Advanced Robotics

ISSN: 0169-1864 (Print) 1568-5535 (Online) Journal homepage: https://www.tandfonline.com/loi/tadr20

Visual detection and tracking with UAVs, following a mobile object

Diego A. Mercado-Ravell, Pedro Castillo & Rogelio Lozano

To cite this article: Diego A. Mercado-Ravell, Pedro Castillo & Rogelio Lozano (2019) Visual detection and tracking with UAVs, following a mobile object, Advanced Robotics, 33:7-8, 388-402, DOI: 10.1080/01691864.2019.1596834

To link to this article: https://doi.org/10.1080/01691864.2019.1596834

Published online: 27 Mar 2019.

ADVANCED ROBOTICS
2019, VOL. 33, NOS. 7–8, 388–402
https://doi.org/10.1080/01691864.2019.1596834

FULL PAPER

Visual detection and tracking with UAVs, following a mobile object

Diego A. Mercado-Ravell [a], Pedro Castillo[b] and Rogelio Lozano[c]

[a]Cátedra CONACyT, Centro de Investigación en Matemáticas CIMAT, Zacatecas, Mexico; [b]Sorbonne universitès, Université de technologie de Compiègne, CNRS, Heudiasyc UMR 7253, CS 60 319, Compiègne cedex, France; [c]Laboratoire Franco Mexicain d'Informatique et Automatique, Unite Mixte Internationale LAFMIA UMI 3175 CINVESTAV-CNRS, Mexico City, Mexico

ABSTRACT

Conception and development of an Unmanned Aerial Vehicle (UAV) capable of detecting, tracking and following a moving object with unknown dynamics is presented in this work, considering a human face as a case of study. Object detection is accomplished by a Haar cascade classifier. Once an object is detected, it is tracked with the help of a Kalman Filter (KF), and an estimation of the relative position with respect to the target is obtained. A linear controller is used to validate the proposed vision scheme and for regulating the aerial robot's position in order to keep a constant distance with respect to the mobile target, employing as well the extra available information from the embedded sensors. The proposed system was extensively tested in real-time experiments, through different conditions, using a commercial quadcopter connected via wireless to a ground station running under the Robot Operative System (ROS). The proposed overall strategy shows a good performance even under disadvantageous conditions as outdoor flight, being robust against illumination changes, image noise and the presence of other people in the scene.

ARTICLE HISTORY
Received 9 October 2018
Revised 14 January 2019
Accepted 6 March 2019

KEYWORDS
Aerial vehicles; object detection and tracking; vision-based control; human–robot interaction

1. Introduction

The astonishing fast developments on the fields of computer science, electronics, mechanics, and more particularly on robotics, over the last recent years, allow us to believe that the science fiction vision of a world surrounded by all kind of robots interacting with humans in all sort of tasks is no longer a fairy dream, but a real possibility in the near future.

Until now, this high risk of injure humans has prevented the use of robots for several applications, constraining them to industrial tasks where the environment is controlled and security can be maximized, or in dangerous scenarios where humans can hardly operate. Otherwise, small size robots that are harmless for humans have to be employed, mainly for entertainment and research. With respect to aerial robots, better known as Unmanned Aerial Vehicles (UAVs), some interesting applications have popped out from new emerged technologies [1], for example, by using the *Myo* bracelet produced by *Thalmic Labs* we are able to control UAVs just with the movements of the arm [2]. This is a devise equipped with Electromyography (EMG) sensors along with a 9-axis Inertial Measurement Unit (IMU) to observe activity from your muscles to detect hand gestures and orientation. Another interesting tool is the *Microsoft Kinect* sensor. It has been conceived to recognize gestures and body postures to friendly control the navigation of UAVs through a Natural User Interface (NUI) [3, 4]. Additionally, haptic devices offer a different way of interaction with human operators by providing force feedback, allowing to improve the piloting experience and assist the user in complicated tasks such as simultaneously controlling multiple UAVs [5]. A study on the integration of speech, touch and gesture for the control of UAVs from a ground control station is also presented in [6].

The use of computer vision for human–drones interaction appears as a powerful alternative, and some examples can be found in the literature, as in [7] where a study on how to naturally interact with drones using hand gestures is presented. More examples can be found in [8], where a survey on computer vision for UAVs is presented. Target tracking using monocular vision for aerial vehicles has also been studied as in [9] where a nonlinear adaptive observer and a guidance law for object tracking are proposed and validated in simulations. Another key related problem is the detection of people in the imagery provided by drones. This is of particular interest in search and rescue missions using UAVs. For instance, Blondel et al. [10] propose a detection algorithm

CONTACT Diego A. Mercado-Ravell diego.mercado@cimat.mx; die.ravell88@gmail.com

using information from visual and infrared spectrum cameras.

In the present work, it is intended to offer a solution to the problem of an UAV capable of following a moving target, in this case a human face, using a camera as main sensor. As a first approximation to the problem, the simplest solution was preferred over the sophisticated one, as long as it was effective in accomplishing its objective and always keeping in mind the users security as the design priority. To do so, a wide variety of problems, all of great interest in the robotics field, were successfully solved and the resulting system can be safely used. They go from perception using monocular computer vision, to automatic control and state estimation, passing through signal filtering.

Detection and tracking of moving objects is envisioned for several interesting applications, including following a suspect or a fugitive in a persecution, or video recording people in hazardous situations, like while practicing extreme sports. This idea was already conceived by the commercial quadcopter *LILY* [11], but in that case the system depends on an external tracking device to localize the human user, while our proposed algorithm depends only on an already integrated frontal camera. Moreover, following a person can be used to interact with the human user, for example, trying to hide and run away from the drone, or even to play with it some popular games like *tag* or the *hot-potato*. Furthermore, the classifier can be trained to detect any other mostly rigid object to be followed by the drone.

Recent efforts towards people following using computer vision can be found in the literature, as is the early case of the joggobot [12], an UAV designed to assist a person while jogging, improving the jogging experience. This was accomplished by using special markers on the jogger's T-shirts. Later on, in [13] the person following problem is tackled using a depth camera, stabilizing the depth image by warping it to a virtual-static camera and feeding the stabilized video to the OpenNI tracker, but a second camera looking upwards is needed to detect special markers and determine the absolute position of the UAV. Simple commands are provided through hand gestures as well. Danelljan et al. [14] propose a detection algorithm combining color and shape information. Detected objects are then used to initialize an Adaptive Color Tracker (ACT) to keep track of multiple objects. The tracking results are verified using the detector and filtered using a Probability Hypothesis Density (PHD) filter. Distance to the target is estimated assuming a horizontal ground plane and a fixed person height. Position control is performed by means of a Proportional controller. In [15], a general object following strategy using a commercial AR Drone 2.0 and OpenTLD tracker is presented. An Image Based Visual Servoing (IBVS) is then applied to follow the target. This approach was tested to follow people using the logos on the target cloths. The advantage relies on the capability to follow different objects, but a human operator must initialize the object to be tracked, and this architecture is not able to estimate the depth to the target. Meanwhile, in [16] a study on multiple object trackers is presented to detect and follow a person using the AR Drone 2.0. In particular, they extended the Discriminative Scale Space Tracker (DSST) and the Kernelized Correlation Filter (KFC) based tracker in order to detect target lost and redetect targets. They employ the same IBVS by [15]. More recently, in [17] an end-to-end human–robot interaction with an uninstrumented human is presented. First, the drone, a commercial Parrot Bebop, detects from far away potential humans waving hands to interact with. Then, it approaches the selected target and obeys simple hand-gesture commands such as taking pictures and landing. Hand gestures are detected using optical flow. During the approaching phase, the same long-term visual tracker from [16] is used. Depth estimation is performed similar to [14]. IBVS is used along with a predictive controller to position the drone, but the lateral movements are not controlled. Finally, Yao et al. [18] present a visual-based human following UAV, but using a blimp instead of a multirotor.

In contrast to other works, as depicted in Table 1, our approach does not require special markers or logos on the human cloths [12, 13, 15]. Also, a simple depth estimation is provided removing the horizontal ground plane assumption [14, 17]. Finally, unlike [15–17] where IBVS is used, we propose a relative position controller after estimation of the relative position between the drone and the human target in the world frame, where the drone is commanded as an omnidirectional robot, controlling lateral movements instead of yaw. The proposed architecture has proved to be effective in real-time experiments, while using computer efficient algorithms well suited for embedded applications on UAVs.

The contribution of this work can be summarized in the following points:

- Several powerful tools and techniques are merged together to offer a full-working solution for a mobile-target tracking drone.
- A complete computer vision algorithm is developed to detect and track a face in 3D, despite the presence of false positives or other faces on the image. This includes estimating the distance from the image plane to the face, using the size of the detected face on the image.

Table 1. Qualitative comparison with related works.

Property/work	Target type	Special markers	Tracking method	Reach	Depth estimation	Prototype
[12]	T-shirt	✓	Unknown	Following in straight lines	X	AR.Drone
[13]	Full-body	✓	OpenNI	Following+ hand gestures	Depth camera	AscTec Pelican
[14]	People	X	ACT + PHD	Following	Horizontal plane assumption	LinkQuad
[15]	Manually selected	T-shirt logo	OpenTLD	Following	X	AR.Drone
[16]	Person	X	DSST & KFC	Following (IBVS [15])	X	AR.Drone
[17]	People	X	[16]	Hand gestures commands	[14]	Bebop
This	Face	X	Haar + KF	Following	✓	AR.Drone

- A real-time relative position control algorithm is proposed for an UAV following a moving object with unknown dynamics, including the observation of the missing state.

The outline of this paper is the following: the general operation of the overall system is described in Section 2. In Section 3, the computer vision algorithm for face detection and tracking, as well as the necessary transformation from the image to the real world are presented. The relative position control is introduced in Section 4, along with some techniques for estimating the altitude speed, needed for the state feedback. Section 5 contains the experimental results of the proposed algorithms, and last but not least, Section 6 concludes with the perspectives and conclusions of this work.

2. Overall system description

A commercial quadcopter, the *AR.Drone* Parrot, shown in Figure 1 is selected for this work. This quadcopter offers a good solution to work close to people without major danger, thanks to its small size and weight of 53×52 cm and 0.42 kg, respectively, and overall because it is protected with a soft hull which also increases its robustness against crashes. It is equipped with three-axis gyroscopes and accelerometers, an ultrasound altimeter, an air pressure sensor and a magnetic compass. Furthermore, it contains two video cameras, one looking downwards and one forward, the former, with a resolution of 320×240 pixels at a rate of 60 fps, is used to estimate the horizontal velocities using optic flow, while the latter, with a resolution of 1280×720 at 30 fps, is normally intended for real-time video streaming and image

Figure 1. Detection and tracking of a mobile object using an UAV, application to follow a human user. The UAV is able to autonomously follow the target of interest, keeping a constant distance with it.

recording, but in this case it is further used for the vision algorithm to detect a moving object of interest.

The drawback of this kind of commercial UAVs resides in its lack of flexibility to be modified, since both software and hardware come in closed architectures and it is not straightforward to customize them. This is solved by designing position control laws that employ reference roll and pitch angles along with altitude and yaw speeds as virtual control inputs (ϕ_d, θ_d, $\dot{z}_d$, $\dot{\psi}_d$). Such virtual control signals are then fed to the inner autopilot as desired references. In this case, a linear PD controller is proposed. All sensor measurements are sent to a ground station at a frequency of 200 Hz. The image processing and the control algorithms are computed in real time on ROS at a rate of 30 Hz.

Three main nodes running in ROS on a ground station computer are employed to achieve object detection and tracking with an UAV. Communication with the drone is performed by means of the *AR.Drone* driver node, already available as open source, which allows to recover information from the embedded sensors on the drone, along with the video streams from both cameras, and to send control inputs. A second node is in charge of the image processing from the frontal camera to detect and track the target, providing the target's position and size. Object detection is accomplished using a Haar feature-based classifier in cascade. Once a target is detected, a Kalman Filter (KF) is used to track it along time, adding robustness against the presence of other target-like objects and false-positive detection. A third node was implemented for relative position control. First, the estimated relative position of the tracked object with respect to the drone is transformed from the image space to the real world, by a suitable coordinates change, where the image depth is estimated using the a-priori knowledge of the object size. A PD controller is used to keep a constant distance between the UAV and the target, where the horizontal velocities ($\dot{x}$, $\dot{y}$) are obtained from the embedded optical flow sensor, and the altitude speed ($\dot{z}$) is estimated by means of a Luenberger observer and the ultrasound altitude sensor. Both the visual object detection and tracking node and the relative position control node were developed for this work.

Moreover, a Graphical User Interface (GUI) node is available for online parameter tuning, switching between operation modes and real-time monitoring. Also, in case any problem arises, and taking the user security as the design priority, manual recovery is possible at any time using a *sixaxis* wireless joystick. The overall system architecture is presented in Figure 2.

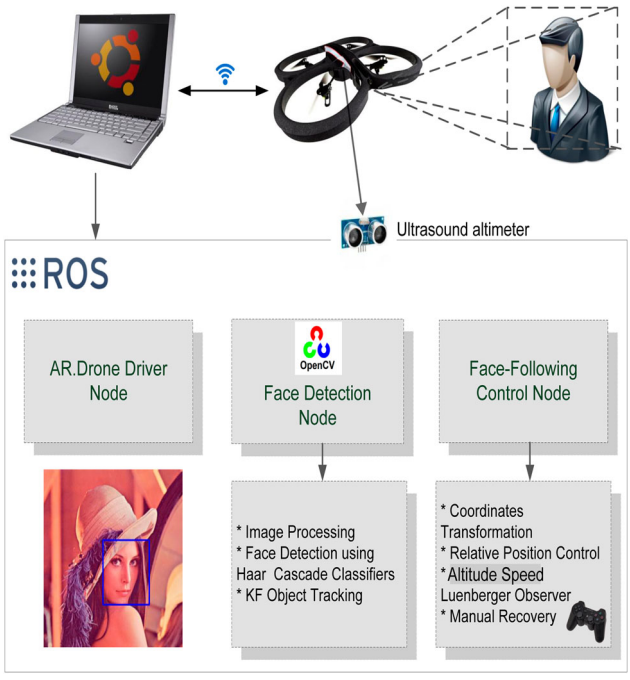

Figure 2. Overall system description. The drone communicates wirelessly with a ground station composed by a computer running ROS. Three main ROS nodes are executed in the ground station: the drone's driver, the face detection using OpenCV and the control node. The drones driver provides information from the embedded sensors, such as the ultrasound altimeter and the optic flow sensor, along with video stream from two cameras. Object detection is accomplished using a Haar cascade classifier on OpenCV, once a target is detected, scale is estimated based on the target size, and a KF is used for tracking the objective. Relative position control is performed through a PD controller, using the extra information from the sensors, where a Luenberger observer is implemented to recover the altitude speed. For safety reasons, manual recovery is available with a joystick.

3. Visual object detection and tracking

3.1. Computer vision algorithm for object detection

Object detection and tracking is performed using computer vision, with the help of the OpenCV libraries on ROS, using the *cv_bridge* node. In brief, the UAV's frontal camera streams a video at 30 Hz in BGR format, which is transformed to grayscale. Then, the image is smoothed by means of a blur filter with a kernel of 5×5, to eliminate white noise on the image, helping to diminish the number of false positive detections due to high frequency noise. Afterwards, a histogram equalization is applied to improve the contrast on the image. Finally, the pretrained object detector is used, and the selected target is tracked with the help of a KF. Outliers are disregarded before the KF. A flow chart describing the complete visual object detection and tracking algorithm is depicted in Figure 3.

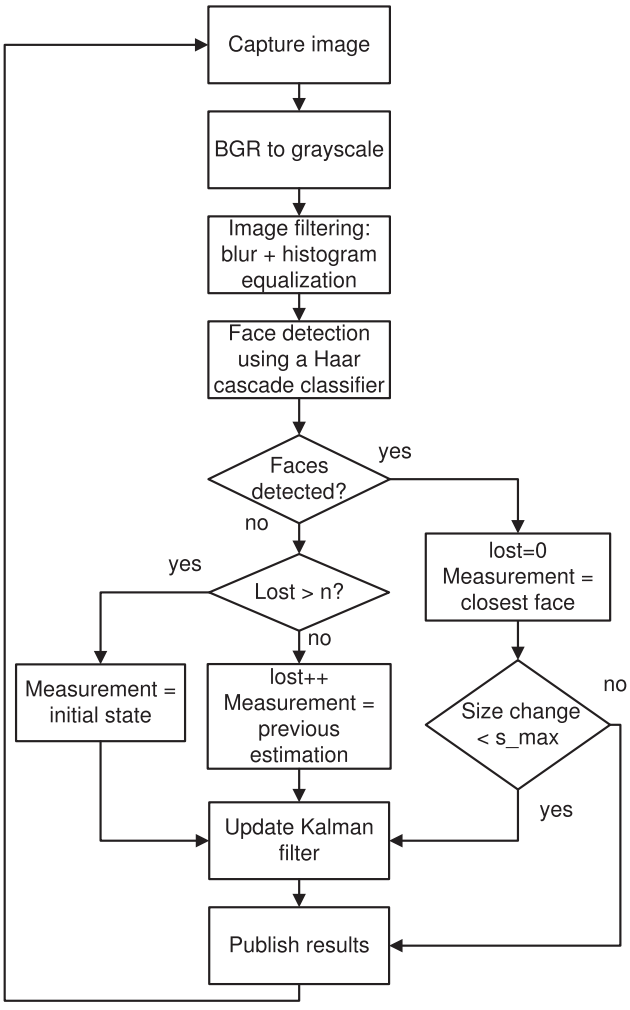

Figure 3. Flowchart of the image processing algorithm. Video stream is provided by the frontal camera on the UAV and converted to grayscale. After a blur filter and histogram equalization, the Haar cascade classifier is used to detect the objective in the scene. A KF filter is used to track the moving object along time, rejecting other similar objects in the image as well as false positives.

Object detection is accomplished through the Haar classifier on OpenCV [19]. It consists in a Machine Learning technique first developed in [20] which uses Haar-like features in cascade through different levels of the image to determine whether or not a pre-specified rigid object, for which it was trained a priori, is present on the image. The Haar classifier is a supervised classifier that uses a form of AdaBoost organized as a rejection cascade and designed to have high detection rate at the cost of low rejection rate, producing many false positives. One of the main advantages of this method is the computational speed achieved in real-time detection, once the classifier was trained off-line for the desired object, in this case a face. It is important to notice that this method can be trained for almost any mostly rigid object with distinguishing views.

In this case, the classifier is trained to detect human faces. The idea is to create an application to interact with the user by keeping a constant distance. Only frontal faces are considered by now. In order to detect different face's poses, a classifier for each pose can be trained and run sequentially. For future developments, it is of particular interest to detect faces rotated in yaw with respect to the drone, to achieve relative yaw control such that the UAV would be able to autonomously locate itself in front of the target.

3.2. Kalman filter for object tracking

One of the main drawbacks of the object detection algorithm is its low rejection rate, resulting in a high number of false positive detections. To overcome this issue and in order to add robustness to the algorithm against missed detections and the presence of other objects similar to the target one on the scene, one solution is to track along time the detected object. To do so, let us consider the use of the Kalman Filter (KF), a powerful technique for optimal state estimation and filtering of linear systems perturbed with Gaussian noise [21].

In this case, the discrete-time version of the KF [22, 23] is applied to the kinematic model of the detected target (see Figure 4), i.e.

$$\begin{bmatrix} \chi_k \\ V_k \end{bmatrix} = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \chi_{k-1} \\ V_{k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ a_{k-1} \end{bmatrix}, \qquad (1)$$

where $\chi = [\bar{x}\ \bar{y}\ \bar{d}]^{\mathrm{T}}$ represents the position of the center of the circle enclosing the object of interest, directly on the image and its diameter d, while $V = [\dot{\bar{x}}\ \dot{\bar{y}}\ \dot{\bar{d}}]^{\mathrm{T}}$ is its time derivative. k is the discrete time index and T_s defines the sampling period. Finally, the process noise $a \in \Re^3$ is used as a tuning parameter, analogous to the acceleration, which determines how fast the variables can move.

The measurement in the KF is updated according to four different cases, depending on the state of the vision algorithm:

- At least one object is detected.
- No objects are detected.
  - Iterations since last detection $\leq n$.
  - Iterations since last detection $> n$.

for certain constant $n \in \Re^+$ denoting the maximum number of iterations without detection before the object is considered lost. For the first case, if more than one object is detected on the same scene, either due to a

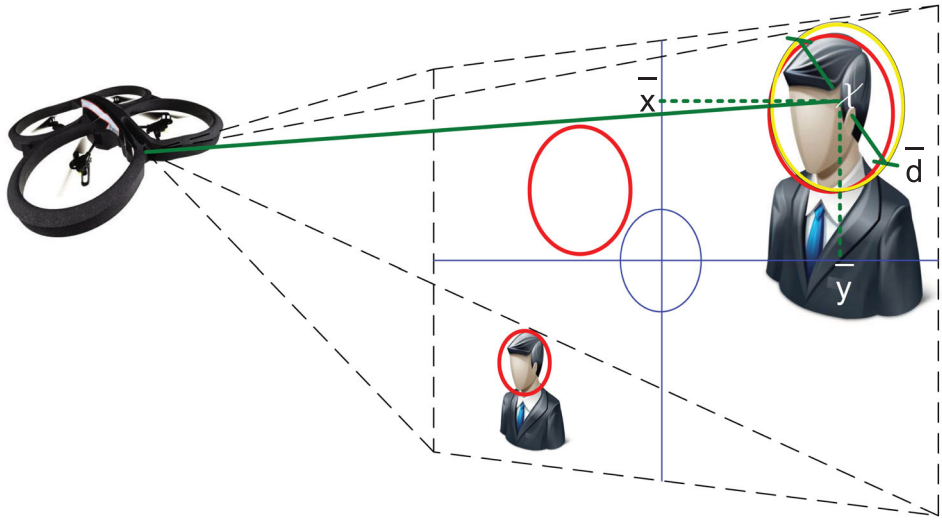

Figure 4. Visual object detection and tracking with a flying quadcopter. Detected objects are represented by dashed red circles, including false detections. The yellow solid circle represents the objective tracked by the KF. The algorithm is robust against false positive detections and the presence of other similar objects in the scene.

false positive detection or to the presence of other similar objects on the image, the detection closest to the previous estimation is chosen. If no object is detected for a few iterations, a missed detection is assumed and the measurement is updated with the last estimation, allowing to continue with the same motion for a while, and avoiding sharp displacements in the closed loop system. However, if the target is not detected for several iterations, the object of interest is assumed lost, consequently, the measurement vector is updated with its initialization value χ_{init}, and the quadcopter would hold its position. Therefore, the update equation for the measurement vector ζ takes the form

$$\zeta_k = \begin{cases} \chi_{\min_k} & N_{\text{faces}} > 0 \\ \chi_{k-1} & (N_{\text{faces}} = 0)\&(\iota_{\text{lost}} \leq n) \\ \chi_{\text{init}} & (N_{\text{faces}} = 0)\&(\iota_{\text{lost}} > n) \end{cases} \tag{2}$$

with

$$\chi_{\min} = \min_{\chi_{\delta_i} \in \Omega} \sqrt{(\bar{x} - x_{\delta_i})^2 + (\bar{y} - y_{\delta_i})^2}, \tag{3}$$

where N_{faces} is the number of objects detected at the present frame, ι_{lost} defines the number of iterations since the last valid detection. Ω represents the set of all the objects detected on the image, described by $\chi_\delta = [x_\delta \; y_\delta \; d_\delta]^{\text{T}}$. $\chi_{\min}$ denotes the closest detection to the previous estimation.

Figure 4 illustrates a possible scenario, where the UAV captures a scene with multiple object detections. Positive detections obtained by the classifier are identified by the dotted red circles, including false positive detections where there are no objects of interest. Then, the KF is employed to track the chosen target along time, and its result is displayed by a unique solid yellow circle with center at $(\bar{x}, \bar{y})$ and diameter $\bar{d}$. Only the closest detection to the previous estimation is used to update the measurement vector of the KF, and the rest are discarded.

3.3. Relative position estimation: from image to real world

Let us consider for simplicity the idealized pinhole camera model to describe the relationship between coordinates in the real world $p_w = (x_w, y_w, z_w)$ to their projection on the image $p_{im} = (x_{im}, y_{im})$, see Figure 5, according to the following expressions [24]:

$$x_{im} = f_x \left[\frac{x_w}{y_w}\right] + c_x; \quad y_{im} = f_y \left[\frac{z_w}{y_w}\right] + c_y; \tag{4}$$

where the constant parameters for the focal lengths $f_x = Fs_x$, $f_y = Fs_y$ are actually the product of the physical focal length F and the number of pixels per meter s_x, s_y, along each image axis. These constants, together with the principal point position c_x, c_y, determine the intrinsic parameter of the camera and are known from calibration. Note that, in contrast to the standard notation used in computer vision, here z stands for the altitude and y for the depth between the camera and the point, this is done just for consistency with the rest of the coordinate frames.

From the previous equation, it is straightforward to recover from the image projection, the real-world position of a point in the x_w and z_w axes, normalized by the depth y_w. However, this depth y_w remains unknown and cannot normally be obtained from the information given

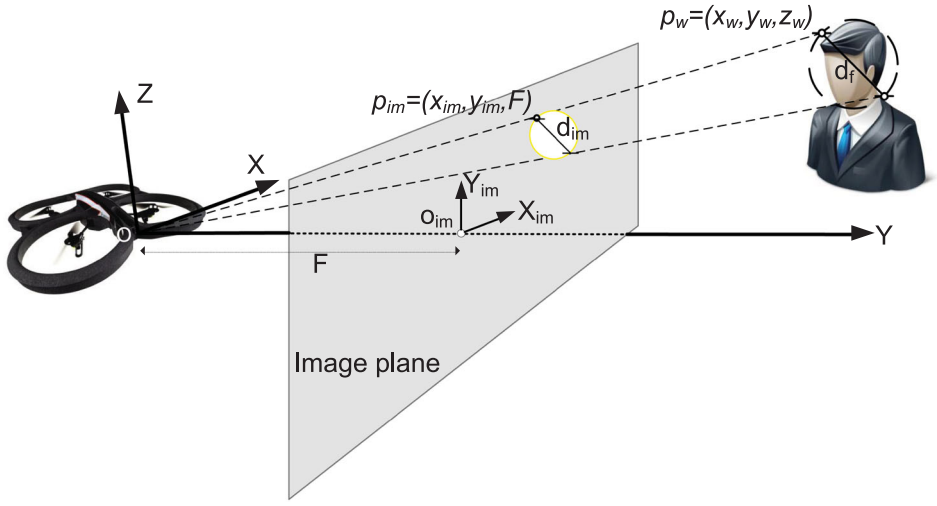

Figure 5. From real world to image coordinates transformation. A pinhole camera model is used to map the image projection to its real-world source, without scale. The scale is estimated using the a-priori knowledge of the average size of the object.

by a single camera. The use of several cameras for stereo-vision is precluded due to the inflexibility of the selected hardware. Another option is to obtain extra information from the already available sensors, such as altitude and inertial measurements, to fusion it with the vision data to estimate this depth, similar to the work developed in [25]. Nevertheless, in our considered scenario, it can be estimated from the a-priori knowledge of the average size of the object of interest. Even though every person's face is different in size and shape, let us assume the face of an adult having an average size whose enclosing circle has a diameter of $d_f \approx 0.241$ m. Also, consider a diameter d of the enclosing circle for the face projected in the image, see Figure 4. Then, it is easy to show that

$$y_w \approx \frac{Fd_f}{d}. \tag{5}$$

Finally, substituting in (4) the position of the object in the physical world can be obtained with

$$x_w = \frac{(x_{im} - c_x)s_x d_f}{d}; \tag{6}$$

$$z_w = \frac{(y_{im} - c_y)s_y d_f}{d}. \tag{7}$$

4. Relative position control

Provided that a full state feedback is available, the mission is to control the drone to autonomously keep a constant distance with respect to a moving target with unknown dynamics. This is equivalent to a three-dimensional position control with a time varying reference. A hierarchical control is proposed to deal with this problem, where a PD position controller with gravity compensation is added in cascade with the inner orientation control loop, available with the autopilot on the selected experimental platform. This strategy is compatible with most commercial autopilots and allows for easy implementation with other drones.

Let us consider a simplified version of the well-known dynamic model of a quadcopter [26]:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} \approx \frac{T}{m} \begin{bmatrix} s\psi s\phi + c\psi s\theta c\phi \\ -c\psi s\phi + s\psi s\theta c\phi \\ c\theta c\phi \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}, \tag{8}$$

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} \approx \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix}, \tag{9}$$

where x, y, z are the position of the quadcopter with respect to an inertial frame, $T \in \Re^+$ defines the total thrust produced by the motors, m and g represent the mass and gravity constant, respectively. ϕ, θ and ψ stand for the Euler angles roll, pitch and yaw, and τ_ϕ, τ_θ and τ_ψ describe the control torques produced by the differential velocities of the rotors. The short notation $s\alpha = \sin(\alpha)$ and $c\alpha = \cos(\alpha)$ is used. Remember that the *AR.Drone* includes an internal autopilot to deal with the attitude (ϕ, θ and ψ) and the altitude (z) controllers, to follow some desired reference. Therefore, our control input vector is $u = [\phi_d\ \theta_d\ \dot{z}_d\ \dot{\psi}_d]^{\mathrm{T}}$, i.e. the desired angles roll and pitch, and desired altitude and yaw rates. Given that the rotational dynamics are much faster than the translational one [27], a time scale separation of the translational and rotational dynamics permits to use the roll and pitch desired references (ϕ_d and θ_d) as virtual control inputs for the unactuated states (x and y).

4.1. Control law

Since the available autopilot already provides a suitable attitude controller, the study herein will focus on the horizontal position dynamics x and y. Assuming small variations in the roll and pitch angles, i.e. only non-aggressive maneuvers are considered, in compliance with the security demands while working close to humans, a linearization around the equilibrium point $\phi_d \approx \theta_d \approx 0$ yields:

$$m\begin{bmatrix}\ddot{x}\\ \ddot{y}\end{bmatrix} \approx T\begin{bmatrix}s\psi & c\psi\\ -c\psi & s\psi\end{bmatrix}\begin{bmatrix}\phi_d\\ \theta_d\end{bmatrix}. \tag{10}$$

Note the introduction of the desired angles instead of the real ones, this is possible since $\phi_d \approx \phi$ and $\theta_d \approx \theta$ due to the action of the autopilot attitude control loop. Denote now

$$\begin{bmatrix}\hat{\phi}\\ \hat{\theta}\end{bmatrix} = \begin{bmatrix}s\psi & c\psi\\ -c\psi & s\psi\end{bmatrix}\begin{bmatrix}\phi_d\\ \theta_d\end{bmatrix} \tag{11}$$

then

$$\begin{bmatrix}\ddot{x}\\ \ddot{y}\end{bmatrix} = \frac{T}{m}\begin{bmatrix}\hat{\phi}\\ \hat{\theta}\end{bmatrix}, \tag{12}$$

where $\hat{\phi}$ and $\hat{\theta}$ are used as the new virtual control inputs for the position, and are chosen such that they follow certain references x_d, y_d. Thus a PD control can be proposed as follows:

$$\begin{bmatrix}\hat{\phi}\\ \hat{\theta}\end{bmatrix} = \frac{m}{T}\begin{bmatrix}-k_{px}(x-x_d) - k_{dx}(\dot{x}-\dot{x}_d)\\ -k_{py}(y-y_d) - k_{dy}(\dot{y}-\dot{y}_d)\end{bmatrix} \tag{13}$$

with the control gains k_{px}, k_{py}, k_{dx}, $k_{dy} \in \Re^+$. Transforming to the original coordinates

$$\begin{bmatrix}\phi\\ \theta\end{bmatrix} = \frac{m}{T}\begin{bmatrix}s\psi & -c\psi\\ c\psi & s\psi\end{bmatrix}\begin{bmatrix}-k_{px}(x-x_d) - k_{dx}(\dot{x}-\dot{x}_d)\\ -k_{py}(y-y_d) - k_{dy}(\dot{y}-\dot{y}_d)\end{bmatrix}. \tag{14}$$

In this work, the interest relies on the vehicle position relative to the mobile target, rather than global localization. Also, it is supposed that the objective pose is aligned with the drone's yaw (the objective is in view from the frontal camera on the vehicle), hence the relative yaw rotation always remains small. To remove this constrain, other classifiers can be trained for different yaw rotations, then a relative yaw control can be implemented to add robustness to rotational movements and increase the applications of the system.

As for the yaw ψ and altitude z controllers, the following control laws are proposed:

$$\dot{\psi}_d = -k_{p\psi}(\psi - \psi_d) - k_{d\psi}\dot{\psi} \tag{15}$$

$$\dot{z}_d = -k_{pz}(z - z_d) - k_{dz}\dot{z} \tag{16}$$

for some desired ψ_d, z_d. Also, $k_{p\psi}$, k_{pz}, $k_{d\psi}$ and $k_{dz} \in \Re^+$ are suitable control gains. For the following problem of a mobile object, the desired position $[x_d\ y_d\ z_d]^{\mathrm{T}}$ is obtained from the relative position between the drone and the object (Equation 7), which is estimated by the vision algorithm presented in Section 3, i.e.

$$\begin{bmatrix}x_d\\ y_d\end{bmatrix} = \begin{bmatrix}x_w\\ y_w + y_{off}\end{bmatrix}, \tag{17}$$

where y_{off} is a predefined safe distance to the objective.

4.2. Altitude velocity estimation

It is important to notice that the previous feedback control strategy relies on a good measurement of the system states and their derivatives. Relative position is used directly from the vision algorithm. However, velocities from the vision algorithm result to be imprecise and noisy, and not to trust for control feedback. This is not a problem for the x and y coordinates since the used platform is equipped with a down looking camera from which the optic flow is calculated to estimate the horizontal velocities $\dot{x}$ and $\dot{y}$. Nevertheless, problems arise because no measurement is available for the altitude velocity $\dot{z}$.

To overcome this issue, two estimation techniques were implemented and tested for the altitude velocity, based on the altitude measurements z_m from the ultrasound sensor integrated on the bottom of the helicopter. The first method consists on the classical Euler derivative of the measurement z_m, i.e.

$$\dot{z}_m(k) = \frac{z_m(k) - z_m(k-\Delta t)}{\Delta t}, \tag{18}$$

where k and Δt stand for the discrete time variable and its increment. Although it offers a simple solution, this derivative is known to amplify the noise from the measurements and cannot be directly used for control feedback. For this reason, it was decided to implement a fourth-order low-pass filter type Chebyshev I, at a cutoff frequency of 8 Hz. This is a kind of recursive filter, and as such, it offers a fast response [28]. The following equation presents the filter algorithm with input $\dot{z}_m$ and output $\dot{\tilde{z}}_m$

$$\dot{\tilde{z}}_m(k) = \sum_{i=0}^{4} a_i\dot{z}_m(k-i\Delta t) + b_i\dot{\tilde{z}}_m(k-i\Delta t), \tag{19}$$

where the constant coefficients a_i and b_i are obtained with the help of *MATLAB/fdatool*.

The result is a smooth and acceptable estimation of the speed, except for a little delay of about 4 steps produced by the filter. This response is good enough for several applications where dynamics are slow or sampling frequency is large enough to neglect this delay. However, for UAV control feedback this can cause instability or poor performance in the closed-loop system.

The second explored method consists of a Luenberger state observer [29]. A state observer provides estimates of the internal states of the system from the input and output measurements. Let us consider the discrete-time kinematic model of the altitude at instant k

$$\begin{bmatrix} z_k \\ \dot{z}_k \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z_{k-1} \\ \dot{z}_{k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ u_z \end{bmatrix}, \tag{20}$$

$$\gamma = [1\ 0] \begin{bmatrix} z_{k-1} \\ \dot{z}_{k-1} \end{bmatrix}, \tag{21}$$

where γ is the output and the input is $u_z = \ddot{z}_k \Delta t$. Then, a Luenberger observer is proposed, according to the following equations:

$$\begin{bmatrix} \hat{z}_k \\ \hat{\dot{z}}_k \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{z}_{k-1} \\ \hat{\dot{z}}_{k-1} \end{bmatrix} + \begin{bmatrix} L_1 \\ L_2 \end{bmatrix} [\gamma - \hat{\gamma}] + \begin{bmatrix} 0 \\ u_z \end{bmatrix}, \tag{22}$$

$$\hat{\gamma} = [1\ 0] \begin{bmatrix} \hat{z}_{k-1} \\ \hat{\dot{z}}_{k-1} \end{bmatrix}. \tag{23}$$

Note the use of a hat on the variables to indicate that they are state estimates rather than the real ones. In order to guarantee that the estimated states $[\hat{z}\ \hat{\dot{z}}]^{\mathrm{T}}$ converge asymptotically to the real ones, the observer gains L_1, L_2 are chosen such that the matrix

$$\begin{bmatrix} 1 - L_1 & \Delta t \\ -L_2 & 1 \end{bmatrix}$$

has all its eigenvalues inside the unit circle.

Figure 6 allows us to compare both estimation techniques, where it is clear that the two of them obey in general the same behavior, however, the estimation given by the Euler derivative combined with the Chebyshev filter (dashed blue line) attenuates a little bit the signal amplitude and introduces a small undesired delay. This is corrected by using only a Luenberger observer instead (solid red line), which proves to be an excellent option for this case.

5. Real-time experimental results

The performance of the full strategy was vastly studied under different conditions, in indoor and outdoor trials, as can be observed in the video at https://youtu.be/xbpMx4o6gY0. There, we can appreciate four stages. First, indoor following is demonstrated moving in the three axis, where the user even covers his face during his motion and uncovers it in a different position, proving the fast response of the system. We consider that covering the face during the motion is a harsher condition than fast user motion. The second stage, also indoors, proves the robustness of the system in the presence of other target-like objects, in this case, other faces. The drone was able to keep constant distance with respect to the objective even in the presence of other person trying to perturb the system. Third scenario accounts for outdoors in the presence of other faces. Note that there is no control over certain outdoors conditions, such as illumination and wind. In this part of the video, we can observe several false positive detections, signaled as red circles, in addition to the target detection and other face in the scene. The use of a KF for tracking allows us to keep the right objective along time. Finally, outdoors operation is tested one more time, in the presence of strong wind gusts that considerably perturb the UAV. In all scenarios, the system was able to successfully accomplish its objective in spite of the imposed harsh conditions.

To quantitatively evaluate the system performance, other experiments were conducted indoors, similar to the first stage in the video. Some results are presented in Figures 7–12. The mission consisted in detecting a human face and keeping a constant distance of 2 m in front of it, while the target moves in a rectangular position, at human walking speed, as can be appreciated in Figure 7. Once the second trajectory is completed, the interacting user descends and rises a few times in order to test the altitude response. An *OptiTrack* motion capture system, composed of 12 infrared cameras, was used only as a ground truth, providing millimeter precision.

Two results are of particular interest in this study, on the one hand, the computer vision detection plus the KF tracking performance to estimate the relative position of the mobile object with respect to the camera can be studied from Figures 8–10. Figure 8 illustrates the effect of the KF for tracking the moving target, mainly to add robustness to false positives and the presence of other target-like objects in the scene (other humans in this case). The result is a smoother estimation, where the KF is able to handle wrong detections (see for example the peak around second 13 in the x coordinate). Please note that this figure is the only one that does not correspond to the same experiment, but to a separate one where harder conditions were imposed in order to highlight the action of the KF, by increasing the number of false positive detections.

The relative position estimation is illustrated in Figure 9, where the visual detection and tracking

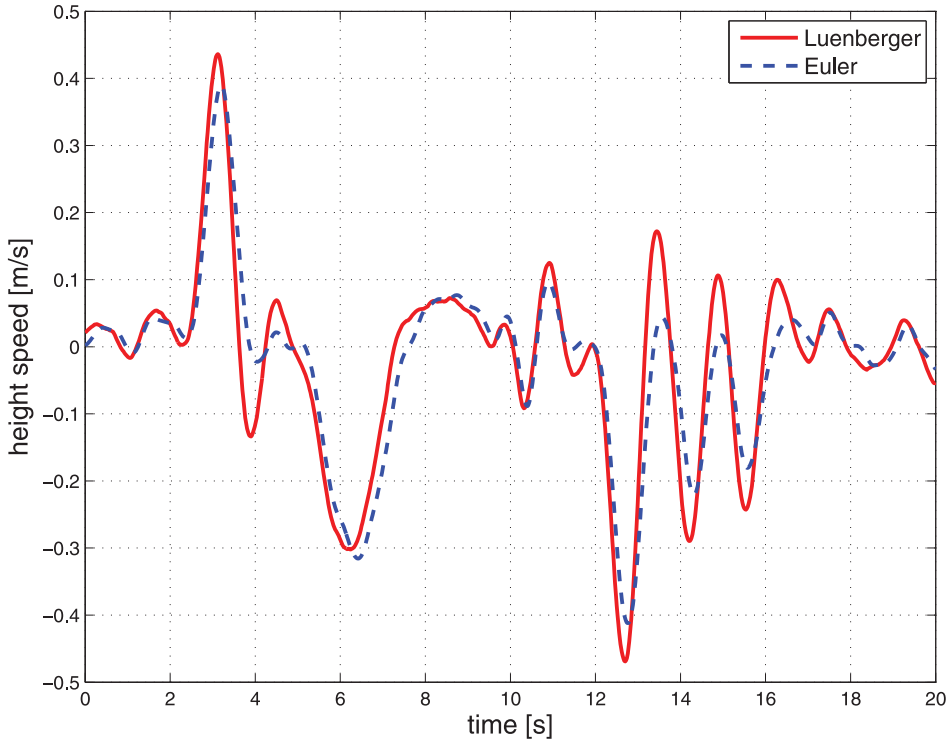

Figure 6. Altitude speed estimation comparison. The Luenberger estimator (solid red line) shows a faster response and smaller attenuation compared to the filtered Euler derivative (dashed blue line).

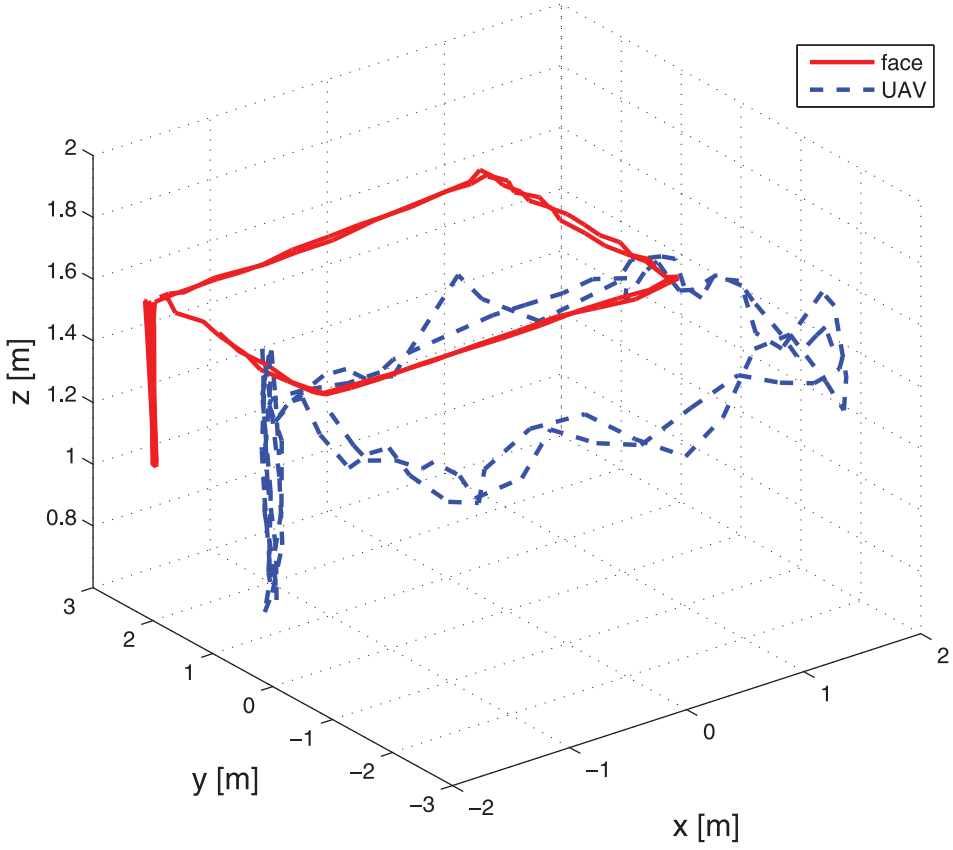

Figure 7. 3D trajectory. The mobile target (solid red line) moves describing a rectangular trajectory and the aerial vehicle (dashed blue line) is able to follow the objective with good performance.

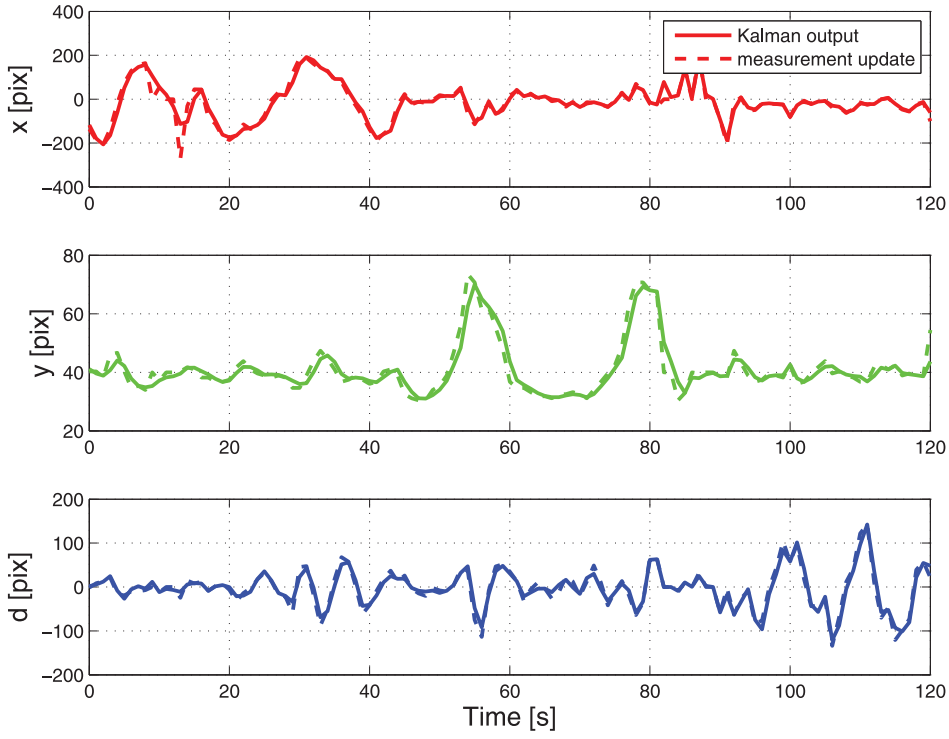

Figure 8. Kalman filter tracking estimation (solid lines) against direct detection by the vision algorithm (dashed lines). The use of a KF to track the detected object helps to filter out false detections and adds robustness against other similar objects in the scene, see for instance second 13 in the *x* variable (top plot).

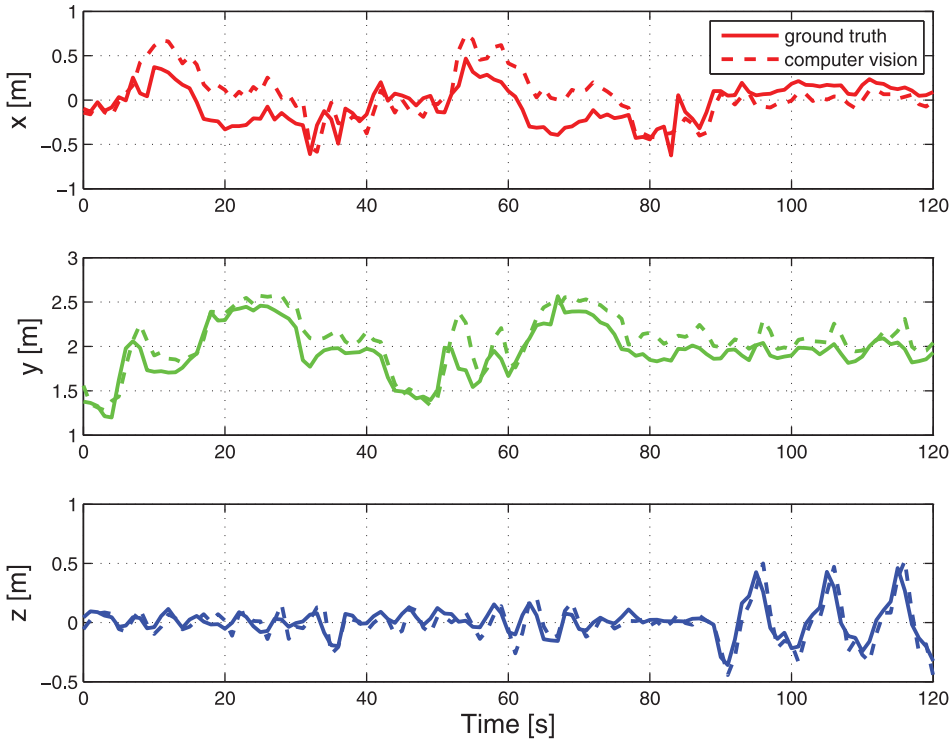

Figure 9. Relative position estimated by the computer vision algorithm (dashed line) versus ground truth from a motion capture system (solid line). The proposed algorithm proves to be consistent with respect to the ground truth, but presents a small error.

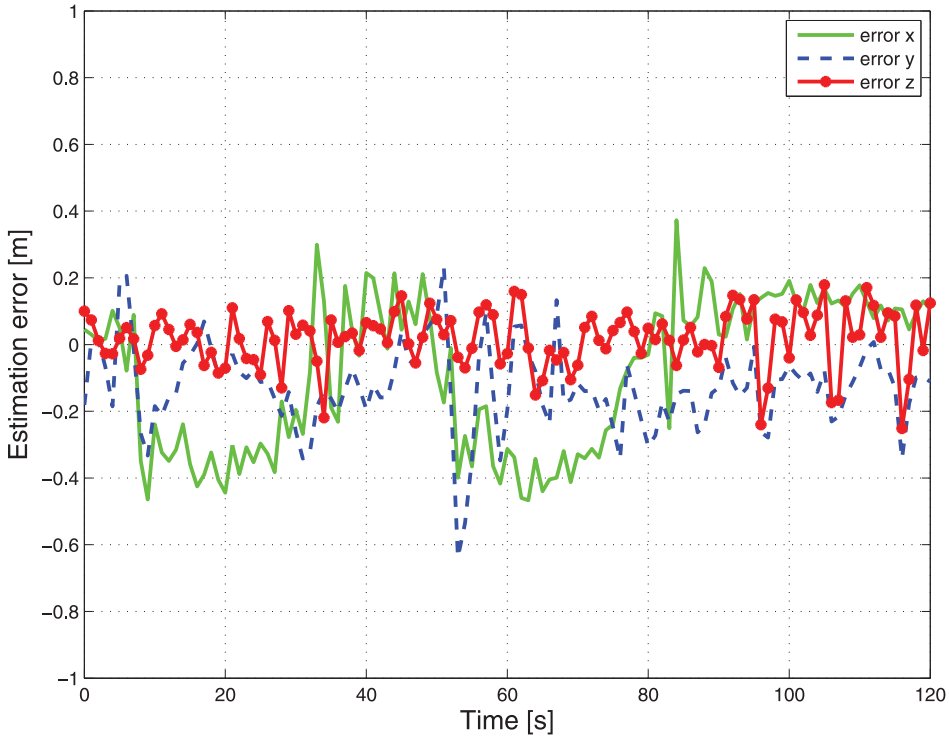

Figure 10. Computer vision estimation error. The RMSE are 0.1294 m in *x*, 0.2392 m in *y* and 0.1802 m in *z*.

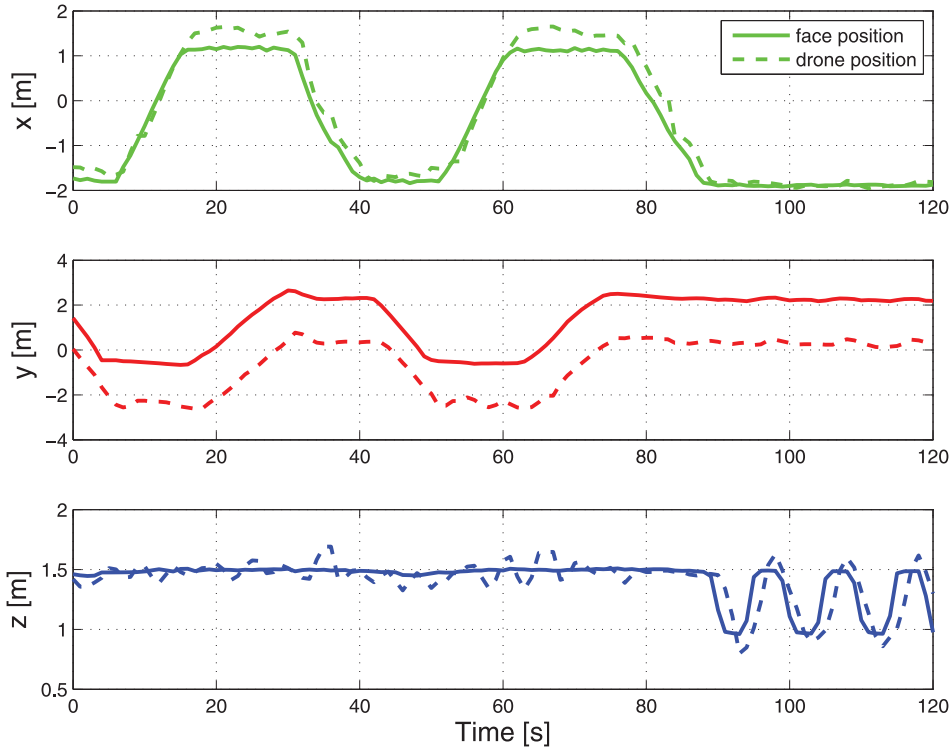

Figure 11. Relative position control performance measured by a motion capture system. The target's position (solid lines) versus the aerial vehicle position (dashed lines). The offset in the *y* coordinate corresponds to a predefined desired distance $y_{off} = 2$ m between the tracked object and the drone.

algorithm is compared against the ground truth measurement for the three axis. There, it can be observed that the estimated relative position is consistent with the reality, but small errors are presented, mainly for the *x* coordinate. It can also be noted the good performance of the depth estimation using the a-priori knowledge of the

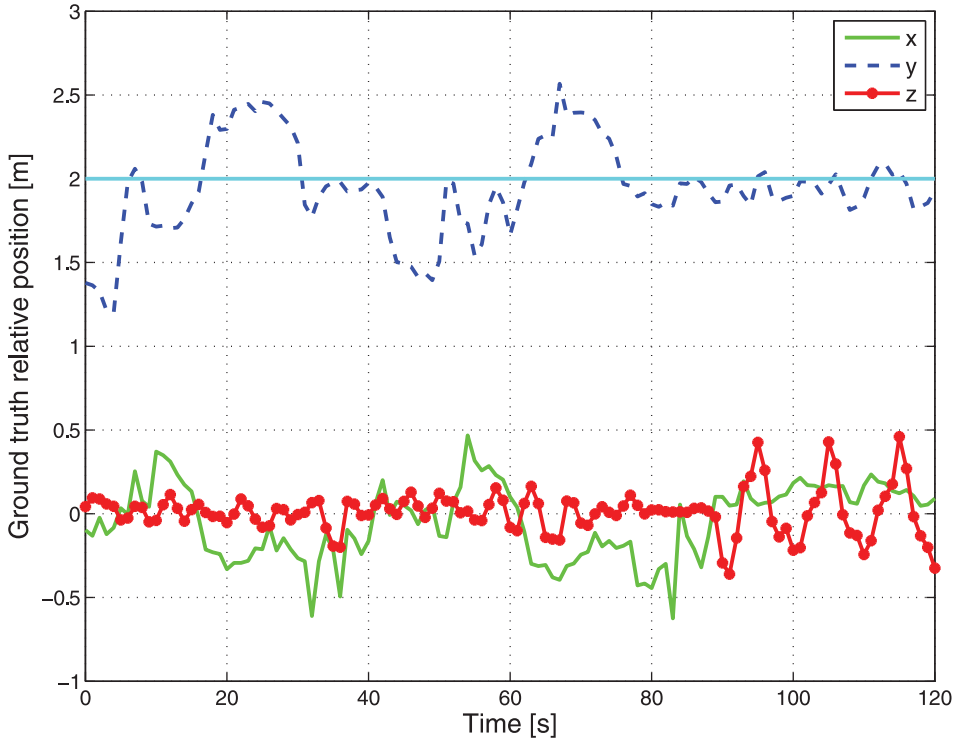

Figure 12. Drone's position relative to the target. The UAV is able to follow the objective and keep a constant distance of 2 m in front of it.

average target size. Figure 10 completes the study showing the relative position estimation errors with a Root Mean Square Error (RMSE) of 0.1294 m in x, 0.2392 m in y and 0.1802 m in z, which are acceptable for this kind of application, taking into account the use of a low-cost camera subject to relatively fast motions.

On the other hand, the performance of the relative position control strategy can be observed from Figures 11 and 12. Figure 11 shows the position of the tracked object along with the position of the UAV as measured by the motion capture system. Please note that the objective is to keep a constant distance of 2 m in front of the target, which explains the offset in the y coordinate. Finally, the relative position between the drone and the moving object is presented in Figure 12, where the validity of the proposed algorithms is confirmed to detect, track and control an UAV to follow a mobile object, with unknown dynamics.

The proposed algorithms demonstrated satisfactory performance during the experiments, even under uncontrolled conditions outdoors where illumination changes and wind gusts are a major concern. The system also proved to function effectively in spite of the presence of other target-like objects in the scene, and false positive detection, thanks to the use of a KF for tracking. Furthermore, the system showed fast response before the objective movement, being able to follow the face as long as it stays in sight. Please note that the range of operation of the detection algorithm is limited to the field of view of the camera lens, and the camera resolution to detect far away targets. Nevertheless, at current stage the system is only capable of detection and tracking of frontal faces, lacking robustness against target rotations, in particular in yaw.

6. Conclusion and future work

An implementation for an UAV which is able to detect, track and follow a moving object with unknown dynamics was conceived and successfully developed in this work, using a human face as a case of study. In order to do so, several tools and techniques were merged together to offer a full-working solution.

Relative position from the mobile target to the quadcopter was estimated by a computer vision algorithm. Object detection was accomplished by means of a Haar cascade classifier, while a KF was implemented to keep track of the object of interest, adding robustness against false positive detections and other similar objects on the image. Then a suitable transformation was proposed, by using the previously known average size of the objective, to compute the depth to the target, normally unknown for monocular vision algorithms.

A PD control strategy was implemented to deal in real-time with the relative position regulation to a time varying reference. To complete the state feedback for the

controller, a Luenberger observer was employed to estimate the missing altitude velocity. The overall system performance was tested in numerous real-time experiments indoors and outdoors, under different conditions, and proved to be a good solution for the studied problem, despite the use of a low-cost quadcopter and the simplicity of the algorithms.

It is envisioned to implement the proposed algorithms in a fully embedded UAV. Also, a wide angle lens would help to increase the field of view, allowing faster motions.

Another important improvement is to extend the computer vision algorithm to detect the object rotation as well, and control the drone's yaw angle accordingly. To do so, it is necessary to train various classifiers for different orientations and run them sequentially.

This work can be used as a base for future applications and developments, looking to improve the human–robot interaction experience. To cite some examples:

- The detection algorithm can be trained to track almost any other kind of mostly rigid object that have distinguishing views. Then, the proposed system can be used as it is to follow the desired object.
- Detecting face gestures and body movements could be used to give further commands to the quadcopter, improving the user experience.
- A powerful and interesting applications would be to detect other quadcopters and use the developed system for formation flight applications.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work has been sponsored by the ScholarOne Manuscript French National Networks of Robotics Platforms ROBOTEX (ANR-10-EQPX-44).

Notes on contributors

Diego Alberto Mercado-Ravell was born in Mexico City. He received his B.S. degree in Mechatronics Engineering from the Universidad Panamericana in Mexico in 2010, the M.Sc. degree in Electrical Engineering option Mechatronics from CINVESTAV-IPN, Mexico City in 2012, and the Ph.D. in Automation, Embedded Systems and Robotics from the University of Technology of Compiègne, France in 2015. Dr. Mercado has held post-doctoral positions at the Mechanical and Aerospace department at Rutgers, the state University of New Jersey and at UMI 3175 LAFMIA laboratory at CINVESTAV Mexico. He is currently full-time professor at the research center in mathematics CIMAT-Zacatecas in Mexico, and member of the Mexican National Research System SNI-C since 2018. His research topics include modeling and control of unmanned aerial and/or underwater vehicles, autonomous navigation, real-time embedded applications, data fusion and computer vision.

Pedro Castillo was born in Morelos, Mexico, on January 8, 1975. He received the B.S. degree in electromechanical engineering from the Instituto Tecnologico de Zacatepec, Morelos, Mexico, in 1997, the M.Sc. degree in electrical engineering from the Centro de Investigación y de Estudios Avanzados (CINVESTAV), Mexico, in 2000, and the Ph.D. degree in automatic control from the University of Technology of Compiègne, France, in 2004. His research topics include real-time control applications, nonlinear dynamics and control, aerospace vehicles, vision, and underactuated mechanical systems.

Rogelio Lozano was born in Monterrey, Mexico, on July 12, 1954. He received the B.S. degree in electronic engineering from the National Polytechnic Institute of Mexico in 1975, the M.S. degree in electrical engineering from Centro de Investigación y de Estudios Avanzados (CINVESTAV), Mexico in 1977, and the Ph.D. degree in automatic control from Laboratoire d'Automatique de Grenoble, France, in 1981. He joined the Department of Electrical Engineering at CINVESTAV, Mexico, in 1981 where he worked until 1989. He was Head of the Section of Automatic Control from June 1985 to August 1987. He has held visiting positions at the University of Newcastle, Australia, from November 1983 to November 1984, NASA Langley Research Center VA, from August 1987 to August 1988, and Laboratoire d'Automatique de Grenoble, France, from February 1989 to July 1990. Since 1990, he is a CNRS (Centre National de la Recherche Scientifique) Research Director at University of Technology of Compiègne, France. He was Associate Editor of Automatica in the period 1987–2000. He is associate Editor of the *Journal of Intelligent and Robotics Systems* since 2012 and Associate Editor in the *International Journal of Adaptive Control and Signal Processing* since 1988. He has coordinated or participated in numerous French projects dealing with UAVs. He has recently organized two international workshops on UAVs (IFAC RED UAS 2013 and IEEE RAS RED UAS 2015). He participates in the organization of the annual international conference ICUAS (International Conference on Unmanned Aerial Systems) since 2010. He is IPC Chairman of the ICSTCC in Rumania since 2012. He was Head of Heudiasyc Laboratory in the period 1995–2007. Since 2008 he is Head of the Joint Mexican-French UMI 3175 CNRS. His areas of expertise include UAVs, mini-submarines, exo-squelettons and Automatic Control. He has been the advisor or co-advisor of more than 35 Ph.D. theses and published more than 130 international journal papers and 10 books.

ORCID

Diego A. Mercado-Ravell http://orcid.org/0000-0002-7416-3190

References

[1] Peshkova E, Hitz M, Kaufmann B. Natural interaction techniques for an unmanned aerial vehicle system. IEEE Pervas Comput. 2017;16(1):34–42.
[2] https://www.myo.com/.
[3] ETH Zurich. Controlling a quadrotor using kinect, 2011.

[4] Sanna A, Lamberti F, Paravati G, et al. A kinect-based natural interface for quadrotor control. Entertain Comput. 2013;4(3):179–186.
[5] Lee D, Franchi A, Hyoung I, et al. Semiautonomous haptic teleoperation control architecture of multiple unmanned aerial vehicles. IEEE ASME Trans Mechatron. 2013;18(4): 1334–1345.
[6] Yao C, Xiaoling L, Zhiyuan L, et al. Research on the UAV multi-channel human–machine interaction system. In: IEEE editor. 2nd Asia-Pacific Conference on Intelligent Robot Systems, Wuhan, China, June 2017.
[7] Cauchard J, Jane LE, Zhai K, et al. Drone & me: an exploration into natural human–drone interaction. In: Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing, Osaka, Japan, September 2015.
[8] Kanellakis C, Nikolakopoulos G. Survey on computer vision for UAVs: current developments and trends. J Intell Robot Syst. 2017;87:141–168.
[9] Choi H, Kim Y. UAV guidance using a monocular-vision sensor for aerial target tracking. Control Eng Pract. 2014;22:10–19.
[10] Blondel P, Potelle A, Pégard C, et al. Dynamic collaboration of far-infrared and visible spectrum for human detection. In: International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 2016.
[11] https://www.lily.camera/.
[12] Graether E, Mueller F. Joggobot: a flying companion as flying companion. In: CHI, Austin, TX, 2012.
[13] Naseer T, Sturm J, Cremers D. Follow me: person following and gesture recognition with a quadrocopter. In: Intelligent Robots and Systems (IROS), Tokyo, Japan, November 2013. p. 624–630.
[14] Danelljan M, Shahbaz F, Felsberg M, et al. A low-level active vision framework for collaborative unmanned aircraft systems. In: Agapito L, Bronstein M, Rother C, editors. Computer Vision – ECCV 2014 Workshops, Zurich, Switzerland, September 2014.
[15] Sanchez-Lopez JL, Pestana J, Saripalli S, et al. Computer vision based general object following for gps-denied multirotor unmanned vehicles. In: American Control Conference (ACC), Portland, USA, June 2014.
[16] Haag K, Dotenco S, Gallwitz F. Correlation filter based visual trackers for person pursuit using a low-cost quadrotor. In: 15th International Conference on Innovations for Community Services (I4CS), Nuremberg, Germany, July 2015.
[17] Monajjemi M, Mohaimenianpour S, Vaughan R. UAV, come to me: End-to-end, multi-scale situated HRI with an uninstrumented human and a distant UAV. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, October 2016.
[18] Yao N, Anaya E, Tao Q, et al. Monocular vision-based human following on miniature robotic blimp. In: IEEE International Conference on Robotics and Automation (ICRA), Singapore, June 2017.
[19] Bradski G, Kaehler A. Learning OpenCV. O'Reilly; 2008. Chapter 13, Machine Learning; p. 459–520.
[20] Viola P, Jones MJ. Rapid object detection using a boosted cascade of simple features. In: IEEE Computer Vision and Pattern Recognition (CVPR), Kauai, Hawaii, USA, 2001.
[21] Kalman R. A new approach to linear filtering and prediction problems. J Basic Eng. 1960;82 (Series D):35–45.
[22] Grewal M, Andrews A. Kalman filtering: theory and practice using MATLAB. New York, USA: John Wiley & Sons; 2001.
[23] Brown R, Hwang Y. Introduction to random signals and applied Kalman filtering. Fourth edition, 2012. Chichester, UK: John Wiley & Sons; 1992.
[24] Bradski G, Kaehler A. Learning OpenCV. 1st ed. O'Reilly; 2008. Chapter, Camera Models and Calibratio; p. 370–404.
[25] Achtelik M, Weiss S, Siegwart R. Onboard IMU and monocular vision based control for MAVS in unknown in- and outdoor environments. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 2011.
[26] Castillo P, Lozano R, Dzul A. Modelling and control of mini-flying machines. 1st ed. Londres: Springer-Verlag; 2005. Chapter 3, The Quad-rotor Rotorcraft; p. 39–59.
[27] Bertrand S, Guénard N, Hamel T, et al. A hierachical controller for miniature VTOL UAVS: Desing and stability analysis using singular perturbation theory. Control Eng Pract. 2011;19(10):1099–1108.
[28] Smith SW. The scientist and engineer's guide to digital signal processing. San Diego, CA: California Technical Publishing; 1997.
[29] Luenberger D. An introduction to observers. IEEE Trans Automat Contr. 1971;AC-16(6):596–602.