



Navigation and Self-Semantic Location of Drones in Indoor Environments by Combining the Visual Bug Algorithm and Entropy-Based Vision

Darío Maravall*, Javier de Lope and Juan P. Fuentes

Department of Artificial Intelligence, Faculty of Computer Science, Universidad Politécnica de Madrid, Madrid, Spain

We introduce a hybrid algorithm for the self-semantic location and autonomous navigation of robots using entropy-based vision and visual topological maps. In visual topological maps the visual landmarks are considered as leave points for guiding the robot to reach a target point (robot homing) in indoor environments. These visual landmarks are defined from images of relevant objects or characteristic scenes in the environment. The entropy of an image is directly related to the presence of a unique object or the presence of several different objects inside it: the lower the entropy the higher the probability of containing a single object inside it and, conversely, the higher the entropy the higher the probability of containing several objects inside it. Consequently, we propose the use of the entropy of images captured by the robot not only for the landmark searching and detection but also for obstacle avoidance. If the detected object corresponds to a landmark, the robot uses the suggestions stored in the visual topological map to reach the next landmark or to finish the mission. Otherwise, the robot considers the object as an obstacle and starts a collision avoidance maneuver. In order to validate the proposal we have defined an experimental framework in which the visual bug algorithm is used by an Unmanned Aerial Vehicle (UAV) in typical indoor navigation tasks.

Keywords: visual bug algorithm, entropy search, visual topological maps, internal models, unmanned aerial vehicles

OPEN ACCESS

Edited by:

Jose Manuel Ferrandez,
Universidad Politécnica de Cartagena,
Spain

Reviewed by:

Félix De La Paz López,
Universidad Nacional de Educación a
Distancia (UNED), Spain

Roberto Iglesias Rodríguez,
Universidade de Santiago de
Compostela, Spain
Rafael Martínez Tomás,
Universidad Nacional de Educación a
Distancia (UNED), Spain

*Correspondence:

Darío Maravall
dmaravall@fi.upm.es

Received: 31 March 2017

Accepted: 14 August 2017

Published: 29 August 2017

Citation:

Maravall D, de Lope J and Fuentes JP
(2017) Navigation and Self-Semantic
Location of Drones in Indoor
Environments by Combining the Visual
Bug Algorithm and Entropy-Based
Vision. *Front. Neurobot.* 11:46.
doi: 10.3389/fnbot.2017.00046

INTRODUCTION

Some of the most challenging behaviors of autonomous robots are related to navigation tasks. According to a policy of safe navigation, the basic level of behaviors are devoted by strategies that allow the obstacle detection and the collision avoidance. Once these tasks have been conveniently solved, the next level is route planning, i.e., the generation of routes or paths that allow the robot to reach specific places in the environment. Some information about the environment is needed for such kind of planning. This information is managed by the robot control system and concerns the fundamental issue of environment mapping. The robot can store a sort of metric data related to the environment, topological information, including relationships between elements of the environment and features, maybe visual, associated to them, or any combination of both.

The aim of our work is to introduce an efficient method for autonomous robot navigation supported by visual topological maps (Maravall et al., 2013a), in which the coordinates of both the robot and the goal are not needed. More specifically, this novel method is meant to drive the robot toward a target landmark using self-semantic location, while simultaneously avoiding any existing obstacle using exclusively vision capacity. By self-semantic location we refer to the concept used generally in topological models that differs from metric navigation models. Therefore, self-semantic location means that the robot control system is able to determine an approximate, semantic self-location when a landmark already presented in the model is detected and recognized. In summary, self-semantic location refers to the cognitive information associated to a particular place of the environment (e.g., “I am at the restaurant,” so that the robot has a specific cognitive framework at this particular place of the environment).

The proposed algorithm is based on a conventional bug algorithm, although in our version we use only visual information as opposite to the classic versions that employ metric information. As it is well-known the bug algorithms are a family of techniques for obstacle avoidance in robot navigation with metric maps in real-time (Lumelsky and Stepanov, 1987; Lumelsky, 2005). These techniques make the robot head toward the goal and, if an obstacle is encountered, it circumnavigates it and remember how close it gets to the goal. Once the obstacle is avoided, the robot returns to the closest point and continues toward the goal. The main drawback of the conventional metric bug algorithms is that they need the knowledge of the robot localization (the hardest constraint) besides the coordinates of the goal in a common reference framework. To update the robot's position coordinates is necessary the use of external positioning systems.

Our method focuses on the search of the target visual landmark based on the entropy maximization of the images captured by the robot (Fuentes et al., 2014), which is used when the robot is in an unknown localization. Our hypothesis is that there is a direct and positive correlation between the entropy of an image and the probability of this image containing one or several objects: the higher the entropy, the higher its probability of containing several objects inside it; and conversely, the lower the entropy, the higher its probability of containing a single object inside it. Afterwards, when the robot has detected a potential landmark, i.e., it is in a known location, a dual architecture is executed with inspiration on cerebellar system of living beings combined with a brain activity. This dual architecture provides the reactive and anticipatory behaviors for the robot autonomous control.

Although the technique can be applied to any type of autonomous robot, we employ an Unmanned Aerial Vehicle (UAV). The application of our algorithm to these flying robots is justified due to occasionally the UAVs only have available an onboard camera as main input sensor for obstacle detection and landmark and location recognition as well as other specific sensors for flying related matters. This is mainly the case with Micro Aerial Vehicles (MAVs).

The rest of the paper is organized as follows. The next section summarizes some related work with the main findings in this work, namely navigation based on topological maps and obstacle detection and avoidance. After the theoretical foundations of the entropy-based search combined with the bug algorithm, we present the experimental work performed for its validation using an UAV. The paper ends with some suggestions for future research work based on the use of the vehicle onboard cameras for vision-based quality inspection and defects detection in different operating environments.

RELATED WORK

Our proposal involves the use of visual graphs, in which each node stores images associated to landmarks, and the arcs represent the paths that the UAV must follow to reach the next node. Therefore, these graphs can be used to generate the best path for an UAV to reach a specific destination, as it has been suggested in other works. Practically each traditional method used in ground robots for trajectory planning has been considered for aerial ones (Goerzen et al., 2010). Some of those methods use graph-like models and generally they use algorithms such as improved versions of the classic A* (MacAllister et al., 2013; Zhan et al., 2014) and Rapidly-exploring Random Tree Star (RRT) (Noreen et al., 2016) or reinforcement learning (RL) (Sharma and Taylor, 2012) for planning. RL is even used by methods that consider the path-planning task in cooperative multi-vehicle systems (Wang and Phillips, 2014), in which coordinated maneuvers are required (Lopez-Guede and Graña, 2015).

On the other hand, the obstacle avoidance task is also addressed here, which is particularly important in the UAV domain because of a collision in flight surely implies a danger and the partial or total destruction of the vehicle. Thus, the Collision Avoidance System (CAS) (Albaker and Rahim, 2009; Pham et al., 2015) is a fundamental part of control systems. Its goal is to allow UAVs to operate safely within the non-segregated civil and military airspace on a routinely basis. Basically, the CAS must detect and predict traffic conflicts in order to perform an avoidance maneuver to avoid a possible collision. Specific approaches are usually defined for outdoor or indoor vehicles. Predefined collision avoidance based on sets of rules and protocols are mainly used outdoors (Bilimoria et al., 1996) although classic methods such as artificial potential fields are also employed (Gudmundsson, 2016). These and other conventional well-known techniques in wheeled and legged robots are also considered for being used in UAVs (Bhavesh, 2015).

Since most of the current UAVs have monocular onboard cameras as main source of information several computer vision techniques are used. A combination of the Canny edge detector and the Hough transform is used to identify corridors and staircases for trajectory planning (Bills et al., 2011). Also, feature points detectors such as SURF (Aguilar et al., 2017) and SIFT (Al-Kaff et al., 2017) are used to analyze the images and to determine free collision trajectories. However, the most usual technique is

optic flow (Zufferey and Floreano, 2006; Zufferey et al., 2006; Beyeler et al., 2007; Green and Oh, 2008; Sagar and Visser, 2014; Bhavesh, 2015; Simpson and Sabo, 2016). Sometimes optic flow is combined with artificial neural networks (Oh et al., 2004) or other computer vision techniques (Soundararaj et al., 2009). Some of those techniques (de Croon, 2012) are based on the analysis of the image textures for estimating the possible number of objects that are captured. Finally, as in other scopes of research, deep learning techniques are also being used to explore alternatives to the traditional approaches (Yang et al., 2017).

THE VISUAL BUG ALGORITHM

In words the visual bug algorithm can be summarized as follows: “search the target landmark and once it is visible then move always toward the target landmark and circumnavigate any existing obstacle when necessary.”

Besides the implementation of the visual search procedure (which is obviously less critical when the robot controller is provided with the orientation Θ to the target landmark), the critical element is the robot odometry (Lumelsky and Stepanov, 1986) and more specifically the robot's dead reckoning or path integration, for this reason the robot planner is then based in the control of the orientation Θ toward the target landmark using topological maps.

Figure 1 shows the pseudocode of the visual bug algorithm (Maravall et al., 2015a):

Besides the basic procedure visual search, this algorithm consists of three additional complex vision-based robot behaviors:

- B1: “go forward to the goal”

- B2: “circumnavigate an obstacle”
- B3: “turns the orientation Θ ”

There are defined three basic environment's states or situations:

- S1: “the goal is visible”
- S2: “there is an obstacle in front of the robot”
- S3: “a landmark from the topological map is visible”

The visual bug algorithm sets out also hard computer vision problems for both the perception of the environment's states S_1 , S_2 , and S_3 and for the implementation of the three basic robot's behaviors B_1 , B_2 , and B_3 , so that it is devoted the bulk of the remaining sections of the paper to describe our proposals aimed at solving these specific computer vision problems.

VISUAL LANDMARKS SEARCH AND RECOGNITION FOR ROBOT SELF-SEMANTIC LOCATION USING VISUAL TOPOLOGICAL MAPS

As it was already pointed out in the pseudocode of the visual bug algorithm, and as it is shown in **Figure 2**, the visual topological map is a connected graph where each node stores images corresponding to the next nodes/landmarks to which it is connected, and toward which the robot must navigate. Besides, the arcs connecting two nodes stores information about the relative orientation, Θ , of one node/landmark with respect to the other. Hence, this hybrid visual and “odometric information” (just orientations and landmarks) can be used by the robot's navigation module for its self-semantic location, allowing

Given a visual topological map (nodes store the landmarks images and arcs store the orientations linking the landmarks) or more specifically, given the image of the target landmark and given the orientation Θ towards the next landmark provided by the topological map:

While the robot does not detect the target landmark specified in the topological map **do**

1. Activate the visual search procedure in a controlled way based on entropy maximization of the captured digital images (*high entropy*).

2. **If** an obstacle is encountered (*low entropy*) **then**

Move following the obstacle contour until the robot meets a leaving point L (which is linked to the goal without hitting the current obstacle itself, using the correct orientation to the next landmark).

3. **If** a landmark is visible **then**

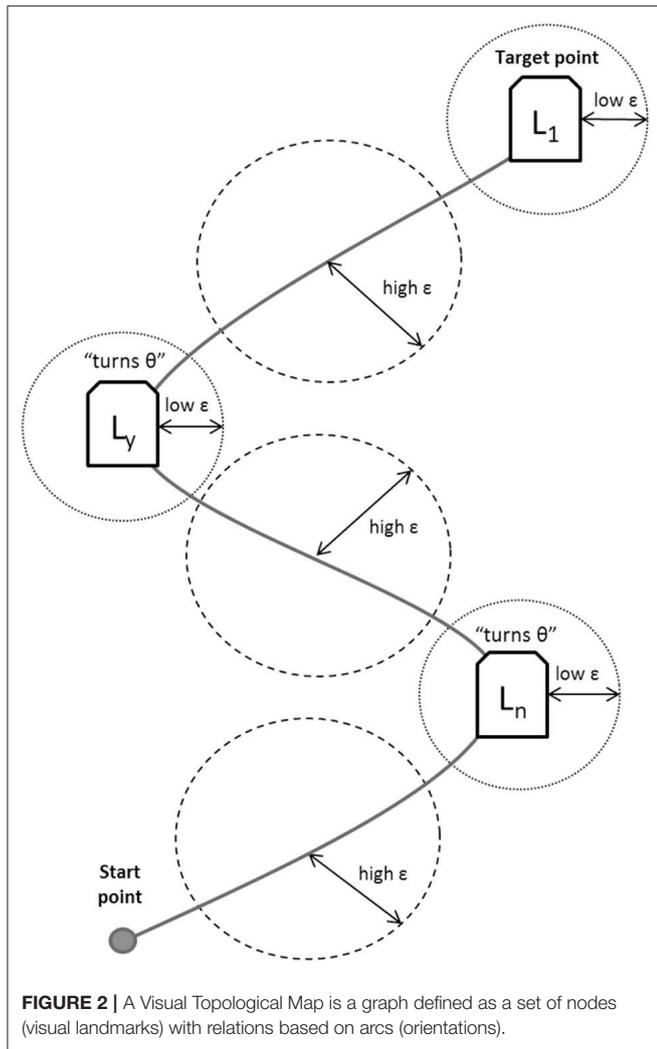
Go forward to the landmark until either happens:

(a) The target landmark is reached, then exit.

(b) Other landmark is reached, then the robot turns the orientation Θ defined in its arc.

End While

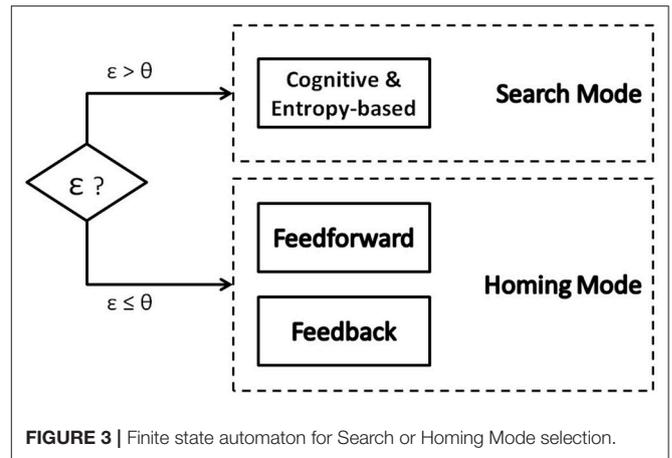
FIGURE 1 | Visual Bug Algorithm.



the search and detection of the sequential nodes/landmarks belonging to a specific route.

Notice to the existence of unknown states (high error ϵ) in which the robot uses entropic vision; known states (low error ϵ) are used when the robot is able to determinate its position with respect to the L_n landmark.

In the sequel a novel method for this self-semantic location task is proposed (i.e., the task of landmark search and recognition; Maravall et al., 2013b) based on the combination of image entropy for landmark search (*Search Mode*), and a dual feedforward/feedback vision-based control loop (*Homing Mode*) for the final landmark homing. **Figure 3** shows the finite state automaton that models the robot's controller between the *Search Mode* [S] and the *Homing Mode* [H] depending on the magnitude of the error ϵ or the difference images (respect to the landmarks defined in the topological map) and according to the simple heuristic rule: If {error is big} Then {S} Else {H}. As previously mentioned, once the image containing the candidates for the target landmark has been obtained by the entropy maximization process in the *Search*



Mode, the robot's controller switches to the *Homing Mode* to guide the UAV toward the target landmark (Maravall et al., 2015a).

Search Mode: Entropy-Based Landmark Search

The main idea behind the entropy-based search is the direct and positive correlation between the entropy of an image and the probability of the image of containing several objects inside (in the case of high entropy) or conversely the probability of the image of containing just a single object (in the case of low entropy) (Fuentes et al., 2014). **Figure 4** shows an example of this idea.

As visual landmarks in topological maps are usually selected as outstanding single objects, normally surrounded by other objects that produce together complex images with a high entropy, the task of visual landmarks search and detection can be formalized as a sequential process of image entropy maximization aimed at converging to an image of high entropy, hopefully containing several candidates of the single target object landmark, followed by a homing process aimed at guiding the robot toward the target landmark by means of a vision-based loop control, as explained below.

Therefore if it is represented by u the robot's control variables (angles of movement as *pitch*, *roll* and *yaw*) this process of entropy maximization can be expressed as follows:

$$\dot{u} = + \frac{\partial H[\text{Hist}(I_k)]}{\partial u} \tag{1}$$

where H is the image entropy, as given by the standard definition (Shannon, 1948) of the entropy of the normalized histogram $\text{Hist}(I_k)$:

$$H[\text{Hist}(I_k)] = - \sum [\text{Hist}(I_k) \cdot \log_2 \text{Hist}(I_k)] \tag{2}$$

These control signals are obtained along the k trials, during the operation of the robot in the flight environment.

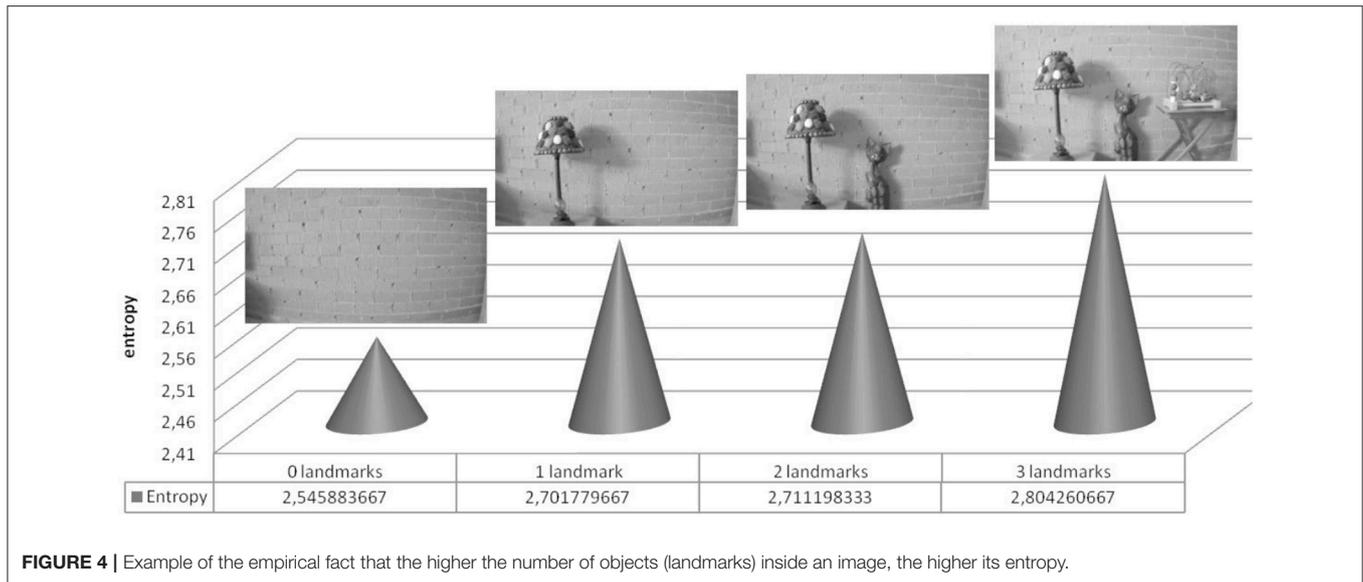


FIGURE 4 | Example of the empirical fact that the higher the number of objects (landmarks) inside an image, the higher its entropy.

Homing Mode: The Feedforward/Feedback Controller for Landmark Homing

This homing mode has been implemented as a dual feedforward/feedback control architecture which is constituted by the combination of a feedback module (either based on a conventional PD control or on error gradient control), and a feedforward module (based on either a neurocontroller or a memory-based controller).

This dual control architecture is shown in **Figure 5**, which shows the block-diagram of the dual feedforward/feedback controller (Kawato, 1990). Notice that the feedback or reactive controller receives as input the ϵ error; this vision-based error signal ϵ is obtained as the difference between the histogram of the recognized landmark or histogram of the goal image $Hist(I_g)$ and the histogram of the current image $Hist(I_k)$ during the k iteration of the controller.

Nevertheless, the feedforward or anticipatory controller receives as input the histogram of the current image $Hist(I_k)$. The feedback controller is implemented as a conventional PD control algorithm (Maravall et al., 2013a) whose control parameters are set experimentally. On the other hand, the feedforward controller is based on an inverse model (Kawato, 1999) using a conventional neural network based on multilayer perceptron (Maravall et al., 2015b), which is trained (using the output of the feedback controller) when error signal ϵ has been reduced in the last iterations. Both output control signals u_{fb} and u_{ff} are combined as follows:

$$u_t = w_{fb} \cdot u_{fb} + w_{ff} \cdot u_{ff} \tag{3}$$

The weight parameters w_{fb} and w_{ff} are set experimentally along the trials with the UAV. The u_t is the vector control signals (*pitch*, *roll*, and *yaw*) which is sent to the robot.

Summarizing, when the error ϵ is high, the robot is around an unknown state and it executes the entropy-based controller

as it has been detailed in the previous section. When the robot recognizes the current location and it is able to detect any landmark already stored in the topological map, it follows the commands generated by the dual feedforward/feedback controller.

EXPERIMENTAL WORK: UAV NAVIGATION BASED ON THE VISUAL BUG ALGORITHM PLUS LANDMARKS SEARCH THROUGH IMAGE ENTROPY

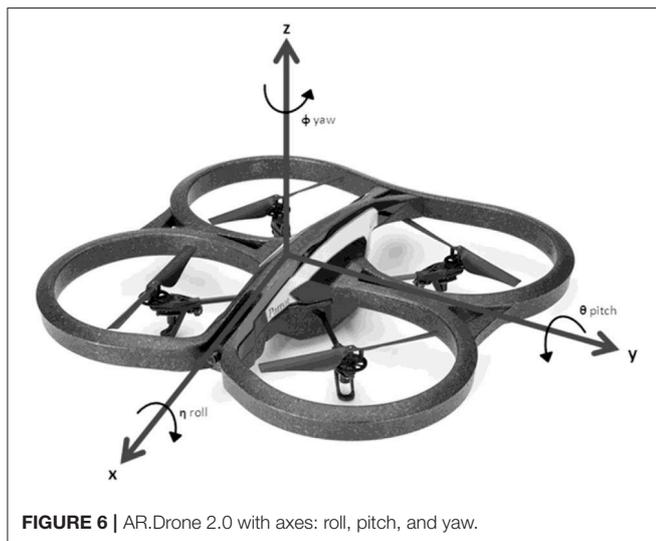
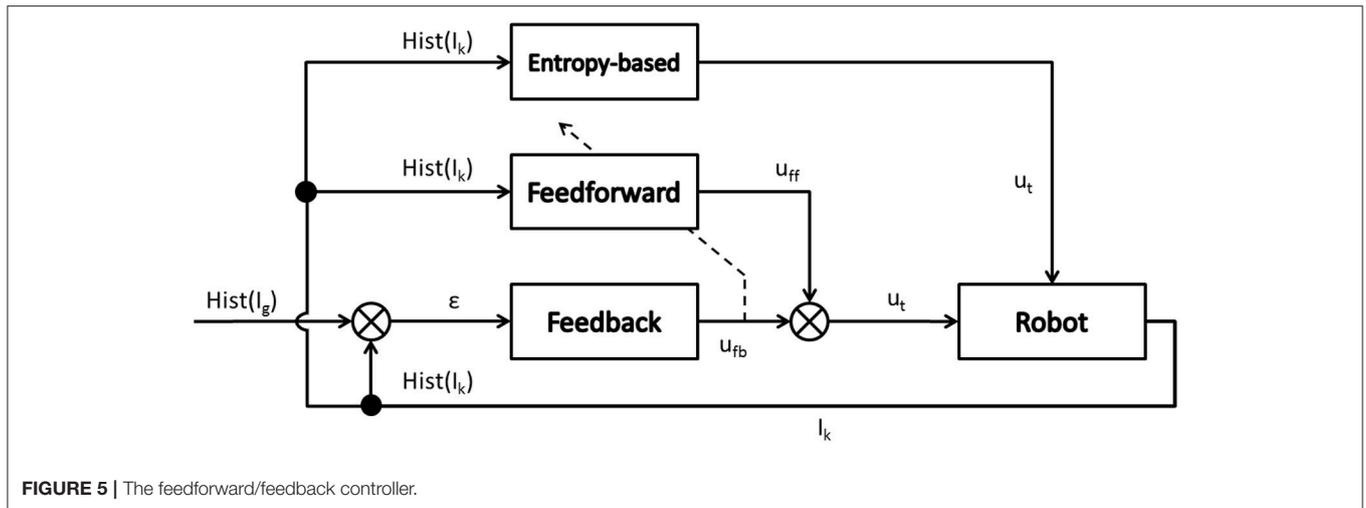
AR.Drone

For the experimental work concerning the testing and validation of the proposed method for visual landmarks search and detection, it has been used the quadrotor Parrot AR.Drone 2.0 (**Figure 6**) as UAV well-established and widely available robotics research platform (Krajník et al., 2011).

The Parrot company launched the project named AR.Drone with the final objective of producing a micro UAV aimed at both the mass market of video games and the home entertainment. AR.Drone 2.0 has been finally released on the market and it is widely available at a low price that make it possible to be used as a unique robotic platform for experimental work on UAVs.

All commands and images can be exchanged with a central controller via an *ad-hoc* Wifi connection. The AR.Drone has an on-board HD camera and it has four motors to fly through the environment. This UAV supports four different control signals or degrees of freedom along the usual axes (*roll*, *pitch*, *gaz*, and *yaw*). The different maneuvers can be executed by the axes grades (*roll*, *pitch*, *gaz*, and *yaw*). The *gaz* variable regulates the altitude control.

This UAV has been extensively used for autonomous navigation at indoor environments (Maravall et al., 2013a).



Description of the Experiments

For the validation of the proposed algorithm, it has been selected a typical indoor environment (see **Figure 7**). In an emergency situation, an UAV can help people to find an exit door along a secure route. For this, a topological map has been defined for the representing the environment, and to verify how the UAV is able to search and reach an exit door when an emergency situation occurs (for example fire in a building, flood in a home, etc.).

Figure 7 shows the environment used during the experiments with the UAV, for which it is defined a visual topological map using a set of landmarks situated along the environment: (a) the experimental indoor environment schema and (b) its associated visual topological map. The UAV navigates along the environment using the entropy-based controller, and when a landmark (a node) is detected through the dual controller, the UAV performs the control signals stored (in the arc), to guide it to the next landmark defined in the map.

For this scenario, the topological map defines three landmarks L_n : the landmark L_3 is an open door, the landmark L_2 is a fire extinguisher and the landmark L_1 or landmark goal is an exit door. **Figure 7** also shows the approximate situation of each landmark along the environment.

Discussion of the Experimental Results

The UAV is located on a given start point (unknown state) in the experimental indoor environment, at a given distance near with respect to a specified landmark in the topological map. The UAV aims reach this first landmark using the entropy-based controller because the error ε is high.

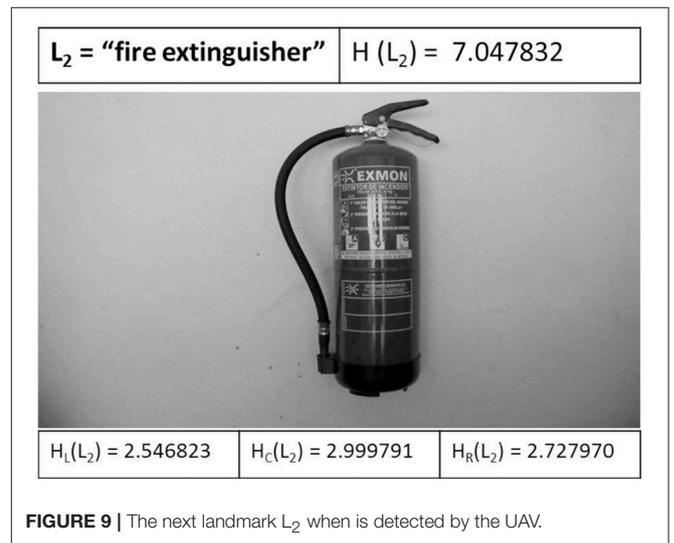
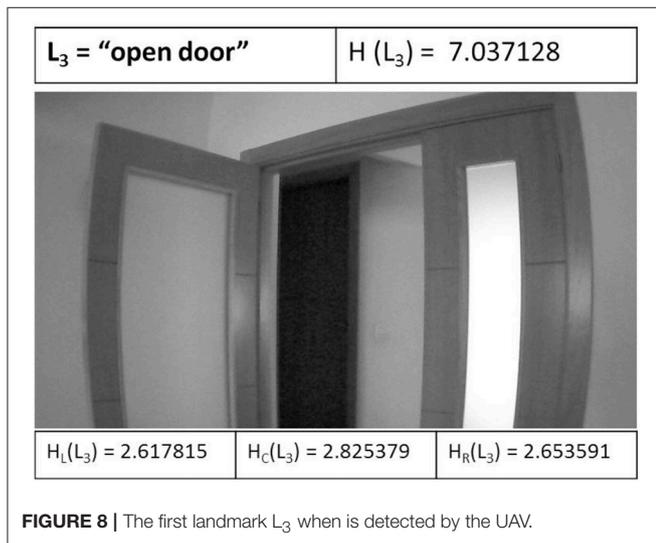
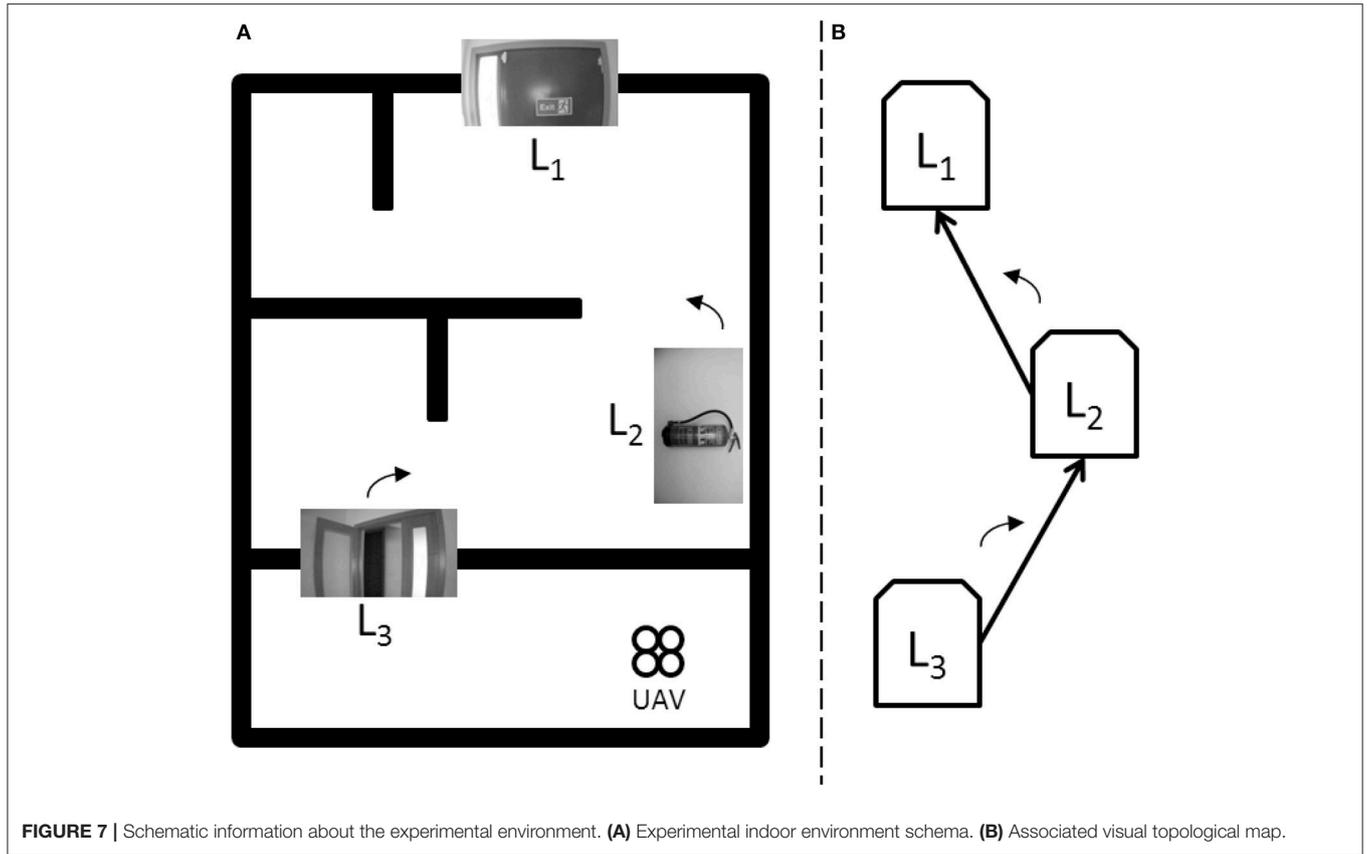
Through the entropy-based controller, it is executed a process of image entropy maximization (Search Mode) aimed at converging to a state of high entropy, hopefully containing landmarks from the visual topological map. This controller uses three entropic values: the left zone (H_L), the right zone (H_R), and the central zone (H_C) of image captured from the UAV.

When the UAV is near of the first landmark L_3 , the error ε decreases, and it is switched to the Homing Mode, through the dual feedforward/feedback controller. Therefore, the dual controller starts generating control signals u_t increasingly optimal, for send them to the UAV.

The signals u_{fb} have been calculated by the feedback controller (reactive behavior) from the different images that are captured by the UAV's onboard camera, as well as the signals u_{ff} provided by the feedforward controller (anticipatory behavior). Both signals have been consolidated adaptively through interaction of the robot with its environment.

Figure 8 shows the first landmark L_3 when is detected by the UAV. The entropic values are shown: the full entropy of image $H(L_3)$, the left entropy $H_L(L_3)$, the center entropy $H_C(L_3)$, and the right entropy $H_R(L_3)$ of the image captured. $H_C(L_3)$ is the higher entropy, therefore the UAV executes a maneuver go forward to the open door.

The UAV recognizes the L_3 landmark at the $k=4$ iteration. When the UAV reaches the specified landmark, it will proceed to execute the specified maneuver in the corresponding arc in



the topological map. It is considered that the approximation maneuver has been executed correctly and the UAV has been able to approximate to this landmark for its identification. The arc stores the corresponding orientation Θ , for which the UAV performs a forward maneuver (*pitch*) plus a right turn (*yaw*), passing this way the open door that

has been identified. The attitude control (*gaz*) remains constant throughout the experiment (about 150 cm above the ground).

At this point the UAV is in an unknown state again and the entropy-based controller is activated. After several iterations, the robot locates a state with higher entropy in the environment and

executes an approximation maneuver to the fire extinguisher. When the error ε decreases, it is switched to the dual controller and the L_2 landmark is detected at the $k = 11$ iteration. The UAV performs a left turn (*yaw*) to the next landmark defined in the visual topological map.

Then, **Figure 9** shows the next landmark L_2 when is detected by the UAV. The entropic values are shown: the full entropy of image $H(L_2)$, the left entropy $H_L(L_2)$, the center entropy $H_C(L_2)$, and the right entropy $H_R(L_2)$ of the image captured. $H_C(L_2)$ is the higher entropy, therefore the UAV executes a maneuver go forward to the fire extinguisher.

Finally, the UAV reaches the target point or L_1 landmark goal defined at the $k = 19$ iteration, and executes the specified maneuver of this arc: go forward to the goal, showing the exit door to an emergency situation.

Figure 10 shows the target landmark L_1 when is detected by the UAV. The entropic values are shown: the full entropy of

image $H(L_1)$, the left entropy $H_L(L_1)$, the center entropy $H_C(L_1)$, and the right entropy $H_R(L_1)$ of the image captured. $H_C(L_1)$ is the higher entropy, therefore the UAV executes a maneuver go forward to the exit door.

During the experimentation, the UAV has used the *pitch* actuator (forward/back), the *gaz* actuator (up/down), and *yaw* actuator (rotation on its axis z). The values of the control signals u_t {*pitch*, *yaw*} that has been generated in each k iteration executed during the experiments, are shown in **Figure 11**. The control signals are generated (*pitch* and *yaw*) at each k iteration during the approximation maneuver of the UAV, from start point until to reach the target point (exit door). The landmarks L_3 , L_2 , and L_1 are detected by the UAV at iteration $k = 4$, $k = 11$, and $k = 19$ respectively.

The entropy-based controller generates the values of the control signals through image entropy maximization (entropic vision), performing a maneuver for guide the robot to the higher entropy state in each iteration. When the left zone of image has the higher entropy (H_L) the robot performs a turn to the left (*yaw* = [-1..0]), else if the right zone of image has the higher entropy (H_R) the robot performs a turn to the right (*yaw* = [0..1]) in this case. If the higher entropy is centered on the image (H_C), the robot goes forward (*pitch* = [-1..0]). The range of values for *yaw* and *pitch* are defined by the AR.Drone SDK (Piskorski et al., 2012).

The dual controller has generated the values of control signals adaptively through interaction of the robot with its environment. Experimentally, for the calculation of the combined signal u_t , it has been established the following weight values: $w_{fb} = 0.7$ and $w_{ff} = 0.3$, for the feedback and feedforward controllers respectively (Maravall et al., 2015b).

From the experimental results obtained in our laboratory it is concluded that the UAV is able to successfully perform in real time the fundamental skills of the visual bug algorithm, guiding the robot toward a goal landmark (in this case exit door) using self-semantic location in each landmark defined in the visual topological map.

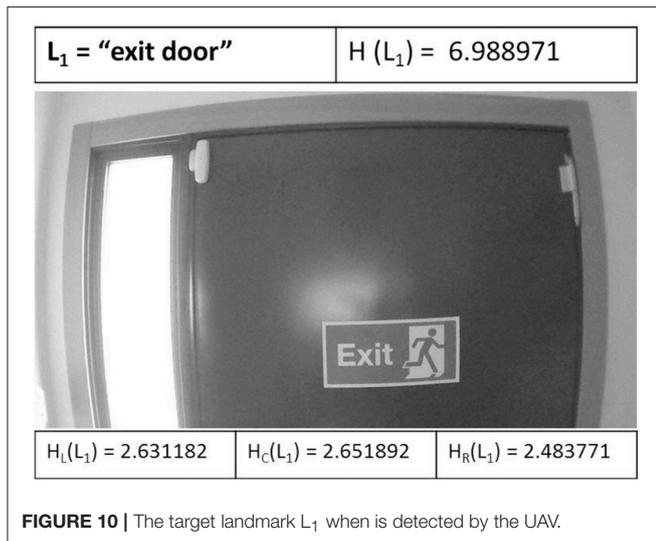


FIGURE 10 | The target landmark L_1 when is detected by the UAV.

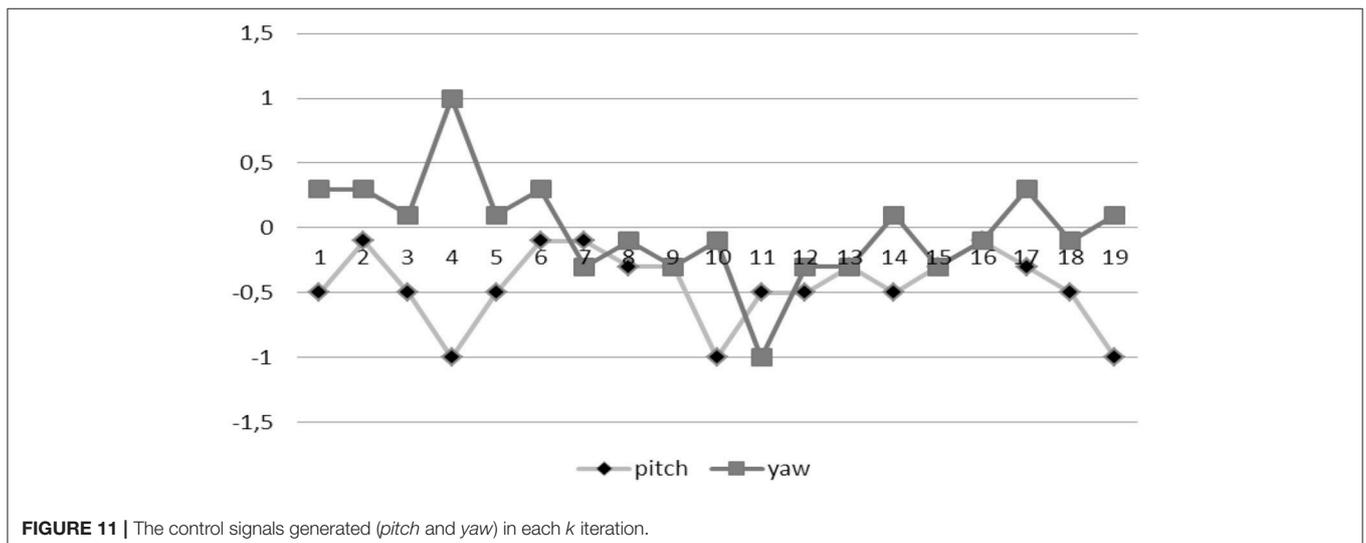


FIGURE 11 | The control signals generated (*pitch* and *yaw*) in each k iteration.

CONCLUSIONS AND FUTURE WORK

A hybrid algorithm for the self-semantic location and autonomous navigation of a robot based on entropic vision and the visual bug algorithm has been presented and tested in a scenario corresponding to a hypothetical emergency situation. The proposed algorithm uses a visual topological map to autonomously navigate in the environment. The nodes in the topological map determine a leave-point or a landmark for the self-location of the robot, in which the robot must re-orient its navigation in order to reach the goal. Unlike the classic bug algorithms, our algorithm does not require any knowledge about the robot's coordinates in the environment since the robot uses its own self-location method during the navigation to know its position in each iteration.

Based on the experimental results, it is concluded that this hybrid algorithm is highly robust when the robot is around an unknown location. The robustness is provided by the concept of entropic vision and the search of zones with high entropy. It is empirically confirmed the direct and positive correlation between the entropy of an image and the probability of the image of containing several objects inside. The performance when the robot encounters an obstacle during its navigation is acceptable, using the maximization of visual entropy as strategy. In addition, both techniques (the visual bug algorithm

and the visual topological maps) together are able to increase the overall solution performance, reducing the number of iterations along the time for reach the goal landmark defined previously.

Future work is planned toward implementation of this hybrid algorithm on other situations in the real world, which an engineering process as self-semantic location of robots is needed: security in building, surveillance of frontiers or critical infrastructure control. We plan to develop further research work concerning the use of the UAV's onboard cameras for vision-based quality inspection and defects detection, taking profit of our experience in vision-based quality inspection and defects detection in the manufacturing industry, where we have introduced the novel concept of the histogram of connected elements as a generalization of the conventional gray level images histogram.

AUTHOR CONTRIBUTIONS

The Authors DM, JdL, and JPF have been working about the paper titled: "Navigation and self-semantic location of drones in indoor environments by combining the visual bug algorithm and entropy-based vision" in the following tasks: definition of abstract, introduction, the dual FF/FB architecture, entropic-vision, and experimental works with drones.

REFERENCES

- Aguilar, W. G., Casaliglla, V. P., and Polit, J. L. (2017). Obstacle avoidance based-visual navigation for micro aerial vehicles. *Electronics* 6:10. doi: 10.3390/electronics6010010
- Albaker, B. M., and Rahim, N. A. (2009). "A survey of collision avoidance approaches for unmanned aerial vehicles," in *Proc. IEEE Int. Conf. Technical Postgraduates* (Kuala Lumpur).
- Al-Kaff, A., García, F., Martín, D., De La Escalera, A., and Armingol, J. M. (2017). Obstacle detection and avoidance system based on monocular camera and size expansion algorithm for UAVs. *Sensors* 17:e1061. doi: 10.3390/s17051061
- Beyeler, A., Zufferey, J.-C., and Floreano, D. (2007). "3D vision-based navigation for indoor microflyers," in *IEEE Int. Conf. Robotics and Automation* (Rome), 1336–1341.
- Bhaves, V. A. (2015). Comparison of various obstacle avoidance algorithms. *Int. J. Eng. Res. Technol.* 4, 629–632. doi: 10.17577/IJERTV4IS120636
- Bilimoria, K., Sridhar, B., and Chatterji, G. (1996). "Effects of conflict resolution maneuvers and traffic density of free flight," in *Proc. AIAA Guidance, Navigation, and Control Conference* (San Diego, CA).
- Bills, C., Chen, J., and Saxena, A. (2011). "Autonomous MAV and flight in indoor environments using single image perspective cues," in *Proc IEEE Int. Conf on Robotics and Automation* (Shanghai).
- de Croon, G. C. H. E., de Weerd, E., De Wagter, C., Remes, B. D. W., and Ruijsink, R. (2012). The appearance variation cue for obstacle avoidance. *IEEE Trans. Robot.* 28, 529–534. doi: 10.1109/TRO.2011.2170754
- Fuentes, J. P., Maravall, D., and de Lope, J. (2014). "Entropy-based search combined with a dual feedforward/feedback controller for landmark search and detection for the navigation of a UAV using visual topological maps," in *Proc. Robot 2013: First Iberian Robotics Conference. Springer Series on Advances in Intelligent Systems and Computing, Vol. 233*, (Heidelberg: Springer), 65–76.
- Goerzen, C., Kong, Z., and Mettler, B. (2010). A survey of motion planning algorithms from the perspective of autonomous UAV guidance. *J. Intell. Robot. Syst.* 57, 65–100. doi: 10.1007/s10846-009-9383-1
- Green, W. E., and Oh, P. Y. (2008). "Optic-flow-based collision avoidance: applications using a hybrid MAV," in *IEEE Robotics and Automation Magazine*, 96–103.
- Gudmundsson, S. (2016). *A Biomimetic, Energy-Harvesting, Obstacle-Avoiding, Path-Planning Algorithm for UAVs*. Ph.D. thesis Dissertation, Department of Mechanical Engineering, Embry-Riddle Aeronautical University. Available online at: <http://commons.erau.edu/edt/306>
- Kawato, M. (1990). "Feedback-error-learning neural network for supervised motor learning," in *Advanced Neural Computers*, ed R. Eckmiller (Amsterdam: Elsevier), 365–372.
- Kawato, M. (1999). Internal models for motor control and trajectory planning. *Neurobiology* 9, 718–727. doi: 10.1016/S0959-4388(99)00028-8
- Krajník, T., Vonásek, V., Fišer, D., and Faigl, J. (2011). "AR-drone as a platform for robotic research and education," in *Research and Education in Robotics - EUROBOT 2011. EUROBOT 2011. Communications in Computer and Information Science*, Vol. 161, eds D. Obdržálek and A. Gottscheber (Berlin; Heidelberg: Springer). doi: 10.1007/978-3-642-21975-7_16
- Lopez-Guede, J. M., and Graña, M. (2015). "Neural modeling of hose dynamics to speedup reinforcement learning experiments," in *Bioinspired Computation in Artificial Systems* (Elche), 311–319.
- Lumelsky, V. J. (2005). *Sensing, Intelligence, Motion: How Robots and Humans Move in an Unstructured World*. New York, NY: John Wiley & Sons.
- Lumelsky, V. J., and Stepanov, A. A. (1986). "Dynamic path planning for a mobile automaton with limited information on the environment," in *IEEE Trans. on Automatic Control*, Vol. AC-31.
- Lumelsky, V. J., and Stepanov, A. A. (1987). Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. *Algorithmica* 2, 403–430.
- MacAllister, B., Butzke, J., Kushleyev, A., Pandey, H., and Likhachev, M. (2013). "Path planning for non-circular micro aerial vehicles in constrained environments," in *Proceeding IEEE International Conference on Robotics and Automation* (Karlsruhe).
- Maravall, D., de Lope, J., and Fuentes, J. P. (2013a). "A vision-based dual anticipatory/reactive control architecture for indoor navigation of an unmanned aerial vehicle using visual topological maps," in *Natural and Artificial Computation in Engineering and Medical Applications, LNCS 7931*, (Heidelberg: Springer), 66–72.

- Maravall, D., de Lope, J., and Fuentes, J. P. (2013b). Fusion of probabilistic knowledge-based classification rules and learning automata for automatic recognition of digital images. *Pattern Recogn. Lett.* 34, 1719–1724. doi: 10.1016/j.patrec.2013.03.019
- Maravall, D., de Lope, J., and Fuentes, J. P. (2015a). “Visual bug algorithm for simultaneous robot homing and obstacle avoidance using visual topological maps in an unmanned ground vehicle,” in *Bio-inspired Computation in Artificial and Natural Systems, LCNS 9108*, (Heidelberg: Springer), 301–310.
- Maravall, D., de Lope, J., and Fuentes, J. P. (2015b). Vision-based anticipatory controller for the autonomous navigation of an UAV using artificial neural networks. *Neurocomputing* 151, 101–107. doi: 10.1016/j.neucom.2014.09.077
- Noreen, I., Khan, A., and Habib, Z. (2016). Optimal path planning using RRT* based approaches: a survey and future directions. *Int. J. Adv. Comput. Sci. Appl.* 7, 97–107. doi: 10.14569/IJACSA.2016.071114
- Oh, P., Green, W. E., and Barrows, G. (2004). “Neural nets and optic flow for autonomous micro-air-vehicle navigation,” in *Proc. Int. Mechanical Engineering Congress and Exposition (Anaheim, CL)*.
- Pham, H., Smolka, S. A., Stoller, S. D., Phan, D., and Yang, J. (2015). A survey on unmanned aerial vehicle collision avoidance systems. *arXiv:1508.07723*.
- Piskorski, S., Brulez, N., and D’Haeyer, F. (2012). *AR-Drone Developer Guide Sdk 2.0*.
- Sagar, J., and Visser, A. (2014). “Obstacle avoidance by combining background subtraction, optical flow and proximity estimation,” in *Proc. IMAV-2014 (Delft)*, 142–149.
- Shannon, C. E. (1948). A mathematical theory of communication. *Bell Syst. Tech. J.* 27, 379–423, 623–656. doi: 10.1002/j.1538-7305.1948.tb00917.x
- Sharma, S., and Taylor, M. E. (2012). “Autonomous waypoint generation strategy for on-line navigation in unknown environments,” in *IROS Workshop on Robot Motion Planning: Online, Reactive, and in Real-Time (Vilamoura-Algarve)*.
- Simpson, A., and Sabo, C. (2016). MAV obstacle avoidance using biomimetic algorithms. *Proc. AIAA SciTech (San Diego, CL)*.
- Soundararaj, S. P., Sujeeth, A. K., and Saxena, A. (2009). “Autonomous indoor helicopter flight using a single onboard camera,” in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (St. Louis)*, 5307–5314.
- Wang, Q., and Phillips, C. (2014). Cooperative path-planning for multi-vehicle systems. *Electronics* 3, 636–660. doi: 10.3390/electronics3040636
- Yang, S., Konam, S., Ma, C., Rosenthal, S., Veloso, M., and Scherer, S. (2017). Obstacle avoidance through deep networks based intermediate perception. *arXiv:1704.08759*.
- Zhan, W., Wang, W., Chen, N., and Wang, C. (2014). Efficient UAV path planning with multiconstraints in a 3D large battlefield environment. *Math. Probl. Eng.* 2014:597092. doi: 10.1155/2014/597092
- Zufferey, J.-C., and Floreano, D. (2006). Fly-inspired visual steering of an ultralight indoor aircraft. *IEEE Trans. Robot.* 22, 137–146. doi: 10.1109/TRO.2005.858857
- Zufferey, J.-C., Klapotcz, A., Beyeler, A., Nicoud, J.-D., and Floreano, D. (2006). “A 10-gram microflyer for vision-based indoor navigation,” in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2017 Maravall, de Lope and Fuentes. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.