

Received March 15, 2022, accepted April 19, 2022, date of publication April 25, 2022, date of current version May 3, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3170481

Extracting Unambiguous Drone Signature Using High-Speed Camera

FRANK BILLY DJUPKEP DIZEU¹, MICHEL PICARD¹, MARC-ANTOINE DROUIN¹,
AND GUILLAUME GAGNÉ²

¹National Research Council Canada, Ottawa, ON K1A 0R6, Canada

²Defence Research and Development Canada, Quebec City, QC G3J 1X5, Canada

Corresponding author: Frank Billy Djupkep Dizeu (dizeubilly@yahoo.fr)

ABSTRACT In recent years, the use of drones for recreational and commercial activities has grown rapidly due to their affordability and performance. This growing use raises concerns about the threats drones pose to the security of sensitive areas such as airports, prisons, industrial and military facilities. In response to these threats, drones detection methods are being actively developed. In particular, most camera-based methods rely on appearance to perform detection. They are therefore prone to error due to the great similarity between drones and some other flying entities such as birds. However, from a kinematic perspective, unlike birds, drones, especially multicopters, have a propeller rotation speed. The method proposed in this paper uses the propeller rotation speed as the key physical parameter on which to rely to unambiguously distinguish drones from other flying entities. The basic idea consists in using discrete Fourier transform to determine the propellers rotation speed from high frame rate videos, and extracting the propellers induced drone signature as a quantitative camera-based drone signature. The proposed algorithm proceeds as follows: flying entities are continuously tracked in the sky; discrete Fourier transform, applied to the video stream within a time window ending at the current instant (frame), is used to extract the propellers induced drone signature which unambiguously confirm each flying entity as being a drone or not. Experimental results obtained using a consumer-grade camera at a frame rate of 240Hz demonstrate the effectiveness and reliability of the proposed method.

INDEX TERMS Drone detection, discrete Fourier transform, high speed camera, propellers induced drone signature, propeller rotation speed, unambiguous drone signature.

I. INTRODUCTION

Drones (unmanned aerial vehicles) are aircraft that can be steered non-autonomously by a ground pilot through radio frequency exchanges, or autonomously by closed loop computer systems. Over the past decade, due to the emergence of companies commercializing different types of affordable drones (multicopters) with increasing performance, there has been an exponential growth in the popularity of drones for recreational purposes and commercial activities, including aerial videos shooting, surveying, cartography, monitoring of public space (for example to enforce social distancing during the COVID-19 pandemic), search and rescue, delivery of goods and medicine, etc. [1]. Experts agree on continued growth in the size of the drone market and forecast five

hundred billion dollars in revenue by 2028 [2]. This will represent tens of millions of drone owners around the world. Unfortunately, this growing use of drones (in particular for recreational activities) raises concerns about privacy, but also security of sensitive areas such as airports, prisons, and industrial and military facilities [3]–[5]. Although in many countries bills are passed and legislation is updated to regulate drone activities [1], [5], [6], it is very likely that an untrained pilot does not know the legislation, and a motivated criminal simply does not care. Illegal/criminal drone activities recently reported include the crash of a drone in front of the white house lawn in january 2015 [7], a collision between a commercial airplane and a drone at the Jean Lesage international airport at Quebec city, Canada in 2017 [8], a mysterious presence of several drones for several days around a nuclear power plant in France [9], the use of drones to bomb a Ukrainian army weapons warehouse [10], an attempt to use

The associate editor coordinating the review of this manuscript and approving it for publication was Sudipta Roy¹.

drones to drop items at prisons [11], for drug smuggling [12], for illegal phones traffic [13], etc. [3]. In the coming years, drones will become a predominant source of intentional and unintentional threats [14]. Therefore, counter-measures are required against illegal and criminal drones activities; one of them is the development of drone detection systems, which is increasingly gaining the attention of the research community, both in academia and in industry.

From a kinematic point of view, flying entities are characterized by their moving speed which allows them to move from one point to another, and their acceleration which allows them to modify their moving direction and speed. Almost all flying entities share the same range of moving speed and acceleration (e.g., we can find birds and drones moving at $3m/s$ with a linear trajectory), making it unreliable to rely on these kinematic parameters to differentiate them. However, in addition to these common kinematic parameters, multicopters (drones) are also characterized by their propeller rotation speed. Therefore, for at least three reasons, it can be very appealing to rely on the propeller rotation speed to differentiate drones from other flying entities. First, the propeller rotation speed has a lower bound different from zero; this means that even when the drone is hovering (moving speed is equal zero), the propeller rotation speed is not null. Second, high speed cameras can be used to capture the fast propeller rotation. By high speed camera, it is meant a camera with a frame rate sufficiently high to capture the propeller rotation in the sense of the sampling theory. Third, the propeller rotation speed can be determined from a high frame rate video capturing the blades in rotation.

In this paper, we use the propeller rotation speed as the key physical parameter on which to rely to unambiguously distinguish drones from other flying entities. The basic idea consists in using discrete Fourier transform to determine the propellers rotation speed from high frame rate videos, and extracting the propeller induced drone signature as an unambiguous quantitative camera-based drone signature. The proposed algorithm proceeds as follows: a steady high speed camera observes the sky and flying entities are detected. These entities are continuously tracked over time and tracking results are stacked to build a stabilized high frame rate video ending at the current frame; discrete Fourier transform, performed pixel per pixel over the entire video sequence, is used to extract the propeller induced drone signature which confirm each flying entity as being a drone or not. The contributions of the paper can be summarized as follows:

- We demonstrate the potential of high speed cameras for the classification of flying entities using their kinematic characteristics.
- We demonstrate the existence of a unique propeller fingerprint related to the propeller rotation speed.
- We introduce the propellers induced drone signature as an unambiguous quantitative camera-based drone signature.
- We propose an efficient algorithm for drone detection in high frame rate videos.

The remaining of the paper is organized as follows. In Section II, we give an overview of the state of the art on drone detection. We elaborate on the propeller fingerprint in Section III. In Section IV, the propellers induced drone signature is introduced and our proposed algorithm for drone detection is presented. Section V addresses the tracking of flying entities. This tracking is necessary to compensate for the motion of these entities in the video supplied as input to the proposed algorithm. The experimental setup and the processing strategy are presented in Section VI followed by the results and discussions in Section VII. We end the paper with a conclusion in Section VIII.

II. PREVIOUS WORKS

So far, various modalities, including radar, audio, radio frequency (RF), camera, have been proposed for detection, tracking, classification for eventual neutralization of drones [15]. Multimodal approaches combining two or more of these individual modalities have also been proposed [16]. Moreover, there is a strong trend towards drone detection systems using machine/deep learning [17], [18].

A. ACOUSTIC-BASED DRONE DETECTION METHODS

The basic principle in acoustic-based drone detection systems is the recognition of the audio signature of the spinning propellers in the ambient noise recorded with a microphone. Feasibility of acoustic-based drone detection using hidden Markov model has been demonstrated [19]. Correlation techniques have also been used, where pre-recorded audio fingerprints of drones are identified in the recorded ambient noise [20]. This requires an audio fingerprint for each existing drone model. Machine/deep learning classifiers are also reported and rely on different architectures, including PIL (plotted image machine learning) and KNN (k-nearest neighbors) [21], multi class SVM (support vector machines) [22], and CNN (convolutional neural network) [23]. An interesting aspect of acoustic-based drone detection systems is their ability to operate day and night. They can alert to the presence of nearby drones, but cannot be used for tracking purposes. Moreover, they are not robust to intentional alteration or camouflage of the drone's acoustic fingerprint, and to noise resulting, for example, from the proximity of an urban area.

B. RADAR-BASED DRONE DETECTION METHODS

Traditional radar sensors deliver information about the distance, the size (radar cross section) and the speed of objects in an active manner, that is, by sending electromagnetic pulses in a given direction in space and analyzing the electromagnetic energy reflected from potential target objects. One strength of radar systems is their ability to perform long range detection, even under unfavorable light and weather conditions [24]. However conventional radar systems are not optimized for detecting small drones moving slowly and flying at low altitude. This led to the development of millimeter wave frequencies systems which can provide better radar cross section resolution depending on the material constituting the

drone [25]. A survey on radar-based drone detection reveals three groups of methods [26]. Methods of the first group aim to understand the radar signatures of drones produced in the micro-Doppler domain ([27]) and characterize the sensed radar cross section in order to set suitable thresholds and detection ranges [28], [29]. The second group includes methods using physics-based criteria ([30], [31]) or neural networks ([32], [33]) for drone detection and classification. The last group is formed by passive radar systems which are less expensive than active radar systems [34], [35].

C. RADIO FREQUENCY-BASED DRONE DETECTION METHODS

Radio frequency (RF)-based systems detect drones by monitoring the radio frequency data exchange between these drones and their controllers [36]. They are transportable and can achieve long-range detection and tracking, making them the most popular anti-drone systems on the market. They can ultimately be designed to locate the drone pilots [37]. Different features have been considered for training a machine learning architecture in RF-based drone detection systems. A statistical analysis of the WiFi fingerprint is performed in [38]. In [36], RF signatures of the body shifting of drone caused by the spinning propellers and that of the body vibration due to environmental factors (wind for example) are exploited. In [39], raw RF signals are converted into frames in the wavelet domain and used as features for the training. Hierarchical learning is used in [40], whereas, in [41], the repetitive synchronization packets in video traffic between drones and controllers are used as features to train a random forest model. Compared to radar systems, radio frequency systems are energy efficient since they use passive RF sensors. Compared to acoustic systems, they are more robust to environment noise due to the strength of RF signals received. The problem with this drone detection modality is that they require radio frequency exchange between the drone and its controller; conceptually, they cannot detect drones pre-programmed for autonomous flight.

D. CAMERA-BASED DRONE DETECTION METHODS

Camera-based drone detection systems include vision-based systems (RGB cameras), thermal-based systems (infrared cameras), and event-based systems (neuromorphic cameras). Vision-based drone detection is being attracting attention due to its good balance between price and detection capability [42], and also because it can provide additional visual informations (drone model and color, dimensions, payload) for easy human interpretation [43]. Unlike radar-based systems, with which they share the need of a line of sight, vision-based systems are passive. Unlike RF-based systems, they can detect autonomous drones. Most vision-based drone detection systems rely on features extraction or deep learning [18]. Features-based approaches use morphological operators/descriptors to extract relevant features which are then used by a classifier [44], [45]. Deep learning approaches exploit various neural network architectures including

CNN [45], Faster region based CNN [46], YOLO (You only look once) [47], [48], etc. [17], [18]. Motion of both drones and camera has also been addressed. In [49], frame difference is used to detect moving objects (flying entities) which are then classified as drones or not. In [50], regression is used for motion stabilization followed by CNN classification. In [51], [52], moving cameras are used for drone detection in the context of drone cooperation, multi-drone autonomous navigation and collision avoidance. Accuracy of vision-based drone detection methods typically decreases with the contrast between the drone and background. It is particularly the case for long range detection where the drone is represented by few pixels and is similar in appearance, shape and size to birds. Some works investigated these scenarios using a multi-camera strategy (one steady camera with a large field of view is used to detect intruders [53], one moving camera with a small field of view follows each intruder to provide high resolution tracking result to the classifier [54]), or deep learning [55]. RGB-D systems, based on either time of flight or stereo vision, have also been proposed for segmenting drones from background using 3D data (depth) [56], [57].

Vision-based systems perform poorly in limited visibility conditions (night, dust, cloud, rain, snow or fog). For such scenarios, thermal cameras can be considered [58], [59]. However, it is very likely that the thermal signature of the drone is degraded by its constituting materials (plastic, carbon fiber), as well as by the thermal shielding of its electric motors. Moreover, for similar specifications, thermal cameras are more expensive than RGB cameras. Systems utilizing near infrared or short wave infrared cameras for drone detection (at night) are also reported [60], [61]. Other devices that seem promising are neuromorphic cameras which captured the rapid changes in intensity, mainly related to motion, occurring in their field of views. In a recent paper, these cameras were used to detect drones by the frequency signature of their propellers [62]. Unfortunately, neuromorphic cameras have, at the moment, a small spatial resolution.

E. MULTIMODAL DRONE DETECTION METHODS

Observing that no drone detection modality is perfect, some authors have proposed multimodal approaches to take advantage of the strengths and reduce the weaknesses of the individual modalities involved. Some multimodal drone detection systems reported in literature include the association of radar and audio sensors [63], a system constituted by a camera array with audio recording [64], the association of infrared and RGB cameras [61], a system combining a radar, a microphone array, and a RGB camera [16], and a system combining LIDAR and cameras (RGB and infrared) [65].

III. PROPELLER FINGERPRINT RELATED TO THE ROTATION SPEED

In this section, we first show that a camera can be used to measure the propeller rotation speed. Then, we rely on a case study to introduce the basic idea of our proposed method:

a unique propeller fingerprint exists, is related to the propeller rotation speed, and can be determined using a camera.

A. MEASURING THE PROPELLER ROTATION SPEED USING A CAMERA

Let us consider a propeller having N_b blades and performing V_P rotation per minute (*rpm*). Let us also consider a camera with frame rate f_c capturing the rotating propeller in a video $\{I(p, t_m)\}_{p=1}^{p=n_p}, m = 1, 2, \dots, n_f$ having n_f frames and n_p pixels per frame. $I(p, t_m)$ is the intensity at pixel $p = 1, 2, \dots, n_p$ in the frame $m = 1, 2, \dots, n_f$ which is captured at time $t_m = (m - 1)/f_c$. We therefore represent the discrete temporal intensity signal extracted at pixel p as $\{I(p, t_m)\}_{m=1}^{m=n_f}$. In the recorded video, if pixel p belongs to an area covered by a blade in rotation, discrete Fourier transform (DFT) of intensity signal $\{I(p, t_m)\}_{m=1}^{m=n_f}$ can be used to determine the propeller rotation speed V_P provided that the Shannon-Nyquist sampling theorem is satisfied, i.e., for:

$$f_c > 2N_b V_P \quad (1)$$

Indeed, the magnitude of the DFT of intensity signal $\{I(p, t_m)\}_{m=1}^{m=n_f}$ will show a peak at the fundamental frequency f_0 given by

$$f_0 = N_b V_P \quad (2)$$

Peaks will also be present at certain harmonic frequencies f_i defined as follows:

$$f_i = if_0 \quad (3)$$

where i , the harmonic order, is a positive integer.

B. PROPELLER FINGERPRINT

Let's proceed with a case study. We consider a propeller made of two blades as illustrated in Fig. 1a and we impose a rotation speed of $66rpm$. Pixels p_1 and p_2 belong to the region covered by the blades during their motion, whereas pixel p_3 belongs to the region not covered by the blades. The following camera frame rates are successively considered: $30Hz$, $12Hz$ and $3Hz$. Only the first two camera frame rates satisfy the Shannon-Nyquist condition (1). Fig. 1b-d show intensity signals extracted at pixels p_1, p_2 and p_3 for frame rates $30Hz$, $12Hz$ and $3Hz$ respectively. In all cases, signals have the same number of samples; this result in a longer acquisition time as the frame rate decreases. The corresponding DFT magnitudes are shown in Fig. 1e-g. We will refer to the frequency (abscissa) axis as the pseudo rotation speed \tilde{V}_P axis to highlight the fact that this axis is related to the propeller rotation speed. When the Shannon-Nyquist condition (1) is satisfied, the pseudo rotation speed \tilde{V}_P is proportional to the rotation speed V_P ($\tilde{V}_P = N_b V_P$). The fundamental frequency $2.2Hz$ ($\tilde{V}_P = 132rpm$) is obtained as the position of the highest peak for frame rates $30Hz$ and $12Hz$. It is used to compute the rotation speed: (2) yields $V_P = 66rpm$. For the frame rate $3Hz$, it is not possible to determine V_P because

the Shannon-Nyquist is not satisfied. However, regardless of the frame rate, we note a great similarity between magnitude spectra corresponding to pixels p_1 and p_2 which belong to the region covered by the blades. This is confirmed in Fig. 1h-j which show the position of the highest peaks in Fig. 1e-g.

This set of peak positions, identical for all pixels covered by the blades in motion, forms the unique propeller fingerprint. We can confirm this by extracting the peak positions in the DFT magnitudes for all pixels along the yellow line in Fig. 1a. The results are shown in Fig. 1k-m for the considered frame rates. Each row in Fig. 1k-m indicates the peak positions obtained for a pixel on the yellow line in Fig. 1a. It is confirmed that pixels covered by the blades have most of their main peaks at the same positions (same pseudo rotation speed) which are the vertical lines indicated by the yellow arrows in Fig. 1k-m. DFT magnitudes for pixels not covered by the blades does not show any peak (black areas on the top and bottom in Fig. 1k-m). These qualitative analysis lead to three major conclusions: (1) It is possible to retrieve the unique propeller fingerprint, closely related to the propeller rotation speed, by computing the DFT magnitude of intensity signals obtained by capturing the propeller rotation using a camera with a given frame rate. (2) A high sampling rate is preferable to reduce the data acquisition time. (3) If the frame rate of the camera satisfies the shannon-Nyquist criterion, the propeller rotation speed can also be determined.

Quantitatively, we can chose the position of the highest peak to represent the propeller fingerprint. The result is shown in Fig. 1n-p for frame rates $30Hz$, $12Hz$ and $3Hz$ respectively. Therefore, in this case study, the propeller fingerprint is quantitatively represented by the following pseudo rotation speed: $132rpm$ for frame rates $30Hz$ and $12Hz$, and $42rpm$ for frame rate $3Hz$. A similar analysis performed with a four-blades propeller led to the same conclusions (see Fig. 2).

IV. EXTRACTING THE PROPELLERS INDUCED DRONE SIGNATURE FROM HIGH FRAME RATE VIDEOS

Drones (multicopters) form a unique class of flying entities characterized by a unique range of propeller rotation speeds with a lower bound of some hundreds *rpm*. We rely on the pseudo propeller rotation speed \tilde{V}_P obtained using DFT to determine the unique fingerprint of each propeller of a drone. Together, these individual propeller fingerprints define the propellers induced drone signature (PIDS), an unambiguous camera-based drone signature. The PIDS discussed in this section is derived in three steps using a high frame rate video sequence. These steps are the static background subtraction, the peaks extraction and the voting consensus. At each step, criteria are used to classify pixels as belonging or not to a drone propeller. Thus, the number of pixels of interest decreases as one advances in the processing. The obtained PIDS represents the pixels most likely to belong to a propeller, or to be impacted by the rotation of the blades.

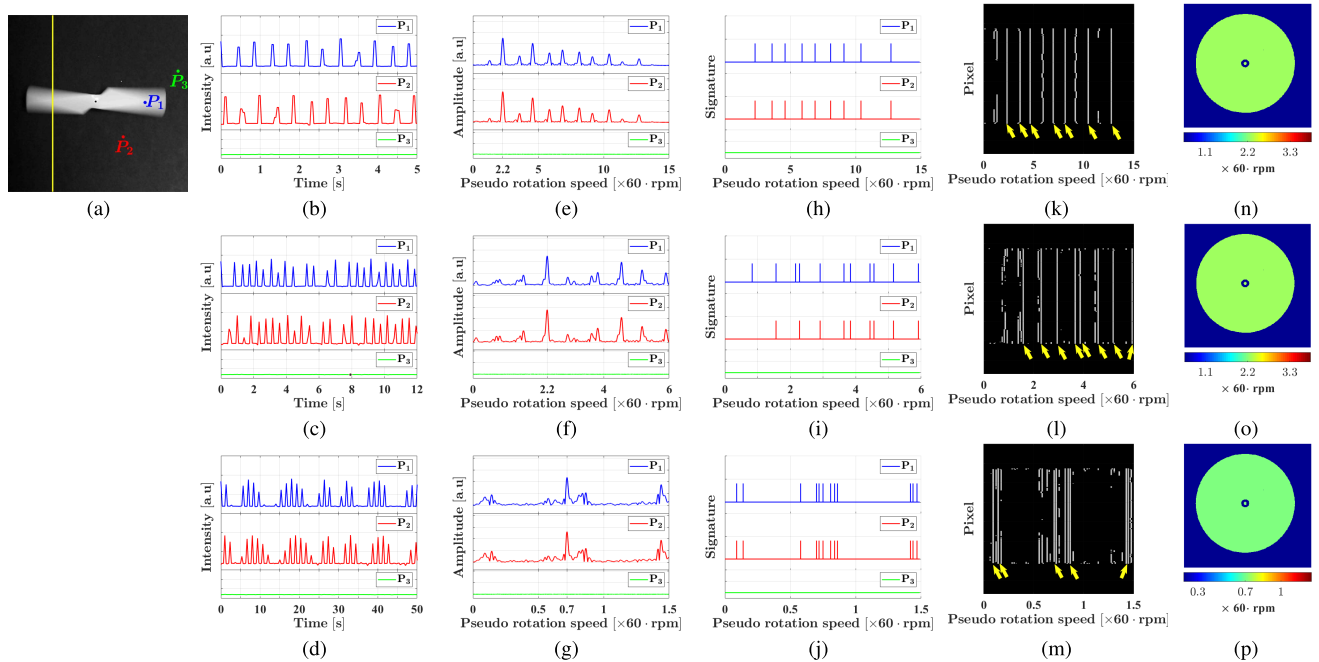


FIGURE 1. Propeller fingerprint: case of two blades. (a) Picture of the blades. The propeller rotation speed is $66rpm$. Pixels p_1 and p_2 belong to the region covered by the blades, whereas pixel p_3 belongs to the background. (b-d) intensity signals captured at pixels p_1 , p_2 and p_3 for frame rates $30Hz$, $12Hz$ and $3Hz$ respectively. Only the first two frame rates satisfy the Shannon-Nyquist theorem. (e-g) DFT magnitudes of the signals in (b-d). The fundamental frequency of $2.2Hz$ (pseudo rotation speed $\tilde{V}_p = 132rpm$) is obtained as the position of the highest peak for frame rates $30Hz$ and $12Hz$. Equation (2) yields $V_p = 66rpm$. For frame rate $3Hz$, it is not possible to determine V_p . (h-j) However, regardless of the frame rate, the highest peaks have the same position for pixels p_1 and p_2 belonging to the region covered by the blades during the rotation. This set of peak positions forms the propeller fingerprint. (k-m) Propeller fingerprint obtained for pixels along the yellow line in (a). Regardless of the frame rate, all pixels belonging to the region covered by the blades during the rotation show peaks at the same positions as indicated by the yellow arrows. (n-p) The position of the highest peaks can be used to quantify the propeller fingerprint: we have (n) $132rpm$ for frame rate $30Hz$, (o) $132rpm$ for frame rate $12Hz$ and (p) $42rpm$ for frame rate $3Hz$.

A. STATIC BACKGROUND SUBTRACTION

In this section, we present the straightforward static background subtraction approach used to extract the PIDS. The effectiveness of this simple approach demonstrates that the PIDS can easily be incorporated into appearance-based drone detection methods that may already include a more sophisticated background subtraction [66], [67].

We define the pseudo intensity $\{\tilde{I}(p, t_m)\}_{m=1}^{m=n_f}$:

$$\tilde{I}(p, t_m) = I(p, t_m) - \bar{I}(p, t_m) \tag{4}$$

where $\{\bar{I}(p, t_m)\}_{m=1}^{m=n_f}$ is the moving average of signal $\{I(p, t_m)\}_{m=1}^{m=n_f}$. The static background is formed by the pixels whose pseudo intensity does not significantly change throughout the video sequence. A pixel belongs to the static background if the following condition is satisfied:

$$\max_{1 \leq m \leq n_f} [\tilde{I}(p, t_m)] - \min_{1 \leq m \leq n_f} [\tilde{I}(p, t_m)] \leq \delta_I \tag{5}$$

Intensity threshold δ_I is chosen according to the sky conditions (blue, cloudy, rainy) and defines the minimum contrast expected between the drone and background. We provide an illustrative example with a video (frame rate of $240Hz$) showing a hovering quadcopter (four propellers) having $N_b = 2$ blades per propeller. Fig. 3d shows one frame extracted from the video. Pixel p_1 belongs to the area covered by a

blade in motion, pixel p_2 belongs to a part of the drone other than propellers, and pixel p_3 does not belong to the drone. Intensity signals measured at these pixels are presented in Fig. 3a, Fig. 3b and Fig. 3c respectively. Their respective pseudo intensity signals, obtained by subtracting the moving average, are shown in Fig. 3e, Fig. 3f and Fig. 3g respectively. For the chosen intensity threshold δ_I , pixel p_3 is classified as static background pixel. Further processing is irrelevant for pixels which, like pixel p_3 , belong to the static background.

We highlight the fact that we perform the static background subtraction by applying a threshold to the difference between the maximum and minimum pixel intensities over the entire video sequence (see Fig. 3h), and not the difference of consecutive frames as traditionally done [66]. Indeed, due to the rotation of the propellers, the pixels covered by the blades can receive background light for several consecutive frames; they can also receive the light reflected by the blades during several consecutive frames. In either cases, consecutive frame difference will yield a very small intensity value at these pixels, causing an error in the estimated static background. Although, in our study, taking the difference between the maximum and minimum pixel intensities over the entire video sequence improved the static background estimation, other approaches could also be considered for more robustness [67].

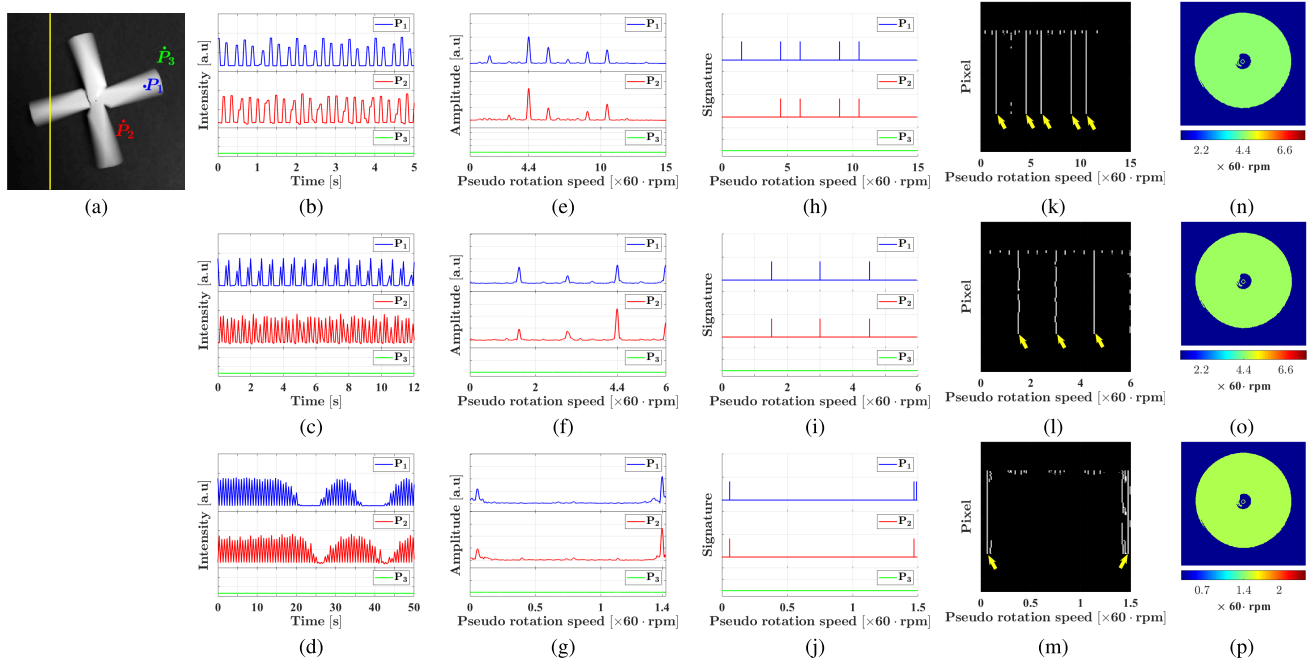


FIGURE 2. Propeller fingerprint: case of four blades. (a) Picture of the blades. The propeller rotation speed is $66rpm$. Pixels p_1 and p_2 belong to the region covered by the blades, whereas pixel p_3 belongs to the background. (b-d) intensity signals captured at pixels p_1, p_2 and p_3 for frame rates $30Hz, 12Hz$ and $3Hz$ respectively. Only the first two frame rates satisfy the Shannon-Nyquist theorem. (e-g) DFT magnitudes of the signals in (b-d). The fundamental frequency of $4.4Hz$ (pseudo rotation speed $\tilde{V}_p = 264rpm$) is obtained as the position of the highest peak for frame rates $30Hz$ and $12Hz$. Equation (2) yields $V_p = 66rpm$. For frame rate $3Hz$, it is not possible to determine V_p . (h-j) However, regardless of the frame rate, the highest peaks have the same position for pixels p_1 and p_2 belonging to the region covered by the blades during the rotation. This set of peak positions forms the propeller fingerprint. (k-m) Propeller fingerprint obtained for pixels along the yellow line in (a). Regardless of the frame rate, all pixels belonging to the region covered by the blades during the rotation show peaks at the same positions as indicated by the yellow arrows. (n-p) The position of the highest peaks can be used to quantify the propeller fingerprint: we have (n) $264rpm$ for frame rate $30Hz$, (o) $264rpm$ for frame rate $12Hz$ and (p) $84rpm$ for frame rate $3Hz$.

B. PEAKS EXTRACTION

We define \mathcal{F}_p as the normalized DFT magnitude of intensity signal $\{I(p, t_m)\}_{m=1}^{m=n_f}$ measured at pixel p (Fig. 3i-j) and $\mathcal{R}_{\tilde{V}} = [\tilde{V}_p^{min}, \tilde{V}_p^{max}]$ as the range of pseudo propeller rotation speed \tilde{V}_p (region delimited by the vertical brown lines in Fig. 3i and Fig. 3j). We remind that the lower bound of $\mathcal{R}_{\tilde{V}}$ is not equal to zero, i.e., $\tilde{V}_p^{min} > 0$.

We compute the mean amplitude \bar{a} of \mathcal{F}_p in interval $\mathcal{R}_{\tilde{V}}$ (Fig. 3k and Fig. 3l) and locate all the peaks, actually the local maximums, present in interval $\mathcal{R}_{\tilde{V}}$. We define A as the amplitude of the highest peak. User-defined coefficient $\delta_{\bar{a}} \geq 1$ is used to ensure that the highest peak is sufficiently high to be relevant; the intention is to avoid any pixel belonging to parts of the drone other than its propellers, belonging to other flying entities, or belonging to a moving background. Thus, a pixel is kept for the next steps if the amplitude A of its highest peak satisfies the following condition:

$$A \geq \bar{a}\delta_{\bar{a}} \tag{6}$$

For the example in Fig. 3, only pixel p_1 is kept for the next steps (Fig. 3k); pixel p_2 does not show a relevant peak although it belongs to the drone (Fig. 3l). For each pixel kept after previous steps, we extract the position of all peaks whose

amplitude A_{pk} satisfies the following condition:

$$A_{pk} \geq A\delta_a \tag{7}$$

where $\delta_a \in]0; 1]$ is a user-defined coefficient. We finally define \mathcal{P}_p as the group of peak positions satisfying (7) at pixel p . Fig. 3m indicates the peaks selected at pixel p_1 in the considered example. These peaks form the signature obtained as outlined in Fig. 3n.

C. VOTING CONSENSUS FOR RETRIEVING THE PIDS

In Section III, for a single propeller, we showed that the positions of the highest peaks in the DFT magnitude are identical for all pixels covered by the blades, and that this set of peak positions forms the propeller fingerprint. We also showed that one of these positions (the position of the highest peak for example) can be used as the quantitative unambiguous propeller fingerprint.

For a multicopter, it is very likely that propellers will have different fingerprints since they individually provide different amount of energy to support the motion (speed and direction) of the drone. Moreover, there will always be a great similarity between the DFT magnitudes of the pixels covered by the same blades, but, unlike the ideal case in section III, we will notice a drift in the peak positions. This drift is due to the drone body shifting and

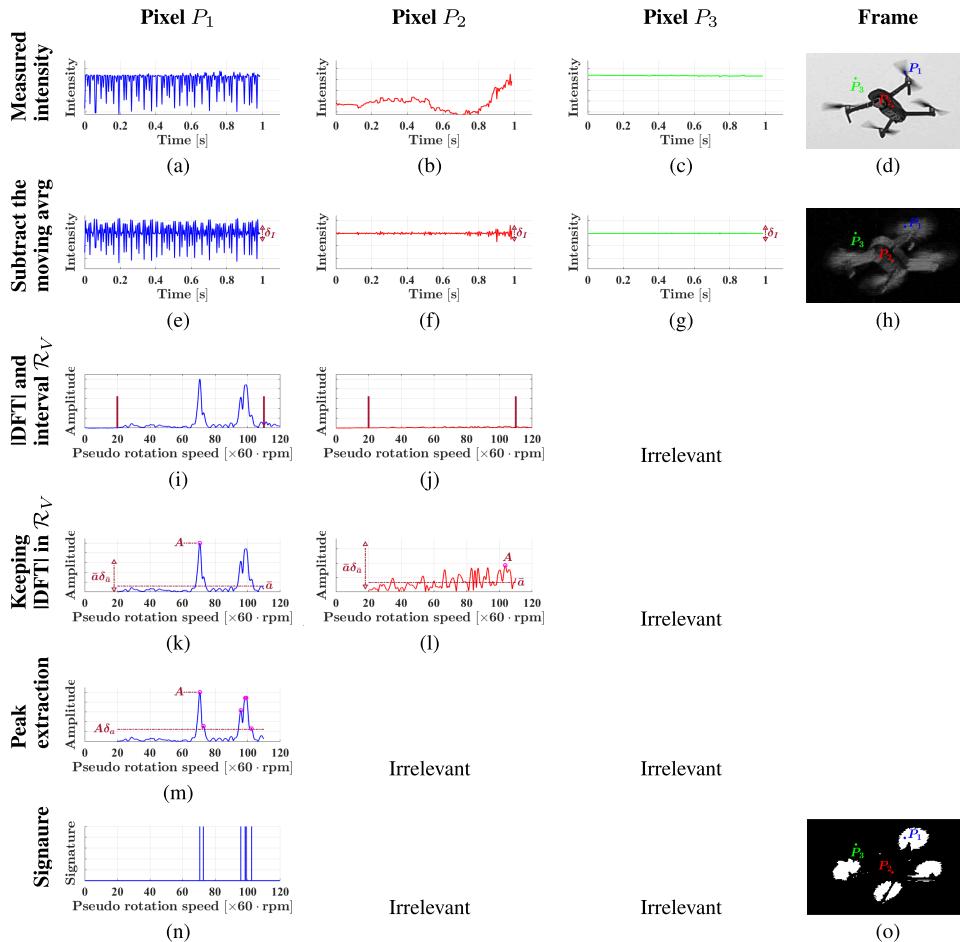


FIGURE 3. Processing steps to retrieve the propellers-induced drone signature (PIDS). (a-c) Measured intensity signals at pixels p_1 , p_2 and p_3 . (d) A frame extracted from the analyzed video. Pixel p_1 belongs to the area covered by the blades of a propeller. Pixel p_2 belongs to a part of the drone other than propellers. Pixel p_3 does not belong to the drone. (e-g) Pseudo intensity signals obtained after subtracting the moving average in intensity signals (a-c). (h) The difference between the maximum and minimum pixel intensities over the entire video sequence. Threshold δ_I is applied and only foreground pixels (e.g. pixels p_1 and p_2 (e-f)) are considered in the next steps. (i-j) DFT magnitudes of pseudo intensity signals in (e-f). Range \mathcal{R}_V of pseudo propeller rotation speeds is also represented and is delimited by the vertical brown lines. (k-l) We pursue the processing only with pixels for which (6) is satisfied. (m) All peaks satisfying (7) are extracted and form the signature in (n). A voting consensus is used to retrieve the propellers-induced drone signature (o).

vibrations caused by the drive of the rotating propellers and the wind. Parameter δ_d is used to take this drift into account. Let us consider the group $\mathcal{P} = \{\mathcal{P}_p\}_{p=1}^{p=n_p}$ formed by all the peak positions $\mathcal{P}_p, p = 1, 2, \dots, n_p$ determined as discussed in Section IV-B. We define \mathcal{L}_p as the label of pixel p . All labels are initialized to zero and background pixels are labeled as -1 . We also define $\mathcal{V}_i, i = 1, 2, \dots, n_V$ as the unique set of peak positions defined as follows:

$$\begin{cases} \mathcal{V}_i \in \mathcal{P} \\ \mathcal{V}_i \neq \mathcal{V}_j, \quad i \neq j \end{cases} \quad (8)$$

Algorithm 1 describes the voting consensus used to determine the propellers-induced drone signature, that is, to classify pixels as belonging or not to a region covered by the blades of the drone (see Fig. 3o).

Algorithm 1 Voting Consensus

- (1): Use (8) to extract $\mathcal{V}_i, i = 1, 2, \dots, n_V$ the unique set of peak positions in \mathcal{P} .
- (2): Use $\mathcal{V}_i, i = 1, 2, \dots, n_V$ as bins to compute the histogram of \mathcal{P} .
- (3): \mathcal{V} is the bin having the highest percentage of occurrences.
- (4): Use the drift parameter δ_d to define interval $\mathcal{R}_d = [\mathcal{V} - \delta_d; \mathcal{V} + \delta_d]$.
- (5): Assign the label \mathcal{V} (set $\mathcal{L}_p = \mathcal{V}$) to all pixels not yet labeled (i.e. for which the label is currently $\mathcal{L}_p = 0$) and for which we can find an element of \mathcal{P}_p belonging to \mathcal{R}_d .

D. ALGORITHM FOR DRONE DETECTION USING PIDS

Algorithm 2 summarizes the steps used to extract the PIDS from high frame rate videos. Input video (frame rate: f_c ,

Algorithm 2 Propellers Induced Drone Signature (PIDS)

- (1) *Inputs*: camera frame rate f_c ; range of propeller rotation speed $\mathcal{R}_{\tilde{v}}$; intensity threshold δ_I ; user-defined coefficients δ_a and $\delta_{\bar{a}}$; drift parameter δ_d ; high frame rate video $\{I(p, t_m)\}_{p=1}^{p=n_{pt}}$, $m = 1, 2, \dots, n_f$.
- (2) Initialize all pixel labels $\mathcal{L}_p, p = 1, 2, \dots, n_{pt}$ to zero.
- (3) *Peak extraction*.
- (a) Take intensity signal $\{I(p, t_m)\}_{m=1}^{m=n_f}$ of next pixel p and compute pseudo intensity signal (4).
- (b) if (5) is satisfied, set $\mathcal{L}_p = -1$ (label for background) and go to step (3a).
- (c) Compute \mathcal{F}_p , the normalized DFT magnitude of pseudo intensity signal (4).
- (d) Compute the mean amplitude \bar{a} of \mathcal{F}_p in interval $\mathcal{R}_{\tilde{v}}$.
- (e) Extract all peaks in interval $\mathcal{R}_{\tilde{v}}$ and find A, the amplitude of the highest peak.
- (f) if $A < \bar{a}\delta_{\bar{a}}$, set $\mathcal{L}_p = -1$ (label for background) and go to step (3a).
- (g) Build \mathcal{P}_p which contains the positions of the peaks whose amplitude is greater than $A\delta_a$ and go to step (3a).
- (4) *Voting consensus*.
- (a) Consider $\mathcal{P} = \{\mathcal{P}_p\}_{p=1}^{p=n_{pt}}$ formed by the peak positions \mathcal{P}_p of pixels not yet labeled (i.e. for which the label is currently $\mathcal{L}_p = 0$).
- (b) if $\exists \mathcal{P}$, Apply Algorithm 1 and go to step (4a).
- (5) *Output: detected drone*. The current video frame is segmented into two groups of pixels: background pixels ($\mathcal{L}_p = -1$) and PIDS ($\mathcal{L}_p > 0$).

n_{pt} pixel per frame, $n_{pt} \leq n_p$) represents a tracked and motion-stabilized flying entity (see Section V) and the PIDS unambiguously confirm this flying entity as being a drone or not.

The computational complexity of the proposed algorithm, including the tracking (see Section V), is given in Table 1. A standard (naive, academic) analysis is performed. For each step of the algorithm, we only report the additional time/space complexity. Except fast Fourier transform which has a linearithmic complexity in terms of the number of frame n_f , all other steps of the proposed algorithm has linear complexity with respect to both the number of pixels n_p and the number of frames n_f . Globally, for n_t tracked flying entities, each captured by n_{pt} ($n_{pt} \leq n_p$) pixels, the time complexity is $\mathcal{O}(\max(n_p n_f, n_t n_{pt} n_f \log n_f))$, whereas the space complexity is $\mathcal{O}(\max(n_p n_f, n_t n_{pt} n_f))$.

V. EVENT-BASED TRACKING AND VIDEO STABILIZATION

We remind that the primary data used in our proposed method is a video stream obtained using a high speed camera observing the sky from a fixed point. However, the algorithm presented in Section IV-D takes as input a high frame rate video capturing a stabilized (motion-compensated) flying entity. This means that the input video of algorithm presented in Section IV-D has all frames registered in the same local system coordinate linked to the flying entity. Such a video

TABLE 1. Computational complexity.

Steps	Time complexity	Space complexity	Comments
Tracking Section V	$\mathcal{O}(n_p n_f)$	$\mathcal{O}(n_p n_f)$	Video stream: n_f frames, each of n_p pixels.
Algo. 2			n_t flying entities tracked. $n_t \ll n_{pt}$.
(2)	$\mathcal{O}(n_t n_{pt})$	$\mathcal{O}(n_t n_{pt})$	
(3a)	$\mathcal{O}(n_t n_{pt} n_f)$	$\mathcal{O}(n_t n_{pt} n_f)$	
(3b)	$\mathcal{O}(n_t n_{pt} n_f)$	$\mathcal{O}(1)$	n_t stabilized input videos: n_f frames, each of n_{pt} pixels. $n_{pt} \leq n_p$.
(3c)	$\mathcal{O}(n_t n_{pt} n_f \log n_f)$	$\mathcal{O}(n_t n_{pt} n_f)$	
(3d)	$\mathcal{O}(n_t n_{pt} n_f)$	$\mathcal{O}(n_t n_{pt})$	
(3e)	$\mathcal{O}(n_t n_{pt} n_f)$	$\mathcal{O}(n_t n_{pt} n_{pk})$	
(3f)	$\mathcal{O}(n_t n_{pt})$	$\mathcal{O}(1)$	
(3g)	$\mathcal{O}(n_t n_{pt} n)$	$\mathcal{O}(1)$	n peaks extracted per pixel. $n \ll n_{pt}$
Algo. 1			
(1)	$\mathcal{O}(n_t n_{pt} n n_{\nu})$	$\mathcal{O}(n_t n_{\nu})$	
(2)	$\mathcal{O}(n_t n_{pt} n n_{\nu})$	$\mathcal{O}(n_t n_{\nu})$	$n_{\nu} \leq n_f$
(3-5)	$\mathcal{O}(n_t n_{pt})$	$\mathcal{O}(1)$	

is built from the primary video stream by tracking the flying entity continuously over time and stacking the frame-by-frame tracking results up to the current frame. We are aware that various tracking methods and image/video stabilization algorithms have been proposed in the literature [68], [69]. However, in the present study, we perform the tracking and the stabilization by utilizing the difference in pixel intensities between consecutive video frames. The resulting event-based approach is straightforward, has a low computation cost and does not require training data.

By event, we mean a rapid intensity change at a given pixel. Neuromorphic cameras can monitor this changes within very short periods of time ($\sim \mu s$) and thus can capture events (at each pixel) with high temporal resolution [70]. In the proposed approach, events are detected by comparing pixel intensities between consecutive frames in the high frame rate video stream. Thus we can capture events with a temporal resolution of $1/f_c$ using a camera with frame rate f_c . Event at a given pixel is determined by applying a threshold to the difference between pixel intensities of consecutive frames.

Let us consider two consecutive frames $\{I(p, t_m)\}_{p=1}^{p=n_p}$ and $\{I(p, t_{m-1})\}_{p=1}^{p=n_p}$ captured with the frame rate f_c . We define the event threshold δ_E as the minimum intensity change related to a motion occurring between these consecutive frames. The event $E(p, t_m)$ at pixel p in the m^{th} frame is determined as follows:

$$\begin{cases} E(p, t_m) = 1, & \text{if } |I(p, t_m) - I(p, t_{m-1})| > \delta_E \\ E(p, t_m) = 0, & \text{otherwise} \end{cases} \quad (9)$$

We are interested by events related to flying entities with strong kinematic parameters (moving speed, rotation speed). Thus δ_E must be sufficiently high to avoid capturing events related to flying entities moving slowly as well as moving backgrounds. The event image $\{E(p, t_m)\}_{p=1}^{p=n_p}$ corresponding to the m^{th} frame can be viewed as the result of the motion-based segmentation of frame $\{I(p, t_m)\}_{p=1}^{p=n_p}$. Flying entities can therefore be tracked and their motion can be compensated using only the temporal information at each pixel.

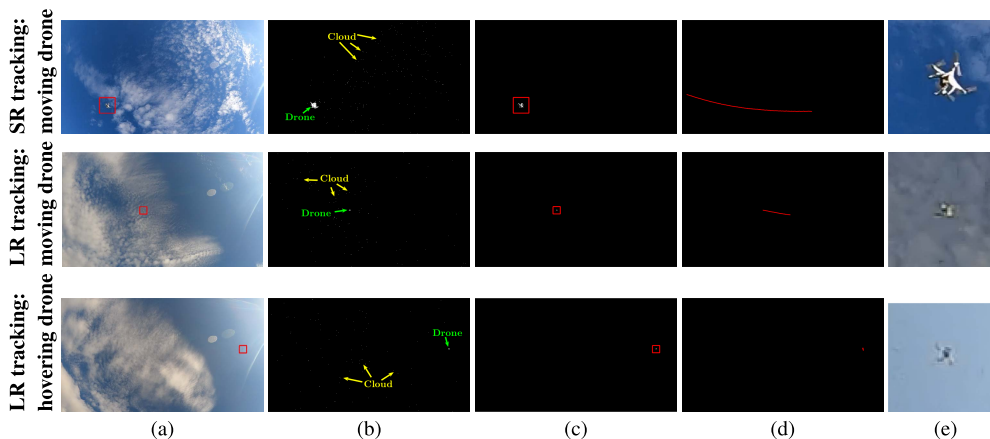


FIGURE 4. Event-based tracking of a flying entity (here a drone). First row: short range (SR) tracking where the drone flies near the camera. Second row: long range (LR) tracking where the drone flies far away from the camera. Third row: long range tracking of a hovering drone. (a) The most recent video frame is sent to the event detector. (b) Event images obtained for a low event threshold δ_{E_1} . Captured events include the moving cloud and the drone. (c) By increasing the event threshold to $\delta_{E_2} > \delta_{E_1}$, we limit the captured events to only those related to the drone. (d) Trajectory of the drone in the video sequence. Note that the trajectory of the hovering drone is a point. The tracking window is centered at the centroid of pixels belonging to the same cluster of events (see the red region of interest in (a) and (c)). (e) Tracking result for the frame in (a).

Thus, using the high frame rate video $\{I(p, t_m)\}_{p=1}^{p=n_p}, m = 1, 2, \dots, n_f$, it is possible to: (1) capture pixel events occurring between consecutive frames; (2) determine the trajectory of flying entities in the sky by aggregating events related to them in the entire video sequence; (3) perform a continuous tracking of moving entities in the sky; (4) compensate for the motion of these entities by stacking the tracking results.

We illustrate the event-based tracking of flying entities in Fig. 4 by considering three scenarios. The first scenario is the short range (SR) tracking where the flying entity is moving close to the camera and is therefore captured with a very good spatial resolution. The second scenario is the long range (LR) tracking where the flying entity moves far away from the camera and is captured with very few pixels (low spatial resolution). In the third scenario the flying entity is hovering far away (long range) from the camera. The current frame (Fig. 4a) is sent to the event detector. Fig. 4b shows the event image obtained using a low threshold δ_{E_1} . Events related to the motion of the flying entity are captured as well as event related to the cloud moving very slowly. Fig. 4c is obtained with a threshold $\delta_{E_2} > \delta_{E_1}$ and shows only events related to the flying entity. Local clusters are formed among pixels associated with an event and the centroid of each cluster is taken as the position of a flying entity in the current video frame. The trajectory of each flying entity is then obtained throughout the video sequence (Fig. 4d): first, the determined positions of flying entities in the previous frames are considered; second, each position in a frame is matched to at most one position in the previous frame and at most one position in the next frame; Third, the matching is performed so that the deviation between the paired positions in consecutive frames is minimal. The tracking result of a flying entity in a given frame is a region of interest centered at the determined centroid (see the red rectangle in Fig. 4a

and Fig. 4c). Fig. 4e shows the tracking result for the frame in Fig. 4a. By stacking these tracking results frame after frame, we build the stabilized video supplied as input to the proposed algorithm (Section IV-D). Note that, there will be as many stabilized videos as there are detected flying entities (centroids) in the current frame. The PIDS extracted from each video will confirm the corresponding flying entity as being a drone or not.

VI. DATA ACQUISITION AND PROCESSING

We used a GoPro Hero 6 black camera ([71]) to collect our data. It was mounted on a tripod and observed the sky from a fixed position as outlined in Fig. 5a. The camera had vertical, horizontal and diagonal field of views of 69.5° , 118.2° and 133.6° respectively. It was configured to capture RGB videos (8 bits, H265 compression) with a resolution of 1080×1920 and a frame rate of $f_c = 240Hz$.

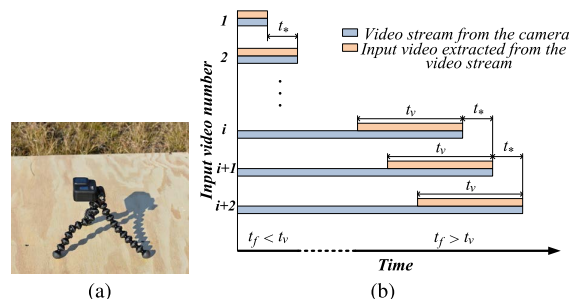


FIGURE 5. Data Collection and processing. (a) We used a GoPro hero 6 black camera mounted on a tripod and observing the sky from a fixed position. (b) In the proposed method, the input video sequence has a length t_v , ends at the current frame at time t_f and is extracted from the video stream provided by the camera. This input video sequence is updated with a periodicity $t_s \geq 1/f_c$.

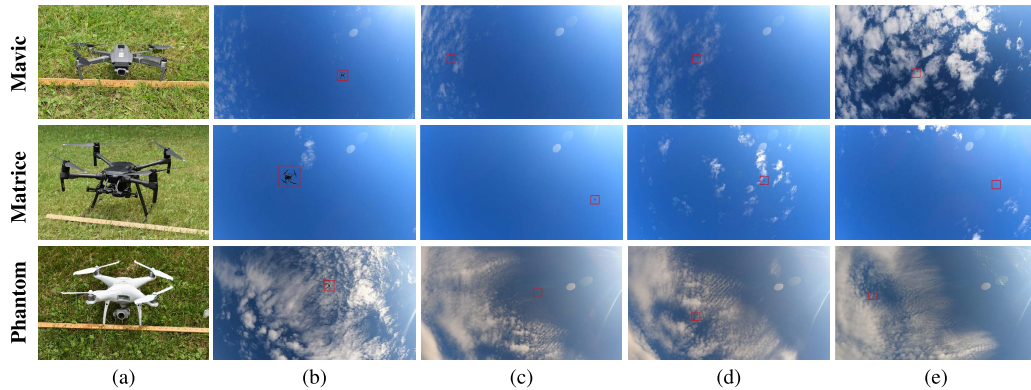


FIGURE 6. Data acquisition. (a) Drone models used: Mavic, Matrice and Phantom. (b-e) Frame extracted from the video sequences used. Drones fly at different altitudes and speeds against backgrounds ranging from blue sky to scattered clouds.

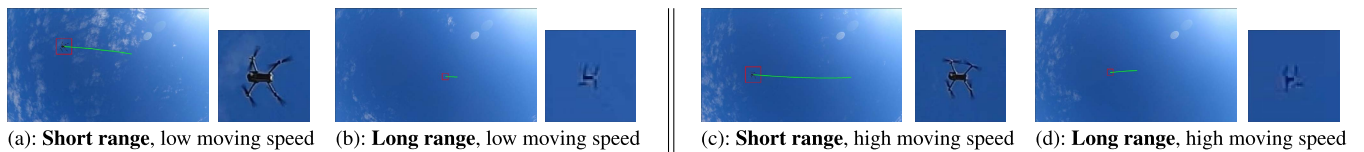


FIGURE 7. Mavic slow/fast in short/long range detection. Left: last frame of the input video sequence (length t_v) showing a detected flying entity (red) and its trajectory within the input video. Right: tracking result.

We started and stopped video recording remotely using the GoPro application installed on a phone. The recorded data was saved on the onboard SD card for offline post-processing. Three drones were used to collect the data, namely, the *Dji Mavic Pro* (Mavic), the *Dji Matrice 210* (Matrice), and the *Dji Phantom 4 Pro V2* (Phantom), all shown in Fig. 6a. The data consist in a collection of video sequences recorded during the flight of one of these drones over the camera. A flight scenario involved a drone moving slowly ($\sim 2m/s$) or quickly ($\sim 6m/s$) at a given altitude chosen such that the drone was represented by very few pixels (we will use "low resolution" to refer to this case) or a sufficient number of pixels (we will use "high resolution" to refer to this case) throughout the video sequence. Fig. 6b-e show some recorded video frames which include backgrounds changing from clear sky to scattered clouds. In some images, it is not easy to see the drone because of its small size and its low contrast against background.

Preliminary tests were carried out to ensure that the (consumer-grade) camera was indeed operating at the set frame rate. First, all automatic settings was disable and we manually adjusted all necessary camera parameters. In particular, we set the exposure time (or shutter speed) to a fixed value $1/f_c$, where f_c is the desired frame rate. Second, we recorded multiple videos of exactly t seconds (e.g. $t = 1s$) at frame rate f_c (e.g. $f_c = 240Hz$), and we verified that there were indeed $t \cdot f_c$ (e.g. 240 frames) frames in each recorded video. Third, we captured the propeller spinning very fast. By playing back the recorded video frame by

frame, we verified that the consecutive frames captured the blades at different positions, which means that an image was indeed captured every $1/f_c$ seconds. Fourth, the experiment described in section III-B was repeated several times under different lighting, including different background brightness and/or different ambient lighting, while maintaining the rotation speed at $66rpm$. We confirmed that the rotation speed obtained using the DFT (DFT assumes that the sampling frequency is constant) was equal to the actual rotation speed for all measurements, meaning that the assumption of constant sampling frequency is true.

We implemented the event-based tracking method (Section V) and the PIDS algorithm (Section IV-D) in Matlab R2018. The data were played back to emulate a continuous video stream at the frame rate $f_c = 240Hz$. A raw full frame video sequence of length t_v was extracted from the video stream between time $t_f - t_v$ and the current time t_f . This raw video sequence was updated with a periodicity $t_* \geq 1/f_c$ as illustrated in Fig. 5b. The first processing step was the event-based tracking of flying entities in each frame of the raw input video sequence (Section V). The output of this step was the stabilized videos, each capturing one flying entity detected. The second processing step was the extraction of the PIDS from the stabilized videos (algorithm in Section IV-D). The supplementary materials emulates the processing and shows the results obtained with the drones moving slowly in short-range / high-resolution detection (*SupMat_MATRICE_SR*, *SupMat_MAVIC_SR*, *SupMat_PHANTOM_SR*), and in long-range / low-resolution

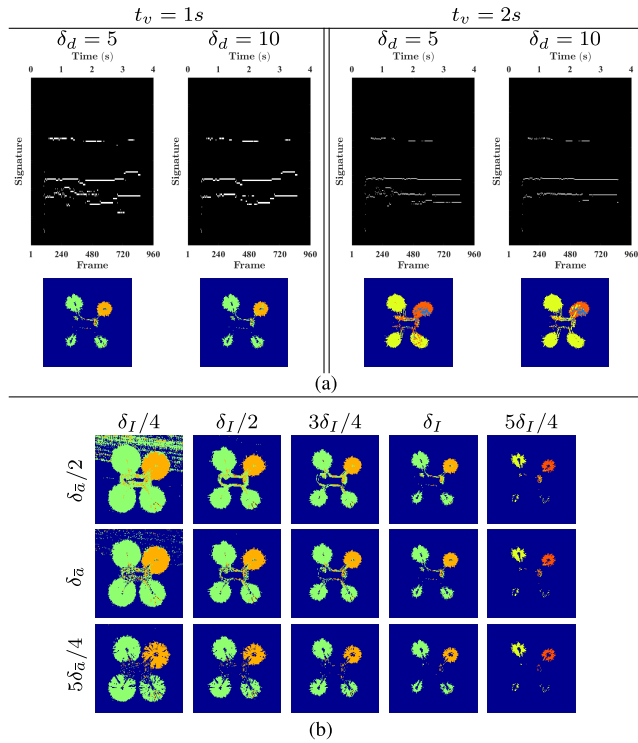


FIGURE 8. Short range PIDS for the Mavic with low moving speed. (a) Top: temporal signature ($t_* = 1/f_c$) obtained for a 4s video stream, with input videos of length $t_v = \{1s; 2s\}$ ($\delta_d = \{5; 10\}$, $\delta_I = 0.35$ and $\delta_a = 3.7$). Bottom: PIDS extracted from the input video ending (frame 600) with the frame in Figure 7a. (b) Influence of coefficients δ_I and δ_a ($\delta_d = 5$, $t_v = 1s$).

detection (*SupMat_MATRICE_LR*, *SupMat_MAVIC_LR*, *SupMat_PHANTOM_LR*).

VII. RESULTS AND DISCUSSIONS

In this section, we present the experimental results of the proposed method and highlight some keys points in a discussion. Results are organized in five groups of figures for each considered drone. The first group of figures (Fig. 7, Fig. 12, Fig. 17) show the four configurations adopted. The four last groups of figures are the results for each of these four configurations, namely, short range detection - low moving speed (Fig. 8, Fig. 13, Fig. 18), long range detection - low moving speed (Fig. 9, Fig. 14, Fig. 19), short range detection - high moving speed (Fig. 10, Fig. 15, Fig. 20), and long range detection - high moving speed (Fig. 11, Fig. 16, Fig. 21). In short range detection, the drone is captured with high resolution (Fig. 7a,c, Fig. 12a,c, Fig. 17a,c), whereas, it is captured with low resolution (around $15pixels \times 15pixels$) in long range detection (Fig. 7b,d, Fig. 12b,d, Fig. 17b,d). Each of the last four groups of figures is constituted by the plots showing the temporal evolution of the extracted PIDS (length of the video stream: 4s, PIDS extracted each $t_* = 1/f_c$ second) and the PIDS obtained for $\delta_d = 5$, $t_v = 1s$ and different values of δ_I and δ_a . The PIDS at a given frame (time) is the signature obtained using an input video of length t_v ending with that frame.

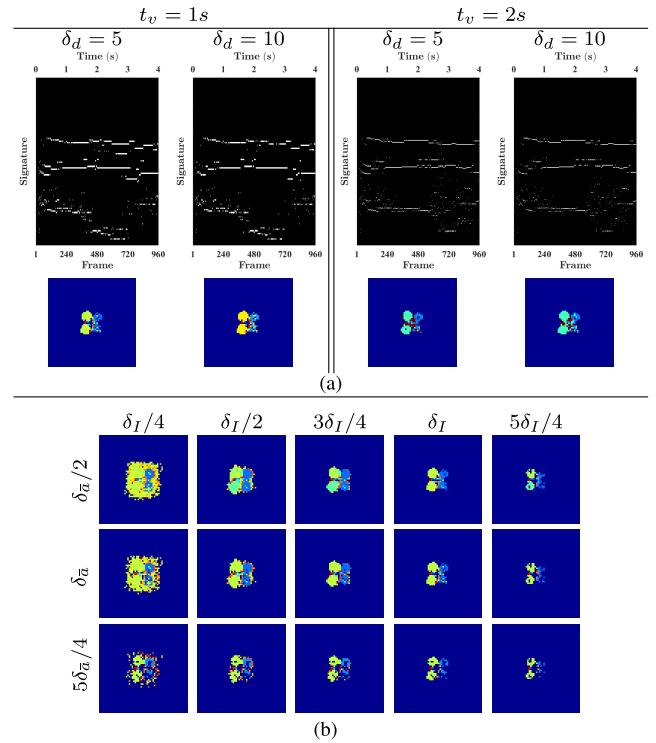


FIGURE 9. Long range PIDS for the Mavic with low moving speed. (a) Top: temporal signature ($t_* = 1/f_c$) obtained for a 4s video stream, with input videos of length $t_v = \{1s; 2s\}$ ($\delta_d = \{5; 10\}$, $\delta_I = 0.15$ and $\delta_a = 3.7$). Bottom: PIDS extracted from the input video ending (frame 745) with the frame in Figure 7b. (b) Influence of coefficients δ_I and δ_a ($\delta_d = 5$, $t_v = 1s$).

A. TRACKING RESULTS

The tracking was performed in two steps. First the event-based method described in Section V was used to determine the coarse position (region of interest in red in Fig. 7, Fig. 12, Fig. 17) of the flying entity in each frame of the video stream between time $t_f - t_v$ and current time (frame) t_f . This set of coarse positions (pixel coordinates) forms the coarse trajectory of the flying entity within this portion of the video stream. Second, the smoothed trajectory (shown in green in Fig. 7, Fig. 12, Fig. 17) was obtained as the moving average of the coarse trajectory. The fine positions (pixel coordinates) on the smoothed trajectory were then used to build the stabilized video provided as input to the PIDS algorithm.

B. TEMPORAL EVOLUTION OF THE PIDS

We have plotted the temporal evolution of the PIDS to evaluate the reliability of the proposed method when all the necessary parameters (moving speed, resolution, parameters of the algorithm) are kept constant. Indeed the temporal PIDS presented show that the signature of the drone is persistent over time. There is almost the same number of main peak at each frame, meaning that the distribution of the pseudo propeller speed is almost the same for the entire video stream (4s long) considered. For high resolution scenarios (Fig. 8a, Fig. 10a, Fig. 13a, Fig. 15a, Fig. 18a, Fig. 20a), the main peaks are almost at the same position, showing that same

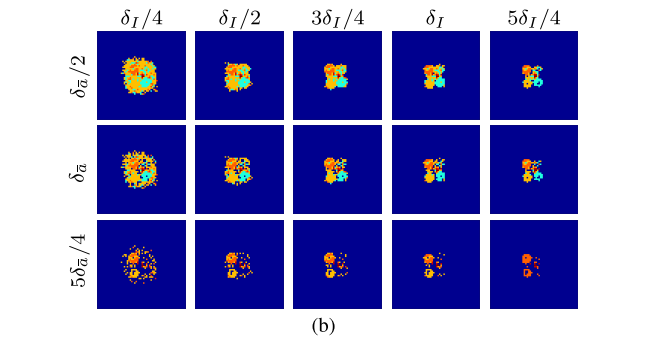
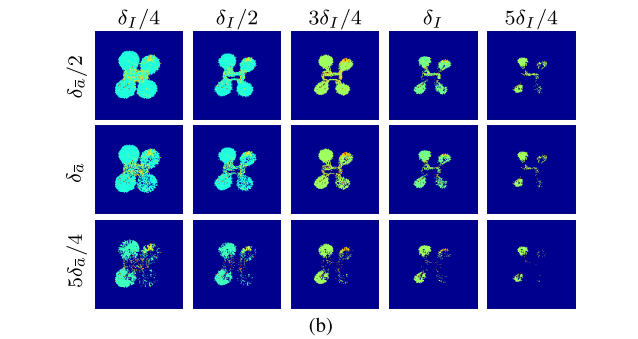
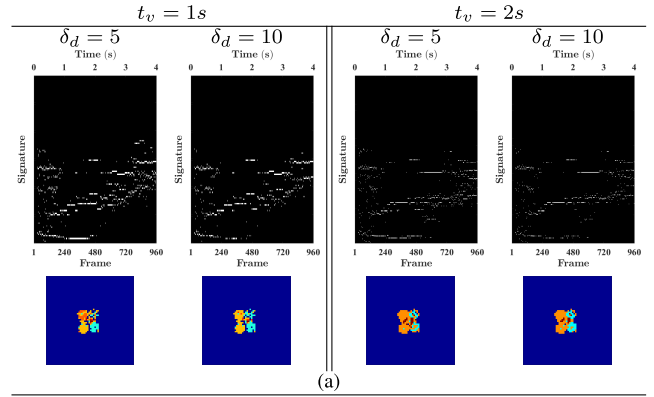
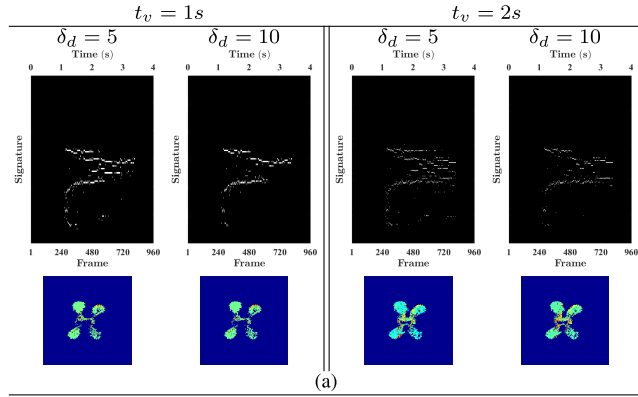


FIGURE 10. Short range PIDS for the Mavic with high moving speed. (a) Top: temporal signature ($t_* = 1/f_c$) obtained for a 4s video stream, with input videos of length $t_v = \{1s; 2s\}$ ($\delta_d = \{5; 10\}$, $\delta_I = 0.27$ and $\delta_{\bar{a}} = 3.3$). Bottom: PIDS extracted from the input video ending (frame 660) with the frame in Figure 7c. (b) Influence of coefficients δ_I and $\delta_{\bar{a}}$ ($\delta_d = 5$, $t_v = 1s$).

FIGURE 11. Long range PIDS for the Mavic with high moving speed. (a) Top: temporal signature ($t_* = 1/f_c$) obtained for a 4s video stream, with input videos of length $t_v = \{1s; 2s\}$ ($\delta_d = \{5; 10\}$, $\delta_I = 0.15$ and $\delta_{\bar{a}} = 3.7$). Bottom: PIDS extracted from the input video ending (frame 480) with the frame in Figure 7d. (b) Influence of coefficients δ_I and $\delta_{\bar{a}}$ ($\delta_d = 5$, $t_v = 1s$).

values of pseudo propeller speed are obtained over time. For low resolution scenarios (Fig. 9a, Fig. 14a, Fig. 16a, Fig. 21a), similar observations are done. However, for the Phantom moving slowly, different values of the pseudo propeller rotation speed are obtained in different parts of the video stream as clearly seen in Fig. 19a ($t_v = 1s$). Moreover, the results obtained with the Mavic moving fast (Fig. 11a) show a drift in the signature from one frame to another. This can be explained by a combined effect of resolution and misalignment of the tracking results. For some videos considered, no drone was present in the first video frames and in the last. For these frames, obviously, no signature (no drone) is found (e.g., see Fig. 15a).

C. INFLUENCE OF ALGORITHM PARAMETERS

The algorithm parameters are the length t_v of the stabilized video, the intensity threshold δ_I , the coefficients $\delta_{\bar{a}}$ and δ_a , and the drift parameter δ_d . In all scenarios, we used $\delta_a = 0.5$. Conceptually, the proposed method is not a single shoot drone detection method in the sense that it needs a certain number of images, actually $t_v f_c$ images, to work properly and provide the PIDS. Using the same video stream (4s long), we have tested the algorithm for different lengths of the stabilized input video, namely $t_v = \{0.5s; 1s; 1.5s; 2s\}$. We obtained the best results with $t_v = 1s$ for all tested scenarios. It is

worth noting that, ideally, each pixel must belong or not to a propeller (region covered by the blades in motion) throughout the entire stabilized video. In practice, the background, the pose of the drone, the resolution with which it is captured, and its moving speed change throughout the video sequence: it is most likely that many pixels belong to a propeller in some frames and to background in other frames. Therefore, the extracted PIDS may include parts of the drone (body) other than the propellers. This is noticeable for long input videos (high value of t_v) as observed for example in Fig. 20a (Bottom: $t_v = 2s$ compare to $t_v = 1s$). Short input videos (a small value for t_v) seems to be a way to improve the PIDS. Indeed, in a short video sequence, we do not expect a significant change in the resolution and the pose between the camera and drone. We also expect an almost constant moving speed. Moreover, for a practical implementation, it is better to choose t_v as small as possible to reduce the computational load. For a given value of the video length t_v , we did not notice any significant difference between PIDS obtained with $\delta_d = 5$ and $\delta_d = 10$. We took $\delta_d = 5$ as the optimum value.

The non-optimal choice of parameters δ_I and $\delta_{\bar{a}}$ affects the PIDS obtained in the same way as t_v , i.e., by increasing the number of misclassified pixels: pixels belonging to background or to the body of the drone are classified as belonging to a propeller (PIDS). Only some combinations

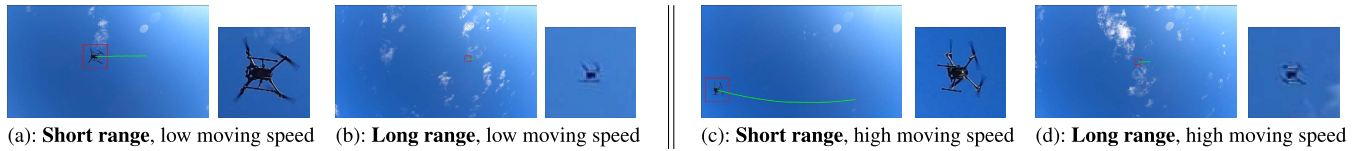


FIGURE 12. Matrice in slow/fast motion in short/long range detection. Left: last frame of the input video sequence (length t_v) showing a detected flying entity (red) and its trajectory within the video. Right: tracking result.

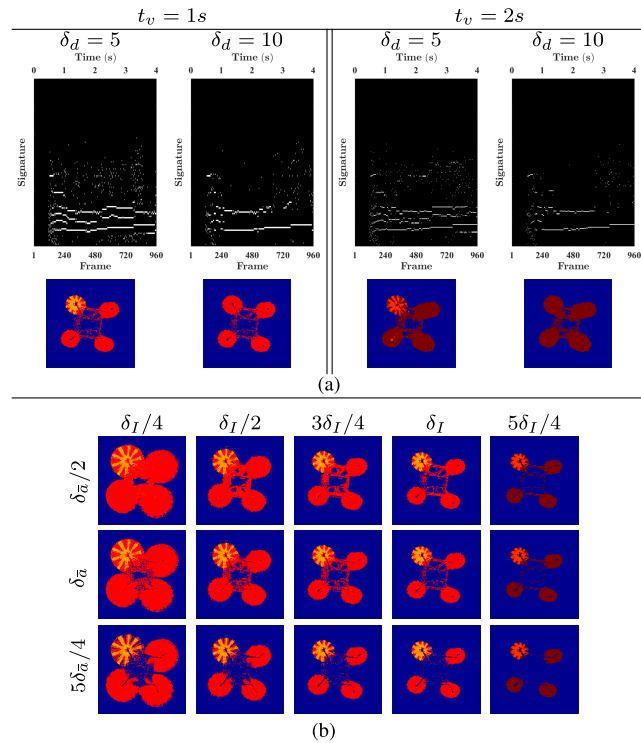


FIGURE 13. Short range PIDS for the Matrice with low moving speed. (a) Top: temporal signature ($t_* = 1/f_c$) obtained for a 4s video stream, with input videos of length $t_v = \{1s; 2s\}$ ($\delta_d = \{5; 10\}$, $\delta_I = 0.25$ and $\delta_{\bar{a}} = 3.5$). Bottom: PIDS extracted from the input video ending (frame 660) with the frame in Figure 12a. (b) Influence of coefficients δ_I and $\delta_{\bar{a}}$ ($\delta_d = 5$, $t_v = 1s$).

of coefficient δ_I and $\delta_{\bar{a}}$ lead to the expected PIDS constituted of four independent regions, each representing a propeller. This is observable ($t_v = 1s$) with the Mavic in Fig. 8b ($\{\delta_I/2; 5\delta_{\bar{a}}/4\}$), in Fig. 9b ($\{3\delta_I/4; \delta_{\bar{a}}/2\}$), in Fig. 10b ($\{\delta_I; \delta_{\bar{a}}\}$), and in Fig. 11b ($\{3\delta_I/4; \delta_{\bar{a}}\}$); with the Matrice in Fig. 13b ($\{5\delta_I/4; 5\delta_{\bar{a}}/4\}$), in Fig. 14b ($\{3\delta_I/4; 5\delta_{\bar{a}}/4\}$), in Fig. 15b ($\{3\delta_I/4; 5\delta_{\bar{a}}/4\}$), and in Fig. 16b ($\{3\delta_I/4; 5\delta_{\bar{a}}/4\}$); with the Phantom in Fig. 18b ($\{\delta_I; \delta_{\bar{a}}\}$) and in Fig. 20b ($\{\delta_I; \delta_{\bar{a}}\}$).

D. INFLUENCE OF RESOLUTION

We remind that, by resolution, we mean the number of pixel capturing the drone in the video. In high resolution, the drone is captured by a high number of pixels, whereas it is captured by very few pixels ($15pixels \times 15pixels$) in low resolution. The proposed method performs very well in all cases (high resolution, low resolution) whatever the moving speed of

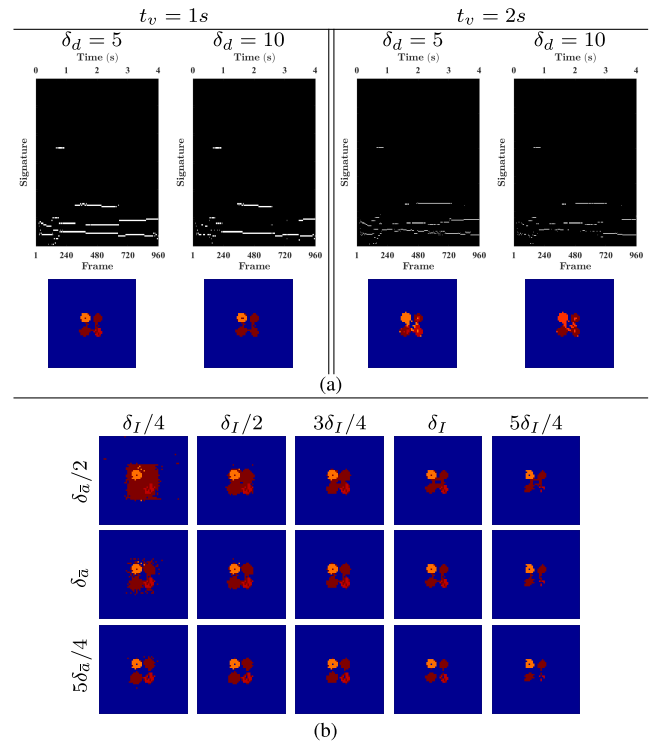


FIGURE 14. Long range PIDS for the Matrice with low moving speed. (a) Top: temporal signature ($t_* = 1/f_c$) obtained for a 4s video stream, with input videos of length $t_v = \{1s; 2s\}$ ($\delta_d = \{5; 10\}$, $\delta_I = 0.15$ and $\delta_{\bar{a}} = 4.6$). Bottom: PIDS extracted from the input video ending (frame 500) with the frame in Figure 12b. (b) Influence of coefficients δ_I and $\delta_{\bar{a}}$ ($\delta_d = 5$, $t_v = 1s$).

the drone. However, the PIDS obtained with the Phantom at low resolution do not clearly show the four propellers (Fig 19b and Fig 21b). This is essentially explained by the low contrast between the white drone and the white cloudy sky constituting the background. The drones used in this study had different sizes and colors. The results obtained show that a resolution of the order of $15pixels \times 15pixels$ (the width of the blades captured by 3 to 5 pixels) is sufficient to extract the PIDS. In all the video sequences used, we had a frontal pose between the camera and the drone (the drone was passing over the camera) and the resolution was almost the same in all the video frames (the drone was moving at a fixed altitude). A minimal resolution greater than $15pixels \times 15pixels$ might be required in other configurations; for example when the resolution changes in the video sequence (drone approaching or moving away from the camera) or when the camera pose is not frontal (drone flying on the horizon). To increase the

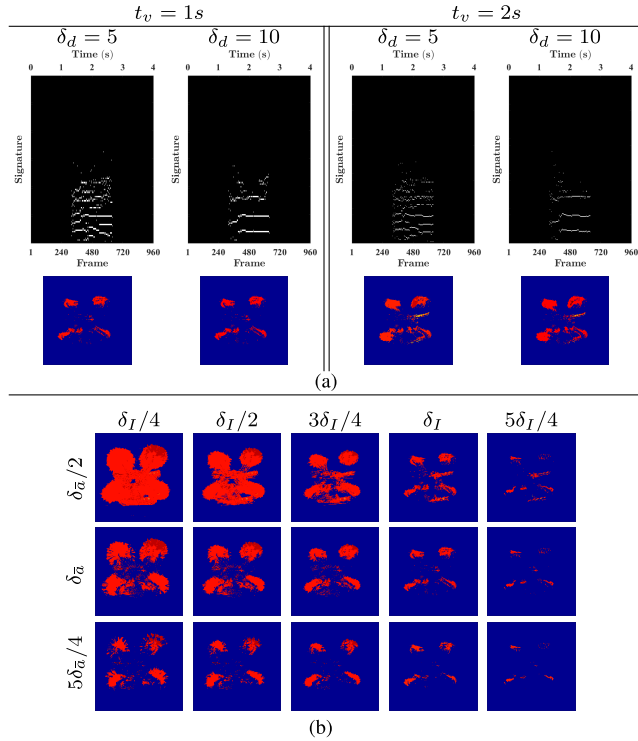


FIGURE 15. Short range PIDS for the Matrice with high moving speed. (a) Top: temporal signature ($t_* = 1/f_c$) obtained for a 4s video stream, with input videos of length $t_v = \{1s; 2s\}$ ($\delta_d = \{5; 10\}$, $\delta_I = 0.44$ and $\delta_a = 5$). Bottom: PIDS extracted from the input video ending (frame 625) with the frame in Figure 12c. (b) Influence of coefficients δ_I and δ_a ($\delta_d = 5$, $t_v = 1s$).

resolution and therefore the detection range while keeping the same coverage (field of view - FoV) of the sky, a strategy could be to use an array of N cameras, each equipped with a small angle lens, having a field of view of FoV/N and covering a different part of the sky. The availability of affordable high-speed cameras, such as the one used in this work, may justify this strategy for practical implementations.

E. INFLUENCE OF DRONE CHARACTERISTICS

Drone characteristics include the span of the drone, its color, the presence of protection elements on the propellers and the load carried by the drone. The span (size) of targeted drones must be taken in account while selecting the camera lens as the detection range is determined by both the camera resolution and span of the drone. The color of the drone is intrinsically related to the contrast obtained with a given background or under given illumination conditions. As with the white Phantom, it might be difficult to extract the PIDS in low resolution videos with a white cloudy sky background.

One requirement of camera-based drone detection methods is the existence of a line of sight between the camera and the drone, more precisely the propellers in the proposed method. The propeller protection elements or a load may obstruct the propeller blades and prevent the correct operation of the proposed method, especially when the drone is flying towards

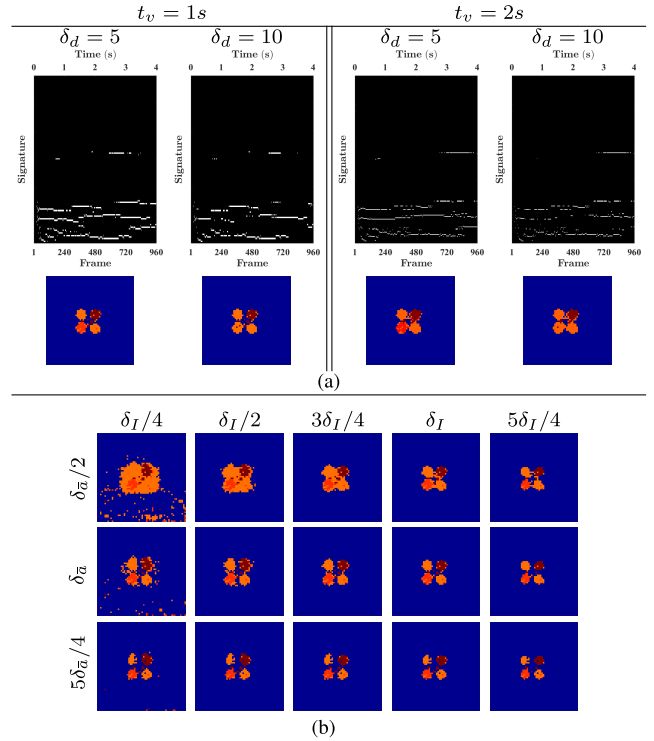


FIGURE 16. Long range PIDS for the Matrice with high moving speed. (a) Top: temporal signature ($t_* = 1/f_c$) obtained for a 4s video stream, with input videos of length $t_v = \{1s; 2s\}$ ($\delta_d = \{5; 10\}$, $\delta_I = 0.15$ and $\delta_a = 4.6$). Bottom: PIDS extracted from the input video ending (frame 500) with the frame in Figure 12d. (b) Influence of coefficients δ_I and δ_a ($\delta_d = 5$, $t_v = 1s$).

or from the horizon. Fortunately, the axis of the propeller must be free to allow air circulation and guarantee the lift and stability of the drone. Therefore, it is very likely that the occlusion will only exist for specific poses between the drone and camera. If the drone is in motion, its pose will change from one video frame to another. Since the stabilized input video is updated with a t_* periodicity, a PIDS will be obtained at some point.

F. POSE OF THE DRONE AND CONTRAST BETWEEN THE DRONE AND THE BACKGROUND

We performed three experiments to investigate the influence of the pose of the drone, and the influence of the contrast between the blades and the background. We characterize the pose of the drone by the angle θ between the camera viewing direction and the propeller rotation axis (see schematic in Figure 24).

The first experiment took place in the laboratory. A propeller with two blades was placed, with different pose ($\theta = \{0; 45; 65; 90\}$), in front of a computer screen (background), whose brightness was changed from zero (background level 0, screen turned off) to the maximum luminosity allowed by the screen (background level 255) with a step of 5. For each background level (BL), the root-mean-square contrast (C_r) was computed as described in [72] using

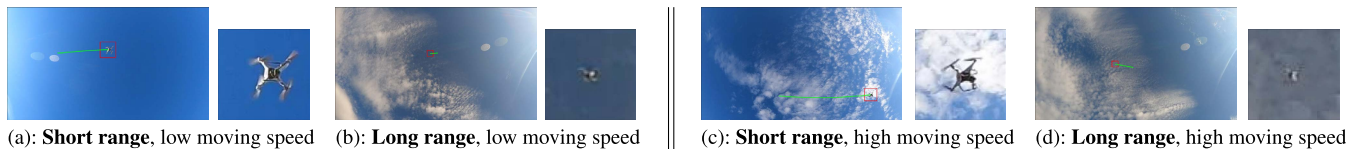


FIGURE 17. Phantom in slow/fast motion in short/long range detection. Left: last frame of the input video sequence (length t_v) showing a detected flying entity (red) and its trajectory within the video. Right: tracking result.

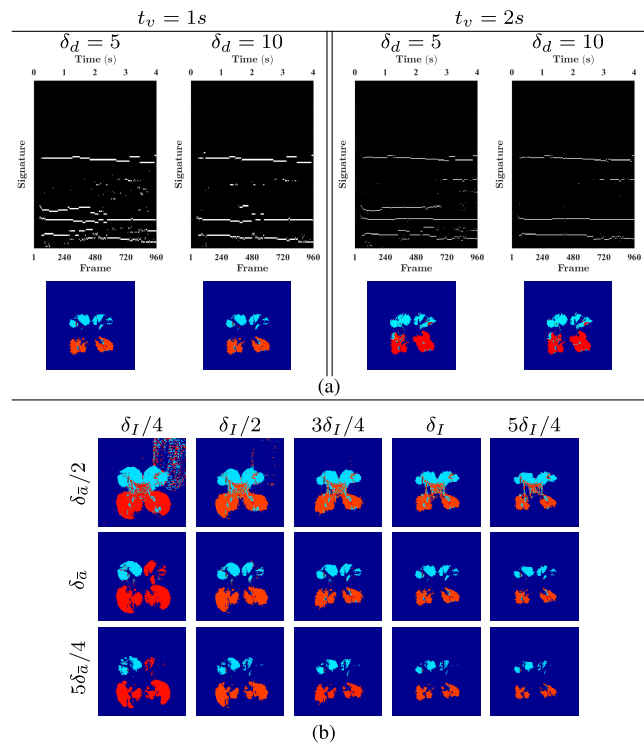


FIGURE 18. Short range PIDS for the Phantom with low moving speed. (a) Top: temporal signature ($t_* = 1/f_c$) obtained for a 4s video stream, with input videos of length $t_v = \{1s; 2s\}$ ($\delta_d = \{5; 10\}$, $\delta_I = 0.2$ and $\delta_a = 5.5$). Bottom: PIDS extracted from the input video ending (frame 565) with the frame in Figure 17a. (b) Influence of coefficients δ_I and δ_a ($\delta_d = 5$, $t_v = 1s$).

intensity signals extracted from background pixels (C_t^{bgd}) and intensity signals extracted from pixels covered by the blades during rotation (C_t^{bla}). For each pose and contrast, the camera captured the spinning propeller during 1s at 240Hz. It turned out that, regardless of the pose of the propeller, the PIDS is always successfully extracted when there is enough contrast between the blades and the background, i.e., when $C_t^{bla} > C_t^{bgd}$. Results of the two extreme poses ($\theta = 0^\circ$, $\theta = 90^\circ$) are reported in Figure 22. The only cases where PIDS is not extracted (not enough contrast) for the frontal pose $\theta = 0^\circ$ is when $BL \in [25; 45[$, or $BL \gg 255$. Note that when $BL \in [0; 25[$, enough contrast exists due to ambient light. For the pose $\theta = 90^\circ$, the PIDS is not extracted when $BL \in [0; 45[$, or $BL \gg 255$. Note also that the case $BL \gg 255$ corresponds to a background excessively bright (e.g., the sun in real scenarios), so that the pixels of the camera are dazzled. It is worth noting that, in the field, the failure

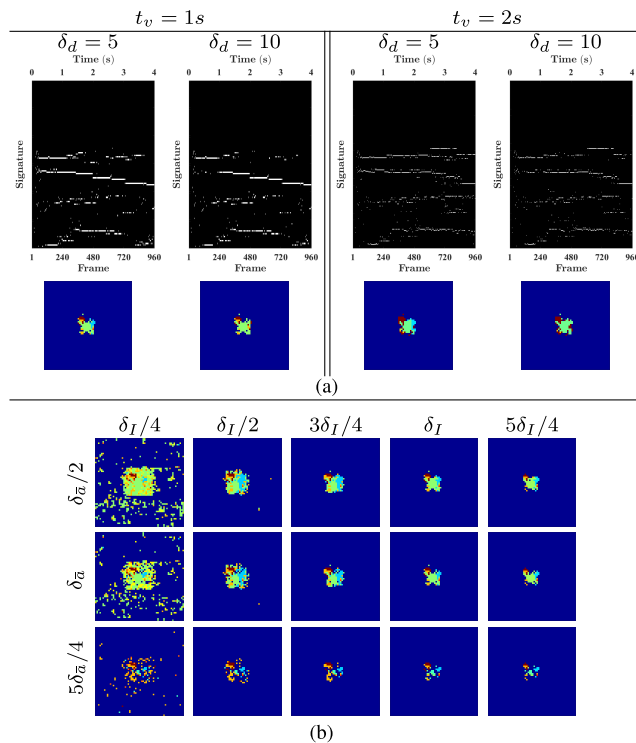


FIGURE 19. Long range PIDS for the Phantom with low moving speed. (a) Top: temporal signature ($t_* = 1/f_c$) obtained for a 4s video stream, with input videos of length $t_v = \{1s; 2s\}$ ($\delta_d = \{5; 10\}$, $\delta_I = 0.15$ and $\delta_a = 3.5$). Bottom: PIDS extracted from the input video ending (frame 590) with the frame in Figure 17b. (b) Influence of coefficients δ_I and δ_a ($\delta_d = 5$, $t_v = 1s$).

case due to very bright background (camera pixels completely dazzled) will only occur if there is perfect alignment between the central area of the sun, the drone (blades captured with few pixels) and the camera. We believe that this extreme case will rarely happen. In real defense applications, we can improve the robustness to camera glare, caused by very bright backgrounds and/or specular reflections from nearby surfaces (urban environment), by monitoring the sky from more than two viewpoints (cameras).

In the second experiment, we confirmed laboratory observations regarding the pose. we captured a drone hovering with different poses. Figure 24 shows the results obtained with the Mavic Mini, one of the smallest drone on the market. In these videos (e.g., see supplementary material *SupMat_MAVIC_Pose90deg*) the width (smallest size) of the blades is captured by 3 to 5 pixels only.

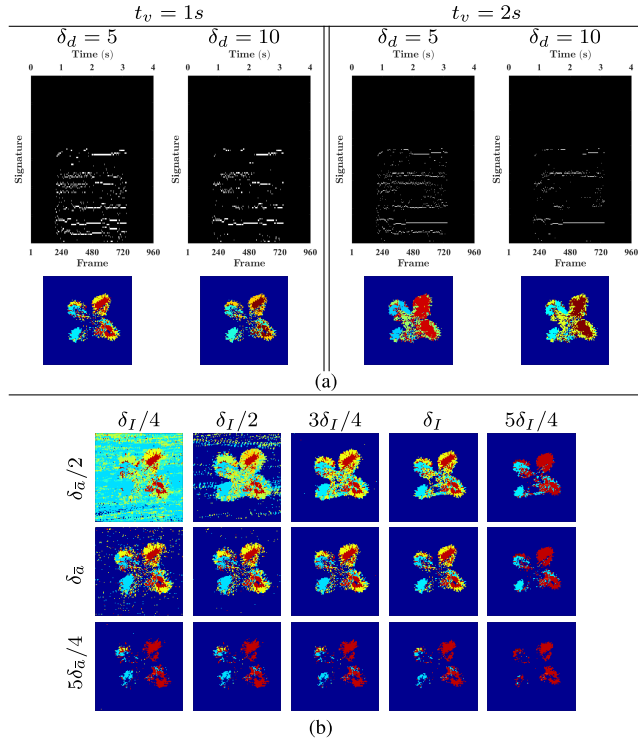


FIGURE 20. Short range PIDS for the Phantom with high moving speed. (a) Top: temporal signature ($t_* = 1/f_c$) obtained for a 4s video stream, with input videos of length $t_v = \{1s; 2s\}$ ($\delta_d = \{5; 10\}$, $\delta_I = 0.3$ and $\delta_{\bar{\alpha}} = 4$). Bottom: PIDS extracted from the input video ending (frame 645) with the frame in Figure 17c. (b) Influence of coefficients δ_I and $\delta_{\bar{\alpha}}$ ($\delta_d = 5$, $t_v = 1s$).

In the third experiment, the flying drone was captured with the sun in the background. Frame by frame results are shown in supplementary material *SupMat_MAVIC_SunAsBackground* and laboratory observations regarding contrast are confirmed as indicated in Figure 23.

G. PRACTICAL IMPLEMENTATION

For practical implementations, a trade-off must be found between the expected detection range for given drone characteristics (color, span) and the camera settings. A multi-cameras system similar to those described in Section VII-D and [53] could be contemplated to extend the detection range although it may result in a complex system. Another important point to consider for a practical implementation is the computing time, that is, the time needed to perform the tracking, build the stabilized videos, each of length t_v , and extract the PIDS. We implemented the proposed algorithm using Matlab and emulate the computations as shown in supplementary materials. The computing time was about 4s for a video of $t_v = 1s$ (240 frames, each of 1080 · 1920 pixels). This time does not include video decoding. We also implemented the proposed algorithm in Microsoft Visual Studio C++ 2019. We used CUDA libraries to take advantage of the parallel computing offered by graphics processing units (GPU). All the computations were performed on a consumer-grade

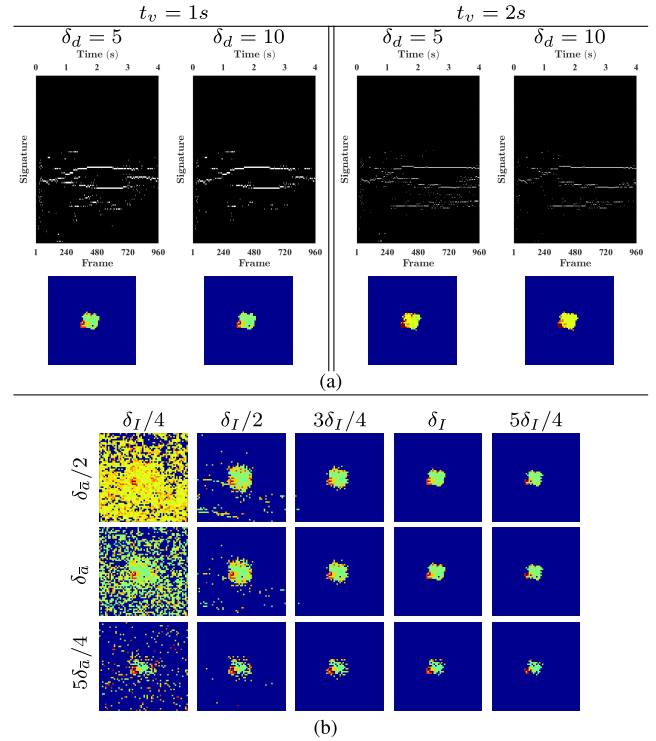


FIGURE 21. Long range PIDS for the Phantom with high moving speed. (a) Top: temporal signature ($t_* = 1/f_c$) obtained for a 4s video stream, with input videos of length $t_v = \{1s; 2s\}$ ($\delta_d = \{5; 10\}$, $\delta_I = 0.15$ and $\delta_{\bar{\alpha}} = 3.5$). Bottom: PIDS extracted from the input video ending (frame 610) with the frame in Figure 7d. (b) Influence of coefficients δ_I and $\delta_{\bar{\alpha}}$ ($\delta_d = 5$, $t_v = 1s$).

computer running Windows 10 (64-bit version) and equipped with a 3.70 GHz Intel(R) Xeon(R) W-2145 CPU, a 32 GB core memory and a GPU (NVIDIA GeForce RTX 3080). The computation time including video loading/decoding, tracking of $n_t = 1$ flying entity, video stabilization, and extraction of the PIDS was 0.93s for $t_v = 1s$ ($n_f = 240$ frames), 1.35s for $t_v = 2s$ ($n_f = 480$ frames), 1.77s for $t_v = 3s$ ($n_f = 720$ frames), and 2.18s for $t_v = 4s$ ($n_f = 960$ frames). Therefore, the proposed algorithm operates in real time. Note that about 30% of processing time was spent on video decoding (GoPro applies H265 compression). Reported computation times can be further reduced if video encoding/decoding is avoided, e.g., by using high-speed machine vision cameras instead of consumer-grade (Gopro) cameras.

H. CONNECTION TO MACHINE/DEEP LEARNING

Camera-based machine/deep learning approaches have been proposed for drone detection [17], [18]. Despite their promising performance, especially for long-range detection, machine/deep learning architectures need training data; they suffer from the reduced amount of annotated data available. Building a training dataset can be a big challenge because, for the sake of robustness, it has to include so many scenarios involving drones (different sizes and colors of drones, various backgrounds and lighting conditions, etc.) as possible. Object

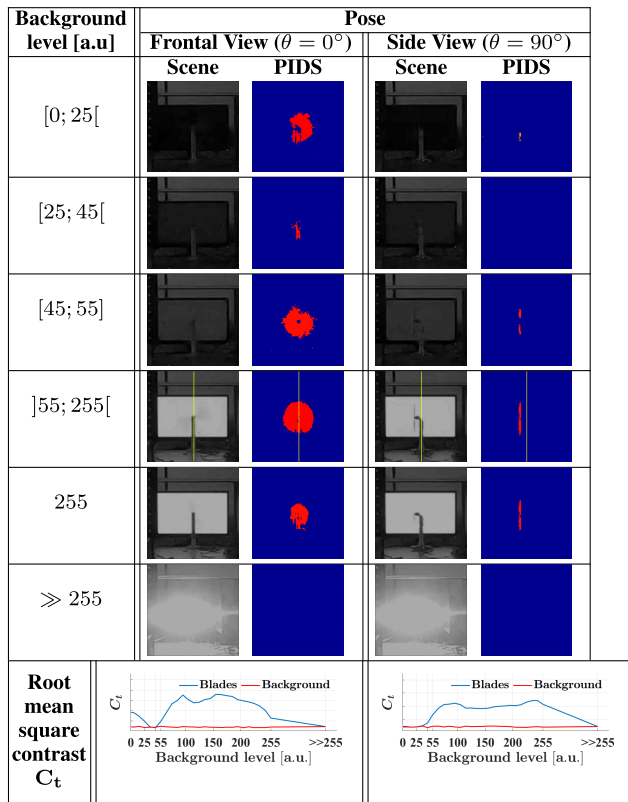


FIGURE 22. Contrast between the propeller blades and the background [72]. θ is the angle between the camera viewing direction and the propeller rotation axis. In this experiment, the PIDS is not extracted when the background level (BL) is between 25 and 45 (frontal view), when it is less than 45 (Side view), or when it is very high. The PIDS is always extracted when there is minimum contrast (e.g., $BL \in [0; 25] \cup [45; 255]$ for $\theta = 0^\circ$, $BL \in [45; 255]$ for $\theta = 90^\circ$) between the blades and the background whatever the pose of the propeller. To avoid confusion, the vertical yellow line ($BL \in]55; 255[$) indicates the position of the bar holding the propeller.

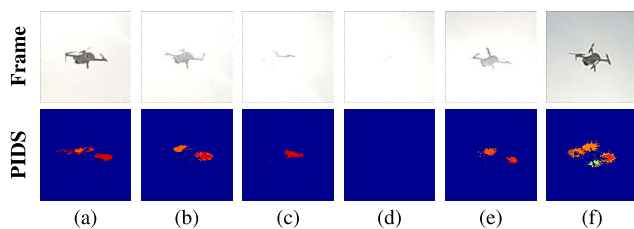


FIGURE 23. Performance with the sun in the background (see supplementary material *SupMat_MAVIC_SunAsBackground*). The PIDS is extracted only when there is a minimum contrast between propeller blades and background. Only (a) three propellers (Frames 388 – 430 in supplementary material), (b) two propellers (Frames 630 – 830), and (c) one propeller (Frames 1000 – 1300) are captured with enough contrast. (d) No contrast, light from bright background glares camera pixels, PIDS is not extracted (Frames 1375 – 1650). Finally, (e) two propellers (Frames 1847 – 1865) and the four propellers (Frames 1900 – 2633) are captured with enough contrast.

detection based on machine/deep learning can be performed using either handcrafted features or learned features. In both cases, features are obtained using the appearance of the object: the object to be detected must look similar as the one

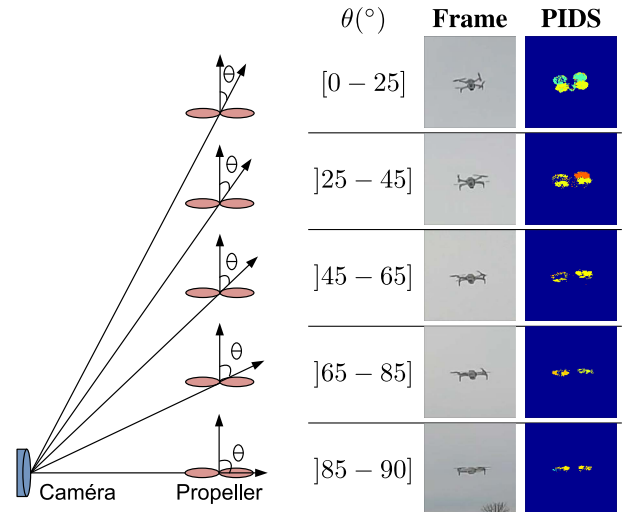


FIGURE 24. PIDS for different poses of the drone. θ is the angle between the camera viewing direction and the propeller rotation axis. If there is minimum contrast between the propeller blades and the background, and the propeller blades are captured with sufficient resolution (at least 3 to 5 pixels for the width of the blades), the PIDS is always extracted regardless of the pose of the drone. Frame by frame results for the extreme pose $\theta \sim 90^\circ$ are given in the supplementary material *SupMat_MAVIC_Pose90deg*.

used for training. Therefore, the training dataset should be updated regularly. It is almost mandatory for drone detection, as new models of drones are regularly released to the market.

The proposed method does not require any training data. It is not an appearance-based method as the drone is detected through the rotation speed of its spinning propellers. In this sense, the PIDS can be extracted as long as the drone is a multicopter, newly released to the market or not. Therefore, one can contemplate machine/deep learning architectures that use the PIDS (kinematic-based features) in addition to appearance-based features. This should compensate, to some extent, for the need for continuous updating of the training dataset. This needs to be confirmed (1) by building a dataset of PIDS extracted from high speed videos capturing drones, (2) by training machine/deep learning architectures with these PIDS dataset, and (3) by testing the robustness of these architectures against new drone models. One can also consider using machine/deep learning for the automatic choice of parameters δ_I , $\delta_{\bar{a}}$, δ_d (parameters of the PIDS algorithm) with respect to sky and illumination conditions. Moreover, in a two-step detection (tracking and classification) such as the one presented in this paper, machine/deep learning can be devoted to the frame by frame tracking of flying entities, while the PIDS is used for classification.

VIII. CONCLUSION

In this paper, we presented a very new end-to-end camera-based drone detection method which uses the propeller rotation speed to extract an unambiguous drone signature. The

method proceeds in three steps. First, a flying entity is detected and tracked frame by frame in a video stream provided by a high speed camera observing the sky from a fixed position. Second, the tracking results are used to build the motion-stabilized input video capturing the tracked flying entity. Third, the propellers induced drone signature (PIDS) is extracted from the input video sequence to confirm the flying entity as being a drone or not. The theoretical fundamentals of the method has been detailed. Experimental results obtained with four different drones (Mavic, Matrice, Phantom, Mavic Mini), captured with low/high resolution while they moving with low/high speed with different poses and backgrounds, demonstrated the effectiveness and the reliability of the proposed method. In particular, the PIDS, which unambiguously identify a drone, can be successfully extracted if: (1) there is a line of sight between the camera and the propeller blades (the protective elements or the body of the drone are the main causes of occlusion of the blades); (2) the blades are captured with enough resolution (at least 3 to 5 pixels for the width of the blades); (3) a minimum contrast exists between the blades and the background. We would like to point out that the proposed method should also work with near infrared cameras, especially for drone detection at night.

All scenarios considered in this paper had the sky (clear or cloudy) or the sun as background. A future work will be to test the performance of the proposed method in urban environment or in the presence of vegetation/forest. It will also be interesting to develop machine/deep learning architectures using PIDS as an extra channel concatenated to the RGB channels (appearance-based) commonly used.

REFERENCES

- [1] H. Shakhathreh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khreishah, and M. Guizani, "Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges," *IEEE Access*, vol. 7, pp. 48572–48634, 2019.
- [2] (Apr. 2021). *Commercial Drone Market Size, Share and Trends Analysis Report by Product (Fixed-Wing, Rotary Blade, Hybrid), by Application, by End-Use, by Region, and Segment Forecasts, 2021–2028*. Report: Grand View Research. Accessed: Mar. 14, 2022. [Online]. Available: <https://www.grandviewresearch.com/industry-analysis/global-commercial-drones-market>
- [3] (2017). *White Paper: Drones: Threat From Above*. Technical Report. G4S North America. Accessed: Mar. 14, 2022. [Online]. Available: https://www.g4s.com/en-ca/-/media/g4s/canada/files/whitepapers/usa/drones_t%hreat_from_above.ashx
- [4] J.-P. Yaacoub, H. Noura, O. Salman, and A. Chehab, "Security analysis of drones systems: Attacks, limitations, and recommendations," *Internet Things*, vol. 11, Sep. 2020, Art. no. 100218.
- [5] K. L. Best, J. Schmid, S. Tierney, J. Awan, N. M. Beyene, M. A. Holliday, R. Khan, and K. Lee, "How to analyze the cyber threat from drones background, analysis frameworks, and analysis tools," Santa Monica, CA, USA: RAND Corporation, 2020, doi: [10.7249/RR2972](https://doi.org/10.7249/RR2972).
- [6] (2017). *Master List of Drone Laws (Organized by State & Country): A Global Directory of Drone Laws and Regulations*. Accessed: Mar. 14, 2022. [Online]. Available: <https://uavcoach.com/drone-laws/>
- [7] B. Jansen. (Jan. 26, 2015). *Drone Crash at White House Reveals Security Risks*. USA Today. Accessed: Mar. 14, 2022. [Online]. Available: <https://www.usatoday.com/story/news/2015/01/26/drone-crash-secret-service-fa%a/22352857/>
- [8] CBC News. (Oct. 15, 2017). *A First in Canada: Drone Collides With Passenger Plane Above Quebec City Airport*. Accessed: Mar. 14, 2022. [Online]. Available: <https://www.cbc.ca/news/canada/montreal/gameau-airport-drone-quebec-1.43557%92>
- [9] A. Jouan. (Nov. 25, 2014). *Survols de Centrales: Un Expert Reconnu S'inquiète*. Figaro. Accessed: Mar. 14, 2022. [Online]. Available: <https://www.lefigaro.fr/actualite-france/2014/11/25/01016-20141125ARTFIG0002%4-survols-de-centrales-un-expert-reconnu-s-inquiete.php>
- [10] J. Robinson. (Mar. 24, 2017). *Kaboom! Incredible Moment Rockets Fizz Across a Ukrainian City After an Arms Warehouse Packed With Ammunition and Weapons Was Bombed by a Drone*. Dailymail. Accessed: Apr. 13, 2021. [Online]. Available: <https://www.dailymail.co.uk/news/article-4346328/Clips-blast-Ukrainian-arms-%depot-bombed-drone.html>
- [11] T. Howell. (Apr. 11, 2020). *3 Suspects Planned to Use Drone to Get Items to Hays Prison Inmate*. SKY21. Accessed: Mar. 14, 2022. [Online]. Available: <https://sky21.com/2020/04/3-suspects-planned-to-use-drone-to-get-items-to-in%mate/>
- [12] S. Dinan. (Aug. 20, 2017). *Mexican Drug Cartels Using Drones to Smuggle Heroin, Meth, Cocaine Into U.S.* Washington Times. Accessed: Mar. 14, 2022. [Online]. Available: <https://www.washingtontimes.com/news/2017/aug/20/mexican-drug-cartels-using-%drones-to-smuggle-heroi/>
- [13] L. Zhang and S. Y. Lee. (Mar. 30, 2018). *China Busts Smugglers Using Drones to Transport Smartphones: State Media*. Reuters. Accessed: Mar. 14, 2022. [Online]. Available: <https://www.reuters.com/article/us-china-crime-smartphones-smugglers/china-b%usts-smugglers-using-drones-to-transport-smartphones-state-media-idUSKBN1H60BT>
- [14] G. S. McNeal. (2015). *Unmanned Aerial System Threats: Exploring Security Implications and Mitigation Technologies*. Hearing Before the Subcommittee on Oversight and Management Efficiency of the Committee on Homeland Security, House of Representatives. Accessed: Mar. 14, 2022. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2741820
- [15] J. Flórez, J. Ortega, A. Betancourt, A. García, M. Bedoya, and J. S. Botero, "A review of algorithms, methods, and techniques for detecting UAVs and UAS using audio, radiofrequency, and video applications," *TecnoLógicas*, vol. 23, no. 48, pp. 269–285, May 2020.
- [16] S. Hengy, M. Laurenzi, S. Schertzer, A. Hommes, F. Kloeppel, A. Shoykhetbrod, T. Geibig, W. Johannes, O. Rassy, and F. Christnacher, "Multimodal UAV detection: Study of various intrusion scenarios," in *Proc. Electro-Opt. Remote Sens. XI*, Oct. 2017, Art. no. 104340P.
- [17] B. Taha and A. Shoufan, "Machine learning-based drone detection and classification: State-of-the-art in research," *IEEE Access*, vol. 7, pp. 138669–138682, 2019.
- [18] A. Coluccia, A. Fascista, A. Schumann, L. Sommer, A. Dimou, D. Zarpalas, M. Méndez, D. de la Iglesia, I. González, J.-P. Mercier, G. Gagné, A. Mitra, and S. Rajashekar, "Drone vs. bird detection: Deep learning algorithms and results from a grand challenge," *Sensors*, vol. 21, no. 8, p. 2824, Apr. 2021.
- [19] M. Nijim and N. Mantrawadi, "Drone classification and identification system by phenome analysis using data mining techniques," in *Proc. IEEE Symp. Technol. Homeland Secur. (HST)*, May 2016, pp. 1–5.
- [20] J. Mezei and A. Molnar, "Drone sound detection by correlation," in *Proc. IEEE 11th Int. Symp. Appl. Comput. Intell. Informat. (SACI)*, May 2016, pp. 509–518.
- [21] J. Kim, C. Park, J. Ahn, Y. Ko, J. Park, and J. C. Gallagher, "Real-time UAV sound detection and analysis system," in *Proc. IEEE Sensors Appl. Symp. (SAS)*, Mar. 2017, pp. 1–5.
- [22] A. Bernardini, F. Mangiatordi, E. Pallotti, and L. Capodiferro, "Drone detection by acoustic signature identification," *Electron. Imag.*, vol. 29, no. 10, pp. 60–64, Jan. 2017.
- [23] Y. Seo, B. Jang, and S. Im, "Drone detection using convolutional neural networks with acoustic STFT features," in *Proc. 15th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Nov. 2018, pp. 1–6.
- [24] N. P. Bhatta and M. GeethaPriya, "Radar and its applications," *IJCTA*, vol. 10, pp. 1–9, Jan. 2017.
- [25] V. Semkin, J. Haarla, T. Pairon, C. Slezak, S. Rangan, V. Viikari, and C. Oestges, "Analyzing radar cross section signatures of diverse drone models at mmWave frequencies," *IEEE Access*, vol. 8, pp. 48958–48969, 2020.

- [26] J. S. Patel, F. Fioranelli, and D. Anderson, "Review of radar classification and RCS characterisation techniques for small UAVs or drones," *IET Radar, Sonar Navigat.*, vol. 12, no. 9, pp. 911–919, 2018.
- [27] P. Klaer, A. Huang, P. Sévigny, S. Rajan, S. Pant, P. Patnaik, and B. Balaji, "An investigation of rotary drone HERM line spectrum under manoeuvring conditions," *Sensors*, vol. 20, no. 20, p. 5940, Oct. 2020.
- [28] M. Ritchie, F. Fioranelli, H. Griffiths, and B. Torvik, "Micro-drone RCS analysis," in *Proc. IEEE Radar Conf.*, Oct. 2015, pp. 452–456.
- [29] R. Guay, G. Drolet, and J. R. Bray, "Measurement and modelling of the dynamic radar cross-section of an unmanned aerial vehicle," *IET Radar, Sonar Navigat.*, vol. 11, no. 7, pp. 1155–1160, 2017.
- [30] J. Ren and X. Jiang, "Regularized 2-D complex-log spectral analysis and subspace reliability analysis of micro-Doppler signature for UAV detection," *Pattern Recognit.*, vol. 69, pp. 225–237, Sep. 2017.
- [31] M. Ritchie, F. Fioranelli, H. Borrión, and H. Griffiths, "Multistatic micro-Doppler radar feature extraction for classification of unloaded/loaded micro-drones," *IET Radar, Sonar Navigat.*, vol. 11, no. 1, pp. 116–124, Jan. 2017.
- [32] B. K. Kim, H.-S. Kang, and S.-O. Park, "Drone classification using convolutional neural networks with merged Doppler images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 1, pp. 38–42, Jan. 2017.
- [33] R. Palamà, F. Fioranelli, M. Ritchie, M. Inggis, S. Lewis, and H. Griffiths, "Measurements and discrimination of drones and birds with a multi-frequency multistatic radar system," *IET Radar, Sonar Navigat.*, vol. 15, no. 8, pp. 841–852, Aug. 2021.
- [34] Y. Liu, X. Wan, H. Tang, J. Yi, Y. Cheng, and X. Zhang, "Digital television based passive bistatic radar system for drone detection," in *Proc. IEEE Radar Conf. (RadarConf)*, May 2017, pp. 1493–1497.
- [35] T. Martelli, F. Murgia, F. Colone, C. Bongioanni, and P. Lombardo, "Detection and 3D localization of ultralight aircrafts and drones with a WiFi-based passive radar," in *Proc. Int. Conf. Radar Syst. (Radar)*, Oct. 2017, pp. 1–6.
- [36] P. Nguyen, H. Truong, M. Ravindranathan, A. Nguyen, R. Han, and T. Vu, "Matthan: Drone presence detection by identifying physical signatures in the Drone's RF communication," in *Proc. 15th Annu. Int. Conf. Mobile Syst., Appl., Services*, Jun. 2017, pp. 211–224.
- [37] A. Shoufan, H. M. Al-Angari, M. F. A. Sheikh, and E. Damiani, "Drone pilot identification by classifying radio-control signals," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 10, pp. 2439–2447, Oct. 2018.
- [38] I. Bisio, C. Garibotto, F. Lavagetto, A. Sciarone, and S. Zappatore, "Unauthorized amateur UAV detection based on WiFi statistical fingerprint analysis," *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 106–111, Apr. 2018.
- [39] M. Ezuma, F. Erden, C. K. Anjinappa, O. Ozdemir, and I. Guvenc, "Micro-UAV detection and classification from RF fingerprints using machine learning techniques," in *Proc. IEEE Aerosp. Conf.*, Mar. 2019, pp. 1–13.
- [40] I. Nemer, T. Sheltami, I. Ahmad, A. U.-H. Yasar, and M. A. R. Abdeen, "RF-based UAV detection and identification using hierarchical learning approach," *Sensors*, vol. 21, no. 6, p. 1947, Mar. 2021.
- [41] A. Alsoliman, G. Rigoni, M. Levorato, C. Pinotti, N. O. Tippenhauer, and M. Conti, "COTS drone detection using video streaming characteristics," in *Proc. Int. Conf. Distrib. Comput. Netw.*, Jan. 2021, pp. 166–175.
- [42] S. Hu, G. H. Goldman, and C. C. Borel-Donohue, "Detection of unmanned aerial vehicles using a visible camera system," *Appl. Opt.*, vol. 56, no. 3, p. B214, 2017.
- [43] U. C. Craye and S. Ardjoune, "Spatio-temporal semantic segmentation for drone detection," in *Proc. AVSS*, 2019, pp. 18–21.
- [44] E. Unlu, E. Zenou, and N. Riviere, "Using shape descriptors for UAV detection," in *Proc. IS&T Int. Symp. Electron. Imag., Intell. Robot. Ind. Appl. Using Computer Vision*, Jan. 2018, pp. 128–1–128–5.
- [45] U. Seidaliyeva, D. Akhmetov, L. Ilipbayeva, and E. T. Matson, "Real-time and accurate drone detection in a video with a static background," *Sensors*, vol. 20, no. 14, p. 3856, Jul. 2020.
- [46] J. Peng, C. Zheng, P. Lv, T. Cui, Y. Cheng, and L. Si, "Using images rendered by PBRT to train faster R-CNN for UAV detection," *Proc. Int. Conf. Central Europe Comput. Graph. Vis. Comput. Vis.*, 2018, pp. 13–18, doi: 10.24132/CSRN.2018.2802.3.
- [47] C. Aker and S. Kalkan, "Using deep networks for drone detection," in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug. 2017, pp. 1–6.
- [48] E. Unlu, E. Zenou, N. Riviere, and P.-E. Dupouy, "Deep learning-based strategies for the detection and tracking of drones using several cameras," *IPSSJ Trans. Comput. Vis. Appl.*, vol. 11, no. 1, pp. 1–13, Dec. 2019.
- [49] S. R. Ganti and Y. Kim, "Implementation of detection and tracking mechanism for small UAS," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2016, pp. 1254–1260.
- [50] A. Rozantsev, V. Lepetit, and P. Fua, "Detecting flying objects using a single moving camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 5, pp. 879–892, 2017.
- [51] M. A. Akhloufi, S. Arola, and A. Bonnet, "Drones chasing drones: Reinforcement learning and deep search area proposal," *Drones*, vol. 3, no. 3, p. 58, Jul. 2019.
- [52] A. Carrio, J. Tordesillas, S. Vemprala, S. Saripalli, P. Campoy, and J. P. How, "Onboard detection and localization of drones using depth maps," *IEEE Access*, vol. 8, pp. 30480–30490, 2020.
- [53] Y. Wu, Y. Sui, and G. Wang, "Vision-based real-time aerial object localization and tracking for UAV sensing system," *IEEE Access*, vol. 5, pp. 23969–23978, 2017.
- [54] X. Zhang and K. Kusriñi, "Autonomous long-range drone detection system for critical infrastructure safety," *Multimedia Tools Appl.*, vol. 80, pp. 23723–23743, Feb. 2021.
- [55] V. Magoulaniotis, D. Ataloglou, A. Dimou, D. Zarpalas, and P. Daras, "Does deep super-resolution enhance UAV detection?" in *Proc. 16th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Sep. 2019, pp. 1–6.
- [56] N. Dorudian, S. Lauria, and S. Swift, "Nonparametric background modelling and segmentation to detect micro air vehicles using RGB-D sensor," *Int. J. Micro Air Vehicles*, vol. 11, pp. 1–20, Jul. 2019.
- [57] A. Carrio, S. Vemprala, A. Ripoll, S. Saripalli, and P. Campoy, "Drone detection using depth maps," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1034–1037.
- [58] P. Andrašić, T. Radišić, M. Muštra, and J. Ivošević, "Night-time detection of UAVs using thermal infrared camera," *Transp. Res. Proc.*, vol. 28, pp. 183–190, Jan. 2017.
- [59] A. Thomas, V. Lebouchier, A. Cotinat, P. Finet, and M. Gilbert, "UAV localization using panoramic thermal cameras," in *Computer Vision Systems (Lecture Notes in Computer Science)*, vol. 11754, D. Tzovaras, D. Giakoumis, M. Vincze, and A. Argyros, Eds. Cham, Switzerland: Springer, 2019, pp. 754–767.
- [60] V. Popovic, B. Ott, P. Wellig, and Y. Leblebici, "Near-infrared high-resolution real-time omnidirectional imaging platform for drone detection," *Proc. SPIE*, vol. 9997, Oct. 2016, Art. no. 999706.
- [61] T. Müller and B. Erdnüß, "Robust drone detection with static VIS and SWIR cameras for day and night counter-UAV," *Proc. SPIE*, vol. 11166, Oct. 2019, Art. no. 111660B.
- [62] T. Stewart, M.-A. Drouin, G. Gagne, and G. Godin, "Drone virtual fence using a neuromorphic camera," in *Proc. Int. Conf. Neuromorphic Syst.*, Jul. 2021, pp. 1–9.
- [63] S. Park, S. Shin, Y. Kim, E. T. Matson, K. Lee, J. C. Slater, M. Scherrek, M. Sam, J. C. Gallagher, P. J. Kolodzy, B. R. Fox, and M. Hopmeier, "Combination of radar and audio sensors for identification of rotor-type unmanned aerial vehicles (UAVs)," in *Proc. IEEE Sensors*, Nov. 2015, pp. 1–4.
- [64] H. Liu, Z. Wei, Y. Chen, J. Pan, L. Lin, and Y. Ren, "Drone detection based on an audio-assisted camera array," in *Proc. IEEE 3rd Int. Conf. Multimedia Big Data (BigMM)*, Apr. 2017, pp. 402–406.
- [65] M. Hammer, B. Borgmann, M. Hebel, and M. Arens, "UAV detection, tracking, and classification by sensor fusion of a 360° lidar system and an alignable classification sensor," *Proc. SPIE*, vol. 11005, May 2019, Art. no. 110050E.
- [66] Y. Benezeth, P. M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger, "Review and evaluation of commonly-implemented background subtraction algorithms," in *Proc. 19th Int. Conf. Pattern Recognit.*, Dec. 2008, pp. 1–4.
- [67] B. Garcia-Garcia, T. Bouwmans, and A. J. R. Silva, "Background subtraction in real applications: Challenges, current models and future directions," *Comput. Sci. Rev.*, vol. 35, Feb. 2020, Art. no. 100204.
- [68] P. Dendorfer, A. Ošep, A. Milan, K. Schindler, D. Cremers, I. Reid, S. Roth, and L. Leal-Taixé, "MOTChallenge: A benchmark for single-camera multiple target tracking," *Int. J. Comput. Vis.*, vol. 129, no. 4, pp. 845–881, Apr. 2021.
- [69] I. Rodríguez-Padilla, B. Castelle, V. Marieu, and D. Morichon, "A simple and efficient image stabilization method for coastal monitoring video systems," *Remote Sens.*, vol. 12, no. 1, p. 70, Dec. 2019.
- [70] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 154–180, Jul. 2022.

- [71] GoPro, "Hero6 Black," [Online]. Available: <https://community.gopro.com/t5/en/HERO6-Black-Field-of-View-FOV-Information/ta-p/390193?profile.language=en>. Accessed: Apr. 13, 2021.
- [72] E. Peli, "Contrast in complex images," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 7, no. 10, pp. 2032–2040, Oct. 1990.



FRANK BILLY DJUPKEP DIZEU received the B.S. degree in physics and the M.S. degree in physics (nonlinear dynamic systems) from the University of Dschang, Cameroon, in 2005 and 2008, respectively, and the Ph.D. degree (electrical engineering) in infrared vision with the Multipolar Infrared Vision (MIVIM) Research Chair, Laval University, in 2015.

In 2012, during his Ph.D. degree, he was selected for an internship with the Italian National Research Council, Padova. After his Ph.D. degree, he obtained a Postdoctoral Fellowship with the Industrial Research Chair CREATION 3D, Laval University, where he led a project combining 3D vision and infrared thermography for nondestructive inspection of objects, between 2015 and 2017. In 2017, he joined the National Research Council Canada as a Postdoctoral Fellow. Since 2019, he has been a Research Officer with the Computer Vision and Graphic Team at the National Research Council Canada. His research interests include around the development of imaging modalities and systems to support Canadian industries, with major interest in signal processing, active triangulation, structured-light imaging, and optical coherence tomography.



MICHEL PICARD is currently a Technical Officer with the National Research Council (NRC). He joined NRC, in 1999, after working in the private sector with a 3D scanner manufacturer. He has contributed to diverse research projects and technology transfers. On multiple occasions, he participated in the development and transfer of advanced 3D scanners that enabled Canadian companies to thrive in the marketplace. He has also played a key role in the development and

transfer of 3D data processing software. He also participated in research projects that were concluded with technology demonstrations recognized on the international stage. He has coauthored numerous scientific papers and contributed to patents.



MARC-ANTOINE DROUIN is currently a Senior Research Officer with the National Research Council (NRC), Computer Vision and Graphic Team. In 2002, he joined, as a student, the Vision3D laboratory from the Department of Computer Science and Operations Research at University of Montréal. During his studies, he became interested in multi-view passive triangulation systems. The results of his work have been presented in the most prestigious computer vision seminars and journals. While working towards his Ph.D. degree, he was with the Université de Montréal, in 2006, he joined NRC as a Guest Worker to perform research on camera-projector systems. In 2007, after being awarded his Ph.D. degree, he joined NRC full-time. His work has focused on active and passive 3D imaging systems, specifically fringe projection systems, and airborne 3D reconstruction. Since, he has been joining NRC, some of his activities have led to the development of many fringe projection prototypes, several patents and software related to 3D imaging. Moreover, he has lead more than 30 client-based research contracts in the security and defence sector.



GUILLAUME GAGNÉ received the B.S. degree in physics, in 2005, and the M.S. degree in physics from Laval University, Quebec City, Canada, in 2007. Between 2007 and 2009, he worked on a group of operational research for the development of a crowd control modeling and simulation tool. Between 2009 and 2017, he worked as a Physicist at AEREX Avionics Inc. He was involved in various projects involving electro-optics sensors from ultra-violet to far infra-red bands, hyperspectral, multi-bands, and airborne. He also developed 3D model and infrared signatures for modeling and simulation applications. Since June 2017, he has been working as a Defense Scientist at DRDC. His research interests include performance evaluation of EO/IR sensors, image processing, computer vision, realist synthetic scene generation in modeling and simulation applications, and assessing electro-optical sensors in counter drone applications.

...