Precise Drone Location and Tracking by Adaptive Matched Filtering from a Top-View ToF Camera

José A. Paredes*, Fernando J. Álvarez, Teodoro Aguilera, Fernando J. Aranda

University of Extremadura, 06006 Badajoz (Spain)

Departament of Eletrical Engineering, Electronics and Automation, Sensory Systems Research Group (GISS)

Abstract

Nowadays the use of drones is rapidly growing, and the autonomous navigation capability depends on knowing their position at any time. The precision and robustness of a local positioning system for these devices is of particular importance in takeoff and landing maneuvers, especially in GPS-denied environments. The main contribution of this work lies in the development of a precise local positioning and tracking system for drones. For this purpose, a ToF camera has been installed on the ceiling in order to make use of its depth maps. Taking as a reference the disturbance caused by a quadrotor in one of these maps, a novel 2D matched filter has been designed based on a Gaussian wavelet. This filter allows the system to quickly detect all drones flying in the scene. Moreover, it is dynamically adapted to the image portion that they occupy, taking into account the variation of this parameter with their flying altitude, which has also been theoretically determined. The whole algorithm leads to a precise 3D drone positioning.

In addition, the proposed system is also robust against short-time occlusions, since the measurements history can be used to predict future positions, thus avoiding the complete loss of tracking.

Keywords: ToF Camera, Drones, Adaptive Matched Filter, Local Positioning and Tracking System

1. Introduction

In the last few years, numerous works on UAVs (unmanned aerial vehicles), also known as drones, have arisen due to the increasing use of these devices, both military and civilian. One of the most important tasks consists in automatizing flights, since the pilot usually stays far away to control it (Kendoul, 2012). Ironically, the main concern is currently connected with the lack of security involved by the presence of an unmanned vehicle, as analyzed in Samland et al. (2012). Hence, knowing its position becomes essential to assemble either an autonomous system or a security one able to assume decisions based on that information.

The global positioning problem has been solved some time ago with the inclusion of GNSS (Global Navigation Satellite System) receivers for the most advanced drones, as can be seen in some off-the-shelf UAVs (ErleRobotics, 2019), (Parrot, 2019), (DeDrone, 2019), (DJI, 2019), or even in works focused on improving measurements from GPS (Global Positioning System) (Tahar & Kamarudin, 2016). Some others have implemented detecting and tracking systems, even determining whether they are hostile or not. For instance, it is manually done in Boddhu et al. (2013), where users send the photos they take of sighted UAVs to a remote server together with hand-typed information about the approximate device size, noise intensity and possible distance at which the drone is located. Finally, all information is processed and sent to other users. At least three observers are needed so as to get a suitable running. Another example can be found in Multerer et al. (2017), which addresses the issue via a 3D MIMO (Multiple Input Multiple Output) radar for jamming signals coming from potentially hazardous drones, and forcing their landing.

However, all researches described above have not been fully interested in knowing the precise position, but only a global detection and tracking. Those applications allowing the device to be as autonomous as possible require a precise and robust knowledge of its position in local environments. Obviously, these local positioning systems, intended either for indoor or outdoor, should improve the precision achieved with the former ones. This work falls within this framework and is aimed at achieving centimetric positioning in local environments and knowing the coordinates of all drones flying in the scene, via a Time-of-Flight (ToF) camera, allowing real time processing. The proposed algorithm has been adapted for better performance in an indoor environment, although the method can be extrapolated to different surroundings.

*Corresponding author

Email addresses: japaredesm@unex.es (José A. Paredes), fafranco@unex.es (Fernando J. Álvarez), teoaguibe@unex.es (Teodoro Aguilera), fer@unex.es (Fernando J. Aranda)

URL: giss.unex.es (Sensory Systems Research Group (GISS))

The rest of the paper is organized as follows: first, literature reviews in positioning drones as well as in ToF camera based positioning are introduced in Section 2; secondly, the working principles of these cameras are described in Section 3; thirdly, the whole algorithm is presented in Section 4; then, experimental tests and results are shown in Section 5; and finally, conclusions are exposed in Section 6.

2. Related Work

2.1. Drone detection and tracking

The performance of 3D positioning systems has notably improved since the number of drones in logistic industry has rocketed. These improvements have been conducted by means of different technologies. For instance, one of them widely applied in the past for local positioning in general such as RF (Radio Frequency) is now adapted to detect UAVs. There exist passive techniques, which take advantage of signals emitted by the UAV, like eavesdropping, reaching 50 m, or vibration pattern analysis (Nguyen et al., 2016); and active techniques, which study the echo from a previously emitted signal, with radar based systems (Multerer et al., 2017), (Caris et al., 2015), (Drozdowicz et al., 2016), (Klare et al., 2017), whose range reaches up to 250 m. However, the main drawback of RF lies in the expensive infrastructure needed.

Acoustic technology has been also applied. Most of the few acoustic based works leverage the well-known buzz produced by propellers, like in Birch et al. (2015), with a detection range of 20 m, or in Benyamin & Goldman (2014), achieving a detection range of 600 m, with a probability of 99.5% and a false positive rate of 3%. In this case, the authors build a complex microphone matrix, so the factor of possible decreasing costs is lost. The same problem occurs in Busset et al. (2015), where a 150 m - 290 m range is reached with an acoustic camera (Distran Omni360). Obviously, these ranges are too wide to achieve precise locations.

Furthermore, one of the most extended technologies in this field is the optical one, since artificial vision techniques can be applied to the images provided by a camera or a set of them (Kong et al., 2014). Today, an important challenge resides in decreasing the processing time for the system to be able to act in real time (at least, as quick as possible) Since the late nineties, the possibility of assembling visual odometry systems detecting UAVs has been researched. Works as Amidi et al. (1999) use an RGB camera (assembled in the drone) pointed to the ground so as to detect templates with geometric shapes and to maintain the position. Merino et al. (2006) goes beyond trying to obtain a position estimation via homography.

Nevertheless, there are also markerless ground-based vision systems, such as in Santos et al. (2015) aimed at landing drones in a ship deck. First, the vehicle is detected with an RGB camera; then, the position is estimated based on an evolutionary method; and finally, a Kalman filter corrects the outliers. A similar procedure was presented in Weiwei et al. (2015), making use of a stereo-camera system. Moreover, there exist prototypes using RGB-D (RGB + depth) cameras devoted to construct local maps of the environment, for the drone to navigate with that information (suitable paths, obstacles, etc.) (Iacono & Sgorbissa, 2018); although it depends on another positioning-navigation system to work properly. Additionally, the 3D position of the objective can be achieved via a pattern recognition algorithm, as presented in Baek et al. (2015).

Likewise, a strategy widely used in the vision field consists in sensor fusion, taking advantage of different technologies, as shown in Corke et al. (2000) and in Paredes et al. (2017) that combine the optical one with ultrasounds, in Kim et al. (2010) with ultrasounds and also an IMU (Inertial Measurement Unit), in Teuliere et al. (2010), Engel et al. (2012) and Curetti et al. (2015) with cameras and IMUs, or in Liu et al. (2017) with audio signals. Nevertheless, sensor fusion has two important drawbacks: the need for a high computing capacity and an algorithm capable of efficiently and precisely merging data.

In short, different researches on drone location have been carried out. Some of them are only focused on detection, others on distant detection and tracking, and the last ones on precise location although with high computation requirements.

2.2. ToF Camera based positioning

Employing a ToF camera could be one of the best and most economical solutions to address the requirements of a precise and robust local positioning system for autonomous UAVs. In fact, some research works can be found in the literature having already tackled the issue of detecting and tracking mobile targets, as explained below.

Just a few years after the emergence of the infrared technology applied to the generation of depth images in Lange (2000), the first positioning application begins to be developed, facing people detection and tracking. An example can be found in Gokturk & Tomasi (2004), where authors are focused on people head recognition by means of a previously trained matched filter and correlator. The results are often erroneous due to the complexity of the algorithm. A slightly simpler system consists in placing the camera on the ceiling, as in Bevilacqua et al. (2006), where people are identified by using the subtraction background technique. The reference pixel is taken in the ROI (Region of Interest) centroid, and then, a conversion between camera coordinates (pixels) and world coordinates (meters) is performed. The main drawback lies in maintaining the scene as stable as possible to prevent background changes. Besides, it does not discriminate between people and other possible new objects entering the scene.

An interesting improvement is presented in Stahlschmidt et al. (2013). Here, the authors try to identify different people in dense crowds by using a scaled matched filter

in combination with height-segmented foreground information. The pattern proposed is a Mexican hat wavelet, which is equal to the second derivative of a 2D Gaussian function:

$$\psi(r) = \frac{2}{\sqrt{3\sigma}\pi^{1/4}} \left(1 - \frac{r^2}{\sigma^2}\right) \exp\left(-\frac{r^2}{2\sigma^2}\right) \qquad (1)$$

where $r = \sqrt{x^2 + y^2}$ and σ is the standard deviation of the distribution, which in a practical sense, points out the bell width. The detection results are combined with a Kalman filter, allowing an effective tracking of several people in the same scene, as shown in a later work (Stahlschmidt et al., 2016).

Apart from detecting people, counting applications have also been developed. A prototype is proposed in Pizzo et al. (2016) trying to count people after crossing an imaginary line. Although the authors make use of a structured light device, the algorithm is similar in the case of a ToF camera. Also, Luna et al. (2017) suggest a design based on the human being morphology from a zenith view. They achieve a mean error rate of 3.1% by means of PCA (Principal Component Analysis) to distinguish people in images. One of the few shortcomings comes from the huge data volume needed to achieve high efficiency after the training step.

Nevertheless, none of these studies are focused on getting the target coordinates, i.e. on the positioning. One of the first works on knowing the location of some elements is Kohoutek et al. (2010) that calculates the distance between the camera and the surrounding furniture. For that purpose, a furniture recognition stage with known positions beforehand is necessary. In a later research, the same authors merge data from the camera and a building information map for the same end. Finally, a regular ToF device is transform into an omnidirectional one thanks the painstaking mirror system in Pirker et al. (2010). The camera-mirror set is placed on the upper part of a robot, which makes the robot able to navigate autonomously knowing its position at all times, though a very precise calibration of the information provided by the camera is needed.

3. Time-of-Flight Cameras

A ToF camera is an active range imaging device that employs time-of-flight techniques. This kind of cameras resolves distances between the scene and itself by measuring the round trip time of an infrared light signal provided by a laser or an LED. Hence, this kind of camera provides not only 2D information in a single image, but also information about the third dimension. Noteworthy is that they do not furnish volumetric data but surfaces in 3D, and for this reason these devices are also known as 2.5D cameras.

This section presents the ToF camera working principle, as well as the different matrices that these devices provide. Also, the coordinates extraction procedure is explained, which comprehends two stages: lens calibration and pixel transformation.

3.1. Working Principle

ToF cameras are composed of a simple lens together with a smart pixel matrix. The working principle consists in emitting continuously a squared modulated infrared wave. The lens receives the returning wave in a matrix of smart pixels (Lange, 2000), where each element ij is capable of computing the following information about this optical signal:

1. $A^{i,j}$ - the returning wave amplitude.
2. $D^{i,j}$ - the distance between the camera and the measured object, calculated as $D^{i,j} = \frac{\mathrm{c}}{2} \frac{(\Delta\varphi)^{i,j}}{2\pi F}$, being $(\Delta\varphi)^{i,j}$ the phase shift between the emitted and the returning wave, F the modulation frequency and c the speed of light.
3. $C^{i,j}$ - the confidence. It is valued in the interval $[0, 1] \in \mathbb{R}$ which provides information about the the measurement reliability. Low values can be due to saturation, underexposure or other reasons.

A detailed mathematical explanation can be found in Langmann (2013), and a clarifying study from the point of view of the CCD/CMOS pixels is included in Hussmann et al. (2012).

3.2. Lens Calibration

An accurate calibration when using cameras is an unavoidable step in order to get precise metrics. A camera is usually considered to be calibrated when the focal length, the principal point and the lens distortion parameters are known. However, a ToF camera needs an additional stage: depth calibration. This section copes with both the x-y calibration and the depth calibration.

The earliest aspect to be considered in the coordinates equations resides in translating the origin, since it is located in the upper left corner of the image, just in the first element in the matrix. Hence, it must be moved from here to the principal point (c_x, c_y). Usually, this point is in the center of the matrix image. Then, every pixel coordinate will be modified in the following way:

$$\left.\begin{aligned} x_p &= x'_p - c_x \\ y_p &= y'_p - c_y \end{aligned}\right\} \qquad (2)$$

where (x'_p, y'_p) is the position of the pixel taken from the upper left corner.

Related to depth, the parameter to be calibrated is the offset, since it depends on the environment illumination, and could be different for each pixel in the matrix, as studied in Kahlmann et al. (2006).

Now regarding x-y calibration, extrinsic and intrinsic parameters should be obtained, in order to fix an undistorted 2D image of the scene. They transform the world

coordinates of a point into its associated pixel coordinates. This calibration deals with radial and tangential distortions. The radial one occurs when light rays bend more the further they cross the lens from the optical center. Its coefficients (k_1, k_2, k_3) are given by:

$$\left.\begin{aligned} dx_r &= x_n \left(k_1 r^2 + k_2 r^4 + k_3 r^6\right) \\ dy_r &= y_n \left(k_1 r^2 + k_2 r^4 + k_3 r^6\right) \end{aligned}\right\} \tag{3}$$

with $(x_n, y_n) = (\frac{x}{z}, \frac{y}{z})$ and $r^2 = x_n^2 + y_n^2$, normalized coordinates. And tangential distortion occurs when the lens and the image plane do not stay parallel. The correction for distorted pixels are:

$$\left.\begin{aligned} dx_t &= 2p_1 x_n y_n + p_2 \left(r^2 + 2x_n^2\right) \\ dy_t &= p_1 \left(r^2 + 2y_n^2\right) + 2p_2 x_n y_n \end{aligned}\right\} \tag{4}$$

Besides, the intrinsic parameters are the focal length (f_x, f_y), the optical center (c_x, c_y) in pixels (also known as the principal point) and the skew coefficient α, which indicates whether the pixel axes are orthogonal (angle α).

$$\begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & \alpha & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_n + dx_r + dx_t \\ y_n + dy_r + dy_t \\ 1 \end{pmatrix} - \begin{pmatrix} c_x \\ c_y \\ 1 \end{pmatrix} \tag{5}$$

Notice that usually $f_x \simeq f_y$, i. e. pixels are square.

3.3. Coordinates Extraction

Now, a model connecting pixels with meters is needed. The model describes the mathematical relationship between the coordinates of a point in three-dimensional space and its projection onto the image plane inside the camera. The aperture is modeled as a point, and no lenses are used to focus light. It does not include geometric distortions or blurring of unfocused objects caused by lenses and finite sized apertures. That is why a previous calibration (Section 3.2) is needed.

Then, taking the geometry from Figure 1, the coordinates can be obtained. First of all, the β angle behind the lens is calculated taking (x_p, y_p) from the principal point (usually at the camera center) and the focal length f, which will be extracted either from the calibration or from the manufacturer if available:

$$\beta = \tan^{-1}\left(\frac{\sqrt{x_p^2 + y_p^2}}{f}\right) \tag{6}$$

Then, the depth coordinate z can be figured out via the β angle before the lens and inside a right triangle:

$$z = d \ \cos\beta \tag{7}$$

where d is the range to the target.

Next, it can be stated that the α angle is the same in both object and image planes, since the inclination respect to the axes is maintained. Then, α is taken from the pixel matrix as:

$$\alpha = \tan^{-1}\left(\frac{y_p}{x_p}\right) \tag{8}$$

From Figure 1, the hypotenuse h can be written as $h = \sqrt{d^2 - z^2}$, and the x and y coordinates are:

$$\left.\begin{aligned} x &= h\sin(2\pi - \alpha) = -h\sin\alpha \\ y &= h\cos(2\pi - \alpha) = h\cos\alpha \end{aligned}\right\} \tag{9}$$

where the 2π factor comes from the inversion in the X and Y axes.

The calibration process related to intrinsic parameters has been already carried out by the camera manufacturer used in this work. So this device provide three complementary matrix, whose elements are in meters:

1. $Z^{i,j}$ - the altitude of the scene.
2. $X^{i,j}$ and $Y^{i,j}$ - the world coordinates.

Hence, only a previous step of straightening is enough to get the world coordinates once the drone is detected.

4. Algorithm Description

This section presents the details regarding the proposed detection algorithm, which basically consists in computing the 2D correlation between a processed version F_{NM} (Section 4.1) of the depth matrix D_{NM} taken from the ToF camera installed on the ceiling, and a 2D wavelet Ψ_{PQ} representing the disturbance caused by an UAV:

$$\Theta_{KL} = F_{NM} \otimes \Psi_{PQ} \tag{10}$$

where $\otimes$ represents the discrete correlation operator. Every element is given by

$$\Theta^{k,l} = \sum_{i=1}^{M}\sum_{j=1}^{N} F^{i,j} \ \left(\Psi^{i-k,j-l}\right)^* \quad \begin{cases} -P+1 \le k \le M-1 \\ -Q+1 \le l \le N-1 \end{cases} \tag{11}$$

The operator $*$ denotes complex conjugation. For this case, k and l constitute shifts in row and columns respectively (if negative, left- or upwards; if positive, right- or downwards). Next, the pixels belonging to the drone are identified and translated into world coordinates. For clarity, this method can be divided into three stages described in detail next: image conditioning, 2D wavelet adjustment, and coordinates extraction. An overview of the algorithm can be seen in Figure 2, where the three stages appear together with the previous step of data acquisition by means of the camera.

4.1. Image Conditioning

As mentioned above, the ToF camera is installed on the ceiling to get information from the scene. The origin of coordinates has been assumed on the floor for simplicity,

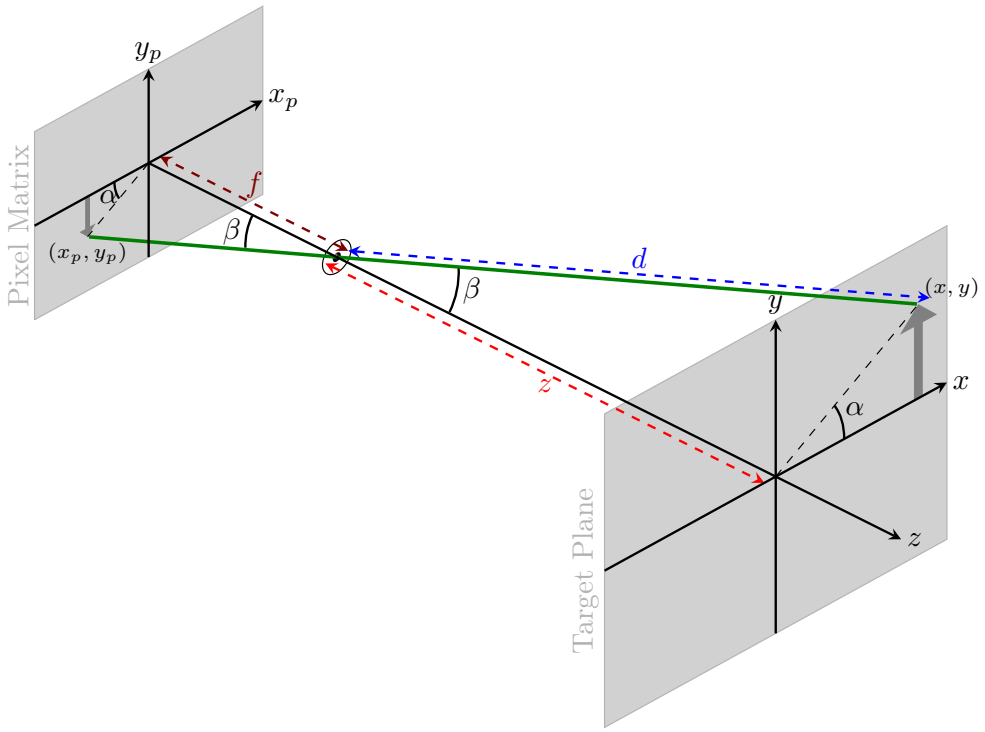


Figure 1: Pinhole camera model. In green, the ray coming from the problem point (x, y, z) and arriving at the image plane through the pinhole. In blue, the data acquired by the ToF camera.

just at the intersection between the orthogonal line coming from the lens and the floor plane.

The first step in the algorithm consists in distinguishing the foreground from the background. For that purpose, a series of preceding depth images $D_{NM}[n-n_0]$ is averaged in order to get the background B_{NM} at instant n:

$$B_{NM}[n] = \frac{1}{n_f} \sum_{n_0=1}^{n_f} [D_{NM}n - n_0] \tag{12}$$

where n_f is the number of measurements, 10 being enough for controlled environments, where the background remains unaltered. One of the advantages of using a ToF camera is that the background does not need to be upgraded over time since it is not affected by environment lights. Now, the foreground $\hat{F}_{NM}[n]$ can be obtained by subtracting the background from every instant frame:

$$\hat{F}_{NM}[n] = |B_{NM}[n] - D_{NM}[n]| \tag{13}$$

in such a way that all elements exceeding a certain threshold correspond to new objects appearing in the scene.

Next, as the confidence matrix $C_{NM}[n]$ is available, the most ambiguous measures can be discarded by multiplying with this matrix:

$$F_{NM} = \hat{F}_{NM} \circ C_{NM} \tag{14}$$

Here, the operator $\circ$ represents the Hadamard product, where each element F^{ij} in the new matrix comes from the product of elements $\hat{F}^{ij} \cdot C^{ij}$.

4.2. Wavelet Dynamic Adjustment

Detecting a drone in an image could be a rather simple task, but when a sequence of consecutive images is taken and several UAVs appear at the same time, the task becomes more complex. On one side, a precise positioning is required to prevent the system from mixing up one trajectory with another belonging to a different drone. On the other side, a drone covers a different area in the image depending on the flying altitude.

To solve the first problem, this work proposes the use of a new correlation pattern: the fourth derivative of the Gaussian function, which coincides with the second derivative $\psi''(R)$ of Equation (1):

$$\Psi(R) = \frac{2}{\sqrt{3\sigma}\pi^{1/4}} \left(\frac{1}{\sigma^4} R^2 \exp\left(\frac{-R^2}{2\sigma^2}\right) - \frac{1}{\sigma^2} \exp\left(\frac{-R^2}{2\sigma^2}\right) \right) \tag{15}$$

The 3D representation of this function is shown in Figure 3 and compared with the disturbance caused by an UAV. Henceforth, this function will be referred to as *Gnome Hat* wavelet, drawing an analogy with the Mexican hat one described above. This new wavelet is also based on an even derivative, thus maintaining the radial symmetry. But since this derivative is of higher order, its slope is also sharper and the bell narrower. Because of these characteristics, the drone location will be always very close to the UAV body center, as long as the second problem is solved. As mentioned, the drone size in the image varies

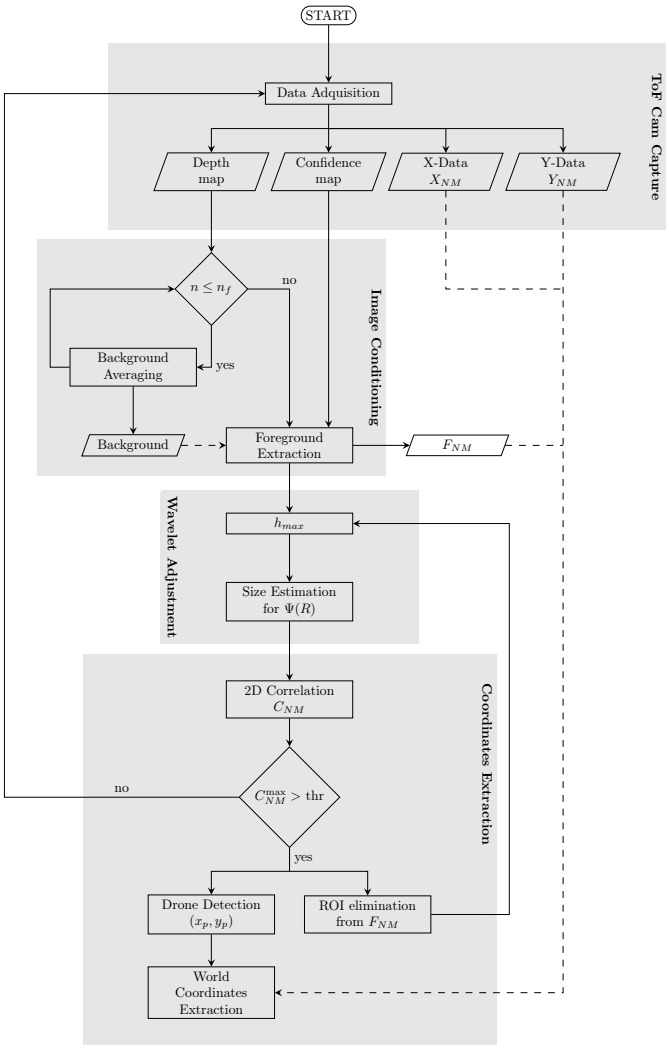
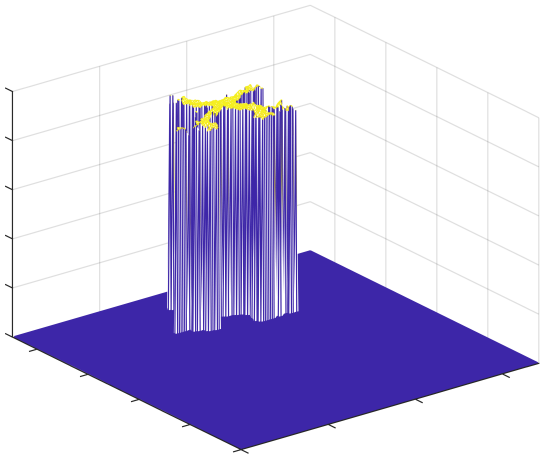
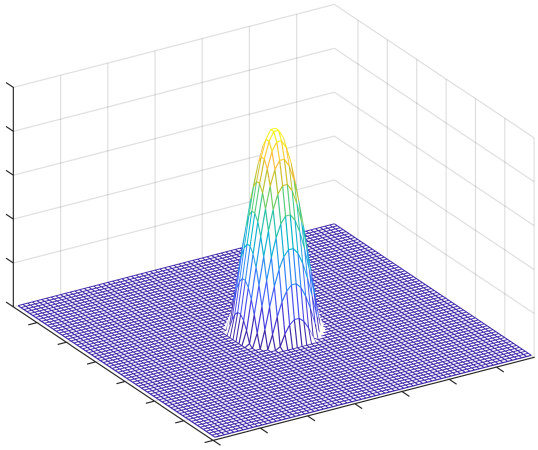


Figure 2: Flowchart depicting the whole procedure. The system starts by capturing information from the scene (ToF Cam Capture), and next the algorithm follows the three stages: image conditioning, wavelet dynamic adjustment and coordinates extraction.



(a) Foreground after image conditioning.



(b) Correlation pattern

Figure 3: (a) The resulting depth matrix after the image conditioning step. (b) Fourth derivative of a 2D Gaussian function. As the disturbance varies with the flying altitude, the size of this function will be adapted.

significantly depending of its flying altitude, in such a way that the correlation with the pattern might not work properly. For instance, an average height of a human being is assumed in Stahlschmidt et al. (2013) for the computation of σ (the parameter related to the function width) according to the expected size of people silhouettes, which in turn also depends on the height at which the camera is positioned above the ground level. Then, the second proposal of this work consists in adapting the pattern size in a dynamic way.

To summarize, this part of the algorithm works as follows:

1. First, the maximum height in the image is computed $h_{\max}$.
2. Secondly, the Gnome Hat pattern is generated Ψ_{PQ}, with $P = Q$ for a quadrotor, since its shape is isotropic. In this way, a circle is defined by its size, whose radius $\mathrm{R}(h_{\max})$ depends directly on the flying altitude. This circle corresponds with the area covered by the drone, the ROI.
3. Then, the 2D correlation is computed its maximum value extracted: $(C_{KL})^{\max} = (x_c, y_c)$. Now, the drone is located at:

$$(x_p, y_p) = \left(x_c - \frac{\mathrm{R}(h_{\max})}{2} + 1, y_c - \frac{\mathrm{R}(h_{\max})}{2} + 1 \right) \tag{16}$$

Note that the factor $\frac{1}{2}$ derives from the fact that C_{KL} reaches its maxima not in the center of where the drones are supposed to be, but in the right-bottom corner.
4. Finally, if this maximum is greater than a threshold previously established, the algorithm goes back to point #1 in order to locate more drones. Otherwise, there are no more drones in the scene.

The final stage of this algorithm consists in extracting the world coordinates in the pixels (x_p, y_p) from the data

maps delivered by the camera, following the procedure already explained in Section 3.3:

$$\begin{aligned} x &= (X_{NM})^{x_p,y_p} \\ y &= (Y_{NM})^{x_p,y_p} \\ z &= (F_{NM})^{x_p,y_p} \end{aligned} \tag{17}$$

This procedure is illustrated with a pseudo-code in Appendix A, complementing the flowchart in Figure 2.

4.3. Time Complexity Analysis

To conclude this section, a time complexity analysis of the proposed algorithm has been carried out, in order to estimate the number of basic operations performed by this algorithm as a function of the input images size η (in pixels). The three stages considered in this analysis have been image conditioning, wavelet adjustment and coordinate extraction, already represented in Figure 2 below the image capture initial stage.

The basic operations in the first one, image conditioning, are only subtractions and multiplications element by element, thus giving a time complexity of $\mathcal{O}(\eta)$[1]. In the next stage, the wavelet adjustment, the number of operations also increases linearly, since the higher the pixel density, the higher the number of them covered by a drone, in the same ratio, and the larger the pattern correlation. Thus, it has an $\mathcal{O}(\eta)$ complexity. And finally, the coordinates extraction stage includes a correlation between two matrices. In the worst case, these matrices (depth matrix and pattern correlation) will have the same size. The number of operations needed for a typical two dimensional correlation could be represented as $\mathcal{O}(\eta^4)$. But, if the calculation is made by means of FFTs (Fast Fourier Transforms), the time complexity is reduced to $\mathcal{O}(\eta^2 \log^2 \eta)$.

Now, considering that the time complexity of an algorithm composed of several stages corresponds to that of the most complex stage, it can be concluded that the proposed algorithm has an $\mathcal{O}(\eta^2 \log^2 \eta)$ complexity.

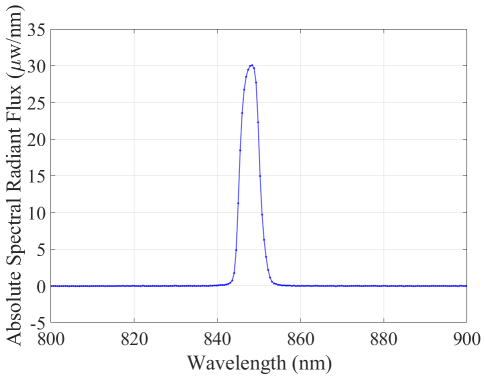
5. Performance Analysis and Results

As final step in this development, this section presents the analysis of the proposed system in a real scenario. The experimental setup will next be described, as well as the pattern size function extraction and the global system performance.

5.1. Experimental Setup

For this purpose, the ToF camera chosen has been the Camboard Picoflexx (Figure 4a), a low cost development kit based on PMD technology, which is connected to the computer via an USB port. The image resolution is $N \times M = 224 \times 171$ pixel^2. Its ToF Sensor is a IRS1145C Infineon REAL 3 with a frame rate up to 45 FPS. The IR illumination source is VCSEL (Vertical-Cavity Surface-Emitting Laser) based, whose main advantage resides in the spectral characteristics. Figure 4b provides the measured spectrum, experimentally obtained in our laboratory with a miniature spectrometer STS-NIR (Ocean Optics, 2019) from Ocean Optics, properly calibrated before. As can be seen, the curve is approximately centered at 848 nm, and it remains stable and narrow around this value.

(a)



(b)

Figure 4: (a) ToF Camera used in the experiments: Camboard Picoflexx from PMD. (b) Absolute Spectral Radiant Flux coming from the VCEL source of the camera.

The drones used in the experimental tests, Syma X15W (Syma, 2019), are shown in Figure 5. They have a wingspan of approximately 31 cm and its total size is $22 \times 22 \times 5$ cm^3. In addition, the algorithm has been programmed in Matlab code and will run under a laptop with an Intel(R) Core (TM) i7-7700HQ CPU at 2.80 GHz and a GeForce GTX 1070 GPU.

5.2. Extraction of the pattern size

As explained in Section 4.2, the expression relating the change in the pattern size as a function of the flying altitude R(h) needs to be extracted, since the adaptive matched filter will be applied by means of this adapted

[1]Here, Big O Notation is used, and η represents the number of operations.

Figure 5: Drone used in the experimental tests. Syma X15W - wingspan: 31 cm, size: $22 \times 22 \times 5$ cm^3.

pattern. In order to do that, this expression will be obtained from Figure 6, where the drone radius has been represented as r. It is straightforward from this figure that the drone sizes in the image are $R_1 = \frac{r\ f}{d_1}$ and $R_2 = \frac{r\ f}{d_2}$, since the angles remain the same, as it has been explained in Section 3.3. Then $R_1 d_1 = R_2 d_2$ and depth and radius show an inversely proportional relationship.

With respect to the flying altitudes h_1 and h_2, the previous expression can be written as $R_1(\mathrm{h_{cam}} - h_1) = R_2(\mathrm{h_{cam}} - h_2)$. If the data related to position 2 are taken as reference values, the expression of the radius at any flying altitude becomes:

$$\mathrm{R}(h) = \mathrm{R_{ref}}\ \frac{\mathrm{h_{cam}} - \mathrm{h_{ref}}}{\mathrm{h_{cam}} - h} \tag{18}$$

whose result is expressed in pixels. This curve is shown in Figure 7 with a solid blue line, for the following values that have been experimentally measured: $\mathrm{h_{cam}} = 2.945$ m, $\mathrm{R_{ref}} = 22$ pixel and $\mathrm{h_{ref}} = 1.16$ m. To verify the validity of equation (18), several snapshots of the UAV in fixed positions have been taken, measuring its size in pixels at different altitudes. These experimental measurements are represented by red crosses in the same figure.

5.3. Results

Next, a performance analysis of the proposed algorithm is carried out. Firstly, some steps of the detection stage are depicted in Figure 8. This procedure starts with a depth image taken from the camera on the ceiling (Figure 8a), and after subtracting the background (Figure 8b), a 2D correlation is carried out (Figure 8c) with a Gnome Hat pattern (Equation 15) whose size is obtained from the maximum height in the depth image through the equation (18). Needless to say, the maximum value of this correlation coincides with the drone body. Hence, going back to the depth map, the algorithm extracts the coordinates (x, y, z). Next, this ROI is removed from the correlation image (Figure 8d), and the procedure starts again. Once all objects have been located, every drone is labeled (Figure 8e). It is interesting to analyze what would happen if a fixed pattern (i.e. fixed height) was taken. In this case, a drone could be lost with a too large pattern o it could turn out in different detections in the opposite situation. These two possibilities of failure are shown in Figure 9.

The labeling stage involves an assignation problem. Every new drone position needs to be assigned to the corresponding drone. Hence, this stage starts by estimating the next position of every drone. To this end, the drone velocity is supposed to be constant in the short period of time between two consecutive measurements. In this way, if the current position is given by $\mathbf{r}[n] = (x[n], y[n], z[n])$, the following estimated one will be:

$$\hat{\mathbf{r}}[n+1] = \mathbf{r}[n] + \mathbf{v}[n]\ \Delta n \tag{19}$$

where the velocity is calculated by averaging the velocity in the last n_i time intervals:

$$\mathbf{v}[n] = \frac{1}{n_i} \sum_{\nu=n-n_i}^{n} \frac{\mathbf{r}[\nu] - \mathbf{r}[\nu-1]}{\Delta n} \tag{20}$$

For this experiment, $n_i = 3$.

Now, a cost matrix is created from the estimation $\hat{\mathbf{r}}$, whose elements are the Euclidean distances to each new measurement in the next image:

$$\Gamma^{w,m} = |\mathbf{r}_w[n] - \hat{\mathbf{r}}_m[n]| \quad \begin{cases} 1 \le w \le W \\ 1 \le m \le M \end{cases} \tag{21}$$

where W represents the number of detected drones in the image and M the number of estimated positions. Next, the Γ_{WM} matrix is used as the input of the Kuhn-Munkres method (Munkres, 1957), also known as the Hungarian method, which is a combinatorial optimization algorithm that solves the assignment problem in polynomial time. In fact, its time complexity is $\mathcal{O}(\bar{\eta}^4)$, where $\bar{\eta}$ is the number of drones flying at the same time. However, as this number is expected to be much less than the number of pixels in each image, its number of operations $\bar{\eta}$ can be neglected with respect to the total amount of operations η explained in Section 4.3. Depending on the matrix elements, this method allows to classify the results in assignments, unassigned estimations and unassigned detections. The assignments are detections belonging to a drone that is already flying in the image. The unassigned detections are those pertaining to new drones appearing in the scene. And finally, the unassigned estimations are associated with the lost drones. These are due to either a drone flying out of the camera viewing area or an occlusion, typically from the crossover of drones flying at different altitudes.

An example of the complete performance, having applied the positioning equations, is depicted in Figure 10. It is composed of five graphs: the first one with the depth map, the second one renders the trajectories followed by each drone in the scene; the other three figures provide the different plane views (XZ, YZ and YX) detailing the positioning. As revealed by these graphs, the drones have

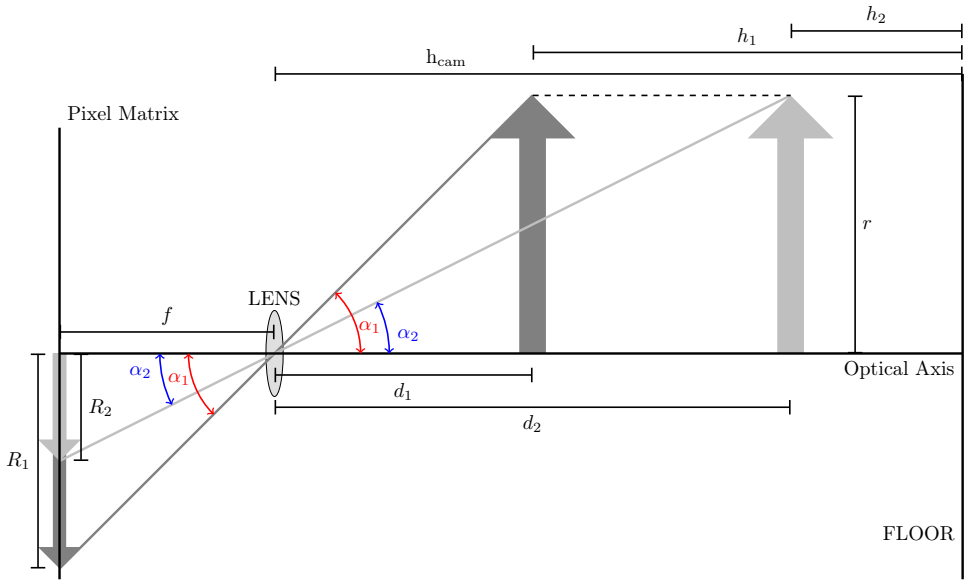


Figure 6: Scene geometry of a drone at different heights. The relationship between the radius in the image and the height is linear, as expected.

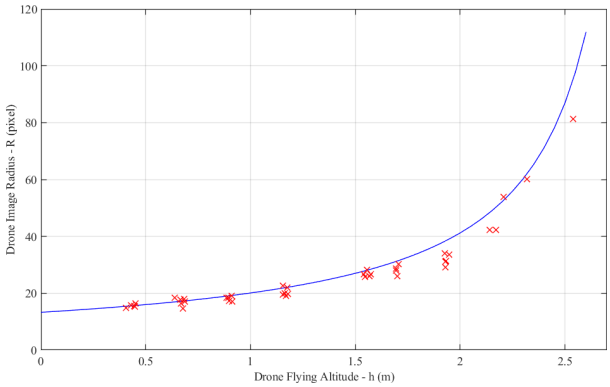


Figure 7: Curve governing the change in the drone radius in the image with respect to the flying altitude [equation (18)].

been positioned and tracked from the beginning of their appearance with centimetric precision. The most fluctuating measurements can be attributed to detections near the border of the image. In those cases, the drone body is partially out of the image, but the measurements are quickly stabilized, only two or three frames later.

This work also deals with the problem of occlusion. If a detected drone is lost, its estimated position will be available for a short period of time (7 frames in tests). During this time, this estimation is part of the variables in the cost matrix, which means that the drone position is still available and it is not totally lost. If the system does not find a real position during these 7 frames, the drone is assumed to be lost. Figure 12 shows the performance in this situation. As can be seen, the detection is estimated during the occlusion, and the real measurements are recovered without losing the label.

The whole performance is shown in a summary video (Appendix B). All the situations described above appears in this short fragment. Noteworthy, the amount of memory allocated in every loop is approximately 2.2 MB, and the algorithm execution time (10 ms on average) remains far below the time the camera spends to get a frame, thus allowing a real time tracking. Indeed, it has been optimized to work in a controlled indoor environment where MAVs can need a precise positioning. More demanding applications would require faster frame rates implying shorter acquisition times. A reliable operation of the algorithm has been experimentally observed with frame rates of up to 35 FPS. Finally, some execution time measurements have been experimentally taken so as to validate the theoretical time complexity presented in Section 4.3. The results are shown in Figure 11, where execution time is measured for different input image sizes. In this figure, the behavior of time complexity can be observed, since it differs from the experimental execution time in a constant term. As can be seen, the execution time fits well a type-$(\eta^2 \log^2 \eta)$ curve, matching the estimations of that section.

Furthermore, the total volume covered by this system configuration is approximately 40 m^3. In a wider environment, a solution of several ToF cameras encompassing all the space could be taken, where a calibration among them would be compulsory. Definitively, the whole system must be configured *in situ*, taking into account the necessities and using a camera that fulfills the minimum performance requirements.

6. Conclusion

Due to the necessity of having precise coordinates in GPS-denied environments, this paper has developed a precise and robust local positioning and tracking system for

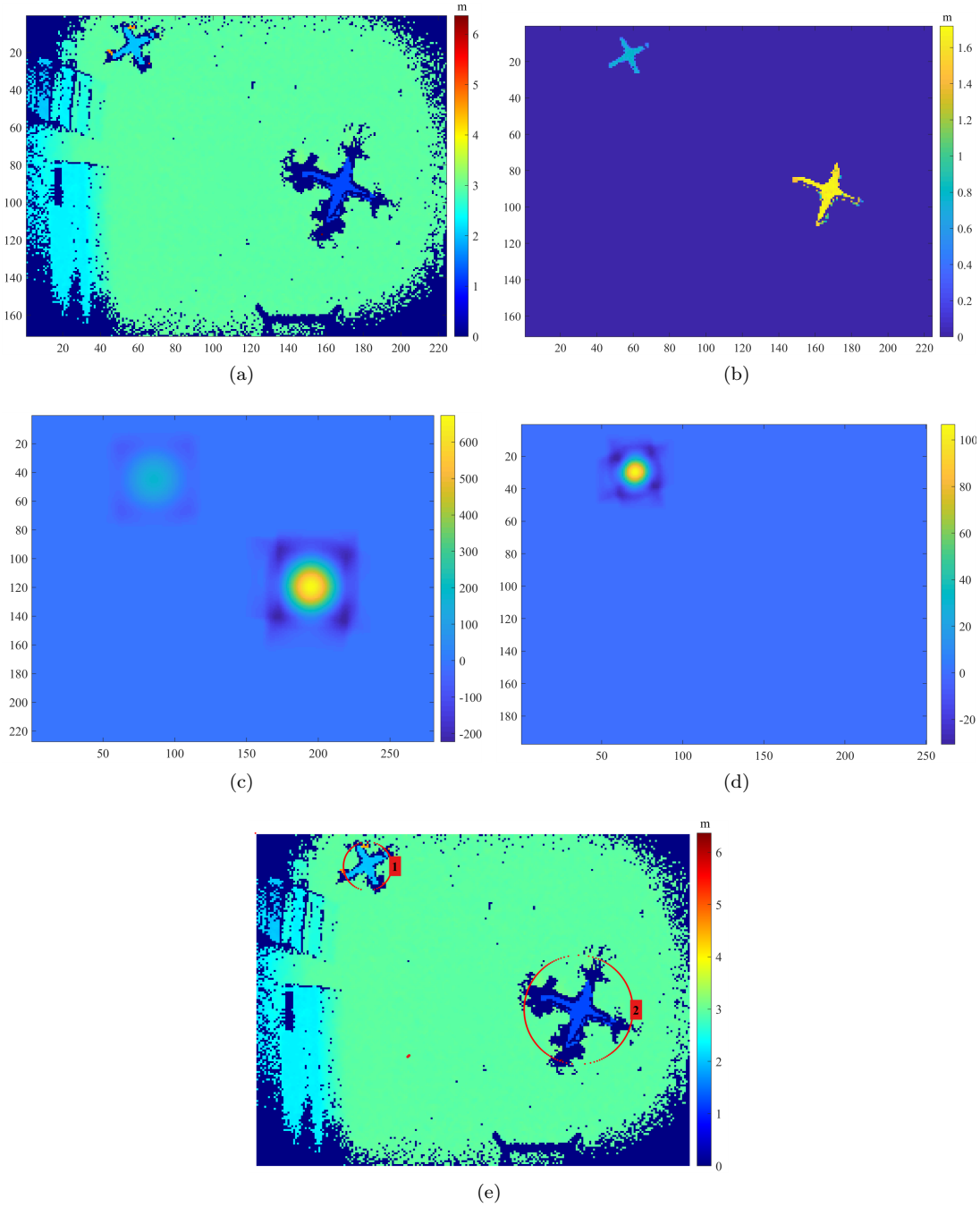
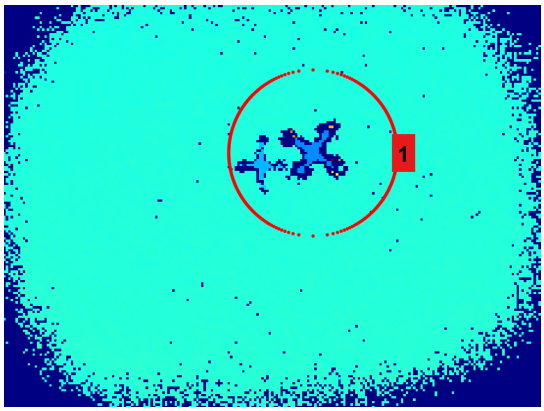


Figure 8: Detection and classification algorithm. (a) Initial depth map. (b) Depth map after background subtraction. (c) The pattern is matched with the maximum height in the last step, to detect and locate a drone. (d) Then, the ROI of the former drone is removed, and the algorithm starts again. (e) If there are no more objects, every drone is labeled.

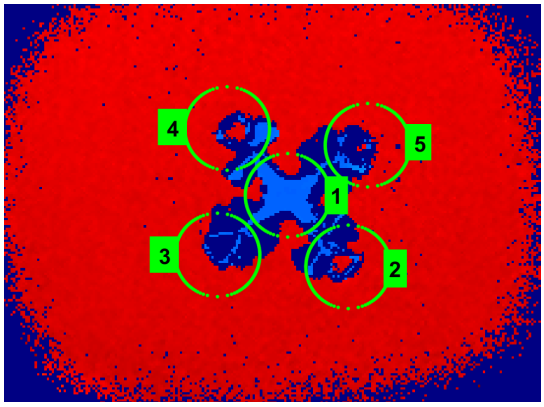
drones. For that purpose, a ToF camera has been used. This camera, installed on the ceiling for a top view, provides a depth map of the environment, the X and Y corresponding data, as well as a confidence map with the reliability of the measurement in every pixel.

One of the main challenges tackled in this work has been the detection of drones flying at different altitudes. To this end, a novel 2D matched filter based on a Gaussian wavelet is dynamically adapted in real time to apply it to a processed version of the scene depth map. The dynamic adjustment is based on a theoretical model for the variation of the image drone size with respect to its flying altitude, which has been also experimentally corroborated.

Furthermore, an accurate tracking has been possible thanks to the position predictions based on the velocity estimation of each UAV. This allows the system to label them, and to avoid the loose of tracking in case of occlusions.



(a)



(b)

Figure 9: Failures on detection when no adaptive height algorithm is used. (a) A drone could be lost in the case of the pattern is too large. (b) A drone could turn out in different detections if the pattern is too small.

Author statement and acknowledgments

The authors' contributions to the paper has been as follows: José A. Paredes: conceptualization, methodology, software and writing - original Draft; Fernando J. Álvarez: formal analysis, writing - review & editing and supervision; Teodoro Aguilera: validation, writing - review & editing; Fernando J. Aranda: writing - review & editing and visualization.

This work has been supported in part by the Spanish Government and the European Regional Development Fund (ERDF) through Project MICROCEBUS under Grant RTI2018-095168-B-C54, and in part by the Regional Government of Extremadura and ERDF-ESF under Project GR18038 and through the Pre- Doctoral Scholarship under Grant 45/2016 Exp. PD16030.

Appendix A. Pseudo-coded Algorithm

Algorithm 1 Proposed algorithm

function TofSnapshot ▷ Retrieving data from camera
  $X_{NM} \leftarrow data.x$
  $Y_{NM} \leftarrow data.y$
  $Z_{NM} \leftarrow data.z$
  $C_{NM} \leftarrow data.c$
  return X_{NM}, Y_{NM}, Z_{NM}, C_{NM}
end function

function MainBody

  for $t_0 \leftarrow 1, 10$ **do** ▷ Background Averaging
    TofSnapshot
    $D_{NM}(t_0) \leftarrow Z_{NM}$
  end for
  $B_{NM} \leftarrow$ average $(D_{NM}(1, 10))$

  while user do not stop it **do**
    TofSnapshot
    $F_{NM} \leftarrow |B_{NM} - Z_{NM}| \circ C_{NM}$ ▷ Foreground Extraction
    while there are drones in image **do**
      $h \leftarrow \max(F_{NM})$
      $R \leftarrow R_{ref}(h_{cam} - h_{ref})/(h_{cam} - h)$
      Gnome Hat Generation $\Psi(R)$
      $\Theta_{KL} \leftarrow F_{NM} \otimes \Psi(R)$ ▷ 2D Correlation
      $(x_\Theta, y_\Theta) \leftarrow \underset{\alpha,\beta}{\operatorname{argmax}} (\Theta_{KL})^{\alpha,\beta}$
      $x_p \leftarrow x_\Theta - R/2 + 1$
      $y_p \leftarrow y_\Theta - R/2 + 1$
      $x \leftarrow (X_{NM})^{x_p, y_p}$ ▷ Coordinates Extraction
      $y \leftarrow (Y_{NM})^{x_p, y_p}$
      $z \leftarrow (Z_{NM})^{x_p, y_p}$
      return x, y, z
      for the drone area defined by a circle with radius R **do**
        $\forall\, i, j \setminus (i - x_p)^2 + (j - y_p)^2 \leq R$, $F^{i,j} \leftarrow 0$ ▷ ROI elimination
      end for
    end while
  end while

end function

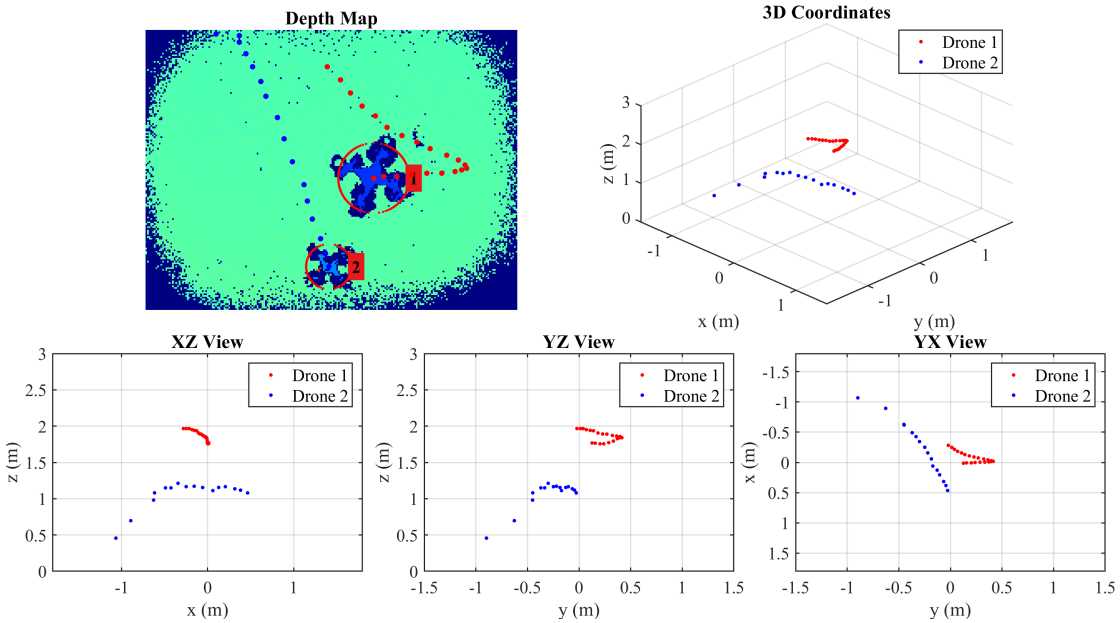


Figure 10: Detail of the whole positioning and tracking. At the top, the depth map with the trajectories of every UAV, and the 3D display. At the bottom, the three different plane views detailing both the positioning and the tracking stages.

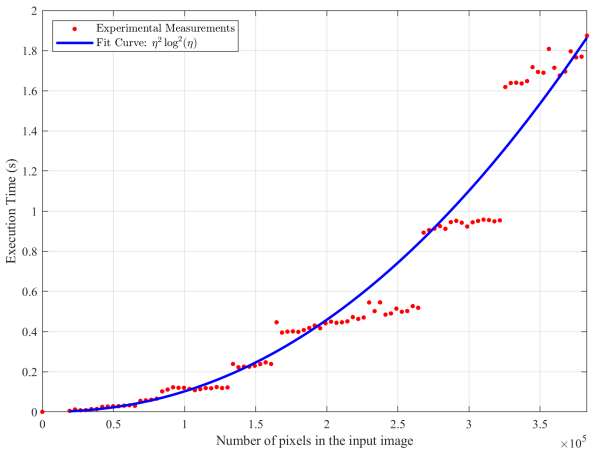


Figure 11: Experimental execution time (red dots) and type-$(\eta^2 \log^2 \eta)$ time complexity theoretical curve (blue line).

Appendix B. Supplementary Data

Supplementary data include a summary video showing the system performance. The depth map with the trajectories of every UAV and the 3D display are shown at the top, and the three different plane views detailing both the positioning and the tracking stages at the bottom. It can be found with this article online at: ——

References

Amidi, O., Kanade, T., & Fujita, K. (1999). A Visual Odometer for Autonomous Helicopter Flight. *Robot. Auton. Syst.*, *28*, 185–193. doi:10.1016/S0921-8890(99)00016-0.

Baek, J.-Y., Park, S.-H., Cho, B.-S., & Lee, M.-C. (2015). Position Tracking System using Single RGB-D Camera for Evaluation of Multi-Rotor UAV Control and Self-Localization. In *IEEE International Conference on Advanced Intelligent Mechatronics (AIM)* (pp. 1283–1288). Busan (South Korea): IEEE. doi:10.1109/AIM.2015.7222715.

Benyamin, M., & Goldman, G. H. (2014). *Acoustic Detection and Tracking of a Class I UAS with a Small Tetrahedral Microphone Array*. Technical Report. URL: http://www.dtic.mil/docs/citations/ADA610599.

Bevilacqua, A., Stefano, L. D., & Azzari, P. (2006). People Tracking Using a Time-Of-Flight Depth Sensor. In *IEEE International Conference on Video and Signal Based Surveillance*. Sidney (Australia): IEEE. doi:10.1109/AVSS.2006.92.

Birch, G. C., Griffin, J. C., & Erdman, M. K. (2015). *UAS Detection Classification and Neutralization: Market Survey 2015*. Technical Report USDOE National Nuclear Security Administration (NNSA). URL: http://www.osti.gov/scitech/biblio/1222445. doi:10.2172/1222445.

Boddhu, S. K., McCartney, M., Ceccopieri, O., & Williams, R. L. (2013). A Collaborative Smartphone Sensing Platform for Detecting and Tracking Hostile Drones. In T. Pham, M. A. Kolodny, & K. L. Priddy (Eds.), *SPIE Defense, Security, and Sensing*. Baltimore (United States): Spie-Int Soc Optical Engineering volume 8742. doi:10.1117/12.2014530.

Busset, J., Perrodin, F., Wellig, P., Ott, B., Heutschi, K., Ruehl, T., & Nussbaumer, T. (2015). Detection and Tracking of Drones Using Advanced Acoustic Cameras. In E. M. Carapezza, P. G. Datskos, C. Tsamis, L. Laycock, & H. J. White (Eds.), *SPIE Security + Defence*. Strasbourg (France) volume 9647. doi:10.1117/12.2194309.

Caris, M., Johannes, W., Stanko, S., & Pohl, N. (2015). Millimeter Wave Radar for Perimeter Surveillance and Detection of MAVs (Micro Aerial Vehicles). In *16th International Radar Symposium (IRS)* (pp. 284–287). Bonn (Germany): German Institute of Navigation (DGON). doi:10.1109/IRS.2015.7226314.

Corke, P., Sikka, P., & Roberts, J. M. (2000). Height Estimation for an Autonomous Helicopter. In *Experimental Robotics VII (ISER 2000)*. Wikiki (United States): Australian Robotics and Automation Association. doi:10.1007/3-540-45118-8_11.

Curetti, M., Serra, M., Bravo, S. G., Flesia, A. G., & Mathé, L. (2015). Use of Inertial and Altimeter Information for Rectified Searches in Image Target Tracking for Drone Applications. In *XVI Workshop on Information Processing and Control (RPIC)* (pp. 1–5). Córdoba (Argentina). doi:10.1109/RPIC.2015.7497177.

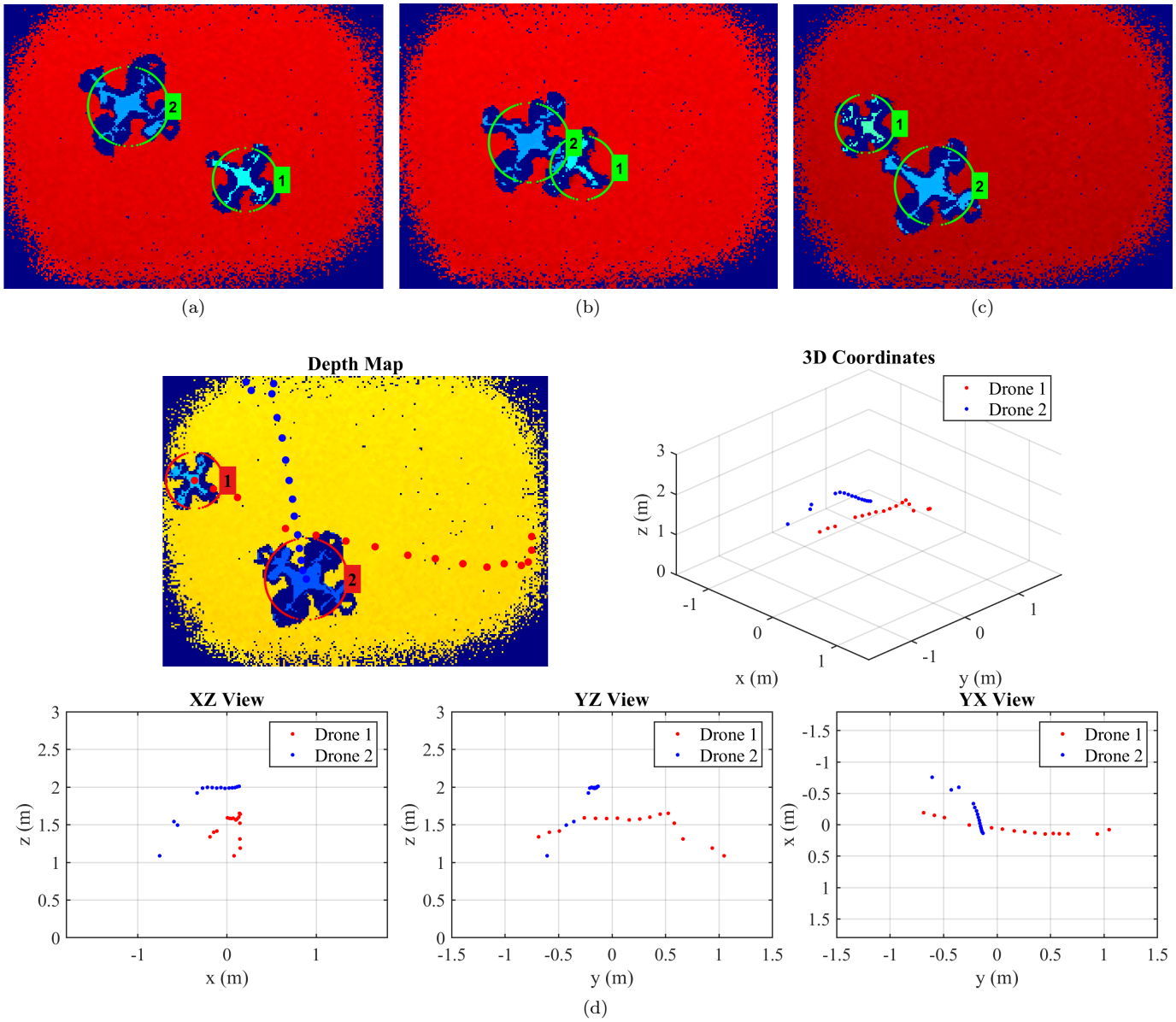


Figure 12: Situation in which two drones cross each other at different altitudes: (a) before (b) during and (c) after the crossing. Finally, (d) shows the complete performance of location and tracking.

DeDrone (2019). RF3000. URL: `https://www.dedrone.com/products/hardware/rf-sensors/overview` Accessed 20 June 2019.

DJI (2019). Phamton 4. URL: `https://www.dji.com/es/phantom-4-pro-v2/info#specs` Accessed 20 June 2019.

Drozdowicz, J., Wielgo, M., Samczynski, P., Kulpa, K., Krzonkalla, J., Mordzonek, M., Bryl, M., & Jakielaszek, Z. (2016). 35 GHz FMCW Drone Detection System. In *17th International Radar Symposium (IRS)* (pp. 1–4). Krakow (Poland): IEEE. doi:`10.1109/IRS.2016.7497351`.

Engel, J., Sturm, J., & Cremers, D. (2012). Camera-Based Navigation of a Low-Cost Quadrocopter. In *IEEE/RSJ International Conference onIntelligent Robots and Systems* (pp. 2815–2821). Vilamoura (Portugal): IEEE. doi:`10.1109/IROS.2012.6385458`.

ErleRobotics (2019). ErleBrain. URL: `http://docs.erlerobotics.com/brains/erle-brain-3/hardware/sensing#erle-gps` Accessed 20 June 2019.

Gokturk, S. B., & Tomasi, C. (2004). 3d Head Tracking Based on Recognition and Interpolation Using a Time-Of-Flight Depth Sensor. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. Washington (United States): IEEE volume 2. doi:`10.1109/CVPR.2004.1315166`.

Hussmann, S., Edeler, T., & Hermanski, A. (2012). Real-Time Processing of 3D-TOF Data in Machine Vision Applications. In F. Solari, M. Chessa, & S. P. Sabatini (Eds.), *Machine Vision* chapter 4. IntechOpen. doi:`10.5772/34160`.

Iacono, M., & Sgorbissa, A. (2018). Path Following and Obstacle Avoidance for an Autonomous Uav Using a Depth Camera. *Robot. Auton. Syst.*, *106*, 38–46. doi:`10.1016/j.robot.2018.04.005`.

Kahlmann, T., Remondino, F., & Ingensand, H. (2006). Calibration for Increased Accuracy of the Range Imaging Camera Swissranger. In *ISPRS Commission V Symposium 'Image Engineering and Vision Metrology'* (pp. 136–141). Dresden (Germany).

Kendoul, F. (2012). Survey of Advances in Guidance, Navigation, and Control of Unmanned Rotorcraft Systems. *J. Field Robot.*, *29*, 315–378. doi:`10.1002/rob.20414`.

Kim, J., Kang, M.-S., & Park, S. (2010). Accurate Modeling and Robust Hovering Control for a Quad–rotor VTOL Aircraft. *J. Intell. Robot. Syst.*, *57*, 9–26. doi:10.1007/s10846-009-9369-z.

Klare, J., Biallawons, O., & Cerutti-Maori, D. (2017). UAV Detection with MIMO Radar. In *18th International Radar Symposium (IRS 2017)* (pp. 1–8). Prague (Czech Republic). doi:10.23919/IRS.2017.8008140.

Kohoutek, T. K., Mautz, R., & Donaubauer, A. (2010). Real-Time Indoor Positioning Using Range Imaging Sensors. In *SPIE - Real-Time Image and Video Processing 2010* (p. 8). Brussels (Belgium) volume 7724. doi:10.1117/12.853688.

Kong, W., Zhou, D., Zhang, D., & Zhang, J. (2014). Vision-Based Autonomous Landing System for Unmanned Aerial Vehicle: A Survey. In *International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI)* (pp. 1–8). Beijing (China): IEEE. doi:10.1109/MFI.2014.6997750.

Lange, R. (2000). *3D Time-Of-Flight Distance Measurement with Custom Solid-State Image Sensors in CMOS/CCD-Technology*. Ph.D. thesis Department of Electrical Engineering and Computer Science at University of Siegen. URL: http://www.ub.uni-siegen.de/pub/diss/fb12/2000/lange/lange.pdf.

Langmann, B. (2013). *Wide Area 2D/3D Imaging. Development, Analysis and Applications*. Springer Vieweg. URL: https://www.springer.com/it/book/9783658064563.

Liu, H., Wei, Z., Chen, Y., Pan, J., Lin, L., & Ren, Y. (2017). Drone Detection Based on an Audio-Assisted Camera Array. In *IEEE Third International Conference on Multimedia Big Data (BigMM)* (pp. 402–406). IEEE. doi:10.1109/BigMM.2017.57.

Luna, C. A., Losada-Gutiérrez, C., Fuentes-Jiménez, D., Fernández-Rincón, Á., Mazo, M., & Macías-Guarasa, J. (2017). Robust People Detection Using Depth Information from an Overhead Time-Of-Flight Camera. *Expert Syst. Appl.*, *71*, 240–256. doi:10.1016/j.eswa.2016.11.019.

Merino, L., Wiklund, J., Caballero, F., Moe, A., De-Dios, J. R. M., Forssen, P. E., Nordberg, K., & Ollero, A. (2006). Vision-Based Multi-UAV Position Estimation. *IEEE Robot. Autom. Mag.*, *13*, 53–62. doi:10.1109/MRA.2006.1678139.

Multerer, T., Ganis, A., Prechtel, U., Miralles, E., Meusling, A., Mietzner, J., Vossiek, M., Loghi, M., & Ziegler, V. (2017). Low-Cost Jamming System Against Small Drones Using a 3D Mimo Radar Based Tracking. In *European Radar Conference (EURAD)* (pp. 299–302). Paris (France): European Microwave Association. doi:10.23919/EURAD.2017.8249206.

Munkres, J. (1957). Algorithms for the Assignment and Transportation Problems. *J. Soc. Ind. Appl. Math.*, *5*, 32–38. doi:10.1137/0105003.

Nguyen, P., Ravindranatha, M., Nguyen, A., Han, R., & Vu, T. (2016). Investigating Cost-Effective RF-Based Detection of Drones. In *DroNet'16* (pp. 17–22). Singapore (Singapore). doi:10.1145/2935620.2935632.

Ocean Optics (2019). Miniature Spectrometer. URL: https://oceanoptics.com/product/sts-nir-microspectrometer/ Accessed 20 June 2019.

Paredes, J. A., Álvarez, F. J., Aguilera, T., & Villadangos, J. M. (2017). 3D Indoor Positioning of UAVs with Spread Spectrum Ultrasound and Time-Of-Flight Cameras. *Sens. MDPI*, *18*, 89. doi:10.3390/s18010089.

Parrot (2019). Bebop 2. URL: https://www.parrot.com/es/drones/parrot-bebop-2-power-pack-fpv#explorez-toujours-plus-loin Accessed 20 June 2019.

Pirker, K., Ruther, M., Bischof, H., Schweighofer, G., & Mayer, H. (2010). An Omnidirectional Time-Of-Flight Camera and Its Application to Indoor Slam. In *11th International Conference on Control Automation Robotics & Vision (ICARCV 2010)* (pp. 988–993). Singapore (Singapore): IEEE. doi:10.1109/ICARCV.2010.5707286.

Pizzo, L. D., Foggia, P., Greco, A., Percannella, G., & Vento, M. (2016). Counting people by RGB or depth overhead cameras. *Pattern Recognit. Lett.*, *81*, 41–50. doi:10.1016/j.patrec.2016.05.033.

Samland, F., Fruth, J., Hildebrandt, M., Hoppe, T., & Dittmann, J. (2012). AR.Drone: Security Threat Analysis and Exemplary Attack to Track Persons. In J. Roning, & D. P. Casasent (Eds.), *SPIE - Intelligent Robots and Computer Vision XXIX: Algorithms and Techniques*. Burlingame (United States): Spie-Int Soc Optical Engineering volume 8301. doi:10.1117/12.902990.

Santos, N. P., Lobo, V., & Bernardino, A. (2015). A Ground-Based Vision System for UAV Tracking. In *OCEANS 2015* (pp. 1–9). Genova (Italy): IEEE. doi:10.1109/OCEANS-Genova.2015.7271349.

Stahlschmidt, C., Gavriilidis, A., Velten, J., & Kummert, A. (2013). People Detection and Tracking from a Top-View Position Using a Time-Of-Flight Camera. In *6th International Conference on Multimedia Communications, Services & Security (MCSS 2013)* (pp. 213–223). Krakow (Poland). doi:10.1007/978-3-642-38559-9_19.

Stahlschmidt, C., Gavriilidis, A., Velten, J., & Kummert, A. (2016). Applications for a People Detection and Tracking Algorithm Using a Time-Of-Flight Camera. *Multimed. Tools Appl.*, *75*, 10769–10786. doi:10.1007/s11042-014-2260-3.

Syma (2019). Syma X15W. URL: http://www.symatoys.com/goodshow/x15w-syma-x15w-everything-in-the-world-is-captured.html Accessed 20 June 2019.

Tahar, K. N., & Kamarudin, S. S. (2016). UAV Onboard GPS in Positioning Determination. *ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, *XLI-B1*, 1037–1042. doi:10.5194/isprs-archives-XLI-B1-1037-2016.

Teuliere, C., Eck, L., Marchand, E., & Guenard, N. (2010). 3D Model-Based Tracking for UAV Position Control. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1084–1089). IEEE. doi:10.1109/IROS.2010.5649700.

Weiwei, K., Daibing, Z., Shulong, Z., Dianle, Z., Boxin, Z., Zhiwei, Z., Zhaowei, M., Dengqing, T., & Jianwei, Z. (2015). Autonomous Track and Land a MAV Using a Modified Tracking-Learning-Detection Framework. In *34th Chinese Control Conference (CCC)* (pp. 5359–5366). doi:10.1109/ChiCC.2015.7260477.