Onboard Drone Detection with Event Cameras

A Study on their Applicability

Alejandro Barberia Chueca

Delft University of Technology

TU Delft

Onboard Drone Detection with Event Cameras

A Study on their Applicability

by

Alejandro Barberia Chueca

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday July 14, 2022 at 13:00.

Student number:	4537998	
Project duration:	March 25, 2021 – July 14, 2022	
Thesis committee:	Dr. G. C. H. E. de Croon,	TU Delft, supervisor
	Dr. ir. B. W. van Oudheusden,	TU Delft, external examiner
	Dr. ir. E. J. J. Smeur,	TU Delft

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Acknowledgements

This thesis marks the end of my chapter in Delft as a student. A six year long journey filled with challenges, adventures, and unforgettable moments. I would like use this chance to express some words of gratitude for all the people that have accompanied me along the way.

First, I would like to thank my supervisors, Julien Dupeyroux and Guido de Croon, for their guidance during this thesis. Thank you for pushing me to think critically and independently and always expecting more from me. On a similar note, I want to thank all the professors I had the pleasure of working with during the BSc and MSc. Thank you for contributing to my personal and professional development, each in your own way.

Secondly, I can only be thankful for all the amazing people I have met here in Delft and that I have the pleasure to call friends. You have made this journey truly wonderful. Thank you for all the adventures, conversations, laughs and trips shared. I can only look back and smile thinking of all the special moments I shared with each one of you.

Lastly, and most importantly, I would like to thank my family for the immense care and support I have received, not only these last six years, but all my life. This would have not been possible without you. Thank you for always wanting the best for me, pushing me always to put all my heart and hard work in everything I do, and setting an example with your daily lives of the values that I want to follow.

Alejandro Barberia Chueca
Delft, July 2022

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

Abbreviation	Definition
AFS	Aggregations of Foraging Swarm
APM	Attention Priority Map
BLE	Broadcast of Local Eligibility
BFS	Breadth First Search
CNN	Convolutional Neural Network
DL	Deep Learning
DBSCAN	Density Based Clustering of Applications with Noise
DFS	Depth First Search
DVS	Dynamic Vision Sensor
FOV	Field of View
FPS	Frames per Second
FSM	Finite State Machine
GSO	Glowworm Swarm Optimisation
ICP	Iterative Closest Point
IR	Infrared
LDSI	Less Data Same Information
LOS	Line of Sight
LQR	Linear Quadratic Regulator
MA-MCTS	Multi-Agent Monte-Carlo Tree Search
MAV	Micro Aerial Vehicle
MOTA	Multiple Object Tracking Accuracy
MPC	Model Predictive Controller
NN	Neural Networks
NSBC	Null-Space-based Behavioural Control
PCA	Principal Component Analysis
PSO	Particle Swarm Optimisation
RSSI	Received Signal Strength Intensity
RPM	Revolutions per Minute
SLAM	Simultaneous Localisation and Mapping
SNN	Spiking Neural Network
SNR	Signal to Noise Ratio
UAV	Unmanned Aerial Vehicle
UWB	Ultra Wide Band

Symbols

Symbol	Definition	Unit
T_{obs}	Camera observation time	[s]
T_{acc}	Camera accumulation time	[s]
N	Analysis frequency	[Hz]
k	Accumulation frames for events	[frames]
C	Event count per pixel	[events]
B	Propeller blades	[-]
f_{prop}	Propeller spinning frequency	[RPMs]

Symbol	Definition	Unit
T_{sym}	Propeller Symmetry period	[s]
P	Propeller symmetries per frame	[-]
R	Propeller radius	[m]
d	Propeller tip projection displacement	[m]
o_c	Count metric	[-]
o_p	Periodicity metric	[-]
$\theta_{max\ OL}$	Maximum overlap angle	[rad]
Δ	Shadow angle	[rad]
λ	Shadow clearance angle	[rad]

1

Introduction

Recent technical advancements in the field of robotics have allowed many new applications that before were unimaginable. Robots are now able to successfully replace human operators for risky and hazardous tasks, such as nuclear and biohazard discovery, mine exploration, planetary exploration or military operations. Robotic automation can also become extremely useful for repetitive tasks such as food sorting, assembly-line operation or warehouse inventory. These different applications can be included in the three D's where robots outperform humans: dull, dangerous and dirty tasks [49].

Despite all this development, current robots are still far from perfect and have plenty of limitations. Limited area coverage for mobile robots, limited weight capabilities for weight carrying robots or limited sensor accuracy for mapping robots are just some examples. One way to deal with such limitations is to design better robots to perform grander tasks. However, this comes at the expense of large investments and long development cycles. An alternative cheaper and faster way to perform these tasks with currently available robots is with swarms. Robotic swarms allow many robots to collaborate to achieve tasks that exceed the capabilities of the individual agents. For that reason, robotic swarms have steadily gained an increasing interest [153].

Looking at the application on drones, swarms become even more relevant. Having to lift all their components off the ground, drones are characterised by more restrictive limitations on sensing, power and operation time than ground or underwater robots. Therefore, drone swarms can leverage the use of many agents to achieve tasks individual drones could not accomplish. Drone swarms become especially useful to, for example, increase coverage, both in terms of time and space, improve sensor readings by combining data from different agents, or increase the carrying capabilities of the individual counterpart.

The gap between independent drones and a swarm is composed of two elements: an intra-swarm relative sensing and avoidance subsystem and a swarm behaviour strategy. The former allows the drones to detect each other and provides them with the ability to coordinate. The latter establishes the coordination rules followed to collectively achieve the desired goal [36].

In the context of relative sensing and avoidance, a wide range of systems have been developed to enable coordination between the different drones. The two main requirements for this system include high accuracy of localisation, specially for close distances to avoid collisions, and robustness to the fast dynamics of a flying drone. Cameras [16, 129], infrared sensors [88, 111], sound [30, 12] or active communication [117, 115] are some examples of different methods used for this purpose. However, as it is shown in this paper, all the mentioned systems have certain drawbacks related to the accuracy or to the operating conditions.

An event camera, also called dynamic vision sensor or neuromorphic camera, is a biologically inspired sensor that asynchronously measures changes in pixel brightness. Instead of capturing full images like a conventional camera, where the colour of every pixel is recorded and stored under a frame, each pixel in an event camera works independently. This enables a high temporal resolution (in the order of μs), low power consumption, very high dynamic range (140 dB vs 60 dB for standard cameras) and high pixel bandwidth (in the order of kHz) [52]. Only becoming commercially available in 2008 [82], event cameras have steadily gained attention and are now starting to show their potential for a wide range of applications, mainly those where the limitations of the conventional cameras was hindering

the technical progress. Examples include object tracking [41, 56], object recognition [107, 113] or shape tracking [144, 4].

1.1. Research Aim

Meeting the need for a relative sensing system that is accurate and robust to fast dynamics, event cameras appear as an obvious candidate for a new method to enable swarming behaviour in drones. Therefore, the research objective of this thesis can be formulated as follows:

to evaluate the suitability of event cameras for relative localisation on drone swarms by means of experimentally assessing their performance for propeller detection

1.2. Thesis Structure

This report is structured as follows. Part I presents a scientific article that describes the main contributions of this research. In such, an overview of related work on relative sensing for drone swarms and object detection with event cameras is presented in section 2.2. Then, in section 2.3, the neuromorphic signal process employed is described emphasizing the correlation between propeller motion and analysis frequency. The experiments conducted are presented in section 2.4, together with the main results. Finally, the conclusions are presented in section 2.5.

Part II presents the preliminary research performed for this thesis. This was conducted as part of a broader project to study the deployment of drone swarms for unknown area exploration. While this thesis focused on the problem of relative sensing, the different challenges regarding the complete implementation are addressed. Chapter 3 offers a biologically inspired introduction to swarming. Then chapter 4 covers previous work on relative sensing. Chapter 5 addresses the coordination strategies that can be followed to achieve the exploration goal.

Finally, Part III offers additional results that support the scientific article. Chapter 6 presents a sensitivity analysis conducted on the analysis frequency for the neuromorphic signal processing. Lastly, chapter 7 shows the complete results for the experiments conducted for this thesis.

Part I

Scientific Article

Study of Onboard Inter-Drone Detection with Event Cameras

A. Barberia Chueca[†*], J. Dupeyroux[‡*], and G.C.H.E. de Croon [‡*]

[†]MSc student, [‡]Supervisor
[*]Control & Simulation, Control & Operations Department
Delft University of Technology, Delft, The Netherlands

***Abstract*— In an effort to develop a new relative sensing method for drone swarms, the suitability of event cameras is assessed for propeller detection. Benchmark tests were conducted for different propellers under different lighting and background conditions, varying the observation distance and spinning frequency. The different tests were evaluated on event count, frequency, and clustering, as these are the most characteristic properties of the propeller-generated signal. A propeller detection metric was derived as a fuzzy classifier to assess detectability. It was observed that the sensor employed is limiting the detection range due to low resolution, with a maximum detection range of 75 cm. While at low spinning frequencies it is possible to detect the propeller at such distance, for higher frequences (6000 to 8000 RPMs) the range decreases to 45 cm for the tests with highest blade to background contrast and two-blade propellers. It was observed that lower contrasts reduce the successful detections only to low frequencies, and three-blade propellers become completely indetectable due to the static overlap between the blades. Therefore, it is concluded that, at this stage of the technology, the use case of event cameras for relative sensing is constrained to close distances with high contrast.**

I. Introduction

Having to lift all their components off the ground, drones are characterised by more restrictive limitations on power and operation than ground or underwater robots. Drone swarms leverage the number of agents to perform tasks that exceed the capabilities of any individual agent.

The gap between independent drones and a swarm is composed of two elements: an intra-swarm relative sensing and avoidance subsystem and a swarm behaviour strategy. The former allows the drones to detect each other and provides them with the ability to coordinate. The latter establishes the coordination rules followed to collectively achieve the desired goal [1].

In the context of relative sensing and avoidance, a wide range of systems have been developed to enable coordination between the different drones. The two main requirements for this system include high accuracy of localisation, specially for close distances to avoid collisions, and robustness to the fast dynamics of a flying drone. Cameras [2], [3], infrared sensors [4], [5], sound [6], [7] or active communication [8], [9] are some examples of different methods used for this purpose. However, as shown in this paper, all the mentioned systems have drawbacks related to the accuracy or to the operating conditions.

An event camera, also called dynamic vision sensor or neuromorphic camera, is a biologically inspired sensor that asynchronously measures changes in pixel brightness. Instead of capturing full images like a conventional camera, where the colour of every pixel is recorded and stored under a frame, each pixel in an event camera works independently. This enables a high temporal resolution (in the order of μs), low power consumption, very high dynamic range (140 dB vs 60 dB for standard cameras) and high pixel bandwidth (in the order of kHz) [10]. Only becoming commercially available in 2008 [11], event cameras have steadily gained attention and are now starting to show their potential for a wide range of applications, mainly those where the limitations of conventional cameras was hindering technical progress. Examples include object tracking [12], [13], object recognition [14], [15] or shape tracking [16], [17].

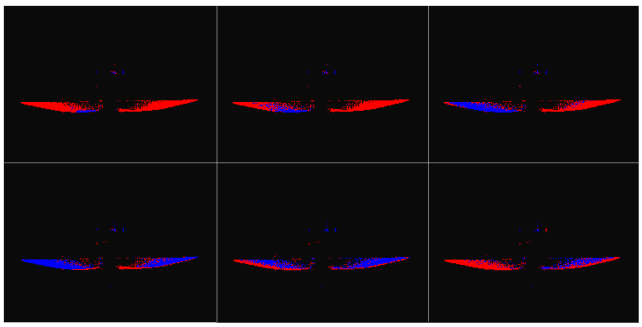


Fig. 1: Time evolution of a propeller spinning at 6000 RPMs captured with an event camera at a 25 cm distance. In blue, the negative events show the advancing black profile against a white background. Complete sequence lasted 0.0037 s.

Meeting the need for a relative sensing system that is accurate and robust to fast dynamics, event cameras appear as an obvious candidate for a new method to enable swarming behaviour in drones. This research focuses on the study of the viability of such system, combining both a theoretical and experimental approach. More particularly, the detection of the drone's propellers is assessed, as this exploits the sensor properties. An example of a rotating propeller captured with an event camera can be found in Figure 1, spinning at 6000 RPMs and recorded from a 25 cm distance. As it can be observed, the suitability of this camera for this challenge is evidenced by its ability to capture the fast dynamics of the propeller. The limitations of the system are analysed in the next sections.

The contributions of this paper are the following:

1) Development of a propeller detection benchmark.
2) Derivation of custom metrics for classification of a neuromorphic signal.
3) Sensitivity analysis of external factors on propeller detection.

The rest of this paper is organised as follows. Section II presents different relative sensing systems used, followed by an introduction to event-based object detection. In Section III, the signal processing method is presented with requirements derived from the propeller motion analysis. Section IV describes the experiments performed, analysing the results found. Lastly, Section V covers concluding remarks and recommendations for future work.

II. Related work

When developing a new method for a specific task, knowledge of already existing methods is critical for a correct assessment of its performance. Understanding the working principle, strengths, and limitations of current relative sensing systems, as well as performance metrics, enables a proper analysis of the value and effectiveness of event cameras for relative sensing on drone swarms. Therefore, this section starts with a review on the main methods used until now for relative sensing and avoidance. These detection systems are typically used together with a tracking algorithm for robust performance. A short overview of the state of the art tracking methods is also included to get a complete understanding of the system. Lastly, the focus is placed on event cameras used for generic object detection and tracking. Event cameras have already been employed for a range of applications. A survey on the different techniques followed is presented as source of inspiration for the development of the system in question.

There are five main methods that have been used in the past for relative sensing on drone swarms [1]:

a) Vision: Different methods have been applied to detect drones on the images captured. Neural Networks and Deep Learning models have been used with the purpose of semantic segmentation, with limited success due to the large processing required for robust detections [2], [3]. Another method is the use of stereo cameras for depth estimation from the image disparity [18]. Finally, visual markers, passive and active, have also been included to the drones to ease the detection [19], [20]. Vision methods can achieve a high accuracy, with the best performance achieved by stereo vision with a 5% distance error for distances up to 50 m [18]. The main limitations of vision-based relative sensing are derived from the physical sensor used. Namely the limited Field of View, motion blur and low dynamic range.

b) Infrared: The use of infrared emitters and receivers allows each drone to send an almost constant signal from its local position, and receive the signals of neighbouring drones in the working range. Most commonly, the Received Signal Strength Intensity is used to derive the location of surrounding drones from the strength of the received signals. The accuracy of this system has been proven to be sufficient for relative sensing, with a distance error of 8% on a range up to 3 m [5]. While it allows fast detections at very low computational power, many sensors are needed for full coverage, it is susceptible to interference from other infrared sources and needs to be tuned for a specific working range.

c) Sound: The disturbances created by the rotating propellers in the form of acoustic noise can be captured by the surrounding drones. Using a microphone array, it is possible to hear neighbouring UAVs and estimate their range and bearing. "Chirp" sounds have been used to distinguish the self-generated noise from the surrounding agents [7]. Moreover, Convolutional Neural Networks have also been employed with limited gain in accuracy and increased weight and computational requirements [6]. These works, however, were focused on presence detection or bearing estimation and did not tackle range estimation, so there is no accuracy performance metric available.

d) Communication-based ranging: Similarly to the infrared sensor, communication-based ranging estimates range and bearing from the received signal strength. Leveraging an existing communication infrastructure, there are two main options regarding the signal type: ultrasound and radio signals. Ultrasound signals offer better performance on short ranges (0.8 cm distance error over a 6.7 m range [8]) at less energy consumption, but are slow to dissipate causing interferences [4]. Radio signals include WiFi, Bluetooth, and Ultra Wide Band (UWB), differing in power consumption, accuracy, and bandwidth. Most importantly, UWB performs best on the first two categories, achieving a cm resolution over ranges longer than 50 m, but is outperformed by WiFi in bandwidth, connecting up to 50 agents [1], [9], [21].

e) Landmark matching: The last relative sensing method also uses communication, but to exchange information regarding the drone's location in the environment. Each drone generates a landmark map (with lidar, radar, cameras...), and broadcasts its position with respect to them. The challenge of relative sensing is then transformed into a localisation problem. In the simplest form, drones share mutually observable features. More complex solutions include the generation of a full map of the environment that is exchanged and updated for global relative localisation [22]. Once more, there is no accuracy metric that can be associated with this method as it is completely dependent on the landmark detection technique employed.

All these relative sensing systems work based on the principle of instantaneous detection. When used independently, these methods only provide information regarding the last detection of the neighbouring drones. To increase robustness against missed detections and to provide more continuous relative sensing, tracking algorithms are commonly also employed. A distinction can be made between the different algorithms used based on the nature of the processing performed: white box vs black box models.

On one hand, in white box models the relationship between inputs, i.e. detections, and outputs, i.e. trajectories, is interpretable and the influence of each variable is clear. Moreover, these generally include hand-crafted features, implementing a desired behaviour. White box models have the

advantage of being generally faster as they rely on simpler and fewer computations. However, their tracking power is lower when detections have low certainty. Some examples of white box tracking algorithms include Kalman Filtering [23], Mean Shift [24], and Particle Filters [25].

On the other hand, black box models do not offer interpretable relationships between inputs and outputs, are highly non linear, and cannot explain the influence of the different variables. Often used end-to-end, black box models can learn update features after initial detection for a more robust tracking. That way, no explicit communication needs to be performed between the detector and the tracker. The most common algorithm employed is Long Short Term Memory [26], [27], followed by other types of Recurrent Neural Networks [28] or Graph Neural Networks [29].

Lastly, it is worth studying the work previously done on object detection using event cameras to understand the detection capabilities of this sensor.

Object detection with event cameras can also be separated into white box and black box models. Early work on the topic focused on white box approaches, with memoryless grouping of events in time and space [30], [31]. Further development employed algorithms such as Density Based Spatial Clustering of Applications with Noise [32] and Mean Shift [33] to elaborate systems capable of performing accurate detections. When tracking is included, Particle Filters [34] and Principal Component Analysis [35] offer a higher detection accuracy due to the filtering on the detection signal.

As the detections become more challenging, the algorithms include feature matching as means for classification of different objects. Hand-picked features are typically used on very controlled environments exploiting probabilistic filters. Detected features are then compared against predefined shapes with Iterative Closest Point [36] or gradient descent algorithms [37], among others. The use of Attention Priority Maps [38] or Lock-In filters [39] on time surfaces generated from event accumulation can help focus the attention on the objects of interest and segment the background. Deterministic filters and activity filters can improve noise rejection, while keeping low computations.

On the black box models, Spiking Neural Networks exploit data learnt features to detect objects [40], [41]. These networks, specifically developed for neuromorphic sensors, are asynchronous models that use exponentially decaying values instead of binary triggers to propagate the calculations. One drawback of such networks is the large training required, as not only spatial but also temporal features should be learnt. However, due to event sparsity, computations on Spiking Neural Networks are reduced, up to 200 times, compared to traditional Neural Networks [40]. Other approaches used include Bag of Words [42] or Support Vector Machines [15], but with more restricted detection capabilities.

III. Methodology

The approach followed to determine the applicability of event-based cameras to drone swarm relative sensing is based on basic motion analysis of the propeller. Previous work on UAV detection with event cameras focused on shape fitting of the whole drone [40]. In such, the event camera was static and the drone was the only moving element in the scene. This however does not exploit the high temporal resolution of the sensor, and does not offer benefits over detection of any other object.

Further work recognised the benefit of focusing on the detection of propellers for drone localisation. Sanket et al. (2021) successfully detected rotating propellers when looking from above with event cameras [43]. For the first time, the uniqueness of propeller dynamics was exploited for detection on a moving platform. However, while this showed great potential for mid-air landing and following, the different drones shall always remain on a particular vertical configuration with small horizontal distance between them. This limits greatly their application for drone swarms, where the independence of the drones may yield many different configurations.

The study here presented also focuses on the detection of rotating propellers, characteristic of all drones. However, we focus on the scenario where all drones fly at the same height and the propeller's axis of rotatation is perpendicular to the camera's normal minimising the projection onto the image, as this presents the most challenging problem with the most applications. By studying the problem from first principles, it is possible to assess the detectability of neighbouring rotating propellers with event cameras.

The analysis performed is subjected to many factors that affect the distinction of the rotating propellers from the background-triggered events. This section first introduces the signal processing employed to assess the detection of rotating propellers with event cameras. Then, the first principles of the rotating propeller are studied to derive requirements on the analysis performed. Lastly, the signal properties of interest for detection are presented.

A. Discrete neuromorphic signal processing

When analysing a neuromorphic signal in a discrete and periodic way, there is inevitably an accumulation of information that occurs. Between each analysis step, there are a series of events that are triggered at asynchronous times. These events are then grouped under the same 'observation frame' and analysed. The method employed exploits this discrete analysis to assess the detection of rotating propellers. First, however, the notation followed throughout the report is introduced.

Shown in Figure 2 is a set of neuromorphic events. Each event is defined by x and y coordinates, a timestamp, and a polarity (represented with red and blue) indicating if the change in brightness that triggered the event was positive or negative. The event notation is described with the following equation, where event e_i belongs to the set E of all events.

$$e_i = ((x_i, y_i), t_i, p_i) \tag{1}$$

At each analysis step, the events grouped within the accumulation window (dashed lines in Figure 2) are evaluated

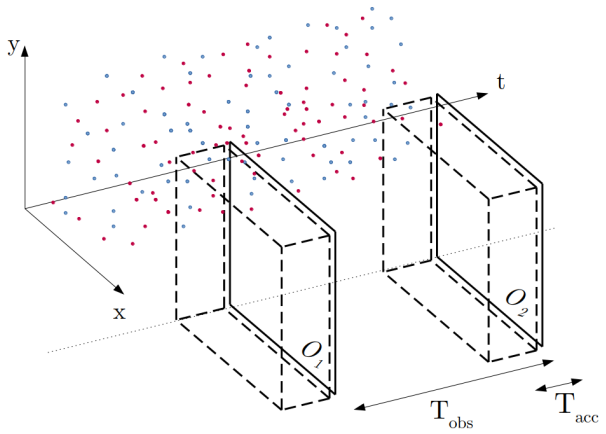


Fig. 2: Relation between observation frame O, observation time T_{obs} and accumulation time T_{acc}.

under observation frames ($o_{1,2}$ continuous lines). Since these observation frames are analysed for the detection of the propeller, the observation time, T_{obs}, sets the frequency at which the analysis is performed. The analysis frequency N is defined as $1/T_{obs}$.

It can also be seen that the accumulation time is independent of the observation time. In order not to loose information and process all events, T_{acc} should be at least equal to T_{obs}. When T_{acc} is different than T_{obs}, events are included into k observation frames equal to T_{acc}/T_{obs}.

Note that the term 'frame' is used in a relaxed manner. Observation frames are not typical frames where all the pixels contain a value. Rather they are a subset of all events that was triggered within the accumulation time. Equation 2 shows how frame i is composed of a subset of events, based solely on the events' timestamp.

$$O_i = \{e | e \in E, t_e \leq \frac{i}{N}, t_e > \frac{i-k}{N}\} \tag{2}$$

It may occur that multiple events in one observation frame have the same x and y coordinates. For each (x,y) coordinate, we define the event count $C(x, y)$ as the number of events triggered in that observation frame in those coordinates. The minimum event count C_{min} is the minimum required event count in any pixel to activate.

Lastly, we define two parameters of the rotating propeller that are frequently used. B is the number of propeller blades. f_{prop} is the rotating speed of the propeller, in Hz.

All the parameters defined, N, k, C, B, f_{prop}, influence the detection in a different way. N sets the analysis frequency. An increase in N translates to a higher temporal resolution. Looking at Equation 2, it is clear that over a one second interval there are N observation frames, each composed of the events triggered on the last k/N seconds. When k increases, more events are grouped into the same frame. One interesting observation is that, when the factor k/N is equal, frames with different values for k and N are equal, given the same i/N. The only difference being higher values of N achieving a higher time resolution.

TABLE I: P-values for $T_{sym} = 0.00256$ s for different N and k combinations.

N \ k	1	2	3	5	10	15	20
30	13	26	39	65	130	195	260
60	6.5	13	19.5	32.5	65	97.5	130
90	4.33	8.67	13	21.67	43.33	65	86.67
150	2.6	5.2	7.8	13	26	39	52
300	1.3	2.6	3.9	6.5	13	19.5	26
600	0.65	1.3	1.95	3.25	6.5	9.75	13
900	0.43	0.87	1.3	2.17	4.33	6.5	8.67
1500	0.26	0.52	0.78	1.3	2.6	3.9	5.2
3000	0.13	0.26	0.39	0.65	1.3	1.95	2.6
6000	0.07	0.13	0.2	0.33	0.65	0.98	1.3

The propeller parameters B and f_{prop} are of critical importance, as they set the duration of the to-be-detected signal. A propeller with B blades that spins at f_{prop} Hz has a symmetry period, T_{sym}, of $1/(B \cdot f_{prop})$, provided all blades are equal. A three-blade propeller spinning at 130 Hz has, for example, an appearance symmetry period of 0.00256 s. Relating this factor to the accumulation time, the periods per observation frame, P-value, is defined as the ratio of accumulation time to symmetry period, or $(k/N)/T_{sym}$. When the accumulation time k/N is smaller than the value of T_{sym} ($P < 1$), observations are grouped for less than a period, allowing the viewer to distinguish the periodic motion. When the accumulation time is higher than the propeller's symmetry period ($P > 1$), all observation frames contain the same events. A period per symmetry table shows the number of appearance symmetry periods for different combinations of N and k. Since T_{sym} is a property of the system observed, it offers a good way to select N and k. For example, for the case mentioned above where T_{sym} equals 0.00256 s, the period per symmetry table for a range of N and k values can be found in Table I. In such, values higher and lower than one are separated by a continuous line. It can be observed how, as N increases, the P-values as a function of k become finer. This is due to the value of k/N getting smaller, increasing the time resolution.

The minimum event count C_{min} sets a threshold on the observation frame for the minimum amount of events triggered at every pixel. On a deterministic sensor, where under the same input the same output is generated, this can be adjusted according to the P-value to filter the signal. When setting $C_{min} \leq P$, specially for higher values of P, the signal filters those events that are not highly periodic, leaving solely the propeller-triggered events. This however does not account for moving objects, as the pixels triggered by the rotating propeller are different as the neighbouring drones move in the scene. On a realistic scenario, where the drones have relative motion with respect to each other and may not have the exact conditions of contrast or lighting in time, the value of C_{min} should remain as low as possible (i.e., one), to avoid filtering the propeller signal and making detection more challenging.

The choice of C_{min} and P-value is also affected by the exact conditions under which pixels are triggered. Never-

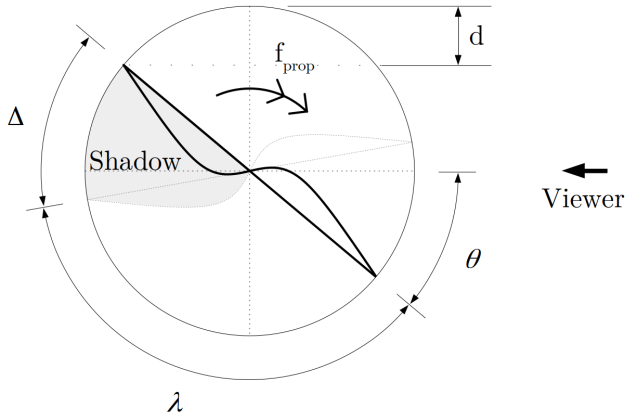


Fig. 3: Diagram of spinning propeller with reference variables.

theless, this is not known. Are the events triggered at the advancing edges of the propellers? Are there events being triggered also when the propeller view changes from leading edge to tip to trailing edge? Are there events being triggered by the vibrations of the observed drone? The sensitivity of the event camera to such subtle changes in brightness needs to be identified. The exact conditions are explored in section IV, after performing experiments on different conditions. However, it is first important to understand what is expected from observing rotating propeller. Section III-B presents a graphical analysis on rotating propellers to compare with the experimental case and establish the requirements needed to be fulfilled by the signal analysis.

B. Graphical propeller analysis

When analysing the rotating propeller, a series of observations can be made based on the premise of discrete analysis. The objective of this subsection is to relate the continuous propeller motion to the discrete analysis of the events. That way, while not making any assumptions on the generation of events, it is possible to associate propeller rotation to the shape of the events observed.

The appearance of the propeller can be determined at all times following a simple graphical analysis. Shown in Figure 3 is a propeller rotating at f_{prop} Hz being observed from the right, in this case with an event camera. The location of the propeller, θ, is defined as the angle between the camera's normal and the first propeller found. In this case, θ is constrained to the range [0, π). First, the scenario with a two blade propeller is presented, but the formulas for the general case (B-blade propeller) are also provided.

As mentioned in section III-B, the analysis is performed at a frequency N and events are accumulated for k/N seconds. Of course, during the time when events accumulate, the propeller rotates generating more events. The angle travelled by the propeller can be defined as Δ. Shown in Figure 3, this corresponds to the shadow region. The location of the propeller at the time of the analysis is depicted with a thick continuous line. In a light dotted line, the location of the propeller when events started accumulating is shown. More precisely, rotating at f_{prop} Hz, the shadow angle can be found using Equation 3.

$$\Delta = f_{prop} \cdot 2\pi \cdot k/N \tag{3}$$

Captured by the event camera, when θ is smaller than Δ, as is the case in Figure 3, there is an overlap on the projection of events onto the image. Events corresponding to different propellers are overlapped onto the same camera pixel. The overlap is maximised when θ equals $\Delta/2$, since at this location all events occur in the same window. This point is referred to as the angle of maximum overlap, $\theta_{max\,OL}$. A limit case exists, beyond which the signal becomes constant. When $\theta_{max\,OL}$ equals $\pi/2$ rad, the combined shadows of both propellers cover the complete circumference. Looking at the definition of $\theta_{max\,OL}$, it can be derived that this occurs when Δ equals π rad. In such case, the accumulation time is equal to, derived from Equation 3, $1/(2f_{prop})$. This result shows that, in order to achieve a distinguishable signal, the accumulation time shall be faster than twice the rotating speed of the propeller. A parallelism can be observed with the Nyquist frequency, where the sampling (i.e., signal accumulation frequency) needs to be performed at twice the spinning rate of the propeller.

Another important parameter that can be observed is the projection's maximum contraction. As the tip advances in the circumference rotating, only the sinusoidal component gets projected onto the camera. Considering once more $\theta_{max\,OL}$, the maximum contraction of the propeller's projection, it can be derived that the projection of the rotating propeller has an oscillatory length. Looking at Figure 3, the tip displacement is represented with d, an oscillating value between 0 m when the propeller is perpendicular to the camera's normal, and d_{max}, shown in Equation 4 as a function of the propeller's radius R, when θ equals $\theta_{max\,OL}$. Of course, if $\Delta/2$ is greater than or equal to $\pi/2$ rad the projection will be static and not oscillate.

$$d_{max} = R(1 - \sin(\Delta/2)) \tag{4}$$

Lastly, there is one more parameter that needs to be considered. As the propeller rotates, it triggers both positive and negative events. Negative events are triggered when the change in brightness is negative, and positive events when the change is positive. On the example of Figure 3, assuming the background is dark (typically drones have black bodies) and clear propellers are used for higher contrast, when the propeller appears on the image (quadrants two and four) the brightness increases causing positive events in those quadrants. Alternatively, when the propeller leaves the scene (quadrants one and three), the image of the clear propeller is replaced by the dark background, triggering negative events. Therefore, on each side of the propeller, it is possible to distinguish the overlapping propellers due to the sign of the events triggered. However, this distinction is limited. In Figure 3, the clearance between the incoming blade and the shadow of the blade in front is described as $\lambda = \pi - \Delta$. When

this angle is larger than $\pi/2$ rad, events of different polarities can be associated with different blades. When it is smaller, there can be no distinction between them, as the signal for both sides of the propeller becomes undistinguishable per polarity.

Before diving into the requirements imposed on the signal analysis, the above analysis can be generalised from a two-blade to a B-blade propeller. This way, a more comprehensive study can be performed on the system requirements. The shadow angle Δ does not change and is not affected by the number of propeller blades, only the speed at which they rotate and the accumulation time. The angle of maximum overlap between triggered events $\theta_{max\,OL}$ can be found with Equation 5. In such, $\pi/2 - \pi/B$ is the maximum static overlap between two blades for a B-blade propeller, to which the dynamic overlap is added. The maximum projected tip displacement d_{max} as a function of the number of propeller blades can be found with Equation 6. Finally, the shadow clearance angle λ can be found with Equation 7.

$$\theta_{max\,OL}(B) = \min\left(\frac{\pi}{2} - \frac{\pi}{B} + f_{prop} \cdot \pi \cdot k/N, \frac{\pi}{2}\right) \quad (5)$$

$$d_{max}(B) = R(1 - \sin\left(\theta_{max\,OL}\right)) \quad (6)$$

$$\lambda(B) = \max\left(\frac{2\pi}{B} - f_{prop} \cdot 2\pi \cdot k/N, 0\right) \quad (7)$$

Specially relevant are Equation 5 and Equation 7. It was presented above how $\theta_{max\,OL}$ affects the event overlap, and λ the per-side polarity association. In the case of the angle of maximum event overlap, an increase of B produces an increase of $\theta_{max\,OL}$. For the shadow clearance, which should be maximised to highlight polarity distinction, an increase of B produces a decrease of λ. Therefore, it is proven that, to ease the detection of propellers, a propeller with low amount of blades should be used.

C. Analysis requirements

The generalised propeller equations can be used to derive requirements on the analysis performed to ensure the propeller detection. These are based on the premise that the detections are performed on the oscillatory nature of the propeller, and are not a purely shape-based detection. This is considered since, on-board a flying drone, a large number of events are triggered in the scene. The periodicity of the propeller-triggered events characterises and highlights the neighbouring drones even on cluttered scenes. The detection therefore becomes a joint problem of detecting event clusters and comparing their polarity periodicity to that of the rotating propellers. If, on the other side, the event periodicity cannot be identified, the challenge increases dramatically.

Two main requirements can be imposed therefore on the signal analysis. One regarding the maximum overlap, and another regarding the polarity of the events.

The angle of maximum event overlap was defined as the angle θ at which the overlap between a blade's-triggered-events with the shadow of the preceding blade is maximum. Naturally, this can range between 0 and $\pi/2$, after which the overlap does not increase and the event count per observation frame becomes constant. Therefore a first requirement is imposed such that $\theta_{max\,OL} < \pi/2$. Substituting Equation 5 for the maximum accumulation time, it is concluded that $k/N < 1/(B \cdot f_{prop})$. This frequency, $B \cdot f_{prop}$, is known as the blade passing frequency, and is a common factor in aeroacoustic.

Considering the shadow clearance angle, a second requirement can be imposed to ensure a distinction of the polarity of the signal. In section III-B, the relation between λ and the polarity detection was introduced. When this angle is larger than $\pi/2$ rad, the accumulation of events still enables the association of event polarity to different blades, creating a periodic signal on the per-polarity events. When, on the other hand, it is smaller, this signal becomes continuous and is indistinguishable. The second requirement therefore imposes that $\lambda > \pi/2$. Once more, substituting the generalised Equation 7, it is derived that $k/N < (4 - B)/(4B \cdot f_{prop})$. One thing to notice from this formula is that, to meet this requirement, only propellers with two or three blades can be employed. If more blades are used, it is not possible to distinguish blades by polarity for each side of the propeller, as there is more than one blade in each quadrant.

D. Signal properties tested

To determine the applicability of event cameras for propeller detection in drone swarms, this study evaluates the signal generated by rotating propellers for cues that can then be identified. By analysing the signal in detail, it is possible to identify its properties and generate different metrics for classification. Only then, after assessing the intensity of these signal properties, can it be determined if event cameras are a suitable option for this use case.

The first and most obvious property analysed is the event count. To evaluate the detectability of rotating propellers, it is clearly important to examine their event count. If it is observed that rotating propellers generate a huge number of events, more than any other object in the scene, then they will be easy to detect. If, on the contrary, they generate a low number of events, they will blend in with the scene and be hard do detect.

The second signal property examined is the event clustering. When detecting rotating propellers, it is desired that the blades generate events in the form of a continuous blob that extends and contracts or that moves horizontally. If so, when a clear cluster is detected and can be traced, its movement can be analysed for resemblance with the expected shapes from the equations derived in section III-B. Otherwise, if the events triggered are sparse and cannot be grouped spatially, the detection task becomes more complex. All the pixels shall then be considered independently as potential propellers without inferring information from the surrounding pixels.

The last property studied is the signal periodicity. The main objective of this research is to exploit the properties of event cameras for the task of drone detection. Their high temporal resolution allows the observation of spinning

propellers, its most distinctive element due to their high periodicity. If an event cluster is observed oscillating periodically at a frequency matching that of a spinning propeller, the classification of the cluster as a propeller is evident. On the other hand, if the cluster does not show a clear periodic behaviour, there will be less certainty in the classification.

IV. Experiments

Having analysed in depth the propeller movement, it is clear how to perform the experiments to determine if event cameras are suited for rotating propeller detection. Being such a high frequency phenomenon, it is key to perform the observations in the appropriate frequency and with the correct accumulation, such that the propeller becomes visible. Studying the propeller in such way will determine if rotating propellers generate enough characteristic events to be detectable in the open space.

The tests are performed with that purpose. Without using flying drones, where the experimental space is very limited to the specific design of the drone used, the study performed focused around finding the limit case for which rotating propellers become distinguishable. Different propellers were recorded under different conditions at a range of distances and rotating speeds. Generating this database in a very controlled environment, it is possible to assess if, and under what conditions, do event cameras become useful for this application. This will also provide insight into techniques to improve the detectability of propellers, understanding what conditions generate the clearest detections.

A. Experimental setup

The experiments were performed in the following manner. The propellers were spun in a test bed controlling the rotating speed. This was recorded by an event camera held static, to record events only generated from the spinning propellers. To test the capabilities of the sensor, two different lighting conditions where tested. Moreover, two different backgrounds were used to evaluate the contrast influence on the detections.

For this experiment, the following hardware was used. First and most importantly, the iniVation (Zurich, Switzerland) DVS 240 [44] event camera was used. This camera was operated in conjunction with the iniVation DV software to record the events. For the test bed, the RC Benchmark Series 1580 [45] was employed. This propeller test rig allowed the control of the propeller through a ESC.

Five different propellers were tested, in order to evaluate the influence of different variables such as number of blades, size, colour and texture. The different propellers used can be seen in Figure 4. Two propellers with three blades and three with two blades were tested. One of the three blade propellers is bright pink, and the other is light blue and translucid. From the two blade propellers, one is light grey, another is shiny black, and the third is matte black. The light blue, grey and shiny black propellers have 6-inch-blades, and the pink and matte black have 5-inch-blades.

Fig. 4: Different propellers tested.

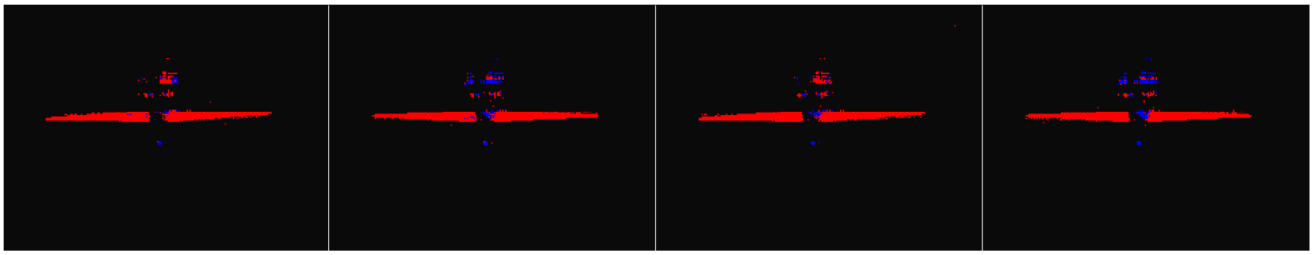
Two different lighting conditions were used for the experiments, one provided by natural sunlight and another by artificial lamps. Finally, two different backgrounds were employed. One was clean, uniform and light coloured, the other was dark grey and cluttered with irregular shapes.

The experiments were then performed testing a combination of all these parameters. All propellers were tested under all lighting and background conditions, for a the same range of distances and rotating speeds. The distances selected were equally distributed from 25 to 125 cm, at intervals of 10 cm. The rotating speeds selected were distributed unevenly: from 500 to 3000 RPMs at intervals of 500 RPMs, and from 3000 to 8000 RPMs at intervals of 1000 RPMs. Considering all variables (five propellers, two lightings, two backgrounds, eleven distances and eleven rotating speeds) a total of 2420 tests were performed.

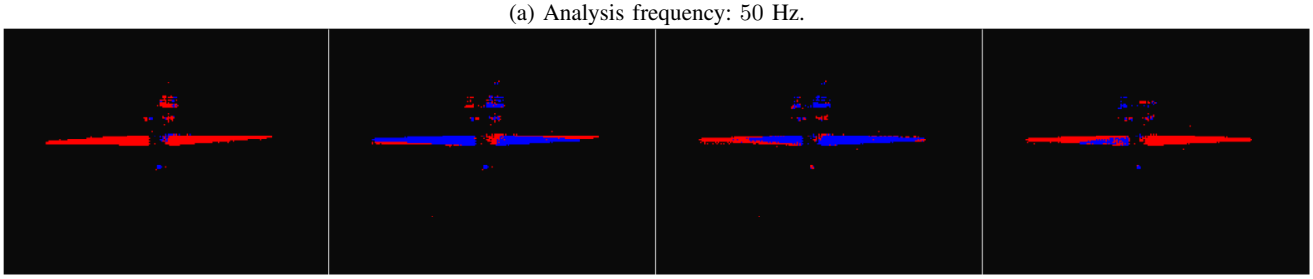
Considering the test variables discussed above, the selection was performed to meet all requirements. While different values for B were selected, to test different cases, the values of k and N are the same for all tests. The accumulation time is selected at 0.0005 s. This is meets the most strict requirement, stating $k/N < (4-B)/(4B \cdot f_{prop})$, for B equal three and f_{prop} equal 8000 RPMs. To keep computations affordable for onboard processing, the lowest resolution was selected, where k is one frame and N is 2000 Hz.

This selection of accumulation time and frequency analysis, together with the theory presented above, can be easily validated. It is desired that the rotations of the propeller are clearly distinguishable. When analysing the raw observation frames visually, all the properties described above can be recognised. Figure 5 shows consecutive observation frames for the two-blade six-inch black propeller spinning at 1500 RPMs with artificial lighting and clear background as an example, when employing different analysis frequencies (and keeping the value of k constant at one). Frames in Figure 5a, Figure 5b, Figure 5c, Figure 5d, Figure 5e and Figure 5f are gathered at 50 Hz, 70 Hz, 100 Hz, 200 Hz, 2000 Hz and 4000 Hz, respectively.

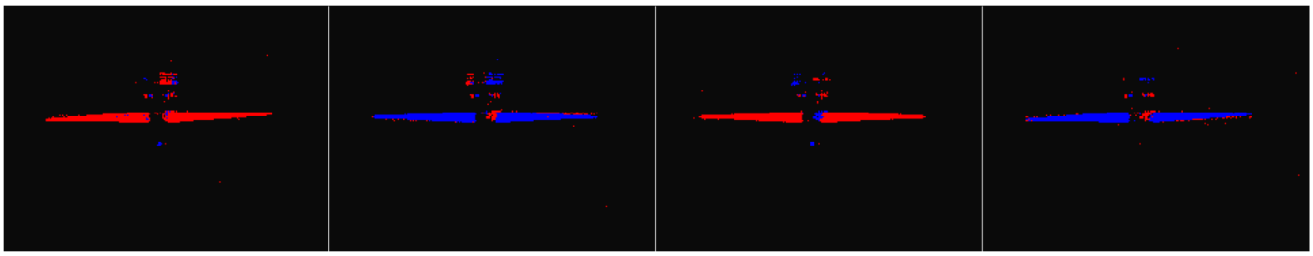
This choice of frequencies illustrates the behaviour of the propeller around the different minimum frequencies presented in section III-C, as well as the actual analysis



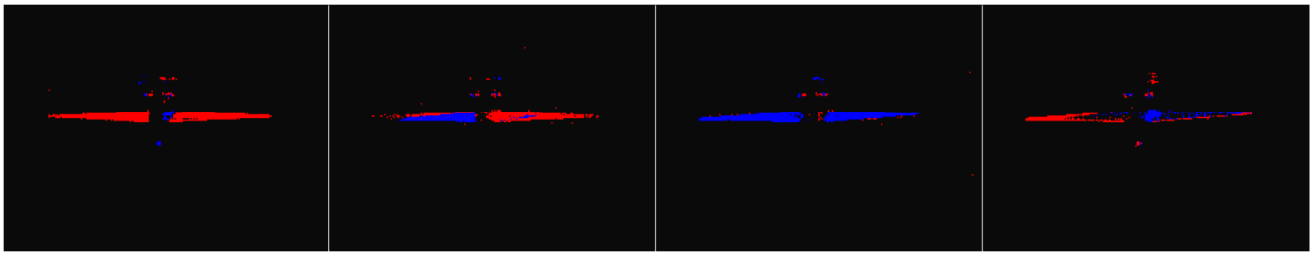
(a) Analysis frequency: 50 Hz.



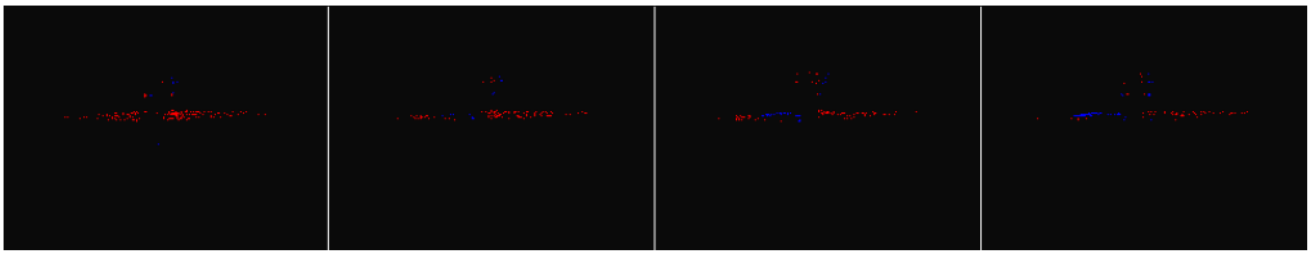
(b) Analysis frequency: 70 Hz.



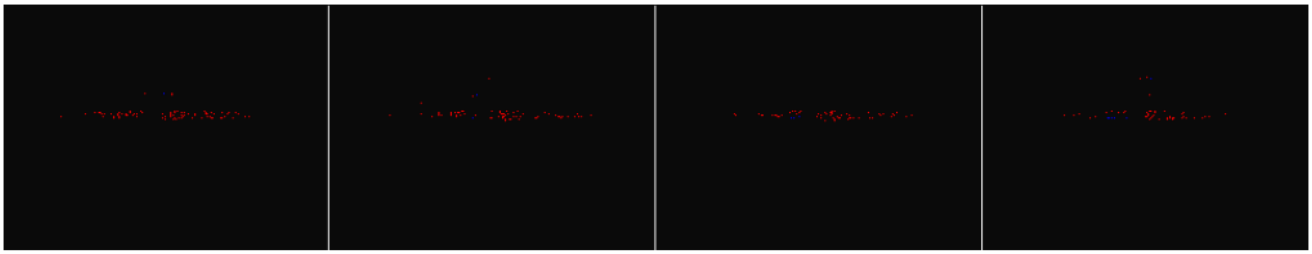
(c) Analysis frequency: 100 Hz.



(d) Analysis frequency: 200 Hz.



(e) Analysis frequency: 2000 Hz.



(f) Analysis frequency: 4000 Hz.

Fig. 5: Four consecutive observation frames of the two-blade six-inch black propeller spinning at 1500 RPMs with artificial lighting and clear background, captured at different analysis frequencies.

frequency and faster. In the experiment performed, the blade passing frequency equals $B \cdot f_{prop} = 50$ Hz. This matches perfectly the result of Figure 5a, as it displays a constant appearance. The next frequency lies between the blade passing frequency and the requirement $\lambda > \pi/2$ that states $(4B \cdot f_{prop})/(4 - B) = 100$ Hz. At 70 Hz, the different polarities can be distinguished, but there is still overlap between the event polarities of different blades. 100 Hz sets the limit of the distinction between the different polarities, as shown in Figure 5c where the accumulation for each blade is high enough to fill up exactly half of the disc. This frequency is also the limit below which aliasing occurs. From this frequency on, at 200 Hz for example, a clear advancing profile can be distinguished, showing the need to sample above the minimum required frequency.

The last two frequencies shown, 2000 Hz and 4000 Hz, are displayed to prove that even at such high speeds the advancing profile can be seen. Of course, at high analysis frequencies there is a smaller event accumulation, causing less events to appear on each observation frame. However, there are still enough to be identified as blades. Even at 4000 Hz the blade is clearly visible. The choice of 2000 Hz for the analysis frequency of all tests is therefore suitable to meet the requirements for all propellers at all spinning rates, while still allowing for blade detection under realistic computational power.

B. Metrics analysed

For all these tests, one second recordings were performed where the propeller was spinning steadily. These recordings were then processed to obtain four different metrics for each test. That way, the huge number of raw events could be converted to meaningful values that can be used to evaluate the detectability of the spinning propeller. These metrics were derived following the keys presented in section III-D, where the importance of event count, signal periodicity and event clustering is described.

The first metric derived is the total event count. For all tests, the total number of events triggered by the propeller was counted, since this is a direct indicator of the detectability of the spinning propeller. When recording a given propeller, the events triggered as a function of time can be observed in Figure 6a. In this case, the signal corresponds to the matte black two-blade propeller, with natural lights and clear background, spinning at 2500 RPMs and recorded from a distance of 35 cm. The event count is calculated as the average of the positive and negative events triggered during the whole one second recording.

The second metric derived is the event count periodic prominence. The periodicity of the signal is critical for its classification as a propeller. This is evaluated on the per-polarity event count, analysing its time evolution. Looking at Figure 6a, a clear periodicity in the signal is observed. It is possible to apply a Fourier Transform and extract the frequency components from the event count time signal. The Fourier Transform of the signals in Figure 6a can be seen in Figure 6b. The spinning frequency of the propeller, in Hz, is shown with a dashed line. The peak at that frequency shows the high energy level of that frequency in the time signal. The zero component of the Fourier Transform just represents the average of the signal, and the higher peaks are due to the discretisation of the signal. The signal periodicity is therefore calculated as the value (energy) of the highest peak in the surrounding of the propeller frequency. Once more, the average of the positive and negative values is used.

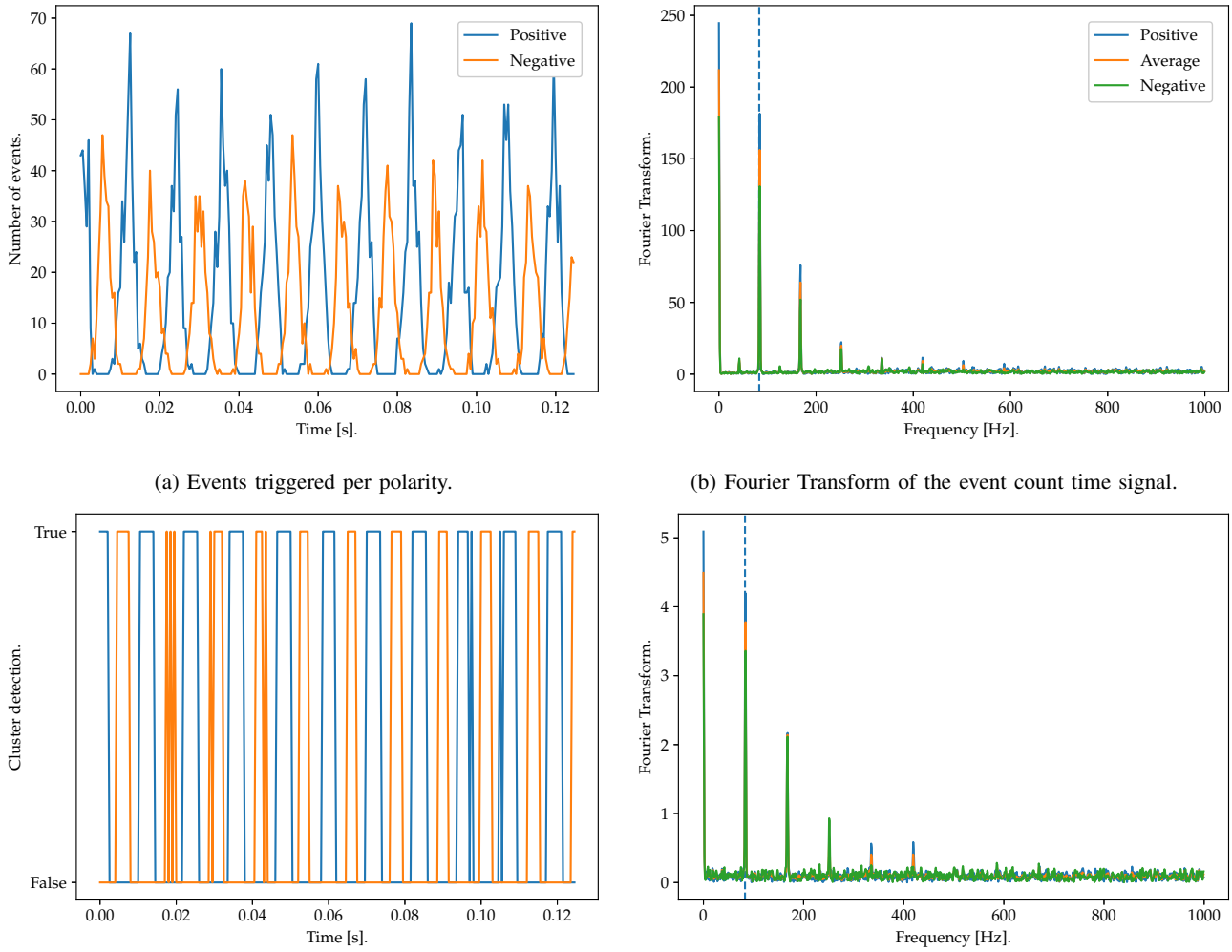
The third metric is the event clustering. This focused on the spatial proximity of the events for each observation frame, to gain insight into the sparsity or concentration of events. Of course, it is desired that the rotating blades generate a continuous and defined versus a scattered and sporadic stream of events. To evaluate this, the DBSCAN algorithm was employed to locate clusters for both, positive and negative events. [46] The cluster with biggest Intersection over Union was then selected as the principal one for that frame. This allowed a coarse classification of frames into positive, negative, or none, depending on the polarity of the biggest cluster. An example of the clustering results can be found in Figure 6c, where the detection of positive and negative cluster is shown for the signal presented in Figure 6a. In this example, a large number of clusters is detected, showing a high concentration of events. The event clustering is calculated as the total number of clusters detected over the one second recording.

The fourth and last metric is the event cluster periodic prominence. Similarly for the case of the event count signal, the periodicity of the event clustering is essential for its characterisation as a propeller. Following the same approach as above, computing the Fourier Transform of the whole one second recording for the clustering in Figure 6c can be shown in Figure 6d. Once more, the propeller spinning frequency is shown in a dashed line matching the frequency with the highest energy (besides the zero frequency representing the average). The event cluster periodic prominence is calculated as the value of the highest frequency in the vicinity of the propeller frequency, taking the average value as reference.

C. Results

The results for the different combination of propeller, lighting, background spinning rate and distance can then be directly calculated. The plots corresponding to three different test setups can be found in Figure 7, Figure 8 and Figure 9. Figure 7 corresponds to the two-blade five-inch matte-black propeller, with artificial lighting and clear background. Figure 8 corresponds to the two-blade six-inch grey propeller, with natural lighting and clear background. Figure 9 corresponds to the three-blade six-inch translucid blue propeller, with natural lighting and clear background. These three examples are chosen as they cover the spectrum of the results obtained.

Figure 7 shows the results for the best performing setup. The tests performed at close distances, between 25 and 45 cm, triggered a huge number of events of up to 140000 with a very high periodicity, reaching values of 700. Moreover, these events were spatially concentrated, allowing a



(a) Events triggered per polarity.

(b) Fourier Transform of the event count time signal.

(c) Clustering of events per polarity.

(d) Fourier Transform of the event clustering time signal.

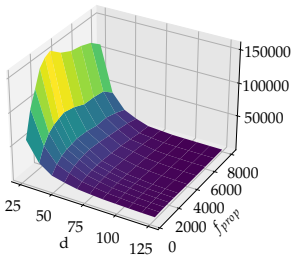
Fig. 6: Signals analysed for the two-blade five-inch matte black propeller, with natural light and clear background, spinning at 2500 RPMs and recorded from a distance of 35 cm.

large amount of cluster detections - over 1500 out of the 2000 frames observed - with also high periodicity. However, this high event count and event periodicity suffers an exponential decrease as the distance get higher. Regarding the clustering, between 45 and 75 cm there is an almost linear decrease of the values, beyond which the values for clustering count and periodicity remain close to zero. As it can be seen, the behaviour described is quasi-independent of the propeller spinning frequency.

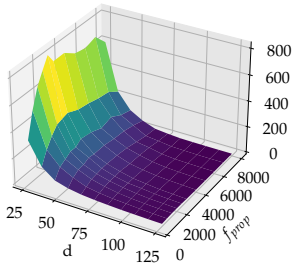
Figure 8 shows the results for a setup with average results. In this case, the behaviour is similar to the one described above, but is more affected by the spinning frequency. For low propeller frequencies, the shape of the different plots is almost identical to Figure 7, but with slightly lower values for the event count and periodicity. However, there is a limit frequency around 5000 RPMs beyond which all values rapidly decrease to zero.

Lastly, Figure 9 shows the results for one of the worst performing setups. For such case, only at very close distances and low spinning frequencies are some events triggered. These have some periodicity, but are extremely sparse, since no clusters are detected. As the propeller is recorded spinning faster or from further distances, the propeller becomes invisible for the camera as no events are recorded.

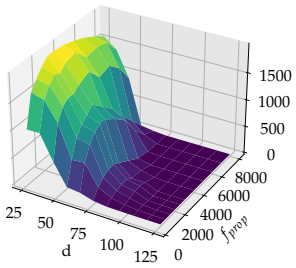
From the differences between the three setups, it is clear that the propeller spinning frequency has a higher effect on the events generated than the recording distance on the different setups. As the contrast decreases between the propeller and the background, it is the higher frequency recordings that stop triggering events first, while the maximum distance at which events are triggered remains constant. In the extreme, in Figure 9, it can be observed how only for the lowest frequency events are still triggered for different distances. Two main ideas can be extracted from this observation.



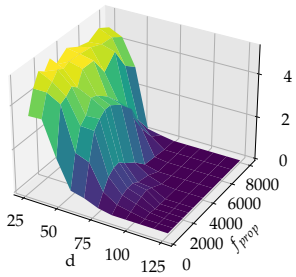
(a) Event count.



(b) Event periodicity.

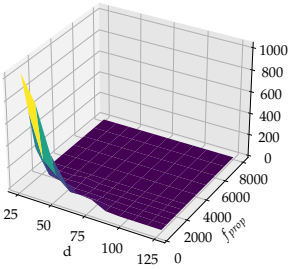


(c) Clustering count.

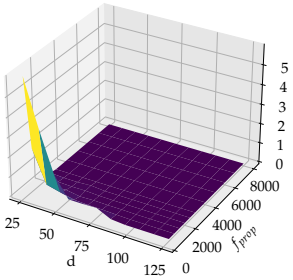


(d) Clustering periodicity.

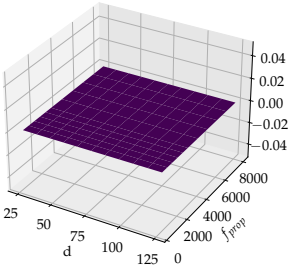
Fig. 7: Results for two-blade five-inch matte blade black propeller, with artificial lighting and clear background versus distance [cm] and propeller frequency [RPMs].



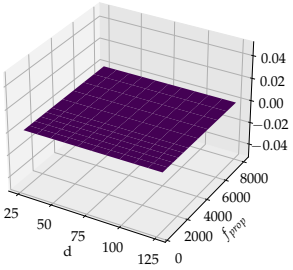
(a) Event count.



(b) Event periodicity.

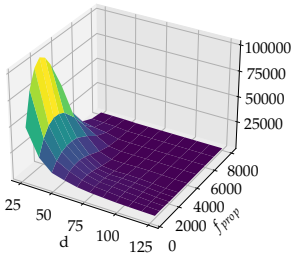


(c) Clustering count.

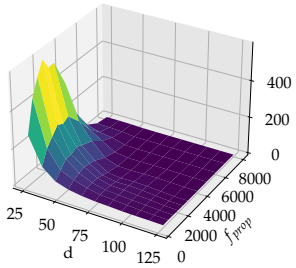


(d) Clustering periodicity.

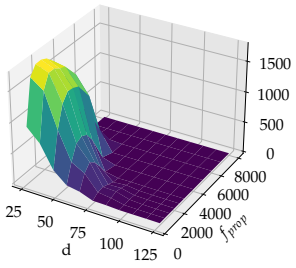
Fig. 9: Results for three-blade six-inch translucid blue propeller, with natural lighting and clear background versus distance [cm] and propeller frequency [RPMs].



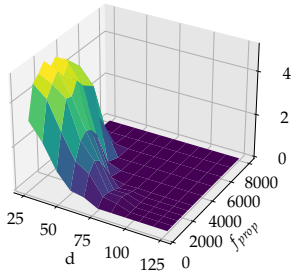
(a) Event count.



(b) Event periodicity.



(c) Clustering count.



(d) Clustering periodicity.

Fig. 8: Results for two-blade six-inch grey propeller, with natural lighting and clear background versus distance [cm] and propeller frequency [RPMs].

Firstly, this indicates that each setup has a limit propeller frequency that can be detected. The contrast between the blade and the background is what establishes this limit. If this is not high enough, the change in the pixel signal at the sensor level oscillating between the blade and the background does not reach the minimum delta to trigger events. Secondly, it suggests that the distance limit is not caused due to the test conditions, but rather on the sensor used. There is a clear limit in the recorded distance at around 85 cm. Beyond this limit, the camera resolution is too low to detect such subtle changes in brightness. This is a consequence of dealing with a sensor in its early stages. Other event camera models do offer a higher resolution, but at the expense of lower temporal resolution. A trade-off must therefore be performed.

One last aspect to highlight is the similarity between the counts and periodicities, for both events and clusters. In all cases there is a high resemblance in the shape of the plots. This is a good indicator that the analysis is performed at the correct frequency and there is no aliasing since the signal period is clearly distinguished. If the analysis was performed at a lower frequency, the event count time signal would not oscillate at the period of the rotating propeller, and such frequency would not have any energy in the Fourier Transform, yielding values of zero periodicity regardless of the event count.

D. CP metric

An interesting observation can be made regarding the differences between the event count and periodicity against the clustering count and periodicity plots. On one hand, the event count plots show an exponentially decreasing value with respect to the distance. On the other hand, three clearly different sections can be observed in the cluster plots: one flat and elevated, one linearly decreasing, and another flat area around zero. The three different regions can be seen most clearly in Figure 7c and Figure 7d.

From this difference in the plots, it is derived that the clustering is fully dependent on the event count, since the regions with no events have obviously no clusters. However, it can be noted that, after a minimum amount of events is triggered, an increase in the number of events does not suppose an increase in the number of detections. That is because the maximum number of detections is constrained by the analysis frequency and the maximum overlap between the different blades. For the case of Figure 7, the number of clusters is very similar between 25 and 45 cm, while the event count decreases from 140000 events at 25 cm to 25000 events at 45 cm. Likewise, there is minimum number of events triggered in order to generate clusters reliably. Beyond 75 cm there are almost no clusters detected, while the number of events detected ranges from 5000 to zero.

Since all four requirements need to be met for a detection to be characterised as a propeller - high number of events and clusters, with high periodicity of both - it is possible to derive one last metric to define propeller detection in a given test using the observation above. A fuzzy classifier can be applied on the event count and periodicity that matches the shape of the clustering plots. By finding the minimum and maximum number of events and corresponding periodicity values that allow clustering, the tests' fuzzy detection membership can be determined on the range $[0, 1]$. This way, it is possible to directly correlate the event count and periodicity and cluster count and periodicity results to a single number that determines if a test is detectable or not. The key is to identify the event count and periodicity that matches the transitions between the three regions in the clustering plots. After averaging these values over all the tests performed, the count metric o_c can be found with Equation 8 as a function of the event count c, and the periodicity metric o_p can be found with Equation 9 as a function of the event periodicity p. These two equations determine the clustering capabilities for a given event count and periodicity. The total Count-Periodicity metric, or CP metric, is then calculated as the multiplication of the individual components, $o_c \cdot o_p$. This metric is therefore used to assess if a propeller can be detected on the tested conditions.

$$o_c = \begin{cases} 1 & \text{if } c > 10000 \\ 0 & \text{if } c < 2500 \\ \frac{c-2500}{10000-2500} & \text{else} \end{cases} \tag{8}$$

$$o_p = \begin{cases} 1 & \text{if } p > 100 \\ 0 & \text{if } p < 40 \\ \frac{p-40}{100-40} & \text{else} \end{cases} \tag{9}$$

With this metric it is possible to show the propeller detection for the different setups. The results can be found in Figure 10, Figure 11, Figure 12, Figure 13, Figure 14 for the three-blade six-inch translucid blue propeller, two-blade six-inch grey propeller, two-blade six-inch black propeller, three-blade five-inch pink propeller and two-blade five-inch matte black propeller, respectively.

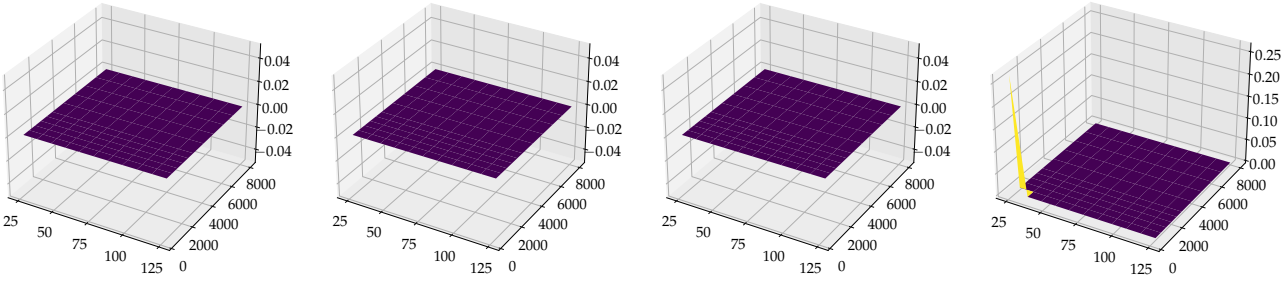
A great difference in the results can be observed between the two- and three-blade propellers. The three blade propellers display, for almost all setups, completely null propeller detection. Only with artificial lighting and cluttered background the metric displays a non-zero value, but only for the lowest frequency at the shortest distance, and with a value of around 0.1 and 0.2 for the five-inch pink propeller and six-inch blue propeller, respectively.

For the two-blade propellers, the performance is more distributed. Once more, the detection limit at 75 cm can be observed. For the higher frequencies, between 6000 and 8000 RPMs, where the drones fly, the best performances are by the six-inch black propeller and five-inch matte black propeller, for the scenario with the artificial lighting. In this case, the propellers are detectable up to 45 cm and 35 cm for the five-inch matte black and six-inch black propeller, respectively. This therefore becomes the maximum detection distance for flying drones.

In order to compare the results, the different setups can be ranked based on the area under the CP metric: the CP area. This value has no physical meaning, but can serve as a metric to compare the different tests and extract conclusions on the influence of the different parameters. The results presented in Table II show the ranked setups, also displaying the normalised area value with respect to the best test.

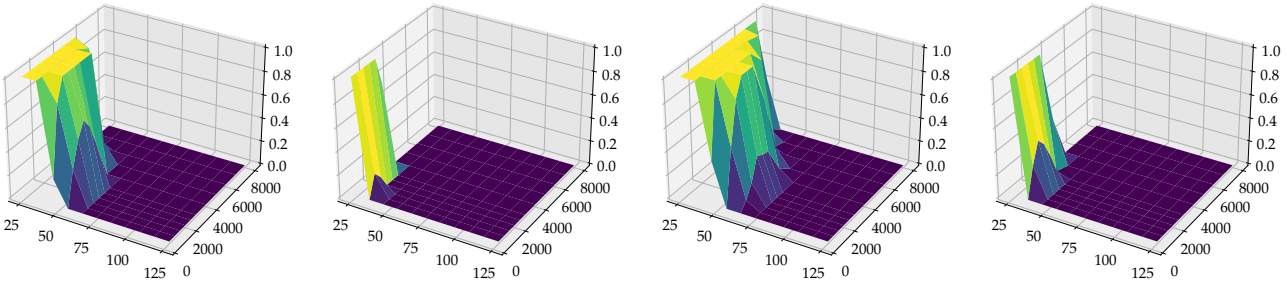
First, it is most noticeable the difference between the three-blade and the two-blade propellers. While two-blade propellers have a varying relative propeller detection, three-blade propellers have absolute zero (with two negligible exceptions). It can therefore be concluded that three blade propellers, with their static blade overlap, do not generate a sufficient number of events or clusters to be detected.

The second most influential factor, as shown by Table II, is the lighting conditions. The top five performing setups have artificial lighting, followed by five setups with natural lighting. The only two exceptions occur with the two-blade six-inch grey propeller. However, this can be explained by the background. The cluttered background, as described in section IV-A, was dark grey with irregular shapes. This background produces much less contrast with the grey propeller, explaining its poor performance. The artificial lighting increases the brightness of the surfaces, allowing larger contrasts. Even for the three-blade propellers, only those with artificial lighting showed a minimum propeller detection, while the rest showed null results.



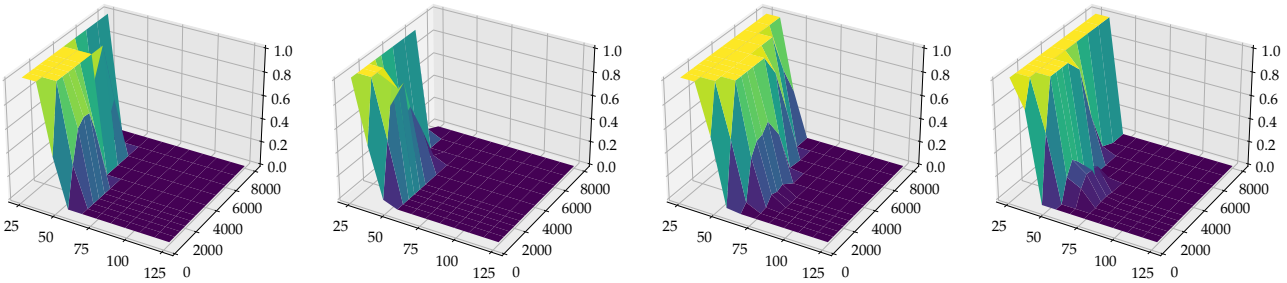
(a) Natural, clear. (b) Natural, cluttered. (c) Artificial, clear. (d) Artificial, cluttered.

Fig. 10: Propeller detection results for the three-blade six-inch translucid blue propeller, for different lightings and backgrounds.



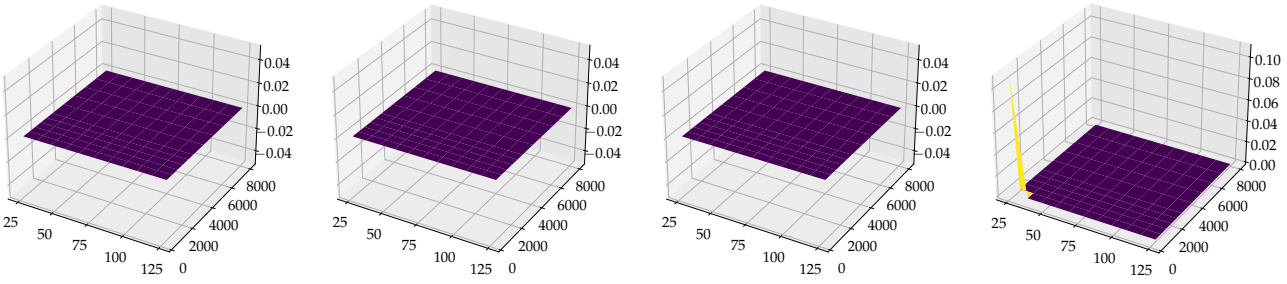
(a) Natural, clear. (b) Natural, cluttered. (c) Artificial, clear. (d) Artificial, cluttered.

Fig. 11: Propeller detection results for the two-blade six-inch grey propeller, for different lightings and backgrounds.



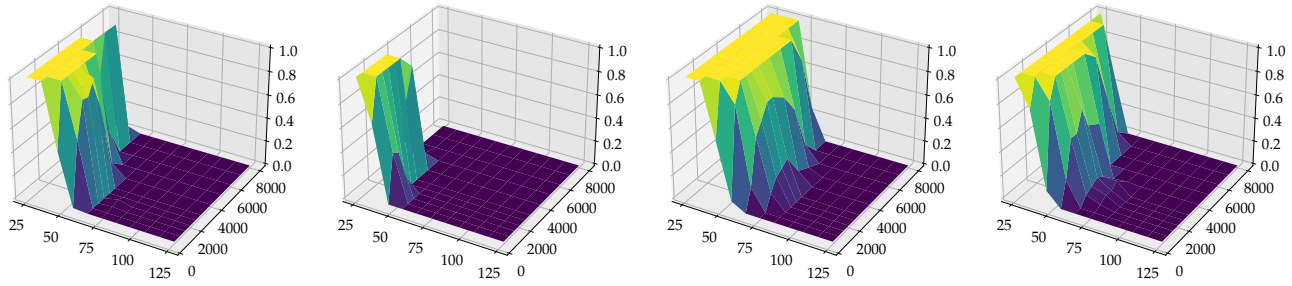
(a) Natural, clear. (b) Natural, cluttered. (c) Artificial, clear. (d) Artificial, cluttered.

Fig. 12: Propeller detection results for the two-blade six-inch black propeller, for different lightings and backgrounds.



(a) Natural, clear. (b) Natural, cluttered. (c) Artificial, clear. (d) Artificial, cluttered.

Fig. 13: Propeller detection results for the three-blade five-inch pink propeller, for different lightings and backgrounds.



(a) Natural, clear. (b) Natural, cluttered. (c) Artificial, clear. (d) Artificial, cluttered.

Fig. 14: Propeller detection results for the two-blade five-inch matte black propeller, for different lightings and backgrounds.

TABLE II: Ranking of the propeller detection of the different setups.

#	CP area (norm.)	Propeller	Lighting	Background
1	1	2-blade 5′ matte black	Artificial	Clear
2	0.84	2-blade 6′ black	Artificial	Clear
3	0.64	2-blade 6′ grey	Artificial	Clear
4	0.63	2-blade 5′ matte black	Artificial	Cluttered
5	0.55	2-blade 6′ black	Artificial	Cluttered
6	0.55	2-blade 5′ matte black	Natural	Clear
7	0.52	2-blade 6′ black	Natural	Clear
8	0.44	2-blade 6′ grey	Natural	Clear
9	0.27	2-blade 6′ black	Natural	Cluttered
10	0.2	2-blade 5′ matte black	Natural	Cluttered
11	0.08	2-blade 6′ grey	Artificial	Cluttered
12	0.05	2-blade 6′ grey	Natural	Cluttered
13	0.001	3-blade 6′ inch blue	Artificial	Cluttered
14	0.0005	3-blade 5′ pink	Artificial	Cluttered
15	0	3-blade 6′ blue	Natural	Clear
16	0	3-blade 6′ blue	Natural	Cluttered
17	0	3-blade 6′ blue	Artificial	Clear
18	0	3-blade 5′ pink	Natural	Clear
19	0	3-blade 5′ pink	Natural	Cluttered
20	0	3-blade 5′ pink	Artificial	Clear

The third variable with the highest influence is the background. The clear background presents more contrast with the dark propellers, generating more events and allowing a higher CP metric, for both artificial and natural lighting. The three blade propellers, on the other hand, had light colours, and therefore presented more contrast with the dark cluttered background, allowing a minimum score in such cases.

Lastly, the parameter with the smallest influence is the propeller shape and texture. For example, the two-blade five-inch matte black has a slightly higher CP score for most scenarios than the two-blade six-inch black propeller. Even for the clear backgrounds, the two-blade six-inch grey propeller joins the other two-blade propellers in the ranking grouping. However, its performance is considerably lower, with a noticeable smaller CP area. To be noted is the little effect of the size difference between the five-inch and six-inch black propellers, as the smaller propeller shows a generally better propeller detection.

V. Conclusions

In this paper, the applicability of event cameras for propeller detection is analysed. Different propellers were tested under different conditions of contrast and luminosity for a range of distances and spinning frequencies to assess the influence of each variables in their detection. A series of metrics were developed to evaluate the propeller detection for the different tests involving event count, periodicity and clustering. Moreover, a fuzzy classifier was developed to classify the different setups and determine if a propeller can be detected.

The detection of the spinning propellers was very limited due to the complexity of the task and the limitations of the sensor employed. Firstly, a maximum detection distance of 75 cm was observed, beyond which almost no events are detected due to low sensor resolution. Employing the CP metric derived, the following results were found for two-blade propellers. For high spinning frequencies, between 6000 and 8000 RPMs, the propeller signal looses energy and becomes almost invisible. A maximum detection distance of 45 cm for propellers with large contrast (black propeller and clear background) and suitable lighting conditions is detected. For the rest of the cases, propellers are only detectable at low frequencies. Three-blade propellers on the other hand do not offer sufficient contrast and are almost undetectable for all cases, regardless of the setup.

These limitations show that this technology is still at an early stage for such a complex application. Currently, detections are limited to two-blade propellers with high proximity and large visual contrasts. In such conditions, swarm coordination is limited, but the system could successfully be employed for collision avoidance, for example, or in conjunction with other relative sensing subsystems to provide further confidence at short distances.

In order to extend the applicability of event cameras to enable swarming behaviour, further work must be focused on the improvement of the sensor itself. Increasing the pixel and time resolution will allow for further detections for a higher range of setups. Lastly, it is also believed that the use of more complex algorithms such as Spiking Neural Networks may improve detections, and the study of such methods is recommended for further work.

REFERENCES

[1] M. Coppola, K. N. McGuire, C. De Wagter, and G. C. de Croon, "A survey on swarming with micro air vehicles: Fundamental challenges and constraints," *Frontiers in Robotics and AI*, vol. 7, p. 18, 2020.

[2] D. K. Behera and A. Bazil Raj, "Drone detection and classification using deep learning," in *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 1012–1016, 2020.

[3] U. Seidaliyeva, D. Akhmetov, L. Ilipbayeva, and E. T. Matson, "Real-time and accurate drone detection in a video with a static background," vol. 20, 2020.

[4] L. Mao, J. Chen, Z. Li, and D. Zhang, "Relative localization method of multiple micro robots based on simple sensors," *International Journal of Advanced Robotic Systems*, vol. 10, no. 2, p. 128, 2013.

[5] J. Pugh and A. Martinoli, "Relative localization and communication module for small-scale multi-robot systems," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pp. 188–193, 2006.

[6] A. A. Cabrera-Ponce, J. Martinez-Carranza, and C. Rascon, "Detection of nearby UAVs using a multi-microphone array on board a UAV," *International Journal of Micro Air Vehicles*, vol. 12, p. 1756829320925748, 2020.

[7] M. Basiri, F. Schill, P. Lima, and D. Floreano, "On-board relative bearing estimation for teams of drones using sound," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 820–827, 2016.

[8] F. Rivard, J. Bisson, F. Michaud, and D. Letourneau, "Ultrasonic relative positioning for multi-robot systems," in *2008 IEEE International Conference on Robotics and Automation*, pp. 323–328, 2008.

[9] G. Retscher, V. Gikas, H. Hofer, H. Perakis, and A. Kealy, "Range validation of uwb and wi-fi for integrated indoor positioning," vol. 11, pp. 187–195, 2019.

[10] G. Gallego, T. Delbruck, G. M. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, and et al., "Event-based vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1–30, 2020.

[11] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128× 128 120 db 15 μs latency asynchronous temporal contrast vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[12] T. Delbruck and M. Lang, "Robotic goalie with 3 ms reaction time at 4% CPU load using event-based dynamic vision sensor," *Frontiers in Neuroscience*, vol. 7, p. 223, 2013.

[13] A. Glover and C. Bartolozzi, "Event-driven ball detection and gaze fixation in clutter," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2203–2208, 2016.

[14] G. Orchard, C. Meyer, R. Etienne-Cummings, C. Posch, N. Thakor, and R. Benosman, "HFirst: a temporal approach to object recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 10, p. 2028–2040, 2015.

[15] B. Ramesh, A. Ussa, L. Della Vedova, H. Yang, and G. Orchard, "Low-power dynamic object detection and classification with freely moving event cameras," *Frontiers in Neuroscience*, vol. 14, p. 135, 2020.

[16] D. R. Valeiras, X. Lagorce, X. Clady, C. Bartolozzi, S.-H. Ieng, and R. Benosman, "An asynchronous neuromorphic event-driven visual part-based shape tracking," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 12, pp. 3045–3059, 2015.

[17] A. Amir, B. Taba, D. Berg, T. Melano, J. McKinstry, C. Di Nolfo, T. Nayak, A. Andreopoulos, G. Garreau, M. Mendoza, J. Kusnitz, M. Debole, S. Esser, T. Delbruck, M. Flickner, and D. Modha, "A low power, fully event-based gesture recognition system," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7388–7397, 2017.

[18] M. Vrba, D. Heřt, and M. Saska, "Onboard marker-less detection and localization of non-cooperating drones for their safe interception by an autonomous aerial system," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3402–3409, 2019.

[19] V. Walter, M. Saska, and A. Franchi, "Fast mutual relative localization of UAVs using ultraviolet LED markers," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1217–1226, 2018.

[20] V. Walter, N. Staub, A. Franchi, and M. Saska, "UVDAR system for visual relative localization with application to leader–follower formations of multirotor UAVs," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2637–2644, 2019.

[21] E. Ferro and F. Potorti, "Bluetooth and wi-fi wireless protocols: a survey and a comparison," *IEEE Wireless Communications*, vol. 12, no. 1, pp. 12–26, 2005.

[22] S. Saeedi, M. Trentini, M. Seto, and H. Li, "Multiple-robot simultaneous localization and mapping: A review," *Journal of Field Robotics*, vol. 33, no. 1, pp. 3–46, 2016.

[23] A. A. Micheal and K. Vani, "Automatic object tracking in optimized UAV video," *The Journal of Supercomputing*, vol. 75, no. 8, 2019.

[24] W. Lei, D. Huang, and X. Cui, "Moving object tracking in video surveillance using yolov3 and meanshift," in *Tenth International Conference on Graphics and Image Processing (ICGIP 2018)*, vol. 11069, p. 86, 2019.

[25] A. Glover and C. Bartolozzi, "Robust visual tracking with a freely-moving event camera," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3769–3776, 2017.

[26] D. Gordon, A. Farhadi, and D. Fox, "Re3: Real-time recurrent regression networks for visual tracking of generic objects," *IEEE Robotics and Automation Letters*, vol. PP, pp. 1–1, 2018.

[27] H. Chen, D. Suter, Q. Wu, and H. Wang, "End-to-end learning of object motion estimation from retinal events for event-based object tracking," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, p. 10534–10541, 2020.

[28] K. Fang, "Track-RNN : joint detection and tracking using recurrent neural networks," 2016.

[29] Y. Wang, K. Kitani, and X. Weng, "Joint object detection and multi-object tracking with graph neural networks," 2020.

[30] D. Drazen, P. Lichtsteiner, P. Häfliger, T. Delbrück, and A. Jensen, "Toward real-time particle tracking using an event-based dynamic vision sensor," *Experiments in Fluids*, vol. 51, 2011.

[31] M. Litzenberger, C. Posch, D. Bauer, A. N. Belbachir, P. Schon, B. Kohn, and H. Garn, "Embedded vision system for real-time object tracking using an asynchronous transient vision sensor," in *2006 IEEE 12th Digital Signal Processing Workshop 4th IEEE Signal Processing Education Workshop*, pp. 173–178, 2006.

[32] H. Patel, C. Iaboni, D. Lobo, J.-w. Choi, and P. Abichandani, "Event camera based real-time detection and tracking of indoor ground robots," *arXiv e-prints*, p. arXiv:2102.11916, 2021.

[33] F. Barranco, C. Fermuller, and E. Ros, "Real-time clustering and multi-target tracking using event-based sensors," 2018.

[34] A. Glover and C. Bartolozzi, "Robust visual tracking with a freely-moving event camera," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3769–3776, 2017.

[35] M. B. Milde, O. J. N. Bertrand, R. Benosman, M. Egelhaaf, and E. Chicca, "Bioinspired event-driven collision avoidance algorithm based on optic flow," in *2015 International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)*, pp. 1–7, 2015.

[36] Z. Ni, A. Bolopion, J. Agnus, R. Benosman, and S. Regnier, "Asynchronous event-based visual shape tracking for stable haptic feedback in microrobotics," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1081–1089, 2012.

[37] Z. Ni, S.-H. Ieng, C. Posch, S. Régnier, and R. Benosman, "Visual tracking using neuromorphic asynchronous event-based cameras," *Neural Computation*, vol. 27, no. 4, pp. 925–953, 2015.

[38] J. P. Rodriguez Gomez, A. Gomez Eguiluz, J. R. Martinez-de Dios, and A. Ollero, "Asynchronous event-based clustering and tracking for intrusion monitoring in UAS," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8518–8524, 2020.

[39] B. Rosas-Flores, A. Hernández-Zavala, and J. Huerta-Ruelas, "Lightweight obstacle detector based on scattered IR and lock-in filtering," *Infrared Physics & Technology*, vol. 105, p. 103157, 2020.

[40] P. Kirkland, G. Di Caterina, J. Soraghan, Y. Andreopoulos, and G. Matich, "UAV detection: a STDP trained deep convolutional spiking neural network retina-neuromorphic approach," in *Artificial Neural Networks and Machine Learning – ICANN 2019: Theoretical Neural Computation*, pp. 724–736, 2019.

[41] E. Stromatias, M. Soto, T. Serrano-Gotarredona, and B. Linares-Barranco, "An event-driven classifier for spiking neural networks fed with synthetic or dynamic vision sensor data," *Frontiers in Neuroscience*, vol. 11, 2017.

[42] B. Ramesh, S. Zhang, Z. W. Lee, Z. Gao, G. Orchard, and C. Xiang, "Long-term object tracking with a moving event camera," in *BMVC*, 2018.

[43] N. Sanket, C. Singh, C. M. Parameshwara, C. Fermüller, G. Croon, and

Y. Aloimonos, "EVPropNet: detecting drones by finding propellers for mid-air landing and following," 2021.
[44] "Camera DVS240, Specification Sheet." https://web.archive.org/web/20201223102302/https://inivation.com/wp-content/uploads/2020/04/DVS240.pdf. (accessed on 1 June 2022).
[45] "RC Benchmark Series 1580, Specification Sheet." https://cdn.rcbenchmark.com/landing_pages/Manuals/Series%201580%20Datasheet.pdf. (accessed on 1 June 2022).
[46] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN revisited, revisited: why and how you should (still) use DBSCAN," *ACM Trans. Database Syst.*, vol. 42, no. 3, 2017.

Part II

Literature Study

3

Swarms

Nature is the largest source of inspiration for robotic development. Most robotic practical applications involve interaction with the physical world, so it is reasonable to look for solutions to current problems on systems that have been perfecting themselves for millions of years. By imitating animal capabilities, new sensors were developed, like sonar for echolocation, new mechanical components were produced, like silicone suckers for object manipulation, and existing components were combined in novel ways, like stereo cameras for distance recognition. Robotic swarms are no different from these other developments. Millions of animals collaborate daily with the objective of achieving tasks they could not accomplish on their own. They leverage the use of many individual agents with limited capabilities to achieve a grander task with a better performance that could not be accomplished by a single party.

3.1. Biological swarms

Before diving into the technicalities of robotic swarms, it is worth carefully analysing biological swarms. Understanding how these multi-agent systems have evolved to exploit the benefits derived from their large number of individuals can result in inspiration for algorithms, subsystems or coordination strategies to be applied on the robotic counterpart.

Precisely speaking, the term 'swarm' refers to a large group of locally interacting individuals with common goals. It is used to describe all types of collective behaviours even though it typically brings up associations to joint movement in space [71]. The specific goal that animals collaborate for vary widely depending on the species. Sometimes, multiple goals benefits are obtained simultaneously.

Ants are social insects and benefit from living in large groups to cover the different roles necessary in their societies. They can collectively take care of their young, construct nests, build bridges and forage for food. Other social insects such as honey bees also show similar interdisciplinary behaviours. When the swarm involves non-social insects, the goals sought are often more specific, mainly four. The first one is mating. Female moths and cockroaches emit pheromones that trigger swarming and searching behaviour in males [137]. Exhibiting a similar behaviour, flies form swarms to promote the attraction of females and perform dancing courtship rituals, a technique called lek mating [17]. The second one is food foraging. An organised example is cockroaches, which leave pheromones for path trailing to discover food sources on non-competitive swarms. Locusts on the other hand explore altogether for suitable feeding spots following the wind direction [68]. The third one is migration. Insects like butterflies, dragonflies, beetles and moths group together in large number of individuals to perform seasonal migrations across huge distances. For such insects, these migration journeys are typically longer than their lifespan, and most of them do not make the entire round trip [112]. Lastly, swarming on insects can also be exhibited as a form of protection against predators. Marines insects have shown that selfish dilution in a big swarm reduces the attacks on the individual, increasing their chances of survival [50].

Beyond insects, plenty of other animal species also exhibit swarming behaviour for different purposes. While their goals may be common with those of the insects, they have a different way of executing them. In birds, for example, migrations are also commonly performed by swarming. Flying in formation, they can save energy by flying on the vortices created by the preceding individuals. Red knots and dunlins

(a) (b) (c)

Figure 3.1: Three basic missions on biological swarms: (a) Trigger behaviour on antelope herds [110]. (b) Path following on an ant swarm [143]. (c) Coordinated movement in a starling murmuration [47].

that fly in formation using this technique have shown to achieve speeds $5km/h$ faster than the singular counterpart [101]. For fish, similar hydrodynamic benefits can be derived when swimming in tight formations [105]. The aforementioned dilution in the swarm for increased safety is also one of the main added benefits for fish schools and mammal herds. However, these can also benefit from collective vigilance. With the increase in herd size, the vigilance required for each individual can diminish with little effect on the overall safety of the group. This way, each individual can focus more on finding food sources and mating [59].

Up until now, all the swarms presented were composed only of conspecific individuals. However, heterogeneous swarms also exist in nature. With the same objectives to drive their collaboration, animals from different species work together in large numbers to combine their unique abilities. It has been shown that mixed zebra-giraffe herds reduce predation risk by eavesdropping on mutual cues. Zebras benefit from the high vigilance levels of the giraffes to detect cues of animal presence [126]. For birds, mixed flocks present some advantages over conspecific. Combining different capabilities, they are more likely to find potential food sources [51], avoid already exploited locations [15] and drive insects out of hiding [152].

For each of the different swarming behaviours presented, the individuals need to have a set of specific abilities that allows them to take part in such collectives. These can range from simple distance-based reactions to complex social interactions. Animals use such abilities to enable the emergence of complex group behaviours from simple local interactions. These group behaviour have, as presented before, many different end goals: mating, migration, food access, safety... To achieve all these different end goals, each species follows a different approach. However, there are a series of behaviours that are commonly exhibited by many different species as means to achieve these end goals. Three main different behaviours can be identified, for which different techniques are employed depending on the species. Looking at these three behaviours and the techniques animals use to address them can be a source of inspiration to tackle the challenges that are faced when developing drone swarms.

First, biological swarms often require a trigger to indicate the start of a certain behaviour. Vision can be of extreme help for many animals. Mammal herds use visual cues of conspecifics as an indicator of predator presence, as shown in figure Figure 3.1(a). Similarly, birds use vision to detect sudden movements of the rest of the flock. Bees use sound to coordinate liftoff when moving to a new nest. This call for swarming is achieved by specific 'piping' sounds that excite the bees to a flight-ready temperature (35ºC) [128]. One third technique animals use to trigger swarming behaviour is direct physical contact. In certain conditions of extreme hunger, locusts start exhibiting cannibalistic behaviour and bite each other on the back. This fear of being bitten on the back triggers a chain reaction that evolves into a swarming search for food [14].

Secondly, either for finding a food source or to trigger path following, swarms are commonly formed to find a specific location or follow a certain direction or path. For that case, the most commonly employed technique is pheromone dropping. This indirect communication system allows insects to mark paths on the ground whenever a food source has been found for others to follow, as shown in figure Figure 3.1(b). Similarly, it is also very frequently used for mating calls, where females emit pheromones to promote searching behaviour in males and select from a larger selection of partners [137]. A different approach to path following is by visual cues. Scouting bees, when looking for a potential new nest location, make dances that describe the options discovered with a dance that indicates their location and quality. If the dance is convincing enough, the other bees will follow to explore the location.

Eventually, the swarm migrates to the preferred location (i.e. that with the most appealing dance) [22].

The third main mission type that swarms perform is coordinated movement. Swarms are most often observed on dynamic systems. Involving large groups of individuals that move close to each other, a coordinated movement must be achieved to ensure safe navigation. Insects like ants or locusts have slow reaction times and are resistant to collisions due to their low speed. Faster animals like flocking birds or swarming bats require more advanced techniques to interact with the neighbours, factor increased due to the 3-dimensional nature of their movement. Bird flock sizes vary widely depending on the species. Very notable are starling murmurations, where reports have shown over $750,000$ individuals flying coordinatedly [58], as seen in figure Figure 3.1(c). Obviously, each starling cannot keep track of the other $749,999$ birds in the murmuration. Rather, observing the closest 7 neighbours and reacting to their movements, regardless of the density of the group, each starling is able to avoid collisions ensuring a safe behaviour in the macroscopic level [31]. In the case of noise emitting animals, like bats for echolocation, flying in swarms causes interferences between the signals sent by the different individuals. When flying in dense groups, bats have shown to reduce their calling rates. With such collective awareness, each bat suffers from a lower obstacle location refresh rate, but benefits from a lower level of interferences due to a less cluttered network and maximised information throughput at the group level [1].

All these collective behaviours are worth considering when discussing the development of robotic swarms. Of special relevance are those involving the coordinated navigation of bird flocks, for the drones shall fly safely without collisions, but inspiration can be drawn from all the biological multi-agent systems described.

3.2. Drone swarms

When addressing the drone swarm development, it is best to start looking at its definition to understand what capabilities must the system have. A robotic swarm does not have the same properties as a biological swarm. Understanding these differences allows the developer to make requirements based on the desired mission objective and system performance, and think about the different subsystems that should be included to enable the swarming behaviour. Finally, inspiration can be drawn from the biological examples presented above to help design a better performing swarm for the task at hand.

A robotic swarm can be defined as:

Definition. *Swarm robotics is the study of how large number of relatively simple physically embodied agents can be designed such that a desired collective behaviour emerges from the local interactions among agents and between the agents and the environment [123].*

From the definition, it is derived that, just like on a biological swarm, the drones operate collaboratively to achieve a common goal. Drone swarms can be used for applications where the task requirements exceed the individual robot's limitations, or for applications where better performance is desired compared to a single drone mission.

When developing robotic swarms, an important consideration must be taken into account that reflects the limitations of robots. Robotic swarm development is constrained by a series of expected advantages that such systems shall have over single agents. Firstly, a swarm shall be robust to the loss or failure of individual robots. Swarms can handle the loss of individual robots or subsystems and still accomplish the desired goal. Secondly, a swarm shall be able to reconfigure itself to tackle flexible environments and tasks. Having a distributed system offers more possibilities for task handling, which gets reflected in the system's adaptability. Lastly, a swarm shall be scalable to different sizes depending on the task. Based on the needs of the system to achieve the goal, more robots can be added to improve performance [124].

There are many different applications for drone swarms. Surveillance and mapping, construction, load transport or infrastructure inspection are just some examples. Of course, based on the application use case, the swarm would need to have different capabilities. These capabilities should always reflect the three advantages swarms have over single robots. For a drone swarm to be robust, flexible and scalable, two requirements can be made on the system. Firstly, the swarm shall be completely decentralised. Taking inspiration from natural systems such as bird flocks or fish schools, it can be seen that cooperation can be achieved without the need for external infrastructure. None of the described biological systems presented relies on a central entity that coordinates the swarming behaviour. Only on social animals with shared responsibilities do some individuals trigger the collaborative behaviour of the whole group,

but these work just on a very high level. Moreover, centralised systems are not robust to failures in the coordinating agent, and limit the system scalability. A second requirement can be imposed to avoid direct communication between the agents. Having a communication-less system ensures the swarm is scalable regardless of the subsystems used. Moreover, a failure in the communication subsystem could be fatal for systems where the coordination strategy is dependent on it. Taking inspiration from the biological systems described, inter-agent communication was only observed to trigger the swarming behaviour. While it can also serve many other purposes, like a form of socialising on hyenas [43] and monkeys [130], communication is not frequently used as a swarm enabling technique. For example, when performing a coordinated attack, a pack of wolves performs a silent approach not to alert their prey [63]. Similarly, a bird flock does not rely on communication to avoid collisions [31]. These two requirements, decentralised and communication-less swarm, limit the coordination capabilities but ensure a cooperating system capable of employing an unbounded number of agents.

To the best of the author's knowledge, a system with decentralised and communication-less requirements has not yet been proposed for unknown area exploration for any kind of robot [3]. Due to the imposed complexity, this design phase must incorporate extensive research from previously proposed swarming methods as well as new ones from other applications or biological inspiration.

The design of a drone swarm is characterised by four main elements, that range from local to global impact. At the hardware level, the Micro Aerial Vehicle (MAV) design defines the basic capabilities of each drone. Then, the local ego-state estimation and control allow the individual drones to operate autonomously handling basic flight operations. At the third level, intra-swarm relative sensing and avoidance allow the different drones to detect each other and provides them with the ability to coordinate. Lastly, the swarm behaviour establishes the coordination rules followed to collectively achieve the desired goal [36]. The low-level design imposes constraints on the performance of the system, while the high-level design proposes requirements on the system capabilities. Still, an emphasis is placed on the two subsystems that allow individual drones to collaborate within a swarm. Relative sensing is the subsystem that allows drones to interact with each other, and the coordination strategy focuses this ability on the targeted problem.

The purpose of chapter 4 and chapter 5 analysis is to understand the requirements that the design of a relative sensing subsystem and a coordination strategy under the decentralised and communication-less conditions present on the individual drone design. However, before addressing the specific technical challenges for each subsystem, it is worth highlighting two points inspired by biological systems that should be considered to enable a better drone swarm design.

On the one hand, it is important to remark on the simplicity of biological systems. Just by looking at its closest 7 neighbours, starlings can group in murmurations with over $750,000$ individuals. Similarly, bats can fly in highly dense formations by simply decreasing the frequency of their echolocation calls. When having a system that is not reliant on complex calculations or strict communication routines, the scalability of the system can be almost unbounded. For robotic systems, simplicity is accompanied by lower error probabilities and higher efficiencies. Therefore, the drone swarm should be as simple as possible.

On the other hand, even if most of the past research on swarm exploration involved direct communication [3], it can be observed on biological systems that it is not necessary for coordination. When the different individuals know how their conspecifics behave, they can act accordingly and prepare in advance for their movements. This does not only apply to dynamic bird flocks, but also to complex social routines like hunting for wolves. These different systems are based on different levels of knowledge of the neighbour's behaviour. An increase in the determinism of the actions of an agent allows for an increase in the coordination level when communication is not employed. Therefore, to compensate for the communication-less requirement, the drone swarm should be as deterministic as possible.

Applying these two points into the specific subsystem design, it is possible to integrate the inspiration from biological systems in the drone swarm. While they do not provide direct inspiration, like the use of sonar or stereo cameras, they provide the guidelines to be followed to ensure the system can handle a large number of individuals in a cooperative manner.

4

Relative Sensing

As stated in the definition of a drone swarm in chapter 3, to have a swarm, the individual agents must be able to interact with other drones and the environment. This chapter investigates the different methods that can be employed to achieve this level of interaction and close the gap between many individual non-cooperating agents and a collaborative swarm.

4.1. Subsystem analysis

Drone swarms are characterised by a certain level of inter-drone interaction. Of course, the degree of interaction determines the capabilities of the system. A relative sensing subsystem that shares current and planned future full state information for all drones can achieve a higher degree of coordination than one that just provides knowledge of the instantaneous relative positions. Therefore, before diving into the subsystem design, it is important to understand the requirements of the relative sensing subsystem derived from the high-level objective of the swarm. These requirements can then be compared with the limitations of the different methods analysed to investigate the utility of each method.

Starting from the mission goal, exploration of an unknown environment with a swarm of drones, it is derived that we cannot make any assumptions on the environment, as no prior information is available about it. The essence of the task is the lack of knowledge about the environment. Besides the basic requirements that the drones shall be able to navigate through the environment, meaning that the drones are physically able to fly everywhere, and it shall have a limited extension, any further assumption would limit the application of the system. If the system was deployed in a truly unknown environment, like planetary cave exploration, reliance on assumptions could be fatal [74]. Therefore, it is derived that the system must be robust to different environments. This applies to both, the physical layout and composition of the environment, as well as the operating atmospheric conditions.

In this first requirement, the variety of possible scenarios is so large that it is absurd to set boundaries on every single variable. Setting hard requirements would focus safe operation only on some environments, prioritising them over others. Rather, for evaluation purposes, the different methods shall be ranked based on the operation range. For example, a method that relies on certain lighting conditions is evaluated worse than one that does not assume anything about such variable. It is then important to clearly describe the environmental requirements for each method to be able to compare them.

A second requirement is that the subsystem is semi-omnidirectional. Drones should be able to sense each other in all directions to prevent inter-drone collisions and increase the coordination of the swarm. To avoid inter-drone interferences such as downwash and facilitate relative localisation, the coordination strategy should enforce drones to fly at (approximately) the same height and not above each other. This means that the relative localisation subsystem shall detect drones with at least a vertical Lambertian coverage.

The third requirement involves the swarm architecture. It was already stated in chapter 3 that the system had to be completely decentralised. Therefore, so shall also be the relative sensing subsystem. This implies all the action is taking place between the drones, in an independent manner, not relying on external infrastructure or a master drone.

Lastly, a fourth requirement ensures operability. Drones are capable of very fast dynamics, which can be exploited for the benefit of the mission. On time-critical applications, it is desired that the drones explore an area as fast as they can. To prevent the relative sensing subsystem from limiting the operation speed, it is required that the subsystem is robust to fast dynamic systems, both in terms of detecting fast-moving drones and detecting other drones while moving fast.

The 4 requirements found for the relative sensing subsystem can be summarised in the list below. The relative sensing subsystem shall be:

1. Robust to different environments
    (a) Composition layout
    (b) Atmospheric conditions
2. Semi-omnidirectional
3. Decentralised
4. Robust to fast dynamic systems

Once the subsystem requirements are clear, the analysis of the different possibilities can begin. For relative sensing, many different methods have been developed using a variety of sensors. Each method has some advantages and disadvantages and is better suited for some applications than others. The most common types are here presented with some application examples. They include standard colour cameras, Infrared (IR), sound, communication-based ranging and landmark matching.

One last comment before analysing the different options. The difficulty of developing such a subsystem should not be undermined. One of the main challenges faced when developing a real physical system is the availability of a relative sensing system that meets all the requirements with the desired accuracy. Most studies on swarms never make it to the real phase and live forever in simulation. Therefore, special attention should be paid to the complexity of the system and the reliance on assumptions.

Vision

One of the most intuitive methods that are used for relative sensing adopts vision cameras. By observing its surroundings, a drone can be able to derive the relative localisation of the neighbouring drones just like humans are able to approximate the location of other people in a cluttered room.

The main challenge of this method is to distinguish, on an image, the drones from the background. Neural Networks (NN) and Deep Learning (DL) (black box models) have been used in the past for drone detection using semantic segmentation, but with limited success due to the large processing required [16], [129]. Another solution to this problem is the addition of stereo cameras, where the depth estimation of each pixel can be derived from the disparity between the images [146].

A third method commonly used to solve this issue is the addition of visual markers to the drone. A distinction can be made on the different visual aids used, depending on if the markers broadcast or not. Passive visual aids provide a distinctive pattern that can be easily identified on an image. Examples include visual markers [106], [55] and coloured balls [119], [46]. Active visual aids, on the other hand, emit light with a characteristic frequency, wavelength or pattern that is easily detectable with a camera. The most notable examples include the use of infrared [140] or ultra-violet light [147] or frequency-based blinking for ID detection [148]. Using the appropriate camera filters, such specific wavelengths are easily detected and segmented from the rest of the scene. Moreover, Walter et al. (2019) were also able to derive the inter-drone relative bearing from the camera reading by employing polarised markers on a predefined setup [148].

The accuracy of this method varies considerably depending on the technique employed (mono or stereo, active or passive markers, white box or black box model). The best performance of the methods here cited was found by using the stereo vision in [146], with a 5% distance error for distances up to $50m$.

While these different techniques may ease the detection of other drones, achieving a robust vision relative localisation system has many complications. As stated in the above-cited requirements, the system must be (semi-) omnidirectional, and robust to fast dynamic systems, environments, and atmospheric conditions. Using standard cameras, the limited Field of View (FOV), motion blur and low dynamic range are known limitations that constrain the robustness of the system. The use of a camera array or omnidirectional lenses can improve its directionality performance, but with a corresponding

addition in processing or decrease in resolution. High-speed cameras and lighting systems could be added to the system, but with a notable increase in weight.

Infrared

The second relative localisation method analysed uses IR. By emitting IR light, each drone sends out a signal to be detected by the other drones. Equipped also with IR receivers, the drone can detect other Unmanned Aerial Vehicle (UAV)s in the working range of the sensor.

One advantage of IR sensors is that the signal is very fast and require very little computational power, which can increase the relative localisation velocity. On the other side, they are uni-directional, so multiple sensors are needed for the drone to detect and be detected from different angles, increasing the weight of the system [88]. The accuracy of such a system has been proved to be sufficient for relative localisation, with a distance error of 8% on a range of up to $3m$ [70], [111].

The most common detection algorithm for IR relative localisation is Received Signal Strength Intensity (RSSI). Assuming the power of the emitters is known and equal for all drones, the strength of the received signal indicates the distance to the neighbouring drones. This technique however is susceptible to interference from other sources of IR such as warm bodies and light and must be tuned for a specific working range [141]. Recent efforts to fight this issue include Lock-In signal filtering, to only pass and amplify frequencies equal to the source, and the use of lenses to increase the range [88].

Sound

A third method that can be used for relative localisation is with sound sensors. By spinning their motors and pushing air down, drones generate disturbances that create noise while flying. Using a microphone array, it is possible to hear neighbouring UAVs and estimate their range and bearing.

An obvious challenge found when using this method is that the listening drone masks the other drone's sounds. Basiri et al. (2016) addressed this by using "chirp" sounds to identify specific drones and Cabrera-Ponce et al. (2020) used Convolutional Neural Network (CNN)s to distinguish between the different sources, but with limited gain in accuracy and increased weight and computational requirements [12], [30].

Communication-based ranging

Communication-based ranging leverages an existing communication infrastructure to obtain the relative localisation of neighbouring drones. Using the antennas available on almost every drone, it is possible to estimate the range and bearing by analysing the strength of the signal received.

When choosing the signal type, the two main options are ultrasound signals or radio signals, each with its advantages and disadvantages. Ultrasound signals are preferred for short ranges as it is easier to accurately measure the signal with less energy consumption. However, they are slow to dissipate in indoor environments, causing interferences [88]. Accuracy of such systems can achieve a $0.8cm$ distance error and a $3°$ bearing error over a $6.7m$ range [117].

Radio signals on the other hand are better suited for longer ranges and are not sensitive to miscalibration. Some examples of radio signals include WiFi, Bluetooth or Ultra Wide Band (UWB). The main differences between the three of them are power consumption, bandwidth and precision, where UWB consumes the least energy during the active mode and WiFi the most, WiFi has the largest bandwidth (up to 50 devices) and Bluetooth the least and UWB has the highest precision (cm resolution) and Bluetooth the least (m resolution) for ranges higher than $50m$ [48] [115] [36].

One drawback common to both signal types is the link between the relative localisation and the communication subsystem. If the communication protocol relies on hearing from all drones or sequential order, issues may arise as drones get out of range [81]. Similarly, the scalability of the system is threatened due to the dependence of the relative sensing on the communication protocol. For such, it was already specified in chapter 3 that the drones shall not use active communication.

Landmark matching

The last of the relative localisation methods analysed is landmark matching. This technique works if the drones can identify landmarks or features in the environment. The drones estimate the location of the surrounding landmarks and broadcast their position with respect to them. The rest of the drones are then able to triangulate the position of the broadcasting drone by finding its position in their landmark

map.

One of the drawbacks of this technique is that the challenge is shifted to relative localisation with respect to landmarks instead of drones. The complexity of this task depends on the implementation used. Achieving its simplest form, Montijano et al. (2016) allowed the drones to share mutually observable features and derive a local relative localisation [95]. On a more complex form, Thrun et al. (2005) enabled the drones to create a full map of the environment they then exchanged and updated for 'global' relative localisation [142].

4.2. Comparison of common methods

The goal of the comparison analysis between the different methods described is to understand their applicability based on the desired system requirements. A critical analysis is here presented to evaluate the suitability of each method according to the global goal.

The main benefit of vision-based relative sensing is the great source of information that such sensors provide. With one sensor, it is possible to extract relative localisation of the drones in the FOV as well as visual information about the environment. Visual information is extremely rich, as it is possible to derive optic flow [90], [120], [134], feature maps [122], [33], [54] or even full navigation paths [34], [35], [80] from it. Ideally, marker-less drones should be able to detect each other to avoid the addition of weight through markers or active components that also consume power. However current state of the art methods are not able to perform reliable relative sensing without the addition of such visual aids. The trade-off then needs to be performed between a rich source of information against the localisation problems and the limited FOV, motion blur and low dynamic range mentioned before.

On the other hand, IR and sound-based relative sensing just provide information specific to the location of other drones. Recent efforts by Boutin et al. (2020) in the field of sound have allowed the use of microphones to detect obstacles where the sound bounces off, but just for single drones [28]. Similarly, recent efforts have been made by Rosas-Flores et al. (2020) on obstacle detection exploiting surface IR reflectance [121]. For both cases, it is more complex to do so in a multi-drone environment because of the number of sources of sound or IR which would interfere with each other and has never been achieved before. In addition, the accuracy of such methods is lower than vision-based or communication-based ranging. On the other hand, they are robust to motion and environmental conditions (provided the system is tuned accordingly).

Communication-based ranging, as well as landmark matching, leverage an existing communication subsystem to perform relative sensing. However, such a system may not be necessary. As stated in chapter 3, the system shall not use active communication. Many efforts have been made to diverge away from reliance on communication for coordination to make the swarm more robust [55], [125], [3]. Reliance on communication can even determine the collaboration strategy limiting the exploration capabilities of the swarm [27]. Moreover, the scalability of the communication subsystem can be of concern, especially when it relies on a predefined sequential order. It is also true that a higher degree of coordination can be achieved when using communication, but since there are also drawbacks derived from their use, it is preferred that the relative sensing is kept independent. Landmark matching also gets disregarded within this category, since it relies not only on the communication subsystem but also on landmark extraction and precise mapping, which comes under heavy computational cost.

As it was shown, none of the commonly implemented methods fully satisfies the requirements presented. This raises concerns about the existence of a relative sensing method that shares all the advantages without the drawbacks. The vision method presents the clearest advantages, while the drawbacks are derived from the limitations of the physical sensor used. If a sensor would exist that does not suffer from the limited FOV, motion blur and low dynamic range, while still being able of providing accurate visual relative sensing and rich visual environment information, the inter-drone awareness problem for drone swarms could be solved.

4.3. Event cameras: a new sensor

The above-mentioned limitations of standard cameras have motivated many researchers to develop new sensors. In 1991, Misha Mahowald together with Prof. Carver Mead presented a new type of camera that worked in a radically different way compared to standard cameras, the event camera.

An event camera, also called dynamic vision sensor, neuromorphic camera or silicon retina, is a bio-inspired sensor that asynchronously measures changes in pixel brightness. Instead of capturing full

images like a conventional camera, where the colour of every pixel is recorded and stored under a frame, each pixel in an event camera works independently, only reporting an output when the local change in brightness exceeds a predefined threshold. The output of such sensors is therefore not a continuous stream of frames at a predefined rate, but rather an asynchronous stream of 'events' composed of the pixel location, time and sign of the change in brightness. Having a different working principle, event cameras show attractive properties compared to standard cameras. Namely, a high temporal resolution (in the order of μs), low power consumption, very high dynamic range (140 dB vs 60 dB for standard cameras) and high pixel bandwidth (in the order of kHz), which allows for a significant reduction in motion blur [52].

While event cameras have such nice properties, they only became commercially available in 2008 [82]. Therefore this sensor is only now starting to show its potential for a wide range of applications, mainly those where the limitations of the standard cameras was hindering the technical progress. Examples of typical scenarios where event cameras outperforms standard cameras include object tracking [41], [56], object recognition [107], [113], shape tracking [144], [4], surveillance and monitoring [83], [118], Simultaneous Localisation and Mapping (SLAM) [94], [145] and visual odometry [96], [97].

As it can be seen, despite their recent appearance, a lot of research has been placed into these sensors to get the most out of them. Especially in the field of robotic applications. While they have not been used before for relative sensing on drone swarms, research was done on robot detection and tracking as a motion capture system [109], [10].

Relative sensing with event cameras

To evaluate the usefulness of employing event cameras for relative sensing, it is worth evaluating the recent work on robot detection and tracking. That way, it is possible to understand if the assumptions held in such methods also apply to our use case.

There exists a wide range of algorithms that can be used for the detection of robots on neuromorphic sensors. These vary in complexity, accuracy and computational requirements. Algorithms of varying complexity are presented to provide an idea of the detection capabilities of this new sensor.

Early work on object detection started with continuous blob grouping of events, without memory [42], [84]. Using Gaussian correlation filters and previous knowledge on the approximate size, events were clustered to determine the location of different targets. Using this same principle, Schubert et al. (2017) were able to use predefined density grouping with Density Based Clustering of Applications with Noise (DBSCAN) to locate objects [127]. Other methods such as mean shift exploit the fact that events are triggered on the contour of objects, and are distant from the centroid. The advantage of such techniques is that no prior information is required on the number or shape of the features, but rather cluster centroids are found in an iterative manner [9]. All the above-cited algorithms can be used for direct object detection without further processing.

More complex algorithms can combine cluster detection with cluster tracking for higher accuracy. Particle filters are commonly used to update the location and shape or size of clusters while maintaining a low computational cost [57], [118]. More elaborate tracking can be achieved with Principal Component Analysis (PCA) used to estimate the direction and strength of variation of the events that belong to a certain cluster [93]. PCA operates using the spatio-temporal information (x, y, t), and the movement of the clusters can be directly extracted from the principal components.

Lastly, object detection algorithms can also use feature matching that uses hand-picked or data-learnt features. Hand-picked feature methods are most typically used on applications with very controlled environments, exploiting probabilistic filters. Events can then be matched against predefined shapes (object outlines) with algorithms such as Iterative Closest Point (ICP) [103] or gradient descent [104]. On the other hand, data-learnt models use training data to learn complex and abstract representations that can then be used to match and track objects [73]. Spiking Neural Network (SNN) are asynchronous NN that were designed specifically for neuromorphic sensors. Instead of a binary trigger that projects through the network, SNN work with exponentially decaying potential values. One limitation of SNN methods is the amount of training required, which should not be underestimated. However, due to the sparsity on events which just trigger on important information, computations on SNN are reduced, up to 200 times, compared to a convolutional NN counterpart [73].

The wide range of methods available and use cases of event cameras proves that they are suited for accurate object detection and tracking. If they were used for relative sensing on the drones, they would need to be mounted with an omnidirectional lens for omnidirectional coverage. Neighbouring

drones would generate some events when their relative velocity is different from zero, but mainly it is the propeller motion that could easily be detected for relative sensing. Compared to prior work on UAV detection with event cameras [64], [73] which was performed with high-resolution vertical recordings and static background greatly simplifying the task, the challenge of using event cameras for relative sensing lies in the segmentation of the neighbouring drones from the rest of the scene.

Mounted on board the moving drone, the background scene would generate many events as the UAV flies around the static environment. Special adjustments would need to be done to ensure the background-triggered events do not mask the drone-generated events. One option to cope with this issue is by classifying events based on their spatio-temporal resolution. An Attention Priority Map (APM) can capture regions that triggered more events within a certain time frame and assign them a higher priority [118]. By applying an APM to the camera output, the priorities can be used as a weighting factor passed to the object detection to improve detection, and as a filter to reduce the number of events to process. A second method to segment the background deals with asynchronous signal processing. It was already mentioned how the main source of events for neighbouring drone detection is the propeller motion. Using the same UAV model for the whole swarm, the propeller Revolutions per Minute (RPM) are the same for all drones. Since the events are triggered asynchronously, those corresponding to the propeller motion could be characterised by a certain specific frequency. A Lock-In filter, previously presented for IR signal detection, could serve as a mask for the background events [121].

One last aspect to consider regarding the use of event-based cameras for relative sensing is the frequent noise associated with such sensors [52]. Deterministic filters such as convolutions and activity filters provide a simple implementation to reduce noise events. More complex methods have also been used, but with the additional goal of reducing the overall number of events to process for highly dynamic applications. Less Data Same Information (LDSI) is an algorithm to achieve a high data reduction ratio while keeping the relevant information at the scene [11]. Representing the scene as a heat-map of events, the value of each pixel is accumulated over time with exponential decay. Only those pixels that exceed a certain threshold, the ones with higher spatio-temporal density, are passed through the filter output.

On the hardware side, some research has been conducted on the effect of temperature on event camera noise. The sensor was analysed at different illumination setups - 2 lux and 10 lux - with and without cooling. The optimal cooling temperature was obtained experimentally at 1.5 deg, on a trade-off between reduction in thermal noise and condensation due to low temperature. A clear improvement is demonstrated when the sensor is cooled. The Signal to Noise Ratio (SNR) is raised by a factor of 1.3, and the number of noise events is dropped by 30%, event at different illuminations. The event probability for a ranging contrast is notably higher for the cooled sensor. The benefit of using a cooled sensor is clear, with a higher SNR and less noisy events [20]. Adding a cooling system on the other hand increases the weight and power consumption of the drone, reducing the mission time. If placed carefully, the event camera could be in the turbulent flow generated by the propellers, using air cooling to increase the SNR.

Comparison with standard cameras

Before settling with event cameras for relative sensing, it is worth analysing their performance compared to standard cameras. By carefully comparing the limitations of the two sensors, it is possible to determine if indeed event cameras outperform standard cameras in all the required aspects and meet the relative sensing requirements.

Event-based sensors can be compared to frame-based sensors on the task of object detection and tracking through a series of experiments that focus on the different limitations of standard cameras, namely motion blur and low dynamic range since the limited FOV is shared for both sensors and can be solved utilising an omnidirectional sensor. Regarding the experimental results, special attention needs to be placed on the analysis, since the algorithms selected also play a big role in the system performance.

Previous research conducted on detection and tracking comparison for both sensors on different conditions includes detection of UAV on static scenes [26]. The tests performed focused on evaluating the dynamic range performance of the sensors and used two different algorithms for the neuromorphic sensors: asynchronous detection and pseudo-frames generated by the accumulation of events. In the daytime, the standard camera offered a similar detection accuracy performance for drone tracking than the asynchronous counterpart. Some birds flew in the background and all the algorithms except the pseudo-frames were also able to identify them as separate objects, although with some data association problems due to detection breaks. During nighttime flying, the frame-based camera was not able to detect and track the drone, while the event-based camera was not affected by ambient light and

accurately detected all drones. It is therefore concluded that event-based cameras can effectively be used for object detection at very high dynamic ranges. This is an extremely important point, as no assumptions can be made regarding the lighting on the unknown environment to be explored by the drone swarm. Therefore, increasing the dynamic range of operation ensures higher operability of the system developed.

Regarding algorithm performance, due to the specific algorithm selection, frame-based had considerably larger memory requirements and shorter processing times than the algorithm for the event camera. However, the object detector was not specifically designed for the task nor optimised in any way. With a more optimised algorithm design, the benefits of the physical sensor can be exploited employing asynchronous object detection.

A second set of experiments focused on object detection at very high speeds. A rotating disk is to be tracked with event cameras and global shutter cameras [62]. Note that standard cameras are not used for this experiment because they are not able to track such fast objects. Instead, a global shutter camera with a frame rate of 1000 FPS is employed to detect the markers in the rotating disk. Two different event cameras are compared: the iniVation DVS240 [44] and the ATIS HVGA Gen3 [45]. The experiment setup alone already shows that the performance of the event cameras is far superior for moving obstacle detection than standard cameras since these are not even considered. While the absolute speeds of the markers remained low, between 0.27 and 21 m/s, due to the small separation between the target and the camera, the mean image speed varied between 3.3 and 260 kpx/s. For all experiments, varying the lighting conditions showed little to no influence. Accurate detection of the marker was achieved on the DVS240 for image speeds up to 50 kpx/s. Beyond this speed, long tails of positive events start to form, marking the limit of the sensor. The ATIS HVGA Gen3, with double the bandwidth, could detect accurate motion for image speeds up to 200 kpx/s. In any case, the expected image speeds to be encountered for relative sensing are well below the 50 kpx/s threshold, and therefore it can be determined that both sensors are suited for the relative sensing application.

Due to the strong dependence of the algorithm selection on the subsystem performance, it is not possible to have an estimate of the accuracy expected when using event cameras for relative sensing. However, the trend in the literature is clear. Sensor capabilities are higher for the event-based (faster dynamics and wider range of ambient conditions), and it boils down to the algorithm to be able to exploit the faster response to derive faster conclusions about object placement. Careful choice of the algorithm would need to be performed to achieve very fast object detection and tracking, and with low processing power.

Algorithm design and evaluation

Considering the application scenario, limited computational power and performance requirements, it is possible to draft from all the literature presented an outline of how the algorithm implemented could look like. Note that this implementation is subjected to the actual performance shown by the event camera plus omnidirectional lens combination. The whole detection and tracking algorithm could be split up into three parts. The first step could consist of an event filter. Depending on the scenario, either a simple noise reduction deterministic filter or the more elaborate LDSI algorithm could be used based on the processing and filtering required. Moreover, the signal could be further processed with an APM or a Lock-In filter to facilitate drone detection. A second step could use the filtered output to perform drone detection. Depending on the filtering applied, a simple DBSCAN clustering algorithm could suffice, or a feature matching algorithm could be developed that then uses ICP for detection. Data-learnt detection takes a considerable amount of time to train, and shall not be used unless strictly required. Lastly, to have a robust relative sensing algorithm, a third object tracking step can be present. A simple particle filter could allow for robust performance against noisy detections and occlusions. This first algorithm idea shall be tested for compliance with the requirements. If tests conclude that the accuracy is not sufficient, more elaborate algorithms would need to be developed and implemented.

One interesting aspect to keep in mind for the evaluation of the system performance is an object tracking metric. With desired minimum performance requirements, it could be used to assess the need of achieving a higher accuracy or accepting the current levels. The Multiple Object Tracking Accuracy (MOTA) metric proposed trades-off missed detections, false positives and mismatches to indicate the relative sensing performance [19]. MOTA is defined as

$$MOTA = 1 - \frac{\sum_k M_k + FP_k + mme_k}{\sum_k g_k} \tag{4.1}$$

where M_k, FP_k and mme_k represent the number of missed sequences, false positives and mismatches in frame k, respectively, and g_k represents the number of ground truth objects in frame k. Individual drone detection metrics can be composed of a match in the number of clusters and the ratio of the detected cluster area. Therefore, detections can be marked as correct when both of these values exceed a certain threshold [109].

One last comment is worth discussing regarding the possibilities that using event cameras as relative sensing sensors open. Some research has been conducted on obstacle detection with neuromorphic vision sensors on autonomous robots, that could be used in the subsystem being developed. Using optic flow directly derived from the camera output, it is possible to construct a nearness map of the environment [21], [18]. Such information could then also be used to guide the navigation and avoid collision with the environment [93].

5

Coordination Strategies

The second characteristic that defines a robot swarm is the coordination between the agents to achieve a desired collective behaviour. A network of drones that can sense each other, but does not have any coordination or any interaction between the agents cannot be defined as a swarm. The interactions required for the achievement of the desired goal can be defined as the coordination strategy, and vary depending on the task to be solved.

5.1. Task analysis

In the context of unknown area exploration, the task to be solved is the complete exploration of a given environment, commonly performed to find an object or person of interest or generate a map of the scene. Search and rescue operations to find a missing person on an unknown area [7] or generating maps of planetary caves [74] are just two examples of situations that would require inter drone coordination when performed with a swarm. As it was done for the relative sensing subsystem, an analysis of the requirements that shall be met by the coordination strategy is first presented. These requirements can then be used to evaluate the utility of the different strategies proposed.

The first requirement is derived from the task to be solved. It was mentioned in chapter 3 how the swarm was meant to explore an unknown area to find the location of different targets. Because the targets are at unknown locations, the whole environment shall be explored to ensure all the targets are identified. It can be then derived that the coordination strategy followed shall allow for complete coverage of the environment. It is the objective of the coordination strategy to ensure at least a drone has flown through all the areas of the environment. Only when the environment is fully explored, the system can accurately determine the location of all the targets.

An important annotation has to be made. It is the responsibility of the coordination strategy to ensure drones explore all areas of the environment such that the targets can be localised. However, the required thoroughness of the exploration is a function of the target detection subsystem. If a target detection subsystem has a range of 10 cm, then it would be required that drones fly with such spacing. Because event cameras are used for relative localisation, they can provide information about the whole scene in the FOV. Such images could be used for online or offline target identification, provided the target is visually recognisable. Then, the level of exploration required shall at least allow for capturing the whole environment with the event cameras.

A second requirement is that the system shall be aware of the moment when it has completed the exploration task. If the swarm is not aware of the task completion, the drones would continuously explore the environment until they run out of battery. If they are triggered to return home before that happens, the system would also not know whether the drones should fly again to continue the exploration. For the system to be practical, it should be aware of when it has successfully explored the whole environment and the task is finished.

The third requirement is common to the relative sensing subsystem. Having set an early requirement on the system to be completely decentralised, so shall be all of the subsystems used. A requirement can be derived that the coordination strategy shall be decentralised, and not use an external infrastructure of a master drone to spread the drones around the environment.

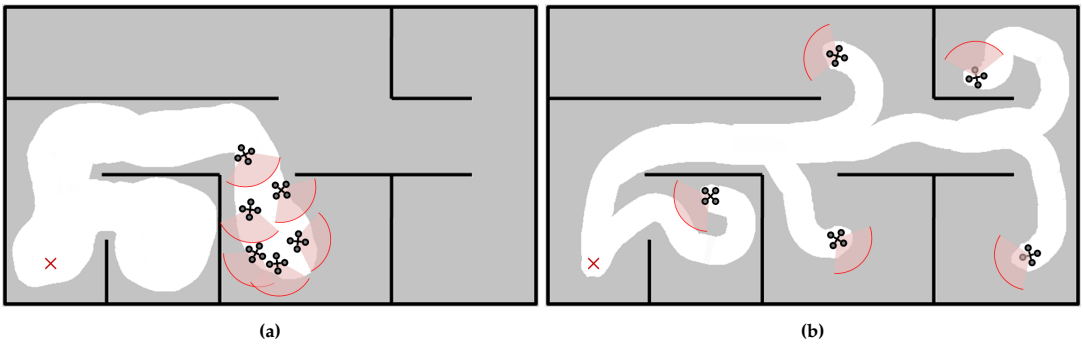


Figure 5.1: Different exploration philosophies for a swarm of drones. The red cross marks the initial position of the drones, and the white trails indicates the discovered area. (a) Flocking exploration shows higher redundancy. (b) Independent exploration promotes faster area coverage.

One last requirement can ease the selection of the algorithm based on the metric performance. Many of the applications that require swarms, such as search and rescue [99], load transport and delivery [108] or reconnaissance [65], require the task to be performed in a timely manner. The fourth requirement is therefore that the coordination strategy shall allow for fast exploration. To help the evaluation of this requirement, it is broken down into two complementary points. Firstly, the coordination strategy shall avoid navigation over already explored areas. Secondly, the coordination strategy shall constantly promote the exploration of new areas.

The different requirements the coordination shall meet to ensure task success can be summarised in the list below. The coordination strategy shall allow:

1. Complete exploration of the environment
2. Awareness of the task completion
3. Decentralised coordination
4. Fast exploration
   (a) Avoid re-exploration
   (b) Promote navigation on new areas

Having stated the requirements to be met, the study of the different possible coordination strategies can be tackled. Two philosophies can be followed for multi-drone exploration coordination. The first philosophy exploits the robustness induced by the swarm by performing flocking exploration. When all the drones explore together by flocking, the system becomes robust to drone failure as the task can still be completed by the rest of the flock. The second coordination philosophy promotes independent exploration of the different agents. By exploring different areas of the environment simultaneously, exploration time is reduced. However, the coordination between the drones requires a higher level of understanding of the environment and the exploration performed by the other drones. Both of these philosophies are equally valid and have their advantages and disadvantages. Each one is presented in detail, addressing the different aspects to consider for their implementation. An example of a flocking exploration swarm can be found in figure Figure 5.1(a). An example of an independent exploration swarm can be found in figure Figure 5.1(b).

Before proceeding with the discussion of the different philosophies, an important aspect must be presented. Both of the coordination philosophies require an exploration strategy. On the flocking exploration, the whole flock explores the environment together following a certain strategy. At the same time, on the independent exploration, each drone follows its own exploration strategy while remaining in coordination with the rest of the swarm. It is therefore beneficial to learn first about exploration strategies for a single drone, and only then extrapolate the acquired knowledge to the multi-drone problem.

5.2. Exploration strategy

Exploration of an unknown area by a single agent is a problem that has been widely studied [2], [35], [89], [86], [138]. A wide variety of methods have been proposed, with varying levels of understanding of

the environment. A classification can be made based on this metric, with some methods presenting a behavioural strategy and others showing a hierarchical strategy [100].

Behavioural strategy

Behavioural control, also known as reactive control, derives navigation commands directly from the current sensory information. The name is derived from the reactive nature of the control commands to the sensory data. In this bottom-up approach, high-level constraints are not considered when planning the robot's actions. Therefore, a low understanding of the environment is sufficient to successfully control the robot. One of the benefits of this strategy is its adaptability to dynamic and unknown environments. Since it is not necessary to build a model of the environment, the strategy success is independent of the environment's complexity and can quickly react to dynamic situations [100].

Typically, multiple behaviours or control commands are generated simultaneously from the different sensory data. That way, complex behaviours can emerge from the combination of simpler and smaller sub-level reactions. Different architectures exist to combine the different control laws generated. Subsumption behaviour-based control sorts the different control laws based on priority and activates the behaviours with the highest priorities [29]. Huang et al. (2019) developed a dynamic priority allocation together with a Null-Space-based Behavioural Control (NSBC) to project low priority actions into the null space of the robot actuation [66]. Motor behaviour-based control on the other hand simultaneously combines the output from the different control laws. The overall response is obtained from the vector summation of the different behaviours allowing a more cooperative, less hierarchical control [6].

Behavioural exploration of unknown environments can take different forms, that depend on the level of complexity integrated into the strategy. On its simplest form, random walk or random navigation promotes the agents to follow paths selected at random. The exploration strategy is therefore abstracted from the sensory input. At the extreme, the whole navigation strategy can be independent of the sensory input making the drones fly blindly, but collisions with other drones and objects can occur. Mulgaonkar et al. (2018) effectively used a swarm of drones for exploration with this technique, employing drones resistant to such collisions [98]. More reactive methods can also employ random walk for environment exploration and combine it with sensory input for collision avoidance or inter-robot coordination [125]. The main limitation of random walk algorithms is the large amount of time taken for exploration, as complete area coverage can only be guaranteed asymptotically with time.

Other efforts have been made to guarantee a quicker solution. Directly guiding the navigation from the sensory input, bug algorithms have shown to be effective for exploration tasks. Bug algorithms are a navigation strategy that leverages relative sensing of obstacles in the environment to progress with the exploration by reacting to contact with such obstacles. There are many different types of bug algorithms with varying complexity, memory and sensing requirements. Simple wall following algorithms can be efficient for maze solving tasks, but it requires conscious navigation to achieve effective unknown area exploration. Loop detection and path integration are proposed as two techniques to avoid navigation over already explored areas and perform accurate homing [91].

More complex behaviours can be achieved with increased sensing. A Finite State Machine (FSM) can allow reactive behaviour when the rules to change between the states are sensory-based. Then, with predefined behaviours, it is possible to achieve a high level of control with mere sensory information that triggers the state transitions. Bug algorithms are a type FSM, where the reactions take place only at the low level. When more complex predefined behaviours are employed, the tasks to be solved become virtually unlimited, including area exploration [150].

The last of the behavioural exploration strategies employs fuzzy control for robot navigation. A set of fuzzy rules are designed to achieve the desired goal: area exploration. Common examples of fuzzy rules include obstacle avoidance [92], goal-reaching/exploration [102], or formation control for multi-agent systems [66]. The sensory information is employed to determine the membership to the different rules, which then project their output onto the actuation space of the robot. For fuzzy control, the combination of the output signals can be done by fuzzy mechanisms of product-sum or min-max inferences [139].

Hierarchical strategy

On the other side of the spectrum of robot control, a hierarchical strategy uses the sensory information perceived to generate a global world model upon which to generate the appropriate robot actions to achieve the desired goal. That way, a hierarchical strategy, also called deliberative or centralised, follows

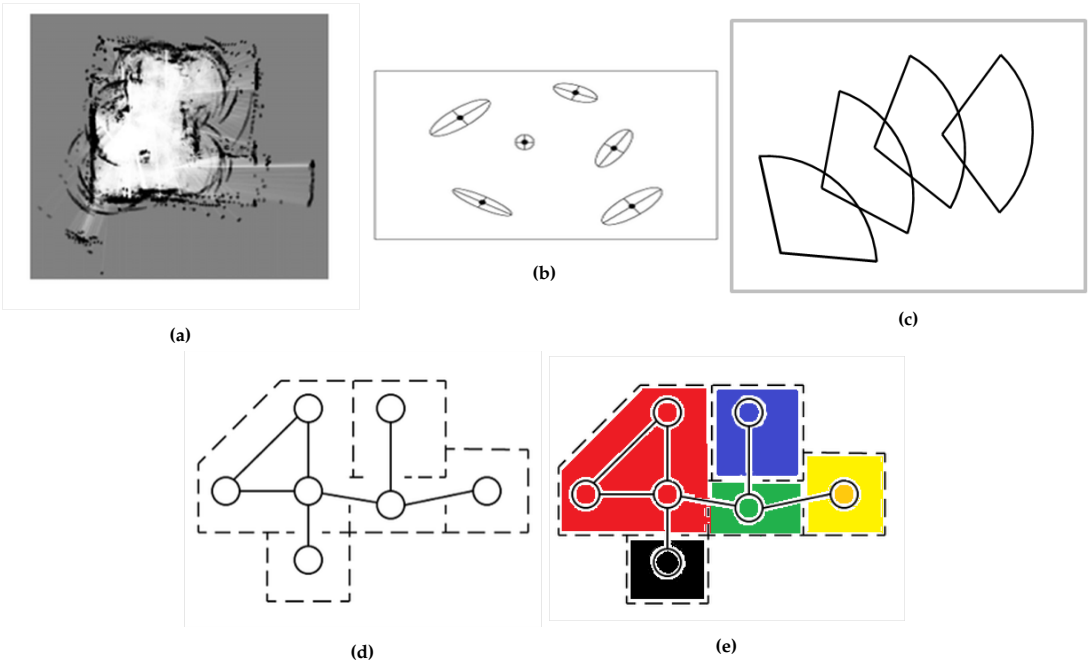


Figure 5.2: Different map representations for hierarchical control [5]. (a) Grid representation is an occupancy grid of the area. (b) Feature representation is a relative location of detectable features. (c) Appearance representation is a concatenation of images. (d) Topological representation is a division of the environment in distinct areas with proximity relations. (e) Semantic maps are like topological maps, but with more detailed information about the places and their composition.

a top-down approach where high-level constraints are broken down into low-level commands. One of the main benefits of such a strategy is the ease to turn goals into tasks. Using the environment model, the control strategy is directly derived from the current and the desired state [100].

A hierarchical strategy is usually comprised of three modules: sensing, planning and action. First, the sensing module employs the sensory information collected by the robot to generate a world model. This world model does not necessarily need to be a map, but rather it can be any abstract description of the world. Then, the planning module uses the world model to search for the optimal path to get from the current state to the desired state. The desired state is selected based on the task to be solved. For exploration, for example, the desired states are all those locations where the drone has not been before. Finally, the action module executes the planned actions to reach the target. Of course, this process is iterative. As the drones navigate the environment, they gather more sensory data which is used to update their world model, around which new optimal paths are calculated and executed.

Even if maps are not the only representation type a drone can have of the environment, they are the most commonly used. For that reason, they are the main focus of this hierarchical strategy study. There has been plenty of research on map-based navigation, and while it is all based on the same principle, there are many different techniques to achieve it [67], [69], [85], [135], [149].

Therefore, before diving into the different map-based exploration strategies, it is worth learning about how the different map-making methods, their strengths and limitations, for there is a strong correlation between the two. As mentioned above, hierarchical exploration is an iterative strategy that senses, models, plans, acts and repeats. Learning about the different techniques to model the environment provides a better understanding of the rest of the modules.

Regarding environment modelling, the most fundamental choice is the map representation type. There exist many different representations, but they can be grouped into six different categories: grid maps, feature maps, appearance maps, topological maps, semantic maps and hybrid maps [122]. This classification is made according to the abstraction and memory required to construct and update the map. As the map types increase in their level of abstraction, there is a consequent decrease in geometric information.

At the lowest level of abstraction, grid maps represent the world with a matrix of cells. Each cell represents a small area of the world. The occupancy of each cell is derived from sensory information.

Range finders are most commonly used, but other sensors to develop such maps, like vision or IR, can also be employed. Based on the map occupancy, the robot can plan the navigation through the free cells of the map, only achieving a low-level of abstraction and comprehension of the scene. Smaller cell size allows for a more accurate model of the environment but at the expense of higher memory demand. Coarser map representations on the other hand still allow for safe navigation at a lower computational cost. Similarly, the grid representation can be 3D, where the cells become voxels instead of pixels, but with the corresponding increase in computational and memory expense [133]. An example of a grid map can be seen in figure Figure 5.2(a).

With a slightly higher comprehension of the physical environment, feature maps or landmark maps represent the world by the features that compose it. This type of mapping does not allow for direct planning to avoid collisions since only partial information about the (feature) occupancy is available, but it is suited for accurate localisation. The feature selection is of high importance, as it determines the robustness of the map. Outstanding features will be more easily detectable when returning to the same location, allowing for an accurate position estimate. On the same line, denser feature maps may ease localisation but at the expense of memory and computational power required [61]. An example of a feature map can be seen in figure Figure 5.2(b).

At a similar level of abstraction, appearance-based maps associate views with different locations. Instead of using relative localisation based on the observable features like in feature maps, the location estimate is provided based on the similarity of the images. A vision system is therefore clearly required for such type of maps. Once again, these maps are more valuable for localisation than for obstacle avoidance planning. The density of the images recorded influences the localisation accuracy, as well as the memory and computational requirements [23]. An example of an appearance map can be seen in figure Figure 5.2(c).

Making a leap in the level of understanding of the environment, topological maps offer a higher-level abstraction. Until now, map descriptions merely consisted of a projection of the real world into the virtual model, either by using simple binary occupancy grids, location of features, or viewpoints. Topological maps allow for an abstract representation of the environment composed of paths and intersections. Topological maps are often defined as symbolic mapping, providing a concise model in the form of a graph. Consequently, the memory and computational demand are much lower compared to the metric maps presented above. In topological maps, there is no need for precise localisation, since there is an absence of metric and geometric information which is in turn replaced by notions of proximity and order. Using such a high-level representation of the environment requires, at the same time, that the drone can navigate between the different topological regions coping with obstacles and other unforeseen situations [25]. An example of a topological map can be seen in figure Figure 5.2(d).

Semantic maps depict the most abstract environment representation. While being very close to topological maps, semantic maps offer more detailed information about objects and places. They prove to be extremely useful for goal-oriented and high-level behaviours that require reasoning about the environment. Semantic maps are especially useful when the robot has to interact with the environment and perform real-time reasoning on the scene. Needing to extract more complex features of the environment, semantic maps have a higher computational and memory demand than topological maps [37]. An example of a semantic map can be seen in figure Figure 5.2(e).

Lastly, hybrid maps encompass all the different combinations of the above-cited types. Combining features from different representations, it is possible to tailor the map to the specific application, sensors and computational power available achieving maximum results. Most common is the combination of a low and a high-level representation, such as hybrid topological-metric maps. By combining different types, a higher-level abstraction can be exploited for planning while achieving accurate relative localisation, for example [24].

For the task at hand, exploration of an unknown environment, topological maps would offer the best trade-off between a concise understanding of the environment at a low computational cost. Chaplot et al. (2020) used semantic maps for destination search in an unknown environment, where prior information on scene structure is leveraged to achieve a faster completion [34]. For exploration, however, all places are equally important and semantic priors are less relevant. Metric representations such as grid, feature or appearance maps have also proved to be effective for navigation but provided the required computational power is available. Biological navigation strategies show a parallel with topological SLAM, storing only important landmark information [54]. Not requiring precise localisation, navigation around the topological map can be performed with approximate notions. It is then derived that

topological maps offer sufficient environment awareness to allow for efficient exploration. Topological maps have been generated with a wide variety of sensors [25], but, to the best of the author's knowledge, never before with event cameras. It must therefore be addressed if it is possible to apply a topological map-making technique from standard cameras to a pseudo-frame event representation, for example. Otherwise, the alternative of adding a specific sensor for such a purpose could also be considered.

Having studied the different types of environment representations, their strength and limitations, it is possible to investigate how are they applied to the hierarchical strategy for unknown area exploration. Map-based exploration, or frontier-based exploration, is performed by constructing a map of the explored environment while keeping track of the areas to be explored (i.e. frontiers). As it was explained before, the map does not need to be an accurate metric map, it can be any representation that holds information about the areas visited and to visit.

Hierarchical single robot exploration is defined by the strategy followed by the planning module to exploit the information from the world model and determine the actions of the robot. Assuming the drone can generate the selected environment representation and frontier determination, the exploration strategy can be boiled down to the frontier assignment problem. For the multi-robot problem, the hierarchical approach must also include the other robots of the swarm in the model of the world. On flocking exploration, as all the drones fly in formation behaving like one individual, there is vast parallelism with single robot exploration. For independent exploration, on the other hand, the frontier assignment problem is a more complex challenge. On a communication-less system, all coordination must be emergent from the control rules based on the actions of the neighbouring drones. On top of that, the drones shall update their visited and to visit areas based, not only on their actions but also on that of the rest of the drones.

A distinction can be made between the different frontier assignment strategies available for multi-robot hierarchical exploration. Depending on the nature of the control commands, the strategies can be discrete or continuous. Discrete hierarchical control distributes the drones around the frontiers as individual missions using a set of predefined, low or high-level, rules until the assigned frontier is fully explored or until more frontiers are discovered. At that moment, a new frontier assignment would be proposed. This process is repeated until the exploration is completed. Continuous hierarchical control on the other hand governs the movement of each drone through some cost or goal functions. Navigation by flocking only makes sense when continuous control is used, at least partially for the formation control. However, more complex coordination strategies can be hard to integrate into a single objective function, and therefore are easier to implement on a discrete control strategy. Lastly, discrete control tends to be less computationally expensive.

Discrete exploration can be performed with different levels of complexity. A greedy frontier assignment is a simple and easily implementable strategy where each robot greedily chooses the best frontier according to a cost function, usually representing travel time or distance [76]. An example of a greedy assignment strategy can be found in figure Figure 5.3(a). To avoid different robots exploring the same goal, assigned frontiers can be discarded with a Broadcast of Local Eligibility (BLE). This method has been frequently implemented in a centralised manner, but can also be used on a communication-less system if the active broadcast gets replaced by passive observations of the neighbouring drone's movements [151]. An example of a BLE frontier assignment strategy can be found in figure Figure 5.3(b).

More sophisticated frontier assignment strategies can include algorithms such as the Hungarian method to optimally assign frontiers to drones [75], but auction algorithms that work on local updates prove to offer better performance when all the frontiers are not known initially (i.e. they are discovered) [13]. K-means clustering algorithms solve the problem of the unknown frontiers by quickly spreading the robots in the environment to sparse areas. This is achieved by clustering the remaining unknown space among the different robots. Initial distribution of the robots into distinct areas of the environment reduces the amount of re-exploration since robots are far from each other [131]. These methods however typically require either the presence of a centralised system for spatial distribution and assignment or a communication method for the auction approach. Such planning with a decentralised communication-less system is more difficult to achieve and requires predefined behavioural rules to coordinate the drones. An example of a K-means clustering frontier assignment can be found in figure Figure 5.3(c).

One last method that uses the discrete strategy is graph exploration. On graph exploration, the environment is modelled as a graph where the frontiers are nodes and the edges are the paths that connect them. Starting from the root node, the objective is to navigate the whole graph. For single

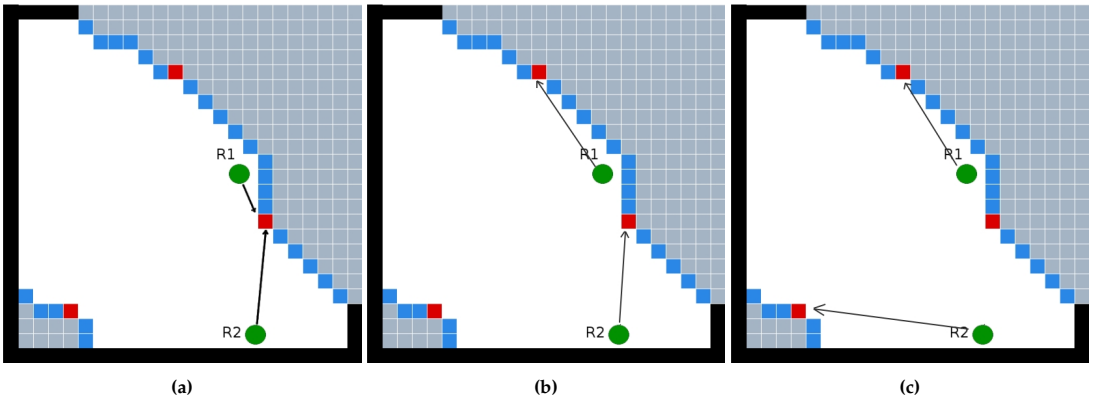


Figure 5.3: Discrete frontier assignment strategies [76]. (a) Greedy assignment leads to potential exploration overlap. (b) BLE assignment prevents overlap while keeping a greedy policy. (c) K-means clustering distributes the agents to different areas of the environment, afar from each other.

robot, two very popular approaches are Depth First Search (DFS) and Breadth First Search (BFS). In DFS, the most recently discovered frontiers are explored first allowing the drone to dive deeper into the exploration faster. On the other hand, BFS promotes exploration of old frontiers first, causing plenty of navigation over already explored paths. For multi-robot exploration, tree strategies have shown to be extremely successful, however they tend to rely on a centralised system [39], direct communication [87] or dropping beacons [32] to avoid re-exploration. Decentralised systems have been employed following this idea by using Multi-Agent Monte-Carlo Tree Search (MA-MCTS), but the coordination is dependent on the knowledge of the whole map which does not apply [40]. Achieving coordination for the graph exploration strategy on a decentralised and communication-less system requires a deterministic predefined behaviour that can be reproduced by all agents. Observations of the locations of the neighbouring drones can then provide information on the parts of the graph that are already explored and those that are not. However, such a method has not yet been proposed for unknown area exploration, to the best of the author's knowledge.

Continuous exploration strategies, on the other hand, do not use predefined assignment rules but rely on cost or goal functions to drive the navigation. Maybe the most notable hierarchical continuous planning strategy for robotic swarms is Particle Swarm Optimisation (PSO). PSO is commonly used for finding optimum locations on a given space [8], [136], and combines the readings from the different agents to advance towards the sensory global maximum. The navigation commands are not rule-based, but rather are dynamically obtained from the environment cost functions. Glowworm Swarm Optimisation (GSO) and Aggregations of Foraging Swarm (AFS) are two examples of more distributed algorithms where an adaptive decision domain evaluates the readings from the neighbouring drones to derive the navigation commands [38]. Even if all these methods are used for finding a specific maximum, exploration can be achieved by iteratively increasing the weight of the frontiers and decreasing the value of the to explore places on the map. Some issues with such techniques are that they rely on random deployment, which cannot be met on unknown area exploration, and they depend on a communication system to update the values of the cost function throughout the map.

A similar approach uses potential fields combined with a diffusive behaviour. A potential function, representing the attraction-repulsion rules that biological swarms display, determines the clustering of the drones in the swarm. High attraction forces allow the drones to fly on a flock while high repulsion forces ensure independent exploration. This diffusive strategy on the other hand promotes navigation towards frontiers. In large environments, subgroups tend to be formed even while flocking, due to the diffusive nature of the exploration algorithm. The combination of both signals is a continuous output that is passed to the drone's controls [72].

Having analysed the different exploration strategies, both behavioural and hierarchical, it is time to return to the multi-drone exploration coordination philosophies. Studying the different implementation details for each of them serves as an analysis of the feasibility and utility of each of the two methods. It is important to recall the imposed decentralised and communication-less requirements, to understand the constraints derived on the coordination strategy.

5.3. Flocking exploration

Taking inspiration from nature, flocking navigation offers each of the individual agents the benefits of navigating on a big group. Increased safety, higher awareness, or easier access to food sources are some examples of benefits biological systems can enjoy while when remaining in the flock. Analysing the behaviour, flocking can be broken down into a list of steering rules. Following a trajectory in an organised manner, the different agents are subjected to separation, alignment and cohesion rules. Separation secures a safe distance from the other drones to avoid collisions. Alignment guarantees all the drones move in an organised manner. Cohesion ensures the flock remains united [116]. More behaviours can be added on top to achieve certain goals and safe navigation around objects, but these three are the pillars that characterise the flocking philosophy. Swarming by flocking can be of special interest when it is desired that all agents remain in range for a certain purpose. For example, for coordination strategies relying on continuous direct communication, flocking can be an enabling mechanism [27]. Moreover, flocking can also be beneficial when deriving high-level goals from low-level interactions. For example, it is possible to perform reliable obstacle avoidance by collaboratively combining the behaviours from many individual drones [78].

There are two main questions to solve when dealing with the implementation of a flocking exploration strategy. The first one involves the preservation of the flock itself. Different mechanisms can be employed to ensure the flock navigates as a single unit and does not diverge into different subgroups. While flocking, such a thing must not occur to exploit the joint exploration in larger groups. The second one involves the exploration leadership. When navigating as a flock, coordination between the formation control and the exploration strategy must take place to ensure both goals are met. Such a technique ensures the mission objective is satisfied promptly without relying on specific formation patterns.

Starting with flock preservation, the different methods encompass pure attraction-repulsion, as well as cost function-based approaches. Attraction-repulsion methods exploit the three basic separation, alignment and cohesion rules as a manner of deriving the control commands that maintain the flocking formation. Potential fields can be employed to integrate the attraction-repulsion forces and maintain a desired relative position. The generalised Lennard-Jones potential function, for example, directly outputs motion vectors that can be translated to control commands from the inter-agent desired distance [72].

Fuzzy control can also be applied as an attraction-repulsion method. When the attraction-repulsion forces are expressed as fuzzy rules, the control commands can be derived from the fuzzy output mechanisms [66]. One benefit of this method is the ease to prioritise different conducts. Splitting the desired behaviours into separate rules with different membership functions, it is possible to completely control the swarm to maintain the flock as desired.

Cost function approaches to flocking control involve the development of a metric to evaluate the state of the formation. This cost function integrates the three flocking rules, separation, alignment and cohesion, to evaluate how are the drones in the swarm are preserving the flock. The path to be followed by the drone is then obtained by maximising the cost function value. Because the cost function is evaluated based on the position of all drones, it is possible to guarantee the preservation of the flock. Model Predictive Controller (MPC) and Linear Quadratic Regulator (LQR) are two examples of optimisation methods that can be used to plan the trajectories of the drones to preserve formation [55], [132]. The main advantage of this method, the increase in the level of formation control at all times, comes at the expense of requiring a centralised system that optimises all the trajectories simultaneously.

The second aspect to consider when implementing a flocking swarm is the exploration leadership. The three basic flocking rules are sufficient for formation control but do not push the swarm to explore the environment. As important as formation control may be, it is just designed as a means to achieve the mission objective, exploration of an unknown environment to find targets. There are two different methods by which exploration can be promoted on a flock: designation of leaders, or implicit in the flocking rules.

The use of the leader-follower technique has been widely adopted for different types of robotic missions. Both, predefined leaders as in [60] and [114] or self-assigned leaders as in [154] are viable options for such implementation. Predefined leaders have the advantage that they can be provided with the necessary equipment to perform the guiding task, such as the presence of distinctive markers [55]. Zongyao et al. (2009) showed hidden predefined leaders also prove effective to control the movement of the swarm by pulling with the follower drones through the attraction-repulsion forces. Self-assigned leaders on the other hand are robust to leader failure, as they can dynamically reassign leaders based

on the needs of the swarm [87]. One aspect to keep in mind when using this method is the leader assignment. Leaders can be assigned based on drone ID when using a communication system [87]. Assigning leaders randomly is also a possibility, but lacking the knowledge of the existence of other leaders [154].

The alternative to designated leaders that are followed by the rest of the flock is to embed the exploration behaviour within the flocking rules. To the three flocking rules, separation, alignment and cohesion, more rules can be added to promote exploration. Bouraqadi et al (2009) added collision avoidance and exploration towards the nearest frontier as two extra rules. Assigning higher priorities to separation and collision avoidance, a weighted combination of the other three rules ensured exploration while preserving the flock formation [27]. Another approach to this method lies at the intersection between auction strategy for frontier assignment and cost function for formation control. Both objectives, exploration and formation control, are promoted simultaneously by combining distance to frontiers and location of neighbouring drones in the navigation planner employing internal bid functions [77].

Considering all the different techniques proposed to promote flocking exploration, a selection of the most appropriate ones can be made based on the problem at hand. Regarding flock preservation, attraction-repulsion forces appear as the only viable option. The formation cost function approach, while achieving higher performance, has a strong requirement on a centralised system that does not apply. Both potential field or fuzzy rules can be used to enforce the basic flocking rules depending on the preferred implementation. For exploration leadership, designated leaders, as well as embedded exploration behaviour in flocking rules, can be used. However, predefined leaders are to be discarded as it is less robust to leader-drone-failure.

5.4. Independent exploration

The second philosophy for area exploration also takes inspiration from nature. Independent exploration can be found on tasks where it is beneficial to split an objective with a big search space among many agents and reduce the search time. Finding a food source is the most typical example, and is tackled by millions of insects daily. The main characteristic of independent exploration is the lack of formation between the agents. The level of coordination between the independent agents can vary and is what determines the effectiveness of the swarm. For a robotic swarm, the main benefit of independent exploration is the reduction in the mission time. When coordinated perfectly, the reduction in the mission time is directly proportional to the increment in the number of agents, something very appealing when using a large number of cheap robots.

The main question to solve when designing a swarm that promotes independent exploration is the coordination between the agents. Independent agent exploration without coordination cannot be referred to as a swarm, since there is no added benefit to the collective from the sum of its parts. On insects, for example, apparent random coordinated-less exploration of swarms of ants is followed by path tracking and following as soon as a food source is found. There are two methods with which to coordinate a swarm. The first one is with explicit rules that determine precisely the interactions between the agents. The second one is to embed the coordination strategy inside the navigation rules.

Explicitly determining the coordination between the agents can allow for precise swarm control. By clearly defining the interaction between the agents, complex tasks can be achieved. Distributed exploration can leverage the use of many robots to speed up tasks such as map generation [39]. Not only drones can directly spread around the environment under predefined rules, but also, for instance, exchange data from commonly visited places to improve the recorded measurements for map generation tasks. Drones can exchange frontier points, and distribute them to increase exploration speed. With a lower level of coordination, this method can also be implemented on communication-less systems, where the agents coordinate solely based on the relative sensing of the surrounding drones. Two very famous examples of such implementation are Conway's Game of Life and Langton's Ant [53], [79]. For exploration, this approach can integrate splitting and grouping rules at intersections to enforce the distribution of the agents around the environment [125].

On the other hand, it is also possible to embed the independent exploration coordination inside the navigation rules of each agent. That way, it is not necessary to explicitly determine what the agent should do when it encounters another drone, but rather the coordination emerges from the attraction-repulsion forces. Attraction-repulsion forces, in this case, are not implemented as for the flocking method. Instead, they are used to derive the coordination strategy. For instance, by having strong separation requirements

as well as exploration curiosity, the agents can achieve a stable distribution around the environment that scans all the areas simultaneously. One drawback of this method is that it is more challenging to obtain the desired behaviour when specific requirements apply. For example, it is not possible to avoid re-exploration of the same areas when not explicitly avoiding it. Furthermore, to the best of the author's knowledge, no prior work has been conducted with such an approach, as most of the independent communication-less systems deal with strong predefined coordination rules.

Taking into account the added difficulty of embedding the coordination strategy into the navigation logic, explicit interaction rules present themselves as the only viable option for independent exploration. Since the coordination capabilities are already limited by using a communication-less system, it is important to exploit such a system to its maximum potential and not limit it by the algorithm selection. Furthermore, explicit rules allow for smart planning and avoiding re-exploration, which can increase notably the exploration time as desired by the requirements.

5.5. Comparison of coordination strategies

Having presented a wide variety of different coordination and exploration strategies used for swarm control, it is worth analysing which of the proposed methods best suits the desired application based on the requirements described in section 5.1. Both the exploration as well as the coordination strategies have to be selected simultaneously, as they are dependent on each other. However, some options can be disregarded due to undesired performance or not compliance with the requirements. Even if a brief preliminary analysis was already conducted on the preferred implementations for the flocking and independent exploration, an integrated discussion also considering the exploration techniques can allow for the design of a complete strategy that meets the system's needs.

Starting with the exploration strategy, there is one requirement that stands out due to the performance constraint it sets. If the system shall be aware of task completion, there is a direct need for a method to keep track of the progress being done. Behavioural exploration, offering only reactive control to sensory input, does not allow for an understanding of the development of the exploration. While allowing navigation with very little computational resources, higher-level tasks are beyond the capabilities of such simple methods. Path integration is the maximum level of abstraction that can be achieved with behavioural exploration, but it only allows for loop detection and homing and not full exploration progress status.

Hierarchical exploration, on the other hand, does offer the tools needed to achieve this level of understanding of the mission evolution. Updating a world model, the agents can keep track of the already explored as well as the to be explored areas. It was already discussed before how a topological representation of the environment is the best alternative for the exploration task under computational constraints. Topological maps can be pictured as a technique to describe a certain place with the least amount of information possible. Only the relationships between the different places are important, not their actual geometric characterisation.

Frontier assignment on topological maps can be performed just like for the other map representations. From the different options considered, discrete or continuous, the selection is dependent on the type of coordination philosophy followed. Obviously, a flocking swarm does not need to share the optimal exploration strategy with an independent exploration swarm, as the system has different properties. Therefore, both options are considered separately and then that which meets the requirements with the best performance can be selected for implementation on a real system.

Exploration by flocking allows all the drones to navigate the environment as a single unit. Attraction-repulsion forces were selected over the cost-function approach for flock preservation due to the need for a centralised system of the latter option. For the exploration strategy, both leader-follower techniques and rules-implicit exploration curiosity were deemed as viable. When using a leader-follower approach with self-assigned leaders, a discrete approach for frontier assignment can ease the frontier selection process and increase the exploration speed. A small amount of leaders is preferred, more so just one, as the benefit of this approach is to reduce the coordination between the agents and allow them to navigate by following. More leaders could generate stronger pulling forces for the flock and achieve faster speeds, but at the expense of inter-leader coordination for frontier assignment. One drawback of this technique is the unsolved issue of the leader assignment for a communication-less swarm. When embedding the exploration curiosity in the flocking rules, a continuous approach to frontier assignment could have more advantages. The same potential function can be employed to integrate both the

flocking preservation logic, exploration leadership and frontier assignment. Accommodating all the different behaviours in one function can allow the swarm to satisfy both objectives at the same time. One drawback from this potential field approach is that, since the exploration is emergent and not explicit, it may take large amounts of time to explore areas that are far from each other and therefore have low potential pulling force. Therefore, self-assigned leaders with discrete frontier assignment stand out as the best performing option for the flocking philosophy. However, if the leader assignment challenge is not solved, the lower performance potential field approach should be used.

Independent exploration of the different drones on the other hand promotes simultaneous exploration of different areas. As discussed before, explicitly determining the interaction rules offers a better alternative to embedding the coordination on the navigation rules for robot coordination. Being able to determine the inter-drone interactions, a discrete approach to frontier assignment could benefit from a deterministic behaviour. A graph exploration algorithm can leverage the knowledge of the interaction rules to make assumptions on parts of the graph that are being explored by other drones. Marking as explored branches of the graph when drones return from a certain area, the exploration speed can be greatly increased. Special planning needs to be employed with cyclic graphs for such assumptions. If an efficient tree exploration algorithm is designed to exploit deterministic multi-drone interactions, a system with a great performance could be achieved. However, such a method has not been proposed before, to the best of the author's knowledge.

Comparing the systems for both philosophies, it is possible to understand why the exploration of an unknown environment with a decentralised, communication-less swarm still remains an open challenge. There is not a technique that stands out as easily implementable with outstanding performance. On one hand, exploration by flocking, while settling with lower performance, could be a strategy that can be designed combining already existing and tested algorithms. On the other hand, independent exploration offers potentially better performance, but at the cost of a still to-be-designed algorithm that can exploit the deterministic inter-drone interactions. One benefit is that the platform required to implement both options is the same. Only the navigation logic changes. Therefore, both options could be implemented. One to prove the basic working solution of a decentralised communication-less drone swarm for exploration of an unknown environment by flocking. And another one, subjected to the development of a multi-agent graph exploration algorithm, to demonstrate a system that simultaneously explores different areas achieving greater performance.

Part III

Additional Results

6

Analysis frequency sensitivity analysis

The analysis frequency N was presented in Part I as one of the most influential variables on the propeller detectability. Moreover, having selected a constant frame accumulation k of one for the events, the parameter N also determines the event accumulation time, given by $1/N$. As the importance of this variable is highlighted, it becomes necessary to perform a sensitivity analysis for two main reasons.

Firstly, it allows the validation of the detection metrics derived. The objective behind performing a high frequency analysis was to allow multiple observations per period on the spinning propellers. Then, the signal classification can be performed given the periodic behaviour of the signal. However, when the analysis frequency is smaller than the rotation frequency of the propeller, multiple rotations periods are grouped in the same observation frames loosing its periodic characterisation. By performing a sensitivity analysis on the analysis frequency it is possible to prove the effect of the signal periodicity on its detectability.

Secondly, it can be employed to improve the design of the onboard design when mounted on the drone. For this research, a value of 2000 Hz was selected to respect the analysis requirements described in section 2.3 on all test cases. However, a very good and expensive processor may be needed to run the drone detection algorithm at such frequency. While searching for a lower value that satisfies the capabilities of the onboard processor, it is important to understand the requirements imposed by the specific system used on the analysis frequency.

6.1. Results for different N values

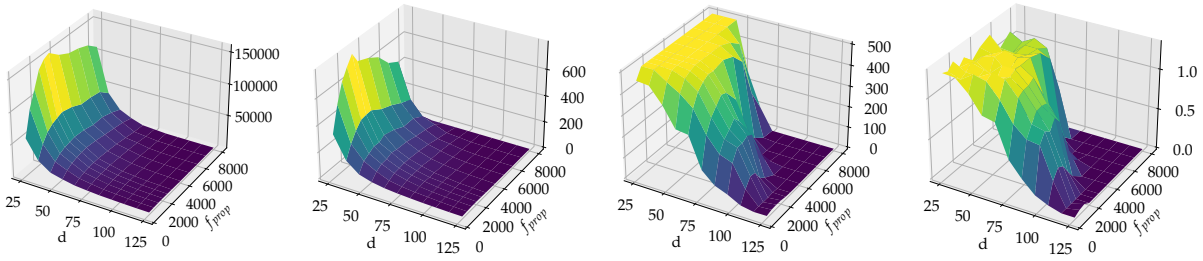
In order to perform the sensitivity analysis, the setup with the two-blade five-inch matte black propeller was selected. These conditions showed the best performance with a high number of events triggered, highlighting the effects of changing the analysis frequency.

The sensitivity analysis is focused on the four metrics studied for every test - event count, event periodicity, clustering count and clustering periodicity - giving higher importance to the periodicity values. As the base results for $N = 2000$ are already shown in section 2.4, the results here presented showcase directly the changed values of the analysis frequency.

Three different frequencies were tested to study the sensitivity on the results. For their selection, it is important to recall the two theoretical requirements derived in section 2.4 on the minimum analysis frequency. Firstly, it was concluded that $1/N < 1/(B \cdot f_{prop})$, having assigned the value of k to one. Secondly, it was also stated that $1/N < (4-B)/(4B \cdot f_{prop})$. Substituting the appropriate number of blades and only looking at the most strict requirement, it is determined that $1/N < 1/(4f_{prop})$. Comparing with the range of spinning frequencies tested, between 500 and 8000 RPMs, the three values must be chosen such that only part of the tests values meet the requirement.

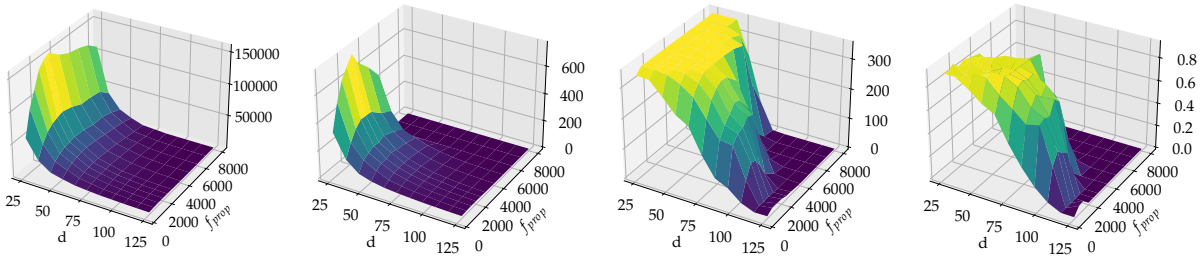
The three values selected for the analysis frequencies are 500, 350 and 200 Hz, with corresponding threshold propeller frequencies of 7500, 5250 and 3000 RPMs, respectively. The results for the metrics obtained can be found in Figure 6.1 for $N = 500$, Figure 6.2 for $N = 350$, and Figure 6.3 for $N = 200$.

It must be noted that only one experimental test was conducted, and it was the raw data that was analysed at different frequencies. This was done to see the effects of changing the analysis frequency alone, without the variability of the specific recordings to play a role.



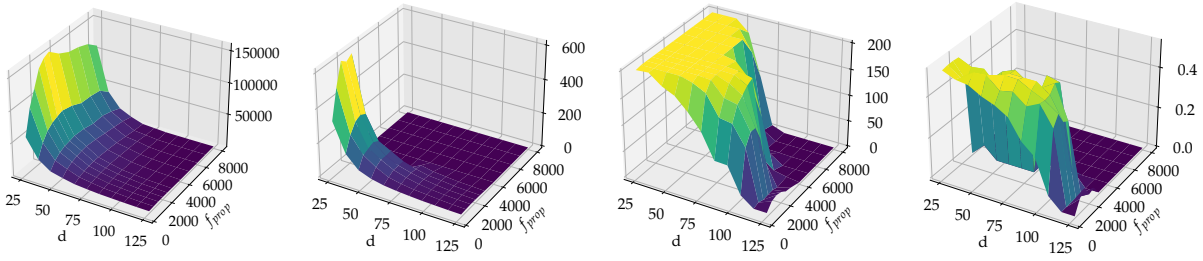
(a) Event count. **(b)** Event periodicity. **(c)** Clustering count. **(d)** Clustering periodicity.

Figure 6.1: Results for two-blade five-inch matte black propeller, with artificial lighting and clear background versus distance [cm] and propeller frequency [RPMs] at an analysis frequency of 500 Hz.



(a) Event count. **(b)** Event periodicity. **(c)** Clustering count. **(d)** Clustering periodicity.

Figure 6.2: Results for two-blade five-inch matte black propeller, with artificial lighting and clear background versus distance [cm] and propeller frequency [RPMs] at an analysis frequency of 350 Hz.



(a) Event count. **(b)** Event periodicity. **(c)** Clustering count. **(d)** Clustering periodicity.

Figure 6.3: Results for two-blade five-inch matte black propeller, with artificial lighting and clear background versus distance [cm] and propeller frequency [RPMs] at an analysis frequency of 200 Hz.

6.2. Analysis of the results

The effect of changing the analysis frequency has a different magnitude depending on the metric analysed. As expected, the event count is unaffected by it. It does not matter how fast or slow the analysis is performed, the number of events triggered is independent. However, this is not the case for the other metrics.

Looking at the event clustering count, it can be observed how there is a twofold effect. On one hand, an increase in the cluster count can be observed with respect to the total number of frames. On the other hand, there is a decrease in the smoothness of the surface with decreasing N. This can be explained because, as N goes down, the event accumulation time increases, boosting clustering due to the larger number of events in every observation frame. However, this behaviour is only shown up to where the propeller is visible. Lower values of N translate therefore to a more polarised clustering count.

Lastly, the effect on the event and clustering periodicity can be discussed. From section 2.4, it is clear that the periodicity results must match the shape of the count results, for both the events and the clusters, because the source that triggers the events is periodic by nature. The results here presented support that statement, but only up until the threshold propeller frequencies. For Figure 6.3, for example, it is observed that the shape of the event periodicity matches that of the event count, but only until 2500 RPMs. Beyond that value, a periodicity of zero is obtained. The same can be said about the clustering periodicity. The same behaviour can be seen in Figure 6.1 and Figure 6.2, but for threshold propeller frequencies of 5250 and 7500 RPMs, respectively. Specially remarkable is the magnitude of the event periodicity for the different figures, which is completely unaffected by it.

These results firstly prove that the theoretical requirements on the analysis frequency are correct. When the analysis frequency does not respect the $1/N < (4 - B)/(4B \cdot f_{prop})$ requirement, the periodicity of the signal is completely lost. Secondly it demonstrates that for the values where the requirement is met, the shape of all of the metrics is unaffected, with the exception of a slight polarisation in the clustering count.

It can therefore be concluded that, as long as the analysis is performed respecting the theoretical requirements derived, the propeller signal can be characterised studying its periodicity.

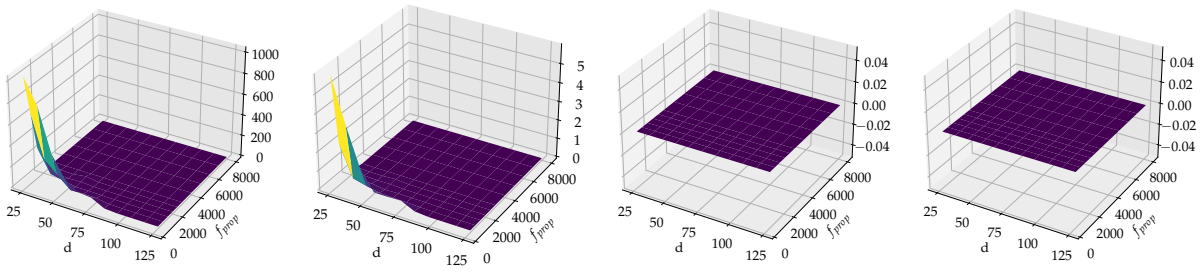
7

Test results

For the completion of this MSc thesis, extensive testing was conducted. The methodology followed, analysis conducted and results obtained were introduced in Part I. The results presented showed the propeller detection for the different test scenarios, for the tested distances and spinning frequencies.

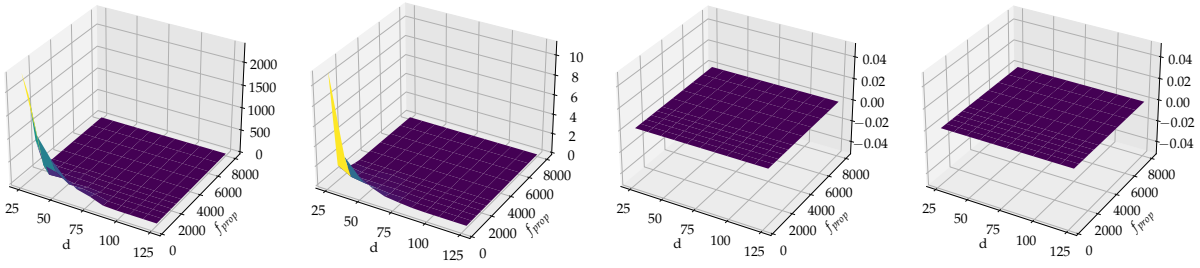
These results, however, are the outcome of the fuzzy classifier developed, which employed the different metrics extracted from the data to assess the detections. In an effort to provide insight into the detection performance of event cameras, the actual data directly extracted from the tests is presented here. The the four metrics that characterise the propeller signal - event count, event periodicity, clustering count and clustering periodicity - are shown for all the setups tested.

Figure 7.1, Figure 7.2, Figure 7.3 and Figure 7.4 present the results for the three-blade, six-inch translucid blue propeller for natural lighting and clear background, natural lighting and cluttered background, artificial lighting and clear background and artificial lighting and cluttered background, respectively. This same order for lighting and background is respected for the all the plots. Figure 7.5, Figure 7.6, Figure 7.7 and Figure 7.8 show the results for the two-blade, six-inch grey propeller. Figure 7.9, Figure 7.10, Figure 7.11 and Figure 7.12 show the results for the two-blade, six-inch black propeller. Figure 7.13, Figure 7.14, Figure 7.15 and Figure 7.16 show the results for the three-blade, five-inch pink propeller. Lastly, the results for the two-blade, five-inch matte black propeller can be found in Figure 7.17, Figure 7.18, Figure 7.19 and Figure 7.20.



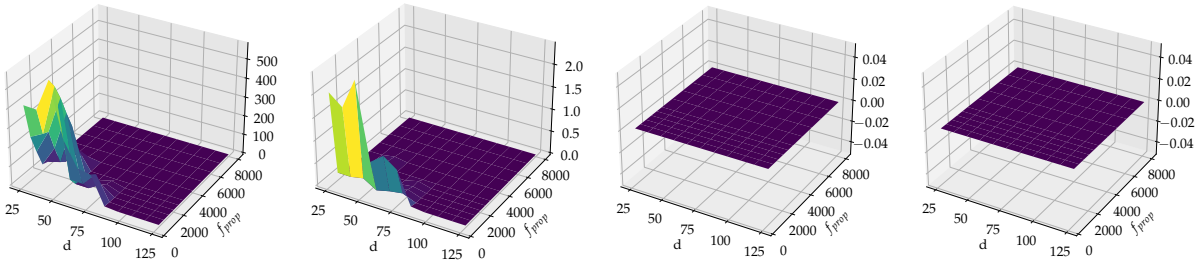
(a) Event count. **(b)** Event periodicity. **(c)** Clustering count. **(d)** Clustering periodicity.

Figure 7.1: Results for three-blade six-inch translucid blue propeller, with natural lighting and clear background versus distance [cm] and propeller frequency [RPMs].



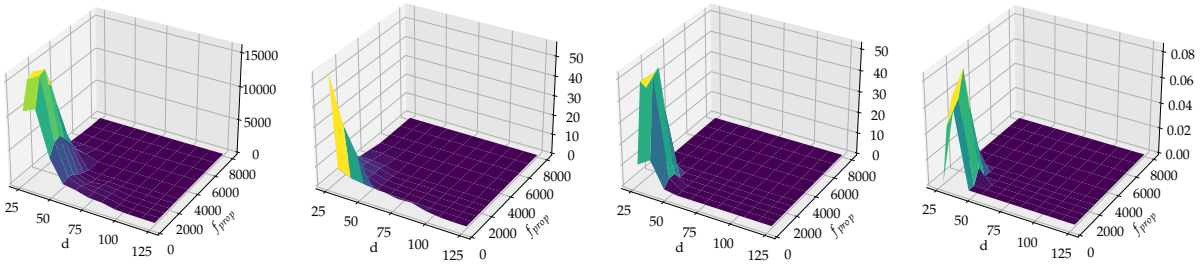
(a) Event count. **(b)** Event periodicity. **(c)** Clustering count. **(d)** Clustering periodicity.

Figure 7.2: Results for three-blade six-inch translucid blue propeller, with natural lighting and cluttered background versus distance [cm] and propeller frequency [RPMs].



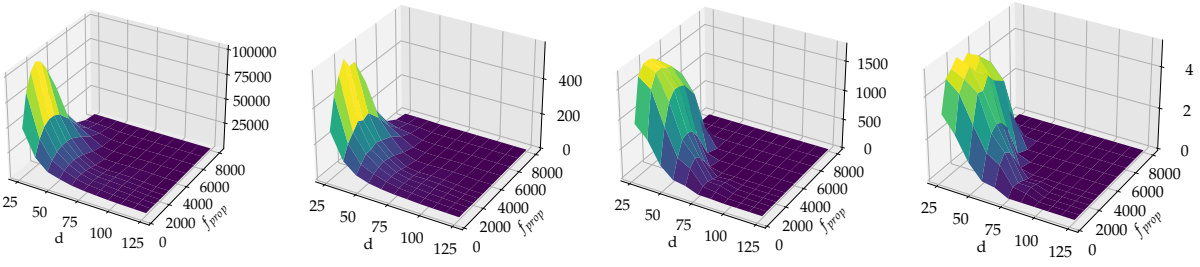
(a) Event count. **(b)** Event periodicity. **(c)** Clustering count. **(d)** Clustering periodicity.

Figure 7.3: Results for three-blade six-inch translucid blue propeller, with artificial lighting and clear background versus distance [cm] and propeller frequency [RPMs].



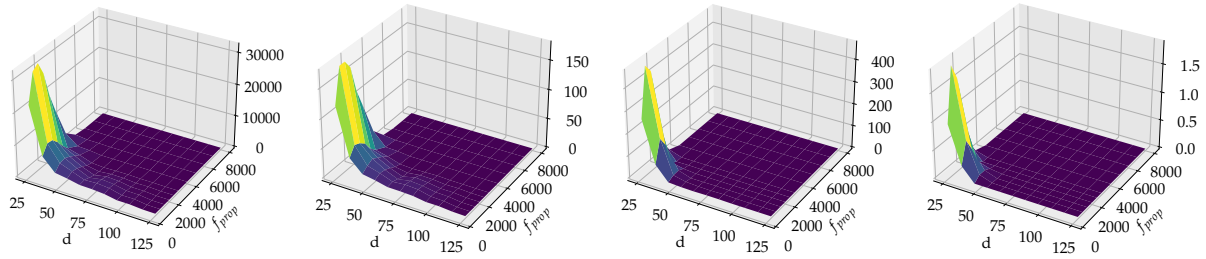
(a) Event count. **(b)** Event periodicity. **(c)** Clustering count. **(d)** Clustering periodicity.

Figure 7.4: Results for three-blade six-inch translucid blue propeller, with artificial lighting and cluttered background versus distance [cm] and propeller frequency [RPMs].



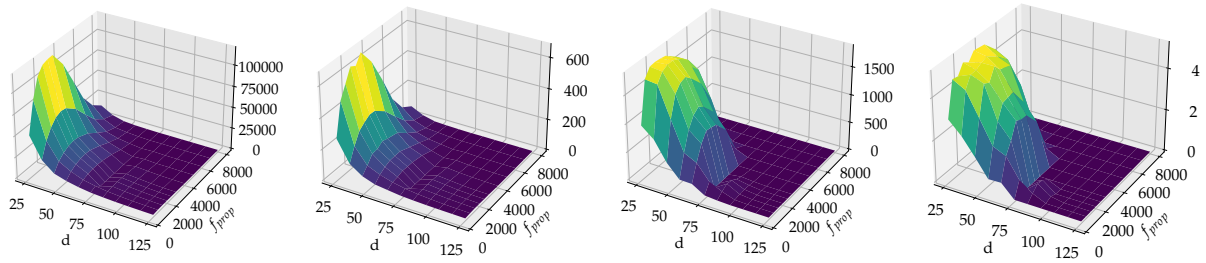
(a) Event count. **(b)** Event periodicity. **(c)** Clustering count. **(d)** Clustering periodicity.

Figure 7.5: Results for two-blade six-inch grey propeller, with natural lighting and clear background versus distance [cm] and propeller frequency [RPMs].



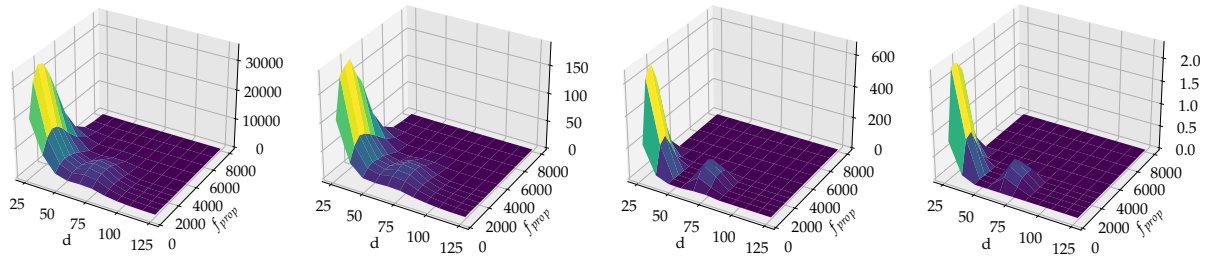
(a) Event count. **(b)** Event periodicity. **(c)** Clustering count. **(d)** Clustering periodicity.

Figure 7.6: Results for two-blade six-inch grey propeller, with natural lighting and cluttered background versus distance [cm] and propeller frequency [RPMs].



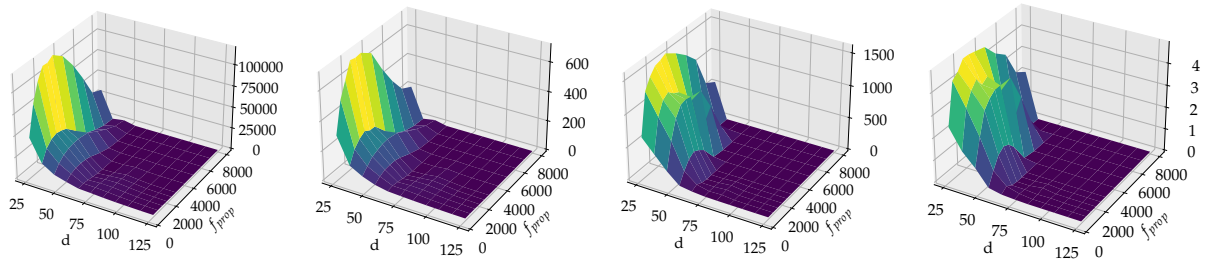
(a) Event count. **(b)** Event periodicity. **(c)** Clustering count. **(d)** Clustering periodicity.

Figure 7.7: Results for two-blade six-inch grey propeller, with artificial lighting and clear background versus distance [cm] and propeller frequency [RPMs].



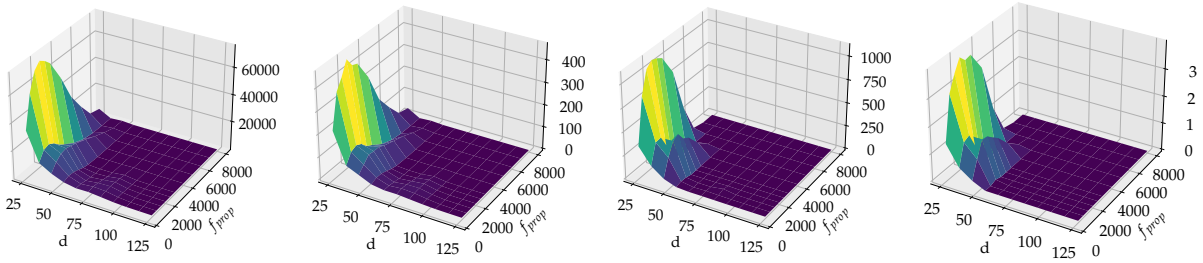
(a) Event count. **(b)** Event periodicity. **(c)** Clustering count. **(d)** Clustering periodicity.

Figure 7.8: Results for two-blade six-inch grey propeller, with artificial lighting and cluttered background versus distance [cm] and propeller frequency [RPMs].



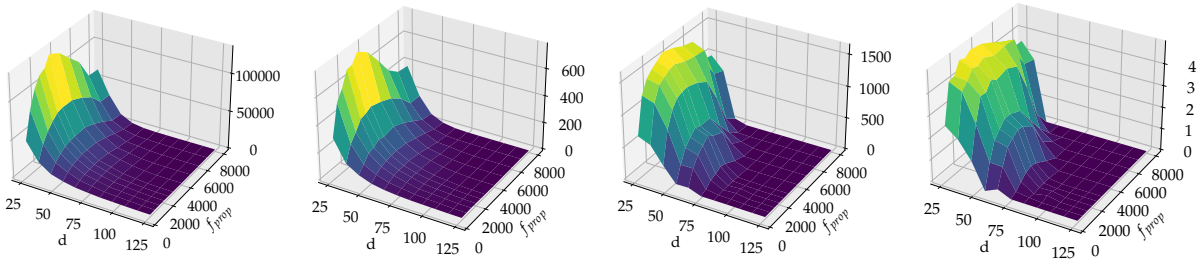
(a) Event count. **(b)** Event periodicity. **(c)** Clustering count. **(d)** Clustering periodicity.

Figure 7.9: Results for two-blade six-inch black propeller, with natural lighting and clear background versus distance [cm] and propeller frequency [RPMs].



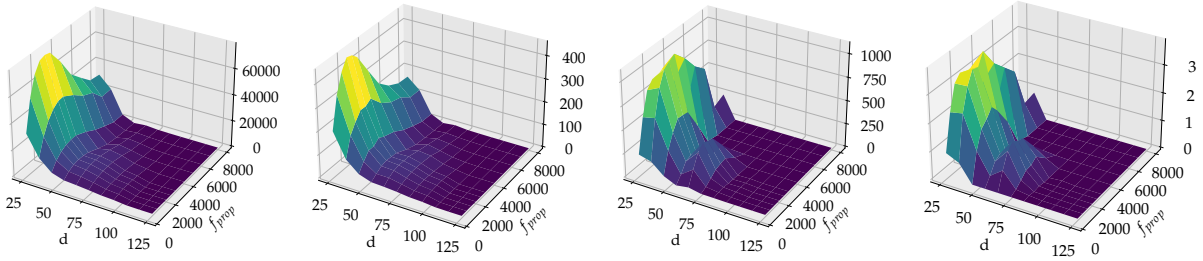
(a) Event count. **(b)** Event periodicity. **(c)** Clustering count. **(d)** Clustering periodicity.

Figure 7.10: Results for two-blade six-inch black propeller, with natural lighting and cluttered background versus distance [cm] and propeller frequency [RPMs].



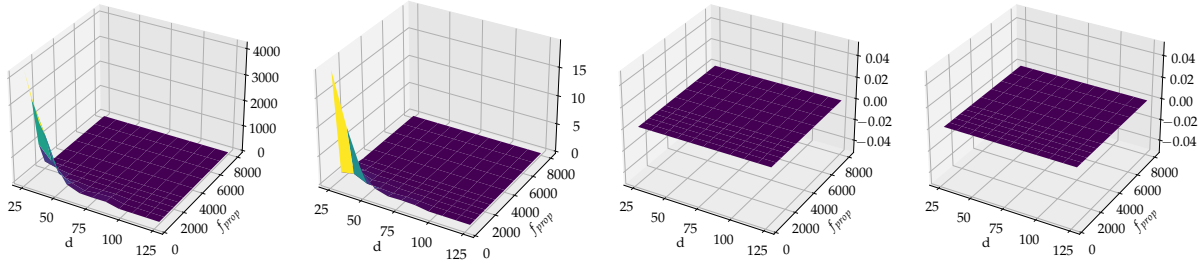
(a) Event count. **(b)** Event periodicity. **(c)** Clustering count. **(d)** Clustering periodicity.

Figure 7.11: Results for two-blade six-inch black propeller, with artificial lighting and clear background versus distance [cm] and propeller frequency [RPMs].



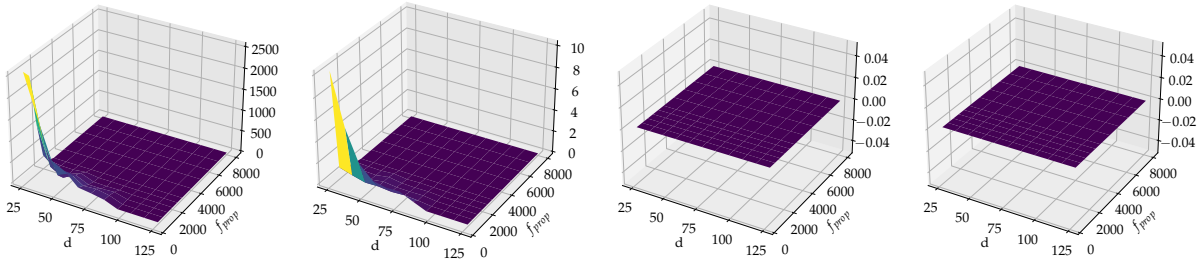
(a) Event count. **(b)** Event periodicity. **(c)** Clustering count. **(d)** Clustering periodicity.

Figure 7.12: Results for two-blade six-inch black propeller, with artificial lighting and cluttered background versus distance [cm] and propeller frequency [RPMs].



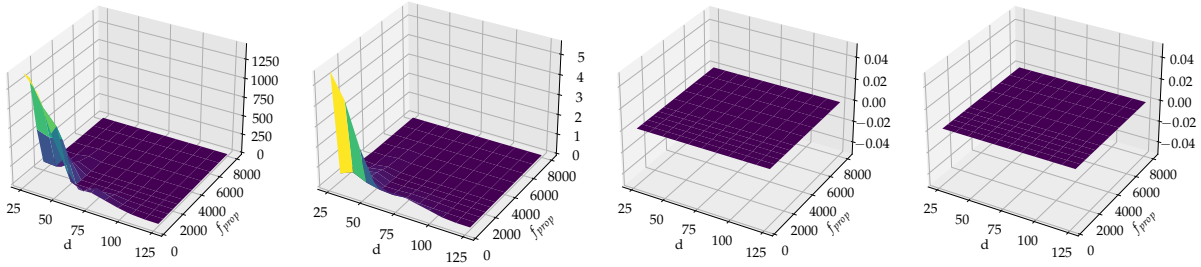
(a) Event count. **(b)** Event periodicity. **(c)** Clustering count. **(d)** Clustering periodicity.

Figure 7.13: Results for three-blade five-inch pink propeller, with natural lighting and clear background versus distance [cm] and propeller frequency [RPMs].



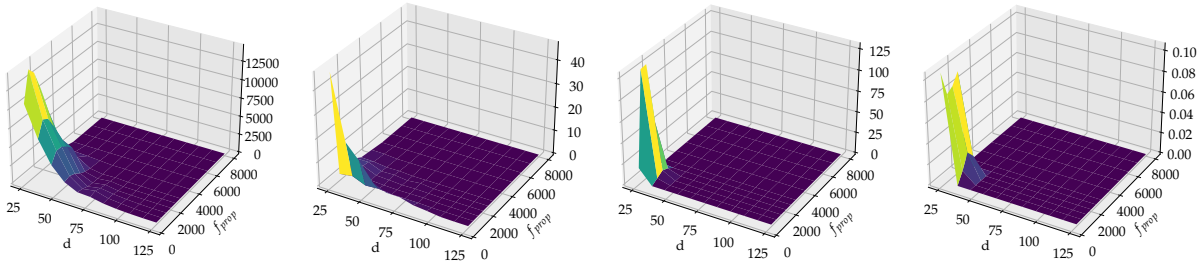
(a) Event count. **(b)** Event periodicity. **(c)** Clustering count. **(d)** Clustering periodicity.

Figure 7.14: Results for three-blade five-inch pink propeller, with natural lighting and cluttered background versus distance [cm] and propeller frequency [RPMs].



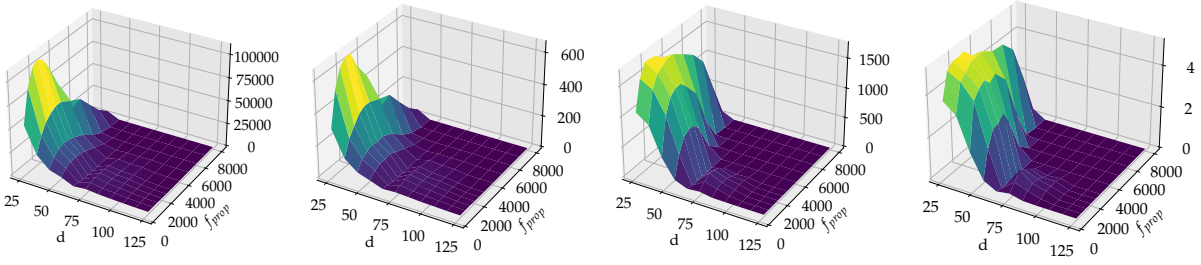
(a) Event count. **(b)** Event periodicity. **(c)** Clustering count. **(d)** Clustering periodicity.

Figure 7.15: Results for three-blade five-inch pink propeller, with artificial lighting and clear background versus distance [cm] and propeller frequency [RPMs].



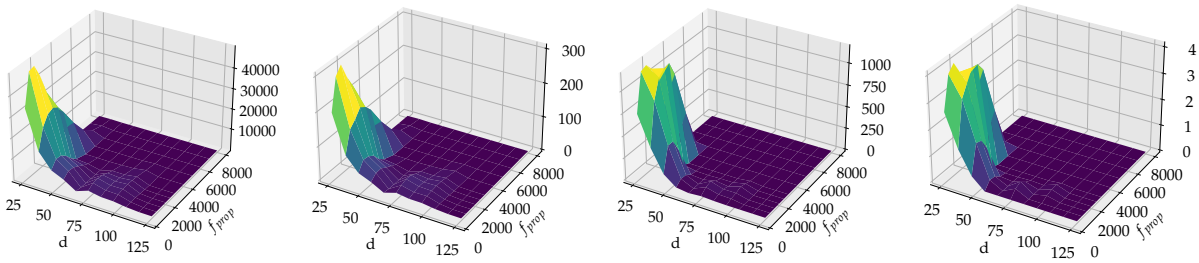
(a) Event count. **(b)** Event periodicity. **(c)** Clustering count. **(d)** Clustering periodicity.

Figure 7.16: Results for three-blade five-inch pink propeller, with artificial lighting and cluttered background versus distance [cm] and propeller frequency [RPMs].



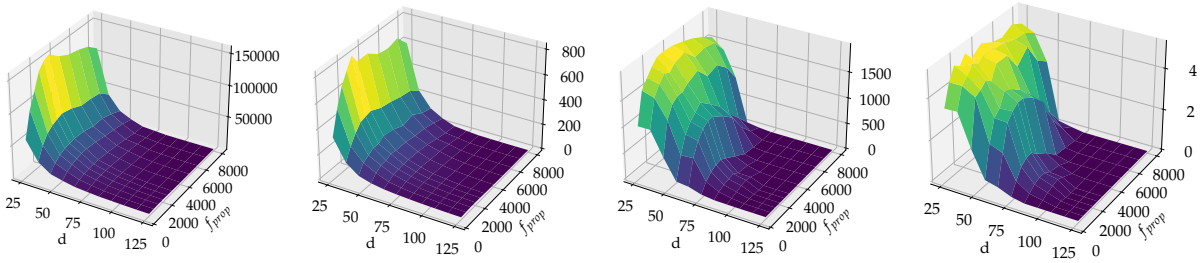
(a) Event count. **(b)** Event periodicity. **(c)** Clustering count. **(d)** Clustering periodicity.

Figure 7.17: Results for two-blade five-inch matte black propeller, with natural lighting and clear background versus distance [cm] and propeller frequency [RPMs].



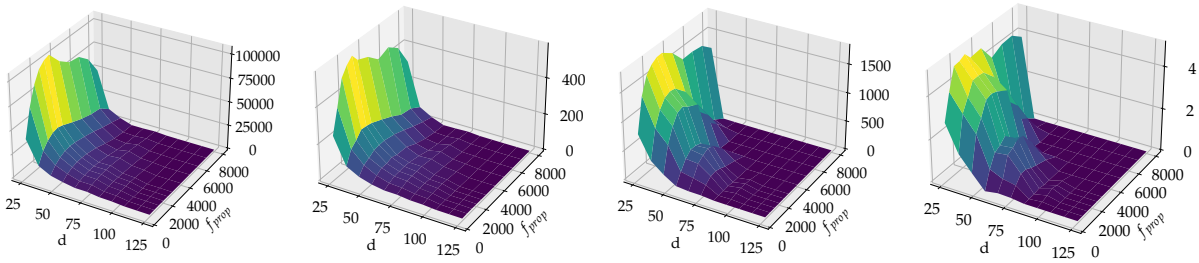
(a) Event count. **(b)** Event periodicity. **(c)** Clustering count. **(d)** Clustering periodicity.

Figure 7.18: Results for two-blade five-inch matte black propeller, with natural lighting and cluttered background versus distance [cm] and propeller frequency [RPMs].



(a) Event count. **(b)** Event periodicity. **(c)** Clustering count. **(d)** Clustering periodicity.

Figure 7.19: Results for two-blade five-inch matte black propeller, with artificial lighting and clear background versus distance [cm] and propeller frequency [RPMs].



(a) Event count. **(b)** Event periodicity. **(c)** Clustering count. **(d)** Clustering periodicity.

Figure 7.20: Results for two-blade five-inch matte black propeller, with artificial lighting and cluttered background versus distance [cm] and propeller frequency [RPMs].

Bibliography

[1] Amanda M. Adams, Kaylee Davis, and Michael Smotherman. Suppression of emission rates improves sonar performance by flying bats. *Scientific Reports*, 7, 2017.

[2] Benjamin Adler, Junhao Xiao, and Jianwei Zhang. Autonomous exploration of urban environments using unmanned aerial vehicles. *Journal of Field Robotics*, 31(6):912–939, 2014.

[3] Francesco Amigoni, Jacopo Banfi, and Nicola Basilico. Multirobot exploration of communication-restricted environments: A survey. *IEEE Intelligent Systems*, 32(6):48–57, 2017.

[4] Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, Jeff Kusnitz, Michael Debole, Steve Esser, Tobi Delbruck, Myron Flickner, and Dharmendra Modha. A low power, fully event-based gesture recognition system. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7388–7397, 2017.

[5] Ilze Andersone. Heterogeneous map merging: State of the art. *Robotics*, 8(3), 2019.

[6] Ronald C. Arkin. Motor schema based navigation for a mobile robot: An approach to programming by behavior. In *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, volume 4, pages 264–271, 1987.

[7] Ross D. Arnold, Hiroyuki Yamaguchi, and Toshiyuki Tanaka. Search and rescue with autonomous flying robots through behavior-based cooperative intelligence. *Journal of International Humanitarian Action*, 3, 2018.

[8] Jan Carlo Barca and Y. Ahmet Sekercioglu. Swarm robotics reviewed. *Robotica*, 31(3):345–359, 2013.

[9] Francisco Barranco, Cornelia Fermuller, and Eduardo Ros. Real-time clustering and multi-target tracking using event-based sensors. 2018.

[10] Juan Barrios-Avilés, Taras Iakymchuk, Jorge Samaniego, Leandro D. Medus, and Alfredo Rosado-Muñoz. Movement detection with event-based cameras: Comparison with frame-based cameras in robot object tracking using powerlink communication. *Electronics*, 7(11), 2018.

[11] Juan Barrios-Avilés, Taras Iakymchuk, Jorge Samaniego, and Alfredo Rosado-Muñoz. An event-based fast movement detection algorithm for a positioning robot using powerlink communication. 2017.

[12] Meysam Basiri, Felix Schill, Pedro Lima, and Dario Floreano. On-board relative bearing estimation for teams of drones using sound. *IEEE Robotics and Automation Letters*, 1(2):820–827, 2016.

[13] Antoine Bautin, Olivier Simonin, and François Charpillet. Minpos : A novel frontier allocation algorithm for multi-robot exploration. In Chun-Yi Su, Subhash Rakheja, and Honghai Liu, editors, *Intelligent Robotics and Applications*, pages 496–508, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[14] Sepideh Bazazi, Jerome Buhl, Joseph J. Hale, Michael L. Anstey, Gregory A. Sword, Stephen J. Simpson, and Iain D. Couzin. Collective motion and cannibalism in locust migratory bands. *Current Biology*, 18(10):735–739, 2008.

[15] Guy Beauchamp. Does group foraging promote efficient exploitation of resources? *Oikos*, 111(2):403–407, 2005.

[16] Dinesh Kumar Behera and Arockia Bazil Raj. Drone detection and classification using deep learning. In *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 1012–1016, 2020.

[17] Giovanni Benelli, Giulia Giunti, Angelo Canale, and Russell H. Messing. Lek dynamics and cues evoking mating behavior in tephritid flies infesting soft fruits: implications for behavior-based control tools. *Applied Entomology and Zoology*, 49:363–373, 2014.

[18] Ryad Benosman, Charles Clercq, Xavier Lagorce, Sio-Hoi Ieng, and Chiara Bartolozzi. Event-based visual flow. *IEEE Transactions on Neural Networks and Learning Systems*, 25(2):407–417, 2014.

[19] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008, 2008.

[20] Xavier Berthelon, Guillaume Chenegros, Thomas Finateu, Sio-Hoi Ieng, and Ryad Benosman. Effects of cooling on the snr and contrast detection of a low-light event-based camera. *IEEE Trans Biomed Circuits Syst*, 12(6):1467–1474, 2018.

[21] Olivier J. N. Bertrand, Jens P. Lindemann, and Martin Egelhaaf. A bio-inspired collision avoidance model based on spatial information derived from motion detectors leads to common routes. *PLOS Computational Biology*, 11(11):1–28, 2015.

[22] Jacobus C. Biesmeijer and Thomas D Seeley. The use of waggle dance information by honey bees throughout their foraging careers. *Behavioral Ecology and Sociobiology*, 59:133–142, 2005.

[23] Suman Raj Bista, Paolo Robuffo Giordano, and François Chaumette. Appearance-based indoor navigation by ibvs using line segments. *IEEE Robotics and Automation Letters*, 1(1):423–430, 2016.

[24] Fabian Blöchliger, Marius Fehr, Marcin Dymczyk, Thomas Schneider, and Roland Siegwart. Topomap: Topological mapping and navigation based on visual slam maps. 2018.

[25] Jaime Boal, Alvaro Sánchez-Miralles, and Alvaro Arranz. Topological simultaneous localization and mapping: a survey. *Robotica*, 32(5):803–821, 2014.

[26] James P. Boettiger. A comparative evaluation of the detection and tracking capability between novel event-based and conventional frame-based sensors. Master's thesis, Air Force Institute of Technology, Dayton, Ohio, 2020.

[27] Noury Bouraqadi and Arnaud Doniec. Flocking-based multi-robot exploration. *4rd National Conference on "Control Architectures of Robots"*, 2009.

[28] Mireille Boutin and Gregor Kemper. A drone can hear the shape of a room. *SIAM Journal on Applied Algebra and Geometry*, 4:123–140, 2020.

[29] Rodney Brooks. A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation*, 2(1):14–23, 1986.

[30] Aldrich A. Cabrera-Ponce, J. Martinez-Carranza, and Caleb Rascon. Detection of nearby uavs using a multi-microphone array on board a uav. *International Journal of Micro Air Vehicles*, 12:1756829320925748, 2020.

[31] Andrea Cavagna and Irene Giardina. The seventh starling. *Significance*, 5(2):62–66, 2008.

[32] Sarat Ch, Ra N, Leena Vachhani, and Arpita Sinha. A decentralized approach for autonomous multi-robot exploration and map building for tree structures. In *2015 Indian Control Conference*, page 274–279, India, 2015.

[33] H. Jacky Chang, C. S. George Lee, Y. Charlie Hu, and Yung-Hsiang Lu. Multi-robot slam with topological/metric maps. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1467–1472, 2007.

[34] Devendra S. Chaplot, Ruslan Salakhutdinov, Abhinav Gupta, and Saurabh Gupta. Neural topological slam for visual navigation. 2020.

[35] Devendra Singh Chaplot, Dhiraj Gandhi, Abhinav Gupta, and Ruslan Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. 2020.

[36] Mario Coppola, Kimberly N. McGuire, Christophe De Wagter, and Guido C.H.E. de Croon. A survey on swarming with micro air vehicles: Fundamental challenges and constraints. *Frontiers in Robotics and AI*, 7:18, 2020.

[37] Akansel Cosgun and Henrik I. Christensen. Context-aware robot navigation using interactively built semantic maps. *Paladyn, Journal of Behavioral Robotics*, 9(1):254–276, 2018.

[38] Micael S. Couceiro, Patricia A. Vargas, Rui P. Rocha, and Nuno M.F. Ferreira. Benchmark of swarm robotics distributed techniques in a search task. *Robotics and Autonomous Systems*, 62(2):200–213, 2014.

[39] Anthony Cowley, Camillo J. Taylor, and Ben Southall. Rapid multi-robot exploration with topometric maps. In *2011 IEEE International Conference on Robotics and Automation*, pages 1044–1049, 2011.

[40] Mohammadreza Daneshvaramoli, Mohammad Sina Kiarostami, Saleh Khalaj Monfared, Helia Karisani, Hamed Khashehchi, Dara Rahmati, and Saeid Gorgin. Decentralized communication-less multi-agent task assignment with cooperative monte-carlo tree search. In *2020 6th International Conference on Control, Automation and Robotics (ICCAR)*, pages 612–616, 2020.

[41] Tobi Delbruck and Manuel Lang. Robotic goalie with 3 ms reaction time at 4% cpu load using event-based dynamic vision sensor. *Frontiers in Neuroscience*, 7:223, 2013.

[42] David Drazen, Patrick Lichtsteiner, Philipp Häfliger, Tobi Delbrück, and Atle Jensen. Toward real-time particle tracking using an event-based dynamic vision sensor. *Experiments in Fluids*, 51, 2011.

[43] Christine M Drea and Laurence G Frank. The social complexity of spotted hyenas. In *Animal social complexity*, pages 121–148. Harvard University Press, 2013.

[44] Camera dvs240, specification sheet. `https://web.archive.org/web/20201223102302/https://inivation.com/wp-content/uploads/2020/04/DVS240.pdf`. [Online; accessed 16-June-2021].

[45] Event-based evaluation kit atis hvga gen3. `https://web.archive.org/web/20201022114024/https://www.prophesee.ai/event-based-evk/`. [Online; accessed 16-June-2021].

[46] Dror Epstein and Dan Feldman. Quadcopter tracks quadcopter via real-time shape fitting. *IEEE Robotics and Automation Letters*, 3(1):544–550, 2018.

[47] Rupert Evershed. The marvel of a starling murmuration. `https://www.hertsad.co.uk/lifestyle/the-marvel-of-a-starling-murmuration-6851868`, 2020. [Online; accessed 30-June-2021].

[48] Erina Ferro and Francesco Potorti. Bluetooth and wi-fi wireless protocols: a survey and a comparison. *IEEE Wireless Communications*, 12(1):12–26, 2005.

[49] Jeremy A. Fishel, Toni Oliver, Michael Eichermueller, Giuseppe Barbieri, Ethan Fowler, Toivo Hartikainen, Luke Moss, and Rich Walker. Tactile telerobots for dull, dirty, dangerous, and inaccessible tasks. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11305–11310, 2020.

[50] W. A. Foster and J. E. Treherne. Evidence for the dilution effect in the selfish herd from fish predation on a marine insect. *Nature*, 293:466–467, 1981.

[51] Todd M. Freeberg, Shannon K. Eppert, Kathryn E. Sieving, and Jeffrey R. Lucas. Diversity in mixed species groups improves success in a novel feeder test in a wild songbird community. *Scientific Reports*, 7, 2017.

[52] Guillermo Gallego, Tobi Delbruck, Garrick Michael Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew Davison, Jorg Conradt, Kostas Daniilidis, and et al. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–30, 2020.

[53] Mathematical Games. The fantastic combinations of john conway's new solitaire game "life" by martin gardner. *Scientific American*, 223:120–123, 1970.

[54] Emilio Garcia-Fidalgo and Alberto Ortiz. Vision-based topological mapping and localization methods: A survey. *Robotics and Autonomous Systems*, 64:1–20, 2015.

[55] Michael Gassner, Titus Cieslewski, and Davide Scaramuzza. Dynamic collaboration without communication: Vision-based cable-suspended load transport with two quadrotors. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5196–5202, 2017.

[56] Arren Glover and Chiara Bartolozzi. Event-driven ball detection and gaze fixation in clutter. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2203–2208, 2016.

[57] Arren Glover and Chiara Bartolozzi. Robust visual tracking with a freely-moving event camera. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3769–3776, 2017.

[58] Anne E. Goodenough, Natasha Little, William S. Carpenter, and Adam G. Hart. Birds of a feather flock together: Insights into starling murmuration behaviour revealed using citizen science. *PLOS ONE*, 12(6):1–18, 2017.

[59] Ashley D. Gosselin-ildari and Andreas Koenig. The effects of group size and reproductive status on vigilance in captive callithrix jacchus. *American Journal of Primatology*, 74(7):613–621, 2012.

[60] Robert Grabowski, Luis E. Navarro-Serment, Christiaan J.J. Paredis, and Pradeep K. Khosla. Heterogeneous teams of modular robots for mapping and exploration. *Autonomous Robots*, 8:293–308, 2000.

[61] Robin P. Guan, Branko Ristic, Liuping Wang, Bill Moran, and Rob Evans. Feature-based robot navigation using a doppler-azimuth radar. *International Journal of Control*, 90(4):888–900, 2017.

[62] Ondřej Holešovský, Radoslav Škoviera, Václav Hlaváč, and Roman Vítek. Experimental comparison between event and global shutter cameras. *Sensors*, 21(4), 2021.

[63] Marybeth Holleman and Gordon Haber. *Among Wolves : Gordon Haber's Insights into Alaska's Most Misunderstood Animal*. University of Alaska Press, 2013.

[64] Sahar Hoseini, Garrick Orchard, Amirreza Yousefzadeh, Balakrishna Deverakonda, Teresa Serrano-Gotarredona, and Bernabé Linares-Barranco. Passive localization and detection of quadcopter uavs by using dynamic vision sensor. In *2017 5th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, pages 81–85, 2017.

[65] Dean F. Hougen, S. Benjaafar, Jordan C. Bonney, J.R. Budenske, M. Dvorak, Maria Gini, H. French, Donald G. Krantz, P.Y. Li, F. Malver, Bradley Nelson, Nikolaos Papanikolopoulos, Paul E. Rybski, Sascha A. Stoeter, Richard M. Voyles, and K.B. Yesin. A miniature robotic system for reconnaissance and surveillance. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 1, pages 501–507, 2000.

[66] Jie Huang, Ning Zhou, and Ming Cao. Adaptive fuzzy behavioral control of second-order autonomous agents with prioritized missions: Theory and experiments. *IEEE Transactions on Industrial Electronics*, 66(12):9612–9622, 2019.

[67] Y.L. Ip, Ahmad B. Rad, and Yiu-Kwong Wong. Autonomous exploration and mapping in an unknown environment. In *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826)*, volume 7, pages 4194–4199 vol.7, 2004.

[68] Mohd. Kamil Usmani and Shahnila Usmani. Locusts. In *Pests and Their Management*, pages 825–869. Springer Singapore, 2018.

[69] Evan Kaufman, Kuya Takami, Taeyoung Lee, and Zhuming Ai. Autonomous exploration with exact inverse sensor models. *J. Intell. Robotics Syst.*, 92(3–4):435–452, 2018.

[70] Ian Kelly and Alcherio Martinoli. A scalable, on-board localisation and communication system for indoor multi-robot experiments. In *Sensor Review*, volume 24, pages 167–180, 2004.

[71] James Kennedy and Russell C. Eberhart. *Swarm Intelligence*. The Morgan Kaufann Series in Evolutionary Computation. Morgan Kaufmann Publishers Inc., San Francisco, 2001.

[72] Muhammad S. A. Khan, Mohammad S. Hasan, and Tarem Ahmed. A new multi-robot search algorithm using probabilistic finite state machine and lennard-jones potential function. In *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 850–855, 2018.

[73] Paul Kirkland, Gaetano Di Caterina, John Soraghan, Yannis Andreopoulos, and George Matich. Uav detection: A stdp trained deep convolutional spiking neural network retina-neuromorphic approach. In *Artificial Neural Networks and Machine Learning – ICANN 2019: Theoretical Neural Computation*, pages 724–736, 2019.

[74] Filip Klaesson, Petter Nilsson, Tiago Vaquero, Scott Tepsuporn, Aaron Ames, and Richard Murray. Planning and optimization for multi-robot planetary cave exploration under intermittent connectivity constraints. 2020.

[75] H.W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[76] M. Kulich, T. Juchelka, and L. Preucil. Comparison of exploration strategies for multi-robot search. *Acta Polytechnica*, 55(3):162–168, 2015.

[77] Ashish Kumar, Sanjeev Sharma, Ritu Tiwari, and Samriddhi Majumdar. Area exploration by flocking of multi robot. *Procedia Engineering*, 41:377–382, 2012. International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012).

[78] A. B. Kurzhanski. Problem of collision avoidance for a team motion with obstacles. *Proceedings of the Steklov Institute of Mathematics*, 293:120–136, 2016.

[79] Christopher G Langton. Studying artificial life with cellular automata. *Physica D: Nonlinear Phenomena*, 22(1):120–149, 1986.

[80] Shuo Li, Michaël M.O.I. Ozo, Christophe De Wagter, and Guido C.H.E. de Croon. Autonomous drone race: A computationally efficient vision-based navigation and control strategy. *Robotics and Autonomous Systems*, 133:103621, 2020.

[81] Shushuai Li, Mario Coppola, Christophe De Wagter, and Guido C.H.E. de Croon. An autonomous swarm of micro flying robots with range-based relative localization. 2020.

[82] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128× 128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, 2008.

[83] Martin Litzenberger, Bernard Kohn, Ahmed Nabil Belbachir, Nikolaus Donath, Gerhard Gritsch, Heinrich Garn, Christoph Posch, and Stephan Schraml. Estimation of vehicle speed based on asynchronous data from a silicon retina optical sensor. In *2006 IEEE Intelligent Transportation Systems Conference*, pages 653–658, 2006.

[84] Martin Litzenberger, Christoph Posch, Daniel Bauer, Ahmed Nabil Belbachir, Peter Schon, Bernhard Kohn, and Heinrich Garn. Embedded vision system for real-time object tracking using an asynchronous transient vision sensor. In *2006 IEEE 12th Digital Signal Processing Workshop 4th IEEE Signal Processing Education Workshop*, pages 173–178, 2006.

[85] Kian Hsiang Low, John M. Dolan, and Pradeep Khosla. Adaptive multi-robot wide-area exploration and mapping. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '08, page 23–30, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems.

[86] Matteo Luperto, Michele Antonazzi, Francesco Amigoni, and N. Alberto Borghese. Robot exploration of indoor environments using incomplete and inaccurate prior knowledge. *Robotics and Autonomous Systems*, 133:103622, 2020.

[87] Nesrine Mahdoui, Vincent Frémont, and Enrico Natalizio. Communicating multi-uav system for cooperative slam-based exploration. *Journal of Intelligent Robotic Systems*, 98:325–343, 2020.

[88] Ling Mao, Jiapin Chen, Zhenbo Li, and Dawei Zhang. Relative localization method of multiple micro robots based on simple sensors. *International Journal of Advanced Robotic Systems*, 10(2):128, 2013.

[89] Ivan Maurović, Marija ðakulović, and Ivan Petrović. Autonomous exploration of large unknown indoor environments for dense 3d model building. *IFAC Proceedings Volumes*, 47(3):10188–10193, 2014. 19th IFAC World Congress.

[90] Alex S. Mauss and Alexander Borst. Optic flow-based course control in insects. *Current Opinion in Neurobiology*, 60:21–27, 2020. Neurobiology of Behavior.

[91] Kimberly N. McGuire, Guido C.H.E. de Croon, and Karl Tuyls. A comparative study of bug algorithms for robot navigation. *Robotics and Autonomous Systems*, 121:103261, 2019.

[92] Kimberly N. McGuire, Christophe De Wagter, Karl Tuyls, Hilbert J. Kappen, and Guido C.H.E. de Croon. Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment. *Science Robotics*, 4(35), 2019.

[93] M. B. Milde, O. J. N. Bertrand, R. Benosman, M. Egelhaaf, and E. Chicca. Bioinspired event-driven collision avoidance algorithm based on optic flow. In *2015 International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)*, pages 1–7, 2015.

[94] Michael Milford, Hanme Kim, Stefan Leutenegger, and Andrew Davison. Towards visual slam with event-based cameras. 2015.

[95] Eduardo Montijano, Eric Cristofalo, Dingjiang Zhou, Mac Schwager, and Carlos Sagüés. Vision-based distributed formation control without an external positioning system. *IEEE Transactions on Robotics*, 32(2):339–351, 2016.

[96] Elias Mueggler, Guillermo Gallego, and Davide Scaramuzza. Continuous-time trajectory estimation for event-based vision sensors. In *Robotics: Science and Systems*, 2015.

[97] Elias Mueggler, Basil Huber, and Davide Scaramuzza. Event-based, 6-dof pose tracking for high-speed maneuvers. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2761–2768, 2014.

[98] Yash Mulgaonkar, Anurag Makineni, Luis Guerrero-Bonilla, and Vijay Kumar. Robust aerial robot swarms without collision avoidance. *IEEE Robotics and Automation Letters*, 3(1):596–603, 2018.

[99] Robin R. Murphy. Human-robot interaction in rescue robotics. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 34(2):138–153, 2004.

[100] Danial Nakhaeinia, S. Tang, Samsul Noor, and O. Motlagh. A review of control architectures for autonomous navigation of mobile robots. *International Journal of Physical Sciences*, 6:169–174, 2011.

[101] Ian Newton and Keith Brockie. *The Migration Ecology of Birds*. Elsevier Science & Technology, 2007.

[102] Thi Thanh Van Nguyen, Manh Duong Phung, and Quang Vinh Tran. Behavior-based navigation of mobile robot in unknown environments using fuzzy logic and multi-objective optimization. *International Journal of Control and Automation*, 10(2):349–364, 2017.

[103] Zhenjiang Ni, Aude Bolopion, Joël Agnus, Ryad Benosman, and Stéphane Regnier. Asynchronous event-based visual shape tracking for stable haptic feedback in microrobotics. *IEEE Transactions on Robotics*, 28(5):1081–1089, 2012.

[104] Zhenjiang Ni, Sio-Hoi Ieng, Christoph Posch, Stéphane Régnier, and Ryad Benosman. Visual tracking using neuromorphic asynchronous event-based cameras. *Neural Computation*, 27(4):925–953, 2015.

[105] Shawn R. Noren, George Biedenbach, Jessica V. Redfern, and Elizabeth F. Edwards. Hitching a ride: the formation locomotion strategy of dolphin calves. *Functional Ecology*, 22(2):278–283, 2008.

[106] Tobias Nägeli, Christian Conte, Alexander Domahidi, Manfred Morari, and Otmar Hilliges. Environment-independent formation flight for micro aerial vehicles. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1141–1146, 2014.

[107] Garrick Orchard, Cedric Meyer, Ralph Etienne-Cummings, Christoph Posch, Nitish Thakor, and Ryad Benosman. Hfirst: A temporal approach to object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(10):2028–2040, 2015.

[108] Ivana Palunko, Rafael Fierro, and Patricio Cruz. Trajectory generation for swing-free maneuvers of a quadrotor with suspended payload: A dynamic programming approach. In *2012 IEEE International Conference on Robotics and Automation*, pages 2691–2697, 2012.

[109] Himanshu Patel, Craig Iaboni, Deepan Lobo, Ji-won Choi, and Pramod Abichandani. Event camera based real-time detection and tracking of indoor ground robots. *arXiv e-prints*, page arXiv:2102.11916, 2021.

[110] Bernd Wasiolka Wildlife Photography. Springbok herd. `http://www.wildphotolife.com/services/print/open-edition-prints/product/174-springbok-herd`. [Online; accessed 30-June-2021].

[111] Jim Pugh and Alcherio Martinoli. Relative localization and communication module for small-scale multi-robot systems. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 188–193, 2006.

[112] Robert Michael Pyle, Massimo Vignelli, and National Audubon Society. *The Audubon Society field guide to North American butterflies*. Alfred A. Knopf, 1981.

[113] Bharath Ramesh, Andrés Ussa, Luca Della Vedova, Hong Yang, and Garrick Orchard. Low-power dynamic object detection and classification with freely moving event cameras. *Frontiers in Neuroscience*, 14:135, 2020.

[114] Alessandro Renzaglia and Agostino Martinelli. Potential field based approach for coordinate exploration with a multi-robot team. In *2010 IEEE Safety Security and Rescue Robotics*, pages 1–6, 2010.

[115] Guenther Retscher, Vassilis Gikas, Hannes Hofer, Harris Perakis, and Allison Kealy. Range validation of uwb and wi-fi for integrated indoor positioning. volume 11, pages 187–195, 2019.

[116] Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, page 25–34, New York, NY, USA, 1987. Association for Computing Machinery.

[117] Frederic Rivard, Jonathan Bisson, Francois Michaud, and Dominic Letourneau. Ultrasonic relative positioning for multi-robot systems. In *2008 IEEE International Conference on Robotics and Automation*, pages 323–328, 2008.

[118] Juan Pablo Rodriguez Gomez, Augusto Gomez Eguiluz, J. Ramiro Martinez-de Dios, and Anibal Ollero. Asynchronous event-based clustering and tracking for intrusion monitoring in uas. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8518–8524, 2020.

[119] Steven Roelofsen, Denis Gillet, and Alcherio Martinoli. Reciprocal collision avoidance for quadrotors using on-board visual detection. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4810–4817, 2015.

[120] Brian Rogers. Optic flow: Perceiving and acting in a 3-d world. *i-Perception*, 12(1), 2021. PMID: 33613957.

[121] B. Rosas-Flores, A. Hernández-Zavala, and J. Huerta-Ruelas. Lightweight obstacle detector based on scattered ir and lock-in filtering. *Infrared Physics & Technology*, 105:103157, 2020.

[122] Sajad Saeedi, Michael Trentini, Mae Seto, and Howard Li. Multiple-robot simultaneous localization and mapping: A review. *Journal of Field Robotics*, 33(1):3–46, 2016.

[123] Erol Şahin. Swarm robotics: From sources of inspiration to domains of application. In Erol Şahin and William M. Spears, editors, *Swarm Robotics*, pages 10–20, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[124] Erol Şahin, Sertan Girgin, Levent Bayindir, and Ali Emre Turgut. Swarm robotics. In Christian Blum and Daniel Merkle, editors, *Swarm Intelligence: Introduction and Applications*, pages 87–100, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[125] Paul Schermerhorn and Matthias Scheutz. Social coordination without communication in multi-agent territory exploration tasks. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '06, page 654–661, New York, NY, USA, 2006. Association for Computing Machinery.

[126] Melissa H. Schmitt, Keenan Stears, and Adrian M. Shrader. Zebra reduce predation risk in mixed-species herds by eavesdropping on cues from giraffe. *Behavioral Ecology*, 27(4):1073–1077, 2016.

[127] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. Dbscan revisited, revisited: Why and how you should (still) use dbscan. *ACM Trans. Database Syst.*, 42(3), 2017.

[128] Thomas Seeley and Juergen Tautz. Worker piping in honey bee swarms and its role in preparing for liftoff. *Journal of comparative physiology. A, Sensory, neural, and behavioral physiology*, pages 667–676, 2001.

[129] Ulzhalgas Seidaliyeva, Daryn Akhmetov, Lyazzat Ilipbayeva, and Eric T Matson. Real-time and accurate drone detection in a video with a static background. 20, 2020.

[130] Stephen V. Shepherd and Winrich A. Freiwald. Functional networks for social communication in the macaque monkey. *Neuron*, 99(2):413–420.e3, 2018.

[131] Agusti Solanas and Miguel Angel Garcia. Coordinated multi-robot exploration through unsupervised clustering of unknown space. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 1, pages 717–721, 2004.

[132] E. Soria, F. Schiano, and D. et al. Floreano. Predictive control of aerial swarms in cluttered environments. *Nature Machine Intelligence*, 2021.

[133] Anderson Souza and Luiz M. G. Gonçalves. Occupancy-elevation grid: an alternative approach for robotic mapping and navigation. *Robotica*, 34(11):2592–2609, 2016.

[134] Mandyam V. Srinivasan. Vision, perception, navigation and 'cognition' in honeybees and applications to aerial robotics. *Biochemical and Biophysical Research Communications*, 2020.

[135] Cyrill Stachniss. *Robotic mapping and exploration*. Springer, 2009.

[136] Jake A. Steiner, Joseph R. Bourne, Xiang He, Donald M. Cropek, and Kam K. Leang. Chemical-source localization using a swarm of decentralized unmanned aerial vehicles for urban/suburban environments. volume 3 of *Dynamic Systems and Control Conference*, 2019.

[137] Tracy L. Stepien, Cole Zmurchok, James B. Hengenius, Rocío Marilyn Caja Rivera, Maria R. D'Orsogna, and Alan E. Lindsay. Moth mating: Modeling female pheromone calling and male navigational strategies to optimize reproductive success. *Applied Sciences*, 10(18), 2020.

[138] Lei Tai, Shaohua Li, and Ming Liu. Autonomous exploration of mobile robots through deep neural networks. *International Journal of Advanced Robotic Systems*, 14(4):1729881417703571, 2017.

[139] C. W. Tao. Simplification of a fuzzy mechanism with rule combination. *Soft Computing*, pages 418–425, 2001.

[140] Lucas Teixeira, Fabiola Maffra, Marco Moos, and Margarita Chli. Vi-rpe: Visual-inertial relative pose estimation for aerial vehicles. *IEEE Robotics and Automation Letters*, 3(4):2770–2777, 2018.

[141] Klaus Tempfli, Gerrit C. Huurneman, Wim H. Bakker, Lucas L.F. Janssen, Wim Feringa, Ambro S.M. Gieske, Kark A. Grabmaier, Chris A. Hecker, and John A. van der Horn. *Principles of Remote Sensing*. Enschede, 2009.

[142] Sebastian Thrun and Yufeng Liu. Multi-robot slam with sparse extended information filers. *The International Journal of Robotics Research*, 15, 2005.

[143] The Unwitting Traveller. Column of marching ants. `https://theunwittingtraveller.wordpress.com/2011/05/16/kruger-park-part-iii-from-skukuza-to-oliphants/`, 2011. [Online; accessed 30-June-2021].

[144] David Reverter Valeiras, Xavier Lagorce, Xavier Clady, Chiara Bartolozzi, Sio-Hoi Ieng, and Ryad Benosman. An asynchronous neuromorphic event-driven visual part-based shape tracking. *IEEE Transactions on Neural Networks and Learning Systems*, 26(12):3045–3059, 2015.

[145] Antoni Rosinol Vidal, Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. Ultimate slam? combining events, images, and imu for robust visual slam in hdr and high-speed scenarios. *IEEE Robotics and Automation Letters*, 3(2):994–1001, 2018.

[146] Matouš Vrba, Daniel Heřt, and Martin Saska. Onboard marker-less detection and localization of non-cooperating drones for their safe interception by an autonomous aerial system. *IEEE Robotics and Automation Letters*, 4(4):3402–3409, 2019.

[147] V. Walter, M. Saska, and A. Franchi. Fast mutual relative localization of uavs using ultraviolet led markers. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1217–1226, 2018.

[148] Viktor Walter, Nicolas Staub, Antonio Franchi, and Martin Saska. Uvdar system for visual relative localization with application to leader–follower formations of multirotor uavs. *IEEE Robotics and Automation Letters*, 4(3):2637–2644, 2019.

[149] Jinkun Wang and Brendan Englot. Autonomous exploration with expectation-maximization. In *Robotics Research*, pages 759–774, Cham, 2020. Springer International Publishing.

[150] Qian Wang, Thomas Weiskircher, and Beshah Ayalew. Hierarchical hybrid predictive control of an autonomous road vehicle. *Dynamic Systems and Control Conference*, 3, 2015.

[151] Barry Brian Werger and Maja J. Matarić. *Broadcast of Local Eligibility for Multi-Target Observation*, pages 347–356. Springer Japan, Tokyo, 2001.

[152] J. M. Winterbottom. On woodland bird parties in northern rhodesia. *Ibis*, 85(4):437–442, 1943.

[153] Guang-Zhong Yang, Jim Bellingham, Pierre E. Dupont, Peer Fischer, Luciano Floridi, Robert Full, Neil Jacobstein, Vijay Kumar, Marcia McNutt, Robert Merrifield, Bradley J. Nelson, Brian Scassellati, Mariarosaria Taddeo, Russell Taylor, Manuela Veloso, Zhong Lin Wang, and Robert Wood. The grand challenges of science robotics. *Science Robotics*, 3(14), 2018.

[154] Hu Yulan, Zhang Qisong, and Xue Pengfei. Study on multi-robot cooperation stalking using finite state machine. *Procedia Engineering*, 29:3502–3506, 2012. 2012 International Workshop on Information and Electronics Engineering.