



A low-cost vision system for online reciprocal collision avoidance with UAVs

Julian Estevez^{a,*}, Endika Nuñez^b, Jose Manuel Lopez-Guede^c, Gorka Garate^a

^a Group of Computational Intelligence, Faculty of Engineering of Gipuzkoa, University of the Basque Country, UPV/EHU, San Sebastian, Spain

^b Group of Computational Intelligence, Faculty of Computer Science, University of the Basque Country, UPV/EHU, San Sebastian, Spain

^c Group of Computational Intelligence, Faculty of Engineering of Vitoria, University of the Basque Country, UPV/EHU, Vitoria, Spain

ARTICLE INFO

Communicated by Christian Circi

Keywords:

Quadrotor

Collision-avoidance

RCA

Autonomous navigation

ABSTRACT

In this article, we propose a reciprocal collision avoidance system for autonomous drones, based on computer vision and using relative positioning in an indoor environment. This dynamic environment represents a demanding challenge, but it is crucial for any future existence of multiple drones operating in urban areas. We use commercial AR Drone 2.0 robots, which represent that our proposal is suitable for low-cost equipment. In our case, we attempt to achieve the collision avoidance of two drones that fly one towards the other and react online autonomously to signals received by their computer vision systems with a decentralized control strategy. We test this in four different experiments with demanding conditions. For this purpose, we get the camera signal of the onboard drones and tune their behavior to react smoothly and precisely. We report encouraging positive results and provide the code we use in the experiments for replication.

1. Introduction

Since the last decade, scientists have made a big effort to research different aspects and applications of UAVs. Some examples can be found in precision agriculture [1], military applications, different objects transportation [2,3], inspection of infrastructures [4] and space surveillance [5]. However, due to the expected increase in the number of UAVs in future services, sky-sharing aircrafts and infrastructure still remain a big operational-safety concern, because the UAVs will be required to operate avoiding obstacles and interacting close to each other [6]. In this scenario, a certain type of robot autonomy and cognitive architecture for dealing with sudden incidents is mandatory [7]. Moreover, the research on the proper interaction and collaboration between UAVs opens the possibility to greater missions and collective intelligence [8].

In UAVs interaction, we find different research aspects, i.e., swarm formation, obstacle avoidance and scenario mapping. In this article, we describe a new proposal for reciprocal collision avoidance based on onboard computer vision using low-cost quadrotors. This approach pursues to provide a robotic system of a higher autonomy and intelligence [9,10]. In the *sense-and-avoid* problem, path planning of the avoiding maneuver is a crucial feature of robotics. Condition-free path-planning

has attracted efforts of researchers since the 1970s [11]. But in spite of the research done, different works in this field show that path planning, in specific situations such as the sudden detection of an obstacle or the mutual collision avoidance among drones, is still a hard and not totally resolved problem [12,13].

Academia has been working on reciprocal collision avoidance (RCA) algorithms in robotics just over the past few years [14,15]. The scenario involves multiple robots navigating in a shared environment, and each robot must take navigation decisions in a decentralized way through a multisensory continuous cycle. During flight, each robot observes the other robots and their trajectories; if necessary, without mutual communication nor coordination, they must compute movement decisions in order to continue along the planned trajectory avoiding collisions [16]. Because of their limited payload, achieving such dependable flying autonomy based on sensing and computation is especially difficult for small UAVs weighing no more than a few kilograms, or even less [17]. Collision avoidance methods are categorized into non-cooperative avoidance approaches and cooperative avoidance approaches. Non-cooperative approaches operate without prior information about obstacles through communication, whereas cooperative approaches utilize shared information through cooperative communication [18,19]. Thus, our RCA algorithm falls into the non-cooperative approach.

* Corresponding author.

E-mail address: julian.estevez@ehu.eus (J. Estevez).

The key feature of most RCA approaches is that the robots specifically share the responsibility of the maneuver to avoid pairwise collisions. Failure of this protocol inevitably carries very undesirable perturbations in the navigation of the robots [20,21]. It is highly convenient to achieve a low computational load for this algorithm [22], as it should check the safety of the robots in the environment several times per second.

Two of the most important aspects for a collision avoidance system are the sensing and detection capability and the collision avoidance strategy [23,24]. In research works of UAVs various sensors have been used, and many different technologies, such as ADS-B, FLARM, or MLAT for collision avoidance [25]. FLARM (flight alarm) is a proprietary electronic system used to selectively alert pilots to potential collisions between aircraft [26]; MLAT is the multilateration method responsible for communications, navigation, and surveillance/air traffic management among airships, and [27] uses it with an ultrawind band sensor.

Researchers in [28] use a Wi-Fi network with a range of 100 m to exchange information in a swarm of UAVs for cooperative exploration tasks. Meanwhile, in [29] authors created a GNSS RTK network that estimates the relative positions of multiple ground robots receivers with respect to a single-base receiver. Similarly, [30] present a split-merge strategy that leverages pigeons' obstacle avoidance behavior and employs a consensus approach to achieve efficient obstacle avoidance and formation reconstruction for multiple quadrotors. In [31] an ADS-B technique is used, which enables the UAV to keep track of information from other UAVs, and opts for a flatness-based approach for trajectories generation. Authors in [32] develop a reactive collision avoidance system for static obstacle detection, based on a Doppler radar sensor. In [33], a fusion of ultrasonic and infrared sensor is used as an input to a collision avoidance controller. This combination of sensors provides better sensor accuracy according to authors. Researchers in [34] employ a LiDAR sensor to implement the RCA maneuver of two UAVs in a corridor. Finally, other types of sensors are also used: [35] use small-sized radar sensor for low-cost micro UAVs, [36] use an optic-flow sensor for miniature drones as well and, finally, [37] choose an ultra-wideband sensor for their experimentation.

A promising research field are the vision based RCA systems that rely on cameras, as some authors have tested [38–41]. Cameras are now used in multiple applications for static obstacle avoidance with UAVs [42]. However, in following lines we will restrict to reciprocal collision avoidance systems in multi-UAVs. Visual sensing is of particular interest as it generally requires less power and is lighter than other solutions [41]. The study detailed in [38] introduces a collision avoidance technique centered around visual detection. The hardware setup encompassed an expensive Hummingbird quadrotor fitted with a prominent marker and dual built-in fish-eye cameras. The fusion of measurements from the two cameras was accomplished using a Gaussian-mixture probability hypothesis density filter, resulting in effective tracking. However, all this required a high cost equipment with a complex and computationally costly algorithmic load. Similarly, in [43] scientists also use Hummingbird quadrotors, and they develop mutual collision avoidance maneuvers in simple trajectories, under controlled lightning conditions and with the help of an external Vicon system.

Next, authors in [39] employ a visual detection system on a Align T-REX 700 N helicopter, which was equipped with a stabilized camera alongside an onboard DM8168 image processing computer. This system utilized a close-minus-open (CMO) filter to isolate the sky region and employed a hidden Markov model (HMM) filter to identify targets against the sky background. In this work again, researchers use expensive hardware and computationally expensive algorithms, and their proposal is designed for long distance collision avoidances, as we can see in their results that UAVs relative distance goes from 20 to 100 m. On the other hand, researchers in [44] use cameras on quadrotors for navigating in a drone swarm. However, in this work humans are teleoperating the multirotors with the help of the cameras visual feedback, and the purpose of the study is to assess how operators perform when

navigating a simulated swarm using two scenarios: one where a camera is placed above the swarm and another where cameras are positioned within the swarm.

In [41], an obstacle avoidance system is developed for situations with limited field of view (FOV) interactions. Incorporating the field of view (FOV) limit is crucial to guarantee that the target consistently resides within the detection area of the onboard sensor, thereby ensuring the fulfillment of the lock-on condition. In this work, the authors introduce a potential-based framework for navigation and collision avoidance. However, it is worth noting that this framework does not consider coordination as a primary objective, the system is designed for fixed-wing UAVs, and they validate it with simulations. In [45] the team of researchers designs a real time deep-learning computer vision algorithm based on YOLOv3-tiny and a low-cost, wide angle monocular camera with real-time computer vision algorithms to detect and track other UAVs. However, their solution is validated only for a 30-40 m separation between UAVs in an outdoors scenario. Moreover, monocular vision is characterized for a bad performance with darkness or strong light conditions [46]. This is a similar approach to the more recent article in [47], whose algorithm is based on YOLOv7, but it is adjusted for fixed wing UAVs and long distances too.

In all these vision based reports, however, due to the noise that the camera receives from the environment, researchers reported inaccuracy of the algorithms and lack of robustness of the vision [19,17,48,47,46]. Moreover, some of the studies rely only on simulated validation, and they all use expensive and often, not affordable, laboratory equipment together with complex filters [19,49]. Therefore, cited bibliography reveals that visual sensing for RCA systems in UAVs share some of following characteristics: expensive drone and cameras equipment, long distance applications, computationally expensive algorithms, fixed-wing UAVs, or human intervention in teleoperation.

Our proposal pursues to fill a not covered niche in this field of research, and we aim to develop a low-cost UAV platform and a vision-based RCA strategy for short distances. The scenario consists of two UAVs interacting in the air without exchanging information and acting based on their onboard sensors in a decentralized manner. In spite of their reduced cost, our solution has been successfully tested on AR Drone 2.0 commercial drones; all the required data for the RCA algorithm is provided by the low-cost sensors of the drones. Although the capability of the drones for onboard processing is small, this work opens the way to research groups that cannot afford other more expensive equipment in the market, such as Vicon or Hummingbird systems [50]. Besides, there is interesting scientific literature using the AR Drone 2.0 for collision avoidance research purposes [51–53,9].

Using a low-cost equipment with low resolution camera and a simple vision-based algorithm, we implement a robust collision-avoidance algorithm for UAVs, which can be easily extrapolated to other type of aerial robots; moreover, we provide the complete code of the solution. The collision avoidance maneuver does not require that UAVs exchange information via communication; instead, they rely solely on onboard sensors and cameras.

Thus, our contributions are the development of a real time RCA algorithm based on low-resolution cameras, which can be integrated in low-cost drones too. This system dispenses with exchange of information between robots and we provide the code of the solution. In our experiments, we try to imitate realistic conditions of UAVs sharing the same scenario in an urban area (uncontrolled lighting conditions and robots flying from different directions), and thus, our contributions are focused on real-world applications, despite the initial work in laboratory. At this step, the algorithm is restricted to a single drone avoiding the collision with another single UAV.

The article is organized as follows: in Section 2, the RCA strategy based on computer vision is detailed. In Section 3, we describe the dynamic modeling equations of the quadrotor. Next, in Section 4, four experiments are presented and results are exposed in Section 5 and



Fig. 1. UAV with the red card mounted. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

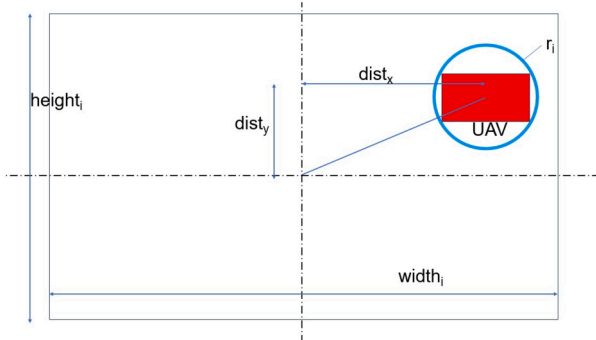


Fig. 2. Schematic image.

discussed in Section 6. We finish the article with some conclusions in Section 7.

2. Collision-avoidance system

Our solution is based on using the cameras of the quadrotors to detect the proximity and location of near quadrotors. The selected cameras are the frontal ones integrated in the AR Drone 2.0 UAVs. The video signal about encountered quadrotors is processed in an OpenCV and Python based algorithm, located in the laptop where the UAV is connected, which contains the RCA algorithm that sends the commands to the quadrotor to avoid the crash in the air on real time.

We assume the next flying and hardware conditions:

- Quadrotor model is the same and equipped camera has the exact same characteristics.
- Flying altitude difference of the UAVs must be kept in a range.
- Quadrotors have real time information about their own state variables.
- They carry the same visual objects / cards, which will be described lately and can be seen in Fig. 1.
- They do not share any information between drones.

2.1. RCA algorithm details

Each UAV is equipped with a colored red card, specially designed to be mounted on the quadrotor, as can be seen on Fig. 1. This card permits the software algorithm to detect the presence of an approaching drone and measure its proximity. The details of this card are explained in Section 3.

Our RCA proposal is based on the usage of color and percentage of colored areas on an image, divided into two hemispheres, left and right.

Our solution is continuously searching for a range of red color on real time, and embeds it on a virtual circumference (as can be seen in Fig. 2). When the radius of the circumference overcomes a preset threshold, the drone starts considering the proximity to the obstacle, by calculating the percentage of colored area appearing on the screen. When the percentage is higher than another established limit, the UAV

assumes that the obstacle is near enough, and the maneuver for the collision avoidance strategy takes place. The decision to turn left or right is made upon the percentage of area appearing on each hemisphere of the camera, providing an autonomous decision making capacity to the robot. If main area is on left one, the UAV will turn to the opposite direction till it does not detect the red area on the image. The decision to turn left or right is executed by acting on the φ degree of freedom of the UAV.

In this article, we propose a reciprocal collision avoidance system for autonomous drones, based on computer vision and using relative positioning in an indoor environment. Thus, our proposed solution proves to be easily scalable and executed as many times as required.

The changing lighting conditions require a precise color range tuning, and cited circumference radius threshold permits filtering noisy image signals and scenario parts, thus enhancing the robustness of the system on real time.

The pseudocode of the RCA algorithm is shown on Algorithm 1 and it is uploaded on an online repository.¹ It can be seen that for the implementation of the RCA algorithm, we use an OpenCV library based algorithm for color related calculus. In this way, the algorithm pursues the crash avoidance by a combination of color and distance detection on the image.

Algorithm 1 RCA algorithm.

Input:

planned_position: [x, y, z] coordinates of the planned position of the UAV.

n: number of UAVs.

$[r, g, b]_{min}^T$: minimum threshold to consider a pixel as target area.

$[r, g, b]_{max}^T$: maximum threshold to consider a pixel as target area.

r_{min} : minimum radius to consider a portion of the image as target area.

$target_area_{min} \in (0, 100)$: minimum percentage of the image filtered to consider that the UAV is close to another UAV.

$target_area_{max} \in (0, 100)$: maximum percentage of the image filtered to consider that the UAV is close to another UAV.

$velocity_rate \in (0, 1)$ rate to reduce the nominal velocity of the UAV.

Output:

$[velocity_x, velocity_y]$: components of the velocity of the UAV adjusted to collision avoidance.

```

while (1) do
  for each i in [1..n] do
    fly UAVi towards planned_positioni
    imagei ← take a color image of size (heighti, widthi)
    filtered_imagei ← pixels of imagei such that  $[r, g, b]_{min}^T \leq [r, g, b] \leq [r, g, b]_{max}^T$ 
     $r_i \leftarrow$  minimum radius whose circumference contains filtered_imagei
    if  $r_i \geq r_{min}$  then
       $circle\_area_i \leftarrow \pi r_i^2$ 
       $percentage_i \leftarrow \frac{circle\_area_i \cdot 100}{(height_i \cdot width_i)}$ 
      if  $percentage_i \geq target\_area_{min}$  then
         $A \leftarrow (percentage_i - target\_area_{max})^2$ 
         $velocity_{i,x} \leftarrow velocity_{i,x} \cdot velocity\_rate \cdot A$ 
         $hypotenuse_i \leftarrow \sqrt{(height_i^2 + width_i^2)}$ 
         $[dist_x, dist_y] \leftarrow$  distance from (0,0) to the center of the circle containing filtered_imagei (Fig. 2)
         $hypotenuse\_dist_i \leftarrow \sqrt{dist_x^2 + dist_y^2}$ 
         $B \leftarrow \frac{hypotenuse\_dist_i}{(0.5 \cdot hypotenuse_i)}$ 
         $velocity_{i,y} \leftarrow velocity_{i,y} \cdot velocity\_rate \cdot (1 - B)$ 
      end if
    end if
  end for
end while

```

¹ https://github.com/Julestevez/Reciprocal_collision_avoidance.

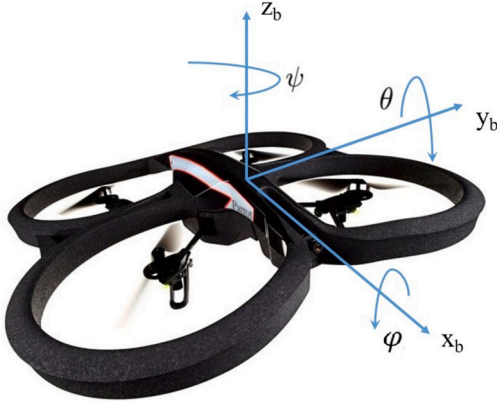


Fig. 3. AR Drone 2.0 motions.

Next, Fig. 2 shows the measures of main variables in the algorithm.

In our experiments, the collision avoidance maneuver will be horizontal, either to left or right, for simplicity in the research purposes. However, parameters of the algorithm can be easily adapted so that the maneuver of the drones becomes vertical too and it would just require the modification of a few parameters in the RCA solution. Moreover, r_{min} , $target_area_{min}$ and $target_area_{max}$ and UAV forward speed will be empirically selected for each camera, quadrotor and laptop system. Our RCA proposal will always guarantee that collision is avoided as far as those parameters are properly calculated. These three parameters are also relevant to modify the maneuver distance, or distance between drones for reaction. Finally, time complexity of the algorithm is $O(npxqxq)$, where n refers to the number of UAVS, while p and q are the height and width pixels respectively.

3. Description of the experimental platform

3.1. Dynamic model of a UAV

There are four basic motions of this UAV: roll, pitch, yaw (φ, θ, ψ respectively) represent the angles around x_b, y_b and z_b axis framed on the body of the UAV, w.r.t. absolute coordinates ($X; Y; Z$), as shown in Fig. 3. The fourth motion is lift, which is the force in the z_b direction of the drone and m is the mass.

Next, a short recall of the dynamic formulas of the UAV is resumed.

The following equations govern the motion of a quadrotor, and relate the linear accelerations to the input commands:

$$\begin{aligned} \ddot{x} &= (\sin \psi \sin \varphi + \cos \psi \sin \theta \cos \varphi) \frac{U_1}{m}, \\ \ddot{y} &= (-\cos \psi \sin \varphi + \sin \psi \sin \theta \cos \varphi) \frac{U_1}{m}, \\ \ddot{z} &= -g + (\cos \theta \cos \varphi) \frac{U_1}{m}, \end{aligned} \quad (1)$$

where U_1 represents the vertical lift in z_b direction of the drone (see Fig. 3). The second system of equations (eq. (2)) specifies how the basic movements are related to the propellers' squared speeds.

$$\begin{aligned} U_1 &= b (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2), \\ U_2 &= l \cdot b (-\Omega_2^2 + \Omega_4^2), \\ U_3 &= l \cdot b (-\Omega_1^2 - \Omega_3^2), \\ U_4 &= d (-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2), \end{aligned} \quad (2)$$

where U_2, U_3 and U_4 are roll, pitch and yaw torques, respectively, and Ω_i^2 is the squared speed of the i -th propeller. The parameter b is the propeller thrust coefficient, l is the quadrotor arm length and parameter d is the drag coefficient.

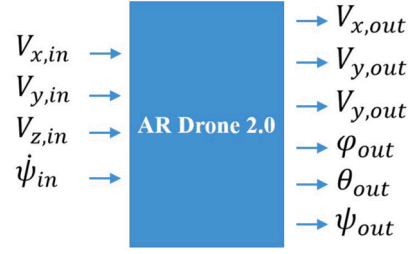


Fig. 4. Input and output variables of AR Drone 2.0.

Finally, in eq. (3) the equations that relate angular accelerations and the three torques are shown, where I_{xx}, I_{yy} and I_{zz} are the three inertia moments:

$$\begin{aligned} \ddot{\varphi} &= \frac{U_2}{I_{xx}}, \\ \ddot{\theta} &= \frac{U_3}{I_{yy}}, \\ \ddot{\psi} &= \frac{U_4}{I_{zz}}. \end{aligned} \quad (3)$$

3.2. Analysis of inputs and outputs and system identification

The AR Drone 2.0 is a UAV designed to be remotely controlled with a smartphone or a tablet by a human operator in a Wi-Fi network. The autopilot (also called firmware) is installed on its main board, and helps stabilizing the robot under some low disturbances and automatically take-off, land and hover [54].

The developed Software Development Kit (SDK) allows the quadrotor to transmit and receive the information about the robots orientation and linear velocities [9]. The system is executed by four inputs, which are the linear velocities on the three axes and yaw angular speed references $\{V_{x,in}, V_{y,in}, V_{z,in}, \dot{\psi}_{in}\}$. In this SDK, the input control signals are normalized and range from -1 to $+1$. At the end of the loop, the UAV transmits the measurement of the three angles around the body-frame axis, its height, and longitudinal and transversal speeds, denoted by $\{V_{x,out}, V_{y,out}, V_{z,out}, \theta_{out}, \varphi_{out}, \psi_{out}\}$ respectively, as depicted in Fig. 4. This information, as well as how to get access to them can be found in [55].

3.3. Equipment and communications

A Python library called PS Drone [56] is in charge of communicating and sending movement commands to the AR Drone 2.0 using a host computer and the command interface (tablet, smartphone, laptop) over a 50 m range Wi-Fi network. The quadcopter is repeatedly transmitting its height and orientation to the host computer in real time. To do this, it is equipped with an IMU sensor responsible for measuring the three main angle measurements and with an ultrasound altimeter with an operating frequency of 15 Hz.

This drone is equipped with two cameras. The first one, at the bottom, has a resolution of 320×240 pixels at 30 frames per second (fps); the second one, at the front, has a resolution of 640×360 pixels at 60 fps with a 92 degree wide-angle lens [9]. The onboard computer is composed of a 1 GHz ARM Cortex A8 processor running Linux, a 1 GB RAM and a Wi-Fi module.

The PS Drone Python library allows the control of the movement of the quadcopter. In order to initiate this movement from the host computer, a command between 0 and 1 is sent to the quadcopter. This command represents a percentage of the maximum drone speed.

Each quadrotor is equipped with a colored red card, specially designed to be mounted on the quadrotor, as can be seen on Fig. 1. Main dimensions and shape of the card are shown in Fig. 5. Among the set of experiments that the authors have developed, we also use a blue card

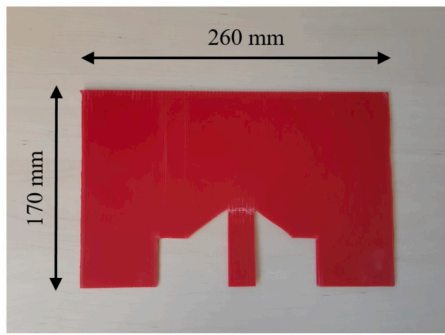


Fig. 5. Dimensions of the red card.



Fig. 6. Initial conditions of the experiment.

with the same dimensions as the red one. However, in the experiments of the current article, the blue card is not used. Red card is not regarded as the only color valid for our experiments. Our solution aims to be universal for low-cost drones and laboratory conditions, and any colored card can be used, by simply modifying the RGB values in the code.

We implement the OpenCV based algorithm for the UAV to search and hover in front of the blue card, and to avoid the red card. This procedure can be observed in following video.²

Finally, the camera signal of the drone is received and visualized on the laptops, where the computer vision based RCA algorithm is hosted.

Two UAVs take part in the experiments, each of them controlled by a single laptop. Details of the procedure are presented in following section.

4. Experiment

In this work, we carry out four experiments to check our RCA proposal. Four tests are based on the aerial facing of two drones in the air to analyze how they deal with the mutual avoidance maneuver, but we change parameters and boundary conditions among the four experiments.

In *Experiment 1*, we locate two AR Drone 2.0 robots landed 15 m away from each other. At this starting point, we program the UAVs to take off and fly one against the other to force the collision. Both quadrotors fly approximately at the same height and are controlled with the same piece of code, which the authors have uploaded in an online open repository.³

The experiment aims to test the developed reciprocal collision algorithm. Since the robots are set perfectly aligned, opposite one to each other and they fly straight ahead, authors initially cannot guess which will be the direction of the avoiding maneuver, whether left or right. The cards described in Fig. 5 are mounted on both quadrotors so that the front color is red. In the absence of external position measurement systems, such as Optitrack, the programmed path is a forward flight during 17 seconds, and finish with a landing.

The dynamic and sensor parameters during this experiment are shown next: $velocity_rate = 0.2$; $r_{min} = 10px$; Size of frame = 640×360 px; $target_area_{max} = 10$; $[r, g, b]_{max}^T = [255, 0, 0]$ and $[r, g, b]_{min}^T = [175, 0, 0]$.

A representation of the initial conditions of the experiment is shown in Fig. 6. Thus, the robots are expected to meet at the middle point of the path.

In *Experiment 2*, we alter the symmetry of the robot parameters and set the $velocity_rate = 0.2$ for first AR Drone 2.0 and set $velocity_rate = \{0.15, 0.10, 0.05\}$ for the second. In these variable conditions, we run the same test as in *Experiment 1*, three times for each $velocity_rate$. This

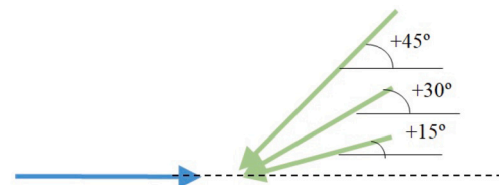


Fig. 7. Paths of Experiment 3.

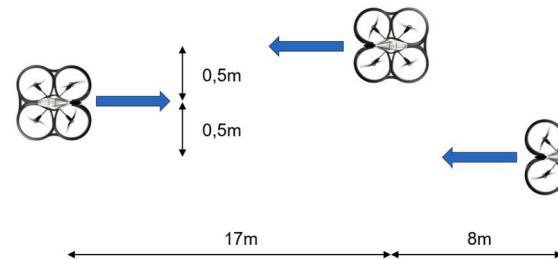


Fig. 8. Initial positions of Experiment 4.

demonstration aims to check whether UAVs with different capacities can execute correctly the RCA algorithm.

In *Experiment 3* multirotors face each other from different angles, so that the straight paths of both robots cross in the air. Considering the zero reference the line that joints two AR drones perfectly faced, we perform the experiment so that the straight paths of the drones create the following angles between them: $\{\pm 15^\circ, \pm 30^\circ, \pm 45^\circ\}$, as shown in Fig. 7. We carry out this analysis once per each angle orientation, and we aim to simulate the random conditions that might happen in a hypothetical situation of multiple drones interacting in the same scenario, where robots can come from any direction.

Finally, in *Experiment 4* we position 3 drones in the scenario. Two UAVs fly in a direction, and the third flies against them. The initial locations of the robots are shown in Fig. 8. In this test the authors try to prove the scalability of the proposed algorithm. Three robots are equal and the colored cards mounted on the AR Drones are shown only forward.

In the four experiments, identical laptops are used to send and receive signals from the UAVs: HP Pavillion with a quad core Intel i7-2410 @2 GHz, with 16 GB RAM memory and running Linux Ubuntu 20.04. The frequency of the IMU of the drones is 400 Hz; however, the authors sampled the data at 20 Hz for reducing computational load (every 50 ms). There were no controlled lighting conditions.

² <https://www.youtube.com/watch?v=rVl0avxx7TU>.

³ https://github.com/Julestevez/Reciprocal_collision_avoidance.



Fig. 9. Resulting path of two UAVs in Experiment 1.

Table 1
Lateral distance [cm] that each UAV executes in the RCA maneuver.

	Velocity rate 0.15			Velocity rate 0.10			Velocity rate 0.05		
UAV ₁	90	94	96	102	105	103	114	118	115
UAV ₂	60	57	64	48	51	52	38	39	41

5. Results

5.1. Experiment 1

The UAVs successfully avoided the collision between each other and ended their flight after 17 seconds since the take-off. The complete video of the experiment can be observed in next URL.⁴ As we mention in Subsection 3.3, despite a blue card appears on the back side of the UAV, it does not participate in the current experiment, and both drones equip red cards at the front. The video⁵ shows the results of the experiment from the point of view of the drone camera and the calculus of the collision-avoiding parameters. Thus, in the absence of a external position measurement system, the stylized path made by two UAVs is described by Fig. 9.

Next, in Fig. 10, the altitude evolution along the experiment for two UAVs is shown, which is stabilized around 80 cm. In Fig. 11, we can see the evolution of the roll angle of the UAVs, which is precisely the main responsible angle for the collision avoidance maneuver. In the experiment, φ angle is positive and then negative in both graphs, meaning that both robots first turn right to avoid the obstacle and then come back to original path by turning left (see Fig. 3).

In Fig. 12 we can see the evolution of the pitch angle, θ , which represents the forward movement of the UAV. As we can see, the pitch angle has a hesitant performance, which we attribute to the signal noise, false negatives of red color detections in the scenario and uncontrolled lighting conditions that were in the experiment.

The authors carried out this experiment for 10 times more, and measured the existing frontal distance between drones when starting the collision avoidance maneuver and the lateral distance that covered. In absence of Vicon system, we calculate these values through odometry and numerical integration of $V_{x,out}$ and $V_{y,out}$ (see Fig. 4). The results are shown in Fig. 13.

5.2. Experiment 2

In this experiment, we performed nine tests in total with a different *velocity_rate* parameter for one of the two multicopters.

All the analysis were successful and UAVs avoided the collision between themselves smoothly. One of those trials can be observed in video.⁶

In Table 1, we can appreciate the lateral distances that each of the drones flies in the avoiding maneuver to left or right, depending on the different *velocity_rate* that UAV₂ has.

Table 2
Lateral distance made by UAVs in Experiment 3.

	Lateral distance in RCA [cm]					
	15°	30°	45°	-15°	-30°	-45°
UAV ₁	60	55	48	64	50	48
UAV ₂	62	50	42	58	52	45

5.3. Experiment 3

In this experiment, two UAVs cross in the air at different angles $\{\pm 15^\circ, \pm 30^\circ, \pm 45^\circ\}$, and same dynamic and sensors parameters. Again, our proposal of RCA algorithm proved robust enough to get all the flights to be successful and drones did not collide.

As a result of this analysis, two aerial robots were able to avoid themselves in all the tested situations. The schematic reaction of both UAVs is represented in Fig. 14.

Moreover, we include Table 2 with the results of the lateral distance flown by each drone for avoiding the collision. Positive angles lead to maneuvers to the right, while negative angles lead to maneuvers to the left.

5.4. Experiment 4

In the final test, three UAVs try to avoid the collision between them. Two multicopters fly in one direction and the other one flies against them. The complete experiment lasts 24 seconds, when the robots finally land. The AR Drone 2.0 on the left of the figure was able to avoid the other 2 robots, and the schematic path three of them followed is shown in Fig. 15

6. Discussion

In this article, we propose a reciprocal collision avoidance (RCA) algorithm applicable using simple technical equipment, sensors and sources of data. In order to achieve this, we use two AR Drones 2.0, equipped with low-cost sensors and cameras, and two colored red cards in order to detect the presence of an incoming airship and to be able to autonomously avoid it on real time. The strategy is decentralized and it is robust enough for different lighting conditions that we test during experiments. We intend for the system to be easily applicable to other robot models, and thus, we have uploaded the programmed code in Python in an open online repository.

In the Results section of the text, Fig. 10 shows that the altitude of both UAVs is quite stable and after losing about 10 cm while performing the avoiding maneuver, they recover immediately the preset height objective.

Next, Fig. 11 and 12 show evolution of pitch and roll angles of drones along the experiment. First image shows a smooth behavior of both UAVs to fly laterally and avoid the collision, and a later compensation of roll angle, φ , to return to planned path. Second image show a bit hesitant behavior of the pitch angle, ϑ , but the video of the experiment shows that despite this, drones fly quite smoothly and we consider that low-cost sensors are the responsible of this issue. We recall that lighting conditions are uncontrolled, and thus, we choose broad limit values for the detection of the color red, denoted with $[r, g, b]$. The authors consider that they sacrificed the perfect performance of flight of the drones in order for the algorithm to work for different light conditions that occurred during the experiments, and undoubtedly the specific values of the algorithm parameters play a role in that performance. Finally, Fig. 13 proves a good versatility of the RCA algorithm in different reaction distances.

Experiment 2 and Experiment 3 aim to prove further capacities of RCA solution and its robustness under demanding conditions, trying to replicate nearer real-world conditions similar to a hypothetical scenario of

⁴ <https://www.youtube.com/watch?v=FCd3-Q1sVhQ>.

⁵ <https://youtu.be/kIHcLJpx5rQ>.

⁶ <https://youtu.be/FCd3-Q1sVhQ>.

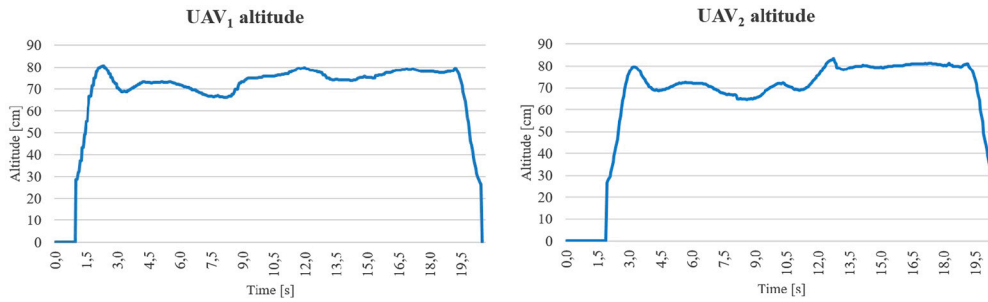


Fig. 10. Altitude evolution for two UAVs in Experiment 1.

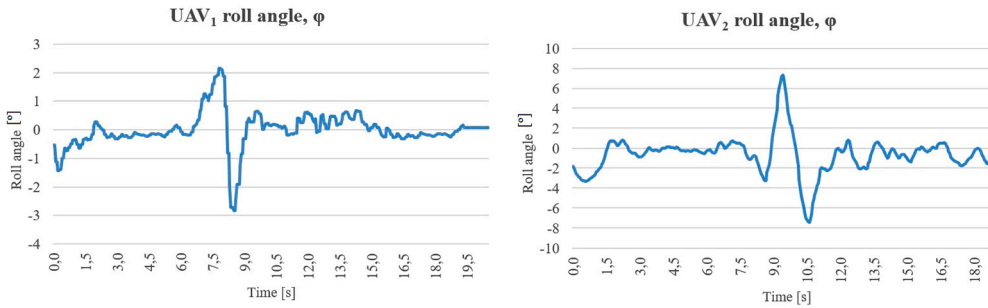


Fig. 11. Roll angle evolution for two UAVs in Experiment 1.

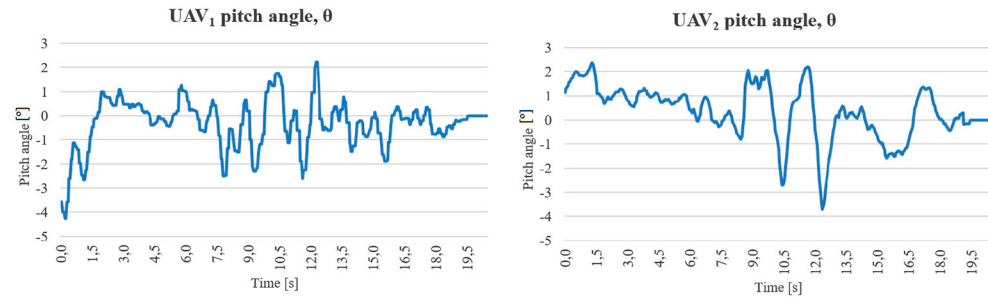


Fig. 12. Pitch angle evolution for two UAVs in Experiment 1.

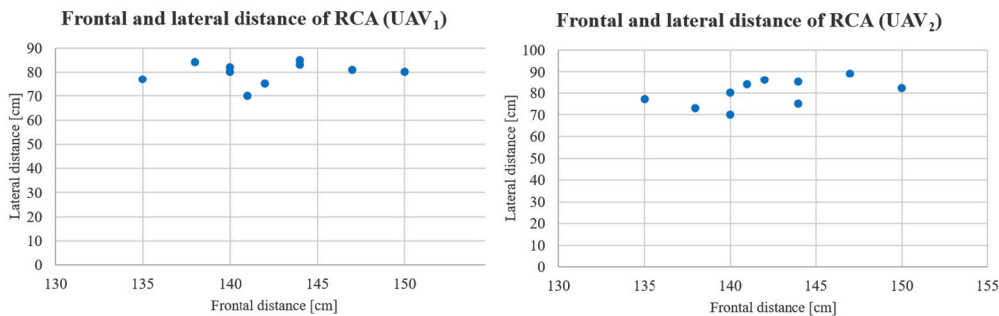


Fig. 13. Lateral and frontal distance for avoiding maneuver in Experiment 1.

drones sharing a scenario in a city. In *Experiment 2*, Table 1 demonstrates that UAVs with different velocity capacities can still avoid the collision between them. In order to show that capacity, we can observe that for every different velocity rates, the values in UAV_1 column are always higher than the values under UAV_2 column, which means that UAV_1 assumes the responsibility in the RCA maneuver to fly a longer distance than the other robot due to its higher velocity rate (0.2).

In *Experiment 3* we test RCA algorithm for UAVs in straight paths that are not aligned. This experiment represents a big challenge for the low-cost sensors equipment, and checks the robustness of the algorithms when identifying colors and thus, it requires a fine tuning

of their thresholds on the code. Colored cards with different orientations tend to make harder to distinguish the color due to shadows and lower frontal surface. However, as can be analyzed on Table 2, resulting data is coherent and multirotors fly the required distance to avoid themselves. Maneuver distances are shorter than in *Experiment 1*, which makes sense due to the lower projected surface of the red card.

Finally, *Experiment 4* shows that the proposed RCA algorithm is able to cope with more than two multirotors. For this purpose we located three UAVs in different coordinates, and we can appreciate that robot in the top left of Fig. 15 is able to avoid the collision with two other apparatus, one after the other. The purpose of this work is the descrip-

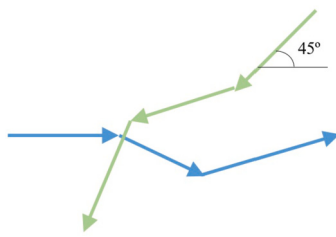


Fig. 14. Collision avoiding reaction of UAVs in *Experiment 3*.

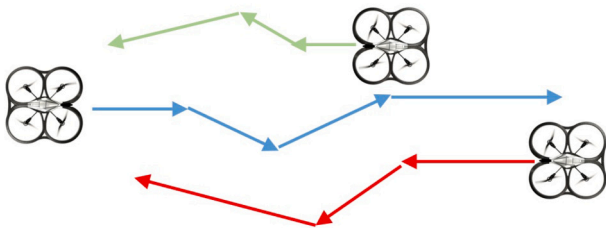


Fig. 15. Schematic paths of three UAVs in *Experiment 4*.

tion of a low-cost vision system for online reciprocal collision avoidance with UAVs. Despite the good results shown in *Experiment 4*, we consider that more tests are necessary as future work in order to find the limitations of the number of simultaneous UAVs flying in the same scenario that this RCA algorithm is able to manage. The low cost of this solution may not be able to handle the presence of a high number of drones in the camera and react to appropriately to avoid them.

As described in the article, we use a laptop per UAV to control the RCA algorithm. This design decision again tries to emphasize the low complexity and low budget of our proposed solution. However, we are aware that in order to extend this solution outdoors and increase the number of multirotors, we will have to enhance the UAV technology and onboard processing power, so that UAVs are able to compute the calculus remotely. In order to keep the focus of current work, we set the implementation of onboard image processing with UAVs as future work.

Although four experiments described in the article performed well, during the testing and fine tuning of the algorithm parameters, the authors found some very specific drawback situations in which our low-cost vision system for multi-UAVs did not perform well. Those situations involved very high lightning conditions and excessive approach flying speed between UAVs. We attribute these inconveniences to a narrow range of $[r, g, b]$ values, which might not be easily detectable in high lightning conditions. Next, we consider that the low-cost cameras harm the required image processing speed, and thus, UAVs can not fly at their maximum speed.

The purpose of this work is the development of a low-cost vision system for online reciprocal collision avoidance with UAVs; it is because of this low-budget objective that we use a laptop per UAV to control the RCA algorithm. In spite of this limitation in budget, the results obtained in all four experiments performed confirm that the system does ensure the collision avoidance in a robust and consistent way while maintaining stable flight conditions. The fact that experiments were repeated tenths of times using different velocities and angles of incidence proves the robustness and consistency of the RCA system presented.

In the context of multi-UAV systems, the preference of collision avoidance systems leans towards cooperative approaches due to their benefits in safety and design flexibility [19]. However, we set ourselves demanding and unfavorable conditions to have a harder benchmark against we could test our RCA system.

In addition to this, we believe that our proposal is supported by three main aspects, apart from the technical performance: it is cheap to build and has a low computational cost. Therefore, it is easily affordable by most of research groups. Moreover, it does not require special

equipment, and our laptop requirements are not demanding and there is a big number of them in most laboratories. Finally, it is a sustainable technical solution and permits the reutilization of old technical equipment.

7. Conclusions

Summing up, we consider that in the light of the results, the benefits of our RCA system are a stable and a smooth dynamic behavior of the drones, and to count with a simple algorithm integrated on low-cost equipment that can be easily embedded in other robotic systems. Obviously, this algorithm is also valid for high quality cameras. Finally, developed RCA algorithm also proved to be an obstacle collision avoidance algorithm: in case the system in one of the drones fails, the other drone is perfectly available to avoid the collision by assuming the whole responsibility of making a longer lateral maneuver till the drone with failure disappears from its camera.

The experiment and results show that, despite the lack of complexity in the sense and avoid system, after a careful tuning of the required parameters in the algorithm, the system did not have any UAV crash in none of the experiments. Besides, there is no data exchange between robots, enhancing the simplicity in communication hardware. The four presented experiments show that the system is able to work with more than two UAVs, and the authors are currently working on new experiments that involve a group of drones as future work.

Therefore, we consider that the proposed RCA algorithm could be an easy first approach for research groups aiming to explore the autonomous navigation of robots and interaction among them, in order to prospect future scenarios with proliferation of robot services in our surroundings. As our solution is designed towards the filling of the niche evidenced in the bibliography of the research field, the limitations of our proposal consist on the inability for long distances object detection, neither high speed UAV conditions, nor fixed-wing drones.

Future work of the authors will go towards enabling a more robust navigation control that could lead to test autonomous navigation outdoors while coping with different wind disturbances and more severe environment conditions. The required technical improvements to cope with this new scenario do not only involve the RCA algorithm, but also the hardware equipment and communications. First of all, authors should explore the navigation controllability limits of the current UAVs to cope with wind disturbances and compensate them, and include vertical maneuvers for collision avoidance. In case it is necessary, authors will need to augment the sensing capacities and add extra sensors (LiDAR or radar) than enhance the computer vision algorithms of the system. We should also optimize the algorithm for energy efficiency, especially in challenging conditions where the UAV may need to consume more power for stability and control. This involves refining the balance between safety and energy consumption.

Also, more robust communications are required that strengthen communication protocols between UAVs if the environment includes areas with signal interference or communication blackouts. Implementation of reliable communication strategies to ensure timely sharing of collision avoidance information is necessary.

Moreover, authors will also research for the enhancement of algorithm for multiple simultaneous collision avoidance maneuvers, this is, when more than one drone are coming towards a single UAV. In order to achieve that, authors will work different research lines, such as the replacement of the UAV models by some other that could easily handle image processing and autonomous collision avoidance techniques onboard.

Code availability

Code is hosted in online repository: github.com/Julestevez/Reciprocal_collision_avoidance.

CRedit authorship contribution statement

Julian Estevez: Writing – original draft, Validation, Supervision, Methodology, Conceptualization. **Endika Nuñez:** Writing – review & editing, Software, Project administration, Methodology, Investigation. **Jose Manuel Lopez-Guede:** Writing – review & editing, Supervision, Funding acquisition. **Gorka Garate:** Writing – review & editing, Software, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

The work in this paper has been partially supported by FEDER funds for the MICIN project PID2020-116346GB-I00, research funds from the Basque Government as the Grupo de Inteligencia Computacional, Universidad del País Vasco, UPV/EHU with code IT1689-22. Additionally, the authors participate in Elkartek projects KK-2022/00051 and KK-2021/00070. Authors have also received support by Fundacion Vitoria-Gasteiz Araba Mobility Lab.

References

- U.M. Rao Mogili, B.B.V.L. Deepak, Review on application of drone systems in precision agriculture, *Proc. Comput. Sci.* 133 (2018) 502–509.
- Daniel K.D. Villa, Alexandre S. Brandao, Mário Sarcinelli-Filho, A survey on load transportation using multirotor UAVs, *J. Intell. Robot. Syst.* 98 (2) (2020) 267–296.
- Julian Estevez, Manuel Graña, Jose Manuel Lopez-Guede, Online fuzzy modulated adaptive PD control for cooperative aerial transportation of deformable linear objects, *Integr. Comput.-Aided Eng.* 24 (1) (2017) 41–55.
- Saeed Rafee Nekoo, José Ángel Acosta, Guillermo Heredia, Anibal Ollero, A benchmark mechatronics platform to assess the inspection around pipes with variable pitch quadrotor for industrial sites, *Mechatronics* 79 (2021) 102641.
- Payam Shafei Gohari, Hossein Mohammadi, Sajjad Taghvaei, Using chaotic maps for 3D boundary surveillance by quadrotor robot, *Appl. Soft Comput.* 76 (2019) 68–77.
- Chris Taylor, Cameron Nowzari, The impact of catastrophic collisions and collision avoidance on a swarming behavior, *Robot. Auton. Syst.* 140 (2021) 103754.
- Abdulmuttalib T. Rashid, Abduladhem A. Ali, Mattia Frasca, Luigi Fortuna, Multi-robot collision-free navigation based on reciprocal orientation, *Robot. Auton. Syst.* 60 (10) (2012) 1221–1230.
- Farhan Mohammed, Ahmed Idries, Nader Mohamed, Jameela Al-Jaroodi, Imad Jawhar, UAVs for smart cities: opportunities and challenges, in: 2014 International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, 2014, pp. 267–273.
- Thi Thoa Mac, Cosmin Copot, Robin De Keyser, Clara M. Ionescu, The development of an autonomous navigation system with optimal control of an UAV in partly unknown indoor environment, *Mechatronics* 49 (2018) 187–196.
- Amir Behjat, Steve Paul, Souma Chowdhury, Learning reciprocal actions for cooperative collision avoidance in quadrotor unmanned aerial vehicles, *Robot. Auton. Syst.* 121 (2019) 103270.
- N. Sariff, Norlida Bunyamin, An overview of autonomous mobile robot path planning algorithms, in: 2006 4th Student Conference on Research and Development, IEEE, 2006, pp. 183–188.
- Jose A. Cobano, Roberto Conde, David Alejo, Anibal Ollero, Path planning based on genetic algorithms and the Monte-Carlo method to avoid aerial vehicle collisions under uncertainties, in: 2011 IEEE International Conference on Robotics and Automation, IEEE, 2011, pp. 4429–4434.
- Mauro Pontani, Bruce A. Conway, Particle swarm optimization applied to space trajectories, *J. Guid. Control Dyn.* 33 (5) (2010) 1429–1441.
- Haotian Niu, Cunbao Ma, Pei Han, Directional optimal reciprocal collision avoidance, *Robot. Auton. Syst.* 136 (2021) 103705.
- Julian Estevez, Gorka Garate, Jose Manuel Lopez-Guede, Mikel Larrea, Review of aerial transportation of suspended-cable payloads with quadrotors, *Drones* 8 (2) (2024) 35.
- Yaohua Guo, Gang Chen, Tao Zhao, Learning-based collision-free coordination for a team of uncertain quadrotor UAVs, *Aerosp. Sci. Technol.* 119 (2021) 107127.
- Takumi Wakabayashi, Yukimasa Suzuki, Satoshi Suzuki, Dynamic obstacle avoidance for multi-rotor UAV using chance-constraints based on obstacle velocity, *Robot. Auton. Syst.* 160 (2023) 104320.
- Jawad N. Yasin, Sherif A.S. Mohamed, Mohammad-Hashem Haghbayan, Jukka Heikkonen, Hannu Tenhunen, Juha Plosila, Unmanned aerial vehicles UAVs: collision avoidance systems and approaches, *IEEE Access* 8 (2020) 105139–105155.
- Sunan Huang, Rodney Swee, Huat Teo, Kok Kiong Tan, Collision avoidance of multi unmanned aerial vehicles: a review, *Annu. Rev. Control* 48 (2019) 147–164.
- Jur Van den Berg, Ming Lin, Dinesh Manocha, Reciprocal velocity obstacles for real-time multi-agent navigation, in: 2008 IEEE International Conference on Robotics and Automation, IEEE, 2008, pp. 1928–1935.
- Panagiotis Rousseas, George C. Karras, Charalampos P. Bechlioulis, Kostas J. Kyriakopoulos, Indoor visual exploration with multi-rotor aerial robotic vehicles, *Sensors* 22 (14) (2022) 5194.
- Hamed Habibi, Ali Safaei, Holger Voos, Mohamed Darouach, Jose Luis Sanchez-Lopez, Safe navigation of a quadrotor UAV with uncertain dynamics and guaranteed collision avoidance using barrier Lyapunov function, *Aerosp. Sci. Technol.* 132 (2023) 108064.
- Hung Pham, Scott A. Smolka, Scott D. Stoller, Dung Phan, Junxing Yang, A survey on unmanned aerial vehicle collision avoidance systems, *arXiv preprint, arXiv:1508.07723*, 2015.
- Jung-Woo Park, Hyon-Dong Oh, Min-Jea Tahk, UAV collision avoidance based on geometric approach, in: 2008 SICE Annual Conference, IEEE, 2008, pp. 2122–2126.
- Juan A. Besada, David Carramiñana, Luca Bergesio, Ivan Campaña, Ana M. Bernardos, Modelling and simulation of collaborative surveillance for unmanned traffic management, *Sensors* 22 (4) (2022) 1498.
- David Carramiñana, Iván Campaña, Luca Bergesio, Ana M. Bernardos, Juan A. Besada, Sensors and communication simulation for unmanned traffic management, *Sensors* 21 (3) (2021) 927.
- Jungwon Kang, Kunwoo Park, Zahra Arjmandi, Gunho Sohn, Mozhdeh Shabbazi, Patrick Ménard, Ultra-wideband aided UAV positioning using incremental smoothing with ranges and multilateration, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2020, pp. 4529–4536.
- Alberto Viseras, Thomas Wiedemann, Christoph Manss, Valentina Karolj, Dmitriy Shutin, Juan Marchal, Beehive-inspired information gathering with a swarm of autonomous drones, *Sensors* 19 (19) (2019) 4349.
- Euiho Kim, Sae-kyeol Kim, Global navigation satellite system real-time kinematic positioning framework for precise operation of a swarm of moving vehicles, *Sensors* 22 (20) (2022) 7939.
- Jinjin Guo, Juntong Qi, Mingming Wang, Chong Wu, Yuan Ping, Shi Li, Jie Jin, Distributed cooperative obstacle avoidance and formation reconfiguration for multiple quadrotors: theory and experiment, *Aerosp. Sci. Technol.* 136 (2023) 108218.
- Yu Fu, Xiang Yu, Youmin Zhang, Sense and collision avoidance of unmanned aerial vehicles using Markov decision process and flatness approach, in: 2015 IEEE International Conference on Information and Automation, IEEE, 2015, pp. 714–719.
- Andrew Viquerat, Lachlan Blackhall, Alistair Reid, Salah Sukkarieh, Graham Brooker, Reactive collision avoidance for unmanned aerial vehicles using Doppler radar, in: *Field and Service Robotics*, Springer, 2008, pp. 245–254.
- Rethnaraj Rambabu, Muhammad Rijaluddin Bahiki, Syaril Azrad, Multi-sensor fusion based UAV collision avoidance system, *J. Teknol.* 76 (8) (2015).
- Enrique Aldao, Luis M. González-de Santos, Higinio González-Jorge, Lidar based detect and avoid system for UAV navigation in UAM corridors, *Drones* 6 (8) (2022) 185.
- Myungwhan Choi, Areeya Rubenecia, Taeshik Shon, Hyo Hyun Choi, Velocity obstacle based 3D collision avoidance scheme for low-cost micro UAVs, *Sustainability* 9 (7) (2017) 1174.
- Stephen Griffiths, Jeff Saunders, Andrew Curtis, B. Barber, T. McLain, R. Beard, Obstacle and terrain avoidance for miniature aerial vehicles, in: *Advances in Unmanned Aerial Vehicles*, Springer, 2007, pp. 213–244.
- Ian Gresham, Alan Jenkins, Robert Egri, Channabasappa Eswarappa, Noyan Kinayman, Nitin Jain, Richard Anderson, Frank Kolak, Ratana Wohlert, Shawn P. Bawell, et al., Ultra-wideband radar sensors for short-range vehicular applications, *IEEE Trans. Microw. Theory Tech.* 52 (9) (2004) 2105–2122.
- Steven Roelofsen, Denis Gillet, Alcherio Martinoli, Reciprocal collision avoidance for quadrotors using on-board visual detection, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2015, pp. 4810–4817.
- Yang Lyu, Quan Pan, Chunhui Zhao, Haifeng Zhu, Tongguo Tang, Yizhai Zhang, A vision based sense and avoid system for small unmanned helicopter, in: 2015 International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, 2015, pp. 586–592.
- Zhouyu Zhang, Yunfeng Cao, Meng Ding, Likui Zhuang, Jiang Tao, Monocular vision based obstacle avoidance trajectory planning for unmanned aerial vehicle, *Aerosp. Sci. Technol.* 106 (2020) 106199.
- Steven Roelofsen, Alcherio Martinoli, Denis Gillet, 3D collision avoidance algorithm for unmanned aerial vehicles with limited field of view constraints, in: 2016 IEEE 55th Conference on Decision and Control (CDC), IEEE, 2016, pp. 2555–2560.
- Liang Lu, Giancarmine Fasano, Adrian Carrio, Maolin Lei, Hriday Bavle, Pascual Campoy, A comprehensive survey on non-cooperative collision avoidance for micro aerial vehicles: sensing and obstacle detection, *J. Field Robot.* (2023).

- [43] David Alejo, Jose A. Cobano, Guillermo Heredia, Anibal Ollero, A reactive method for collision avoidance in industrial environments, *J. Intell. Robot. Syst.* 84 (2016) 745–758.
- [44] Carmine Tommaso Recchiuto, Antonio Sgorbissa, Renato Zaccaria, Visual feedback with multiple cameras in a UAVs human–swarm interface, *Robot. Auton. Syst.* 80 (2016) 43–54.
- [45] Wai Lun Leong, Pengfei Wang, Sunan Huang, Zhengtian Ma, Hong Yang, Jingxuan Sun, Yu Zhou, Mohamed Redhwan Abdul Hamid, Sutthiphong Srigrarom, Rodney Teo, Vision-based sense and avoid with monocular vision and real-time object detection for UAVs, in: 2021 International Conference on Unmanned Aircraft Systems ICUAS, IEEE, 2021, pp. 1345–1354.
- [46] Zhiqing Wei, Zeyang Meng, Meichen Lai, Huici Wu, Jiarong Han, Zhiyong Feng, Anti-collision technologies for unmanned aerial vehicles: recent advances and future trends, *IEEE Int. Things J.* 9 (10) (2021) 7619–7638.
- [47] Xiangpeng Xu, Sheng Zhuge, Chujun Li, Chenghao Ning, Lijun Zhong, Bin Lin, Xia Yang, Xiaohu Zhang, A vision-only relative distance calculation method for multi-UAV systems, *Aerosp. Sci. Technol.* 142 (2023) 108665.
- [48] Massimiliano Iacono, Antonio Sgorbissa, Path following and obstacle avoidance for an autonomous UAV using a depth camera, *Robot. Auton. Syst.* 106 (2018) 38–46.
- [49] Vitaly Shalunov, Reciprocal guidance approach for suspicious collision point identification in a multi-agent unmanned aerial vehicle scenario, *Aerosp. Sci. Technol.* 112 (2021) 106656.
- [50] Lucas Vago Santana, Alexandre Santos Brandão, Mário Sarcinelli-Filho, Navigation and cooperative control using the AR Drone quadrotor, *J. Intell. Robot. Syst.* 84 (1) (2016) 327–350.
- [51] Milton Cesar Paes Santos, Claudio Dario Rosales, Mário Sarcinelli-Filho, Ricardo Carelli, A novel null-space-based UAV trajectory tracking controller with collision avoidance, *IEEE/ASME Trans. Mechatron.* 22 (6) (2017) 2543–2553.
- [52] Paula Fraga-Lamas, Lucía Ramos, Víctor Mondéjar-Guerra, Tiago M. Fernández-Caramés, A review on IoT deep learning UAV systems for autonomous obstacle detection and collision avoidance, *Remote Sens.* 11 (18) (2019) 2144.
- [53] Jan Dentler, Martin Rosalie, Grégoire Danoy, Pascal Bouvry, Somasundar Kannan, Miguel Olivares-Mendez, Holger Voos, Collision avoidance effects on the mobility of a UAV swarm using chaotic ant colony with model predictive control, *J. Intell. Robot. Syst.* 93 (2019) 227–243.
- [54] Pierre-Jean Bristeau, François Callou, David Vissière, Nicolas Petit, The navigation and control technology inside the AR Drone micro UAV, *IFAC Proc. Vol.* 44 (1) (2011) 1477–1484.
- [55] Stephane Piskorski, Nicolas Brulez, Pierre Eline, Frederic DHaeyer, AR Drone Developer Guide, Parrot, SDK, 1, 2012.
- [56] J. Graff, The PS-drone-API: Programming a Parrot AR Drone 2.0 with Python-the easy way, 2012.