UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS

JORDÃO NATAL DE OLIVEIRA JÚNIOR

Avaliação Estatística de Interações Dinâmicas Envolvendo Abelhas: Rastreamento Bayesiano e Informação Mútua

São Carlos
2019

JORDÃO NATAL DE OLIVEIRA JÚNIOR

Avaliação Estatística de Interações Dinâmicas Envolvendo Abelhas: Rastreamento Bayesiano e Informação Mútua

Dissertação apresentada à Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Mestre em Ciências – Programa de Pós-Graduação em Engenharia Elétrica.

Orientador: Prof. Dr. Carlos Dias Maciel.

Trata-se da versão corrigida da dissertação. A versão original se encontra disponível na EESC/USP que aloja o Programa de Pós-Graduação de Engenharia Elétrica.

São Carlos

2019

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da EESC/USP com os dados inseridos pelo(a) autor(a).

Oliveira Júnior, Jordão Natal
O48a Avaliação Estatística de Interações Dinâmicas Envolvendo Abelhas Melíferas: Rastreamento Bayesiano e Informação Mútua / Jordão Natal Oliveira Júnior; orientador Carlos Dias Maciel. São Carlos, 2019.

Dissertação (Mestrado) - Programa de Pós-Graduação em Engenharia Elétrica e Área de Concentração em Sistemas Dinâmicos -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2019.

1. Rastreamento de múltiplos alvos. 2. inferência bayesiana. 3. abelhas. 4. agroquímicos. I. Título.

Eduardo Graziosi Silva - CRB - 8/8907

FOLHA DE JULGAMENTO

Candidato: Engenheiro **JORDÃO NATAL DE OLIVEIRA JÚNIOR.**

Título da dissertação: "Avaliação estatística de interações dinâmicas envolvendo abelha: rastreamento bayesiano e informação mútua".

Data da defesa: 25/07/2019.

Comissão Julgadora: **Resultado:**

Prof. Associado **Carlos Dias Maciel** APROVADO
(Orientador)
(Escola de Engenharia de São Carlos/EESC)

Prof. Dr. **Ailton Akira Shinoda** APROVADO
(Universidade Estadual Paulista "Júlio de Mesquita Filho"/UNESP – Ilha Solteira)

Prof. Titular **Jorge Alberto Achcar** APROVADO
(Instituto de Ciências Matemáticas e de Computação/ICMC-USP)

Coordenador do Programa de Pós-Graduação em Engenharia Elétrica:
Prof. Associado **Ivan Nunes da Silva**

Presidente da Comissão de Pós-Graduação:
Prof. Titular **Murilo Araujo Romero**

Resumo

OLIVEIRA JR., JORDÃO NATAL **Avaliação Estatística de Interações Dinâmicas Envolvendo Abelhas: Rastreamento Bayesiano e Informação Mútua**. 2019. 71p. Dissertação de Mestrado - , São Carlos, Brasil, 2019.

Rastrear objetos em vídeo é um método barato para obter informações sobre as partes de um sistema. No entanto, quando há muitos objetos simultaneamente no rastreamento, alguns problemas podem ocorrer, como sobreposição e troca de rótulos, comprometendo a eficiência geral. Recentemente, novas abordagens para resolver estes problemas foram desenvolvidas, por exemplo, Redes Neurais Convulacionais, mas o custo computacional ainda é muito alto. Neste trabalho foi desenvolvido um algoritmo de rastreamento Bayesiano para monitorar objetos em quadros de vídeo. O algoritmo permite a avaliação da Função de Distribuição de Probabilidade (PDF) dos objetos que estão sendo rastreados, combinando o rastreamento com o KDE (Kernel Density Estimation). O algoritmo proposto foi avaliado através de simulação e comparação com abordagens semelhantes, uma vez que as bases de dados convencionais (Princeton Tracking Benchmark) não apresentam semelhança com o problema daquele abordado nesta dissertação. O algoritmo é capaz de rastrear os objetos com grande precisão, sendo capaz de avaliar dinamicamente a entropia e energia, usando coordenadas polares e assumindo uma distribuição de Mises para a previsão de variação de ângulo e uma distribuição não informativa para a predição de raio. Em seguida, com as informações obtidas a partir do algoritmo, foi feita a análise de resiliência abordando os efeitos de dois agroquímicos nas abelhas: o inseticida imidaclopride e o fungicida cerconil. Informações adicionais sobre como elas afetam as abelhas foram obtidas através de Informações Mútuas sobre doses letais, reforçando os resultados anteriores.

Palavras-chave: Rastreamento de múltiplos alvos, inferência bayesiana, abelhas de abelhas, agroquímicos.

Abstract

OLIVEIRA JR., JORDÃO NATAL **Statistical Evaluation of Dynamical Interaction Involving Bees: Bayesian Tracking and Mutual Information**. 2019. 71p. Dissertação de Mestrado - , São Carlos, Brasil, 2019.

Tracking objects in video is a cheap method to obtain information about the parts of a system. However, when there are many objects simultaneously in the tracking some problems can happen, such as overlapping and swap of labels, compromising the overall efficiency. Recently new approaches for solving these problems were developed e.g. Convolutional Neural Networks, but the computational cost is still very high. Here, a Bayesian tracking algorithm to supervise objects on video frames is described. The algorithm allows the evaluation of and Probability Distribution Function (PDF) of the objects being tracked by combining the tracking with the Kernel Density Estimation (KDE). The proposed algorithm was evaluated through simulation and comparison with similar approaches, since the conventional databases (as Princeton Tracking Benchmark) lacks similarity with the problem of the one approached in this dissertation. The algorithm is able to track the objects with great precision, thus being able to dynamically evaluate the entropy and energy, by using polar coordinates and assuming a von Mises distribution for the angle variation prediction and a non-informative distribution for the radius prediction. Then, with the information obtained from the algorithm, a resilience analysis was made approaching the effects of two agrochemicals in the honey bees: the insecticide imidacloprid and the fungicide cerconil. Additional information about how they affect honey bees was obtained via Mutual Information on lethal doses, reinforcing the previous results.

Keywords: Multi-target tracking, Bayesian inference, Bee swarm Honey bees, Agrochemicals.

List of Figures

List of Tables

List of abbreviations and acronyms

KE	Kinetic Energy
PDF	Probability Distribution Function
CDF	Cumulative Distribution Function
KDE	Kernel Density Estimation

Contents

1 Introduction

Bees' participate in nearly 75% of food cultivation through pollination and generate 26 to 65 billion dollars globally (AFFEK, 2018; ISAACS et al., 2017; POTTS et al., 2010; ALLEN-WARDELL et al., 1998). However, their populations are in decline in many regions of the world (BECHER et al., 2018; BRETTELL; MARTIN, 2017; CAMERON et al., 2011). The cause of this decline is still under debate, but agrochemicals are pointed to as the most probable cause of this phenomenon (TOMé et al., 2017; ALLEN-WARDELL et al., 1998). In larger doses, these substances cause death (TOMÉ et al., 2017). On the other hand, when insects come into contact with some substances in lower doses, it influences their behaviour, causing movement far beyond their original condition (THOMPSON, 2003). Animal behaviour analysis uses different techniques to track them, including RF transmitters (MAINWARING et al., 2002; SCHWARTZKOPF-GENSWEIN; HUISMA; MCALLISTER, 1999), GPS (BENNISON et al., 2018; BROWNING et al., 2018), and video tracking (ROSSETTI et al., 2018; CHO et al., 2018; OH; BARR; HURT, 2015; AGUIAR; MENDONÇA; GALHARDO, 2007).

Video tracking is a technique that is constantly under development because it is applied in a wide range of fields (TEKALP, 2015). It is still the cheapest form of tracking (BAZHENOV; KORZUN, 2018), and with the increase in computational power, it has become possible to use more and more sophisticated techniques in image processing (XUE; XU; FENG, 2018; HARE et al., 2016; HENRIQUES et al., 2015). However, it has some drawbacks, such as the identification of several similar objects near each other (PÉREZ-ESCUDERO et al., 2014) and background noise (TEKALP, 2015). These objects can be automated using deep learning or neural networks (FORYS et al., 2018), but these approaches have high computational system costs and contain many elements (ANIL et al., 2018; HU; SHEN; SUN, 2018). The use of animal markings is still significant for distinguishing between objects to be tracked. Other modern approaches to object tracking include hierarchical learned features for tracking (WANG et al., 2015), machine learning (HONG et al., 2015) and cognitive vision (GULER et al., 2018), which all have the same problem: the algorithm is limited to tracking only a few objects at the same time for accuracy. The multi-tracking algorithm faces the challenge (MAGGIO; CAVALLARO, 2011) of not swapping the labels when two objects (for example, animals being tracked) cross. These labels are associated with each animal and must remain the same during video stream analysis.

In the context of an agent changing the operation of a system, as a hive, it is natural to evoke the concept of resilience. It can be defined as the ability of a system to recover itself after an external impact (HOLLING, 1973). However, it is also a quantitative measurement being used in engineering for compute the vulnerability and recoverability of the affected system (DEHGHANIAN et al., 2019; FOTOUHI; MORYADEE; MILLER-HOOKS, 2017; WANG et al., 2016; ARGHANDEH et al., 2016). As a graphical analyses, it needs a physical dynamical quantity to be target of calculations.

The video stream is decomposed into frames in a time span, and these frames are processed to detect the mark colour of each animal. A simple algorithm is used to decide whether or not these marks indicate an animal. In each frame, all colour marks are assigned to a label. These labels are evaluated over time to track all the animals. During this analysis, many labels can become swapped because animals cross each other's pathways, or some of them stop moving (BERNARDIN; STIEFELHAGEN, 2008). Bee movements are usually at a low angle; they keep going, closely following a straight line, and depending on the animal, these variations may or may not be significant (CARTAR; REAL, 1997). Considering this movement to be decomposed in polar coordinates (r, θ), each one can be modelled as a random variable

and its update based on Bayes inference to discern the most probable trajectory. This type of learning prevents the usual swap problem present in video tracking when objects cross their paths.

The proposed algorithm deals with the preceding limitations and provides the theoretical framework to incorporate prior knowledge in this model to track several animals from video streams, modelling their trajectories with a combination of Gaussian distribution for the radius and a von Mises (GATTO; JAMMALAMADAKA, 2007) distribution for the angle. The proposed algorithm starts with uniform non-informative prior distribution functions and the corrections are based on the fact that the bee movement is smooth.

This method was accurate even when prior information was scarce (BOX; TIAO, 2011). The inference is made dynamically, with the PDFs being updated every time; then, the Bayesian algorithm learned the correct distribution for each animal independently. This method is done by associating a PDF for each animal and inferring a distinct PDF for each one of them over the interactive updates. This way, when their paths cross, the algorithm can recognise the correct trajectory of each individual using the accumulated knowledge.

The trajectory validation was done using simulation to understand the general algorithm performance, and the dataset to be analysed is from a bee nucleus. Conventional tracking databases, such as the Princeton Tracking Benchmark[1] or OTB[2], do not provide similar data as the one approached in this dissertation. Also, these results are compared with similar approaches in Ahmed *et al.* (AHMED et al., 2018) and Bozek *et al.* (BOZEK et al., 2018).

Based on the trajectory of the marked bees, the overall nucleus is evaluated for sanity, considering entropy (H) and kinetic energy (KE). The KE is estimated on the bee velocity, and H is based on the PDF estimated via Kernel Density Estimation (KDE) on the evaluation of the Iterative Bayes algorithm. A quantitative measurement of the impact of imidacloprid and cerconil are also calculated using resilience tools, in function of the kinetic energy of contaminated bees compared with a control group. The spatial distribution of the bees are obtained and discussed in therms of metabolic changes from the agrochemicals.

The mutual information between imidacloprid and cerconil was also theme of this dissertation, and it was applied Copula Theory from statistics to obtain the joint distribution from the Cumulative Distribution Functions. The material used for the tests were bees submitted to lethal doses of both agrochemicals.

1.1 Objectives

The objectives of this work are the following:

- identify and track contaminated bees in each video of the supplied material;
- to obtain measurements of average kinetic energy for each contaminant, as well as for control, in order to compare the curves and calculate the resilience that the honey bees develop in each situation;
- evaluate the entropy of the contaminated bees, a second measurement of dynamical evolution under the agrochemicals;
- model the spatial distribution of contaminated bees in order to identify possible states and monitor possible changes metabolic effects caused by agrochemicals;

[1] http://tracking.cs.princeton.edu/eval.php
[2] http://cvlab.hanyang.ac.kr/tracker_benchmark/

- from a different experiment, in which the bees were submitted to lethal doses of imidacloprid and cerconil, evaluated the Mutual Information (MI) between the two agrochemicals.

1.2 Structure of the monograph

This text is divided into six chapters plus a list with bibliographical references. In addition to the introduction, the chapters have the following contents: in Chapter 2 the motivation to work, as well as the main techniques in statistics and signal processing already presented to address the subject.

In chapter 3 the mathematical tools used to solve the problem, ranging from the most basic definitions in image processing, to an introduction to the statistical model to be developed. A discussion of the filmed nuclei is also present, in order to elucidate some of the hypotheses admitted initially.

In chapter 4 the materials to be used are presented in detail, from the video that was processed to the hardware used in the solution. In a second step, and presented the algorithms constructed to find the desired measurements, such as trajectory, kinetic energy and position.

In chapter 5, the results of the processing described in the previous chapter are presented, always followed by a discussion involving them. Finally, in chapter 6, there is a conclusion of the researches done to obtain the Master's degree is written, resuming all the results.

2 Theory

In this section, the mathematical framework used in this work will be described. A few formulas are introduced here to standardise the equation notation through the text. The pre-processing has the objective of identifying the centroids of the bees, and then the Bayesian algorithm get this data and process the labelling.

2.1 Economic importance of bees

The more than 20,000 specimens of existing bees are the main plant pollinators of economic interest in the world; in some cases (1), are the only pollinators. All tables in this chapter were taken from "Plants visited by bees and pollination" (ALMEIDA et al., 2003)

Table 1 – Increase in productivity in presence of bees.

Common name	Scientific name	Increase in productivity (%)
Pumpkin	*Corcubita maxima*	76.9
Coffee	*Coffea arabica*	36.9
Onion	*Allium cepa*	89.3
Apple	*Puros malus* (Wealthy)	75
Apple	*Corcubita maxima* (Johnathan)	94.4
Peach	*Pirus persica*	94
Orange	*Citus cenensis* (Hamlin)	36.3
Orange	*Citus cenensis* (Natal)	15.5

As it can be seen in Tables 1 to 3, bees are a fundamental part of the agriculture scheme of mass production. In fact, some cultivation have a great dependence of bees to be perpetuated (Table 3).

Table 2 – Benefits of bees for some agricultural crops.

Plantation	Increased productivity	Other benefits
Bean	21 % (Free, 1996) No increase (Moreti et al, 1994)	Increase of 18% in the protein content (Moreti et al, 1994).
Sunflower	300% to 600% (Schelotto & Pereyras, 1971)	Increase of 25% in the oil content (Schelotto Pereyras, 1971).
Soy	60 to 230% (Moreti et al, 1998)	

Table 3 – Distance traveled by each bee (second column), average velocity (third column) and standard deviation of the velocities (fourth column)

Discontinue pollinators	Are benefited in several degrees by pollinators	Only produce with pollinators intervention
Banana	Coffee	Fig
Sugar Cane	Beans	Wheatgrass
Leafy Vegetables	Sunflower	Passion Fruit
Manioc	Orange	Melon
Corn	Soy	Cucumber
Wheat	Tomatoes	Chayote

2.2 Statistical modelling importance

Similar problems involving tracking of individual objects, such as bees or people, were approached using primarily neural networks and deterministic reasoning (see (BOZEK et al., 2018) and (AHMED et al., 2018)). However, a more coherent approach necessarily has to include statistical aspects of uncertainties in the system. As said by Pear (PEARL; MACKENZIE, 2018): "In recent years, the most remarkable progress in AI has taken place in an area called *deep learning*, which uses methods like convolutional neural networks. These networks do not follow the rules of probability; they do not deal with uncertainty in a rigorous and transparent way. Still less do they incorporate any explicit representation of the environment in which they operate. Instead, the architecture of the network is left free to evolve on its own. When finished training a new network, the programmer has no idea what computations it is performing or why do they work. If the network fails, the programmer has no idea how to fix it."

A statistical model is, therefore, a mathematical model which compress uncertainties and function that describes the expected value of a given parameter. It has been proven useful in all situation in which one cannot give the exact measurement of variable.

2.3 Entropy and complexity as dynamical measurement

Entropy is a measure largely used in science and engineering (COVER; THOMAS, 2012) and can be defined as the amount of information needed to fully describe a system. Having been first introduced in thermodynamics by Clausius (GREVEN; KELLER; WARNECKE, 2014) and improved by Boltzmann and Gibbs still in nineteenth century (WEHRL, 1978). The concept was generalised by Shannon in the twentieth century (SHANNON, 1948). Today, its applications can be found in biology (BROOKS; WILEY; BROOKS, 1988; MARTINO; MARTINO, 2018; CARO; VALENTINE; WAND, 2018), cosmology, in which it is the center of one of the biggest open problems of science (MALDACENA, 2018; XIAO et al., 2018; BOUSSO, 2018), economics (BOSSOMAIER et al., 2018; GU; XIONG; LI, 2015), engineering (ZEESHAN et al., 2017; ROSTAGHI; AZAMI, 2016; HE et al., 2016) and even linguistics (DEGAETANO-ORTLIEB; TEICH, 2017; REYNAR; RATNAPARKHI, 1997; CAMPBELL, 1982).

With such a wide range of applications, it raises the natural question: what is the difference, if any, among the "entropies" used in each field? It can be observed in several papers a misunderstanding of the meaning of entropy when applied to different areas other than physics and information theory (TAME, 2019; ADAMI, 2016; KOVALEV, 2016; HAYFLICK, 2007; MOROWITZ, 1986). However, sometimes even in these areas the concept is misused (MARTYUSHEV, 2013; HENDERSON, 2003) and university students have many misconceptions about the theme (SOZBILIR, 2003).

There have been attempts to conciliate the entropy of thermodynamics with that of information theory. The most common approach is defining entropy as "disorder" (WRIGHT, 1970; SCHRODINGER, 1968), something that is introduced as soon as the high school for thermodynamics. However, if it has, at first, a didactic appeal, it is not a good analogy since "order" is a subjective human concept and "disorder" is not the measurement that can be obtained always with entropy (SOUBANE et al., 2018).

Since the concept of entropy is key in this dissertation, more information about it can be found in the Appendix.

2.4 Image pre-processing

An introduction to the essentials of image processing, used in this dissertation, is presented.

2.4.1 Fundamentals

A digital image is defined mathematically as a matrix $m\ x\ n$, where a colour code associates each position of the matrix with a different colour. Thus, a binary image consists of only two states, corresponding to a matrix in which the inputs are 0 and 1, respectively "black" and "white". Greyscale images correspond to 8-bit entries, again 0 totally black and 255 all white, with all integers between 0 and 255 corresponding to different levels of grey. Color images correspond to matrices whose entries have 24 bits, 8 bits for each colour matrix, red, green or blue, resulting in the RGB code. In this case, the primary colours are mixed until the desired colour simulation of a real colour is reached, with $255^3 = 16581375$ of different shades (Fig. 1).

Figure 1 – Same image displayed in three different modes: RGB (24 bits), grey levels (8 bits) and binary (1 bits). The greyscale image is converted to binary using a threshold corresponding to the interval [0, 255], whereby matrix values below the threshold become valued 0 and values above the threshold become valued 1

(a) RGB Image. (b) Greyscale. (c) Binary image.

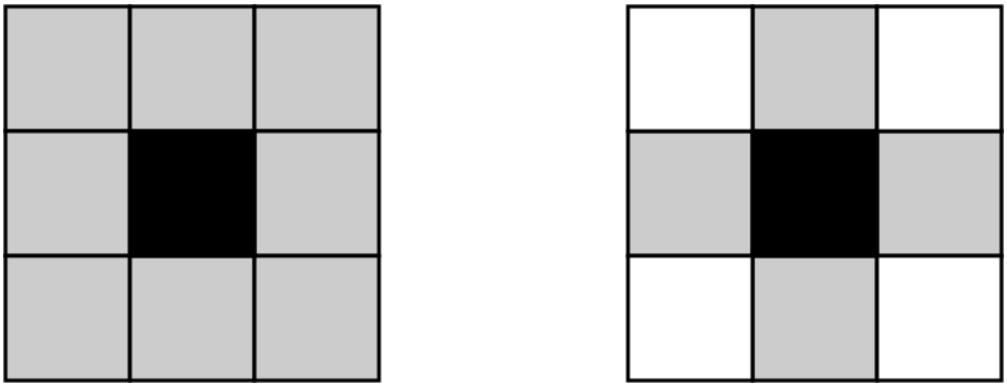
It is possible to define the intensity function $f(i, j)$ as the function $f(i, j)$ that associates each position (i, j) of the matrix corresponding to the pixel of the image with an integer. It is called, therefore, the intensity function, the function

$$f(i,j) = \begin{cases} [0,1] & \text{if the image is binary} \\ [0,255] & \text{if the image is 8 bits (grey levels)} \\ [0,255]^3 & \text{if the image is 24-bit (RGB)} \end{cases} \tag{2.1}$$

One way of identifying objects in an image is to separate them from the background if they have different colours. Once the image is processed as an array, it can be traversed and the pixels can be identified with the colours of interest, separated from the background using a loop, and converted the RGB matrix into a binary array in which pixels are worth 1 if they match a pixel of the object. This is done by identifying distinct objects by connectivity criteria between pixels: when a pixel with the desired colour is found, it is checked if any of the adjacent pixels have the same colour; while there are pixels that

meet this requirement, it is said to be the same object (figure 2). Once the whole object is obtained, the center of mass of the object is calculated (equation 2.3), a reference measure of the spatial location.

Figure 2 – Connectivity between pixels (grey pixels relative to the center black pixel): 8-connected (left) and 4-connected (right) regions. Given any pixel of an object, all of the 8-connected region are considered to belong to the same object in this work.



The center of mass of an object in an image is defined as the ratio between the corresponding moments for the axes x e *and*. The general expression for the moment of order $p + q$ is:

$$M_{pq} = \sum_{x} \sum_{y} x^p y^q f(x, y) \tag{2.2}$$

and the coordinates of the center of mass are defined as

$$(x_c, y_c) = \left(\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}} \right) \tag{2.3}$$

Figure 3 – Example of identification of objects by connectivity between pixels: in this case, the criterion of 4-connected regions is used (neighbours diagonal pixels are not considered part of the same object). If the criterion were 8-connected regions, all pixels 1 would be considered part of the same object.

0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0
0	0	1	1	1	0	1	0	0
0	0	1	0	1	0	0	0	0
0	0	1	1	0	1	0	0	0
0	0	0	0	1	1	1	0	0
0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

2.4.2 Application in the material

Fig. 4 (a) illustrates the objects identification and Fig. 4 (b) shows a region of interest (ROI) for a frame where there are white and yellow marked objects with the background (the experimental apparatus

is from a hive). Each pixel is separated based on its RGB value; if its value is in the colour range desired for the tracking, then it is turned white; otherwise this pixel is turned to black. The result from this image processing is shown in Fig. 4 (b). With the objects defined, their positions can be evaluated calculating its center of mass.

After the identification, the processed frames were traversed from top to bottom and from left to right, and each object of the swarm is labelled in order of occurrence. Since they are constantly moving, their identification (or labels) change on each frame. If in the first frame, an element is labelled as A, in the next frame, due to their movement, it can be labelled as B. The tracked objects labels are swapped, and this must be corrected to obtain their trajectory. The label that appears first must be conserved until the end of the video stream.

However, even limiting the radius of displacement of the objects, there can be more than one possible position in the next instant due to near bees. Bayesian inference was used to solve this issue: with a *prior* information about the path travelled by the individuals, the next correct point of the trajectory can be calculated to maximise the likelihood of the *prior* information, giving a *posterior* estimation for the variable (MARITZ; LWIN, 2018).

Figure 4 – Example of the image processing algorithm for the experimental case of the bees. (a) An original frame from the hive (zoomed). (b) The result of the processing. White regions correspond to yellow marked bees. (a) Original image. (b) Result of the image processing.

2.5 Labelling Process, Bayesian inference and *Maximum a Posterior* (MAP)

The movement of the objects to be tracked on each frame was realised in polar coordinates with radius (r), and angle (θ); both were assumed to be random variables. For the correct labelling of each object movement, it was used Bayesian inference to indicate the most probable displacement of marked bees on each frame. This method uses Bayes' theorem, Eq. 2.4, to update the probability of a hypothesis as more information becomes available. Bayesian updating is important in the dynamic analysis of a time series (HAMILTON, 1994).

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2.4}$$

Here, $P(A)$ and $P(B)$ are the probabilities of observing the events A and B independently; $P(A|B)$ is the probability of the event A, given that B happened.

In many applications, the event B is fixed, and it represents the impact on our belief in the occurrence of events A. In this case, the denominator $P(B)$, is fixed; what we want to vary is A. Bayes' theorem then shows that the posterior probabilities are proportional to the numerator: $P(A \mid B) \propto P(B \mid A) \cdot P(A)$. Also, the Bayes' Theorem is valid to Probability Distribution Function (PDF) (BOX; TIAO, 2011) and is stated as follows: let $f(x|\alpha)$ be the PDF of a random variable X (here, r or θ) with parameter α and samples x. Let $\pi(\alpha)$ be the *prior* PDF for α. Then, the *posterior* distribution for α given X, $\pi(\alpha)$, is defined as (BOX; TIAO, 2011):

$$\pi(\alpha|x) = \frac{f(x|\alpha)\pi(\alpha)}{\int f(x|\alpha)\pi(\alpha)dx} \tag{2.5}$$

The estimation of the parameter α (that is not observed) can be done via the Maximum a Posterior (MAP) method. Given the observed values of X that generates a PDF $f(x|\alpha)$, the MAP is the maximum value of the *posterior* function:

$$\hat{\alpha}_{\mathrm{MAP}}(x) = \arg\max_{\theta} \ \pi(\alpha|x) \tag{2.6}$$

and $\pi(\alpha|x)$ is calculated using Eq. 2.5.

There were used polar coordinates to modelling the movement of the particles. For the swept angle θ, the natural choice as *prior* was the von Mises distribution (RISKEN, 1996), since it is the circular analogue of the normal distribution. It is defined as follows:

$$f(x|\mu,\kappa) = \frac{e^{\kappa cos(x-\mu)}}{2\pi I_0(\kappa)} \tag{2.7}$$

where $I_0(x) = \frac{1}{\pi}\int_0^{\pi} e^{cos(xcos(\theta))}d\theta$ is the modified Bessel function of order 0. In Eq. 2.7, μ is the mean of the distribution and κ defines the curve's format (Fig. 5).

For the radius, a uniform non-informative *prior* was chose. The algorithm has its maximum efficiency when $r \sim \lim_{n\to 0} U(0,n)$[1]. In each iteration, a new vector $v_i = (r_i\cos\theta, r_i\sin\theta_i)$ will be generated (and the null vector is possible) so that the j-th particle, after all iterations, have a trajectory given by $\bigcup_{i=1}^{\infty} v_i$

[1] Of course the discrete calculations performed in real world problems with image inputs are limited by frame-rate, or absence of necessity of such many iterations to realistically modelling a system.

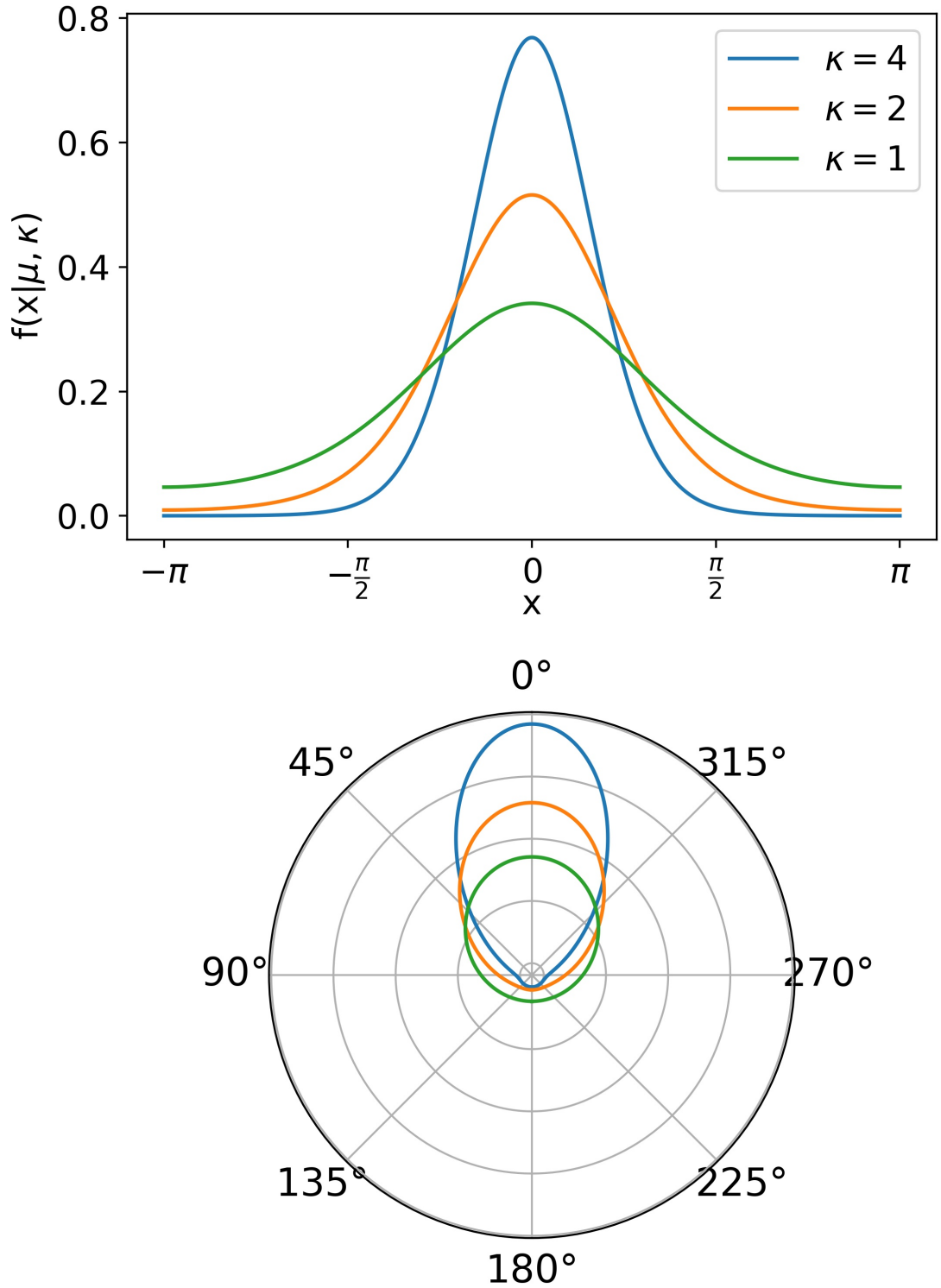


Figure 5 – Comparison of the von Mises PDF for $\mu = 0$ and different values of κ. (a) If the angle variation is smooth, a good *prior* for the variable is $\kappa = 4$. The Bayesian algorithm will constantly update the PDF, and at the end of the process, it should give rise to the best von Mises that estimates the angular behaviour for each object being tracked. (b) Polar distribution of the probability for each value of κ.

2.6 Dynamical evaluation

The Shannon entropy is used in this study to evaluate the randomness of the tracked object, and thus infer about the movement of them in space. The Shannon's entropy (SHANNON, 1948) H of an event X, with discrete PDF is given by:

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x) \tag{2.8}$$

the unit of H is *bits*. One classical interpretation of its meaning is: the entropy of an event with n distinct outcomes (given by the variable $X = (x_1, \ldots, x_n)$) measures how much information is needed to describe the entire space state. In practice, it means that the higher the entropy, less predictable is the event, and events that are certain have zero entropy. The equivalent expression to continuous distributions is given by the differential entropy:

$$h(f) = -\int_{x \in X} f(x) \log_2 f(x) dx \tag{2.9}$$

This measurement is a generalisation of the Shannon entropy, and since the PDF can be greater than 1, even though negative entropy is difficult to attribute physical meaning. Computational computations of continuous distributions of probability inevitably go through quantization processes. Let the continuous random variable X, with distribution $f(x)$. Because, by constructing the ordered body of real numbers (LAM, 1983), $P(X = x) = 0$ for all x, the probability of continuous events can only be evaluated in a given interval. Therefore, any interval would need infinite bits to be fully described. However, by dividing *bins* (ASH, 1990) of size Δ, there exists a x_i within each interval (*bin*) such that by the Mean Value Theorem for integrals (GOLDFELD; HOFFSTEIN, 1985), one can write

$$f(x_i)\Delta = \int_{i\Delta}^{(i+1)\Delta} f(x) dx \tag{2.10}$$

Now taking the random variable $X^\Delta = x_i$ if $i\Delta \leq X \leq (i+1)\Delta$,

$$p_i = \int_{i\Delta}^{(i+1)\Delta} f(x) dx = f(x_i)\Delta \tag{2.11}$$

Therefore the entropy of the quantized variable is

$$H(X^\Delta) = -\lim_{N\to\infty} \sum_{i=-N}^{N} p_i \log p_i \tag{2.12}$$

$$= -\lim_{N\to\infty} \sum_{i=-N}^{N} f(x_i)\Delta \log(f(x_i)\Delta) \tag{2.13}$$

$$= -\lim_{N\to\infty} \sum_{i=-N}^{N} \Delta f(x_i) \log f(x_i) - \lim_{N\to\infty} \sum_{i=-N}^{N} f(x_i)\Delta \log \Delta \tag{2.14}$$

$$= -\lim_{N\to\infty} \sum_{i=-N}^{N} \Delta f(x_i) \log f(x_i) - \log \Delta \tag{2.15}$$

Theorem: If the density f(x) of the random variable X is Riemman integrable[2], then

$$H(X^{\Delta}) + \log \Delta \rightarrow h(f) \tag{2.16}$$

This solves the asymptotic case. A more practical formulation for estimate the entropy of a continuous random variable is the Jaynes correction (JAYNES, 1963; JAYNES, 1968) based on the limiting density of discrete points (LDDP). Let $m(x)$ be the *invariant measurement*; then

$$\lim_{N\rightarrow\infty} \frac{1}{N} \left(\#C[a,b]\right) = \int_a^b m(x)\, dx \tag{2.17}$$

in which $\#C[a,b]$ denotes the cardinality of C, or the number of discrete elements in the set ranging from a to b. By Jaynes definition,

$$H(X) = -\int p(x) \log \frac{p(x)}{m(x)} dx \tag{2.18}$$

In the case that $m(x)$ is constant in some size range r, where $p(x)$ is essentially zero outside this range, then the discrete point boundary density is closely related to the differential entropy $h(f)$, and the Shannon entropy can be approximated as

$$H(X) \approx \log(N) - \log(r) + h(f) \tag{2.19}$$

Since usually $N \gg r$, the Jaynes correction tries to ensure that the entropy is always positive, surpassing the difficulties of defining the meaning of the generalisation of discrete entropy in the negative cases.

The entropy of a system can provide insights into the nature of the evaluated variables since it can be interpreted as a measurement of randomness of the variable (SHANNON, 1948). In the case of motion analysis, greater entropy indicates that path travelled by the object is less predictable. A discussion of the degrees of freedom in biological systems can be found in Popovic *et al.* (POPOVIC, 2017). A similar calculation of entropy given the energy of a system can be found in Meirovitch *et al.* (MEIROVITCH, 2007); however, the scale used there is molecular, and here, the level is macroscopic.

The KE is a measure of how much tracked objects are moving, and it is given by the formula $KE = mv^2/2$. Therefore, KE is proportional to the velocity squared. Lower energy indicates lower entropy, since the movement become less random and can have important insights in different real systems, as the bees in the application. A lower KE indicates that a bee spends more time not moving, and shows, together with entropy, abnormal behaviour of the insect (BLACQUIERE et al., 2012; MULLIN et al., 2010).

2.7 Kernel Density Estimation (KDE)

This mathematical method was used to generate the continuous PDF of the empirical distributions for the angles and radius travelled by each bee, allowing for the entropy calculation. The KDE is a non-

[2] In fact, modern approaches of differential entropy usually requires the function to be integrable in the sense of Lebesgue, since it can deal with an infinite amount of discontinuities (BARTLE; BARTLE, 1995).

parametric algorithm to find a continuous better estimate of PDF from discretely observed data. As can be found in Terrell *et al.* (TERRELL; SCOTT, 1992), the estimated PDF is given by

$$\hat{f}(\mathbf{y}) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x_i - \mathbf{y}}{h}\right) \tag{2.20}$$

In equation 2.20, $\mathbf{y}$ is the variable of the domain, x_i are the observed data, K is an interpolation function generally chosen as a Gaussian. $K : R^d \to R$ is a function centred in 0 that integrates to 1, and h is a smoothing parameter that would usually tend to 0 as the sample size n tends to infinite. Several algorithms are implemented to increase the performance of the KDE (CACOULLOS, 1966; PARZEN, 1962; ROSENBLATT, 1956). The one that will be used here is described by Loftsgaarden *et al.* (LOFTSGAARDEN; QUESENBERRY et al., 1965) is based in the k-nearest neighbours algorithm. It consists in, for any point of a set, its class will be assigned comparing the classes of its k nearest points. In a ball centred at x, the ratio of observations is k/n, and the KDE is given by:

$$\hat{f}(\mathbf{y}) = \frac{k}{nV_d h_k(\mathbf{y})^d} \tag{2.21}$$

In equation 2.21, $h_k(\mathbf{y}^d)$ is the Euclidean distance between $\mathbf{y}$ and the k-th nearest neighbour, and V_d is the volume of the unit sphere centered in the origin of the R^d space. If K is an uniform density on the unit d-sphere S_d; then

$$\hat{f}(\mathbf{y}) = \frac{k}{nh_k(\mathbf{y})^d} \sum_{i=1}^{n} K\left(\frac{x_i - \mathbf{y}}{h_k(\mathbf{y})}\right) \tag{2.22}$$

The objective of the KDE algorithm in this dissertation is to obtain a continuous PDF for the moving objects, since it is know that their movement is continuous, but the discretisation makes the calculations return discrete histograms.

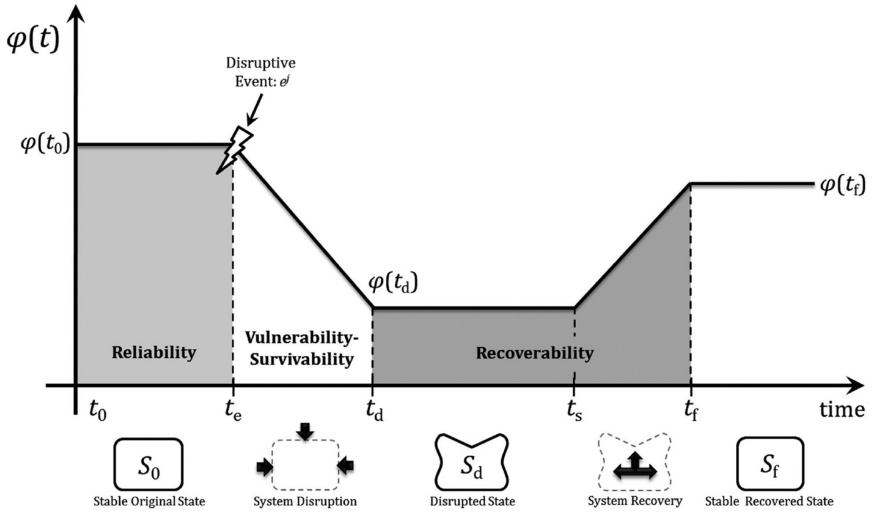


Figure 6 – Parameters used in the resilience analysis of a system (from (DESSAVRE; RAMIREZ-MARQUEZ; BARKER, 2016)). The reliability is the area bellow the curve in which the system is operating in its normal condition; the vulnerability is the area of the interval between the interference of an external factor and the stabilization in the lower level; the recoverability is the area bellow the curve in which the system goes from the lower level state and recover itself to an upper state, that can be the initial state or not.

2.8 Resilience of a system

Resilience is a concept present in several areas of knowledge (THORÉN; OLSSON, 2018). In this text, however, its definition will be the same as in physical systems (ALEXANDER, 2013): resilience is the ability of a system to recover itself after an external perturbation and adapt after the recovery, which can be total or partial. This approach is being used to model the vulnerability of power systems under extreme climate conditions (WANG et al., 2016; PANTELI; MANCARELLA, 2015; OUYANG; DUENAS-OSORIO, 2014).

Fig. 6 represents the stages of a system under resilience analyses: a physical quantity ($\phi(t)$) is stable in the original state S_0, until an external event e^j changes the operation of the system. The quantity being measured drops its value until a new state S_d, and the area under the curve between these two instants is called "vulnerability". The system stabilizes in $\phi(t_d)$ and, after some time, tries to recover itself to the initial state, stopping its value in $\phi(t_f)$, that can be bellow, the same or above the initial state, and, for this reason, the area under the curve between the instants t_d and t_f is called "recoverability".

2.9 Mutual information

Mutual information $I(X;Y)$ between the random discrete variables X and Y is a measure of information theory (SHANNON, 1948) for the distance, in the sense of Kullback-Leibler (KULLBACK; LEIBLER, 1951) of the joint distribution $p(x,y)$ and the product $p(x)p(y)$, and can be defined as

$$I(X;Y) = \sum_{x\in X}\sum_{y\in Y} p(x,y)\log_2\frac{p(x,y)}{p(x)p(y)} \tag{2.23}$$

Mutual information quantifies the amount of information in the sense of Shannon entropy (in bits) obtained about one random variable through observing the other random variable. Here, it will be used to calculate how a mixture of agrochemicals affects honey bees in comparison with individual applications.

3 Materials and Methods

3.1 Exposure to sub-lethal doses

Bees of up to one day of life of the *A. mellifera* species were captured in four colonies already established in the experimental apiary of the Federal University of Viçosa (UFV, Viçosa, MG, Brazil) to perform the bio-assays. The pronotum of each bee was stained with ink so that it was possible to identify the treated individuals, and for each treatment a colour was defined. The bees were kept under controlled conditions, similar to those found in the colony (34 ± 2, $70 \pm 10\%$ relative humidity); packed in transparent plastic bottles (non-toxic polypropylene) of 500 mL with perforations in the lid; fed with distilled water and sucrose solution (50% v / v) and kept in the dark until the experiments were carried out.

Four replicates were performed per treatment, with each replicate corresponding to one colony. The bees were fasted for one hour prior to submission to treatments equivalent to CL 30 of each product (imidacloprid and cerconil). The fasting period before exposure to pesticides was necessary to standardize diet consumption by the bees tested. In addition, a control group was fed only with distilled water and sucrose solution (50% v / v), being available for five hours under controlled conditions (34 ± 2, $70 \pm 10\%$ RH). After this period, a total of 200 bees tagged and exposed to each treatment were reintroduced into colonies established in observation nuclei. Each nucleus received only bees from a single treatment. The two selected pesticides are products widely used in commercial crops in Brazil: the neonicotinoid insecticide imidacloprid [Evidence ® 700 WG, 700 g active ingredient (a.i.) / L, water-dispersible granules; Bayer CropScience Ltda, São Paulo, SP, Brazil] and the fungicide cerconil [Cerconil WP ®, 700 g i.a./Kg, wettable powder (WP); Iharabras S.A. Chemical Industries, Sorocaba, SP, Brazil]. The treatments were corresponding concentrations of imidacloprid $8.96 \cdot 10^{-4}$ g ai / mL and for cerconil $4.35 \cdot 10^{-4}$ g ai / mL, which is equivalent to 30 each compound LC established in studies by tome2017agrochemical.

Four colonies were established in observation nuclei, consisting of a wooden frame (52.5 x 43.5 x 5.5 cm) with two glasses on both sides and two frames (a frame of honey and a frame of creation). The observation cores were placed inside a protected shed and connected to the outside through tunnels of transparent acrylics. Filming was performed in three stages between May and June 2017. The treatments were alternated so that each colony received different treatments (imidacloprid, cerconil and control) at different periods, totaling four replicates for each treatment. After the reintroduction of the bees, treated and control, in the colonies established in the observation nuclei, 10 minutes were filmed on both sides of the nuclei simultaneously, during the first five days and one to the tenth day.

3.2 Exposure to lethal doses

The assays being described here are the same described in (TOMÉ et al., 2017). Twenty bees of 4 different nucleus were feed, separately and mixed, with different doses of the pesticides “diluted in honey-based syrup solution (50%, v/v) and offered to bees in 2 mL Eppendorf tubes inserted into low-density plastic containers (250 mL). Each plastic container was used as an experimental unit containing 20 forager bees fed on 1 mL of pesticide-contaminated honey solution” (TOMÉ et al., 2017). As it was discovered, the mixture of two agrochemicals multiply their lethally in roughly 1000 times. The mutual information was applied to measure how much information (in the Shannon sense) is shared between the two agrochemicals. To obtain the mutual information by Eq. 8, it is necessary to first estimate the joint probability. With the data from tome2017agrochemical, it is possible to generate the Cumulative

Figure 7 – Example of experimental apparatus (nucleus) used to record the bees behaviour. This image shows imidacloprid exposed bees. Similar set up was used for cerconil exposed bees.

Distribution Functions (CDF), i.e. the rate of dead individuals by each agrochemical, in function of their concentration. With the CDF $F(x)$, it is possible to generate the Probability Density Function (PDF) $f(x)$, using Eq. 3.1.

$$f(x) = \frac{dF(x)}{dx} \tag{3.1}$$

The marginal PDFs ($p(x)$ for imidacloprid and $p(y)$ for cerconil, for example) are therefore easy to find. The joint distribution $p(x, y)$ is harder, since there are correlations between the random variables X and Y that requires a mapping to generate their intersection. Copula Theory was used to surpass this difficulty.

Copulas are joint distribution of random variables $U = U_1, ..., U_n$, such as $(U_1, ..., U_n) \sim U(0, 1)$ (DANAHER; SMITH, 2011). The joint cumulative density and the copula function are related by Sklar's Theorem (LO, 2018) in Eq. 3.2

$$\frac{p(x, y)}{p(x)p(y)} = c(P(x), P(y)) \tag{3.2}$$

in which c is the PDF of the copula:

$$c(u_1, u_2) = \frac{\partial^2}{\partial u_1 \partial u_2} C(u_1, u_2) \tag{3.3}$$

Using 100,000 points in the interval [0, 0.07] makes the approximation of the partial derivative for the discrete function C very precise. The copulas were used, therefore, to obtain the joint distribution function for the mixture of agrochemicals, and then, calculate their mutual information. Since the later calculation needs the joint distribution, this task would be impossible without the application of the Copula Theory.

3.3 Synthetic video simulation

Since the conventional databases could not provide material for quality test of the algorithm, it was tested in a simulation of a particle swarm. A sequence of frames with several dimensions with 200, 300 and 600 randomly distributed circular objects moving through the area, during 500 seconds was generated using *openCV*. The PDFs of the angle swept by the objects was defined as a von Mises, with $\mu = 0$ and $\kappa = 4$. The radius of movement was given by a non-informative Uniform distribution ($r \sim U(0, 10)$). For each configuration, number of objects/background area, the simulation was executed a thousand times. Then, the radius was changed to verify how it affects the quality of the tracking, and identify the problems involved.

The simulation behaves exactly like the marked bees of the previous section: there were chosen n objects to take place in a set of frames of same dimensions of a Full HD video, i.e., 1080 pixels in the vertical direction, and 1920 in the horizontal direction, since the objective is to apply the method for real life videos of particle swarms.

3.4 Bayesian inference and dynamical evaluation

With the position of the bees stored, the next step is to trace their correct trajectory. Here we are faced with two problems: choosing the right place and the overlapping issue. The latter is caused by the fact that some bees are overlapped by its partners and therefore are not visible for some time. To surpass this hindrance, it is assumed that the bee does not move while overlapped by others, maintaining their positions. This was empirically confirmed via visual inspection.

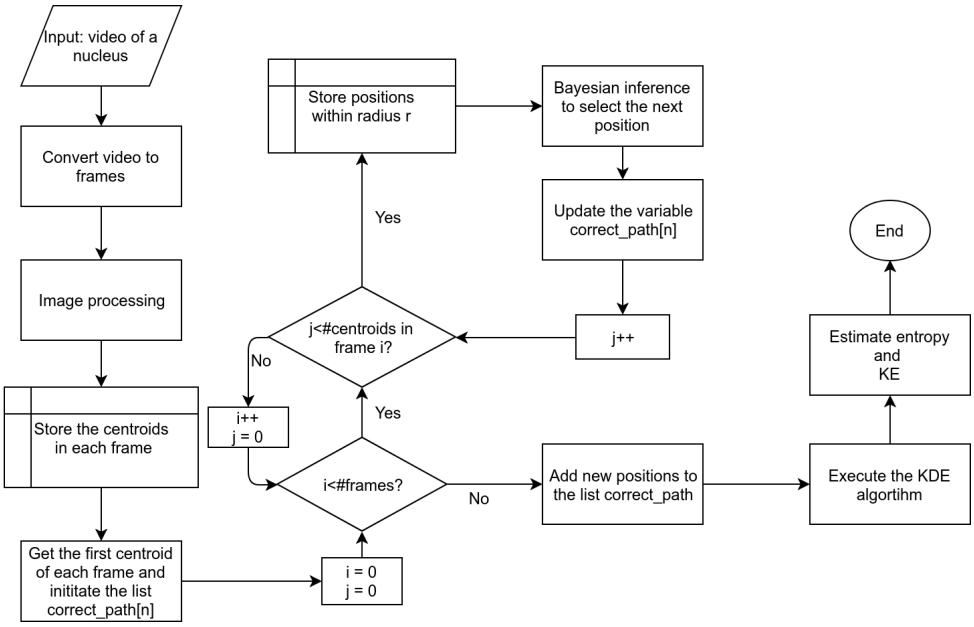


Figure 8 – Fluxogram of the algorithm: the data consists of videos, that are divided in frames and processed, obtain the centroids in each second. The variable *correct_path* is initialized with the size of the number of elements in the first frame. The frames are traversed adding new elements and using Bayes Theorem to correct the path of the bees. With the positions, the KDE algorithm is executed and allows to estimate the entropy, while the trajectories allows for the estimation of the KE.

It was adopted that the right next position of the bee's trajectory tends to be smooth. It corresponds to a *prior* for the angle between two consecutive points is small or close to zero. The PDF for

the angle is assumed to be a von Mises with $\mu = 0$ and $\kappa = 4$ (Fig. 2.7). The inference is calculated for each bee in each frame as follows: it is evaluated the angle variation between the possible future positions of the bee and the current one. Then, the next position is chosen to be the one to maximise the value of $f(x|\mu, \kappa)$. The mean μ is updated as the weighted average between the current mean and the angular variation are discovered, and κ is modelled as $\pi(\kappa) \propto U(0, 20)$. Then, Eq. 2.7 is applied to the *prior* to generate the *posterior* for κ. The inference process delimits an area around the bee being tracked at each moment, with radius d equals to the maximum radius in the *prior* for the radius. Then, it chooses, between the bees inside the radius in the next frame, the one which returns the Maximum Posterior for the von Mises of the angle.

At the end of the process, it is generated a von Mises PDF for the angular displacement for every bee, as well as a distribution for the radius. Then, the KDE algorithm takes the empirical data of the angular displacement and the translational displacement, using the corrected path of each bee, and generates a continuous PDF for them. Finally, the MAP is calculated, with the *prior* assumed to be the von Mises of the last session for the angle.

In the process of returning the PDF for angle and radius, it also becomes possible to calculate the entropy of the movement of the contaminated bees in different days. Since the expected action of the agrochemicals is to change the metabolism of the insects (BLACQUIERE et al., 2012; MULLIN et al., 2010), this entropy should decrease with time[1]. The entire methodology is illustrated in the Fig. 9

[1] The entropy being estimated here is not the thermodynamical entropy, which should increase for the contaminated bees, because they are dying and tending to balance with the environment (EVANS, 1963). The entropy measured in molecular scale certainly would grow (SCHNEIDER; KAY, 1994).

4 Results

4.1 Sub-lethal doses

4.1.1 Spatial distribution

The distribution of bees in the nucleus over time was obtained by simple quantification of the number of centroids in each region.

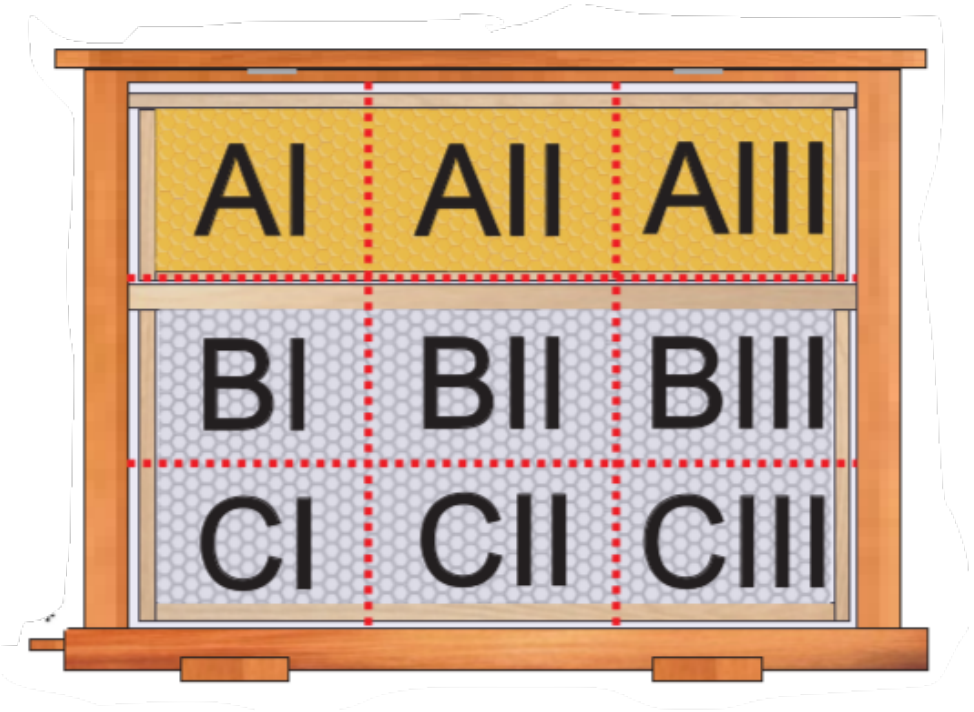


Figure 9 – Regions of the nucleus, as indicated by the specialists of the UFV. The upper and yellow regions are called honey frames, and the bottom and grey regions are called reproduction frames.

As expected, there is a natural tendency for the contaminated bees to rapidly spread out from the center to the boards, since they are “ageing” faster (day 1 for control is missing).

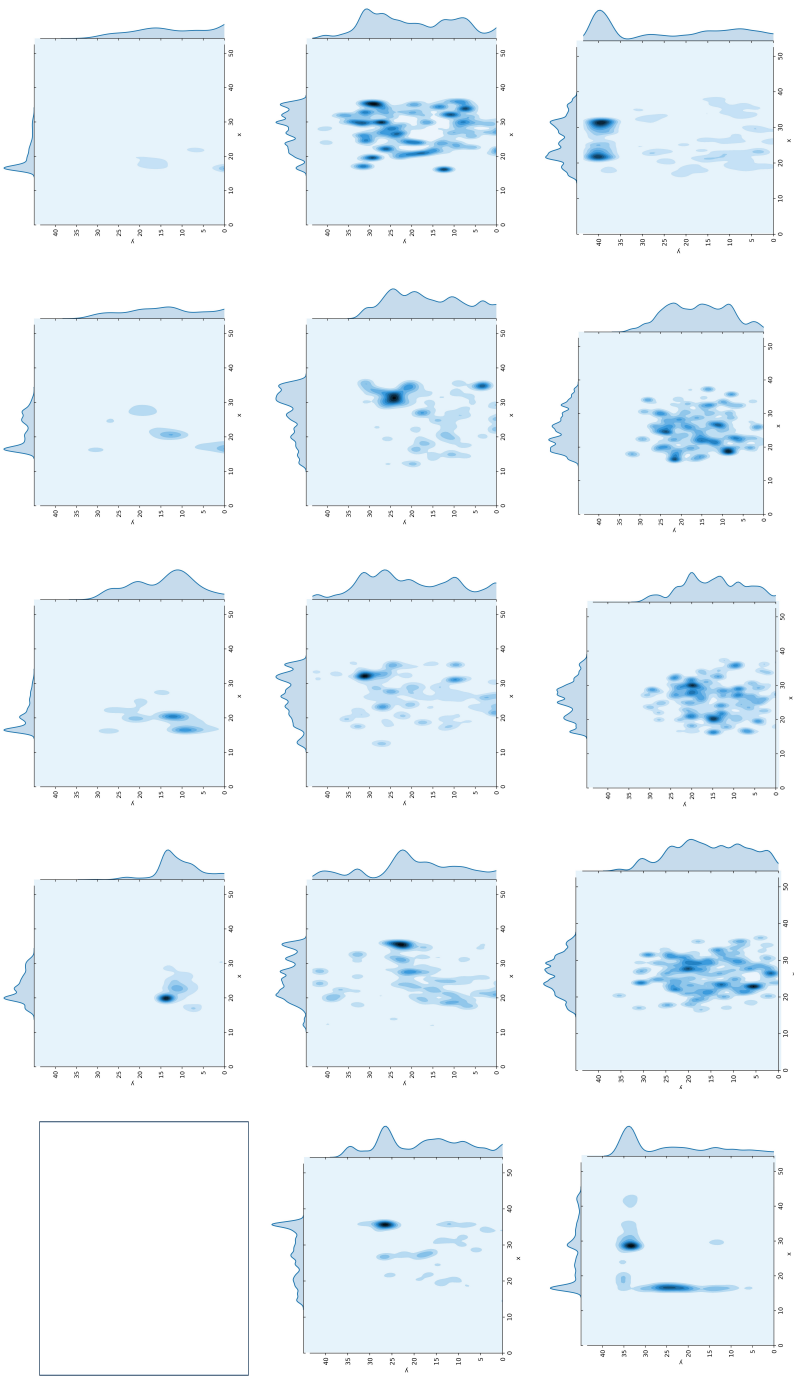


Figure 10 – Spatial evolution of bees distribution. From above to bellow, the heat-maps represents the distribution of the control group, the cerconil group and the imidacloprid group. From left to right, the samples are from the days 1,2,3,5 and 10 (Day 1 is missing for the control). The normal behaviour is a concentration in the center in the first days, and with time, the distribution spreads out to the borders. The contaminated bees spreads out faster than the control ones, demonstrating the effect of the agrochemicals in their metabolism, making them "age" faster

4.1.2 Synthetic video simulation

The efficiency of the Bayesian algorithm was measured in the simulation. With $r \sim U(0, 50)$, the results were fast, but hit rate was low, as can be seen in Table 7. This choice for the radius was taking in account a division of 1 frame per second. However, video tracking is made by equipment capable of filming with at least 30 frames per second, and, increasing the number of frames should increase the hit rate of the algorithm, as well as the execution time.

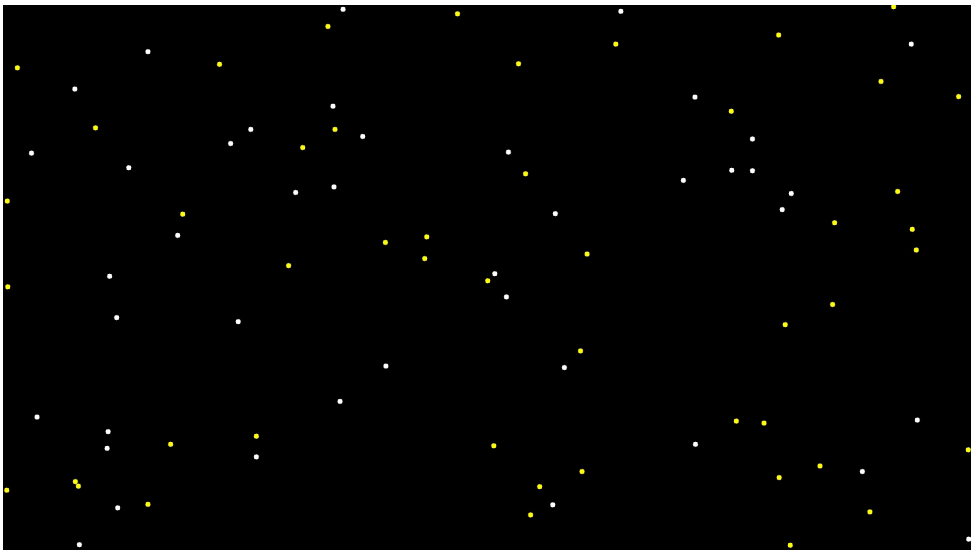


Figure 11 – A frame from the video simulation. The objects generated are points with radius of 5 pixels and with colours white and yellow, and they move randomly in the background.

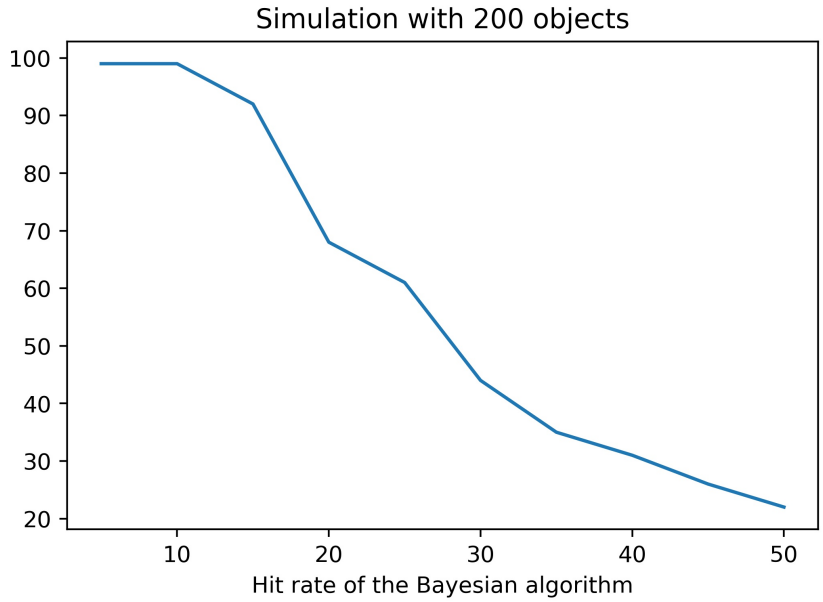


Figure 12 – Efficiency of the algorithm, for a simulation with 200 objects, in function of the maximum radius of the *prior*.

Table 6 – Efficiency of the geometric algorithm in the simulation. The problems which appear in a real nucleus (overlapping and rotation) are also treated here.

Objects simulated	Trajectory hit rate (%)	Min/Max (%)
100	96.2	(92.3/100)
150	88.7	(82.0/93.1)
200	66.7	(48.6/86.9)

Table 4 – Efficiency of the Bayesian algorithm in the simulation, for $r \sim U(0, 50)$ and dimensions 1080x1920.

Objects simulated	Trajectory hit rate (%)	Execution time (s)
200	26.1	13.7
300	19.9	15.4
600	12.4	48.2

The main reason for the difference of times is that, as can be seen in Eq.2.5, an integral needs to be calculated each iteration, for each bee. Changing the radius increases the number of iterations (Fig. 9)

Table 5 – Efficiency of the Bayesian algorithm in the simulation, for $r \sim U(0, 10)$ and dimensions 1080x1920.

Objects simulated	Trajectory hit rate (%)	Execution time (s)
200	99.8	50.0
300	99.8	76.7
600	97.6	288.1

4.1.3 Comparison with the unsupervised geometric algorithm

The same simulation was used as input to an geometric algorithm, similar to the one described by Ahmed *et al* (AHMED et al., 2018): the possible bees in the frame $n + 1$ connect to the correspondent one in the frame n if they are inside a circumference with fixed radius (here, 30 pixels) and their segment forms the smallest angle from the ones inside the circumference. The main difference of the test is that the objects are assumed to be the same size, and the estimation of Eq. 4 in Ahmed *et al.* is not used.

The reason for the inferior hit rate in this geometric approach is the fact that the algorithm does not learn from the upcoming data, and the model is nearly static, keeping its *prior* assumptions from the beginning to end.

4.1.4 Dynamical measurements

The KDE calculation using the *scipy* library has a parameter *bw*, that, when not assigned (assumed infinity) returns a smooth approximation of the PDF. When the *bw* is decreased, the estimate is smoother, but, if the value is too small, it overfits the histogram. The smooth solutions have the inconvenience of having a negative residual part, even when the histogram doesn't have one (Fig. 13). To solve this problem, the negative part in Fig. 13 was manually transported to the positive side, taking the image of the negative x-axis and concatenating with the positive values.

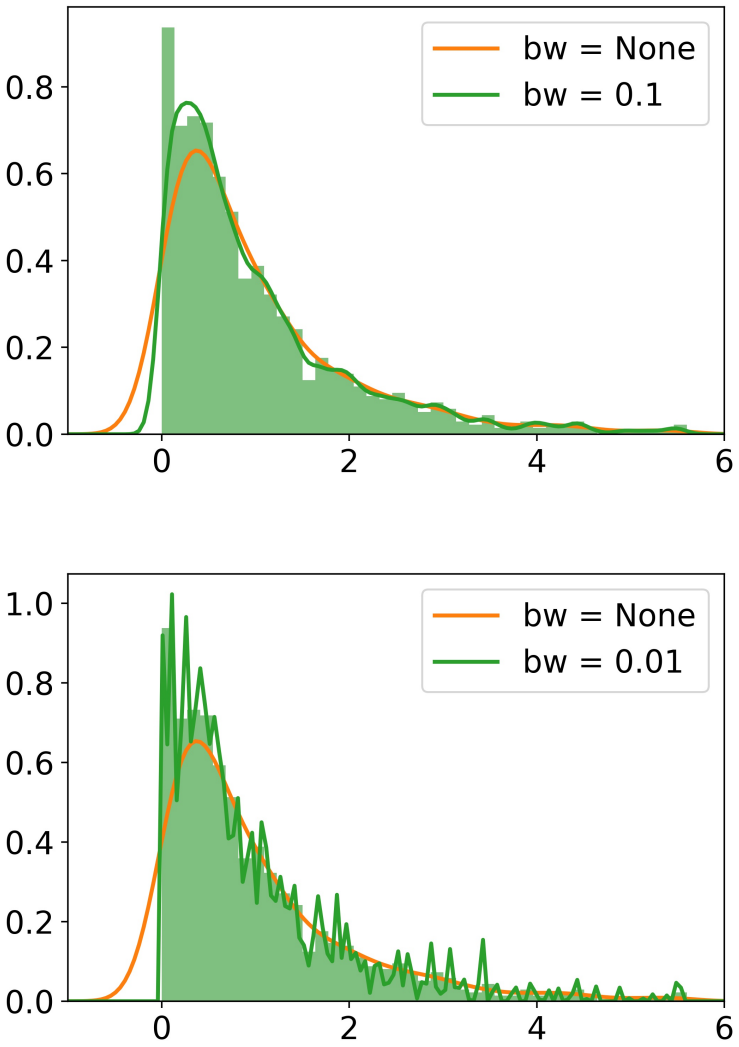


Figure 13 – Example of the KDE algorithm taking a histogram and converting it in PDFs. The parameter *bw* regulates the fitting of the resulting curve. With small values, the smoothness increases, but for very small *bw*, overfitting can happen. (a) KDE for *bw* not asserted (*None*) and $bw = 0.1$. (b) KDE for $bw = 0.01$

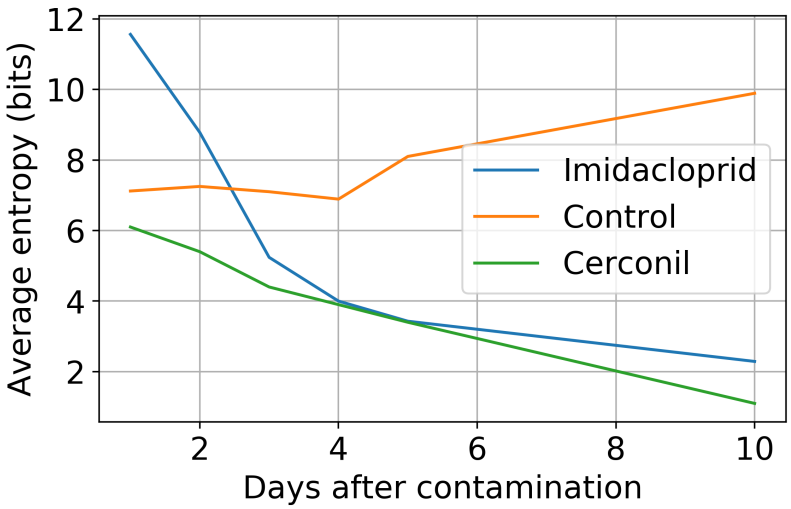


Figure 14 – Average entropy for the contaminated bees. With less KE, the movement becomes more predictable, as expected. Since the entropy is a logarithmic function, the reduction of 6 bits in the first day to 1 bit in the last day means a reduction of 32 times in the degrees of freedom of the system.

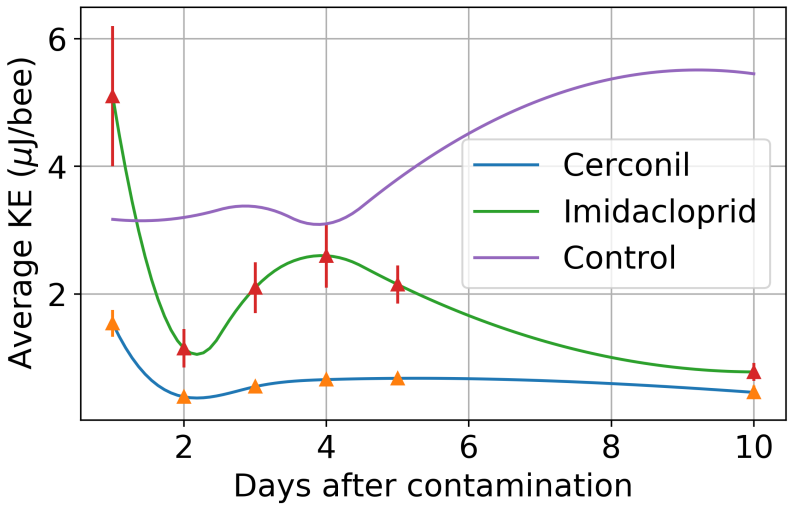


Figure 15 – Average KE for the populations. The energy in the control group is approximately constant until day 5, when it experiments a increase until day 10. The contaminated bees presents a fast decline in their KE (more intense with imidacloprid) related to the mortality. An attempt to recover can be observed from day 2 to day 4, without success.

With the trajectories obtained from the Bayesian algorithm, the KE was calculated using the average velocity, and then directly applying the formula for the KE. As expected, the energy drops from the first to the last day. However, there is an "attempt" of recovering in the second and third day. Fig. 15 exhibits the average value of KE for each day, after an interpolation to smooth the curves. The resilience analysis of these curves will be discussed in the next section. It is noticeable that the KE for the non-contaminated bees increase in the last days; this can be explained looking at Fig 10 and Fig. 16: most of the bees are alive and now occupy a wider area, having more possibility of locomotion, thus increasing its average KE.

With the PDFs of each bee, it becomes possible to evaluate the average entropy for the marked bees in each day. The Bayesian algorithm is robust enough to calculate the number of live bees each day. The dead bees are naturally ejected from the colony by other bees. Since the nucleus has an opening to the exterior environment, the number of bees can vary, but with a general downward trend (Fig. 16), which can be better explained as mortality by the agrochemical.

The entropy was calculated using the algebraic formulation of the PDF obtained via KDE for each bee. For every bee, the PDF was inserted in Eq. 2.9 and its entropy was evaluated. The average value, as expected, dropped from the first to the last day, corroborating the hypothesis that the movement becomes more predictable, with less KE (Fig. 15 and Fig. 14). Analogously with the concept of temperature, one can imagine the bees as molecules, and with less KE, their movement becomes more predictable. In absolute zero (total death), the movement is entirely predictable (the bees stay at the same position), and the entropy is zero.

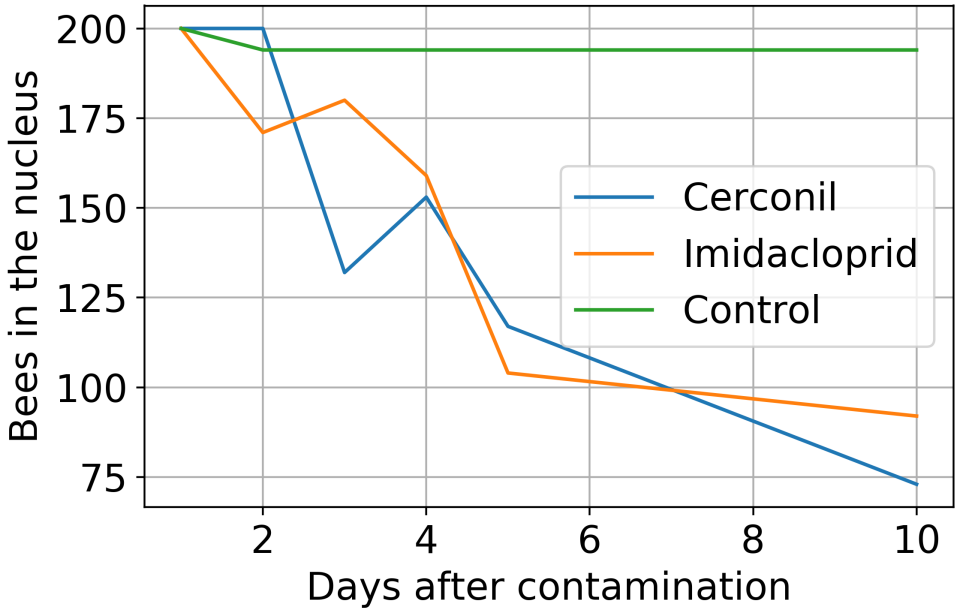


Figure 16 – Dynamical results of the contaminated bees over time: average number of contaminated bees for each day in the nucleus. The trend is downwards, but on some days, there can be an increase of the total number, because the bees are free to go out of the hives.

4.1.5 Resilience measurements

With the curves of the KE obtained from the Bayesian algorithm (Fig. 15), the same reasoning present in Fig. 6 is applied to measure how the agrochemicals affects the nucleus.

Table 7 – Vulnerability of the contaminated bees. Since the physical quantity being analysed is the KE, and the vulnerability is the area under the curve (Fig. 15), it has units of $\mu J \cdot$ day/bee, and measures the total loss of KE during the period of the 10 days.

Agrochemical	Vulnerability ($\mu J \cdot$ day/bee)
Imidacloprid	2.52
Cerconil	0.74

4.2 Lethal doses

The CDF of the lethal experiments are exhibited in Fig. 17. Since the mixture is exponentially more lethal than the isolated substances, the CDF for this case is almost a step function, and reaches 100% of dead bees a thousand times faster.

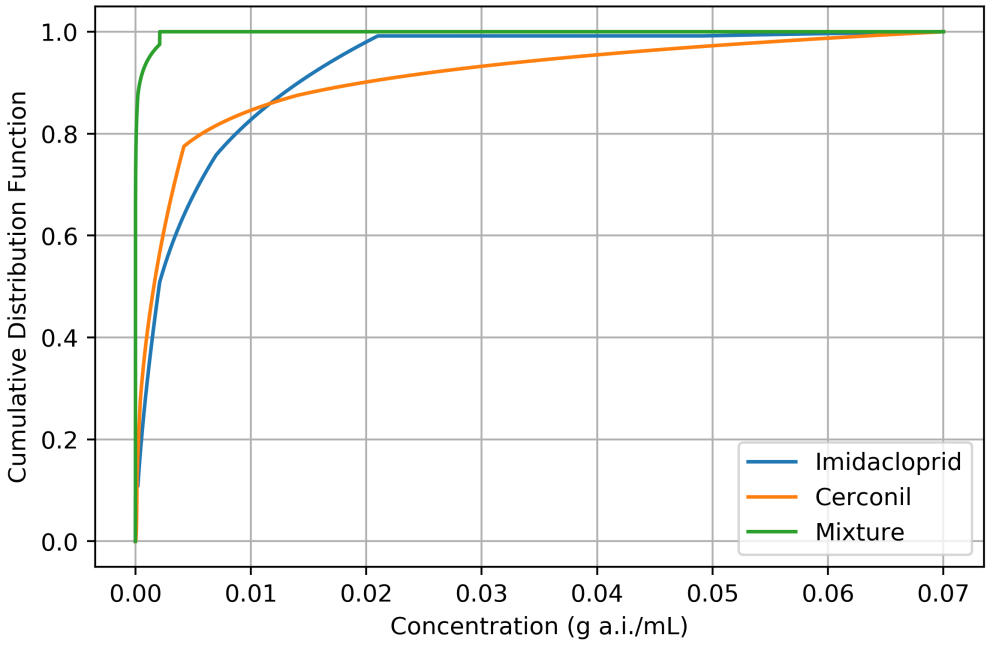


Figure 17 – Cumulative Distribution Function measuring the average percentile of the 20 bees in each of the assays. The mixture has a much faster slew rate, and it is therefore much more powerful.

The correlations in the information theoretical content of the PDFs of imidacloprid and cerconil are greater for the CDF in the concentrations nearly the minimum and the maximum (Fig. 18), since their behaviour is similar in both cases.

Table 8 – Measures of entropy: the joint distribution was calculated via Sklar's Theorem as in Eq. 3.2, and the mutual information is defined only for it. The entropy of the two isolated agrochemicals was calculated with the logarithmic interpolation.

Agrochemical	Entropy of PDF (bits)	Mutual Information (bits)
Imidacloprid	1.0009	———
Cerconil	0.9633	———
Mixture	———	0.2326

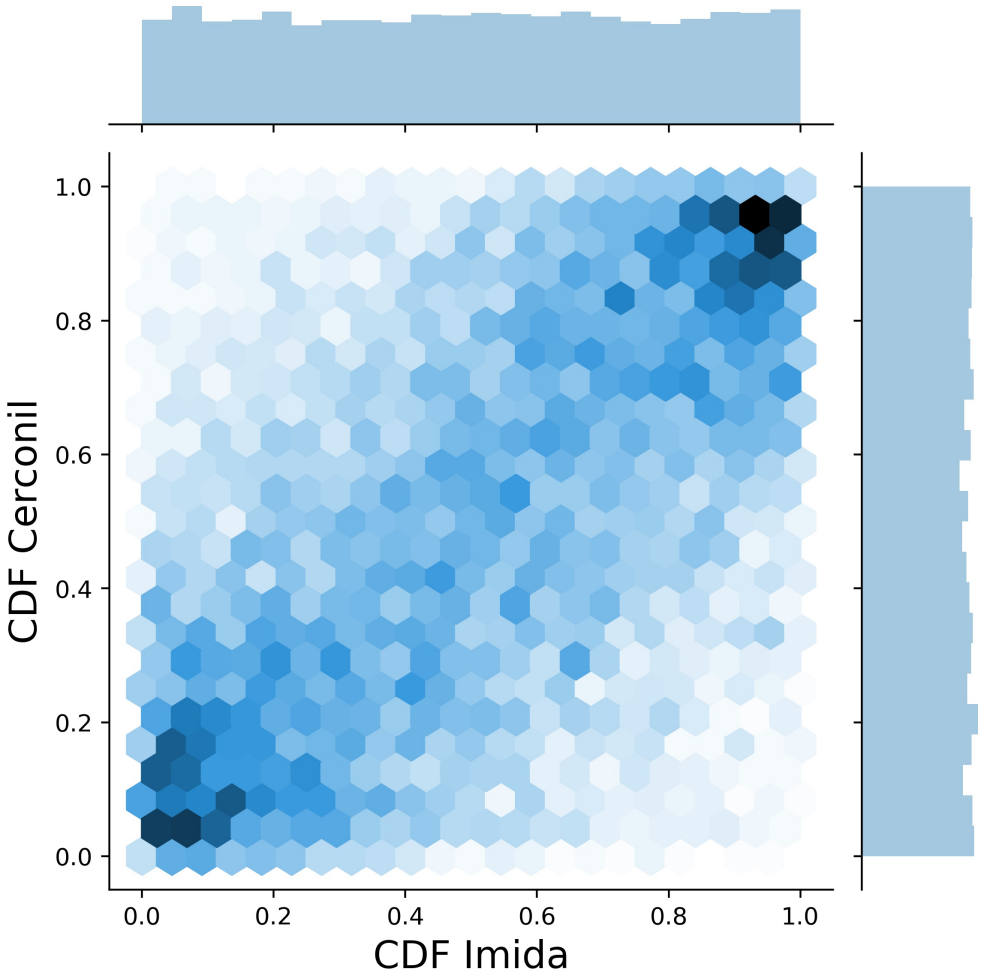


Figure 18 – Copula between the CDFs of the two agrochemicals. Both have roughly the same lethality in in the low and high concentrations, therefore, their correlation is maximum at theses regions of the copula.

5 Discussion and conclusion

Based on the simulations, the Bayesian tracking algorithm reached great precision when the radius was maximum at 10 (Tab. 5), but did not perform well when it was maximum at 50 (Tab. 7). Since the radius is larger in the second case, there is a good probability that the label swap happens in the first iterations, before the algorithm has learned the correct PDF for the individuals. These swaps are less frequently with smaller radius, making the precision increase; however, there is an integral being calculated for each iteration as in Eq. 2.5, and the execution time also increase.

Is it possible, therefore, to achieve a trajectory hit-rate of above 99%, dividing the maximum radius by two and sacrificing the execution time. As it is common a frame-rate of 30 or 60 Hz, the radius can be divided even more, resulting in increasing precision (and time). When the swap problem happens, the code had learned the PDF of the objects movement and did not mistake their trajectories. One of the performance evaluations of the algorithm is the correct recognition of the 200 bees on the first day in the nuclei (Fig. 16 (above)). This number is much higher than the standard biology EthosVision can track (NOLDUS; SPINK; TEGELENBOSCH, 2001), and many other tracking algorithms (PARKER et al., 2015; HONG et al., 2015), keeping the accuracy above 99% for the stipulated radius.

The code runs in about 90 minutes for a 12 minutes bee-video in the hardware used. Previous approaches tracked the objects using only geometric rules just like Ahmed *et al.* (AHMED et al., 2018) and ran in about 15 minutes for the same 12 minutes video. The disadvantage of the former method was the inability of learning: it always had the same assumptions for choosing the best next position for every bee. As can be seen in Table 3 of Ahmed *et al.*, the hit rate of their code is approximately the same as ours if the radius is taken as some value between 20 and 30 pixels. The algorithm in this work has its efficiency adjustable at the cost of more processing and time.

The Bayesian algorithm, on the other hand, detects all the variations in the distribution. This characteristic makes the code usable for any application in which the objective is to track objects of distinct colours of the background, for example, drone swarms, or satellite monitoring of moving objects. The tracking allowed the estimation of the KE, and the Bayesian Theorem enabled the KDE to calculate the entropy. The complexity of the algorithm described here is smaller than the one described in Wang *et al* (WANG et al., 2015), since it does not have a deep learning aspect, and the learning is totally Bayesian. Also, the overlap problem is treated with Bayes' Theorem, assuming the object is in rest state during the overlap. Future work can include a better treatment for this issue.

The algorithms were all written in open source software. The *openCV* library for *Python* contains several tools for image processing, and the *scipy* library for scientific computation saved time once it has the complex KDE algorithms already implemented.

The estimation of entropy using Bayesian Inference is not found in the literature, and can be an important measurement for dynamical evolving systems monitored via video, as traffic, radio-frequency tracking or GPS. The association between Bayes' Theorem and Information Theory can broader, and this algorithm only needs slight modifications to embody them. One possible change can be done as in Zhou *et al.* (ZHOU; LI; HE, 2014) and associate Game Theory to better discriminate objects. Further developments in our algorithm can mix the high quality of background treatment in Ning *et al.* (NING et al., 2012) and the unmarked tracking of Bozek *et al.* (BOZEK et al., 2018) to achieve roughly the same performance in much noisier videos. Even though the Bozek algorithm is very precise, it has the disadvantages of needing a low noise image, and, since the identification is made by a convolutional neural

network (a black-box model) it cannot extract dynamical features of the system.

The practical results obtained here shows that despite cerconil being sold as a fungicide, it does affect the metabolism of bees; in fact, its mortality rate is arguable greater than that of imidacloprid (Fig. 16 (a)), which is notable, since theoretically, it only affects fungi (while imidacloprid is sold as an insecticide). And even if the mortality is not 100%, the effects in the KE and entropy are noticeable and change the dynamics of the hives, as seen in Fig. 16 (b) and Fig. 14. The effects of cerconil in bees were previously analysed in Tome *et al.* (TOMÉ et al., 2017) and the results here corroborate the results. The KE is also affected by the cerconil. This result is in accord with a previous work that measured the fitness of honey bees (BLACQUIERE et al., 2012; MULLIN et al., 2010). However, the impact in the dynamics of the hives is more noticeable for the imidacloprid contamination (as expected by its nature), as can be seen in Fig. 14

The entropy evaluation shows that even the remaining bees had their behaviour changed: the movement became more predictable, and it is not due to the presence of dead bees in the nucleus since the other bees remove them. Its state space exponentially decreased (PINCUS, 1991) and is pointing towards death, since zero entropy would indicate that all the contaminated bees have died (BROOKS et al., 1989; ZADEH, 1968).

The mutual information calculated between the imidacloprid and cerconil (Table 8) reveals that the two agrochemicals shares about 21% of the physical information concerning the mortality rate, which is significantly higher than expected from a fungicide and a insecticide. This result corroborates even more the hypothesis of cerconil being lethal to honey bees.

The Bayesian tracking algorithm was effective and made possible the estimations of dynamical quantities. This method has two main advantages over classical inference: it can translate previous knowledge about the system into mathematical equations and it can learn patterns once more data is inserted. These properties allowed for the code to correct possible errors during the processing, and also predict paths and distributions in other cases of segmentation by colour, making it flexible.

A Algorithms

A.1 Conversion of the videos in sequence of frames

```
import cv2
import os
import time

start_time = time.time()
v = [1,2,3,4,5,10]
for i in v:
    folder = 'folder'
    os.mkdir(folder)

    vidcap = cv2.VideoCapture('folder/dia'+str(i))

    count = 0
    while True:
            vidcap.set(cv2.CAP_PROP_POS_MSEC,(count*1000))
            success,image = vidcap.read()


            if not success:
                break
            cv2.imwrite(os.path.join(folder,"frame{:d}.jpg".format(count)), image)     #salva o frame no formato jpeg
            x = "frame{:d}.jpg".format(count)
            count += 1


end_time = time.time()
elapsed = end_time-start_time
print(elapsed)
```

A.2 Convert RGB frames to binary frames

```
import cv2
import numpy as np
from glob import glob
import time

start = time.time()
# construct the argument parse and parse the arguments

v = [1]

for i in v:
    img_mask = 'folder_mask'
    img_names = glob(img_mask)

    boundaries = [
    #    ([40, 40, 100], [80, 80, 200])
         #([120, 180, 180], [180, 255, 255])
         ([0, 0, 150], [100, 100, 255])
        ]
    # load the image
    count = 0
    if i == 10:
        count = 18

    while count<len(img_names):
        fn = 'folder/frame'+str(count)+'.jpg'
        im_gray = cv2.imread(fn,0)
        thresh = 240
        im_bw = cv2.threshold(im_gray, thresh, 255, cv2.THRESH_BINARY)[1]
        cv2.imwrite('/home/lps/Downloads/bw_image.png', im_bw)
        for (lower, upper) in boundaries:
            lower = np.array(lower, dtype = "uint8")
            upper = np.array(upper, dtype = "uint8")
            mask = cv2.inRange(image, lower, upper)
            output = cv2.bitwise_and(image, image, mask = mask)
            cv2.imwrite('folder/frames_bw/frame'+str(count)+'.png',output)
            saida = cv2.imread('folder/frame'+str(count)+'.png',0)
            cv2.imwrite('folder/bw/frame'+str(count)+'.png',saida)
        count += 1

end = time.time()

elapsed = end - start
print(elapsed)
```

A.3 Extract the centroids

```
import cv2
from glob import glob
import math
from skimage import measure
import numpy as np
import time


start = time.time()

pi = 3.141592654

def dist (pt1,pt2):
    dist = math.sqrt((pt1[0]-pt2[0])*(pt1[0]-pt2[0]) + (pt1[1]-pt2[1])*(pt1[1]-pt2[1]))
    return dist

rastreio = []
direcao = []
frame = []
traj = []

i = 0


maximo = 0
raio = 30
count = 0
img_mask = '/home/lps/Downloads/bees/branco/dia 4/bw manha/*.jpg'
img_names = glob(img_mask)

while(i<len(img_names)):
    frame.append([])
    i += 1

while count<len(img_names):
    fn = 'folder/bw'+str(count)+'.jpg'
    img = cv2.imread(fn, 0)
    ret,bw = cv2.threshold(img,210,255,cv2.THRESH_BINARY)
    bw[:,1400:1920] = 0
    bw[0:400,:] = 0
    bw[:,0:300] = 0
    bw[800:1080,:] = 0

    labels = measure.label(bw)


    properties = measure.regionprops(labels)
    x = [prop.area for prop in properties]
    y = [prop.centroid for prop in properties]


    objetos = []
    centroides = []


    for i in range(0,len(x)-1):
        if x[i]>=5 and x[i]<50:
            objetos.append(x[i])
            centroides.append(y[i])
    frame.append(centroides)
    count += 1


j = 0
while j < len(img_names):
    x = str(j)
    with open('folder/frames/frame'+x+'.txt', 'w') as fp0:
        fp0.write('\n'.join('%s %s' % x for x in frame[j]))
        j += 1

end = time.time()


elapsed = end - start
print(elapsed)
```

A.4 Tracking with the Bayesian algorithm

```
import numpy as np
#import cv2
import math
import time

start =time.time()

import os

nframes = next(os.walk('folder/dia n/file_tarde'))[2]
nframes = len(nframes)
traj = []
posicao = []
for i in range(nframes):
    traj.append([])
    posicao.append([])
i = 0
for i in range(nframes-1):
    ind = str(i)
    with open('folder/frame'+ind+'.txt',"r") as infile:
        for line in infile:
            line = line.split()
            traj[i].append((float(line[0]),float(line[1])))
        #imp
rd = 50

pi = 3.141592654

def dist(coord1, coord2):
    return math.sqrt((coord1[0]-coord2[0])**2+(coord1[1]-coord2[1])**2)

def angulo(anterior, atual, proximo):
    if anterior == atual or anterior == proximo or atual == proximo:
        return 0
    else:
        anterior = np.array(anterior)
        atual = np.array(atual)
        proximo = np.array(proximo)
        l1 = np.linalg.norm(anterior-atual)
        l2 = np.linalg.norm(anterior-proximo)
        l3 = np.linalg.norm(proximo-atual)
        if abs(-(l2**2-l1**2-l3**2)/(2*l1*l3))>1:
            return pi/2
        elif proximo[1] < anterior[1]:

            return -(pi-math.acos(-(l2**2-l1**2-l3**2)/(2*l1*l3)))
        else:
            return (pi-math.acos(-(l2**2-l1**2-l3**2)/(2*l1*l3)))


#
def modified_bessel(k, alpha):
    theta = np.linspace(0,pi,1000)
    if alpha == 0:
        integrante = [math.exp(k*math.cos(i)) for i in theta]
        return (1/pi)*np.trapz(integrante,theta,dx = (theta[1]-theta[0]))
    else:
        t = np.linspace(0,100,1000000)
        integrante1 = [(1/pi)*math.exp(k*math.cos(i))*math.cos(alpha*i) for i in theta]
        integrante2 = [(1/pi)*math.sin(alpha*pi)*math.exp(-k*math.cosh(i) - alpha*i) for i in t]
        return np.trapz(integrante1,theta,dx = theta[1]-theta[0])+np.trapz(integrante2,t,dx = t[1]-t[0])
def vonMises(mu,k):
    x = np.linspace(-pi,pi,1000)
    return [(1/(2*pi*modified_bessel(k, 0)))*(math.exp(k*math.cos(i-mu))) for i in x]


def vonMises_y(mu,k,x):
    return (1/(2*pi*modified_bessel(k, 0)))*(math.exp(k*math.cos(x-mu)))

def normal(mu,sigma):
    x = np.linspace(-pi,pi,1000)
    return [(1/(sigma*math.sqrt(2*pi)))*math.exp((-(i-mu)**2)/(2*sigma**2)) for i in x]

def uniform(a,b):
    u = b-a
    x = np.linspace(0,u,1000)
    return [1/u for i in x]
```

```
def posterior(f,prior):
    x = np.linspace(-pi,pi,1000)
    prod = []
    for i in range(1000):
        prod.append(f[i]*prior[i])
    posterior = prod/np.trapz(prod,x,dx = (x[1]-x[0]))
    return posterior

correct_path = []
correct_fdp = []
correct_angle = []

for i in range(len(traj[0])):
    correct_path.append([])
    correct_fdp.append([])
    correct_angle.append([])

j = 0
while j < len(traj[0]):
    correct_fdp[j].append((0,4))
    correct_path[j].append(traj[0][j])
    correct_angle[j].append(0)
    j += 1

onlyfiles = nframes
radius = 10
fdp = vonMises(0,4)
ang = 0
prior = uniform(-pi,pi)
for i in range(1,onlyfiles):
    for j in range(len(correct_path)):
        add = True
        if i==1 or correct_path[j][-1] == correct_path[j][-2] or dist(correct_path[j][-1],correct_path[j][-2]) < radi
            possible = []
            for k in range(len(traj[i])):
                if dist(traj[i][k],correct_path[j][-1]) < radius:
                    possible.append((traj[i][k],k))
            if len(possible) != 0:
                correct_path[j].append(correct_path[j][-1])
                min_dist = 1000
                for tam in range(len(possible)):
                    space = dist(correct_path[j][-1], possible[tam][0])
                    if space < min_dist:
                        min_dist = space
                        correct_path[j][-1] = possible[tam][0]
                        index = possible[tam][1]
                del traj[i][index]
                correct_fdp[j].append(correct_fdp[j][-1])
                correct_angle[j].append(0)
            else:
                correct_path[j].append(correct_path[j][-1])
                correct_angle[j].append(0)
        else:
            possible = []
            mu = correct_fdp[j][-1][0]
            kappa = correct_fdp[j][-1][1]
            correct_angle[j].append([])
            for k in range(len(traj[i])):
                if dist(traj[i][k],correct_path[j][-1]) < radius:
                    possible.append((traj[i][k],k))   #caso 1: tr s pontos distintos consecutivos

            if len(possible) != 0 and add:
                atual = correct_path[j][-1]
                anterior = correct_path[j][-2]
                mle = 0
                for tam in range(len(possible)):
                    angle = angulo(anterior,atual,possible[tam][0])
                    y = vonMises_y(mu,kappa,angle)
                    if y > mle:
                        mle = y
                        correct_angle[j][-1] = angle
                        index = possible[tam][1]
                correct_path[j].append(possible[tam][0])
                mu = np.mean(correct_angle[j])
                kappa = correct_fdp[j][-1][1]
                f = vonMises(mu,kappa)
                kappa =max(posterior(f,prior))
                correct_fdp[j].append((mu,kappa))
                del traj[i][index]
                index = None
```

```
            elif len(possible) == 0 and add:
                    pos = correct_path[j][-1]
                    correct_path[j].append(pos)
                    correct_angle[j][-1] = 0    if len(traj[i]) != 0:                                   #caso 3: novos pontos
        size_list = len(correct_path)
        size = size_list
        for count in range(len(traj[i])):
            correct_path.append([])
            correct_angle.append([])
            correct_fdp.append([])
        for l in range(len(traj[i])):
            for count in range(i+1):
                correct_path[size].append(traj[i][l])
                correct_angle[size].append(0)
                correct_fdp[size].append((0,4))
            size += 1

j = 0
while j < len(correct_path):
    x = str(j)
    with open('folder/trajetorias/'+x+'.txt', 'w') as fp0:
        fp0.write('\n'.join('%s %s' % x for x in correct_path[j]))
        j += 1


end = time.time()
print(end-start)
```

A.5 Simulation

```
import numpy as np
import random
import math
import time

start =time.time()
x = []
y = []
rd = 5
nframes = 500
bees = 200
image=np.zeros((1920,1080,3),np.uint8)
xcoord = np.zeros((bees,nframes))
ycoord = np.zeros((bees,nframes))


for i in range(bees):
    x = random.randint(0,1920)
    y = random.randint(0,1080)
    xcoord[i][0] = x
    ycoord[i][0] = y


pi = 3.141592654
def new_point(x0,y0,r,m):
    x = x0+r*math.sqrt(1/(1+m**2))
    y = y0+m*r*math.sqrt(1/(1+m**2))
    return (x,y)


for i in range(bees):
    for j in range(1,nframes):
        x0 = xcoord[i][j-1]
        y0 = ycoord[i][j-1]
        r = random.randint(-rd,rd)
        mu = 6
        kappa = 8
        m = np.random.normal(mu,kappa)
        xcoord[i][j] = new_point(x0,y0,r,m)[0]
        ycoord[i][j] = new_point(x0,y0,r,m)[1]




def dist(coord1, coord2):
    return math.sqrt((coord1[0]-coord2[0])**2+(coord1[1]-coord2[1])**2)

def angulo(anterior, atual, proximo):
    if anterior == atual or anterior == proximo or atual == proximo:
        return 0
    else:
        anterior = np.array(anterior)
        atual = np.array(atual)
        proximo = np.array(proximo)
        l1 = np.linalg.norm(anterior-atual)
        l2 = np.linalg.norm(anterior-proximo)
        l3 = np.linalg.norm(proximo-atual)
        if abs(-(l2**2-l1**2-l3**2)/(2*l1*l3))>1:
            return pi/2
        elif proximo[1] < anterior[1]:

            return -(pi-math.acos(-(l2**2-l1**2-l3**2)/(2*l1*l3)))
        else:
            return (pi-math.acos(-(l2**2-l1**2-l3**2)/(2*l1*l3)))



def modified_bessel(k, alpha):
    theta = np.linspace(0,pi,1000)
    if alpha == 0:
        integrante = [math.exp(k*math.cos(i)) for i in theta]
        return (1/pi)*np.trapz(integrante,theta,dx = (theta[1]-theta[0]))
    else:
        t = np.linspace(0,100,1000000)
        integrante1 = [(1/pi)*math.exp(k*math.cos(i))*math.cos(alpha*i) for i in theta]
        integrante2 = [(1/pi)*math.sin(alpha*pi)*math.exp(-k*math.cosh(i) - alpha*i) for i in t]
        return np.trapz(integrante1,theta,dx = theta[1]-theta[0])+np.trapz(integrante2,t,dx = t[1]-t[0])
```

```
def vonMises(mu,k):
    x = np.linspace(-pi,pi,1000)
    return [(1/(2*pi*modified_bessel(k, 0)))*(math.exp(k*math.cos(i-mu))) for i in x]


def vonMises_y(mu,k,x):
    return (1/(2*pi*modified_bessel(k, 0)))*(math.exp(k*math.cos(x-mu)))

def normal(mu,sigma):
    x = np.linspace(-pi,pi,1000)
    return [(1/(sigma*math.sqrt(2*pi)))*math.exp((-(i-mu)**2)/(2*sigma**2)) for i in x]

def uniform(a,b):
    u = b-a
    x = np.linspace(0,u,1000)
    return [1/u for i in x]

def posterior(f,prior):
    x = np.linspace(-pi,pi,1000)
    prod = []
    for i in range(1000):
        prod.append(f[i]*prior[i])
    posterior = prod/np.trapz(prod,x,dx = (x[1]-x[0]))
    return posterior

traj = []
for i in range(nframes):
    traj.append([])

for j in range(xcoord.shape[1]):
    for i in range(xcoord.shape[0]):
        traj[j].append((xcoord[i][j],ycoord[i][j]))

correct_path = []
correct_fdp = []
correct_angle = []

for i in range(len(traj[0])):
    correct_path.append([])
    correct_fdp.append([])
    correct_angle.append([])



j = 0
while j < len(traj[0]):
    correct_fdp[j].append((0,4))
    correct_path[j].append(traj[0][j])
    correct_angle[j].append(0)
    j += 1

onlyfiles = nframes
radius = 10
fdp = vonMises(0,4)
ang = 0
prior = uniform(-pi,pi)
for i in range(1,onlyfiles):
    for j in range(len(correct_path)):
        add = True
        if i==1 or correct_path[j][-1] == correct_path[j][-2] or dist(correct_path[j][-1],correct_path[j][-2]) < radius:
            possible = []
            for k in range(len(traj[i])):
                if dist(traj[i][k],correct_path[j][-1]) < radius:
                    possible.append((traj[i][k],k))
            if len(possible) != 0:
                correct_path[j].append(correct_path[j][-1])
                min_dist = 1000
                for tam in range(len(possible)):
                    space = dist(correct_path[j][-1], possible[tam][0])
                    if space < min_dist:
                        min_dist = space
                        correct_path[j][-1] = possible[tam][0]
                        index = possible[tam][1]
                del traj[i][index]
                correct_fdp[j].append(correct_fdp[j][-1])
                correct_angle[j].append(0)
            else:
                correct_path[j].append(correct_path[j][-1])
                correct_angle[j].append(0)
        else:
```

```
            possible = []
            mu = correct_fdp[j][-1][0]
            kappa = correct_fdp[j][-1][1]
            correct_angle[j].append([])
            for k in range(len(traj[i])):
                if dist(traj[i][k],correct_path[j][-1]) < radius:
                    possible.append((traj[i][k],k))

            if len(possible) != 0 and add:
                atual = correct_path[j][-1]
                anterior = correct_path[j][-2]
                mle = 0
                for tam in range(len(possible)):
                    angle = angulo(anterior,atual,possible[tam][0])
                    y = vonMises_y(mu,kappa,angle)
                    if y > mle:
                        mle = y
                        correct_angle[j][-1] = angle
                        index = possible[tam][1]
                correct_path[j].append(possible[tam][0])
                mu = np.mean(correct_angle[j])
                kappa = correct_fdp[j][-1][1]
                f = vonMises(mu,kappa)
                kappa =max(posterior(f,prior))
                correct_fdp[j].append((mu,kappa))
                del traj[i][index]
                index = None
            elif len(possible) == 0 and add:
                    pos = correct_path[j][-1]
                    correct_path[j].append(pos)
                    correct_angle[j][-1] = 0
    if len(traj[i]) != 0:
        size_list = len(correct_path)
        size = size_list
        for count in range(len(traj[i])):
            correct_path.append([])
            correct_angle.append([])
            correct_fdp.append([])
        for l in range(len(traj[i])):
            for count in range(i+1):
                correct_path[size].append(traj[i][l])
                correct_angle[size].append(0)
                correct_fdp[size].append((0,4))
            size += 1



l = bees
a = []
b = []
for i in range(l):
    a = a + correct_path[i]

for i in range(l):
    for j in range(500):
        b.append((xcoord[i,j],ycoord[i,j]))

from difflib import SequenceMatcher
print(SequenceMatcher(None, a, b).ratio())

end = time.time()
print(end-start)
```

A.6 Maximum a Posteriori

```
import numpy as np
import math
import os
from matplotlib import pyplot as plt
import scipy.stats
import warnings
import time
warnings.filterwarnings('ignore')

start = time.time()


pi = 3.141592654
def new_point(x0,y0,r,m):
    x = x0+r*math.sqrt(1/(1+m**2))
    y = y0+m*r*math.sqrt(1/(1+m**2))
    return (x,y)


def dist(coord1, coord2):
    return math.sqrt((coord1[0]-coord2[0])**2+(coord1[1]-coord2[1])**2)

def angulo(anterior, atual, proximo):
    if anterior == atual or anterior == proximo or atual == proximo:
        return 0
    else:
        anterior = np.array(anterior)
        atual = np.array(atual)
        proximo = np.array(proximo)
        l1 = np.linalg.norm(anterior-atual)
        l2 = np.linalg.norm(anterior-proximo)
        l3 = np.linalg.norm(proximo-atual)
        if abs(-(l2**2-l1**2-l3**2)/(2*l1*l3))>1:
            return pi/2
        elif proximo[1] < anterior[1]:

            return -(pi-math.acos(-(l2**2-l1**2-l3**2)/(2*l1*l3)))
        else:
            return (pi-math.acos(-(l2**2-l1**2-l3**2)/(2*l1*l3)))



traj = []
trajexp = []
pdf = []

frame = next(os.walk('/home/lps/Downloads/bees/branco/dia 2/trajetorias'))[2]
onlyfiles = next(os.walk('/home/lps/Downloads/bayesiano/pdf'))[2]
tam = len(onlyfiles)
for i in range(len(onlyfiles)):
    traj.append([])
#     pdf.append([])

for i in range(len(frame)):
    traj[i] = np.genfromtxt(r'/home/lps/Downloads/bayesiano/path/abelha'+str(i)+'.txt', delimiter=' ')
#     pdf[i] = np.genfromtxt(r'/home/lps/Downloads/bayesiano/pdf/pdf'+str(i)+'.txt', delimiter=' ')

g = []
for i in range(len(pdf)):
    g.append(pdf[i][-1])

ang = []
space = []
fdp_ang = []
fdp_space = []
kde1= []
kde2 = []

tam = 0
while tam < len(traj):
    ang.append([])
    space.append([])
    fdp_ang.append([])
    fdp_space.append([])
    kde1.append([])
    kde2.append([])
    tam += 1
```

```
for k in range(len(traj)):
    for i in range(2,len(traj[k])):
        anterior = traj[k][i-2][0],traj[k][i-2][1]
        atual = traj[k][i-1][0],traj[k][i-1][1]
        proximo = traj[k][i][0],traj[k][i][1]
        ang[k].append(angulo(anterior,atual,proximo))
        space[k].append(dist(atual, proximo))

for i in range(len(traj)):
    data1 = np.asarray(ang[0])
    data2 = np.asarray(space[0])
    bw_values =  [None, 0.1, 0.01]
    kde1[i] = [scipy.stats.gaussian_kde(data1,bw_method=bw) for bw in bw_values]
    kde2[i] = [scipy.stats.gaussian_kde(data2,bw_method=bw) for bw in bw_values]


t_range = np.linspace(-pi,pi,1000)
s_range = np.linspace(0,50,1000)

for i in range(len(traj)):
    fdp_ang[i] = kde1[i][-1](t_range)
    fdp_space[i] = kde2[i][-1](s_range)

theta_hat = [((max(fdp_ang[i])*100/(2*pi))%10)/10 for i in range(len(fdp_ang))]

Sang = 0
Sspace = 0
dtheta = np.linspace(-pi,pi,1000)
delx = np.linspace(0,50,1000)
for i in range(len(fdp_ang)):
    integrante = [fdp_ang[i][j] for j in range(len(fdp_ang[i]))]
    real = []
    for k in range(len(integrante)):
        if integrante[k] > 0.001:
            real.append(integrante[k])
    integrante = [real[l]*math.log(real[l]) for l in range(len(real))]
    for k in range(len(real),1000):
        integrante.append(0)
    Sang = Sang + np.trapz(integrante,dtheta,dx = dtheta[1]-dtheta[0])

for i in range(1,len(fdp_space)):
    integrante = [fdp_space[i][j] for j in range(len(fdp_space[i]))]
    real = []
    for k in range(len(integrante)):
        if integrante[k] > 0.001:
            real.append(integrante[k])
    integrante = [real[l]*math.log(real[l]) for l in range(len(real))]
    for k in range(len(real),1000):
        integrante.append(0)
    Sspace = Sspace + np.trapz(integrante,delx,dx = delx[1]-delx[0])

S = (-Sang - Sspace)/(len(fdp_space)*math.log(2))
print('Entropy per bee (bits): '+str(S))
print('\n')
end = time.time()
print('Execution time (s): '+str(end-start))
```

B Relationship between the different types of entropy

B.1 Deriving Boltzmann's and Gibbs entropy

Lets take the system for the ideal gas and divide it in two parts. It is know that $S = S_1 + S_2$ and $W = W_1 W_2$. Therefore, the deduction is as follows:

$$S(W_1) + S(W_2) = S(W_1 W_2) \tag{B.1}$$

Deriving both sides with respect to W_1 and keeping W_2 constant results in

$$S'(W_1) = W_2 S'(W_1 W_2) \tag{B.2}$$

Deriving now in W_2 keeping W_1 constant, applying the chain rule:

$$0 = S'(W_1 W_2) + W_1 W_2 S''(W_1 W_2) \tag{B.3}$$

$$0 = S'(W) + W S''(W) \tag{B.4}$$

Replacing $S'(W) = f(W)$:

$$f(W) + W \frac{df(W)}{dW} = 0 \tag{B.5}$$

$$f(W)dW + W df(W) = 0 \tag{B.6}$$

$$(fW)' = 0 \tag{B.7}$$

Integrating both sides, returns

$$fW = k \tag{B.8}$$

which is the same as

$$W \frac{dS}{dW} = k \tag{B.9}$$

$$\int dS = k \int \frac{dW}{W} \tag{B.10}$$

$$S = k \log W + c \tag{B.11}$$

But it is known that a crystal at 0 K has 0 entropy, and only one microstate. Replacing this fact in the equation B.11:

$$0 = k \log 1 + c \tag{B.12}$$

from which we conclude that $c = 0$ and $S = k \log W$ is the entropy of an ideal gas, where k is the Boltzmann constant. Gibbs jaynes1965gibbs extended the concept of Boltzmann entropy to the cases in which the microstates are not equally likely:

$$S = -k \sum_i p_i \log p_i \tag{B.13}$$

where p_i is the probability of the i-nth microstate (if the W microstates are equally likely, then $p_{i=(1,2,3,\ldots,n)} = 1/W$ and Eq. B.13 is the same as the Boltzmann entropy).

B.1.1 Information theoretic proof that Gibbs entropy is the same as Clausius

With the development of information theory in the twentieth century and the concept of maximum entropy for statistical mechanics which states by the second law of thermodynamics that a system in thermodynamic equilibrium has reached its maximum entropy (and therefore, it is in the macrostate that has the most microstates, corresponding to gas velocities), it is possible to show that Shannon entropy is the same as Clausius entropy as well.

Using Eq. B.13, and the unitarity principle, $\sum_i p_i = 1$, in which i is the i-nth state, we can write the average energy of a system is

$$E = \sum_i p_i E_i = U \tag{B.14}$$

Applying Lagrange multipliers, we have

$$\begin{aligned} \mathcal{L} = & -k \sum_i p_i \log p_i + \lambda_1 \left(\sum_i p_i - 1 \right) \\ & + \lambda_2 \left(\sum_i p_i E_i - U \right) \end{aligned} \tag{B.15}$$

Differentiating and equaling zero

$$-k \log p_i - k + \lambda_1 + \lambda_2 E_i = 0 \tag{B.16}$$

Isolating p_i

$$p_i = \exp\left(\frac{-k + \lambda_1 + \lambda_2 E_2}{k} \right) \tag{B.17}$$

Using the canonical partition function Z, defined as

$$Z = \sum_i \exp\left(\frac{\lambda_2}{k} E_i \right) \tag{B.18}$$

The partition function combines state functions, such as temperature and energy for the microstates, and has a central role in statistical mechanics. Differentiating $\log Z$ with respect to λ_2 returns

$$\frac{\partial \log Z}{\partial \lambda_2} = \frac{E}{k} \tag{B.19}$$

Using unitarity again, Eq.B.17 can be written as

$$\exp\left(\frac{-k+\lambda_1}{k}\right) Z = 1 \tag{B.20}$$

Therefore,

$$\log\left(\frac{1}{Z}\right) + 1 = \frac{\lambda_1}{k} \tag{B.21}$$

Rewriting Eq. B.13 in terms of Z, results in

$$S = -k \sum_i p_i \left(\frac{\lambda_2}{k} E_2 - \log Z\right) \tag{B.22}$$

Using B.19 in B.22, give us

$$\begin{aligned} S &= -\lambda_2 \sum_i p_i E_i + k \log Z \sum_i p_i \\ &= -\lambda_2 U + k \log Z \end{aligned} \tag{B.23}$$

Using the definition of thermodynamics temperature

$$\frac{1}{T} = \frac{\partial S}{\partial U} \tag{B.24}$$

Since $\frac{\partial S}{\partial U} = -\lambda_2$, Eq. B.13 can be written as

$$S = \frac{U}{T} + k \log Z \tag{B.25}$$

Now, lets change the energy of the system by δQ. Every microstate will increase is energy by q_i. Calculating the change in the entropy results in

$$dS = \frac{\delta U}{T} + k\delta \log Z \tag{B.26}$$

Calculating the second term

$$\delta \log Z = \frac{d \log Z}{dZ} \delta Z = \frac{\delta Z}{Z} \tag{B.27}$$

Noticing that $Z = \sum_i \exp(-E_i/kT)$, the new partition function can be written as

$$Z = \sum_i \exp\left(-\frac{E_i + q_i}{kT}\right) \tag{B.28}$$

Applying Taylor expansion to $e^{-q_i/kt}$, since q_i is infinitesimal, the partition function is

$$Z = \sum_i \exp\left(\frac{E_i}{kT}\right)\left(1 - \frac{q_i}{kT}\right) = Z_0 + \delta Z \tag{B.29}$$

Therefore, the variation of the partition function is given by

$$\delta Z = -\frac{1}{kT}\sum_i q_i \exp\left(-\frac{E_i}{kT}\right) \tag{B.30}$$

Using the first law of thermodynamics, the change in U can be expressed as

$$\delta U = \sum_i \delta E_i p_i + \sum_i q_i p_i = \delta Q + \delta W \tag{B.31}$$

Calculating $\delta \log Z$, replacing B.30 in B.27:

$$\delta \log Z = -\frac{1}{kT}\sum p_i q_i \tag{B.32}$$

This value is exactly $\delta W/kT$. Replacing this relation in Eq. B.26, we get

$$dS = \frac{\delta Q}{T} \tag{B.33}$$

which is the Clausius first definition of entropy.

Bibliography

ADAMI, C. What is information? *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, The Royal Society Publishing, v. 374, n. 2063, p. 20150230, 2016.

AFFEK, A. N. Indicators of ecosystem potential for pollination and honey production. *Ecological Indicators*, Elsevier, v. 94, p. 33–45, 2018.

AGUIAR, P.; MENDONÇA, L.; GALHARDO, V. Opencontrol: a free opensource software for video tracking and automated control of behavioral mazes. *Journal of neuroscience methods*, Elsevier, v. 166, n. 1, p. 66–72, 2007.

AHMED, S. A. et al. Unsupervised classification of erroneous video object trajectories. *Soft Computing*, Springer, v. 22, n. 14, p. 4703–4721, 2018.

ALEXANDER, D. E. Resilience and disaster risk reduction: an etymological journey. *Natural hazards and earth system sciences*, Copernicus GmbH, v. 13, n. 11, p. 2707–2716, 2013.

ALLEN-WARDELL, G. et al. The potential consequences of pollinator declines on the conservation of biodiversity and stability of food crop yields. *Conservation biology*, JSTOR, p. 8–17, 1998.

ALMEIDA, D. d. et al. Plantas visitadas por abelhas e polinização. *Piracicaba: Esalq-Divisão de biblioteca e documentação*, 2003.

ANIL, R. et al. Large scale distributed neural network training through online distillation. *arXiv preprint arXiv:1804.03235*, 2018.

ARGHANDEH, R. et al. On the definition of cyber-physical resilience in power systems. *Renewable and Sustainable Energy Reviews*, Elsevier, v. 58, p. 1060–1069, 2016.

ASH, B. *Ash. Information Theory*. [S.l.]: Dover Publications Inc., New York, 1990.

BARTLE, R. G.; BARTLE, R. G. *The elements of integration and Lebesgue measure*. [S.l.]: Wiley Online Library, 1995. v. 27.

BAZHENOV, N. A.; KORZUN, D. G. Use of everyday mobile video cameras in iot applications. In: FRUCT OY. *Proceedings of the 22st Conference of Open Innovations Association FRUCT*. [S.l.], 2018. p. 42.

BECHER, M. A. et al. Bumble-beehave: A systems model for exploring multifactorial causes of bumblebee decline at individual, colony, population and community level. *Journal of Applied Ecology*, Wiley Online Library, 2018.

BENNISON, A. et al. Search and foraging behaviors from movement data: A comparison of methods. *Ecology and evolution*, Wiley Online Library, 2018.

BERNARDIN, K.; STIEFELHAGEN, R. Evaluating multiple object tracking performance: the clear mot metrics. *Journal on Image and Video Processing*, Hindawi Publishing Corp., v. 2008, p. 1, 2008.

BLACQUIERE, T. et al. Neonicotinoids in bees: a review on concentrations, side-effects and risk assessment. *Ecotoxicology*, Springer, v. 21, n. 4, p. 973–992, 2012.

BOSSOMAIER, T. et al. Information flow around stock market collapse. *Accounting & Finance*, Wiley Online Library, v. 58, p. 45–58, 2018.

BOUSSO, R. Black hole entropy and the bekenstein bound. *arXiv preprint arXiv:1810.01880*, 2018.

BOX, G. E.; TIAO, G. C. *Bayesian inference in statistical analysis*. [S.l.]: John Wiley & Sons, 2011. v. 40.

BOZEK, K. et al. Towards dense object tracking in a 2d honeybee hive. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2018. p. 4185–4193.

BRETTELL, L. E.; MARTIN, S. J. Oldest varroa tolerant honey bee population provides insight into the origins of the global decline of honey bees. *Scientific reports*, Nature Publishing Group, v. 7, p. 45953, 2017.

BROOKS, D. R. et al. Entropy and information in evolving biological systems. *Biology and philosophy*, Springer, v. 4, n. 4, p. 407–432, 1989.

BROOKS, D. R.; WILEY, E. O.; BROOKS, D. *Evolution as entropy*. [S.l.]: University of Chicago Press Chicago, 1988.

BROWNING, E. et al. Predicting animal behaviour using deep learning: Gps data alone accurately predict diving in seabirds. *Methods in Ecology and Evolution*, Wiley Online Library, v. 9, n. 3, p. 681–692, 2018.

CACOULLOS, T. Estimation of a multivariate density. *Annals of the Institute of Statistical Mathematics*, Springer, v. 18, n. 1, p. 179–189, 1966.

CAMERON, S. A. et al. Patterns of widespread decline in north american bumble bees. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 108, n. 2, p. 662–667, 2011.

CAMPBELL, J. *Grammatical man: Information, entropy, language, and life*. [S.l.]: Simon and Schuster New York, 1982.

CARO, J. A.; VALENTINE, K. G.; WAND, A. J. Role of conformational entropy in extremely high affinity protein interactions. *Biophysical Journal*, Elsevier, v. 114, n. 3, p. 67a, 2018.

CARTAR, R. V.; REAL, L. A. Habitat structure and animal movement: the behaviour of bumble bees in uniform and random spatial resource distributions. *Oecologia*, Springer, v. 112, n. 3, p. 430–434, 1997.

CHO, H.-J. et al. Newly developed method for mouse olfactory behavior tests using an automatic video tracking system. *Auris Nasus Larynx*, Elsevier, v. 45, n. 1, p. 103–110, 2018.

COVER, T. M.; THOMAS, J. A. *Elements of information theory*. [S.l.]: John Wiley & Sons, 2012.

DANAHER, P. J.; SMITH, M. S. Modeling multivariate distributions using copulas: applications in marketing. *Marketing Science*, INFORMS, v. 30, n. 1, p. 4–21, 2011.

DEGAETANO-ORTLIEB, S.; TEICH, E. Modeling intra-textual variation with entropy and surprisal: topical vs. stylistic patterns. In: *Proceedings of the Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*. [S.l.: s.n.], 2017. p. 68–77.

DEHGHANIAN, P. et al. Predictive risk analytics for weather-resilient operation of electric power systems. *IEEE Transactions on Sustainable Energy*, IEEE, v. 10, n. 1, p. 3–15, 2019.

DESSAVRE, D. G.; RAMIREZ-MARQUEZ, J. E.; BARKER, K. Multidimensional approach to complex system resilience analysis. *Reliability Engineering & System Safety*, Elsevier, v. 149, p. 34–43, 2016.

EVANS, W. E. D. *The chemistry of death*. [S.l.]: Charles C Thomas Pub Ltd, 1963.

FORYS, B. et al. Real-time markerless video tracking of body parts in mice using deep neural networks. *bioRxiv*, Cold Spring Harbor Laboratory, p. 482349, 2018.

FOTOUHI, H.; MORYADEE, S.; MILLER-HOOKS, E. Quantifying the resilience of an urban traffic-electric power coupled system. *Reliability Engineering & System Safety*, Elsevier, v. 163, p. 79–94, 2017.

GATTO, R.; JAMMALAMADAKA, S. R. The generalized von mises distribution. *Statistical Methodology*, Elsevier, v. 4, n. 3, p. 341–353, 2007.

GOLDFELD, D.; HOFFSTEIN, J. Eisenstein series of 1/2-integral weight and the mean value of real dirichletl-series. *Inventiones mathematicae*, Springer, v. 80, n. 2, p. 185–208, 1985.

GREVEN, A.; KELLER, G.; WARNECKE, G. *Entropy*. [S.l.]: Princeton University Press, 2014. v. 47.

GU, R.; XIONG, W.; LI, X. Does the singular value decomposition entropy have predictive power for stock market?—evidence from the shenzhen stock market. *Physica A: Statistical Mechanics and its Applications*, Elsevier, v. 439, p. 103–113, 2015.

GULER, S. Z. et al. *Video tracking systems and methods employing cognitive vision.* 2018. US Patent App. 14/545,365.

HAMILTON, J. D. *Time series analysis.* [S.l.]: Princeton university press Princeton, NJ, 1994. v. 2.

HARE, S. et al. Struck: Structured output tracking with kernels. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 38, n. 10, p. 2096–2109, 2016.

HAYFLICK, L. *Entropy explains aging, genetic determinism explains longevity, and undefined terminology explains misunderstanding both.* [S.l.]: Public Library of Science, 2007.

HE, D. et al. Identification of multiple faults in rotating machinery based on minimum entropy deconvolution combined with spectral kurtosis. *Mechanical Systems and Signal Processing*, Elsevier, v. 81, p. 235–249, 2016.

HENDERSON, L. The von neumann entropy: A reply to shenker. *The British journal for the philosophy of science*, Oxford University Press, v. 54, n. 2, p. 291–296, 2003.

HENRIQUES, J. F. et al. High-speed tracking with kernelized correlation filters. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 37, n. 3, p. 583–596, 2015.

HOLLING, C. S. Resilience and stability of ecological systems. *Annual review of ecology and systematics*, Annual Reviews 4139 El Camino Way, PO Box 10139, Palo Alto, CA 94303-0139, USA, v. 4, n. 1, p. 1–23, 1973.

HONG, W. et al. Automated measurement of mouse social behaviors using depth sensing, video tracking, and machine learning. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 112, n. 38, p. E5351–E5360, 2015.

HU, J.; SHEN, L.; SUN, G. Squeeze-and-excitation networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* [S.l.: s.n.], 2018. p. 7132–7141.

ISAACS, R. et al. Integrated crop pollination: combining strategies to ensure stable and sustainable yields of pollination-dependent crops. *Basic and Applied Ecology*, Elsevier, v. 22, p. 44–60, 2017.

JAYNES, E. T. Information theory and statistical mechanics (notes by the lecturer). In: *Statistical physics 3.* [S.l.: s.n.], 1963. p. 181.

JAYNES, E. T. Prior probabilities. *IEEE Trans. Systems Science and Cybernetics*, v. 4, n. 3, p. 227–241, 1968.

KOVALEV, A. V. Misuse of thermodynamic entropy in economics. *Energy*, Elsevier, v. 100, p. 129–136, 2016.

KULLBACK, S.; LEIBLER, R. A. On information and sufficiency. *The annals of mathematical statistics*, JSTOR, v. 22, n. 1, p. 79–86, 1951.

LAM, T.-Y. *Orderings, valuations and quadratic forms.* [S.l.]: American Mathematical Soc., 1983. v. 52.

LO, G. S. A simple proof of the theorem of sklar and its extension to distribution functions. *arXiv preprint arXiv:1803.00409*, 2018.

LOFTSGAARDEN, D. O.; QUESENBERRY, C. P. et al. A nonparametric estimate of a multivariate density function. *The Annals of Mathematical Statistics*, Institute of Mathematical Statistics, v. 36, n. 3, p. 1049–1051, 1965.

MAGGIO, E.; CAVALLARO, A. *Video tracking: theory and practice.* [S.l.]: John Wiley & Sons, 2011.

MAINWARING, A. et al. Wireless sensor networks for habitat monitoring. In: ACM. *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications.* [S.l.], 2002. p. 88–97.

MALDACENA, J. Black hole entropy and quantum mechanics. *arXiv preprint arXiv:1810.11492*, 2018.

MARITZ, J. S.; LWIN, T. *Empirical bayes methods.* [S.l.]: Routledge, 2018.

MARTINO, A. D.; MARTINO, D. D. An introduction to the maximum entropy approach and its application to inference problems in biology. *Heliyon*, Elsevier, v. 4, n. 4, p. e00596, 2018.

MARTYUSHEV, L. Entropy and entropy production: old misconceptions and new breakthroughs. *Entropy*, Multidisciplinary Digital Publishing Institute, v. 15, n. 4, p. 1152–1170, 2013.

MEIROVITCH, H. Recent developments in methodologies for calculating the entropy and free energy of biological systems by computer simulation. *Current opinion in structural biology*, Elsevier, v. 17, n. 2, p. 181–186, 2007.

MOROWITZ, H. Entropy and nonsense. *Biology and Philosophy*, Springer, v. 1, n. 4, p. 473–476, 1986.

MULLIN, C. A. et al. High levels of miticides and agrochemicals in north american apiaries: implications for honey bee health. *PloS one*, Public Library of Science, v. 5, n. 3, p. e9754, 2010.

NING, J. et al. Robust mean-shift tracking with corrected background-weighted histogram. *IET computer vision*, IET, v. 6, n. 1, p. 62–69, 2012.

NOLDUS, L. P.; SPINK, A. J.; TEGELENBOSCH, R. A. Ethovision: a versatile video tracking system for automation of behavioral experiments. *Behavior Research Methods, Instruments, & Computers*, Springer, v. 33, n. 3, p. 398–414, 2001.

OH, D. Y.; BARR, I. G.; HURT, A. C. A novel video tracking method to evaluate the effect of influenza infection and antiviral treatment on ferret activity. *PloS one*, Public Library of Science, v. 10, n. 3, p. e0118780, 2015.

OUYANG, M.; DUENAS-OSORIO, L. Multi-dimensional hurricane resilience assessment of electric power systems. *Structural Safety*, Elsevier, v. 48, p. 15–24, 2014.

PANTELI, M.; MANCARELLA, P. Influence of extreme weather and climate change on the resilience of power systems: Impacts and possible mitigation strategies. *Electric Power Systems Research*, Elsevier, v. 127, p. 259–270, 2015.

PARKER, J. E. et al. Infrared video tracking of anopheles gambiae at insecticide-treated bed nets reveals rapid decisive impact after brief localised net contact. *Scientific reports*, Nature Publishing Group, v. 5, p. 13392, 2015.

PARZEN, E. On estimation of a probability density function and mode. *The annals of mathematical statistics*, JSTOR, v. 33, n. 3, p. 1065–1076, 1962.

PEARL, J.; MACKENZIE, D. *The book of why: the new science of cause and effect*. [S.l.]: Basic Books, 2018.

PÉREZ-ESCUDERO, A. et al. idtracker: tracking individuals in a group by automatic identification of unmarked animals. *Nature methods*, Nature Publishing Group, v. 11, n. 7, p. 743, 2014.

PINCUS, S. M. Approximate entropy as a measure of system complexity. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 88, n. 6, p. 2297–2301, 1991.

POPOVIC, M. Researchers in an entropy wonderland: A review of the entropy concept. *arXiv preprint arXiv:1711.07326*, 2017.

POTTS, S. G. et al. Global pollinator declines: trends, impacts and drivers. *Trends in ecology & evolution*, Elsevier, v. 25, n. 6, p. 345–353, 2010.

REYNAR, J. C.; RATNAPARKHI, A. A maximum entropy approach to identifying sentence boundaries. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the fifth conference on Applied natural language processing*. [S.l.], 1997. p. 16–19.

RISKEN, H. Fokker-planck equation. In: *The Fokker-Planck Equation*. [S.l.]: Springer, 1996. p. 63–95.

ROSENBLATT, M. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, JSTOR, p. 832–837, 1956.

ROSSETTI, B. J. et al. Graphite: A graphical environment for scalable in situ video tracking of moving insects. *Methods in Ecology and Evolution*, Wiley Online Library, v. 9, n. 4, p. 956–964, 2018.

ROSTAGHI, M.; AZAMI, H. Dispersion entropy: A measure for time-series analysis. *IEEE Signal Processing Letters*, IEEE, v. 23, n. 5, p. 610–614, 2016.

SCHNEIDER, E. D.; KAY, J. J. Life as a manifestation of the second law of thermodynamics. *Mathematical and computer modelling*, Elsevier, v. 19, n. 6-8, p. 25–48, 1994.

SCHRODINGER, E. Order, disorder and entropy. *Modern Systems Research for the Behavioral Scientist. Chicago: Aldine*, p. 143–146, 1968.

SCHWARTZKOPF-GENSWEIN, K.; HUISMA, C.; MCALLISTER, T. Validation of a radio frequency identification system for monitoring the feeding patterns of feedlot cattle. *Livestock production science*, Elsevier, v. 60, n. 1, p. 27–31, 1999.

SHANNON, C. E. A mathematical theory of communication. *Bell system technical journal*, Wiley Online Library, v. 27, n. 3, p. 379–423, 1948.

SOUBANE, D. et al. Hidden information, energy dispersion and disorder: Does entropy really measure disorder? *World*, v. 8, p. 197–202, 2018.

SOZBILIR, M. What students' understand from entropy?: A review of selected literature. *Journal of Baltic Science Education*, v. 2, n. 1, p. 21–27, 2003.

TAME, J. R. On entropy as mixed-up-ness. In: *Approaches to Entropy*. [S.l.]: Springer, 2019. p. 153–170.

TEKALP, A. M. *Digital video processing*. [S.l.]: Prentice Hall Press, 2015.

TERRELL, G. R.; SCOTT, D. W. Variable kernel density estimation. *The Annals of Statistics*, JSTOR, p. 1236–1265, 1992.

THOMPSON, H. M. Behavioural effects of pesticides in bees–their potential for use in risk assessment. *Ecotoxicology*, Springer, v. 12, n. 1-4, p. 317–330, 2003.

THORÉN, H.; OLSSON, L. Is resilience a normative concept? *Resilience*, Taylor & Francis, v. 6, n. 2, p. 112–128, 2018.

TOMÉ, H. V. et al. Agrochemical synergism imposes higher risk to neotropical bees than to honeybees. *Royal Society open science*, The Royal Society, v. 4, n. 1, p. 160866, 2017.

TOMé, H. V. V. et al. Agrochemical synergism imposes higher risk to neotropical bees than to honeybees. *Royal Society Open Science*, v. 4, n. 1, p. 160866, 2017.

WANG, L. et al. Video tracking using learned hierarchical features. *IEEE Transactions on Image Processing*, IEEE, v. 24, n. 4, p. 1424–1435, 2015.

WANG, Y. et al. Research on resilience of power systems under natural disasters—a review. *IEEE Transactions on Power Systems*, IEEE, v. 31, n. 2, p. 1604–1613, 2016.

WEHRL, A. General properties of entropy. *Reviews of Modern Physics*, APS, v. 50, n. 2, p. 221, 1978.

WRIGHT, P. Entropy and disorder. *Contemporary Physics*, Taylor & Francis, v. 11, n. 6, p. 581–588, 1970.

XIAO, M. et al. Supermassive black holes with high accretion rates in active galactic nuclei. vii. reconstruction of velocity-delay maps by the maximum entropy method. *The Astrophysical Journal*, IOP Publishing, v. 864, n. 2, p. 109, 2018.

XUE, W.; XU, C.; FENG, Z. Robust visual tracking via multi-scale spatio-temporal context learning. *IEEE Transactions on Circuits and Systems for Video Technology*, IEEE, v. 28, n. 10, p. 2849–2860, 2018.

ZADEH, L. A. Probability measures of fuzzy events. *Journal of mathematical analysis and applications*, Academic Press, v. 23, n. 2, p. 421–427, 1968.

ZEESHAN, A. et al. Shape effect of nanosize particles in unsteady mixed convection flow of nanofluid over disk with entropy generation. *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering*, SAGE Publications Sage UK: London, England, v. 231, n. 4, p. 871–879, 2017.

ZHOU, X.; LI, Y.; HE, B. Entropy distribution and coverage rate-based birth intensity estimation in gm-phd filter for multi-target visual tracking. *Signal Processing*, Elsevier, v. 94, p. 650–660, 2014.