation. One of the most widely used probabilistic methods is the Kalman Filter (KF),[81] which deals with noisy readings using dynamical system, control and measurement models.

Consider χ_t as the state vector $n \times 1$ in the time t, χ_{t-1} the previous state vector and U_t the control vector $m \times 1$ in the linear equation

$$\begin{aligned} \chi_t &= A_t \chi_{t-1} + B_t U_t + \varepsilon_t \\ \mathfrak{z}_t &= C_t \chi_t + \delta_t \end{aligned} \tag{21}$$

A_t is the state matrix of size $n \times n$, where n is the size of χ_t. B_t is the control matrix $n \times m$, ε_t the Gaussian noise of the same size as the state vector, C_t is the measurement matrix $k \times n$, where k is the size of the measurement vector $\mathfrak{z}_t$.

Considering equation in (21), the KF is defined as

$$\begin{aligned} \bar{\mu}_t &= A_t \mu_{t-1} + B_t U_t \\ \bar{\Sigma}_t &= A_t \Sigma_{t-1} A_t^{\mathrm{T}} + R_t \\ K_t &= \bar{\Sigma}_t C_t^{\mathrm{T}} (C_t \bar{\Sigma}_t C_t^{\mathrm{T}} + Q_t)^{-1} \\ \mu_t &= \bar{\mu}_t + K_t (\mathfrak{z}_t - C_t \bar{\mu}_t) \\ \Sigma_t &= (I - K_t C_t) \bar{\Sigma}_t \end{aligned} \tag{22}$$

where the first and second lines are the prediction phase, $\bar{\mu}$ and $\bar{\Sigma}$ are the estimated predicted state mean and its covariance, μ_{t-1} is the previous state mean and R_t is the error covariance. The third line corresponds to the Kalman Gain K_t, where Q_t is the covariance matrix of the measurement noise. The fourth and fifth lines are the update phase, where μ and Σ are the output of the filter.

https://doi.org/10.1017/S0263574718000358 Published online by Cambridge University Press

KF is useful for odometry and mapping applications like those in the MUAV studies conducted by Fu *et al.*,[74] Omari *et al.*,[59] Schauwecker and Zell,[72] Garcia Carrillo *et al.*,[66] Tomic *et al.*,[70] Schmid *et al.*[73] and Griesbach *et al.*[71] They used this probabilistic method to localise both the robot and features, being aware of the uncertainties, and to reduce the error by fusing data from different sensors. Moreover, the error can be significantly reduced when finding landmarks, leading to uncertainty reduction of the previous states.

A landmark (robust feature) is a feature that corresponds to a common physical object such as a tree, a corners or any specific object in an environment from which its range and bearing can be measured. Landmarks are useful for MUAV pose estimation by identifying and tracking the landmarks. Hence, a MUAV can estimate its pose and build a map more precisely because with the recognised landmarks, the other features are associated with a more general rather than local scope.

$$\begin{aligned} f(\mathfrak{z}_t) &= \{f_t^1, f_t^2, \ldots\} \\ p(f(\mathfrak{z}_t)|\chi_t, m) &= \prod_i p(\chi_t^i|\chi_t, m) \end{aligned} \tag{23}$$

In Eq. (23), $f(\mathfrak{z}_t)$ are the pose ($(x, y, z, \phi, \theta, \psi)^{\mathrm{T}} \in \mathbb{R}^3$) of the landmarks at that specific time t; consecutively, these are added to the map consisting of a list of features $m = \{m_1, m_2, \ldots\}$. The pose of the MUAV is χ_t^i ($(X, Y, Z, \Phi, \Theta, \Psi)^{\mathrm{T}} \in \mathbb{R}^3$) relative to the map. The probability of the features, given the current state and map m, is as stated in the second line, assuming that the measurements are conditionally independent.

The possibility of recognising landmarks in the scene permits the MUAV to estimate its pose in unknown environments more reliably (no access to a map). This is helpful for the vehicle to self-navigate and to build a map because it provides local information from the scene during navigation. Such a task is known as concurrent mapping and localisation or SLAM.[30]

4.4. Towards autonomous navigation

Vision-based autonomous navigation requires the aforementioned elements to work together so that the drone can complete a mission by on its own despite being in an unknown environment. The MUAV must estimate its pose within a map while building the map, and it must navigate towards the goal accord to that map. An algorithm that addresses the self-localisation and mapping problem is SLAM. This algorithm is typically implemented in the state estimation, landmark extraction, data association, state update and landmark update stages.

The first stage uses the motion model to get an estimation of the state at time t, as seen in Section 2. This concerns only the robot pose—not the map elements. The motion model retrieves the belief, which is the posteriori distribution, given the previous belief and the control actions.[82]

$$p(\chi_t|\chi_{t-1}, U_t) \tag{24}$$

One of the most studied SLAM algorithms is the EKF SLAM,[28] which considers a linearisation of the vehicle motion

$$\begin{aligned} p(\chi_t|\chi_{t-1}, U_t) &\Longleftrightarrow \quad \chi_k = F(\chi_{t-1}, U_t) + W_t \\ p(\mathfrak{z}_t|\chi_t, m) &\Longleftrightarrow \quad \mathfrak{z}(t) = h(\chi_t, m) + V_t \end{aligned} \tag{25}$$

where m is the map, $F(\chi_{t-1}, U_t)$ is the linearised model of the MUAV kinematics, W_t are Gaussian disturbances with covariance Q_t, $h(\chi_t, m)$ is the linearised model of the observations and V_t, the Gaussian observation errors with covariance R_t. The EKF on the SLAM algorithm estimates the mean

$$\begin{bmatrix} \hat{\chi}_t \\ \hat{m}_t \end{bmatrix} = E\left[\begin{matrix} \chi_t \\ m \end{matrix} |\mathfrak{z}_{0:t}\right] \tag{26}$$

https://doi.org/10.1017/S0263574718000358 Published online by Cambridge University Press

and the covariance is

$$\begin{aligned} P_t &= \begin{bmatrix} P_{\chi\chi} & P_{\chi m} \\ P_{\chi m}^{\mathrm{T}} & P_{mm} \end{bmatrix}_t \\ &= E\left[\begin{pmatrix} \chi_t - \hat{\chi}_t \\ m - \hat{m}_t \end{pmatrix} \begin{pmatrix} \chi_t - \hat{\chi}_t \\ m - \hat{m}_t \end{pmatrix}^{\mathrm{T}} |_{\mathfrak{z}0:t} \right]_t \end{aligned} \tag{27}$$

These predictions of the posteriori distributions are updated in time t

$$\begin{aligned} \hat{\chi}_t &= F(\chi_{t-1}, U_k) \\ P_{\chi\chi,t} &= \nabla F\, P_{\chi\chi,t-1} \nabla F^{\mathrm{T}} + Q_k \end{aligned} \tag{28}$$

where ∇F is the Jacobian of F at the previous estimated state.

In the landmark extraction stage, SLAM aims to get easily traceable features. It involves visual information extraction, and positions of features are retrieved, as explained in Sections 3 and 4. The feature detector gets features from each of the stereo images; then these are matched to calculate their depth. The feature descriptor and the depth are saved in a data structure (e.g., a vector) for further use.

Data association involves matching observations from a new view with previously known landmarks or with landmarks in previous views. In the event that the SLAM has known correspondences (i.e., known pose of landmarks), the data association is done by comparing the correspondence information with the observation, as in Tomic's paper.[70] In the case of unknown correspondences (i.e., unknown pose of landmarks), the SLAM algorithm itself must create the landmarks, as in the studies of Thamrin *et al.*,[25] Schauwecker and Zell,[72] Schneider[68] and Yang.[83] If there is a match with some of the features, the correspondence exists and is saved in a vector. Whenever this happens, the corresponding points can be estimated and saved within the map variable.

The state and landmark update stages consider both the motion and sensor estimations. From the previous stage, the landmarks improve the pose estimation, thereby, reducing the uncertainty of landmarks as well.

$$\begin{aligned} \begin{bmatrix} \chi_t \\ m_t \end{bmatrix} &= \begin{bmatrix} \hat{\chi}_t \\ \hat{m}_t \end{bmatrix} + W_t[\mathfrak{z}_t - h(\hat{\chi}_t, \hat{m}_t)] \\ P_t &= P_t - W_t S_t W_t^{\mathrm{T}} \end{aligned} \tag{29}$$

from where S_t and W_t are

$$\begin{aligned} S_t &= \nabla h P_t \nabla h^{\mathrm{T}} + R_t \\ W_t &= P_t \nabla h^{\mathrm{T}} S_t^{-1} \end{aligned} \tag{30}$$

Hence, in (31), the whole problem is the estimation of the posterior pose and map, given the sensor's measurements and robot's controls.

$$\begin{aligned} &p(\chi_t, m|_{\mathfrak{z}1:t}, U_{1:t}) \\ &= \int\int \cdots \int p(\chi_{1:t}, m|_{\mathfrak{z}1:t}, U_{1:t}) d\chi_1 d\chi_2 \ldots d\chi_{t-1} \end{aligned} \tag{31}$$

$$\begin{aligned} y_t &= \begin{pmatrix} \chi_t \\ m \end{pmatrix} \\ &= (\chi \quad m_1\, s_1 \quad m_2\, s_2 \quad \ldots \quad m_N\, s_N)^{\mathrm{T}} \end{aligned} \tag{32}$$

In these equations, χ_t is the pose at time t, m is the map, $z_{1:t}$ is the sensors' estimations, $U_{1:t}$ are the control actions, y_t is the state vector with size (4N+6), N is the number of landmarks and s is the landmark size.

https://doi.org/10.1017/S0263574718000358 Published online by Cambridge University Press

Besides performing mapping and self-localising, the SLAM algorithm also looks for convergence. When a robot moves through an unknown environment, it observes landmarks and keeps track of them. Eventually, the robot may encounter a previously seen landmark, so the acquired information can be propagated to the entire map, significantly reducing uncertainty. Therefore, it is important to maintain the correlations of the pose and map. Studies that sought the loop closure for visual SLAM applied to MUAVs include those by Tweedale and Jeffrey,[78] Schmid *et al.*[73] and Gomez-Ojeda *et al.*[84]

Schauwecker *et al.*[72] and Fu *et al.*[74] implemented BA at a given frequency to correct map drift continuously. Schneider *et al.*[68] presented a study that used two pairs of stereo cameras, one looking backwards and the other looking frontwards, to cover a large field of view. Similarly, Zhang *et al.*[1] and Gomez *et al.*[84] implemented a local BA. They tracked landmarks and their corresponding projections to get the new estimations with BA. Besides, they proposed a further BA at each landmark location. Incremental BA implementations permit the MUAV to use the adjusted estimations because they use ongoing observations rather than dealing with a batch of views frequently.

From the surveyed papers, we deem it important to mention those in which the main algorithm was SLAM because they considered the elements involved in autonomous navigation. These are papers of Thamrin *et al.*,[25] Cook,[42] Kwon *et al.*,[64] Garcia Carrillo *et al.*,[66] Schmid *et al.*,[73] Schauwecker and Zell[72] and Tomic *et al.*[70]

Thamrin *et al.* presented a stereo-vision approach for agricultural applications for which the landmarks were papaya trees commonly located in zones with no GPS signal. Therefore, a SLAM implementation was an appropriate solution. They also proposed a virtual point projection (VPP) that calculates an asymptote point parallel to the trees on the plantation so that the MUAV could follow them and navigate between the rows of trees.

Cook proposed a system with an onboard computer doing both the image acquisition and the processing. The main purpose was to use a stereo thermal camera to localise hot spots, which are common in disaster areas. The simulations were performed by setting the starting and target points and navigating between them. The experiments demonstrated successful navigation with no collision. Another experiment considered thermal sources as nearer obstacles than non-thermal ones so that the quad-rotor could detect and avoid them.

Kwon *et al.* significantly changed the baseline distance of a stereo rig to test its effects over detected obstacles—false alarms and missed detections. They recorded 11 false alarms and 1 missed detection under small/medium baseline lengths (124 mm–1,000 mm) and 6 false alarms and 0 missed detections under large baseline lengths (1,000 mm–2,132 mm).

Garcia Carrillo *et al.* proposed performing autonomous take-off, self-positioning, navigation and landing in unknown environments. The mounted sensors were a stereo camera, a laser range finder and an IMU. While the IMU worked fine with fast movement, the visual odometry (VO) worked better with slow changes in the field of view. A laser sensor was used to estimate de UAV's altitude. For the VO, they performed their experiments by finding 2D-to-2D correspondences between the current and previous stereo pairs. Next, via triangulation, the current and previous states were calculated to estimate the evolution of the vehicle recursively. To confirm correspondences between the left and right images, the equation in (6) was repeatedly evaluated. Simultaneously, they did a 3D reconstruction by extracting the rotation and translation from the previous state to the current one. The first experiment consisted of Euler angles estimation via VO against IMU (as the ground truth). The aircraft was given instructions about tilt movements. The results showed a maximum error on the Euler angle estimation of 5°. In the second experiment, a manually controlled flight over a square trajectory of 150 cm × 300 cm at an altitude of 50 cm was done. The VO estimation retrieved a smoother trajectory than the IMU trajectory estimation. The maximum error was of 25 cm. In their third experiment, they manually controlled the flight, varying the altitude from 50 cm to 200 cm. The results showed a maximum error of about 60 cm. The authors concluded that this big drift on the estimation was the result of deficiencies in the VO algorithm.

A study by Schmid *et al.* used an onboard stereo camera and an IMU. The MUAV navigated using Dijkstra's algorithm, avoiding obstacles. It also performed mapping and retrieved the odometry. This study used a quad-rotor that performed navigation in complex indoor and outdoor scenarios. It depended on a MoCAP system that limits autonomous navigation to a confined zone. They evaluated the quality of the VO using both 2D-to-2D and 3D-to-3D feature correspondences. Eventually, they were able to demonstrate autonomous navigation functionalities. Their first experiments proved that for delays in the measurements around 200 ms, the RMSE was lower than 0.15 m, and for delays

https://doi.org/10.1017/S0263574718000358 Published online by Cambridge University Press

around 1 s, the RMSE was higher than 0.25 m. The error remained around 0.1 m on frequencies between 15 Hz and 10 Hz but increased beyond 0.2 m on frequencies lower than 5 Hz. Similar behaviour was observed for velocity estimation. For the second experiment, they developed a hand-held device, apart from the UAV, to validate the system proposal. They used a tachymeter for ground truth comparison. The average error for the VO position was less than 1m for seven of the runs and above 7.5 m for four of them. In the third experiment, they validated the system over the UAV. The criterion for evaluating the system was the drift of take-off versus landing, considering that they are supposed to be at the same place. The loop closure error was smaller than 2.2%.

Schauwecker and Zell[72] and Tomic *et al.*[70] used a stereo SLAM algorithm and achieved autonomous navigation. Schauwecker and Zell's drone used two stereoscopic cameras—one facing frontwards to detect obstacles and the other facing downwards to detect the ground for altitude estimation and hover control via sparse feature detection. They performed autonomous take-offs and landings inside a laboratory. They used the facing frontwards camera to track 2D-to-2D features to estimate the MUAV pose with the VO. The downfacing camera was used to detect the ground plane using RANSAC[47] and to estimate the height as well as the pitch and roll angles. They performed two experiments. First, the UAV was instructed to take-off, hover and land. They then instructed it to navigate on a squared trajectory. They compared the inertial, visual-inertial SLAM and odometry estimations using only one pair of cameras facing downwards against the ground truth. The error for the first experiment was 1.44 cm for the onboard estimation, 3.15 cm for the local SLAM and 23.7 cm with one pair of downward-facing cameras. For the second experiment, the average error was 2.2 cm for the onboard estimation, 4.39 cm for the local SLAM and 41.3 cm with one pair of downward-facing cameras.

The approach of Tomic *et al.*[70] added a laser sensor to the vision system. This method identified robust landmarks by vision and laser information to reduce deviation from the environment. VO and laser odometry were compared by using covariance to determine the best odometry system either indoors or outdoors. The achieved autonomy was presented in two levels: low level (e.g., flight control and stabilisation) and high level (e.g., awareness of surroundings and mission planning). Their experiments were developed by alternating the vision system and the laser sensor for obstacle avoidance. For the VO, they used three recognisers: 2D-to-2D feature correspondence for the landing spot search, 2D-to-3D for a chimney search and 3D-to-3D for the house search during the fly-over. They used laser odometry for the take-off task because the stereo vision was limited as a result of the closeness to the ground. The system activated the stereoscopic odometry once the MUAV was hovering. Both odometry methods were used to estimate the current location relative to space and objects.

5. Conclusion

The aim of this survey was to demonstrate the importance of stereo cameras in the attempts at MUAV autonomous navigation. Stereo cameras, along with LiDARs, ultrasonic, GPS and IMU, are among the most used sensors for performing navigation. The first reason is that stereo cameras are low-weight devices suitable for mounting on board MUAVs with limited payload capacity. The second reason is the possibility of getting 3D information because of the physical configuration of the camera. Another reason is that, because of the mobility of MUAVs (i.e., through the air), these vehicles could escape either from the pilot's visual field or from transmission range (e.g., control and image transmission). Moreover, the pilot's view from ground level could be ambiguous, and the resulting confusion could cause accidents. Hence, giving vision (i.e., cameras) and processing (e.g., single-board computers) capabilities to a drone would make it accomplish its missions more safely.

However, there are important challenges because drones must perform in changing situations that involve real-time limitations (i.e., dealing with image processing, 3D interpretation, sensor-fusion, navigation planning and mapping). This explains researchers' increased interest in using stereo cameras to achieve autonomous navigation successfully in aerial vehicles. However, stereo vision cannot handle all the possible situations in autonomous navigation because they have physical (i.e., pixel resolution) and computational (i.e., processing rate) limitations. Hence, other sensors may be fused with the stereo camera to achieve the goal, but energy, payload and application limitations must be always considered.

https://doi.org/10.1017/S0263574718000358 Published online by Cambridge University Press

Acknowledgements
This work was supported partially by the Mexican National Council of Science and Technology (CONACYT).

References

1. J. Zhang, W. Liu and Y. Wu, "Novel technique for vision-based UAV navigation," *IEEE Trans. Aerosp. Electron. Syst.* **47**(4), 2731–2741 (2011).
2. K. Zhu and F. Lin, "Image Super-Resolution Reconstruction by Sparse Decomposition and Scale-Invariant Feature Retrieval in Micro-UAV Stereo Vision," *Proceedings of the 11th IEEE International Conference on Control and Automation ICCA*, Taichung, Taiwan Image (Jun. 18–20, 2014) pp. 705–710.
3. V. B. Azevedo, A. F. De Souza, L. P. Veronese, C. Badue and M. Berger, "Real-Time Road Surface Mapping Using Stereo Matching, V-Disparity and Machine Learning," *Proceedings of the IEEE International Joint Conference on Neural Networks IJCNN* (Aug. 2013) pp. 1–8.
4. A. Dawadee, J. Chahl and D. N. Nandagopal, "An algorithm for autonomous aerial navigation using landmarks," *J. Aerosp. Eng.* **29** (May 2016) pp. 1–27.
5. A. R. Vetrella and G. Fasano, "Cooperative UAV Navigation Under Nominal GPS Coverage and in GPS-Challenging Environments," *Proceedings of the IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a Better Tomorrow RTSI* (Sep. 2016), pp. 1–5.
6. R. Skulstad, C. L. Syversen, M. Merz, N. Sokolova, T. I. Fossen and T. A. Johansen, "Net Recovery of UAV with Single-Frequency RTK GPS," *Proceedings of the IEEE Aerospace Conference* (Mar. 2015), pp. 1–10.
7. A. Aditya, "Implementation of a 4D Fast SLAM Including Volumetric Sum of the UAV," *Proceedings of the International Conference on Sensing Technology ICST* (2012) pp. 78–83.
8. J. Wu, W. Fei and Q. Li, "An Integrated Measure and Location Method Based on Airborne 2D Laser Scanning Sensor for UAV's Power Line Inspection," *Proceedings of the 5th Conference on Measuring Technology and Mechatronics Automation ICMTMA* (2013) pp. 213–217.
9. R. Li, J. Liu, L. Zhang and Y. Hang, "LIDAR/MEMS IMU Integrated Navigation (SLAM) Method for a Small UAV in Indoor Environments," *Proceedings of the IEEE DGON Inertial Sensors and Systems ISS* (Sep. 2014) pp. 1–15.
10. F. Wang, K. Wang, S. Lai, S. K. Phang, B. M. Chen and T. H. Lee, "An Efficient UAV Navigation Solution for Confined but Partially Known Indoor Environments," *Proceedings of the IEEE International Conference on Control and Automation ICCA* (2014) pp. 1351–1356.
11. J. R. G. Braga, H. F. D. C. Velho and E. H. Shiguemori, "Estimation of UAV Position Using LiDAR Images for Autonomous Navigation Over the Ocean," *Proceedings of the IEEE 9th International Conference on Sensing Technology ICST*, (Dec. 2015) pp. 811–816.
12. R. Opromolla, G. Fasano, G. Rufino, M. Grassi and A. Savvaris, "LIDAR-Inertial Integration for UAV Localization and Mapping in Complex Environments," *Proceedings of the International Conference on Unmanned Aircraft Systems ICUAS* (2016) pp. 649–656.
13. K. Masuko, I. Takahashi, S. Ogawa, M.-H. Wu, A. Oosedo, T. Matsumoto, K. Go, F. Sugai, A. Konno and M. Uchiyama, "Autonomous Takeoff and Landing of an Unmanned Aerial Vehicle," *Proceedings of the IEEE/SICE International Symposium on System Integration* (2010) pp. 248–253.
14. Z. Yao, X. Bin, Y. Qiang, L. Yang and W. Fu, "Autonomous Control System for the Quadrotor Unmanned Aerial Vehicle," *Proceedings of the 31st Chinese Control Conference* (Jul. 2012) pp. 4862–4867.
15. N. Gageik, M. Strohmeier and S. Montenegro, "An autonomous UAV with an optical flow sensor for positioning and navigation," *Int. J. Adv. Robot. Syst.* **10**, p. 341 (Oct. 2013).
16. V. More, H. Kumar, S. Kaingade, P. Gaidhani and N. Gupta, "Visual Odometry Using Optic Flow for Unmanned Aerial Vehicles," *Proceedings of the IEEE International Conference on Cognitive Computing and Information Processing CCIP* (Mar. 2015) pp. 1–6.
17. F. Jesus and R. Ventura, "Combining Monocular and Stereo Vision in 6D-SLAM for the Localization of a Tracked Wheel Robot," *Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics SSRR* (Nov. 2012) pp. 1–6.
18. E. Mattar, K. Al-Mutib, M. Alsulaiman, H. Ramdane and M. Emaduddin, "A Survey: Intelligent Based Mobile Robots Stereo Vision Maps Synthesis and Learning Methodologies," *Proceedings of the 5th International Conference on Intelligent Systems, Modelling and Simulation* (2014) pp. 94–98.
19. D. Moratuwage, D. Wang, A. Rao, N. Senarathne and H. Wang, "RFS collaborative multivehicle slam: Slam in dynamic high-clutter environments," *IEEE Robot. Autom. Mag.* **21**(2), 53–59 (2014).
20. O. Stasse, A. Davison, R. Sellaouti and K. Yokoi, "Real-Time 3D SLAM for Humanoid Robot Considering Pattern Generator Information," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (Oct. 2006) pp. 348–355.
21. S. Ahn, S. Yoon, S. Hyung, N. Kwak and K. S. Roh, "On-Board Odometry Estimation For 3D Vision-Based SLAM of Humanoid Robot," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (Oct. 2012) pp. 4006–4012.
22. Lili Meng, C. W. de Silva and Jie Zhang, "3D Visual SLAM for an Assistive Robot in Indoor Environments Using RGB-D Cameras," *Proceedings of the 9th IEEE International Conference on Computer Science & Education* (Aug. 2014) pp. 32–37.

https://doi.org/10.1017/S0263574718000358 Published online by Cambridge University Press

23. M. F. Fallon, J. Folkesson, H. McClelland and J. J. Leonard, "Relocating Underwater Features Autonomously Using Sonar-Based SLAM," *IEEE J. Ocean. Eng.* **38**, 500–513 (Jul. 2013).
24. F. Hidalgo and T. Braunl, "Review of Underwater SLAM Techniques," *Proceedings of the IEEE 6th International Conference on Automation, Robotics and Applications ICARA* (Feb. 2015) pp. 306–311.
25. N. M. Thamrin, N. H. M. Arshad, R. Adnan, R. Sam, N. A. Razak, M. F. Misnan and S. F. Mahmud, "Simultaneous Localization and Mapping Based Real-Time Inter-Row Tree Tracking Technique for Unmanned Aerial Vehicle," *Proceedings of the IEEE International Conference on Control System, Computing and Engineering* (Nov. 2012) pp. 322–327.
26. M. Bryson and S. Sukkarieh, "Observability analysis and active control for airborne SLAM," *IEEE Trans. Aerosp. Electron. Syst.* **44**, 261–280 (Jan. 2008).
27. R. Li, J. Liu, L. Zhang and Y. Hang, "LIDAR/MEMS IMU Integrated Navigation (SLAM) Method for a Small UAV in Indoor Environments," *Proceedings of the IEEE DGON Inertial Sensors and Systems ISS* (Sep. 2014) pp. 1–15.
28. H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robot. Autom. Mag.* **13**, 99–110 (Jun. 2006).
29. T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robot. Autom. Mag.* **13**, 108–117 (Sep. 2006).
30. S. Thrun, *Probabilistic Robotics* (The MIT Press, Cambridge, MA; London, UK, 2005).
31. A. J. Davison, I. D. Reid, N. D. Molton and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.* **29**, 1052–1067 (Jun. 2007).
32. P. Corke, *Robotics, Vision and Control*, Springer Tracts in Advanced Robotics, vol. 73 (Springer, Berlin Heidelberg, 2011).
33. M. Klingensmith, "Overview of motion planning," Accessed on 19 August 2016, http://www.gamasutra.com/blogs/MattKlingensmith/20130907/199787/Overview_of_Motion_Planning.php (2013).
34. J. Denny, S. Rodriguez and N. Amato, "Algorithms & applications group. Adapting RRT growth for heterogeneous environments," Accessed on 19 August 2016, https://parasol.tamu.edu/groups/amatogroup/research/AdaptiveRRT/, (2013).
35. R. J. Geraerts, Sampling-Based Motion Planning: Analysis and Path Quality *Ph.D. Thesis* (Utrecht University, 2006).
36. H. Choset, "Robotic Motion Planning: Roadmap Methods," Tech. Rep. 16-735, (Carnegie Mellon University, 2007), Pittsburg, PA, USA, Accessed on 20 August 2016, http://www.cs.cmu.edu/∼motionplanning/lecture/Chap5-RoadMap-Methods_howie.pdf
37. H. Choi, Y. Kim, Y. Lee and E. T. KIM, "A reactive collision avoidance algorithm for multiple midair unmanned aerial vehicles," *Trans. Jpn. Soc. Aeronaut. Sp. Sci.* **56**(1), 15–24 (2013).
38. D. Maravall, J. de Lope and J. Pablo Fuentes, "Vision-based anticipatory controller for the autonomous navigation of an UAV using artificial neural networks," *Neurocomputing* **151**, 101–107 (Mar. 2015).
39. S. Emel'yanov, D. Makarov, A. I. Panov and K. Yakovlev, "Multilayer cognitive architecture for UAV control," *Cogn. Syst. Res.* **39**, 58–72 (2016).
40. S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation," *Artif. Intell.* **99**, 21–71 (Feb. 1998).
41. J. Park, Y. Kim and S. Kim, "Landing site searching and selection algorithm development using vision system and its application to quadrotor," *IEEE Trans. Control Syst. Technol.* **23**, 488–503 (Mar. 2015).
42. Z. Cook, L. Zhao, J. Lee and W. Yim, "Unmanned Aerial System for First Responders," *Proceedings of the IEEE 12th International Conference on Ubiquitous Robots and Ambient Intelligence URAI* (Oct. 2015) pp. 306–310.
43. K.G. Derpanis, "The Harris Corner Detector", Technical Report, York University, Available: www.cse.yorku.ca/kosta/CompVis Notes/ harris detector.pdf (Oct. 2004), pp. 2–3.
44. M. Brown, R. Szeliski and S. Winder, "Multi-Image Matching Using Multi-Scale Oriented Patches," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR,* vol. 1 (2005) pp. 510–517.
45. D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.* **60**, 91–110 (Nov. 2004).
46. H. Bay, A. Ess, T. Tuytelaars and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Underst.* **110**, 346–359 (Jun. 2008).
47. M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM* **24**(6), 381–395 (1981).
48. P. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach Intell.* **14**, 239–256 (Feb. 1992).
49. S. Gold, A. Rangarajan, C.-P. Lu, S. Pappu and E. Mjolsness, "New algorithms for 2D and 3D point matching," *Pattern Recog.* **31**, 1019–1031 (Aug. 1998).
50. Y. Tsin and T. Kanade, *A Correlation-Based Approach to Robust Point Set Registration* (Springer, Berlin Heidelberg, 2004) pp. 558–569.
51. A. Myronenko and Xubo Song, "Point set registration: Coherent point drift," *IEEE Trans. Pattern Anal. Mach. Intell* **32**, 2262–2275 (Dec. 2010).
52. Thomson Reuters, "Web of science," Accessed on 13 February 2017, http://apps.webofknowledge.com/

https://doi.org/10.1017/S0263574718000358 Published online by Cambridge University Press

53. H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature* **293**, 133–135 (Sep. 1981).
54. R. Hartley and A. Ziserman, *Multiple View Geometry in Computer Vision*, vol. 2, 2nd ed. (Cambridge: Cambridge University Press, 2004).
55. V. Lepetit, F. Moreno-Noguer and P. Fua, "EPNP: An accurate o(n) solution to the PNP problem," *Int. J. Comput. Vis.* **81**(2), (2009) pp. 155–166.
56. A. Makadia, C. Geyer and K. Daniilidis, "Correspondence-free structure from motion," *Int. J. Comput. Vis.* **75**, 311–327 (Sep. 2007).
57. J. Lee, K. Lee, S. Park, S. Im and J. Park, "Obstacle avoidance for small UAVs using monocular vision," *Aircr. Aerosp. Technol.* **83**, 397–406 (Oct. 2011).
58. A. Ferworn, S. Herman, J. Tran, A. Ufkes and R. Mcdonald, "Disaster Scene Reconstruction: Modeling And Simulating Urban Building Collapse Rubble Within A Game Engine," *Proceedings of the Summer Computer Simulation Conference SCSC,* Vista, CA (Society for Modeling, Simulation International, 2013) pp. 18:1–18:6.
59. S. Omari, M. Bloesch, P. Gohl and R. Siegwart, "Dense Visual-Inertial Navigation System For Mobile Robots," *Proceedings of the International Conference on Robotics and Automation ICRA* (2015) pp. 2634–2640.
60. R. Carloni, V. Lippiello, M. DAuria, M. Fumagalli, A. Y. Mersha, S. Stramigioli and B. Siciliano, "Robot vision: Obstacle-avoidance techniques for unmanned aerial vehicles," *IEEE Robot. Autom. Mag.* **20**(4), 22–31 (2013).
61. A. Harmat and I. Sharf, "Towards Full Omnidirectional Depth Sensing Using Active Vision for Small Unmanned Aerial Vehicles," *Proceedings of the Canadian Conference on Computer and Robot Vision* (2014) pp. 24–31.
62. D. Warren Mellinger, Trajectory Generation and Control for Quadrotors. *Dissertation for Doctor of Philosophy Ph.D. Thesis* (University of Pennsylvania, 2012).
63. S. Shen, Autonomous Navigation In Complex Indoor and Outdoor Environments With Micro Aerial Vehicles. *A Dissertation for the Degree of Doctor of Philosophy Ph.D. Thesis* (University of Pennsylvania 2014).
64. J.-W. Kwon, J. Seo and J. H. Kim, "Multi-UAV-based stereo vision system without GPS for ground obstacle mapping to assist path planning of UGV," *Electron. Lett.* **50**, 1431–1432 (Sep. 2014).
65. B. Charrow, Information-Theoretic Active For Multi-Robot Teams. A Dissertation for Degree of Doctor of Philosophy *Ph.D. Thesis* (University of Pennsylvania, 2015).
66. L. R. García Carrillo, A. E. Dzul López, R. Lozano and C. Pégard, "Combining stereo vision and inertial navigation system for a quad-rotor UAV," *J. Intell. Robot. Syst.* **65**, 373–387 (Jan. 2012).
67. B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique With an Application to Stereo Vision," *Proceedings of the 7th International Joint Conference on Artificial Intelligence* (1981) pp. 674–679.
68. J. Schneider, F. Schindler, T. Läbe and W. Förstner, "Bundle Adjustment for Multi-Camera Systems with Points at Infinity," *ISPRS Annals of the Photogrammetry, Remote Sensing, and Spatial Information Sciences*, vol. 1–3, (Sep. 2012) pp. 75–80.
69. N. Vandapel, J. Kuffner and O. Amidi, "Planning 3-D Path Networks in Unstructured Environments," *Proceedings of the IEEE International Conference on Robotics and Automation* (2005) pp. 4624–4629.
70. T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. Grixa, F. Ruess, M. Suppa and D. Burschka, "Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue," *IEEE Robot. Autom. Mag.* **19**(3), 46–56 (2012).
71. D. Griesbach, D. Baumbach and S. Zuev, "Stereo-Vision-Aided Inertial Navigation for Unknown Indoor and Outdoor Environments," *Proceedings of the IEEE International Conference on Indoor Positioning and Indoor Navigation IPIN* (Oct. 2014) pp. 709–716.
72. K. Schauwecker and A. Zell, "On-board dual-stereo-vision for the navigation of an autonomous MAV," *J. Intell. Robot. Syst.* **74**(1–2), 1–16 (2014).
73. K. Schmid, P. Lutz, T. Tomi, E. Mair and H. Hirschmller, "Autonomous vision-based micro air vehicle for indoor and outdoor navigation," *J. Field Robot.* **31**, 537–570 (Jul. 2014).
74. C. Fu, A. Carrio and P. Campoy, "Efficient Visual Odometry and Mapping for Unmanned Aerial Vehicle Using Arm-Based Stereo Vision Pre-Processing System," *Proceedings of the International Conference on Unmanned Aircraft Systems ICUAS Denver Marriott Tech Center Denver*, Colorado, USA (Jun. 9–12, 2015) pp. 957–962.
75. N. Roma, J. Santos-Victor, and J. Tomé, "A Comparative Analysis of Cross-Correlation Matching Algorithms Using a Pyramidal Resolution Approach," Empir. Eval. Methods Comput. Vis., vol. 6, (2012), pp. 117–142.
76. R. Jain, R. Kasturi and B. Schunck, *Machine Vision*. Computer Science Series (USA: McGraw-Hill, 1995).
77. E. Frontoni, A. Mancini and P. Zingaretti, "UAVs Safe Landing Using Range Images," *Proceedings of the ASME/IEEE International Conference on Mechatronic and Embedded Systems and Applications, Parts A and B*, vol. 3 (2011) pp. 1047–1052.
78. J. W. Tweedale, "Fuzzy Control Loop in an Autonomous Landing System for Unmanned Air Vehicles," *Proceedings of the IEEE International Conference on Fuzzy Systems* (2012) pp. 1–8.

https://doi.org/10.1017/S0263574718000358 Published online by Cambridge University Press

79. G. B. Kim, T. K. Nguyen, A. Budiyono, J. K. Park, K. J. Yoon and J. Shin, "Design and development of a class of rotorcraft-based UAV," *Int. J. Adv. Robot. Syst.* **10** (2013) pp. 1–9.
80. A. Budiyono, G. Lee, G. B. Kim, J. Park, T. Kang and K. J. Yoon, "Control system design of a quad-rotor with collision detection," *Aircr. Eng. Aerosp. Technol.* **87**(1), 59–66 (2015).
81. R. E. Kalman *et al.*, "A new approach to linear filtering and prediction problems," *J. Basic Eng.* **82**, 35–45 (1960).
82. S. Thrun, R. Brooks and H. Durrant-Whyte, *Robotics Research*, Springer Tracts in Advanced Robotics, vol. 28 (Springer, Berlin Heidelberg, 2007).
83. S. Yang, S. A. Scherer, X. Yi and A. Zell, "Multi-camera visual SLAM for autonomous navigation of micro aerial vehicles," *Robot. Auton. Syst.* **93**, 116–134 (2017).
84. R. Gomez-Ojeda, F. Moreno, D. Scaramuzza and J. G. Jiménez, "PL-SLAM: A Stereo SLAM System Through the Combination of Points and Line Segments," CoRR. arXiv: 1705.09479, (2017), pp. 1–13.

https://doi.org/10.1017/S0263574718000358 Published online by Cambridge University Press