Wireless vision-based fuzzy controllers for moving object tracking using a quadcopter

Mohammed Algabri, Hassan Mathkour, Mohamed Amine Mekhtiche, Mohamed Abdelkader Bencherif, Mansour Alsulaiman, Mohammed Amer Arafah and Hamid Ghaleb

Abstract
During the last few decades, the utilization of unmanned aerial vehicles has grown in military and has seen new civil applications because of their reduced cost and hovering capabilities. This article presents a visual servoing system for detecting and tracking a moving object using an unmanned aerial vehicle. The system consists of two sequential components. The first component addresses the detection of a moving object, using the unmanned aerial vehicle based on color features. The second component addresses the visual servoing control that is designed to guide the unmanned aerial vehicle according to the target position and inclination. Three fuzzy logic modules are implemented in order to control the unmanned aerial vehicle in tracking a moving object. The proposed method is validated on a real flight using an AR.Drone 2.0 quadcopter. The obtained results show that the performance of the proposed method is suitable for object-following tasks in surveillance applications. This research investigates the use of unmanned aerial vehicle technology for crowd monitoring during Hajj and it could also be used for border surveillance to monitor Saudi Arabia’s borders.

Keywords
Unmanned aerial vehicles, object detection and tracking, fuzzy logic controller, fuzzylite, AR.Drone

Date received: 2 November 2016; accepted: 23 March 2017

Academic Editor: Miltiadis Lytras

Introduction
Nowadays, unmanned aerial vehicles (UAVs) are widely used in diverse applications, such as in civil engineering,1–3 in remote sensing applications,4,5 in three-dimensional (3D) mapping applications,6 in surveillance7 border monitoring,8–10 in traffic monitoring,11,12 in security and reconnaissance,2,13 in agriculture and vegetation monitoring using thermal and multispectral cameras,14–16 and in disaster management.17,18 Other applications are specific surface displacement monitoring such as the Himalayan debris-covered glacier,19 photogrammetry of cultural heritage and archeological sites,20 and data collection through a network of UAVs.21 Autonomous navigation and tracking is becoming more essential for UAVs. Indeed, tracking a moving target using a UAV is an important task for surveillance applications.22,23 UAV object tracking is the process of determining the position of a target while using the UAV camera to enable the UAV to stay near of it,24 or in some cases landing on...
Automatic target or object detection refers to the precise recognition of the signature of a specific object in a desired image or sensor data. Many techniques are used for this purpose such as scale invariant feature transform (SIFT), speeded up robust features (SURF), background subtraction, Kernel-based tracking, Bayesian estimation, hierarchical particle filtering, and robust object tracking with multiple instance learning.

This article focuses on designing vision-based control to detect and track a moving object using a quadcopter for surveillance and other potential applications using a fuzzy logic controller approach as detailed in Berenji and used in diverse applications such as robotic navigation and intelligent systems.

Our proposed system uses three fuzzy logic controller modules, using different membership functions, for the quadcopter-based object tracking in an unknown environment. A pitch fuzzy logic controller (PFLC) is designed to move the UAV forward or backward depending on object movement. A yaw fuzzy logic controller (YFLC) is employed to turn the UAV clockwise or counterclockwise based on the direction in which the object is heading. A GAZ fuzzy logic controller (GFLC) is proposed to perform up/down UAV movement. The other contribution of this article is to present the design of control approaches that constitute a trade-off between simplicity and robustness.

The article is structured as follows. Section “Methodology” presents the proposed method. Section “Fuzzy logic controllers” provides details of the fuzzy logic controllers used to control the quadcopter. Section “Experimental setup” contains details of the experimental setup. Section “Experimental results” presents the experimental results of the proposed controls to validate our method. Section “Conclusion” concludes the article and suggests some future work.

Related work

Soft computing techniques such as fuzzy logic are widely used in robot navigation. The application of fuzzy logic in autonomous navigation is based on the knowledge representation achieved through If-Then rules that contain linguistic variables describing the problem. Many controllers are described in the literature. The development of a new nonlinear controller for a UAV based on the use of a neural network has been described. Numerical results confirmed the tracking ability of the UAV and showed that proposed controller outperformed the conventional linear controller. A robust path-tracking controller for a UAV has been also proposed. The authors compared and analyzed a conventional proportional–integral–derivative (PID) controller and self-tuning PID based on fuzzy logic. The simulation results showed that both controllers are acceptable in terms of no variation in load. The PID controller is easy to design and has good performance in the case of a specific system. An adaptive neuro-fuzzy system for a UAV using ANFIS has been described. The simulation results showed the feasibility of the proposed controller for the autonomous flight control of UAVs. The navigation and control of UAVs based on fuzzy logic has further been presented. Two-module fuzzy logic was used to control small manned UAVs. The simulation results used an Aerosim Aeronautical Simulation in a MATLAB environment. The implementation of two fuzzy logic controllers for a pan-tilt camera on a UAV has been presented in Olivares-Méndez et al. A total of 94 rules were used for two inputs and one output on those controllers.

Methodology

Figure 1 depicts a block diagram of the proposed object-following system using the UAV. The input of the system is an image acquired by the UAV. In the object detection step, an object using OpenCV was detected by finding the features in the image that could be used to recognize objects. In the parameter generation step, the distance between the UAV and the object and the heading angle were calculated, including the altitude of objects based on the altitude of the UAV. These parameters were used to enable the controllers to control UAV movement.

Moving target detection

Object detection is an important object-tracking task. In this work, color-based detection was used to detect the object. A green object was detected using color-matching methods as shown in Figure 2(a). First, the RGB image was converted into an HSV color model as in Figure 2(b), where the HSV parameters used to select the specified color in the lower boundary were (45, 100, 50) for the hue, saturation, and value, respectively, and (75, 255, 255) for the upper boundary. Then, morphological operations were performed to remove small objects and fill small holes in the detecting region as in Figure 1.

Figure 1. Proposed method.
Figure 2(c). Figure 2(d) shows the detected objects that have the same pre-specified color.

**Parameters’ extraction**

The purpose of this step is to extract useful information from the image. The first parameter was the distance between the UAV and the object using equation (1)

\[
\text{Dist. (mm)} = \frac{\text{focal length} \times \text{real object height}}{\text{object height} \times \text{Pixel per mm}}
\]

where focal length is the focal length of the camera, which equals 3.85 mm; real object height is the height of the real object (mm); object height is the height of the object in an image (pixel); and Pixel per mm is the size of the pixel in millimeters. This value is computed from the camera matrix in the camera calibration file. It is equal to 0.00675 mm.

The second parameter calculated the orientation from the center of the camera to the center of the detected object. These two parameters were used as inputs to the PFLC and YFLC controllers. Then, the altitude of the object related to the UAV was calculated to control the upward and downward movement of the UAV using the GFLC.

**Fuzzy logic**

This section presents the implementation of the fuzzy logic controllers for object tracking using the fuzzylite toolbox. Fuzzylite is a free and open-source fuzzy logic controller programmed in C++ by Juan Rada-Vilela.

**PFLC**

The PFLC is used to control forward/backward UAV movement. This controller has two inputs and one output. The first input of this controller is the distance, in meters, between the UAV and the object. The range of this variable is divided into four linguistic variables as shown in Figure 3(a). The second input variable is the angle deviation, in degrees, between the center of the image captured by the UAV and the center of the object. The deviation is divided into five linguistic variables as in Figure 3(b). The PFLC is implemented with four membership functions for distance and five membership functions for the angle deviation as illustrated in Figure 1. The linguistic variables used for the distance are as follows: Very Near: the object is very close to the UAV; Near: the object is near the UAV; Far: the object is far from the UAV; and Very Far: the object is very far from the UAV. The linguistic variables used for the angle deviation are as follows: Big Left: the object is located to the leftmost of the UAV; Left: the object is located to the leftmost of the UAV; Center: the object is in front of the UAV; Right: the object is to the right of the UAV; and Big Right: the object is to the rightmost of the UAV. The PFLC sends the linear velocity on the X-axis to the UAV, to fly forward or backward quickly if the object is very far and slowly if the object is near the UAV. The output provided by the PFLC is divided into four linguistic variables, namely,
Backward (BW), Slow (S), Fast (F), and Very Fast (VF), as shown in Figure 3(c).

**YFLC**

The main purpose of the YFLC is to turn the UAV to the left or right in order to follow object rotation. The same inputs that were accepted by the previous controller, distance and deviation, were used to design this controller, as shown in Figure 4. The output of the YFLC is the angular velocity on the Z-axis to rotate the UAV clockwise or counterclockwise depending on the rotation of the detected object. The output of this controller is divided into five linguistic variables, namely, Big Right (BR), Right (R), Zero (Z), Left (L), and Big Left (BL), as shown in Figure 4(c). The linguistic variables used for the outputs of the PFLC and YFLC are listed in Table 1. The number of rules is reduced and the performance is improved by building the knowledge base using the 20 rules listed in Table 1.

**GFLC**

The GFLC is proposed to compute the linear velocity along the Z-axis in order to align the position of the
Figure 4. Membership functions for the YFLC: (a) fuzzy input—estimated distance between the UAV and the object (m), (b) fuzzy input—estimated angle between the UAV and the object (°), and (c) fuzzy output—angular velocity control heading of the UAV.

Table 1. Rule base for PFLC and YFLC controllers.

| Distance  | Deviation | Big left | Left | Center | Right | Big right |
|-----------|-----------|----------|------|--------|--------|-----------|
|           |           | Linear   | Angular | Linear | Angular | Linear   | Angular |
| Very near | BW        | BL       | BW    | L      | BW     | Z         | BW       | R       | BW       | BR       |
| Near      | S         | BL       | S     | L      | S      | Z         | S        | R       | S        | BR       |
| Far       | F         | BL       | F     | L      | F      | Z         | F        | R       | F        | BR       |
| Very far  | VF        | BL       | VF    | L      | VF     | Z         | VF       | R       | VF       | BR       |

PFLC: pitch fuzzy logic controller; YFLC: yaw fuzzy logic controller; BW: backward; BL: big left; L: left; Z: zero; R: right; BR: big right; S: slow; F: fast; VF: very fast.
UAV with that of the object. The GFLC is designed for altitude control. The GFLC accepts as its input the difference in the position of an object relative to the center of the image captured by the camera of the UAV. This distance is computed from the pixel coordinates. The output is the linear velocity along the Z-axis, where (−) means flying downward and (+) flying upward. The GFLC is implemented with four triangle membership functions as inputs, as shown in Figure 5. The linguistic variables used to compute the error in position are Very Up, Up, Down, and Very Down. The GFLC is designed to use the four triangle membership functions to produce the output shown in Figure 5. The linguistic variables used to compute the linear velocity along the Z-axis are as follows: Big Fly Down to fly downward quickly; Fly Down to fly downward; Fly Up to fly upward; and Big Fly Up to fly upward quickly.

**Experimental setup**

This section presents the experiments conducted using the proposed controllers to test the UAV performance.

**Parrot AR.Drone 2.0**

Experiments were conducted using a Parrot AR.Drone 2.0 quadrotor (power edition). The AR.Drone has four brushless motors that are used to power four propellers. The total weight is 380 g with the outdoor hull and 420 g with the indoor hull as shown in Figure 6. The UAV is equipped with multiple sensors: two cameras in front and a vertical camera, a three-axis accelerometer, a three-axis gyroscope with 2000°/s precision, a three-axis magnetometer with 6° precision, a pressure sensor, and ultrasonic sensor ground altitude measurement. The AR.Drone 2.0 on-board technology contains a 1 GHz ARM processor with 32-bit RAM and 1-Gbit DDR2 RAM at 200 MHz. It supports high-speed USB 2.0 and Wi-Fi.
**Robotic operating system**

The robotic operating system (ROS) is a meta-operating system that runs on Linux (Ubuntu) and is a flexible framework for writing robot software. It provides software frameworks for robot software development and was originally developed in 2007 by the Stanford Artificial Intelligence Laboratory. The main purpose of the ROS is to encourage collaborative robotics software developments (e.g. if one research group had expertise in mapping, and other groups were experts in navigation, and yet another group had experience in computer vision). Thus, the ROS was designed for collaboration between groups to enable researchers to build upon each other’s work.\(^{52,53}\) It is an agent-based support programming language (C++, Python, Lisp, Java, and more). More than 50 robots use the ROS.\(^{54}\) The ROS has a core known as the ROS Master, which provides name registration and look up to nodes. The processes that perform the computation are called nodes. Topics are the streams of data with publish/subscribe semantics. Request/replay is accomplished by services. Messages are used for data representation such as a data structure. The message is used to enable C++ to communicate with Python. The nodes communicate with the ROS or other nodes using topics (publish/subscribe) and services (request/response).

**Parrot AR-Drone 2.0 ROS driver**

AR-Drone has an ROS driver named ardrone_autonomy. This driver is based on AR-Drone SDK version 2.0.1. This package was developed in the Autonomy Lab of Simon Fraser University.\(^{55}\) The information received from the UAV is published on the topic “ardrone/navdata.” This contains information related to the current state of the drone, namely taking off, landing, and flying, including information about the rotation of the drone around the X-, Y-, and Z-axes. It also contains information about the linear velocity, acceleration, environmental temperature, and wind relating to the drone. After drone takeoff, this enables the drone to fly forward, backward, to the left, to the right, upward, downward, and by rotating clockwise/counterclockwise. The topic “cmd_vel” can be used to publish a message of the type “geometry_msgs::Twist.”

**Experimental results**

The designed controllers were tested for target tracking using the UAV by using a green ball as the object tracker, as shown in Figure 2. The experimental results were obtained using a Parrot AR.Drone 2.0 quadrotor in the experiments, as shown in Figure 6.

| Figure | Distance to the target (m) | Angle between UAV and target (°) |
|--------|---------------------------|---------------------------------|
| a      | 2.5                       | 9                               |
| b      | 3.01                      | -21.74                          |
| c      | 3.8                       | 33.05                           |
| d      | 7.33                      | -10.49                          |
| e      | 5.4                       | -13.28                          |
| f      | 7.89                      | 25.47                           |

UAV: unmanned aerial vehicle.

**Parameters extraction**

This section presents the results of the different experiments conducted to calculate the distance and angle between the UAV and target. Figure 7(a) clearly shows that the distance from the UAV to the target is 2.5 m and the angle is equal to 9°, with the black circle indicating the center of the camera of the UAV. Figure 7(b) shows that the distance is equal to 3 m and the heading angle between the UAV and target is −21.7°. The minus sign in the angle signifies a mark to the left of the UAV. By contrast, in Figure 7(c), the angle between the UAV and the target is 33°, which means a target to the right of the UAV. In terms of computing the input of the GAZ controller, Figure 7(d) shows when the target is located above the UAV and below the UAV, as shown in Figure 7(e). Figure 7(e) shows the distance and angle calculated when the target is located to the right and above the UAV.

The extracted information is listed in Table 2.

**Testing the proposed controllers**

In the flight experiment in an indoor environment, the target is moved by a human. After the UAV takes off from the initial location (0, 0, 0) to an altitude of 0.6 m, it starts searching for the moving target. When the moving target is not in the field of the UAV camera, the UAV rotates clockwise until it detects the object, as shown in Figure 8. The trajectory of the UAV during the flight experiment is shown in Figure 8.

Figure 9 shows the actual tracking distance between the UAV and the target. We specified the desired distance between the UAV and target as 1 m. If the distance is less than the desired distance, the UAV moves backward. In Figure 8, we can clearly see that the trajectory of the desired distance varies between 1 and 2 m most of the time during the flight experiment.

Note to the reader: the video of the flight experiment where the UAV tracks the ball placed on the moving robot is available via the following link: https://www.youtube.com/watch?v=koEpVvHDa7I
Figure 7. Computed controller parameters. Note. The first line describe the distance between the UAV and the target, the second line describe the angle between the front axis of the UAV and the target and the third line shows if the target is above or under the UAV.

Figure 8. Tracking trajectory of the UAV.
Conclusion

This article presents a moving object-tracking system using a UAV and fuzzy visual servoing. The proposed work intends to introduce UAVs for monitoring and surveillance in military applications as well as for monitoring crowds participating in Hajj rituals. The proposed system can detect a moving object in front of the camera of the UAV using a simple color-matching algorithm. Likewise, the system can guide the UAV to track the moving object stably. Experiments were performed on an AR.Drone 2.0 quadrotor using a mobile robot (Scout-II) to move the object. The experimental results demonstrate the efficiency of the UAV to detect the object at different distances and different tilts. The UAV was shown to be capable of detecting the moving object at a distance of 8 m and with an angle of 33°. Furthermore, real tests were carried out to validate the proposed fuzzy logic controllers in different scenarios. The proposed visual serving could be used to detect any another object with different colors and shapes. For future work, the proposed system could be enhanced by tuning the fuzzy logic using optimization techniques.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The authors extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this work through Research Group no. RGP-1436-002.

References

1. Liu P, Chen AY, Huang Y-N, et al. A review of rotorcraft unmanned aerial vehicle (UAV) developments and applications in civil engineering. Smart Struct Syst 2014; 13(6): 1065–1094.
2. Maza I, Caballero F, Capitán J, et al. Experimental results in multi-UAV coordination for disaster management and civil security applications. J Intell Robot Syst 2011; 61(1): 563–585.
3. Kharchenko V and Prusov D. Analysis of unmanned aircraft systems application in the civil field. Transport 2012; 27(3): 335–343.
4. Pujares G. Overview and current status of remote sensing applications based on unmanned aerial vehicles (UAVs). Photogramm Eng Rem S 2015; 81(4): 281–292.
5. Colomina I and Molina P. Unmanned aerial systems for photogrammetry and remote sensing: a review. ISPRS J Photogramm Rem S 2014; 92: 79–97.
6. Nex F and Remondino F. UAV for 3D mapping applications: a review. Appl Geomat 2014; 6(1): 1–15.
7. Nigam N, Bieniawski S, Kroo I, et al. Control of multiple UAVs for persistent surveillance: algorithm and flight test results. IEEE T Contr Syst T 2012; 20(5): 1236–1251.
8. Girard AR, Howell AS and Hedrick JK (eds). Border patrol and surveillance missions using multiple unmanned air vehicles. In: Proceedings of the 43rd IEEE conference on decision and control, 2004 CDC, Nassau, Bahamas, 14–17 December 2004. New York: IEEE.
9. Sun Z, Wang P, Vuran MC, et al. BorderSense: border patrol through advanced wireless sensor networks. Ad Hoc Netw 2011; 9(3): 468–477.
10. Haddal CC and Gertler J. Homeland security: unmanned aerial vehicles and border surveillance. DTIC document. ADA524297, 8 July 2010. Fort Belvoir, VA: Defense Technical Information Center.
11. Kanistras K, Martins G, Rutherford MJ, et al. Survey of unmanned aerial vehicles (UAVs) for traffic monitoring. In: Valavanis KP and Vachtsevanos GJ (eds) Handbook of unmanned aerial vehicles. Dordrecht: Springer, 2014, pp.2643–2666.
12. Srinivasan S, Latchman H, Shea J, et al. Airborne traffic surveillance systems: video surveillance of highway traffic. In: Proceedings of the ACM 2nd international workshop on video surveillance & sensor networks, New York, 15 October 2004. New York: ACM.
13. Kuiper E and Nadjm-Tehrani S (eds). Mobility models for UAV group reconnaissance applications. In: Proceedings of the international conference on wireless and mobile communications, 2006 (ICWMC’06), Bucharest, Romania, 29–31 July 2006. New York: IEEE.
14. Torres-Sánchez J, Peña J, De Castro A, et al. Multi-temporal mapping of the vegetation fraction in early-season
18. Tuna G, Nefzi B and Conte G. Unmanned aerial vehicle-aided communications system for disaster management applications. e & i Elektrotech Informationstechnik 2010; 127(3): 56–63.

19. Meijer S. Surface displacement of a Himalayan debris-covered glacier derived from automated cross-correlations using imagery acquired with unmanned aerial vehicles. Master’s Thesis, Utrecht University, Utrecht, 2015.

20. Eisenbeiss H and Sauerbier M. Investigation of UAV systems and flight modes for photogrammetric applications. Photogramm Rec 2011; 26(136): 400–421.

21. Jawhar I, Mohamed N, Al-Jaroodi J, et al. A framework for using unmanned aerial vehicles for data collection in linear wireless sensor networks. J Intell Robot Syst 2014; 74(1–2): 437.

22. Shaferman V and Shima T. Unmanned aerial vehicles cooperative tracking of moving ground target in urban environments. J Guid Control Dyn 2008; 31(5): 1360–1371.

23. Rodriguez-Canosa GR, Thomas S, del Cerro J, et al. A real-time method to detect and track moving objects (DATMO) from unmanned aerial vehicles (UAVs) using a single camera. Remote Sens 2012; 4(4): 1090–1111.

24. Prevost CG, Desbiens A and Gagnon E (eds). Extended Kalman filter for state estimation and trajectory prediction of a moving object detected by an unmanned aerial vehicle. In: Proceedings of the American control conference, 2007 (ACC’07), New York, 9–13 July 2007. New York: IEEE.

25. Saripalli S and Sukhatme GS. Landing on a moving target using an autonomous helicopter. In: Yuta S, Asama H, Prassler E, et al. (eds) Field and service robotics. Berlin, Heidelberg: Springer, 2003, pp.277–286.

26. Saripalli S, Montgomery JF and Sukhatme GS. Visually guided landing of an unmanned aerial vehicle. IEEE T Robotic Autom 2003; 19(3): 371–380.

27. Javidi B. Image recognition and classification: algorithms, systems, and applications. Boca Raton, FL: CRC Press, 2002.

28. Nguyen T, Park E-A, Han J, et al. Object detection using scale invariant feature transform. In: Pan JS, Krömer P and Snäsel V (eds) Genetic and evolutionary computing. Cham: Springer, 2014, pp.65–72.

29. Zhou H, Yuan Y and Shi C. Object tracking using SIFT features and mean shift. Comput Vis Image Und 2009; 113(3): 345–352.

30. Bay H, Tuyltaelars T and Van Gool L. SURF: speeded up robust features. In: Leonardis A, Bischof H and Pinz A (eds) Computer vision—ECCV 2006. Berlin, Heidelberg: Springer, 2006, pp.404–417.

31. Hosaka T, Kobayashi T and Otsu N. Object detection using background subtraction and foreground motion estimation. IPSJ Trans Comput Vis Appl 2011; 3: 9–20.

32. Comaniciu D, Ramesh V and Meer P. Kernel-based object tracking. IEEE T Pattern Anal 2003; 25(5): 564–577.

33. Tao H, Sawhney HS and Kumar R. Object tracking with Bayesian estimation of dynamic layer representations. IEEE T Pattern Anal 2002; 24(1): 75–89.

34. Sheikh Y and Shah M (eds). Bayesian object detection in dynamic scenes. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition, 2005 (CVPR 2005), San Diego, CA, 20–25 June 2005. New York: IEEE.

35. Yang C, Duraiswami R and Davis L (eds). Fast multiple object tracking via a hierarchical particle filter. In: Proceedings of the 2005 tenth IEEE international conference on computer vision, 2005 (ICCV 2005), Beijing, China, 17–21 October 2005. New York: IEEE.

36. Babenko B, Yang M-H and Belongie S. Robust object tracking with online multiple instance learning. IEEE T Pattern Anal 2011; 33(8): 1619–1632.

37. Berenji HR. Fuzzy logic controllers. In: Yager RR and Zadeh LA (eds) An introduction to fuzzy logic applications in intelligent systems. New York: Springer Science & Business Media, 1992, pp.69–96.

38. Saffiotti A. The uses of fuzzy logic in autonomous robot navigation. Soft Comput 1997; 1(4): 180–197.

39. Yager RR and Zadeh LA. An introduction to fuzzy logic applications in intelligent systems. New York: Springer Science & Business Media, 2012.

40. Algabri M, Mathkour H, Ramdane H, et al. Comparative study of soft computing techniques for mobile robot navigation in an unknown environment. Comput Hum Behav 2015; 50: 42–56.

41. Algabri M, Ramdane H, Mathkour H, et al. Optimization of fuzzy logic controller using PSO for mobile robot navigation in an unknown environment. Appl Mech 2014; 541–542: 1053–1061.

42. Algabri M, Mathkour H, Hedjar R, et al. Self-learning mobile robot navigation in unknown environment using evolutionary learning. J UCS 2014; 20(10): 1459–1468.

43. Rusu P, Petriu EM, Whalen TE, et al. Behavior-based neuro-fuzzy controller for mobile robot navigation. IEEE T Instrum Meas 2003; 52(4): 1335–1340.

44. Tsourveloudis NC, Doitsidis L and Valavanis KP. Autonomous navigation of unmanned vehicles: a fuzzy logic perspective. In: Kordic V, Lazinica A and Morden M (eds) Cutting edge robotics. Rijeka, Croatia: InTech, 2005, pp.291–310.

45. Dierks T and Jagannathan S. Output feedback control of a quadrotor UAV using neural networks. IEEE T Neural Netw 2010; 21(1): 50–66.

46. Sangyam T, Laohapiengsak P, Chongcharoen W, et al. Path tracking of UAV using self-tuning PID controller based on fuzzy logic. In: Proceedings of the SICE
Kurnaz S, Cetin O and Kaynak O. Adaptive neuro-fuzzy inference system based autonomous flight control of unmanned air vehicles. *Expert Syst Appl* 2010; 37(2): 1229–1234.

Doitsidis L, Valavanis KP, Tsourveloudis NC, et al. (eds). *A framework for fuzzy logic based UAV navigation and control*. In: *Proceedings of the ICRA ’04 2004 IEEE international conference on robotics and automation*, New Orleans, LA, 26 April–1 May 2004. New York: IEEE.

Olivares-Méndez M, Campoy P, Martínez C, et al. (eds). A pan-tilt camera fuzzy vision controller on an unmanned aerial vehicle. In: *Proceedings of the 2009 IROS 2009 IEEE/RSJ international conference on intelligent robots and systems*, St. Louis, MO, 10–15 October 2009. New York: IEEE.

Rada-Vilela J. *Fuzzylite: a fuzzy logic control library*. 2014, http://www.fuzzylite.com

Cheng X, Li N, Zhang S, et al. Robust visual tracking with SIFT features and fragments based on particle swarm optimization. *Circ Syst Signal Pr* 2014; 33(5): 1507–1526.

Wu Y, Lim J and Yang M-H. Object tracking benchmark. *IEEE T Pattern Anal* 2015; 37(9): 1834–1848.

Quigley M, Conley K, Gerkey B, et al. ROS: an open-source Robot Operating System. In: *Proceedings of the ICRA workshop on open source software*, Kobe, Japan, 12–17 May 2009.

Danelljan M, Hager G, Shahbaz Khan F, et al. (eds). Learning spatially regularized correlation filters for visual tracking. In: *Proceedings of the IEEE international conference on computer vision*, Santiago, Chile, 7–13 December 2015. New York: IEEE.

Monajemi M. *ardrone_autonomy: a ROS driver for ardrone 1.0 & 2.0*. 2012, https://github.com/AutonomyLab/ardrone_autonomy