2019 IEEE Symposium on Security and Privacy

# Drones' Cryptanalysis - Smashing Cryptography with a Flicker

Ben Nassi[1], Raz Ben-Netanel[1], Adi Shamir[2], Yuval Elovici[1]

**Video 1** - https://youtu.be/4icQwducz68 **Video 2** - https://youtu.be/9PVaDpMsyQE

[1] Ben-Gurion University of the Negev, [2] Weizmann Institute of Science
nassib@post.bgu.ac.il, razx@post.bgu.ac.il, adi.shamir@weizmann.ac.il, and elovici@inter.net.il

## Abstract

In an "open skies" era in which drones fly among us, a new question arises: how can we tell whether a passing drone is being used by its operator for a legitimate purpose (e.g., delivering pizza) or an illegitimate purpose (e.g., taking a peek at a person showering in his/her own house)? Over the years, many methods have been suggested to detect the presence of a drone in a specific location, however since populated areas are no longer off limits for drone flights, the previously suggested methods for detecting a privacy invasion attack are irrelevant. In this paper, we present a new method that can detect whether a specific POI (point of interest) is being video streamed by a drone. We show that applying a periodic physical stimulus on a target/victim being video streamed by a drone causes a watermark to be added to the encrypted video traffic that is sent from the drone to its operator and how this watermark can be detected using interception. Based on this method, we present an algorithm for detecting a privacy invasion attack. We analyze the performance of our algorithm using four commercial drones (DJI Mavic Air, Parrot Bebop 2, DJI Spark, and DJI Mavic Pro). We show how our method can be used to (1) determine whether a detected FPV (first-person view) channel is being used to video stream a POI by a drone, and (2) locate a spying drone in space; we also demonstrate how the physical stimulus can be applied covertly. In addition, we present a classification algorithm that differentiates FPV transmissions from other suspicious radio transmissions. We implement this algorithm in a new invasion attack detection system which we evaluate in two use cases (when the victim is inside his/her house and when the victim is being tracked by a drone while driving his/her car); our evaluation shows that a privacy invasion attack can be detected by our system in about 2-3 seconds.

## I. Introduction

The proliferation of consumer drones over the last few years [1], [2] has created a new privacy threat [3], [4], [5], [6], [7]. We are living in an era in which anyone with a drone equipped with a video camera can use it to perform a privacy invasion attack by flying the drone in order to: detect a cheating spouse [3], spy on people [4], [5] or celebrities [6], or video stream a neighbor's sunbathing daughter [7]. The president of the United States signed a memo allowing drones to fly in populated/urban areas in 2017 [8] as part of the new "open skies" policy, an act which is expected to make the detection of privacy invasion attacks more challenging, as increasing numbers of business and companies begin to adopt drones for various legitimate purposes. Drones are now being used for pizza delivery [9], the shipment of goods [10], filming [11], and many other legitimate purposes [12], and their presence is no longer restricted in populated areas. Given that, **how can we tell whether a drone that is passing near a house is being used for a legitimate purpose (e.g., delivering pizza) or an illegitimate purpose (e.g., taking a peek at a person showering in his/her own house)?**

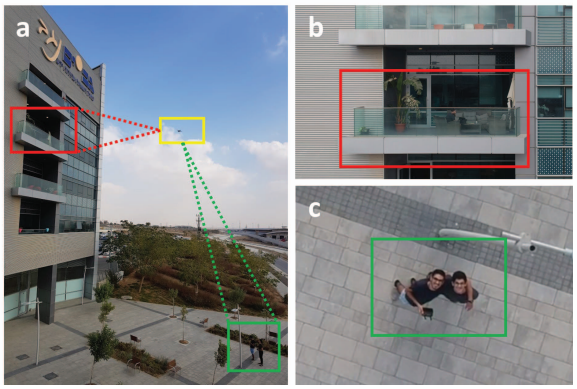


Fig. 1. Legitimate and illegitimate use of a drone from the same location: (a) A drone (framed in yellow), two people (framed in green), and a window of an organization (framed in red), (b) Illegitimate use of the drone camera to film an organization, and (c) Legitimate use for selfie purposes.

Geofencing methods for drone detection based on the drone's location have been suggested in recent years [13], [14], [15], [16], [17], [18], [19] as a means of detecting drones used for malicious purposes in restricted areas (e.g., in order to drop weapons and drugs into prison yards [20], smuggle goods and drugs between countries over borders [21], and crash on the White House lawn [22], [23]). However, the use of a traditional geofencing method as a means of detecting a privacy invasion attack in non-restricted areas (e.g., residential neighborhoods) will fail to distinguish between the legitimate use of a nearby drone and illegitimate use that invades a subject's privacy, a distinction that depends on the orientation of the drone's video camera rather than on the drone's location. Differentiation


DOI 10.1109/SP.2019.00051

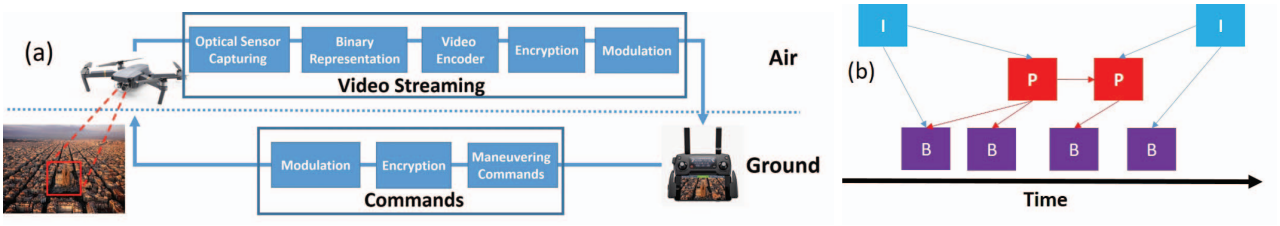


Fig. 2. (a) Secured FPV channel scheme, and (b) GOP structure - I, B, and P-frames.

between illegitimate and legitimate use of a drone can only be accomplished by determining the exact POI (point of interest) being streamed over the video channel and not according to the drone's location, as demonstrated in Figure 1.

The detection of a POI that is being video streamed by a drone/unmanned aerial vehicle (UAV) from an FPV (first-person view) channel has interested armies and militaries for many years. There are several known cases in which an army managed to detect the streamed POI from an intercepted unencrypted FPV channel of a rival's drone [24], [25], [26], [27]. However, there is only one known case in which a video stream was extracted from encrypted UAV traffic [28]. In general, detecting whether a target POI is being streamed from an intercepted encrypted video stream, without prior knowledge about the keys, remains a challenge.

In this paper, we present a new method that can detect whether a specific POI is being video streamed by a drone. We show that applying a periodic physical stimulus on a target/victim that is being video streamed by a drone causes a watermark to be added to the video traffic that is sent from the drone to its operator, a watermark that can be detected using interception. Based on this method, we present algorithms for (1) detecting a privacy invasion attack, and (2) locating a spying drone in space. We evaluate their performance using four commercial drones (DJI Mavic Air, Parrot Bebop 2, DJI Spark, and DJI Mavic Pro) in two use cases (when the victim is inside his/her house and when the victim is being tracked by a drone while driving his/her car) and show that a privacy invasion attack can be detected in about 2-3 seconds.

In this paper, we make the following contributions: First, we (1) present an improved method for classifying a suspicious transmission as an FPV channel and show that it can be used to distinguish between a drone and other moving IoT devices in just a few seconds. Then, we prove that the watermark added after applying a periodic physical stimulus (flickering) on an object for just two seconds enables us to (2) detect a spying drone (from a distance of 100 meters) and (3) identify its GPS coordinates and altitude (using a single Wi-Fi receiver), and can be used to (4) distinguish between the legitimate use of a drone that does not invade a subject's privacy and illegitimate use, even (5) when the target is moving. In contrast to the anti-drone market (expected to grow to a $1.85 billion [29] by 2024) which offers very expensive hardware solutions [30], we present a method that can be implemented using (6) inexpensive devices: a single Wi-Fi receiver and LED strips. In addition, we show how to (7) disguise the flickering so it will be invisible to the drone's operator. Finally, we (8) shatter the commonly held belief that the use of encryption to secure an FPV channel prevents a passive eavesdropper from extracting the POI that is being video streamed.

## II. First-Person View Channel

Modern drones provide video piloting capabilities (FPV channel), in which a live video stream is sent from the drone to the pilot (operator) on the ground, enabling the pilot to fly the drone as if he/she was onboard (instead of looking at the drone from the pilot's actual ground position). This allows a pilot to control a drone using a remote controller, as demonstrated in Figure 2a. A typical FPV channel is intended to be used for two purposes: **video streaming** using data that is captured by the drone's camera and sent to the pilot's controller, and **maneuvering and controlling** the drone using commands sent from the controller to the drone. In the following subsections, we describe the stages of video streaming.

### A. Video Encoding Algorithms

Video encoding [31], [32], [33], [34] begins with a raw image captured from a camera. The camera converts analog signals generated by striking photons into a digital image format. Video is simply a series of such images generally captured five to 120 times per second (referred to as frames per second or FPS). The stream of raw digital data is then processed by a video encoder in order to decrease the amount of traffic that is required to transmit a video stream. Video encoders use two techniques to compress a video: intra-frame coding and inter-frame coding.

**Intra-frame coding** creates an **I-frame**, a time periodic reference frame that is strictly intra-coded. The receiver decodes an I-frame without additional information. Intra-frame prediction exploits spatial redundancy, i.e., correlation among pixels within a frame, by calculating prediction values through extrapolation from already coded pixels, for effective delta coding (the process is described in Appendix XIV). **Inter-frame coding** exploits temporal redundancy by using a buffer of neighboring frames that contains the last M number of frames and creates a delta frame. A delta frame is a description of a frame as a delta of another frame in the buffer. The receiver decodes a delta frame using a previously received reference frame. There are two main types of delta frames: P-frames and B-frames. **P-frames** can use previous frames as data in the decompressing process and are more compressible

TABLE I
PURE WI-FI DRONES

| Manufacturer | Models | Wi-Fi Video Downlink | Distance (FCC compliance) | Weight | Price |
|---|---|---|---|---|---|
| DJI | Spark | 2.4/5.8 GHz | 4 KM | 300 g | $399 |
| | Phantom 3 SE | 2.4/5.8 GHz | 4 KM | 1236 g | $555 |
| Go-Pro | Karma | 2.4 GHz | 3 KM | 1006 g | $899 |
| Parrot | Bebop 2 FPV | 2.4/5.8 GHz | 2 KM | 500 g | $499 |
| | Disco | 2.4/5.8 GHz | 2 KM | 750 g | $499 |
| Xiro | Xplorer 2 | 2.4 Ghz | 1 KM | 1400 g | $1,499 |
| | Xplorer V | 2.4 GHz | 0.5 KM | 1202 g | $500 |
| Husban | H501A X4 | 2.4 Ghz | 0.4 KM | 500 g | $209 |
| | H507A | 2.4 Ghz | 0.3 KM | 450 g | $109 |

than I-frames. **B-frames** can use both previous and upcoming frames for data reference to obtain the greatest amount of data compression (the process is described in Appendix XIV).

The order in which I, B, and P-frames are arranged is specified by a GOP (group of pictures) structure. A GOP is a collection of successive pictures within a coded video stream. It consists of two I-frames, one at the beginning and one at the end. In the middle of the GOP structure, P and B-frames are ordered periodically. An example of a GOP structure can be seen in Figure 2b. Occasionally B-frames are not used in real-time streaming in order to minimize delays.

Video compression techniques were integrated into the MPEG-1 standard in the 1990s and boosted the transmission rate from 1.5 Mbps (MPEG-1) to 150 Mbps (MPEG-4). Naturally, integrating these techniques into the protocol creates a variable bitrate (VBR) in the transmission of a video which is influenced by changes between frames and the content of the frame itself. A frame that can be represented as a set of prediction blocks of a similar neighboring frame (that belongs to the same GOP) requires a smaller amount of data to be represented. On the other hand, a frame with less similarity to other neighboring frames (e.g., as a result of the movement of several objects) necessitates that a larger amount of data be represented as a set of prediction blocks of other frames.

### B. Wi-Fi FPV

There are two types of technologies dominating the FPV market: Wi-Fi FPV and analog FPV [35]. Wi-Fi FPV is, by far, the most popular method used to include FPV in budget RC drones (according to [35], [36]) because: (1) any Android/iOS smartphone (or tablet) on the market can be used to operate the drone; (2) the only additional hardware required is a Wi-Fi FPV transmitter (which is connected to the camera of the drone), instead of an additional controller with a screen that is equipped with a dedicated radio transceiver which is required by other types of FPV (e.g., 2.4/5.8 GHz analog FPV); (3) drone manufacturers were able to boost the Wi-Fi FPV drone flight range to four kilometers using dedicated hardware [37], [38], [39]; and (4) Wi-Fi FPV drones support 4K resolution. Some types of drones are considered pure Wi-Fi FPV drones (e.g., DJI Spark, DJI Phantom 3 SE, Parrot Bebop 2), and other kinds contain Wi-Fi FPV along with their dedicated analog FPV (e.g., DJI Mavic pro, DJI Mavic Air). Almost every FPV-enabled drone selling for less than $100 uses Wi-Fi FPV [35], and there are dozens of kinds of Wi-Fi FPV drones available for purchase [40], [41], [42], ranging from $30 to hundreds and thousands of dollars.

TABLE II
INFORMATION LEAKAGE FROM VBR STREAMS - RELATED WORK

| Transmitter | Purpose | Publication | Required Stream Duration | Analyzed Protocols | Interception |
|---|---|---|---|---|---|
| Video Hosting Services (Netflix, YouTube, etc..) | Classify video stream | [43] - USENIX 07<br>[44] - ISC 10<br>[45] - USENIX 17<br>[46] - CODASPY 17<br>[47] - CCNC 16 | Minutes | DASH | Internal |
| IPTV | Classify video stream | [48] - GLOBECOM 08 | Minutes | RTP | Internal |
| IP Camera | Lights on/off, Hand movement Detecting hidden camera | [49] - GLOBECOM 15<br>[50] - MobiSys 18 | Immediate | RTP | Internal |
| PC | Language extraction, Phrase detection, Transcripts | [51] - USENIX 07<br>[52] - S&P 08<br>[53] - S&P 11 | Varies | VoIP | Internal |
| **Drone** | **Detecting streamed POI** | S&P 19 | **2 seconds** | **RTP** | **External** |

In order to boost the flight range of Wi-Fi FPV drones, manufacturers sell a dedicated controller (without a screen) that broadcasts/receives the signal with a much more powerful transceiver (25-27 dB) than the one that is integrated into smartphones/tablets (10-15 dB). When Wi-Fi communication is sent from the drone to the smartphone via the dedicated controller and vice versa, the controller is used as a proxy between the drone and the smartphone (which is used mainly as a screen). In this study, we focus on Wi-Fi FPV drones. Table I lists various types of **pure commercial Wi-Fi FPV drones** and their properties and prices.

Wi-Fi communication between the drone and the controller (dedicated controller/smartphone) is sent over a secured access point (WPA 2) that is opened by either the drone or the controller (both parties are connected to the access point) and follows the OSI model. The video that is captured by the drone camera is streamed to its controller using real-time end-to-end media streaming protocols (RTP) through UDP packets. The last layer of encryption is applied on layer 2 of the OSI model according to IEEE 802.11 standards.

## III. RELATED WORK

In this section, we describe: (1) methods that exploit information leakage of an encrypted video stream to extract insights about the stream, and (2) methods for nearby drone detection. In the area of **video hosting services** and **IPTV**, several studies exploited variable bitrate (VBR) protocols of video streams to classify a video stream sent from a video hosting service (e.g., YouTube, Netflix , etc.) [45], [47], [46], [43], [54], [44]. In the area of **VoIP**, several studies showed that VBR leakage in encrypted VoIP communication can be used for the detection of the speaker's language [51] and phrases [52], and to extract conversation transcripts [53]. In terms of the attack model, the abovementioned studies [45], [47], [46], [43], [54], [44], [51], [52], [53] require the attacker to: (1) create a large dictionary of video streams that must be classified before classification is applied, (2) intercept a few minutes of video stream in order to obtain good results, and (3) compromise a computer in the targeted network in order to capture network traffic. Our study does not require the abovementioned conditions, since we watermark a video stream and observe the changes instead of comparing the video stream to an existing dictionary. In addition, only a

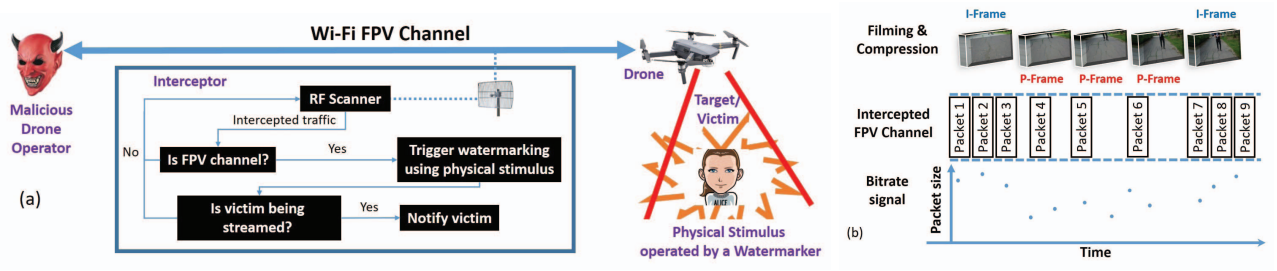


Fig. 3. (a) Detection scheme, and (b) Creating a Bitrate Signal using interception.

few seconds of interception are required to determine whether the captured video stream is watermarked or not. In research using **IP cameras**, a recent study analyzed VBR video streams and showed that it is possible to detect hand movement and ambient light changes [49]. Two studies which were performed in parallel to our work showed a method to detect hidden Wi-Fi cameras using probing [50] and traffic analysis [55]. Table II summarizes studies on information leakage from VBR streams.

In the area of **drone detection**, various methods were introduced over the last few years to detect a nearby consumer drone. These methods are widely used for geofencing purposes such as detecting the presence of a drone in restricted areas (e.g., drones that were used for purposes of dropping weapons and other contraband into prison yards [20], smuggling goods and drugs between countries over borders [21], and crashing on the White House lawn [22], [23]). **Active radar** is a traditional method of detecting drones, however the detection of small consumer drones requires expensive high frequency radar systems [13], in order to prevent drones from evading such systems [23]. Two other methods suggested using **passive radar** (i.e., a Wi-Fi receiver) to detect a consumer/civilian drone controlled using Wi-Fi signals. The first method [18], [56] analyzes the protocol signatures of the Wi-Fi connection between the drone and its controller. The second method [19] analyzes the received signal strength (RSS) using a Wi-Fi receiver. Several studies suggested **computer vision techniques** that use a camera to analyze motion cues [14], [15], in order to detect a drone. However, these methods suffer from false positive detection due to: (1) the increasing number of drone models, and (2) similarities between the movements of drones and birds [15]. Several studies used **sound techniques** to analyze the noise of the rotors captured by microphones [16], [15]. However, very expensive equipment is required in order to address the challenges arising from the ambient noise and the distance between the drone and the microphone [16]. A **hybrid method** was suggested by [17], however this method is very expensive to deploy. A recent study [30] from DEF CON 25 that reviewed 33 commercial products that implement the abovementioned methods [13], [14], [15], [16], [17], [18], [19] called these systems "overkill" due to their expensive price compared to the price of a drone.

Populated areas are no longer considered restricted for drones; in 2017, the president of the United States signed a memo allowing drones to fly in urban areas [8]. As a result, applying geofencing methods as a means of detecting privacy invasion attacks is irrelevant. All of the methods for drone detection described in this section [13], [14], [15], [16], [17], [18], [19] fail to distinguish between the act of taking a selfie and spying on an organization, as demonstrated in Figure 1. In contrast to the abovementioned drone detection mechanisms, our method does not have this weakness. In this research, we demonstrate a method for determining whether a specific POI is being filmed, that comparing to other commercial drone detection mechanisms does not require an expensive hardware.

## IV. Adversary Model & Detection Scheme

There are four parties involved in a privacy invasion attack perpetrated by drones: a malicious operator that controls the drone, a target/victim, an interceptor, and a watermarker. In this study, we define the **malicious operator** as any person who uses a drone for the illegitimate purpose of streaming a victim for any reason. We assume that the **malicious operator** is using a Wi-Fi FPV drone and is located up to four kilometers from the victim. We consider the **target/victim** any subject, building, or facility that is of interest to a malicious operator and being video streamed by the drone, and consider the **interceptor** an automated model (described in Algorithm 1) for privacy invasion attack detection that runs on a PC/laptop/smartphone with a connected RF scanner (e.g., network interface card, software-defined radio) and an adequate antenna and amplifier. The **watermarker** is a laptop/microcontroller that controls a device that can launch a periodic physical stimulus (flickering) and turn it on and off. In practical deployment, the victim/target may decide to activate the physical stimulus (flickering) only when needed, e.g., when a drone is detected (based on the drone's RF transmission) and it is unclear whether the drone is being used to spy on the victim. In addition, flickering can be launched using a variety of devices, including LED strips, smart bulbs, a portable projector, smart film, and other devices that can be programmed to change their color and force pixel changes between consecutive frames. The watermarker can be deployed inside or outside the target house/car. In cases in which the watermarker is deployed inside a house/car, infrared lighting can be used for flickering, so it will be invisible to people in the house/car (in Section IX we show that a drone's camera is sensitive to infrared lighting, meaning that infrared flickering can be used to watermark the

FPV channel). In cases in which the watermarker is deployed outside the target house/car, there is likely no need for an additional device given existing visible programmable lighting and its infrastructure. One example of this involves the use of exterior building lights, commonly used these days for decoration in many buildings (residential, offices, government) and facilities (stadiums); often such existing lighting uses a changing lighting pattern which can be leveraged.

**Algorithm 1** Detecting Privacy Invasion Attack

```
1:  procedure UNDERDETECTION?
2:      enableMonitoringMode()
3:      suspiciousNetworkList = getNetworksInRange()
4:      for (network : suspiciousNetworkList) do
5:          if isFpvChannel(network) then
6:              // Draw stimulus frequency and duration
7:              fs = getRandomFloat(1,6)
8:              duration = getRandomFloat(1,10)*1000
9:              // Store stimulus beginning time
10:             time = currentTimeInMillis()
11:             // Launch watermarker and determine spying
12:             watermarker(fs,duration)
13:             if isTargetFilmed?(network,fs,time) then
14:                 notifyVictim()
```

Figure 3a presents the proposed target detection scheme and the parties involved. The interceptor's model for detecting a privacy invasion attack is presented in Algorithm 1. First, suspicious transmissions are intercepted (line 3) and extracted to a *suspiciousNetworkList*. For each suspicious transmission *network*, we apply the Boolean function *isFpvChannel* to determine whether the *network* is an FPV channel (line 5). If the *network* is classified as an FPV channel, the algorithm triggers a periodic physical stimulus at a given frequency for a given *duration* (in milliseconds) by calling the method *watermarker*. Finally, the method *isTargetFilmed?* is called to determine whether the FPV channel *network* is being used to film the target/victim, and a notification is sent to the victim upon detection of a privacy invasion attack (line 14).

## V. Interception & Creating Bitrate Signal

We used four types of drones in our experiments: two pure Wi-Fi FPV drones (DJI Spark and Parrot Bebop 2) and two drones which support Wi-Fi and analog FPV (DJI Mavic Pro and DJI Mavic Air). These drones were among the top 10 most sold drones when this research was performed [57]. All of the drones' access points are secured by WPA 2, in order to guarantee that the transmitted video stream is only available for watching by the connected parties (controller).

We applied interception as follows: we used a laptop (Dell Latitude 7480) that runs Kali Linux with a standard NIC (Intel Dual Band Wireless-AC 8265 Wi-Fi) as the Wi-Fi receiver.

1) We enabled "monitor mode" on the Wi-Fi receiver (used by the interceptor) using Airmon-ng [58].

2) We detected Wi-Fi networks within the range of the Wi-Fi receiver used.

3) We used a Wi-Fi sniffer (Airodump-ng) [59] to intercept packets of a specific Wi-Fi network.

We consider this process external interception, i.e., we intercept a specific network's transmissions without being connected to the network. By intercepting packets this way, we cannot observe encrypted layers of captured packets (since we do not have the required key). The interception range can be extended to detect transmissions from drones up to a few kilometers from the victim using additional hardware such as a dedicated antenna or amplifier, however we did not use additional hardware to extend the range in this study.

The process of creating an intercepted bitrate signal from the captured packets is as follows:

4) From each captured packet we extracted the following information: (a) Packet's arrival time in nanoseconds - information added to each captured packet by Airodump-ng, and (b) Packet's size - information that was extracted from the unencrypted meta-data (PLCP header) from the data link layer.

5) Finally, we changed the signal's resolution from nanoseconds to milliseconds by aggregating all packets captured in each millisecond.

The two bash scripts that implement stages 1-3 and 4-5 are presented in Appendix XV. In the rest of this paper we refer to the output of this process as the **intercepted bitrate signal**. The FFT graphs and spectrograms (power spectral density) presented in this paper were extracted from the intercepted bitrate signal. Figure 3b depicts this process.

## VI. Detecting FPV Channel

In this section, we show how a suspicious transmission can be classified as an FPV channel and how to extract details about its quality. We present an improved passive radar method that relies on two detection stages: (1) moving object detection, and (2) video channel detection. Two additional benefits from using our method are that unlike similar passive radar methods, we can distinguish between a drone and other moving IoT devices, and we are able to extract the FPV channel quality (FPS and resolution) as well.

### A. Detecting Moving Objects

Passive radar methods for classifying an intercepted transmission as an FPV channel were suggested by [18], [56], [19]. These methods analyzed RSSI (received signal strength indicator) measurements that are added by a static radio receiver (e.g., NIC, SDR, etc.) in order to detect a moving drone. These studies presented classification methods based on unique RSSI patterns that are the result of a drone's movement. However, these studies did not validate the quality of their methods against other ubiquitous moving IoT devices that transmit radio signals such as robotic vacuum cleaners, smartwatches, smartphones, etc. In this paper, we show that a drone's RSSI behavior can be similar to other moving IoT devices and argue that moving object detection is not adequate for distinguishing a drone from other moving IoT devices.

*1) Experimental Setup:* In this experiment, a laptop was placed on the ground and used as passive radar. One of the authors walked a distance of 25 meters from the laptop for 100 seconds (at a very slow speed) with a smartphone (Galaxy S8) in his pocket and a smartwatch (LG smartwatch Urbane 2nd

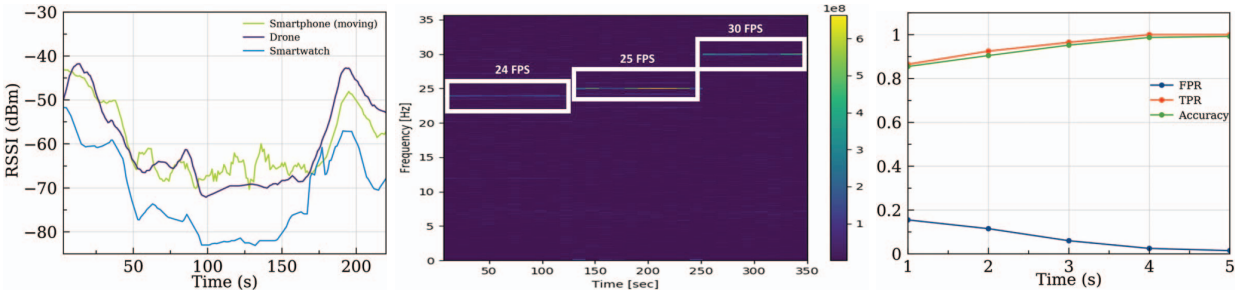


Fig. 4. (a) Similar RSSI patterns obtained from smartwatch, smartphone, and drone, (b) A spectrogram (power spectral density) of the intercepted bitrate signal of a drone access point when the FPS was changed, and (c) Classification results of Algorithm 2.

Edition) on his wrist. After 100 seconds, he returned to the laptop on the same path. We flew a drone (DJI Mavic Pro), at an altitude of two meters from the ground, along the same path (the operator stood near the laptop). In addition, we intercepted the traffic sent by the smartwatch, smartphone, and drone via the laptop (as described in Section V).

*2) Results:* Figure 4a presents the RSSI measurements from transmissions sent from the drone, smartwatch, and smartphone as they were captured by the laptop using external interception. As can be seen from the results, the RSSI measurements and patterns are similar for the smartphone, smartwatch, and drone. This experiment proves that relying on moving object detection methods as a means of classifying an FPV channel using RSSI analysis requires an additional stage to filter out moving IoT devices that are not drones.

### B. Detecting Video Stream & Extracting its Quality

In this subsection, we present a new method for classifying an intercepted transmission as a video stream that can extract details about the video stream's quality (FPS and resolution).

*1) Experimental Setup:* We conducted the following experiment using the Bebop Parrot 2 which supports three FPV transmission rates (24, 25, and 30 FPS). We positioned the drone on the ground and used its application to change the FPS rate every two minutes (from 24 FPS to 25 FPS and then from 25 FPS to 30 FPS). We intercepted the traffic that was sent from the drone and created the intercepted bitrate signal (as described in Section V).

*2) Results:* As can be seen from the spectrogram extracted from the intercepted bitrate signal (presented in Figure 4b), the power around each of the FPV transmission frequencies (FPSs) outperforms any other frequency. Video streams can be detected by comparing the frequency with the strongest magnitude of the intercepted bitrate signal to known FPS rates used by video streams. By detecting the FPS of a captured video stream, we can also use the intercepted bitrate signal to infer the resolution of the video stream, and find the resolution for the H-264 standard published in [60], [61], [62].

### C. Classifying FPV Channels

Algorithm 2 presents a method for classifying FPV channels based on the observations mentioned above. It receives a suspicious intercepted *network*, and it classifies the network as an FPV channel if a connected MAC address was found to be a moving object (line 5) that transmits traffic at known drone FPS video rates. (line 10). In prior research, methods to classify an IoT device as a moving object based on RSSI analysis have been applied to detect moving smartphones [63] and smartwatches [64]. The distance between a moving radio transmitter and a static receiver can be derived from RSSI measurements, and this has been used for indoor localization of smartphone users [63]. However, we are interested in detecting moving objects, a task which is much simpler than localizing objects. Therefore, we implemented an algorithm for object detection suggested in a prior study that is based on RSSI measurements obtained from the receiver [65].

**Algorithm 2** Classifying an FPV Channel

```
1:  procedure ISFPVCHANNEL?(network,time)
2:      frequency = 70
3:      for (macAddress : network) do
4:          //Detecting Moving Objects
5:          if (isMovingObject(macAddress)) then
6:              bitrate[] = extractBitrateSignal(macAddress)
7:              fft [] = FFT(bitrateArray,frequency)
8:              index = frequencyWithStrongestMagnitude(fft)
9:              //Detecting video channel
10:             if (index==24 || index==25 || index==30) then
11:                 return true
12:     return false
```

*1) Experimental Setup:* We evaluate the performance of Algorithm 2 given a device that was already found to be a moving object; therefore, we are aiming to determine how much time it takes to classify a moving object as a drone. In order to accomplish this, in this experiment we intercepted 1000 seconds of traffic (as described in Section V) from the Bebop Parrot 2 and DJI Spark (500 seconds from each drone) while they flew in the air (at an altitude of 30 meters). We also intercepted 1000 seconds of traffic from moving IoT devices as follows: 290 seconds from a robotic vacuum cleaner (Roborock S50) as it was performing routine home cleaning, 290 seconds of traffic from a smartwatch (LG G W150), and 420 seconds of traffic from a smartphone (OnePlus 5). The smartwatch was worn on the wrist of a person walking with a smartphone in his pocket.

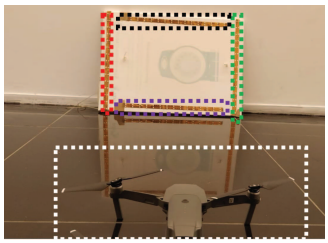

Fig. 5. Experimental setup - drone (framed in white) located in front of a white board with four LED strips (framed in red, green, purple, and black)

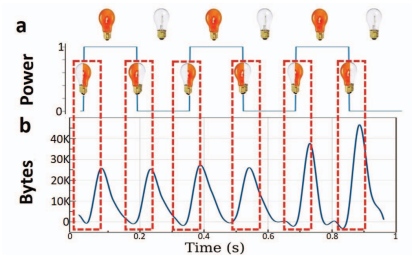


Fig. 6. influence of flickering: (a) a bulb according to a 3 Hz square wave, (b) six bursts from one second of the intercepted bitrate signal of a drone that streams a 3 Hz flickering LED strip.

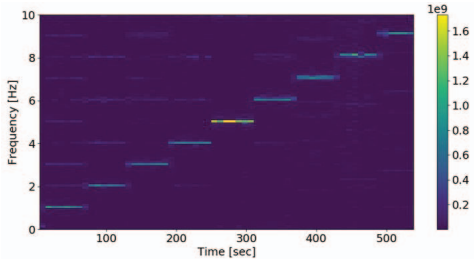


Fig. 7. A spectrogram (power spectral density) of the intercepted bitrate signal of a drone located in front of an LED strip that flickers for one minute at frequencies of 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5 Hz.

*2) Results:* We obtained the intercepted bitrate signals for each of the devices and divided the intercepted signals into smaller signals (each signal was five seconds long). This process resulted in 200 intercepted bitrate signals obtained from drones and 200 intercepted bitrate signals obtained from other moving IoT devices. Figure 4c presents the results (accuracy, TPR, and FPR) after applying Algorithm 2 on the data with various interception windows (1-5 seconds). As can be seen in Figure 4c, once a device has been identified as a moving object it takes just four seconds to classify its transmissions as an FPV channel. After four seconds, accuracy of 0.99 and a true positive rate (i.e., drone detection rate) of 1.0 is obtained. The confusion matrices from this experiment are presented in Table VI in Appendix XVII.

In the remainder of the paper, we assume that (1) a suspicious transmission can be classified as an FPV channel by applying Algorithm 2 on the intercepted bitrate signal (extracted as described in Section V), and (2) the quality of the FPV channel (FPS and resolution) can be extracted from the intercepted bitrate signal.

## VII. Watermarking FPV Channel

In this section, we assess the influence of a periodic physical stimulus which is applied to a target/victim that is being streamed by a drone, by analyzing the intercepted bitrate signal. We consider the algorithm that controls the periodic physical stimulus a **watermarker** (described in Algorithm 3).

**Algorithm 3** Physical Watermarking

```
1: procedure WATERMARKER(frequency,duration)
2:     onOffDuration = 1000/(2*frequency), N = duration/onOffDuration
3:     for (i = 0; i < N; i++) do
4:         if (i%2 == 0) then
5:             turnOnPhysicalStimulus()
6:         else turnOffPhysicalStimulus()
7:         sleep(onOffDuration)
8:     turnOffPhysicalStimulus()
```

Algorithm 3, which runs from a computer/controller, controls a device that creates a periodic stimulus (e.g., flickering) whose frequency can be programmed. The algorithm receives two parameters: *frequency* (amount of stimuli per second) and *duration* (in milliseconds). The algorithm creates a square wave at the given *frequency*, and based on this, turns a physical stimulus on and off for the specified *duration*.

*1) Experimental Setup:* We attached four LED strips, each of which was connected to a microcontroller, to a white board (as can be seen in Figure 5) and performed the following experiment. We programmed the microcontroller that was connected to the top LED strip (framed by black dots in Figure 5) so that it would flicker at various frequencies (0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5 Hz) for one minute per frequency. We then positioned a DJI Mavic Pro [66] consumer drone in front of the board at a distance of 1.5 meters (as can be seen in Figure 5), intercepted the traffic sent from the drone, and created the intercepted bitrate signal (as described in Section V).

*2) Results:* Figure 6b presents one second from the intercepted bitrate signal that was captured during the time that the top LED strip flickered at 3 Hz (see Figure 6a). As can be seen from Figures 6a and 6b, a 3 Hz flickering LED strip creates a 6 Hz phenomena within the intercepted bitrate signal by producing six bursts per second. Each time the LED strip was turned on/off a larger amount of data was sent from the drone which is expressed as a burst of bytes in the time domain. This is due to the fact that a larger amount of P-frames was required to encode the changing macroblocks (changing pixels) compared to an unchanging video stream. Figure 7 presents a spectrogram that was produced from the intercepted bitrate signal of the entire experiment. As can be seen, frequencies of 1-9 Hz were influenced by this experiment. The flickering LED watermarks the frequency of the intercepted bitrate array exactly at the point which is twice its flickering frequency. We concluded that the flickering object's frequency can be detected using this method by analyzing FPV traffic, and moreover, that it can even be used as a means of detecting whether the drone's camera is being used to stream a flickering object when the channel is encrypted. However, since the slowest FPS rate among the four drones supports just 24 FPS, the maximum frequency of a flicker that can be detected by analyzing the intercepted bitrate signal is limited to a 6 Hz flickering rate that watermarks the 12 Hz frequency of the intercepted bitrate array (Nyquist frequency). In the rest of

the paper we measure the influence of a flickering object on the intercepted bitrate signal. We refer to the ratio between the magnitude after a flicker was triggered (noise) and the magnitude before a flicker was triggered (signal) around the influenced frequency as the signal to noise ratio (SNR).

## VIII. Locating Drone in Space

In this section we first show how to calculate the distance and angle between the watermarker and the drone. Then, we leverage our findings to create a drone locating model and evaluate its performance.

### A. Detecting Drone's Distance

*1) Influence of Distance on SNR:* Here we show the influence of distance on a fixed sized flickering object.

*Experimental Setup*: We aimed a portable projector [67] at the exterior wall of a building; the projector was used to project a video of a flicker (3.5 Hz) onto a specific portion of the wall (a rectangle 2.5×2.5 meters in size). We flew a DJI Mavic Pro various distances (10m, 20m, .., 90m, 100m) from the flickering rectangle. As in real surveillance, we zoomed the drone's camera (2x) on the flickering rectangle (that was considered as the target in this experiment). A laptop was placed near the projector to intercept the traffic sent from the drone during the experiment.

*Results*: Figure 8 presents the SNR as a function of distance. As can be seen, using a rectangle of 2.5×2.5 meters leaves a watermark with an SNR that is greater than one from a distance of 50 meters. Since the amount of pixels that are changed as a result of a flickering object is greater from a shorter distance, many more macroblocks are changed (as a result of the flickering) and the SNR is greater; in contrast, greater distances cause the flickering object to be smaller and result in changing fewer macroblocks and a lower SNR. However, the new DJI Mavic 2 Zoom supports 4x zoom, which has twice the zoom capacity of the drone we used. The DJI Mavic 2 Zoom can be detected from a greater distance, because a fixed size object that is captured by a drone with 2x zoom from a distance of 50 meters can be captured by a drone with 4x zoom from a distance of 100 meters [68].

*2) Extracting Drone's Distance:* We aimed to extract the distance between the drone and the flickering object. In order to do so, we must first learn the effect of changing the percentage of captured pixels on the traffic.

*Experimental Setup*: We placed the DJI Mavic Pro (configured to 24 FPS and 720p) in front of a laptop monitor located 0.5 meters away. We conducted 11 experiments using this setup, and in each experiment a flickering rectangle (at 3Hz) of a different size was presented in the middle of the monitor (10%, 20%, ..., 90%, 100%). In each experiment, we intercepted traffic (as described in Section V) sent from the drone. We obtained the 11 intercepted bitrate signals and applied FFT to each of them.

*Results*: As can be seen by the SNR that was computed from the magnitudes around 6 Hz in the experiments (presented in Figure 9), the SNR increases as a function of the percentage of changing pixels. By increasing the size of the rectangle, we increased the amount of macroblocks that were changed between consecutive frames. Encoding a larger amount of macroblocks increases the bitrate which improves the SNR. Based on the results of the experiments, we compared the performance of four regression methods (polynomial, linear, exponential, and logarithmic) to predict the percentage of changing pixels given a specific magnitude.

Table III presents the residual sum of squares (RSS) and coefficient of determination ($R^2$) of the percentage of changing pixel prediction for each regression method. The function of the polynomial regression that yielded the best prediction result among the tested methods is presented in Equation 1:

TABLE III
Error of Distance Prediction Based on Regression Methods

| Method | RSS | $R^2$ |
|---|---|---|
| Polynomial Regression | 56 | 0.994 |
| Linear Regression | 464 | 0.957 |
| Exponential Regression | 581 | 0.947 |
| Logarithmic Regression | 2523 | 0.770 |

$$\textbf{\% Changing Pixels (SNR=s)} = 1.12 - 3.14\times10^{-7}s^4 + 6.96\times10^{-5}s^3 - 5.12\times10^{-3}s^2 + 1.87\times10^{-1}s \quad (1)$$

By applying a physical stimulus using a square shaped flicker at a specific frequency, the interceptor can calculate the height and width of the flickering object (in terms of pixels) in a frame (picture) by applying the following steps:

1) Determining the $FPVresolution$ of the FPV channel (as explained in Section VI).
2) Triggering a physical stimulus using a square flickering at a specific frequency (e.g., 3 Hz).
3) Calculating the percentage of changing pixels from the intercepted bitrate signal using Equation 1.
4) Inferring the amount of $changingpixels$ from the $FPVresolution$.
5) Inferring the $height$ and $width$ (in terms of pixels) of the flickering object in a frame.

For a square flickering object we conclude that the:

$$\textbf{height (in pixels) = width (in pixels)} = \sqrt{\%ChangingPixels(m) \times FPVResolution} \quad (2)$$

By calculating the height and width (in pixels) of a flickering object (for which the real size is known), the interceptor can infer the distance between the drone's camera to the flickering object [69] from the intercepted FPV channel (for which the resolution was also determined) using Equation 3:

$$\textbf{Distance (mm)} = factor(p) \times factor(d) \quad (3)$$

$factor(p)$ is defined as follows (Equation 4):

$$\textbf{factor(p)} = \frac{realObjectHeight(mm) \times imageHeight(pixels)}{objectHeight(pixels)} \quad (4)$$

The parameters required to calculate $factor(p)$ have already been calculated. $factor(d)$ is drone dependent and defined as follows (Equation 5):

$$\textbf{factor(d)} = \frac{f(mm)}{sensorHeight(mm)} \quad (5)$$

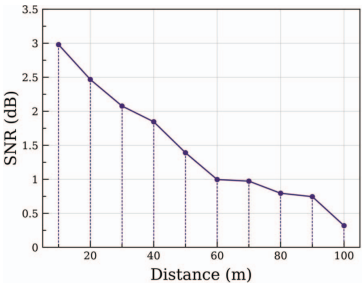


Fig. 8. SNR - magnitudes around 7 Hz as a function of the distance between a drone and a flickering object.

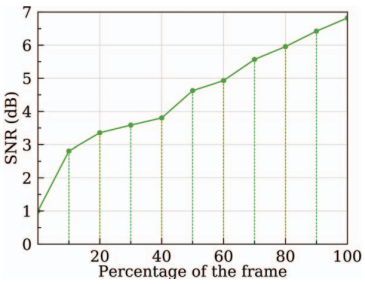


Fig. 9. SNR - magnitudes around 6 Hz as a function of the percentage of changing pixels.

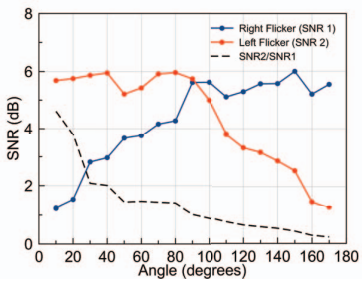


Fig. 10. SNR - magnitudes around 7 Hz (SNR1) and 6 Hz (SNR2) as a function of the angle at the midpoint between two flickering LED strips.

The parameters $f(mm)$ and $sensorHeight(mm)$ are published online in the specifications for each of the drones [66], [70], [71]. The $sensorHeight(mm)$ for each drone is 1/2.3" (11.0434783 millimeters). The lens' length of each drone varies between 24 and 35mm, so $factor(d)$ is in the range of (Equation 6):

$$0.31 < \textbf{factor(d)} < 0.46) \tag{6}$$

Based on Equations 6 and 3, we can see that the distance between the drone and the flickering object varied in the range of (Equation 7):

$$0.31 \times factor(p) < \textbf{Distance (mm)} < 0.46 \times factor(p) \tag{7}$$

For $factor(d) = 0.385$, we obtain a maximum error of $0.075 \times factor(p)$ for the distance estimation. If the exact type of drone can be detected from the intercepted FPV channel (e.g., according to a unique FPS rate), the computed distance is accurate.

### B. Detecting Drone's Angle

Next, we aimed to investigate the effect of the angle between the flickering object and the drone.

*1) Experimental Setup:* Using the white board presented in Figure 5, we programmed the microcontrollers of the LED strip on the left to flicker at 3 Hz and those of the LED strip on the right to flicker at 3.5 Hz simultaneously. We positioned the drone at 17 different angles (10°, 20°, ..., 160°, 170°). The distance between the drone and the middle of the strips was the same for each of the 17 positions. We intercepted the traffic sent from the drone and created the intercepted bitrate signal (as described in Section V).

*2) Results:* The SNR around the frequencies of 7 Hz (referred to as SNR1, i.e., the SNR around the frequency that is influenced by the left flickering LED) and 6 Hz (referred to as SNR2, i.e., the SNR around the frequency that is influenced by the right flickering LED) is presented in Figure 10. As can be seen, the SNR at those frequencies behaves as a mirror around 90°(due to the fact that flickering objects of the same size have the same effect). However, the magnitude of the LED strip that was far from the camera when the drone was located diagonal to the white board decreases, since a flickering object that is farther away is smaller compared to a flickering object that is closer. The ratio between SNR2 and SNR1 ($\frac{SNR2}{SNR1}$) is also presented in Figure 10. As can be seen, the ratio decreases as the angle increases. We compared the performance of four regression methods (polynomial, linear, exponential, and logarithmic) to predict the angle between the drone and the middle of the two LED strips, based on the ratio between SNR2 and SNR1. Table IV presents the residual sum of squares (RSS) and coefficient of determination ($R^2$) of angle prediction for each regression method. The function obtained based on exponential regression is presented in Equation 8:

TABLE IV
ERROR OF ANGLE PREDICTION BASED ON REGRESSION METHODS

| Method | RSS | $R^2$ |
|---|---|---|
| Exponential Regression | 979 | 0.976 |
| Polynomial Regression | 1062 | 0.973 |
| Logarithmic Regression | 1450 | 0.964 |
| Linear Regression | 10011 | 0.754 |

$$Angle(SNR1, SNR2) = 192.72 * e^{-0.71 * \frac{SNR2}{SNR1}} \tag{8}$$

### C. Locating Drone's Location

In Subsection VIII-A, we obtained a formula to detect the distance *r* between a drone and a flickering object. In Subsection VIII-B, we obtained a formula to detect the angle of a planner that spreads from a drone to the middle of two parallel flickering objects attached to a white board. Figure 11 leverages our findings for locating a drone in space using a white board (framed in yellow) with two pairs of parallel flickering objects. As can be seen in the figure, the objects that comprise the first pair of parallel flickering objects (marked with red dots) are located at the top and bottom of a rectangle board (framed in yellow) and spread a red planner with angle $\phi$ along the x-axis. The objects comprising the second pair of parallel flickering objects (marked with green dots) are located on the left and right sides of the same board (framed in yellow), and they spread a green planner with angle $\phi$ along the z-axis. We consider ($r$,$\theta$,$\phi$) spherical coordinates that give the relative location of a drone from a rectangle

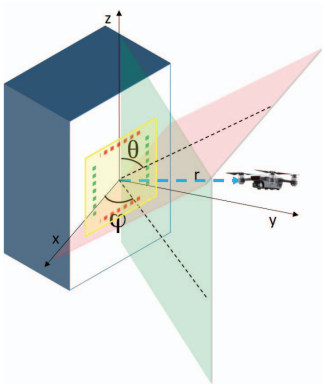


Fig. 11. Locating a drone based on four flickering LED strips that creates r, $\theta$, and $\phi$.

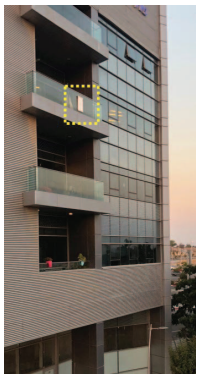

Fig. 12. A board with four flickering LED strips installed on a balcony.

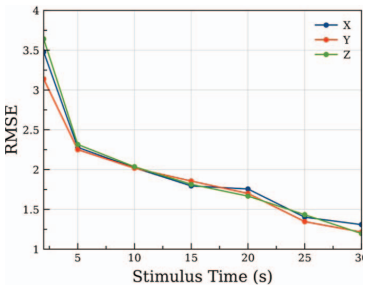


Fig. 13. Root mean square error (RMSE) results of the locating drone experiment as a function of the amount of time that flickering was applied.

board that contains two pairs of parallel flickering LED strips. The Cartesian coordinates (x,y,z) can be retrieved from the spherical coordinates ($r$,$\theta$,$\phi$) using known formulas [72].

*1) Experimental Setup:* In order to evaluate the accuracy of a mechanism for locating a spying drone in space according to our formulas, we conducted the following experiment. The white board presented in Figure 5, which has an LED strip connected to a microcontroller on each edge, was attached to a balcony located on the third floor of a building (21 meters from the ground) so that the side of the board with the LED strips was facing outward, as can be seen in Figure 12. We flew the DJI Mavic Pro drone between 30 different locations at various altitudes and distances from the balcony while the drone conducted a privacy invasion attack against the organization (i.e., the drone's video camera streamed the balcony). The exact 30 locations, as measured by the DJI-Go application (longitude, latitude, and altitude), are listed in Table VII (Appendix XVIII) and marked by blue dots in Figure 14. Each of the four LED strips was programmed to flicker at a different frequency for 30 seconds. We intercepted the drone's FPV channel at each of the 30 locations and extracted 30 bitrate signals.

*2) Results:* Using the previously mentioned formulas, we computed the spherical coordinates ($r$,$\theta$,$\phi$) for each of the locations and computed the Cartesian coordinates (x,y,z) from

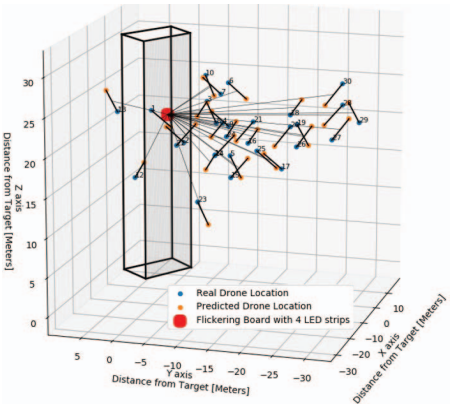


Fig. 14. Results of the locating a drone in space experiment when applying a physical stimulus for two seconds.

the spherical coordinates according to [72]. Based on the computed Cartesian coordinates, we calculated the GPS coordinates (latitude, longitude) and altitude. Finally, we computed the error between the actual location and the predicated location. Figure 13 presents the mean square error (RMSE) results for the x, y, and z-axes as a function of the amount of time the physical stimulus was applied. As can be seen, the accuracy along each axis is improved from an average error of 3.5 meters (by applying flickering for two seconds) to an average error of 1.2 meters (by applying flickering for 30 seconds). The actual locations and predicted locations (by applying two seconds of flickering) are presented in Figure 14 and Table VII (Appendix XVIII). Considering the fact that the measurements of the 30 real locations were obtained from the drone's GPS (using its application) and the known average error of GPS devices (a 4.9 meter radius in the open sky [73]), we can accurately locate a spying drone in space using four flickering LED strips and a single Wi-Fi receiver by applying flickering for two seconds.

## IX. Hiding the Physical Stimulus

In this section, we investigate whether a physical stimulus can be produced in such a way that it is undetectable to the human eye. An undetectable physical stimulus should fulfill the following three requirements: (1) it should be undetectable by direct observation by the drone's operator via the naked eye, (2) it should be undetectable by indirect observation by the drone's operator via the controller screen, and (3) it should watermark the FPV channel. One method that was considered takes advantage of the eye's limited ability to capture infrared and UV frequencies. We tested the influence of using infrared LEDs as a means of creating a physical stimulus. As can be seen in Figure 15a, the drone's camera is sensitive to infrared frequencies and can capture them; therefore, this method does not meet the second requirement. However, infrared flickering can be used in cases in which the watermarker is deployed inside a house/car, and there is no need to hide the flickering from the drone's operator in order to create invisible flickering that will not disturb the people in the house/car.

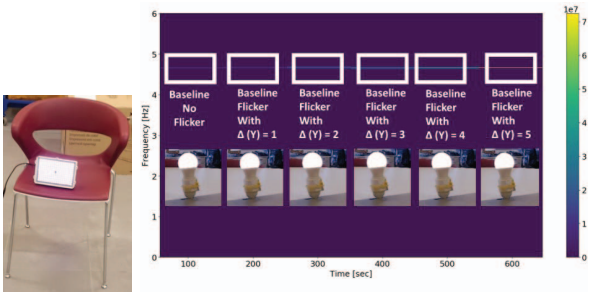


Fig. 15. (a) A picture of an infrared LED projector captured by the DJI Mavic, and (b) A spectrogram (power spectral density) of the intercepted bitrate signal from an experiment in which a smart bulb flickered between a baseline color and five similar hues (as can be seen in Table V).

*1) Experimental Setup:* We decided to test another method that takes advantage of a different limitation of the human eye: its inability to distinguish between two almost identical hues of the same color. In this experiment we aimed to determine whether a physical stimulus that both flickers between two similar hues (with different RGB values) and is undetectable to the human eye can be produced and leave a noticeable (distinguishing) effect on the FPV channel.

TABLE V
YUV AND RGB VALUES USED IN OUR EXPERIMENTS

| Luma (Δ) | YUV | RGB |
|---|---|---|
| Baseline | **231**,26,143 | 253,255,51 |
| 1 | **230**,26,143 | 252,254,50 |
| 2 | **229**,26,143 | 251,253,49 |
| 3 | **228**,26,143 | 250,252,48 |
| 4 | **227**,26,143 | 249,251,47 |
| 5 | **226**,26,143 | 248,250,46 |

We conducted two experiments. In the first experiment, we picked a random RGB color (253,255,51) as the baseline and transformed it to the YUV color space (231,26,143). We created five new hues similar to the baseline color by reducing the luma component (see Table V). We placed the DJI Mavic Pro in front of, and .5 meters away from, a smart LED bulb (Magic Blue) that provides the BLE protocol for controlling. We programmed the Magic Blue to flicker between two similar hues as follows: For the first minute, the Magic Blue was set at the baseline color (231,26,143). For the second minute, the Magic Blue was set to flicker at 2.3 Hz between the baseline color and the color that we created by reducing the luma component by one (230,26,143). For the third minute, the Magic Blue was set to flicker at the same frequency between the baseline color and the color that we created by reducing the luma component by two (229,26,143). This pattern continued until the flickering included the last color that we created (226,26,143). In the second experiment, we positioned the DJI Mavic Pro at various distances (3m, 6m, 10m, 15m, 20m, 25m, 30m) from the Magic Blue bulb that was programmed to flicker between two similar hues: (231,26,143) and (226,26,143). In both experiments, we intercepted the traffic sent from the drone and extracted the intercepted bitrate signal (as described in Section V).

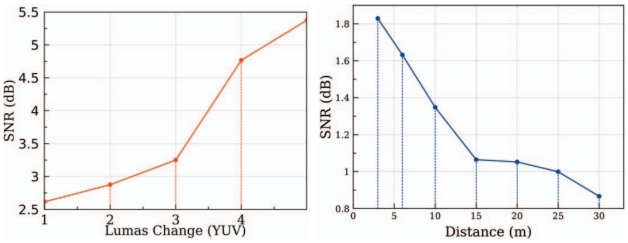


Fig. 16. (a) SNR as a function of the change in the luma component and (b) SNR as a function of the change in the distance

*2) Results:* The hues, as they were captured by the drone's video camera in the first experiment, are presented in Figure 15b. The flickering cannot be detected by the human eye, because human vision is not sensitive enough to detect such subtle changes. We compared the magnitude of the intercepted bitrate signal around 4.6 Hz during the entire experiment. As can be seen in the spectrogram presented in Figure 15b which was extracted from the intercepted traffic, the power of the magnitude around 4.6 Hz increases as much as the delta between the baseline and the second flickering color increases. The SNR as a function of the delta is presented is Figure 15a. Figure 15b shows the results of the second experiment. As can be seen, the SNR is greater than one up to a distance of 15 meters, so this method is only effective for a range shorter than the range of visible flickering (up to 50 meters). Based on this experiment we concluded that the physical stimulus can be disguised in a way that watermarks the intercepted traffic without the awareness of the drone's operator for much shorter ranges. In Appendix XVI, we discuss a method for hiding the physical stimulus.

## X. INFLUENCE OF AMBIENT FACTORS

In this section we investigate the influence of ambient light and wind on the intercepted watermarked signal.

### A. Influence of Wind

The camera of a drone is installed on a stabilizer that is designed to compensate for unwanted camera movement, so the captured picture won't be affected by movement resulting from wind or maneuvering. We start by comparing the video stream of an object obtained from the air and a stand. We conducted two experiments. In the first experiment we positioned the drone on a 1.5 meter high stand on top of a four story building and filmed the landscape for 10 minutes at a frequency of 24 FPS. We repeated the same experiment with a minor change; this time the drone flew to an altitude of 1.5 meters (the same altitude as the stand), and the same landscape was streamed for the same amount of time. As can be seen from the results presented in Figure 17a, the wind mainly affects the low frequencies (below 6 Hz), which are more noisy compared to a static stream. In order to test whether this observation is wind independent, we conducted another experiment in which we positioned a fan behind a flying drone. We used the fan to produce 14 wind speeds. We measured the

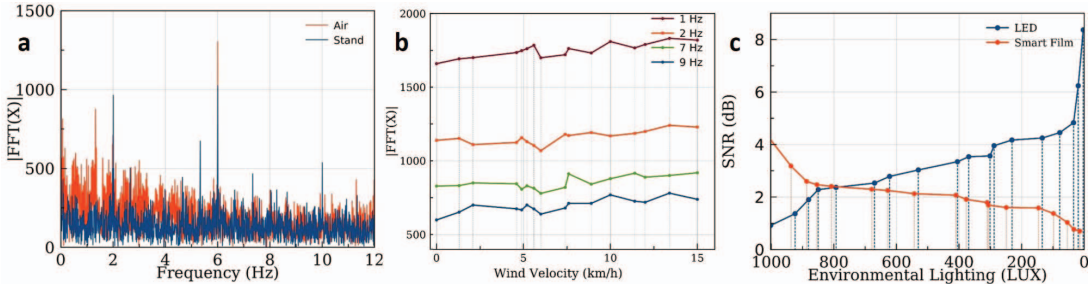


Fig. 17. From left to right: (a) FFT graphs of streaming an object statically from a stand (orange) and air (blue), (b) four magnitudes extracted from external interception to a drone at different wind speeds, (c) SNR as a function of the ambient light (measured in lux).

wind that hit the drone using a wind meter and observed the magnitudes of four frequencies. As can be seen in Figure 17b, the order of the magnitudes of the four frequencies remains the same following the stabilizer operation. In addition, their ranges are also stable and not highly influenced by wind. Therefore, we concluded that frequencies that are higher than 6 Hz are immune to the wind's influence and are less noisy for any given altitude compared to low frequencies. In terms of optimizing the SNR, it is better to use a physical stimulus that can affect frequencies that are higher than 6 Hz.

### B. Influence of Light

Next, we aimed to learn about the influence of ambient light on the SNR. We placed the drone outside the lab, in front of a flickering LED strip (at 3.5 Hz) that was connected to an Arduino Uno and conducted 18 experiments using various levels of ambient light, from morning to night. We repeated the experiment for flickering smart film. In each experiment, we used the intercepted bitrate signal to measure the magnitude around 7 Hz for one minute of flickering and compared it with one minute during which there was no flickering. As can be seen in Figure 17c which presents the SNR as a function of the ambient light (in lux) calculated in the 18 experiments, the SNR that was created from the intercepted bitrate signal from the LED strip experiment improves as a function of the ambient darkness. In contrast, the flickering smart film improves as a function of the ambient light. From this set of experiments, we concluded that by using both an LED strip and smart film, the influence of the watermarker can be felt at all hours of day and night.

## XI. System Evaluation

In this section, we present the final component of our proposed method for detecting privacy invasion attacks: a classification algorithm that uses watermark detection in order to determine whether a given FPV transmission is being used to video stream a victim/target. We evaluate the performance of the proposed privacy invasion attack detection method for two use cases: when the target is a private house (as was the case in [5], [7]) and when the target is a subject driving in his/her car (as was the case in [3]).

Algorithm 4 compares the ratio between the magnitude around the flickering frequency after the periodic physical stimulus was launched (the signal) to the baseline magnitude around the same frequency before the periodic physical stimulus was launched (the baseline/noise). Algorithm 4 is applied after the *Watermarker* method has been called. The algorithm receives a suspicious FPV transmission (*FpvChannel*) and two parameters regarding the periodic physical stimulus: (1) its *startingTime* (EPOC time): the time that the physical stimulus was launched, and (2) *frequency* of operation. A *bitrate* signal is extracted from the intercepted *FpvChannel* (line 3). A bandpass filter is applied (line 4) to the *bitrate* signal around the operated *frequency*. The *filtered* signal is divided into two signals: *before* (line 5) and *after* (line 6) the periodic physical stimulus was launched. The magnitude around the operated frequency before the periodic physical stimulus was launched is given to *noiseMagnitude* (line 8), and accordingly, the magnitude around *frequency* after the periodic physical stimulus was launched is given to *noiseMagnitude* (line 9). Finally, the *FpvChannel* is classified as being used to stream the victim if the SNR is greater than a *threshold* (line 11).

**Algorithm 4** Detecting Whether a POI is Being Streamed

```
1:  procedure isTargetFilmed?(FpvChannel,
2:  frequency,startingTime)
3:      bitrate [] = extractBitrateSignal(FpvChannel)
4:      filtered [] = bandpassFilter(frequency,bitrate)
5:      before [] = subArray(bitrate,0,startingTime)
6:      after [] = subArray(bitrate,startingTime,N)
7:      N = length(bitrate)
8:      noiseMagnitude = FFT(before,30)[frequency]
9:      signalMagnitude = FFT(after,30)[frequency]
10:     SNR = signalMagnitude/noiseMagnitude
11:     return (SNR >= threshold)
```

*1) Experimental Setup:* In order to evaluate the performance of our method, we conducted two sets of experiments. The first set demonstrates how smart film attached to a window can be used as a means of detecting a privacy invasion attack conducted against a private house from a neighboring property (simulating privacy invasion attacks previously published in the media [7], [5]). Figure 18a presents the experimental setup in which the **target** is the **victim's** living room which is being video streamed by a **malicious drone operator** (nosey subject) who uses a **DJI Mavic Pro** (configured to video stream at 30 FPS and 720p) from his/her property (framed

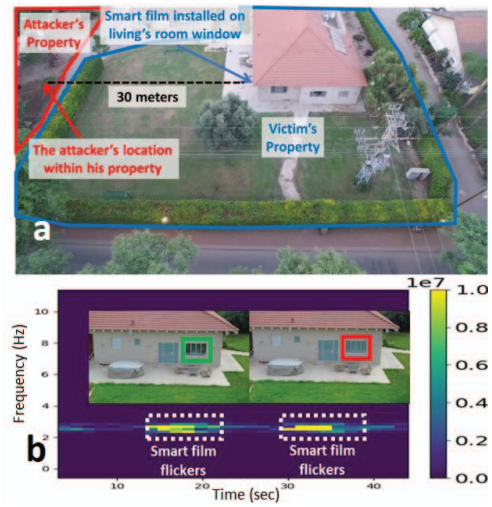


Fig. 18. (a) The attacker's location, Smart film installed on the living room window, and (b) A spectrogram of the intercepted bitrate signal from a drone that was used by a malicious operator to spy on his/her neighbor (smart film was used to flicker twice).

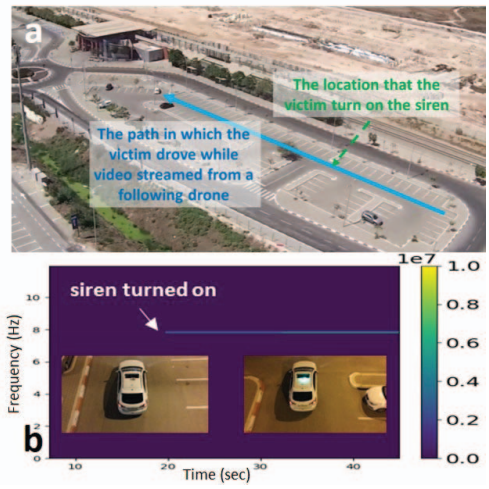


Fig. 19. (a) The point at which the siren was turned on, The route taken by the victim as the victim was being video streamed by a drone, and (b) A spectrogram of the intercepted bitrate signal from a drone that was used by a malicious operator to spy on a person driving a car (a siren was turned on after 20 seconds of driving).

in red), at a distance of around 30 meters from the **victim's** window. We consider smart film (film that changes its state from transparent to mat and vice versa) that has been installed on the **victim's** window and connected to an RF controller as a **Watermarker**. We consider a laptop (Dell Latitude) with an integrated NIC (used as a radio receiver) that is located in the **victim's** home and controls the smart film (i.e., can flicker it from mat to transparent at a given frequency) using a radio transmitter (we used a HackRF One, a software-defined radio) as an **interceptor**. The experiments (which can be viewed on an uploaded video[1]) show how a nosey subject (**malicious drone operator**) uses a drone to film his/her own yard (a legitimate use of a drone), and later uses the same drone, positioned within his/her own airspace, to peek at his/her neighbor (illegitimate use of a drone) by focusing on the neighbor's living room window. In this set of experiments, the drone was located on the property (and within the airspace) of the **malicious drone operator** (framed in red), 30 meters away from the neighbor's living room. The smart film is used as a flickering object that is operated at a frequency of 1.3 Hz. The spectrogram of the intercepted bitrate array from the entire experiment, with a bandpass filter around the 2.6 Hz frequency, is presented in Figure 18b.

The second set of experiments demonstrates how a siren installed on the top of a car can be used as a means of detecting a privacy invasion attack conducted against a subject while he/she is driving in his/her car (simulating the privacy invasion attack that was conducted against a cheating spouse in [3]). Figure 19a presents the experimental setup. We consider a **target** (the **victim's** moving car) that is being video streamed by a **malicious drone operator** who uses a **DJI Spark** (configured to video stream at 30 FPS) to spy on the **victim**. For the **Watermarker** we use a siren that is an LED strip connected to an Arudino Uno microcontroller (used to flicker the siren at a given frequency). We utilize a laptop (Dell Latitude) with an integrated NIC (used as a radio receiver) that is located in the **victim's** car and can trigger the siren as an interceptor. The experiments (see the uploaded video[2]) show how a **victim** that is being followed by a nosey **malicious drone operator** who uses the drone to video stream the **victim** while driving (the **victim's** route is presented in Figure 19a). After 20 seconds of driving, the laptop triggers a green siren that is operated at a frequency of 3.9 Hz. The spectrogram of the intercepted bitrate array (intercepted by the laptop) from the entire experiment with a bandpass filter around the 7.8 Hz frequency is presented in Figure 19.

*2) Results:* Based on the intercepted bitrate arrays that were obtained from the two experiments, we extracted magnitudes around the watermarked frequencies before and after the physical stimulus was started for durations of 1-5 seconds. The results are presented in Figure 20a. As can be seen in the figure, two seconds of the physical stimulus are sufficient for increasing the signal's magnitude (after the physical stimulus began) over the baseline magnitude (before the physical stimulus began). In addition to the experiments that simulated illegitimate uses of a drone, we conducted experiments that simulate legitimate drone use as follows. In the private house experiment, we conducted an additional set of experiments in which the neighbor used his/her drone to film his/her own garden (legitimate use of a drone). In the car experiment, we conducted an additional set of experiments in which the drone was used to film its operator (legitimate use of a drone).

We consider a privacy invasion attack detection system a system that can detect every privacy invasion attack. In order to accomplish this, we tuned the **threshold** variable from line 11 of Algorithm 4 to the minimum SNR calculated from the set of

[1] https://youtu.be/4icQwducz68

[2] https://youtu.be/9PVaDpMsyQE

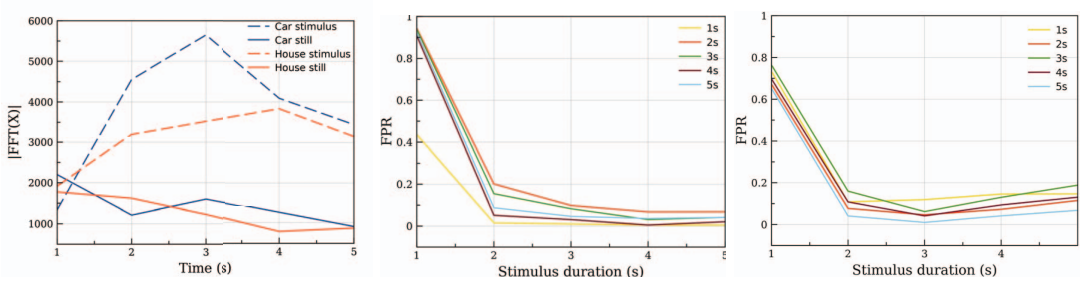


Fig. 20. (a) Extracted magnitudes around the flickering frequency as a function of the duration, (b) False positive rate obtained by applying Algorithm 1 for a legitimate purpose in the private house experiment, and (c) False positive rate obtained by applying Algorithm 1 for a legitimate purpose in the car experiment.

experiments that we conducted previously on the illegitimate use of a drone. By setting the parameter's **threshold** at the minimal SNR value observed by triggering a physical stimulus, we force the detection of each privacy invasion attack that occurs. In order to test the false alarms associated with this, we used the set of experiments that simulated legitimate drone use. We divided the intercepted bitrate array from the experiments that we conducted previously on the legitimate use of a drone into a duration of 10 seconds. We applied Algorithm 1 on the intercepted signals with the baseline and signal magnitudes that were extracted from various durations (15 seconds for each). The FPR results of the private house experiment are presented in Figure 20b, and the FPR results of the car experiment are presented in Figure 20c. As can be seen from the results, the FPR rate goes below the value of 0.1 very quickly: within 2-3 seconds of flickering in the car experiment and within two seconds in the private house experiment. Based on these results, we concluded that a privacy invasion attack detection system that detects every privacy invasion attack can be tuned so that it leads to a minimal amount of false alarms.

## XII. Countermeasures

In this section, we discuss countermeasure methods that can be used by the drone's operator to evade detection resulting from the bursty bitrate that is caused by our carefully crafted physical stimulus. The most effective way to evade detection is by eliminating video compression, i.e., transmitting the raw video stream from the drone to its operator (transmitting just I-frames). However, none of the commercial drones sold today support the functionality of switching the transmitted video stream to a constant bit rate instead of a variable bitrate. This is likely due to the fact that providing a high-quality resolution video stream to the drone's operator is important in order to enable the operator to safely maneuver the drone and avoid collision, and a high resolution video stream requires compression. In addition, since the 1990s, applying compression to a video stream prior to transmission has been mandatory, and it is supported by all next generation video encoders; therefore, the variable bit rate side effect is not about to disappear anytime soon.

Another option for evading detection involves using a drone equipped with two video cameras. The first video camera is used for maneuvering the drone by transmitting the raw video at a constant bitrate (CBR) at a very low resolution without applying any compression. This camera is not focused on the target, in order to prevent it from capturing the flickering object. The second video camera is used to spy on the target and stores the video stream on the SD card at a high resolution using compression. While this method might be effective for static objects (e.g., the window of a building), its main disadvantage is that it is not effective with a moving target (e.g., a passing car), since the video stream presented to the drone's operator is transmitted from a video camera that does not capture the moving target. The abovementioned reasons also explain why evading detection by occasionally disabling the video channel from the drone's operator when using a single camera won't be an effective countermeasure.

## XIII. Limitations & Future Work

Drones manufactures use various protocols other than Wi-Fi for FPV transmission. DJI, for example, uses DSSS and FHSS modulations with its own protocol for FPV transmission [74]. Some additional knowledge regarding the modulation is required in order to apply our method and create a bitrate array from intercepted traffic, i.e., demodulating the signal from the physical layer (radio) to the data link layer (binary). In the case of DSSS, the chip sequence is required in order to demodulate the radio transmission to data, otherwise our method can not be applied. In future work, it would be interesting to implement a technique that was suggested at DEF CON 25 [75] and extracts chip sequence from DSSS transmission. In addition, our work can be extended by adding a threshold for a maximal amount of time for reasonable snooping behavior (defined by the victim) that will ensure that the system only issues alerts for video capturing that exceeds the threshold; this will allow reasonable maneuvering near the target.

## References

[1] B. Insider, "Commercial unmanned aerial vehicle (uav) market analysis," http://www.businessinsider.com/commercial-uav-market-analysis-2017-8, 2017.

[2] ——, "Drone market shows positive outlook with strong industry growth and trends," http://www.businessinsider.com/drone-industry-analysis-market-trends-growth-forecasts-2017-7, 2017.

[3] N. Y. Post, "Husband uses drone to catch cheating wife," https://nypost.com/2016/11/16/husband-uses-drone-to-catch-cheating-wife/, 2016.

[4] kiro7, "Woman terrified by drone outside her window," http://www.kiro7.com/news/woman-terrified-drone-outside-her-window/81721261, 2014.

[5] D. Mail, "Woman grabs gun shoots nosy neighbour's drone," http://www.dailymail.co.uk/news/article-4283486/Woman-grabs-gun-shoots-nosy-neighbour-s-drone.html.

[6] N. Washington, "Virginia woman shoots down drone near actor robert duvalls home," http://www.nbcwashington.com/news/local/Virginia-Woman-Shoots-Down-Drone-Near-Actor-Robert-Duvalls-Home-391423411.html.

[7] N. News, "Kentucky man arrested after shooting down neighbor's drone," http://www.nbcnews.com/news/us-news/not-my-backyard-man-arrested-after-shooting-drone-down-n402271.

[8] Wired, "President trump moves to fill america's skies with drones," https://www.wired.com/story/faa-trump-drones-regulations/, 2017.

[9] Newsweek, "Pizza delivery by drone launched by dominoâĂŹs," http://www.newsweek.com/pizza-delivery-drone-dominos-493371.

[10] B. Insider, "Amazon and ups are betting big on drone delivery," http://www.businessinsider.com/amazon-and-ups-are-betting-big-on-drone-delivery-2018-3.

[11] Fortune, "Cnn just got approved to fly drones over crowds of people," http://fortune.com/2017/10/18/cnn-drones-faa-crowds/, 2017.

[12] A. D. Craze, "Top 12 non military uses for drone," https://www.airdronecraze.com/drones-action-top-12-non-military-uses/.

[13] T. Eshel, "Mobile radar optimized to detect uavs, precision guided weapons," *Defense Update*, 2013.

[14] A. Rozantsev, V. Lepetit, and P. Fua, "Flying objects detection from a single moving camera," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4128–4136.

[15] J. Busset, F. Perrodin, P. Wellig, B. Ott, K. Heutschi, T. Rühl, and T. Nussbaumer, "Detection and tracking of drones using advanced acoustic cameras," in *Unmanned/Unattended Sensors and Sensor Networks XI; and Advanced Free-Space Optical Communication Techniques and Applications*, vol. 9647. International Society for Optics and Photonics, 2015, p. 96470F.

[16] E. E. Case, A. M. Zelnio, and B. D. Rigling, "Low-cost acoustic array for small uav detection and tracking," in *Aerospace and Electronics Conference, 2008. NAECON 2008. IEEE National*. IEEE, 2008, pp. 110–113.

[17] J. R. Vasquez, K. M. Tarplee, E. E. Case, A. M. Zelnio, and B. D. Rigling, "Multisensor 3d tracking for counter small unmanned air vehicles (csuav)," in *Proc. SPIE*, vol. 6971, 2008, p. 697107.

[18] M. Peacock and M. N. Johnstone, "Towards detection and control of civilian unmanned aerial vehicles," 2013.

[19] S. Birnbach, R. Baker, and I. Martinovic, "Wi-fly?: Detecting privacy invasion attacks by consumer drones," *NDSS*, 2017.

[20] BBC, "Big rise in drone jail smuggling incidents," http://www.bbc.com/news/uk-35641453.

[21] L. A. Times, "Two plead guilty in border drug smuggling by drone," http://www.latimes.com/local/california/la-me-drone-drugs-20150813-story.html.

[22] N.-Y. Times, "Secret service arrests man after drone flies near white house," https://www.nytimes.com/2015/05/15/us/white-house-drone-secret-service.html.

[23] ——, "A drone, too small for radar to detect, rattles the white house," https://www.nytimes.com/2015/01/27/us/white-house-drone.html.

[24] Ynet, "Nasrallah describes 1997 ambush," http://www.ynetnews.com/articles/0,7340,L-3932886,00.html.

[25] ——, "What really went wrong in botched 1997 shayetet 13 operation?" http://www.ynetnews.com/articles/0,7340,L-4977429,00.html.

[26] Wired, "Insurgents intercept drone video in king-size security breach (updated, with video)," https://www.wired.com/2009/12/insurgents-intercept-drone-video-in-king-sized-security-breach/.

[27] ——, "Most u.s. drones openly broadcast secret video feeds," https://www.wired.com/2012/10/hack-proof-drone/.

[28] Telegraph, "British and us intelligence 'hacked into israeli drones'," http://www.telegraph.co.uk/news/worldnews/middleeast/israel/12128855/British-and-US-intelligence-hacked-into-Israeli-drones.html.

[29] "The global anti-drone market size is anticipated to reach usd 1.85 billion by 2024," https://www.prnewswire.com/news-releases/the-global-anti-drone-market-size-is-anticipated-to-reach-usd-1-85-billion-by-2024--300673188.html, 2018.

[30] F. Brown, "Game of drones," *DefCon 25*, 2017.

[31] I. E. Richardson, *H. 264 and MPEG-4 video compression: video coding for next-generation multimedia*. John Wiley & Sons, 2004.

[32] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h. 264/avc video coding standard," *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 560–576, 2003.

[33] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with h. 264/avc: tools, performance, and complexity," *IEEE Circuits and Systems magazine*, vol. 4, no. 1, pp. 7–28, 2004.

[34] K. Jack, *Video demystified: a handbook for the digital engineer*. Elsevier, 2011.

[35] rcdronearena, "Wifi fpv vs 5.8ghz fpv vs 2.4ghz fpv: Ultimate guide," http://www.rcdronearena.com/2016/03/15/wifi-fpv-vs-5-8ghz-fpv-vs-2-4ghz-fpv-explained/.

[36] B. Quadcopter, "Wifi fpv vs 5.8ghz fpv vs 2.4ghz fpv," https://www.best-quadcopter.com/versus-zone/2016/04/wifi-fpv-vs-5-8ghz-fpv-vs-2-4ghz-fpv/.

[37] DJI, "Spark remote controller," https://store.dji.com/product/spark-remote-controller.

[38] Parrot, "Parrot skycontroller," https://www.parrot.com/global/accessories/drones/parrot-skycontroller#parrot-skycontroller.

[39] droneuplift, "Top 5 best dji phantom signal range boosters 2017," http://www.droneuplift.com/top-5-dji-phantom-signal-range-extenders/.

[40] auselectronicsdirect, "Wifi fpv drones," https://www.auselectronicsdirect.com.au/drones/fpv-drone/wifi-fpv-drones/.

[41] androidauthority, "8 fun drones you can control with your smartphone," https://www.androidauthority.com/best-smartphone-controlled-drones-744632/.

[42] dronesglobe, "8 drones than can be controlled by a smartphone (fully or partially)," http://www.dronesglobe.com/guide/smartphone-drones/.

[43] T. S. Saponas, J. Lester, C. Hartung, S. Agarwal, T. Kohno *et al.*, "Devices that tell on you: Privacy trends in consumer ubiquitous computing." in *USENIX Security Symposium*, 2007, pp. 55–70.

[44] Y. Liu, A.-R. Sadeghi, D. Ghosal, and B. Mukherjee, "Video streaming forensic-content identification with traffic snooping." in *ISC*. Springer, 2010, pp. 129–135.

[45] R. Schuster, V. Shmatikov, and E. Tromer, "Beauty and the burst: Remote identification of encrypted video streams," in *26th USENIX Security Symposium (USENIX Security 17)*. Vancouver, BC: USENIX Association, 2017, pp. 1357–1374. [Online]. Available: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/schuster

[46] A. Reed and M. Kranch, "Identifying https-protected netflix videos in real-time," in *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy*. ACM, 2017, pp. 361–368.

[47] A. Reed and B. Klimkowski, "Leaky streams: Identifying variable bitrate dash videos streamed over encrypted 802.11 n connections," in *Consumer Communications & Networking Conference (CCNC), 2016 13th IEEE Annual*. IEEE, 2016, pp. 1107–1112.

[48] Y. Liu, C. Ou, Z. Li, C. Corbett, B. Mukherjee, and D. Ghosal, "Wavelet-based traffic analysis for identifying video streams over broadband networks," in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*. IEEE, 2008, pp. 1–6.

[49] C. Wampler, S. Uluagac, and R. Beyah, "Information leakage in encrypted ip video traffic," in *2015 IEEE Global Communications Conference (GLOBECOM)*, Dec 2015, pp. 1–7.

[50] T. Liu, Z. Liu, J. Huang, R. Tan, and Z. Tan, "Detecting wireless spy cameras via stimulating and probing," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2018, pp. 243–255.

[51] C. V. Wright, L. Ballard, F. Monrose, and G. M. Masson, "Language identification of encrypted voip traffic: Alejandra y roberto or alice and bob?" in *USENIX Security Symposium*, vol. 3, 2007, pp. 43–54.

[52] C. V. Wright, L. Ballard, S. E. Coull, F. Monrose, and G. M. Masson, "Spot me if you can: Uncovering spoken phrases in encrypted voip conversations," in *Security and Privacy, 2008. SP 2008. IEEE Symposium on*. IEEE, 2008, pp. 35–49.

[53] A. M. White, A. R. Matthews, K. Z. Snow, and F. Monrose, "Phonotactic reconstruction of encrypted voip conversations: Hookt on fon-iks," in *Security and Privacy (SP), 2011 IEEE Symposium on*. IEEE, 2011, pp. 3–18.

[54] R. Dubin, A. Dvir, O. Pele, and O. Hadar, "I know what you saw last minute;encrypted http adaptive video streaming title classification," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 12, pp. 3039–3049, Dec 2017.

[55] Y. Cheng, X. Ji, T. Lu, and W. Xu, "Dewicam: Detecting hidden wireless cameras via smartphones," in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, ser. ASIACCS

'18. New York, NY, USA: ACM, 2018, pp. 1–13. [Online]. Available: http://doi.acm.org/10.1145/3196494.3196509
[56] 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 *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2017, pp. 211–224.
[57] pcmag, "The best drones of 2018," https://www.pcmag.com/roundup/337251/the-best-drones.
[58] "Airmon-ng," https://www.aircrack-ng.org/doku.php?id=airmon-ng.
[59] "Airodump-ng," https://www.aircrack-ng.org/doku.php?id=airodump-ng.
[60] lighterra, "Video encoding settings for h.264 excellence," http://www.lighterra.com/papers/videoencodingh264/.
[61] I. Video and Control, "Industrial video and control bandwidth calculator," http://www.ivcco.com/bandwidth-calculator/?page=calc.
[62] IBM, "Planning audio and video network bandwidth requirements," https://www.ibm.com/support/knowledgecenter/en/SSKTXQ_9.0.0/admin/plan/plan_av_net_bandwidth_req.html.
[63] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *CoRR*, vol. abs/1709.01015, 2017. [Online]. Available: http://arxiv.org/abs/1709.01015
[64] Ó. Belmonte-Fernández, A. Puertas-Cabedo, J. Torres-Sospedra, R. Montoliu-Colás, and S. Trilles-Oliver, "An indoor positioning system based on wearables for ambient-assisted living," *Sensors*, vol. 17, no. 1, p. 36, 2016.
[65] K. Muthukrishnan, M. Lijding, N. Meratnia, and P. Havinga, "Sensing motion using spectral and spatial analysis of wlan rssi," in *European Conference on Smart Sensing and Context*. Springer, 2007, pp. 62–76.
[66] DJI, "Mavic pro," https://www.dji.com/mavic.
[67] "Nebula capsule," https://www.amazon.com/Projector-Anker-Portable-High-Contrast-Playtime/dp/B076Q3GBJK.
[68] Scantips, "Calculate distance or size of an object in a photo image," https://www.scantips.com/lights/subjectdistance.html.
[69] scantips, "Calculate distance or size of an object in a photo image," https://www.scantips.com/lights/subjectdistance.html.
[70] DJI, "Mavic air," https://www.dji.com/mavic-air/info#specs.
[71] ——, "Spark," https://www.dji.com/spark/info#specs.
[72] "Spherical coordinate system," https://en.wikipedia.org/wiki/Spherical_coordinate_system.
[73] "Gps accuracy," https://www.gps.gov/systems/gps/performance/accuracy/.
[74] A. Luo, "Drones hijacking - multi-dimensional attack vectors and countermeasures," *DefCon 24*.
[75] M. Szczys, "Michael ossmann pulls dsss out of nowhere," https://hackaday.com/2017/07/29/michael-ossmann-pulls-dsss-out-of-nowhere/, 2017.
[76] D. Mitrovic, "Video compression," *University of Edinburgh*.
[77] nickkolenda, "Subliminal messages: Do they really work?" https://www.nickkolenda.com/subliminal-messages/.
[78] L. Wang, "Are you being manipulated by subliminal messages?" http://blog.visme.co/subliminal-messages/.

## XIV. Appendix - Stages of Video Compression Algorithms

### A. Intra-Coding Process (Creating I-Frames)

Intra-coding contains the following stages [76], [34]:

### B. Inter-Coding Process (Creating B and P-Frames)

The process of generating a delta frame is similar to the intra-coding process, with minor changes. Between the quantization stage and the entropy coding stage the following stages are added:

1) Reference block matching - A similar block in another frame is identified.
2) Motion vector extraction - The difference between the two blocks is extracted by calculating the prediction error. This is the data that is used to describe the delta frame (prediction error from another frame).

## XV. Appendix - Interception & Creating Bitrate Signal Script

Listing 1 presents the bash script that implements the process of interception (stages 1-3 in Section V).

```
#!/bin/bash
#start monitor mode
airmon-ng check kill
airmon-ng start wlan0
#Capture packets of specific network
airodump-ng --bssid $1 --write capture.pcap wlan0mon
read -p "Press any key to exit monitor mode... " -n1 -s
#exit monitor mode
airmon-ng stop wlan0mon
service network-manager start
rfkill unblock all
```

Listing 1. Applying interception script.

The bash script presented in Listing 1 received the *BSSID* as the argument and creates a PCAP file that contains packets captured from the BSSID's network.

Listing 2 presents the bash script that implements the bitrate signal interception process (stages 4-5 in Section V).

```
#!/bin/bash
prefix=12
suffix=1
interval=0.041666666667
tshark -q -z 'io,stat,'"$interval" -r "$1" > "$1".txt -2
lines=$(wc -l < "$1".txt)
line_2_remove="$((lines - prefix))"
echo $line_2_remove
echo $lines
tail --lines=$line_2_remove "$1".txt > tmp.txt
lines=$(wc -l < tmp.txt)
line_2_remove="$((lines - sefix))"
head --lines=$line_2_remove tmp.txt > tmp2.txt
cut -f 3 -d '|' tmp2.txt > tmp3.txt
cut -f 4 -d '|' tmp2.txt > tmp4.txt
cat tmp3.txt > packets.txt  | tr -d "\t\n\r"
cat tmp4.txt > bytes.txt  | tr -d "\t\n\r"
echo 'packets' | cat - packets.txt > temp && mv temp packets.txt
echo 'bytes' | cat - bytes.txt > temp && mv temp bytes.txt
paste -d "," packets.txt bytes.txt  >> "$1".csv
rm packets.txt bytes.txt "$1".txt tmp4.txt tmp3.txt tmp2.txt tmp.txt
paste -d "," *csv  >> all.txt
rm *.csv
```

TABLE VI
CLASSIFICATION RESULTS BASED ON VARIOUS INTERCEPTION PERIODS

| | | Actual | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Interception Period | 1 second | | 2 seconds | | 3 seconds | | 4 seconds | | 5 seconds | |
| | Predicted/Actual Moving IoT Device | Drone | Others | Drone | Others | Drone | Others | Drone | Others | Drone | Others |
| Predict | Drone | 173 | 31 | 185 | 23 | 193 | 12 | 200 | 5 | 200 | 3 |
| | Other Moving IoT Devices | 27 | 169 | 15 | 177 | 7 | 188 | 0 | 195 | 0 | 197 |

```
24tr -d " \t" < all.txt > aggregation.
    txt
25rm all.txt
```

Listing 2. Interception and creating bitrate signal script.

The script presented in Listing 2 receives the path to the PCAP as the argument and creates a bitrate signal by aggregating all of the packets according to an *interval* parameter.

## XVI. APPENDIX - HIDING THE PHYSICAL STIMULUS EXTENSION

One of the methods that we considered takes advantage of the limited capturing speed of the human eye (48-60 Hz) by creating a stimulus that is too brief to be detected by the human eye. This method is popular in advertising where it is used in order to affect the viewer's subconscious using visual cues that appear so briefly (for just a few milliseconds) that people don't perceive them [77], [78]. However, most drones use an FPV channel that supports 25-30 FPS. Hence, even if a drone's optical sensor captures a brief stimulus, the change will be presented to the drone's operator in his/her controller for 33-40 milliseconds, revealing the physical stimulus and thereby disqualifying this method.

## XVII. APPENDIX - CONFUSION MATRICES FROM CLASSIFYING FPV CHANNEL EXPERIMENT

Table VI presents confusion matrices resulting from the application of Algorithm 2 with various interception windows on the following moving IoT devices: drone, smartwatch, smartphone, and robotic vacuum cleaner.

## XVIII. APPENDIX - LOCATING DRONE EXPERIMENT

TABLE VII
LOCATING DRONE EXPERIMENT - RESULTS

| | Real Drone Location | | | | | | | | | Predicted Drone Location | | | | | | | | | Error | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Latitude** | **Longitude** | **Altitude (m)** | **Δ x (m)** | **Δ y (m)** | **Δ z (m)** | **r (m)** | **theta (°)** | **phi (°)** | **Latitude** | **Longitude** | **Altitude (m)** | **Δ x (m)** | **Δ y (m)** | **Δ z (m)** | **r (m)** | **theta (°)** | **phi (°)** | **x (m)** | **y (m)** | **z (m)** |
| 1 | 31.26310 | 34.81051 | 0 | -10 | 0 | 0 | 10.0 | 0.0 | 0.0 | 31.26314 | 34.81048 | -3.4 | -13.6 | 3.1 | -3.4 | 14.3 | 13.9 | -12.8 | 12.7 | 9.5 | 11.9 |
| 2 | 31.26290 | 34.81051 | -4 | -9 | -5 | -4 | 11.0 | 21.2 | 29.1 | 31.26314 | 34.81059 | -7.8 | -12.4 | -8.1 | -7.8 | 16.7 | 27.9 | 33.1 | 11.5 | 9.4 | 14.6 |
| 3 | 31.26292 | 34.81041 | 1 | -6 | -8 | 1 | 10.0 | -5.7 | 53.1 | 31.26307 | 34.81056 | 4.7 | -2.8 | -5.0 | 4.7 | 7.3 | -39.4 | 60.9 | 10.5 | 9.3 | 13.4 |
| 4 | 31.26298 | 34.81035 | -2 | -4 | -10 | -2 | 11.0 | 10.5 | 68.2 | 31.26314 | 34.81064 | -5.6 | -7.6 | -13.1 | -5.6 | 16.1 | 20.2 | 60.0 | 12.7 | 9.6 | 12.7 |
| 5 | 31.26302 | 34.81031 | -7 | 1 | -10 | -7 | 12.2 | 34.9 | -84.3 | 31.26307 | 34.81058 | -10.6 | 4.6 | -7.0 | -10.6 | 13.5 | 51.7 | -56.3 | 13.3 | 9.2 | 13.1 |
| 6 | 31.26312 | 34.81031 | 2 | 4 | -9 | 2 | 10.0 | -11.5 | -66.0 | 31.26307 | 34.81057 | 5.5 | 7.5 | -5.7 | 5.5 | 10.9 | -30.5 | -37.0 | 12.3 | 11.2 | 12.5 |
| 7 | 31.26318 | 34.81033 | 0 | 7 | -7 | 0 | 9.9 | 0.0 | -45.0 | 31.26314 | 34.81055 | 3.6 | 3.5 | -3.8 | 3.6 | 6.3 | -35.2 | -47.4 | 12.3 | 10.2 | 13.3 |
| 8 | 31.26324 | 34.81037 | -4 | 8 | -6 | -4 | 10.8 | 21.8 | -36.9 | 31.26314 | 34.81061 | -7.7 | 4.3 | -9.3 | -7.7 | 12.8 | 37.0 | -65.1 | 13.6 | 10.9 | 13.8 |
| 9 | 31.26326 | 34.81039 | -3 | 0 | -10 | -3 | 10.4 | 16.7 | 0.0 | 31.26314 | 34.81065 | 0.9 | -3.4 | -13.3 | 0.9 | 13.7 | -3.7 | 75.5 | 11.8 | 10.7 | 15.1 |
| 10 | 31.26310 | 34.81031 | 2 | 9 | -4 | 2 | 10.0 | -11.5 | -24.0 | 31.26313 | 34.81058 | -1.8 | 5.7 | -6.9 | -1.8 | 9.1 | 11.2 | -50.3 | 10.7 | 8.4 | 14.3 |
| 11 | 31.26328 | 34.81043 | -6 | -5 | -9 | -6 | 11.9 | 30.2 | 60.9 | 31.26313 | 34.81064 | -9.4 | -8.3 | -12.3 | -9.4 | 17.6 | 32.5 | 56.0 | 10.7 | 10.7 | 11.9 |
| 12 | 31.26300 | 34.81033 | -7 | -20 | 0 | -7 | 21.2 | 19.3 | 0.0 | 31.26313 | 34.81048 | -3.6 | -23.2 | 3.0 | -3.6 | 23.7 | 8.8 | -7.3 | 10.5 | 9.0 | 11.5 |
| 13 | 31.26270 | 34.81051 | 1 | -19 | 3 | 1 | 19.3 | -3.0 | -9.0 | 31.26307 | 34.81045 | -2.8 | -15.5 | 6.0 | -2.8 | 16.9 | 9.6 | -21.0 | 12.0 | 8.9 | 14.7 |
| 14 | 31.26272 | 34.81057 | -4 | -17 | -12 | -4 | 21.2 | 10.9 | 35.2 | 31.26307 | 34.81060 | -7.6 | -13.6 | -8.7 | -7.6 | 17.8 | 25.4 | 32.7 | 11.8 | 10.8 | 13.3 |
| 15 | 31.26276 | 34.81027 | -7 | -15 | -14 | -7 | 21.7 | 18.8 | 43.0 | 31.26307 | 34.81068 | -10.2 | -11.4 | -17.2 | -10.2 | 23.0 | 26.3 | 56.5 | 13.3 | 10.0 | 10.1 |
| 16 | 31.26280 | 34.81023 | -3 | -12 | -16 | -3 | 20.2 | 8.5 | 53.1 | 31.26313 | 34.81064 | -6.4 | -15.0 | -12.7 | -6.4 | 20.7 | 18.0 | 40.4 | 8.8 | 10.7 | 11.4 |
| 17 | 31.26286 | 34.81019 | -7 | -6 | -20 | -7 | 22.0 | 18.5 | 73.3 | 31.26314 | 34.81074 | -3.5 | -9.4 | -23.2 | -3.5 | 25.3 | 7.9 | 67.9 | 11.9 | 10.5 | 12.2 |
| 18 | 31.26298 | 34.81011 | -1 | 0 | -20 | -1 | 20.0 | 2.9 | 0.0 | 31.26313 | 34.81068 | -4.6 | -3.3 | -17.0 | -4.6 | 18.0 | 15.0 | 79.0 | 10.9 | 8.8 | 13.2 |
| 19 | 31.26310 | 34.81011 | -3 | 5 | -20 | -3 | 20.8 | 8.3 | -76.0 | 31.26314 | 34.81068 | 0.2 | 1.1 | -16.6 | 0.2 | 16.6 | -0.7 | -86.2 | 15.1 | 11.6 | 10.3 |
| 20 | 31.26320 | 34.81011 | -4 | 9 | -18 | -4 | 20.5 | 11.2 | -63.4 | 31.26307 | 34.81066 | -0.4 | 12.6 | -14.8 | -0.4 | 19.5 | 1.1 | -49.6 | 13.0 | 10.0 | 13.0 |
| 21 | 31.26328 | 34.81015 | -5 | 17 | -10 | -5 | 20.3 | 14.2 | -30.5 | 31.26314 | 34.81058 | -8.7 | 13.6 | -6.7 | -8.7 | 17.4 | 30.0 | -26.3 | 11.9 | 10.9 | 13.8 |
| 22 | 31.26344 | 34.81031 | -1 | -29 | -9 | -1 | 30.4 | 1.9 | 17.2 | 31.26314 | 34.81064 | -5.1 | -32.5 | -12.3 | -5.1 | 35.1 | 8.3 | 20.7 | 11.9 | 10.6 | 16.4 |
| 23 | 31.26252 | 34.81033 | -8 | -28 | -12 | -8 | 31.5 | 14.7 | 23.2 | 31.26313 | 34.81066 | -4.4 | -31.2 | -14.8 | -4.4 | 34.9 | 7.2 | 25.4 | 10.5 | 8.1 | 13.3 |
| 24 | 31.26254 | 34.81027 | 0 | -26 | -16 | 0 | 30.5 | 0.0 | 31.6 | 31.26307 | 34.81064 | -3.7 | -22.6 | -13.1 | -3.7 | 26.3 | 8.1 | 30.1 | 11.9 | 8.5 | 13.9 |
| 25 | 31.26258 | 34.81019 | -2 | -23 | -20 | -2 | 30.5 | 3.8 | 41.0 | 31.26306 | 34.81074 | 2.1 | -19.3 | -23.0 | 2.1 | 30.1 | -3.9 | 50.0 | 13.9 | 8.9 | 16.5 |
| 26 | 31.26264 | 34.81011 | -2 | -18 | -25 | -2 | 30.9 | 3.7 | 54.2 | 31.26307 | 34.81073 | -5.8 | -14.6 | -21.9 | -5.8 | 26.9 | 12.5 | 56.4 | 11.8 | 9.6 | 14.7 |
| 27 | 31.26274 | 34.81001 | -2 | -11 | -29 | -2 | 31.1 | 3.7 | 69.2 | 31.26314 | 34.81077 | 1.7 | -14.4 | -26.0 | 1.7 | 29.8 | -3.2 | 61.0 | 11.8 | 8.8 | 13.6 |
| 28 | 31.26288 | 34.80993 | 2 | -8 | -30 | 2 | 31.1 | -3.7 | 75.1 | 31.26314 | 34.81078 | 5.8 | -11.5 | -26.9 | 5.8 | 29.8 | -11.2 | 66.8 | 12.4 | 9.9 | 14.4 |
| 29 | 31.26294 | 34.80991 | -3 | 10 | -29 | -3 | 30.8 | 5.6 | -71.0 | 31.26307 | 34.81083 | -6.6 | 13.6 | -32.2 | -6.6 | 35.5 | 10.7 | -67.1 | 12.9 | 9.9 | 12.8 |
| 30 | 31.26330 | 34.80993 | 1 | 16 | -25 | 1 | 29.7 | -1.9 | -57.4 | 31.26306 | 34.81073 | 4.5 | 19.8 | -21.7 | 4.5 | 29.8 | -8.7 | -47.6 | 14.8 | 10.8 | 12.3 |