

Article

# Obstacle Detection and Avoidance System Based on Monocular Camera and Size Expansion Algorithm for UAVs

Abdulla Al-Kaff <sup>\*,†,‡</sup>, Fernando García <sup>‡</sup>, David Martín <sup>‡</sup>, Arturo De La Escalera <sup>‡</sup> and José María Armingol <sup>‡</sup>

Intelligent Systems Lab, Universidad Carlos III de Madrid, Leganes, 28911 Madrid, Spain; fegarcia@ing.uc3m.es (F.G.); dmgoomez@ing.uc3m.es (D.M.); escalera@ing.uc3m.es (A.D.L.E.); armingol@ing.uc3m.es (J.M.A.)

\* Correspondence: akaff@ing.uc3m.es; Tel.: +34-916-246-217

† Current address: Department of System Engineering and Automation, Avenida de la Universidad, 30, Leganes, 28911 Madrid, Spain.

‡ These authors contributed equally to this work.

Academic Editors: Felipe Gonzalez Toro and Antonios Tsourdos

Received: 19 December 2016; Accepted: 4 May 2017; Published: 7 May 2017

**Abstract:** One of the most challenging problems in the domain of autonomous aerial vehicles is the designing of a robust real-time obstacle detection and avoidance system. This problem is complex, especially for the micro and small aerial vehicles, that is due to the Size, Weight and Power (SWaP) constraints. Therefore, using lightweight sensors (i.e., Digital camera) can be the best choice comparing with other sensors; such as laser or radar. For real-time applications, different works are based on stereo cameras in order to obtain a 3D model of the obstacles, or to estimate their depth. Instead, in this paper, a method that mimics the human behavior of detecting the collision state of the approaching obstacles using monocular camera is proposed. The key of the proposed algorithm is to analyze the size changes of the detected feature points, combined with the expansion ratios of the convex hull constructed around the detected feature points from consecutive frames. During the Aerial Vehicle (UAV) motion, the detection algorithm estimates the changes in the size of the area of the approaching obstacles. First, the method detects the feature points of the obstacles, then extracts the obstacles that have the probability of getting close toward the UAV. Secondly, by comparing the area ratio of the obstacle and the position of the UAV, the method decides if the detected obstacle may cause a collision. Finally, by estimating the obstacle 2D position in the image and combining with the tracked waypoints, the UAV performs the avoidance maneuver. The proposed algorithm was evaluated by performing real indoor and outdoor flights, and the obtained results show the accuracy of the proposed algorithm compared with other related works.

**Keywords:** obstacle detection; collision avoidance; size expansion; feature points; UAV; monocular vision

## 1. Introduction

During the last decade, with the developments in microelectronics and the increase of computing efficiency, the use of Unmanned Aerial Vehicles (UAVs) is no longer restricted to the military purposes only. Recently, with the advent of small and micro aerial vehicles (sUAVs and MAVs), the requirements of operations and applications in low altitudes are increased.

Due to their ability to operate in remote, dangerous and dull situations, sUAVs and MAVs especially helicopters and vertical take-off and landing (VTOL) rotor-craft systems are increasingly used in many applications; such as surveying and mapping, rescue operation in

disasters [1], spatial information acquisition, data collection from inaccessible areas and geophysics exploration [2,3], cooperate in manipulation and transportation [4], buildings inspection [5,6] and navigation purposes [7,8].

Nowadays, with the current technology and the variety and complexity of the tasks, modern UAVs aim at higher levels of autonomy and performing flight stabilization. For the autonomous UAVs, the ability of detection and avoidance of obstacles with high level of accuracy is considered to be a challenging problem.

The difficulty appears because of the size of UAVs is getting smaller and thus, the weight is getting lighter. Therefore, taking into account these properties, sUAVs and MAVs have not the ability of carrying heavy sensors such as laser [9–11] or radar [12]. Hence, the suitable solution is to use the on-board cameras due to its advantage of lightweight and low power consumption.

In addition to its lightweight and the low power consumption, the cameras provide rich information of the environment. Therefore, they are considered as important sensors mounted on the small and micro UAVs.

In vision-based navigation systems, different approaches were presented to solve the problem of obstacle detection and avoidance. Approaches such as [13–15], built a 3D model of the obstacle in the environment. Other works calculate the depth (distance) of the obstacles, such as in [16,17]. A technique based on stereo cameras, in order to estimate the proximity of the obstacles, was introduced in [18]. At which, the system detects the size and the position of the obstacles based on the disparity images and the view angle. Furthermore, this technique calculates the relation of the size and the distance of the detected obstacle to the UAV. All these approaches have the disadvantage of the high cost in the computational time.

Whilst bio-inspired (insect, animal or human like) approaches estimate the presence of the obstacle efficiently, without calculating the 3D model, such as using optical flow [19–21] or perspective cues [7,22,23]. However, optical flow approaches cannot identify the forward movement, due to the aperture problem, thus frontal obstacles would provide only movement component normal to the detected edges in the image, not providing frontal movement information *per se*. Perspective cues approaches work well in the structured environments [24].

Detecting and avoiding frontal obstacles using monocular camera is considered a challenging problem because of the absence of the optical flow or the motion parallax. However, size expansion provides useful information for detecting the obstacles that are moving towards the UAV.

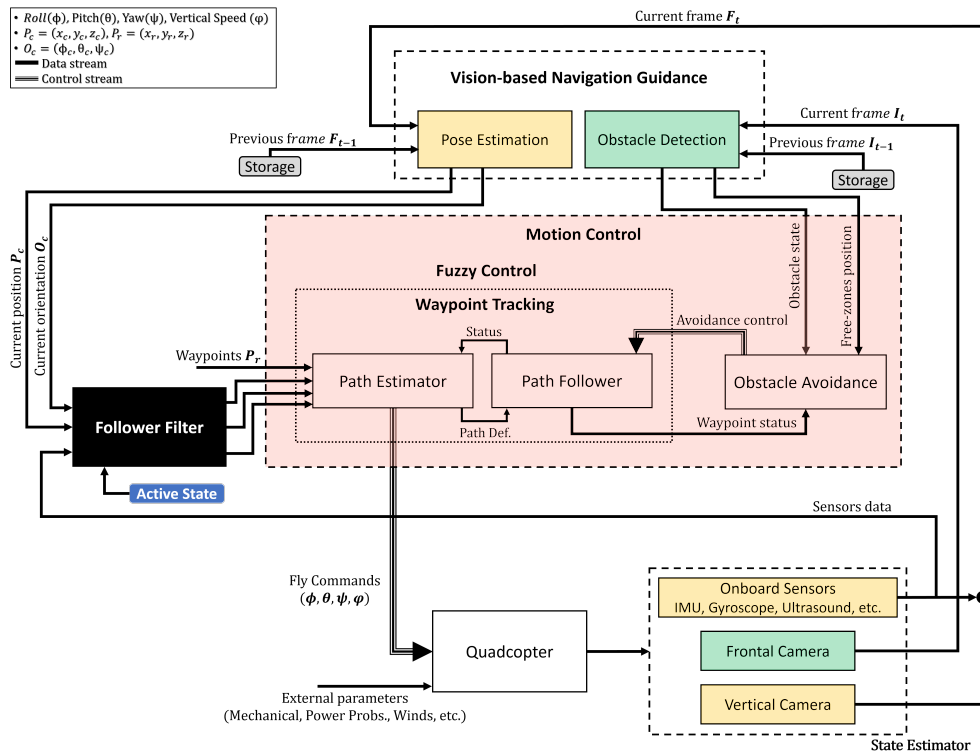
From the bio-inspired point of view, the human visual system has the ability to extract information correctly of the objects that are moving toward them [25]. In addition, Gibson illustrated the ability of the human visual system to identify the approaching of the objects related to the expansion of its size, by both eyes or even one eye [26].

From this aspect, in this paper, a bio-inspired approach using a monocular camera is presented to mimic the human behavior of obstacle detection and avoidance applied on UAVs. The system is divided into two main stages: **Vision-Based Navigation and Guidance** in which, the obstacle detection algorithm is performed based on the input images captured from the front camera. In addition, **Motion Control**, where the avoidance decision is taken and sent to the UAV. Figure 1a shows the general overview of the system, whilst Figure 1b depicts the subsystem which focuses on the detection and avoidance stages.

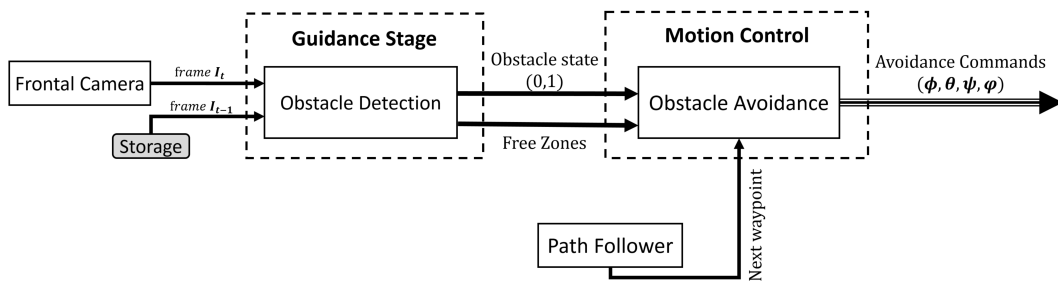
The novelty in this paper is based on two main lines: First, the use of the size changes of the detected feature points, in order to provide object detection. Second, the changes in the size ratio in consecutive frames of the convex hull constructed from these points allow reliable frontal obstacle detection by means of a monocular camera and the motion of the UAV. The presence of approaching obstacles is estimated from these size expansion ratios, avoiding the need of complex 3D models, as shown in Figure 2. Reducing considerably the computation cost of the detection algorithm.

The remainder of this paper is organized as follows; Section 2 introduces the state-of-the-art work related to obstacle detection and avoidance approaches, followed by the proposed obstacle

detection algorithm in Section 3. Section 4 presents the avoidance algorithm, then Section 5 discusses the experimental results. Finally, in Section 6 conclusion is summarized.



(a) System Overview.



(b) Detection and Avoidance Subsystem.

Figure 1. General overview of the Detection and Avoidance phases.

## 2. Related Work

Obstacle detection and avoidance plays an important role in any autonomous navigation system. Different works were presented to solve this challenging process especially in vision-based systems.

In [27], it was presented an approach based on the texture and color variation cue to detect obstacles for indoor environments. However, this approach works with detailed textures. Furthermore, their experiments were limited to indoor environments.

Working with Hybrid MAVs, Green et al. proposed an optical flow approach, mimicking the biological flying insects, by using dual cameras mounted on a fixed-wing UAV, in order to detect and avoid lateral obstacles [28]. Besides the detection of lateral obstacles only, some limitations appeared in avoiding large obstacles like walls. In addition, from the experiments, the avoidance algorithm is insufficient if the UAV flies in a straight path.

In [29], SIFT descriptor and Multi-scale Oriented-Patches (MOPS) are combined to show 3D information of the obstacles. At which, the edges and corners of the object are extracted using MOPS by obtaining and matching the MOPS feature points of the corners, then the 3D spatial information of the MOPS points is extracted. After that, SIFT is used to detect the internal outline information. However, the presented approach has expensive computational time (577 ms).

Bills et al. [7] proposed an approach for indoor environments with a uniform structure characteristics. In this work, Hough Transform is used to detect the edges that are used to classify the essence of the scene based on a trained classifier. However, their experiments were limited to corridors and stairs areas.

A saliency method based on Discrete Cosine Transform (DCT) is presented in [30] for obstacle detection purposes. From the input images, the system assumes that the obstacle is a unique content in a repeated redundant background, then by applying amplitude spectrum suppression, the method can remove the background. Finally, by using the Inverse Discrete Cosine Transform (IDCT) and a threshold algorithm, the center of the obstacle is obtained. Furthermore, a pinhole camera model is used to estimate the relative angle between the UAV and the obstacle, this angle is used with a PD controller to control the heading of the UAV for obstacle avoidance.

In [17], the authors presented an approach for measuring the relative distance to the obstacle. At which, the camera position is estimated based on the Extended Kalman Filter (EKF) and the IMU data. Then the 3D position of the obstacle can be calculated by back projecting the detected features of the obstacle from its images.

An expansion segmentation method was presented in [31], in which a conditional Markov Random Field (MRF) is used to distinguish if the frontal object may represent a collision or not. Additionally, an inertial system is used to estimate the collision time. However, the experiments of this work was limited to simulations.

Another approach presented in [24] used the feature detection algorithm in conjunction with the template matching to detect the size expansions of the obstacles. However, the experiments were limited to tree-like obstacles and did not show results of other shapes.

Kim et al. presented a block-based motion estimation approach for detecting the moving obstacles (humans) [32]. In which, the input image is divided into smaller blocks, then the system compare the motion in each block through consecutive images. However, their experiments were limited in detecting large size obstacles (humans) in indoor environments.

In addition, surveys of different approaches of UAVs guidance, navigation and collision avoidance methods and technologies are presented in [33,34]. Recently, Mcfadyen et al. presented a literature review of the vision-based collision avoidance systems [35].

The work presented in this paper represents a step forward in the obstacle detection and avoidance by the use of the convex hull (area) of the obstacle identified by means of image features. The approach is able to work in real time with all different kind of obstacles and in different tested scenarios.

### 3. Obstacle Detection

The proposed obstacle detection algorithm mimics the human behavior of detecting the obstacles that are located in front of the UAV during motion. At which point, the collision state of the approaching obstacles is estimated instead of building 3D models, or calculating the depth of the obstacle.

The novelty and the key of this algorithm is to estimate the size ratios of the approaching obstacles from the consecutive frames during the flight as shown in Figure 2. This is achieved by estimating the change in the size property of the detected feature points (diameter), and the size of the convex hull (area) which is constructed from these points as well. When the size ratios exceed certain empirical values (explained in Section 3.2), it means that there is an obstacle detected, and can cause a danger to the UAV as shown in Algorithm 1, and Figure 3.

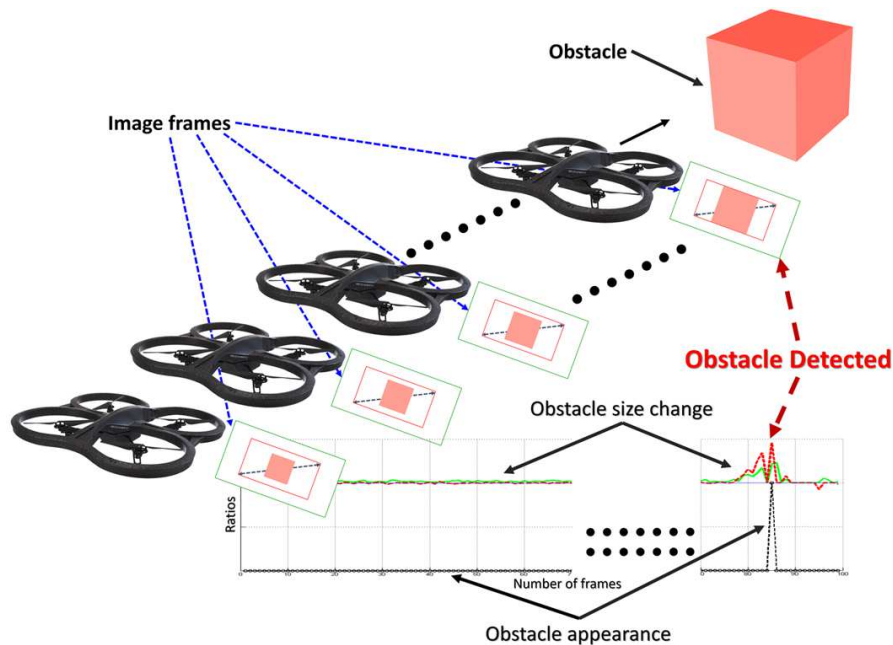
**Algorithm 1:** Obstacle Detection

---

**Input:** Image frames  $F$   
**Output:** Collision state  $Obs$ , Obstacle position  $\tau$

- 1 **Define:** Current frame  $F_t$ , Previous frame  $F_{t-1}$ , Current workspace  $ROI_t$ , Previous workspace  $ROI_{t-1}$ , Current keypoints  $KP_t[ ]$ , Previous keypoints  $KP_{t-1}[ ]$ , Number of keypoints  $(N, M)$ ,  $n \in N, m \in M, n = m, x \in n$ , Matched points  $pts$ , Distance ratio  $thresh$ , Object of interest  $Convex$
- 2 **begin**
- 3     **while**  $isFlying()$  **do**
- 4          $F_{t-1} \leftarrow getNewFrame()$
- 5         **if**  $Obstacle\ Detection\ isActivated()$  **then**
- 6              $F_t \leftarrow getNewFrame()$
- 7              $(ROI_{t-1}, ROI_t) \leftarrow DefWrkspc(F_{t-1}, F_t)$
- 8              $(KP_{t-1}(N), KP_t(M)) \leftarrow DetectKeypoints(F_{t-1}, F_t)$
- 9              $(pts_{t-1}(n), pts_t(m)) \leftarrow MatchSymKeypoints(KP_{t-1}(N), KP_t(M), thresh)$
- 10              $(pts_{t-1}(x), pts_t(x)) \iff size(pts_t(m)) > size(pts_{t-1}(n))$
- 11              $(Convex_1, Convex_2) \leftarrow Create(pts_{t-1}(x), pts_t(x))$
- 12             **if**  $AreaScale(Convex_2 : Convex_1) \geq 1.7$  and  $SizeScale(pts_t(x) : pts_{t-1}(x)) \geq 1.2$  **then**
- 13                  $Obs \leftarrow true$
- 14                  $\tau \leftarrow CalcPos(Obs)$
- 15                 **goto** Algorithm 3
- 16             **end**
- 17         **end**
- 18          $F_{t-1} \leftarrow F_t$
- 19     **end**
- 20 **end**

---



**Figure 2.** Concept of approaching obstacle detection.

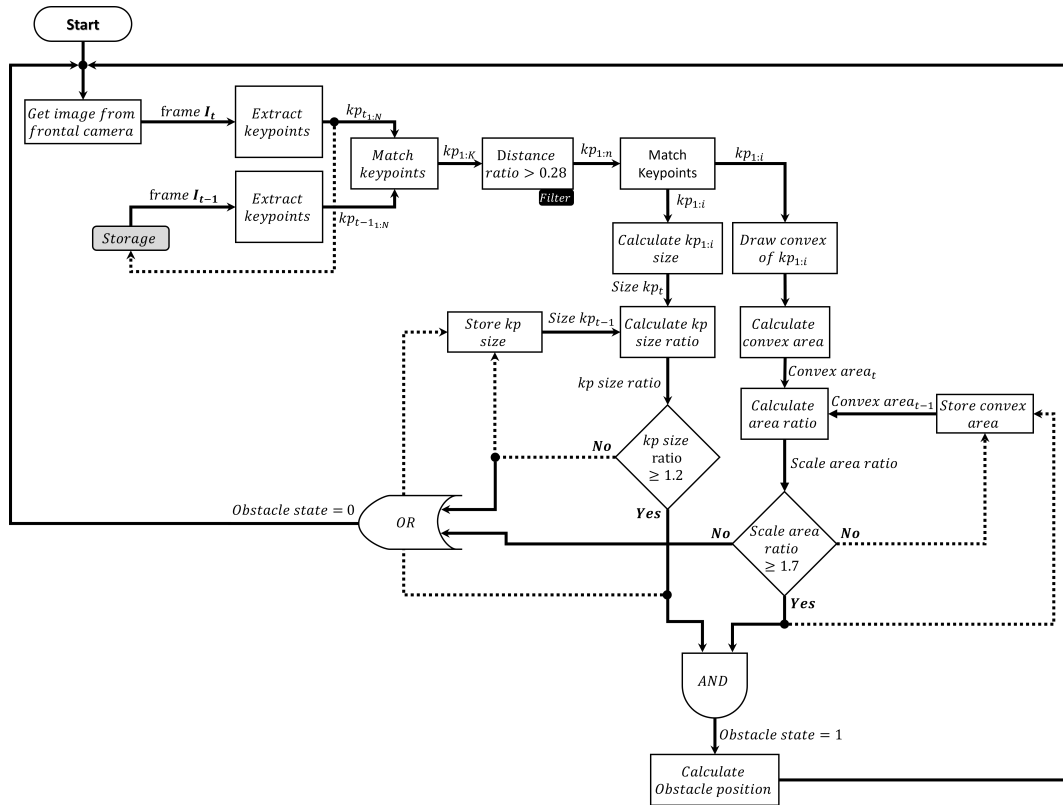


Figure 3. Obstacle detection approach flowchart.

3.1. Feature Detection and Description

In this step, an image Region Of Interest (ROI) of diagonal 62° Field of View (FOV) is taken, in order to be processed instead of the whole image, as shown in Figure 4. The selection of the diagonal 62° ROI is based on the results that are obtained from the experiments. Where, it has been found that any object detected out of the area of this ROI will not cause any danger to the UAV, and only the objects that are detected in the scope of this diagonal 62° ROI can be considered as an obstacle. Furthermore, processing the diagonal 62° ROI instead of the whole diagonal 92° image, leads to a significant minimizing in computational time. Test performed proved the viability of this approach, and the results will be discussed in following sections.

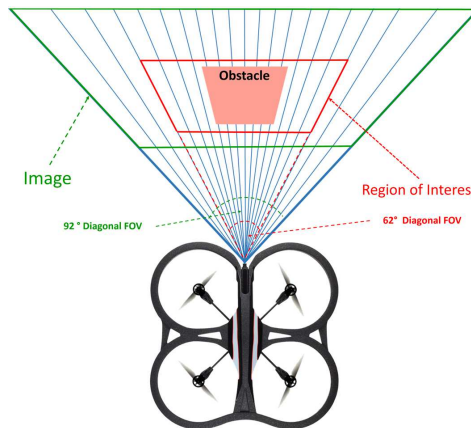


Figure 4. Define the diagonal 62° patch from the whole 92° image FOV.

Due to flying in unknown environments, the captured frames are affected by different conditions; such as illumination variation which may induce to noise and error. However, the keypoints need to be extracted accurately even under these conditions. Therefore the SIFT detector algorithm is used; because of the ability to identify and localize accurately the feature points even under different image condition specially scale and rotation properties.

According to Algorithm 1, all the keypoints are detected and its descriptors are extracted from the two consecutive frames as shown in Figure 5, then a vector of the position  $(x, y)$  and the size  $(s)$  of each keypoint is obtained.

After detection the keypoints, the Brute-Force algorithm is applied to match the keypoints from the two frames, and only the points that are found in both frames are returned. Algorithm 2 illustrates the concept of the Brute-Force method to match, and find the smallest distance of a pair of points.

---

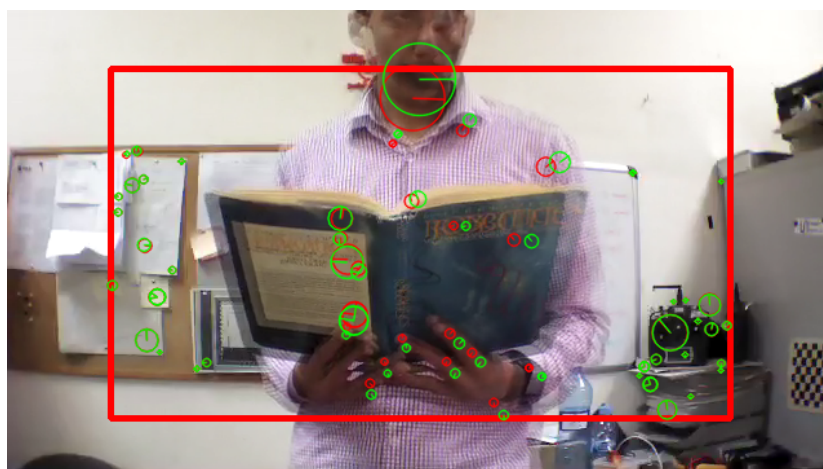
**Algorithm 2:** Brute-Force Matcher
 

---

**Input:** Array of keypoints in first image  $kp_{t-1}[N]$ ,  
 Array of keypoints in second image  $kp_t[M]$   
**Output:** Indices of the matched points  $index_1, index_2$

- 1 **Define:** Keypoints  $kp_1 = (x_1, y_1), \dots, kp_n = (x_n, y_n)$ , Minimum distance  $d_{min}$ , Distance between keypoints  $d$
- 2 **begin**
- 3      $d_{min} \leftarrow \infty$
- 4     **for**  $i \leftarrow 1, N$  **do**
- 5         **for**  $j \leftarrow i + 1, N$  **do**
- 6              $d \leftarrow \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$
- 7             **if**  $d < d_{min}$  **then**
- 8                  $d_{min} \leftarrow d$
- 9                  $index_1 \leftarrow i$
- 10                  $index_2 \leftarrow j$
- 11             **end**
- 12         **end**
- 13     **end**
- 14 **end**

---



**Figure 5.** Keypoints extraction from two consecutive frames; keypoints extracted from frame  $f_{t-1}$  (red) and keypoints extracted from frame  $f_t$  (green).

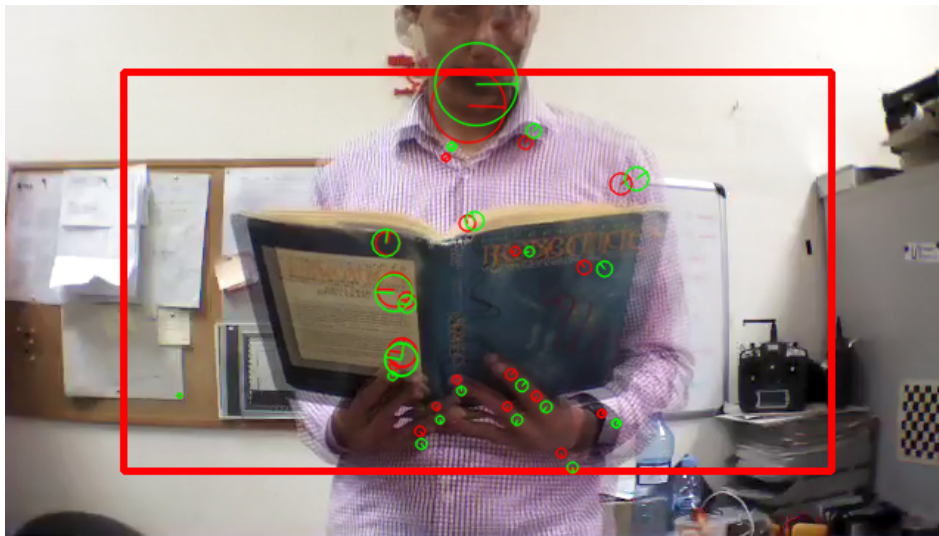
For more accuracy, the matched keypoints are filtered, by eliminating the ones that have a minimum distance ratio bigger than an empirical threshold value (0.28). Let  $mkp$  is the filtered-matched keypoint which are calculated as follows:

$$mkp(n) = \begin{cases} (x, y, s), & \text{distratio} \leq 0.28 \\ 0, & \text{otherwise} \end{cases} \quad \forall n \in K \quad (1)$$

where,  $s$  is the size of the keypoint (diameter),  $distratio$  is the minimum distance ratio of the matched keypoints, and  $K$  is the total number of the matched keypoints.

Afterwards, the obtained keypoints by Equation (1) are compared from the second to the first frame, and then the algorithm return the matched keypoints if and only if its size is growing, as shown in Figure 6:

$$mkp(i) = \begin{cases} (x, y, s), & \text{Size}(mkp_2(i)) > mkp_1(i) \\ 0, & \text{otherwise} \end{cases} \quad \forall i \in n \quad (2)$$



**Figure 6.** Filtered keypoints where the size expand from the second frame to the first frame; keypoints extracted from frame  $f_{i-1}$  (red) and keypoints extracted from frame  $f_i$  (green).

### 3.2. Object of Interest (OOI)

The next step of the detection algorithm is to determine the probability to detect a frontal obstacle. Hence, from the extracted and filtered keypoints by Equation (2), an Object of Interest (OOI) is created around these keypoints in both frames, by creating a convex hull of the corresponding points, as it is shown in Figure 7:

$$C = \sum_{i=1}^N \lambda_i mpk_i | (\forall i : \lambda_i \geq 0) \quad (3)$$

where  $C$  defines the convex hull, and  $\lambda_i$  is a non-negative weight assigned to the keypoints  $mpk_i \in N$  and  $\sum_{i=1}^N \lambda_i = 1$ .

Next, in order to estimate the changes in the size of the area of the detected obstacles, it is considered that each convex hull as an irregular polygon. Therefore, for a given  $C$  as a convex hull, the area of  $C$  can be calculated as follows:



$$C_{area} = \frac{1}{2} \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ \vdots & \vdots \\ x_n & y_n \\ x_1 & y_1 \end{vmatrix} = \frac{1}{2} [(x_1y_2 + x_2y_3 + x_3y_4 + \dots + x_ny_1) - (y_1x_2 + y_2x_3 + y_3x_4 + \dots + y_nx_1)] \quad (4)$$

where  $x_{(1:n)}$  and  $y_{(1:n)}$  are vertices, and  $n$  is the number of sides of the polygon.

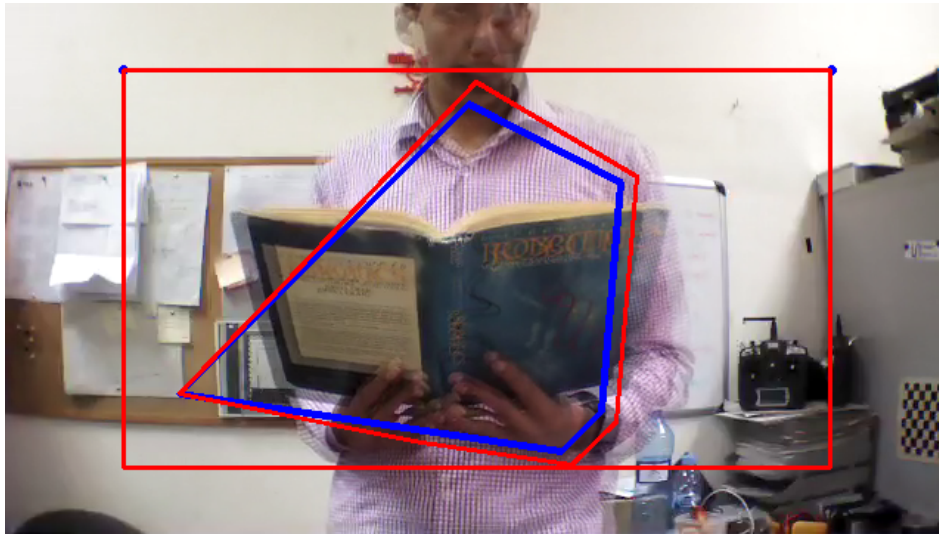
Finally, the size ratio of the matched keypoints, and the area of the convex hull from the second to the first frame are calculated respectively as follows:

$$ratio(mkp) = \frac{1}{N} \sum_{i=1}^N \frac{Size(mkp_2(i))}{Size(mkp_1(i))} \quad (5)$$

$$ratio(C) = \frac{Size(C_2)}{Size(C_1)} \quad (6)$$

Then, the algorithm estimates the collision state, if the approaching obstacle may represent a collision or not.

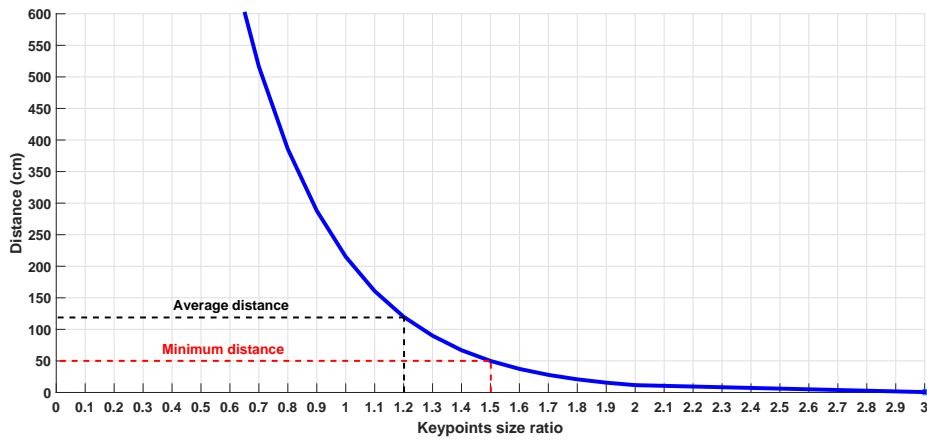
$$State = \begin{cases} 1, & ratio(mkp) \geq 1.2 \wedge ratio(C) \geq 1.7 \\ 0, & otherwise \end{cases} \quad (7)$$



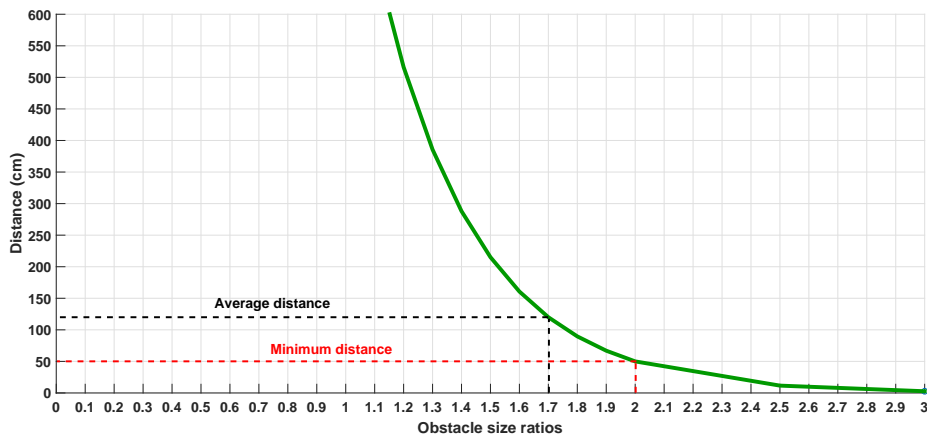
**Figure 7.** Convex Hull construction from detected keypoints in both frames; frame  $f_{t-1}$  (blue) and  $f_t$  (red).

Next, an empirical study about the relation of the ratios between the size of the keypoints, the area of the obstacle and the distance of the approaching obstacle has been developed and the results are illustrated in Figure 8. This relation has been estimated by performing different indoor and outdoor experiments. Assuming that the UAV is flying at a constant velocity, the best ratios are in the range of [1.2–1.5], and [1.7–2.0] for keypoints size and obstacle size area respectively, at which the obstacle can be detected in a distance of [120–50] cm.

Figure 9 shows the collision state of the detected obstacles by the monocular camera, where it provides 1 if there is an obstacle, or it provides 0 if there is no obstacle detected.



(a) Keypoints.



(b) Obstacle Area.

Figure 8. Distance and Ratios relation.

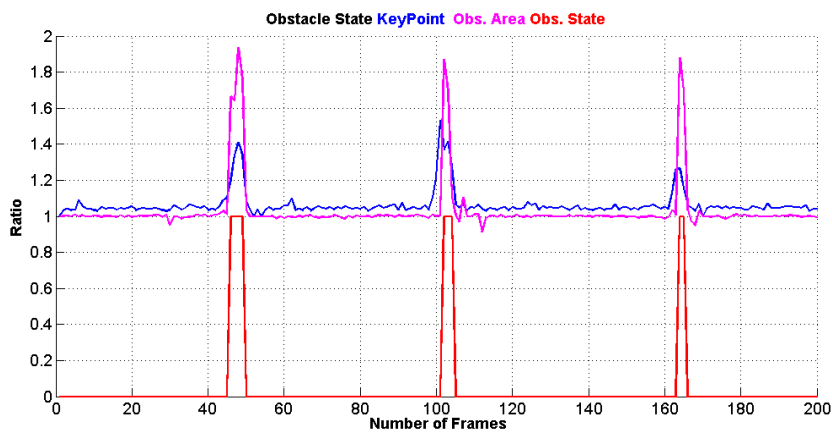


Figure 9. Obstacle State: Keypoint size ratio (Blue), Convex hull area ratio (Magenta) and Obstacle State (0) not found (1) found (Red).

In this step, after detecting the obstacles with a collision state value 1, the algorithm estimates the position of the extremely outer points that construct the obstacle in the image ( $P_l, P_r, P_u, P_d$ ), as it is

shown in Figure 10, where  $P_l$  is the point the of a position that has the minimum  $x$  value,  $P_r$  has the maximum  $x$  value, and similarly,  $P_u$  and  $P_d$  have the  $y$  minimum and maximum values respectively.

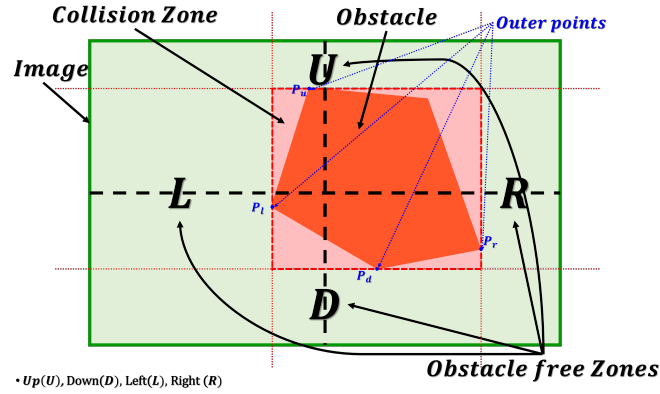


Figure 10. Estimating Obstacle outer points.

Finally, the collision-free zones *Left*, *Right*, *Up* and *Down* (in case of hanged or flying obstacles) are calculated as four rectangles surrounding the obstacle as shown in Equation (8):

$$\begin{aligned} \tau &= (\tau_l, \tau_r, \tau_u, \tau_d) \\ &= (\text{Zone}_{L_{width}}, \text{Zone}_{R_{width}}, \text{Zone}_{U_{height}}, \text{Zone}_{D_{height}}) \end{aligned} \quad (8)$$

where,  $\text{Zone}_{L_{width}}$ ,  $\text{Zone}_{R_{width}}$ ,  $\text{Zone}_{U_{height}}$  and  $\text{Zone}_{D_{height}}$  are the width and the height of the rectangles that are created by the points  $(P_l, P_r, P_u, P_d)$ , as follows:

$$\begin{aligned} \text{Zone}_L &= \text{Rectangle} \left[ (0, P_{u_y}), (P_{l_x}, P_{u_y}), (0, P_{d_y}), (P_{l_x}, P_{d_y}) \right] \\ \text{Zone}_R &= \text{Rectangle} \left[ (P_{r_x}, P_{u_y}), (ROI_w, P_{u_y}), (P_{r_x}, P_{d_y}), (ROI_w, P_{d_y}) \right] \\ \text{Zone}_U &= \text{Rectangle} \left[ (P_{l_x}, 0), (P_{r_x}, 0), (P_{l_x}, P_{u_y}), (P_{r_x}, P_{u_y}) \right] \\ \text{Zone}_D &= \text{Rectangle} \left[ (P_{l_x}, P_{d_y}), (P_{r_x}, P_{d_y}), (P_{l_x}, ROI_h), (P_{r_x}, ROI_h) \right] \end{aligned} \quad (9)$$

where,  $w$  and  $h$  are the width and the height of the ROI respectively.

#### 4. Obstacle Avoidance

In this section, the combined mission of the waypoint tracking and the avoidance method is described. The geometrical problem is shown in Figure 11 where the avoidance technique is summarized in Algorithm 3.

To define the problem of waypoint tracking, let the UAV  $X$  flying at a velocity  $V$ , considering the UAV flies forward at a constant velocity along its  $x$ -axis, where:

$$\begin{aligned} X &= [x_d \quad y_d \quad z_d]^T \quad \text{and,} \\ V &= [u_d \quad v_d \quad w_d]^T \end{aligned} \quad (10)$$

On the other hand, let a waypoint:

$$WP = [x_w \quad y_w \quad z_w]^T \tag{11}$$

hence the waypoint is assumed to be tracked if  $x_d$  is achieved when both  $y_d$  and  $z_d$  are satisfied, where:

$$\begin{aligned} x_d &= x_w \pm \mu_x \\ y_d &= y_w \pm \mu_y \\ z_d &= z_w \pm \mu_z \end{aligned} \tag{12}$$

where,  $\mu$  is the tolerance area around the waypoint position with a radius of 10 cm from the waypoint.

Let a frontal obstacle *obs* be detected by Algorithm 1, situated in the UAV path and surrounded by the collision-free zones  $\tau = (\tau_l, \tau_r, \tau_u, \tau_d)$ .

First, the avoidance algorithm checks all the free zones and differentiate which zone is the best to be followed. This is done by reading the position of the next waypoint and by comparing the size of the free zones, where the final maneuver will be in term of **Left-Right** or **Up-Down** motion or a combination of both. After that, a safety boundary surrounding the obstacle is assumed as shown in Figure 11, which is based on the dimensions of the UAV. This safety region is estimated to be:

$$Safety_{lr} = \left(\frac{w_{UAV}}{2}\right) + 20 \text{ [cm]} \tag{13}$$

and

$$Safety_{ud} = \left(\frac{h_{UAV}}{2}\right) + 20 \text{ [cm]} \tag{14}$$

where,  $w$  and  $h$  defines the width and the height of the UAV respectively.

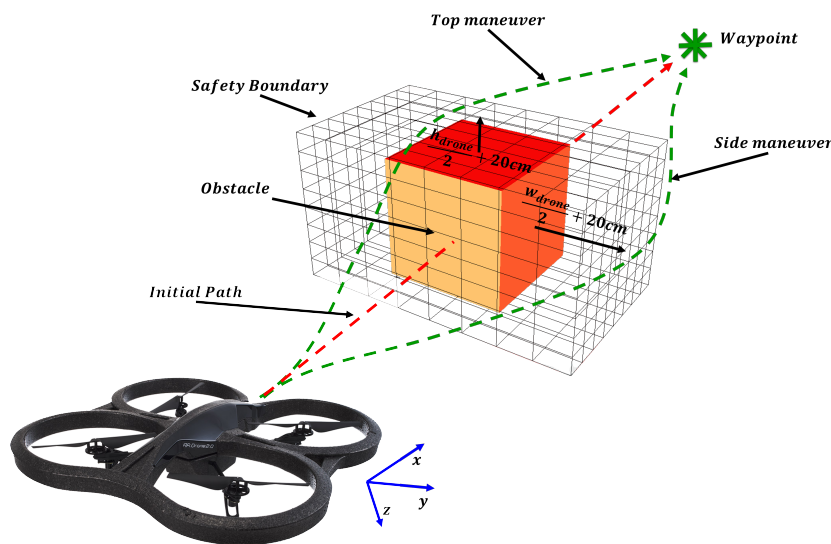


Figure 11. Obstacle Avoidance Path.

Afterwards, the algorithm reads the position of the predefined next waypoint, and calculates the new waypoint out of the path (in order to avoid the obstacle), and sends a control command (velocity control) to the UAV for maneuvering according to the waypoint position as follows:

- Horizontal maneuver (*Right* or *Left*)

$$V_{lr} = \kappa(y_d \pm Safety_{lr}) \tag{15}$$

- Vertical maneuver (*Top or bottom*)

$$V_{ud} = \kappa(z_d \pm \text{Safety}_{ud}) \quad (16)$$

where  $y_d$  and  $z_d$  are the UAV position in  $(y, z)$ -coordinates, and  $\kappa$  is a control coefficient.

---

**Algorithm 3: Obstacle Avoidance Algorithm**


---

**Input:** Collision-free zone  $\tau$ , UAV position  $X = (x_d, y_d, z_d)$

**Output:** Navigation command  $Nav(\phi, \theta, \psi, \vartheta)$

1 **Define:** Image zones  $(\tau_l, \tau_r, \tau_u, \tau_d)$ , Roll  $\phi$ , Pitch  $\theta$ , Yaw  $\psi$ , Vertical speed  $\vartheta$ , Next waypoint  $WP = (x_w, y_w, z_w)$ , Tolerance  $(\text{thresh}_1, \text{thresh}_2)$  and Maneuver behavior  $(Mr_{rl}, Mr_{ud})$

2 **Control Commands:**

$$3 \quad +\Delta H \leftarrow z_d + \left(\frac{h_{UAV}}{2}\right) + 20 \text{ cm} \quad -\Delta H \leftarrow z_d - \left(\frac{h_{UAV}}{2}\right) + 20 \text{ cm}$$

$$4 \quad +\Delta Y \leftarrow y_d + \left(\frac{w_{UAV}}{2}\right) + 20 \text{ cm} \quad -\Delta Y \leftarrow y_d - \left(\frac{w_{UAV}}{2}\right) + 20 \text{ cm}$$

5 **begin**

6 **while**  $!atGoal$  **do**

7 **if**  $ObAreaRatio \leq \underline{2}$  and  $KPSizeRatio \leq \underline{1.5}$  **then**

8  $WP \leftarrow Read()$

9 **if**  $\tau_l$  and  $\tau_r = 0$  **then** // No free-zones in Right or Left

10  $Mr_{rl} \leftarrow 0$

11 **else if**  $\tau_l = \tau_r$  **then** // Right and Left free-zones are equal

12 **if**  $y_w > y_d$  **then** // Compare  $y$  position of the UAV and next waypoint

13  $Mr_{rl} \leftarrow \tau_r$  // Go Right

14 **else**

15  $Mr_{rl} \leftarrow \tau_l$  // Go Left

16 **else**

17  $Mr_{rl} \leftarrow \max(\tau_l, \tau_r)$  // Choose the wider free-zones

18  $\phi \leftarrow CalcControl(Mr_{rl}, \tau_l, \tau_r)$  // See Algorithm 4

19 **if**  $\tau_u$  and  $\tau_d = 0$  **then** // No free-zones in UP or Down

20  $Mr_{ud} \leftarrow 0$

21 **else if**  $\tau_u = \tau_d$  **then** // Up and Down free-zones are equal

22 **if**  $z_w > z_d$  **then** // Compare  $z$  position of the UAV and next waypoint

23  $Mr_{ud} \leftarrow \tau_u$  // Go Up

24 **else**

25  $Mr_{ud} \leftarrow \tau_d$  // Go Down

26 **else**

27  $Mr_{ud} \leftarrow \max(\tau_u, \tau_d)$  // Choose the wider free-zones

28  $\vartheta \leftarrow CalcControl(Mr_{ud}, \tau_u, \tau_d)$  // See Algorithm 4

29  $\theta \leftarrow 1, \psi \leftarrow 0$

30  $X \leftarrow Nav(\phi, \theta, \psi, \vartheta)$

31 **else**

32  $X \leftarrow Nav(0, 0, 0, 0)$

33 **goto** Algorithm 1 // See Detection Algorithm 1

---

**Algorithm 4:** Calculate Avoidance Control

---

**Input:** Image zones  $(\tau_l, \tau_r, \tau_u, \tau_d)$  and Maneuver behavior  $(Mr_{rl}, Mr_{ud})$   
**Output:** Roll  $\phi$ , Pitch  $\theta$ , Yaw  $\psi$ , Vertical speed  $\vartheta$

- 1 **Define:** Control gains  $K_{ph}, K_{p\phi}$ , UAV position  $X = (x_d, y_d, z_d)$
- 2 **if**  $Mr_{rl} = \tau_l$  **then**
- 3      $\phi \leftarrow K_{p\phi} (-\Delta Y - y_d)$
- 4 **else if**  $Mr_{rl} = \tau_r$  **then**
- 5      $\phi \leftarrow K_{p\phi} (+\Delta Y - y_d)$
- 6 **else**
- 7      $\phi \leftarrow 0$
- 8 **if**  $Mr_{ud} = \tau_u$  **then**
- 9      $\vartheta \leftarrow K_{ph} (+\Delta H - z_d)$
- 10 **else if**  $Mr_{ud} = \tau_d$  **then**
- 11      $\vartheta \leftarrow K_{ph} (-\Delta H - z_d)$
- 12 **else**
- 13      $\vartheta \leftarrow 0$

---

Finally, by estimating the new UAV position after avoidance, the algorithm recalculates the new waypoints in order the UAV to be able to return back to its predefined path and activate the detection process.

In the case that the *AreaScale* is greater than 2 and the *SizeScale* of the keypoints is greater than 1.5, a "Hover" command is sent to the UAV. That is because if the ratios exceed these limits, this means that the obstacle is very close to the UAV (less than 50 cm), as it is shown in Figure 8.

## 5. Experimental Results

In order to evaluate the performance of the proposed algorithms in the previous sections, 100 different real flight experiments have been carried out, in both indoor and outdoor environments, with a total number of 1000 obstacles, taking in consideration the visual conditions (the illumination and the texture of the obstacles) which affect the accuracy of the detection.

### 5.1. Platform

The processing in the ground station is performed in Intel i7-3770 at 3.4 GHz CPU, with 6 GB DDR3 RAM. The connection with the UAV is established via a standard 802.11n wireless LAN card.

The experiments have been performed with a Parrot AR.Drone 2.0 quadcopter [36]. This control system is governed by the inputs (roll, pitch, yaw angles, and vertical speed), therefore the implemented controller realize the UAV actual position, orientation and velocity.

One of the most important aspects in the avoidance phase is based on the robust control system for the UAV at which it is necessary to know its dynamic model. However, to avoid the complexity in modeling, the avoidance control was applied over the internal control of the system, modifying the roll and the vertical speed in order to perform the maneuvers in  $y$  and  $z$  directions.

### 5.2. Scenarios

Two different scenarios have been conducted, in order to evaluate the performance of the proposed algorithms in both motion and stability, with data gathered from the experiments to test the detection and the estimation of the position of the obstacle. In each scenario, different types of obstacles (people, obstacles, pillars, trees and walls), (static and dynamic) are situated.

The first scenario is a predefined straight flight, where the UAV flies in a straight line from the starting point to the end point. Different types of obstacles with unknown previous position were situated in the UAV path. The goal of this scenario is to evaluate the accuracy, and robustness in detecting, and avoiding the obstacles in motion.

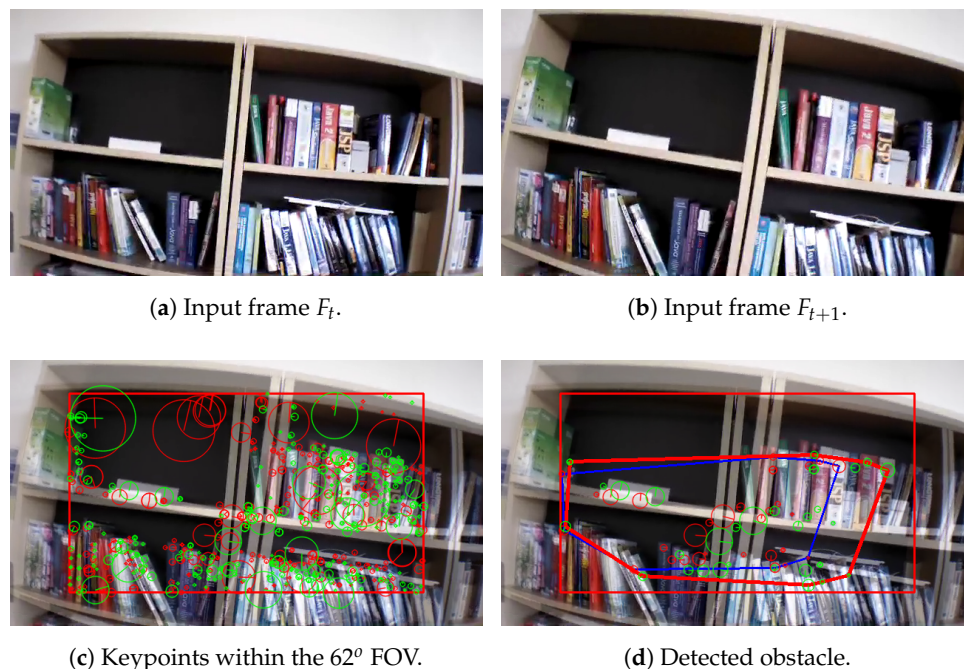
The second scenario is a hover stability flight. At which, the UAV enters to the hover flight mode, and different obstacle are approaching to it. Once an obstacle is detected, the UAV flights in the opposite direction of the obstacle (*Backward maneuver*).

### 5.3. Results (Obstacle Detection)

From the experiments, the obtained results demonstrate that the algorithm is able to detect the obstacles with different sizes (areas) between 8500 and 200,000 pixels, and at a distance range between 90 and 120 cm. It is shown that the minimum accuracy of the algorithm is 95.0%, and the overall accuracy is 97.4% as it is demonstrated in Table 1.

**Table 1.** Accuracy of Detection Algorithm.

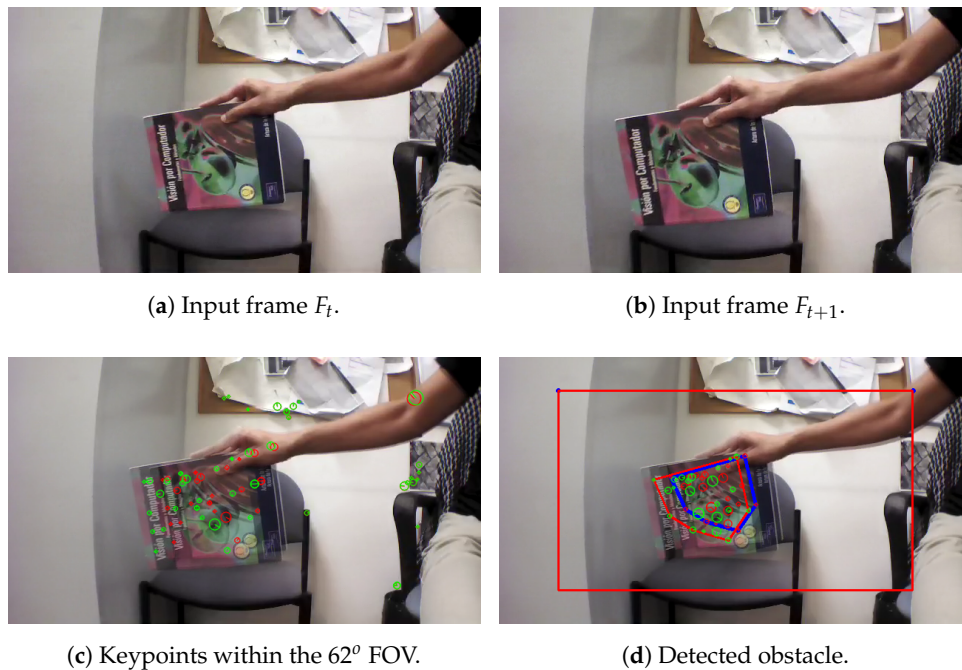
	Indoor				Outdoor				Total
	People	Obstacle	Pillar	Wall	People	Obstacle	Tree	Wall	
Situated	200	110	80	80	200	120	140	70	1000
Detected	196	107	79	76	196	116	135	69	974
Fail	4	3	1	4	4	4	5	1	26
Accuracy (%)									
Object	98.0	97.3	98.8	95.0	98.0	96.7	96.4	98.6	97.4
Environment		97.3				97.4			



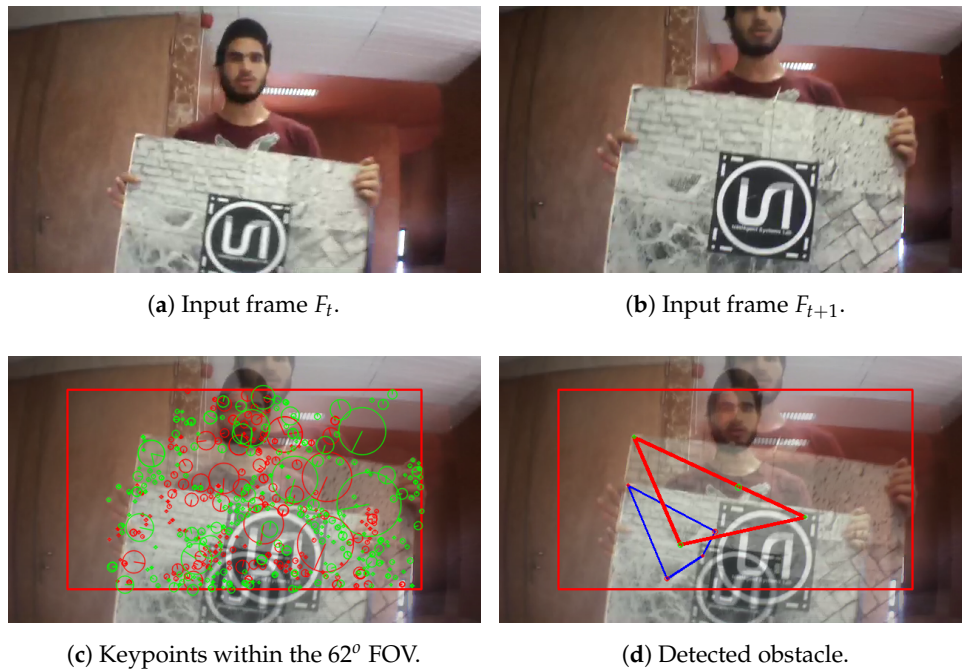
**Figure 12.** Obstacle detection:  $ratio(mkp) = 1.27$ ,  $ratio(C) = 1.76$  and  $distance = 114$  cm.

Figures 12–14 illustrate the detection process of various approaching obstacles, with different size ratios. Where, Figures 12a,b, 13a,b and 14a,b are showing the two input consecutive frames to be processed. In Figures 12c, 13c or 14c it is shown the total number of the detected and matched

keypoints before filtering its size expansion property. Finally, the filtered keypoints and the constructed polygon of the detected obstacle are shown in Figures 12d, 13d and 14d.



**Figure 13.** Obstacle detection:  $ratio(mkp) = 1.25$ ,  $ratio(C) = 1.71$  and  $distance = 92$  cm.



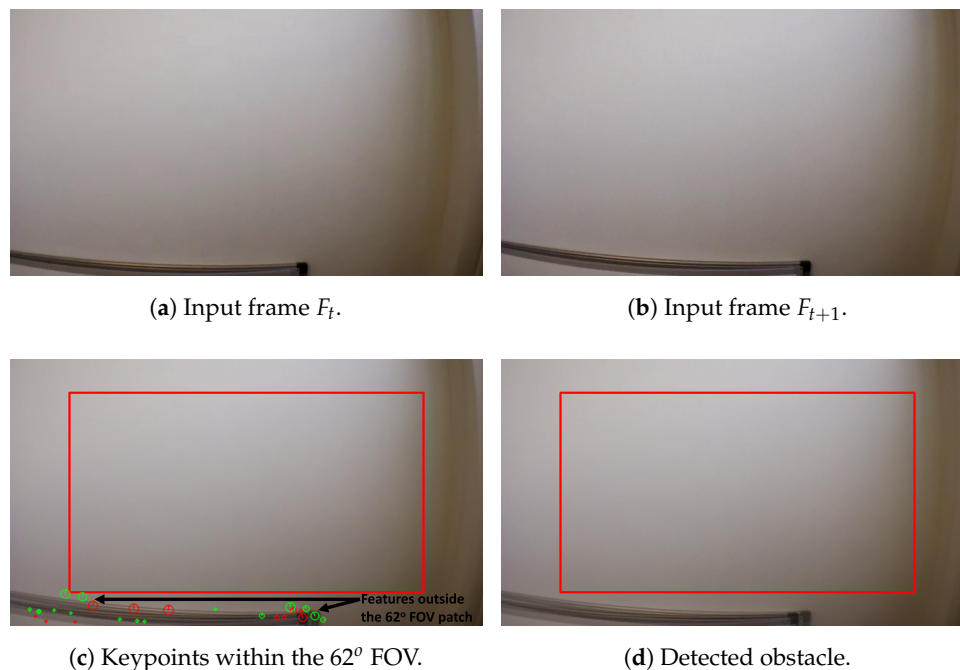
**Figure 14.** Obstacle detection:  $ratio(mkp) = 1.20$ ,  $ratio(C) = 2.15$  and  $distance = 126$  cm.

Table 1 summarizes the accuracy of the detection algorithm. The table shows the total number of the obstacles that either situated in the UAV path (*first scenario*) or moving towards the UAV (*second scenario*), the number of the detected obstacles and the number of fails.



From the table, it is illustrated that the accuracy of the detection process in the indoor scenarios is better than the accuracy in outdoor environments. This is due to the constancy of the light conditions in indoors rather than outdoors, which are suffered from various lighting effects.

Two main reasons for the fail of detection; the first one is the disability of extracting sufficient number of keypoints, and that is either because of the low light conditions or because of the absence of the texture on the obstacle surface such as in the case of some pillars and walls as shown in Figure 15.



**Figure 15.** Obstacle detection fail (wall) (absence of texture):  $ratio(mkp) = 1$  and  $ratio(C) = 1$ .

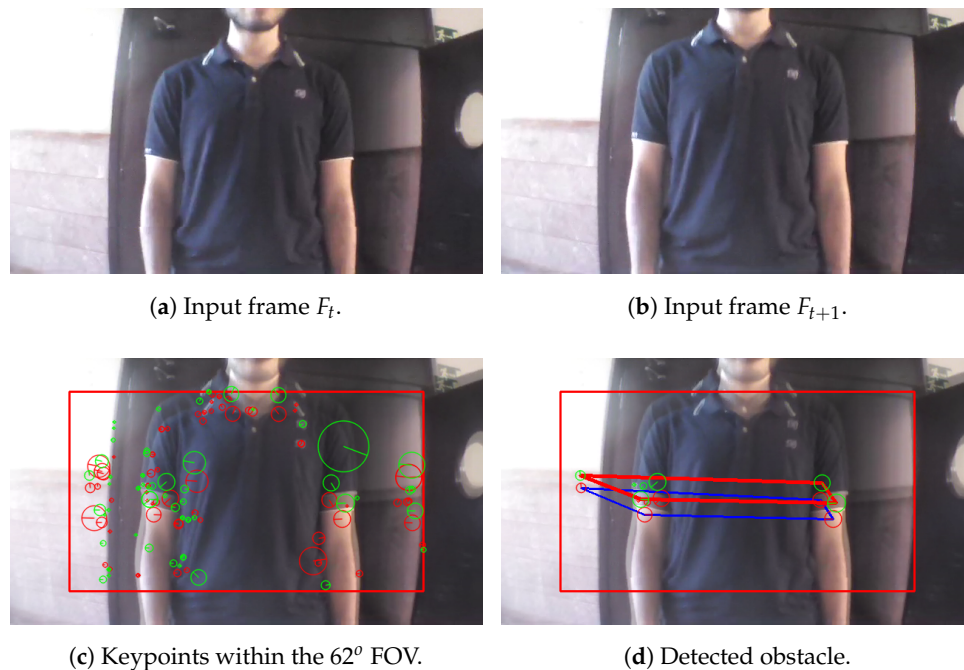
The second reason is the direction of the motion of the obstacle, the algorithm is able to detect the moving obstacle if the motion is towards the UAV.

Figure 16 shows an example of the second scenario, where the UAV flies in hover mode, and the object is moving, however, this movement is not in the direction of the UAV. Therefore, it does not consider as an obstacle.

However, in most cases of the moving obstacles according to Table 1, the algorithm could not detect the appearance of the obstacles if the motion is around the UAV such as in the case of the people and obstacles.

In addition, the proposed algorithm is evaluated against two related works of detecting frontal obstacles based on monocular vision. As it shown in Table 2, the proposed algorithm provides more accuracy (97.4%) comparing to *SURF + Template matching* method [24] which provides 97%, and *relative distance estimation* approach [17] that provides 97.1% of accuracy.

Furthermore, the computational time of the detection algorithm is estimated around 52.4 ms. This is due to the processing ROIs of 62° FOV, which leads to decrease the processing time up to 50% from 106.1 ms comparing to processing the whole 92° FOV images. In addition, this computational time is estimated for the detection of 800–1200 keypoints. On the one hand, if the number of detected keypoints exceeds 6000, the computational time peaks to a maximum of 100 ms, on the other hand, if it is below 300 keypoints, then the required computation time is reduced to 30 ms.



**Figure 16.** Obstacle detection fail (people) (motion around the UAV)-KPSizeRatio = 1.07 and ObAreaRatio = 1.03.

**Table 2.** Comparison of Frontal Obstacle Detection.

Algorithm	Total	Detected	Fail	Accuracy (%)
SURF + Template matching [24]	107	104	3	97
Relative distance estimation [17]	35	34	1	97.1
Proposed Algorithm	1000	974	26	97.4

#### 5.4. Results (Obstacle Avoidance)

Figures 17 and 18 demonstrate an example of a set of experiments presenting the first scenario. In these experiments, the UAV is flying in a velocity of 2 m/s. All the started from the same *start* point, and during the the flight, an obstacle is situated in the UAV path. Figure 17 illustrates the UAV ability to perform avoidance maneuvers in the Left or Right directions of a total number of 9 experiments.

Similarly, in Figure 18, the success in avoiding hanged obstacles performing vertical maneuvers in the *z* direction by passing above and under the obstacle in a total number of 10 experiments is represented.

Finally, Table 3 shows a comparison of avoidance accuracy between two methods from the bibliography and the proposed algorithm. The accuracy results display that the best performance belongs to the proposed algorithm reaching 93% of accuracy.

**Table 3.** Comparison of Avoidance Accuracy.

Algorithm	Total	Success	Failure	Accuracy (%)
SURF + Template matching [24]	23	20	3	87
Relative Distance Estimation [17]	35	31	4	88.57
Proposed Algorithm	100	93	7	93

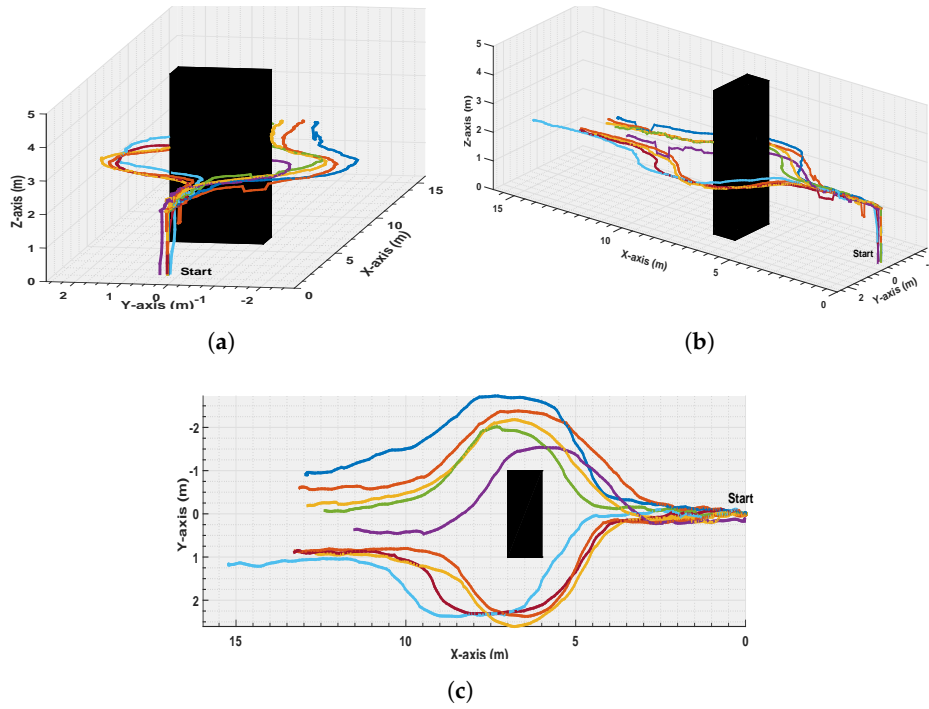


Figure 17. Left-Right Avoidance Maneuver, 9 experiments; (a): Front, (b): 3D, and (c): 2D.

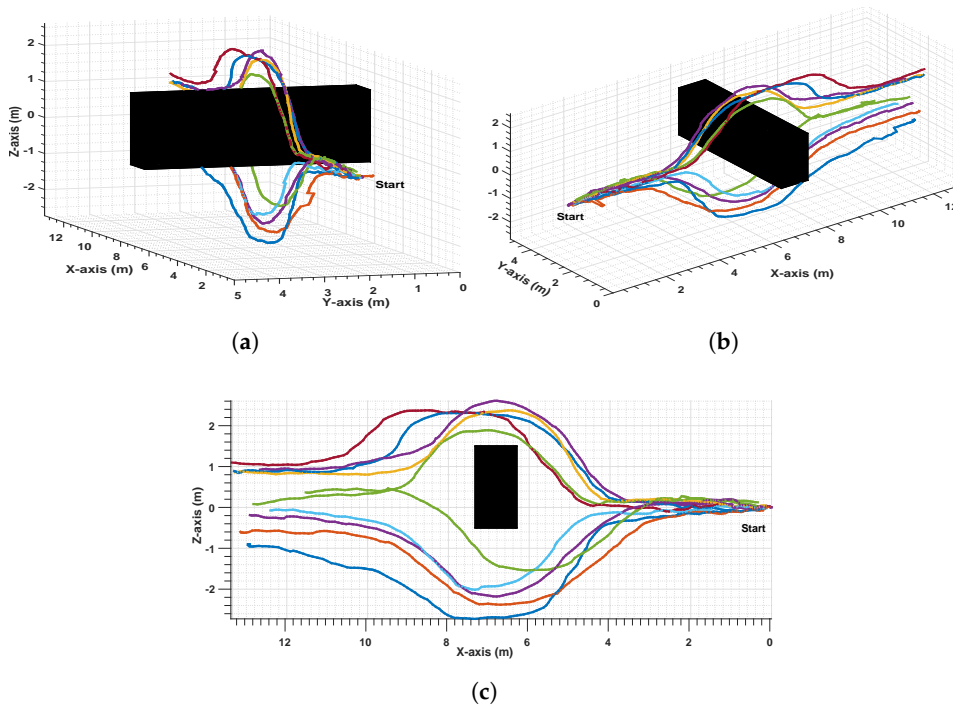


Figure 18. Up-Down Avoidance Maneuver, 10 experiments; (a): Front, (b): 3D, and (c): 2D.

## 6. Conclusions

In this paper, two algorithms have been presented as a framework to cope with cutting-edge UAVs technology. Real-time obstacle detection and avoidance is studied as a complex and essential task for

intelligent aerial vehicles in transportation systems. The proposed algorithms take the advantages of onboard camera to accomplish complex tasks, that is, safe obstacle sensing and detection tasks.

The selected configuration of  $62^\circ$  ensured the capabilities of detecting the border of an object of the size of the actual drone ( $58.4 \times 1.3 \times 54.4$ ) located in the center of the image at distances higher than 15 cm, which allows to avoid obstacles, for higher obstacles or closer distances, the drone proved to be able to stop and perform hover movement, avoiding the collision. Bigger obstacles located at longer distances were avoided due to the use of a high quality camera able to detect obstacle at long distances. In case of faster speeds required, the frame rate calculation and the angle should be adjusted, to allow the drone to do the calculation at proper detection. However, the change of the field of view of the camera, would only be advisable in order to allow further maneuverability in extremely dense scenarios, with short distance detection requirements which are not common in aerial scenarios where UAVs are deployed.

Keypoints used approach is based on the use of SIFT features. The performance obtained proved to be good in both computational time and overall detection performance scheme. The nature of the approach made it possible to adapt it to different set of features beyond the use of SIFT, such as the FAST-BRIEF pair [37] and BRISK [38] which proved to provide better performance in different scenarios. Future works will try to analyze the advantages of adding these sets of features to the presented approach.

The usefulness and advantages of the presented reliable solutions are demonstrated through real results under demanding circumstances, such as, complex artificial and human obstacles. Hence, complex scenarios are evaluated and difficulties are successfully overcome by means of monocular camera processing, where the relative size expansion of the obstacles are estimated and the approaching obstacles are detected from a distance between 90 and 120 cm with 97.4% of total accuracy. The various performed tests proved both, the trustable performance of the algorithms provided and the improvements in comparison to the previous works presented in literature.

The strengths of the presented applications are clearly stated in the paper, where according to the UAV path, the obstacle detection and avoidance algorithms demonstrate high accuracy in the detection process and send maneuver commands to the UAV based on the obstacle and UAV positions. However, the specific drawbacks that should be taken into account are mainly related with the nature of the sensing devices used, that is, the monocular camera has the drawback of the high sensitivity to lighting conditions; such as direct sun light may lead to lack of information.

Future works will focus on two main lines. The tests of the application with different features, as commented. The second line is focused on the development of advanced control system, with higher capacities. These include the use of fuzzy logic, and other advanced AI approaches.

**Acknowledgments:** Research supported by the Spanish Government through the Cicyt project ADAS ROAD-EYE (TRA2013-48314-C3-1-R).

**Author Contributions:** A. Al-Kaff, F. Garcia and D. Martin conceived and designed the experiments; A. Al-Kaff performed the experiments; A. Al-Kaff, F. Garcia and D. Martin analyzed the data; A. de la Escalera and J.M Armingol contributed reagents/materials/analysis tools; A. Al-Kaff, F. Garcia and D. Martin wrote the paper. Authorship must be limited to those who have contributed substantially to the work reported.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Erdos, D.; Erdos, A.; Watkins, S. An experimental UAV system for search and rescue challenge. *IEEE Aerosp. Electron. Syst. Mag.* **2013**, *28*, 32–37.
2. Kamel, B.; Santana, M.C.S.; De Almeida, T.C. Position estimation of autonomous aerial navigation based on Hough transform and Harris corners detection. In Proceedings of the 9th WSEAS International Conference On Circuits, Systems, Electronics, Control & Signal Processing, Athens, Greece, 29–31 December 2010; pp. 148–153.

3. Fraundorfer, F.; Heng, L.; Honegger, D.; Lee, G.H.; Meier, L.; Tanskanen, P.; Pollefeys, M. Vision-based autonomous mapping and exploration using a quadrotor MAV. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Algarve, Portugal, 7–12 October 2012; pp. 4557–4564.
4. Michael, N.; Fink, J.; Kumar, V. Cooperative manipulation and transportation with aerial robots. *Auton. Robots* **2011**, *30*, 73–86.
5. Eschmann, C.; Kuo, C.M.; Kuo, C.H.; Boller, C. Unmanned aircraft systems for remote building inspection and monitoring. In Proceedings of the 6th European Workshop on Structural Health Monitoring, Dresden, Germany, 2–6 July 2012.
6. Choi, S.; Kim, E. Image Acquisition System for Construction Inspection Based on Small Unmanned Aerial Vehicle. In *Advanced Multimedia and Ubiquitous Engineering*; Springer: Heidelberg, Germany, 2015; pp. 273–280.
7. Bills, C.; Chen, J.; Saxena, A. Autonomous MAV flight in indoor environments using single image perspective cues. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011; pp. 5776–5783.
8. Bloesch, M.; Weiss, S.; Scaramuzza, D.; Siegwart, R. Vision based MAV navigation in unknown and unstructured environments. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA), Anchorage, AK, USA, 3–8 May 2010; pp. 21–28.
9. Shim, D.; Chung, H.; Sastry, S. Conflict-free navigation in unknown urban environments. *IEEE Robot. Autom. Mag.* **2006**, *13*, 27–33.
10. Luo, D.; Wang, F.; Wang, B.; Chen, B. Implementation of obstacle avoidance technique for indoor coaxial rotorcraft with Scanning Laser Range Finder. In Proceedings of the 2012 31st Chinese Control Conference (CCC), Hefei, China, 25–27 July 2012; pp. 5135–5140.
11. Shang, E.; An, X.; Li, J.; He, H. A novel setup method of 3D LIDAR for negative obstacle detection in field environment. In Proceedings of the 2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC), Qingdao, China, 8–11 October 2014; pp. 1436–1441.
12. Ariyur, K.; Lommel, P.; Enns, D. Reactive inflight obstacle avoidance via radar feedback. In Proceedings of the American Control Conference, Portland, OR, USA, 8–10 June 2005; Volume 4, pp. 2978–2982.
13. Broggi, A.; Cattani, S.; Patander, M.; Sabbatelli, M.; Zani, P. A full-3D voxel-based dynamic obstacle detection for urban scenario using stereo vision. In Proceedings of the 2013 16th International IEEE Conference on Intelligent Transportation Systems (ITSC), The Hague, Netherlands, 6–9 October 2013; pp. 71–76.
14. Gao, Y.; Ai, X.; Rarity, J.; Dahnoun, N. Obstacle detection with 3D camera using U-V-Disparity. In Proceedings of the 2011 7th International Workshop on Systems, Signal Processing and their Applications (WOSSPA), Tipaza, Algeria, 9–11 May 2011; pp. 239–242.
15. Na, I.; Han, S.H.; Jeong, H. Stereo-based road obstacle detection and tracking. In Proceedings of the 2011 13th International Conference on Advanced Communication Technology (ICACT), PyeongChang, South Korea, 19–22 February 2011; pp. 1181–1184.
16. Li, J.; Li, X.-M. Vision-based navigation and obstacle detection for UAV. In Proceedings of the 2011 International Conference on Electronics, Communications and Control (ICECC), Ningbo, China, 9–11 September 2011; pp. 1771–1774.
17. Saha, S.; Natraj, A.; Waharte, S. A real-time monocular vision-based frontal obstacle detection and avoidance for low cost UAVs in GPS denied environment. In Proceedings of the 2014 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES), Yogyakarta, Indonesia, 13–14 November 2014; pp. 189–195.
18. Majumder, S.; Shankar, R.; Prasad, M. Obstacle size and proximity detection using stereo images for agile aerial robots. In Proceedings of the 2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN), Noida, Delhi-NCR, India, 19–20 February 2015; pp. 437–442.
19. Merrell, P.C.; Lee, D.J.; Beard, R.W. Obstacle avoidance for unmanned air vehicles using optical flow probability distributions. *Mob. Robots XVII* **2004**, *5609*, 13–22.
20. Hrabar, S.; Sukhatme, G.; Corke, P.; Usher, K.; Roberts, J. Combined optic-flow and stereo-based navigation of urban canyons for a UAV. In Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005), Edmonton, AB, Canada, 2–6 August 2005; pp. 3309–3316.

21. Beyeler, A.; Zufferey, J.C.; Floreano, D. 3D Vision-based Navigation for Indoor Microflyers. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation, Rome, Italy, 10–14 April 2007; pp. 1336–1341.
22. Celik, K.; Chung, S.J.; Clausman, M.; Somani, A. Monocular vision SLAM for indoor aerial vehicles. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 10–15 October 2009; pp. 1566–1573.
23. Chavez, A.; Gustafson, D. Vision-based obstacle avoidance using SIFT features. In Proceedings of the 5th International Symposium on Advances in Visual Computing: Part II, ISVC'09, Las Vegas, NV, USA, 30 November–2 December 2009; Springer: Berlin, Heidelberg, 2009; pp. 550–557.
24. Mori, T.; Scherer, S. First results in detecting and avoiding frontal obstacles from a monocular camera for micro unmanned aerial vehicles. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 6–10 May 2013; pp. 1750–1757.
25. Shirai, N.; Yamaguchi, M.K. Asymmetry in the perception of motion-in-depth. *Vis. Res.* **2004**, *44*, 1003–1011.
26. Gibson, J.J. *The Ecological Approach to Visual Perception: Classic Edition*; Psychology Press: Boston, MA, USA, 2014.
27. De Croon, G.; de Weerd, E.; De Wagter, C.; Remes, B. The appearance variation cue for obstacle avoidance. In Proceedings of the 2010 IEEE International Conference on Robotics and Biomimetics (ROBIO), Tianjin, China, 14–18 December 2010; pp. 1606–1611.
28. Green, W.; Oh, P. Optic-Flow-Based Collision Avoidance. *IEEE Robot. Autom. Mag.* **2008**, *15*, 96–103.
29. Lee, J.O.; Lee, K.H.; Park, S.H.; Im, S.G.; Park, J. Obstacle avoidance for small UAVs using monocular vision. *Aircr. Eng. Aerosp. Technol.* **2011**, *83*, 397–406.
30. Ma, Z.; Hu, T.; Shen, L.; Kong, W.; Zhao, B. A detection and relative direction estimation method for UAV in sense-and-avoid. In Proceedings of the 2015 IEEE International Conference on Information and Automation, Gothenburg, Sweden, 24–28 August 2015; pp. 2677–2682.
31. Byrne, J.; Taylor, C.J. Expansion segmentation for visual collision detection and estimation. In Proceedings of the International Conference on Robotics and Automation (ICRA'09), Kobe, Japan, 6–7 August 2009; pp. 875–882.
32. Kim, J.; Do, Y. Moving Obstacle Avoidance of a Mobile Robot Using a Single Camera. *Procedia Eng.* **2012**, *41*, 911–916.
33. Kendoul, F. A Survey of Advances in Guidance, Navigation and Control of Unmanned Rotorcraft Systems. *J. Field Rob.* **2012**, *29*, 315–378.
34. Yu, X.; Zhang, Y. Sense and avoid technologies with applications to unmanned aircraft systems: Review and prospects. *Prog. Aerosp. Sci.* **2015**, *74*, 152–166.
35. Mcfadyen, A.; Mejias, L. A survey of autonomous vision-based See and Avoid for Unmanned Aircraft Systems. *Prog. Aerosp. Sci.* **2016**, *80*, 1–17.
36. Krajnik, T.; Vonasek, V.; Fiser, D.; Faigl, J. AR-drone as a platform for robotic research and education. In *Research and Education in Robotics-EUROBOT*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 172–186.
37. Schmidt, A.; Kraft, M.; Fularz, M.; Domagala, Z. The comparison of point feature detectors and descriptors in the context of robot navigation. *J. Autom. Mob. Robot. Intell. Syst.* **2013**, *7*, 11–20.
38. Jeong, C.Y.; Choi, S. A comparison of keypoint detectors in the context of pedestrian counting. In Proceedings of the 2016 International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, 31 July–26 August 2016; pp. 1179–1181.

