



HAL
open science

Multiple honey bees tracking and trajectory modeling

Baptiste Magnier, Eliyahou Gabbay, Faysal Bougamale, Behrang Moradi, François Pfister, Pierre Slangen

► **To cite this version:**

Baptiste Magnier, Eliyahou Gabbay, Faysal Bougamale, Behrang Moradi, François Pfister, et al.. Multiple honey bees tracking and trajectory modeling. Multimodal Sensing and Artificial Intelligence: Technologies and Applications, Jun 2019, Munich, France. pp.29, <10.1117/12.2526120>. <hal-02187412>

HAL Id: hal-02187412

<https://hal.science/hal-02187412v1>

Submitted on 26 Feb 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Multiple Honey Bees Tracking and Trajectory Modeling

Baptiste Magnier^a, Eliyahou Gabbay^a, Faysal Bougamale^a, Behrang Moradi^a, François Pfister^a, and Pierre Slangen^b

^aIMT Mines Alès, LGI2P, 6. avenue de Clavières 30000 Alès, France

^bIMT Mines Alès, LGEI, 6. avenue de Clavières 30000 Alès, France

ABSTRACT

The current context of biodiversity loss is particularly marked by the Colony Collapse Disorder (CCD) of honeybees due to multiple causes, toxicological, parasitic and viral. The beekeeper has to face these difficulties in order to maintain the population of bees to save the species but also to make its exploitation profitable. According studies, one can understand what is happening inside the hive by observing what is going on outside. In this context, we propose to individually capture by video the flight trajectories of bees and then characterize the pace of the global activity in front of the hive to infer observations that will be consolidated and made available to apicultural data scientists. Thus bees are detected and tracked using image and video processing methods, then the trajectory are modeled.

Then, from the extracted data outcome of the videos, curves are fitted as the ideal trajectories of each bee path in order to study and classify their behaviors. Thus, for each tracked bee, the points of extracted centered positions are time-ordered approximated on a plan. The chosen method interpolates the abscissae separately from the ordinates as time-dependent functions before plotting the parametric curve for each bee path individually. Thus, the abscissae as the ordinates are interpolated using cubic splines. The consecutive points to be interpolated are connected by polynomials of degree three. The first and second derivatives of these polynomials must be connected too. This allows the curve to look more natural by avoiding tingling and convexity discontinuities. Finally, it represents the continuity of the speed of the bees too. Experiments on synthetic and real videos show precise detections of the bee paths. Looking forward, through the collected data, the bee behavior could be understood by using machine learning and the semi supervised method must be one way to proceed.

Keywords: Bee detection, tracking, motion analysis, trajectory measurement, bee behavior modeling, interpolation, cubic spline, fitting

1. INTRODUCTION AND MOTIVATIONS

The current context of biodiversity loss is particularly marked by the Colony Collapse Disorder (CCD) of honeybees. The CCD combines toxicological (phytosanitary treatments), parasitic (*Varroa Destructor*), predators (Asian hornets), and also bacteriological and viral causes. Consequently, the beekeeper has to face these difficulties in order to maintain the population of bees, to save the species but also to make its exploitation profitable. A new discipline, known as connected beekeeping allows him to collect a lot of field data in the apiary. By leveraging the Internet of Things (IoT) to collect through sensors a lot of measured data on hives, and submitting these data to machine learning based on algorithms, the apiarist is able to get a fine knowledge of his livestock and also decision-making elements to improve the management of his exploitation, in the long term (selection of strains) or short-term (operational actions). One of the field data is the behavior of bees at the hive entrance. Storch and Mosbeux¹ consider that one can understand what is happening inside the hive by observing what is going on outside, near the hive. We propose to individually capture the flight trajectories of bees and then characterize the pace of the global activity in front of the hive to infer observations that will be consolidated and made available to apicultural data scientists. Thus, by submitting these data to machine learning based

Further author information: (Send correspondence to B. M.)

B.M. : E-mail: baptiste.magnier@mines-ales.fr, Telephone: +33 (0)4 34 24 62 90

F. P.: E-mail: pfister@connecthive.com

algorithms, the apiarist is able to get a fine knowledge of his livestock and also decision-making elements to improve the management of his exploitation. Before submitting visual data to machine learning, one have to extract the interesting features by the mean of classical artificial vision methods.

Firstly, using a non-invasive camera placed above a beehive entrance such that the entire entrance width is covered by the camera bee are detected and tracked. Fig. 1 shows three consecutive images extracted from a video (a-c), and the center of each bee are plotted in the image fusion (d). Several techniques exist to detect flying bees²⁻⁹ in order to observe bees to monitor abnormal behaviors; these observations can be used to diagnose the health of a hive; but the trajectory modeling had not been developed yet. The behavior modeling issued of the bee trajectories represents the motivation of this work. Trajectories of bees are recorded by the mean of standard cameras and then characterize the pace of the global activity in front of the hive to infer observations that will be consolidated and made available to apicultural data scientists. In this work, the approach to detect and track bees is an improvement of the method developed by Magnier et al.⁶ Indeed, a standard camera is installed at the entrance of the hive. This study had limitations due to the flight speed of the bees compared to the performance of the camera. This tracking technique, estimating ellipse orientations (tied to the shape of a bee) and displacements, allows to extract positions of the flying bees, even in the presence of shadows created by bees themselves or objects close to the hive.

Secondly, from the extracted data outcome of the videos, curves are fitted as the ideal trajectories of each bee path in order to study and classify their behaviors. For each tracked bee, the points of extracted centered positions are time-ordered approximated on a plan (coordinates are in two dimensions). Actually, the detection step gives discrete real bees positions. Therefore, an interpolation method is required to approximate continuous curves which do not pass necessarily through each detected points. Hence, the presented study makes it possible to reconstruct the natural trajectories by polynomial interpolation. The chosen method interpolates the abscissae separately from the ordinates as time-dependent functions before plotting the parametric curve for each bee path individually. Thus, the abscissae as the ordinates are interpolated using cubic splines.¹⁰ The consecutive points to be interpolated are connected by polynomials of degree three. The first and second derivatives of these polynomials must be connected too. This allows the curve to look more natural by avoiding tingling and convexity discontinuities. Finally, it represents the continuity of the speed of the bees too. Experiments on synthetic and real videos show precise detections of the bee paths.

Looking forward, through the collected data, the bee behavior could be understood by using machine learning

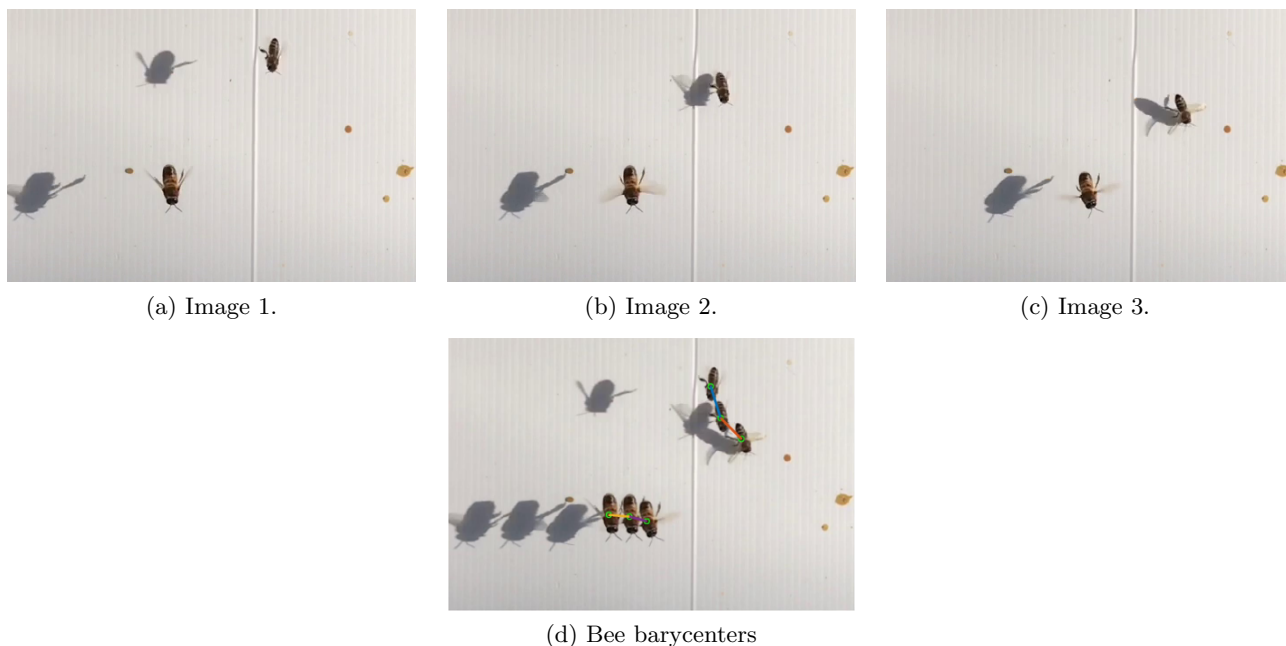


Figure 1. Example of tracked bees in front of the hive entrance.

and the semi supervised method must be one way to proceed.

During the experiment, the camera acquires 60 images per second (fps) in order to build a video. At this stage, the position of each detected bee in the current image is computed. The aim is to compute curves which fit the best each tied bee trajectory. As discrete positions are extracted, curves must pass through each of these points. That is why an interpolation method is used contrary to another approximation approach which has for objective to be as close as possible of each point. As the trajectories of bees are two-dimensional and bee positions are time ordered: the chosen method interpolates the abscissae separately from the ordinates as time-dependent functions before plotting the parametric curve for each bee path individually. So, theoretically,¹⁰ the two-dimensional interpolation problem becomes two unidimensional interpolation problems. Now, considering the a discrete trajectory of a bee path has been extracted, the interpolation methods are detailed below.

2. DETECTION AND TRACKING OF BEES

Usually, bees go out of the hive to visit flowers. The daily activity starts at sunrise and stops at sunset. The phases of high activity often match with very sunny periods and this periods is fundamental for evaluating the hive health. Nevertheless, as the camera is placed verticality above the landing board, the bee shadows are projected against the white background and also recorded and a bee could be counted twice: the bee and its associated shadow (as shown in Fig. 2 (c)). The closer the bee is to the background, the darker the shadow will appear and could be detected whereas it corresponds to an outlier. Hence, the used algorithm for bee detection and tracking corresponds to the method developed by Magnier et al.,⁶ this algorithm is detailed in the following paragraphs.

Firstly, the blue color of the sky is due to Rayleigh scattering. As the light moves through the atmosphere, most of the wavelengths pass straight through. However, much of the shorter wavelength light is absorbed by the gas molecules.¹¹ The absorbed blue light is then radiated in different directions, and, the sky looks blue. It gets scattered all around the sky and contributes to the illumination of objects in the natural shadow (i.e., hidden from the sun). This scenario assumes that shadows are mainly illuminated by the sky light.¹¹ Consequently, in a natural environment, shadows have high saturation in blue channels and low intensity. Fortunately, bee are not blue, but rather brown, orange or yellow (in term of the pixel values). Thus, the white background remains very useful because shadows can be easily removed when color pixel contains more blue than the other colors.

The main features computed by the method of Magnier et al.⁶ correspond to ellipses.¹² These ellipses are fitted from the edges extracted from the bees, after the thin edges have been thresholded.¹³ To detect the targets (bees), the algorithms computes a difference between I_t and I_M , a median image of several previous images (for example 30 previous frames). Unfortunately, shadows are always present and some parts remain visible in the background, resulting in the creation of undesirable ellipses with the difference of images. Thus, the method developed in⁶ considers the difference between I_t and I_M , removes pixels that are mostly blue and pixels with a high variance V_c between each color channels:

$$V_c(x, y) = \frac{1}{n} \cdot \sum_{i=1}^n (c_i(x, y) - \mu(x, y)), \quad (1)$$

with

$$\mu(x, y) = \sum_{i=1}^n \frac{c_i(x, y)}{n}, \quad (2)$$

where (x, y) represents the pixel coordinates and n corresponds to the number of channels c_i (3 in our case: Red, Green and Blue). Finally, these algorithm correctly extracts bees in the image for pixels having a high variance V_c (brown, orange... pixels tied to the bee colors, and, removing white parts with small V_c , i.e., V_c less than a threshold) and no blue parts (i.e., shadows) followed by an image difference (illustrated in Fig. 2 (d)).

Once the bees are detected, the next step consists on extracting the movement of these bees throughout the video, from their appearance to their disappearance. Standard objects trackers fail to follow the bees^{14, 15} because of their numbers in the image. Moreover, they appear at different scales, turn around themselves and bees all look the same. The general idea proposed in Magnier et al.⁶ is to determine the assumed position of

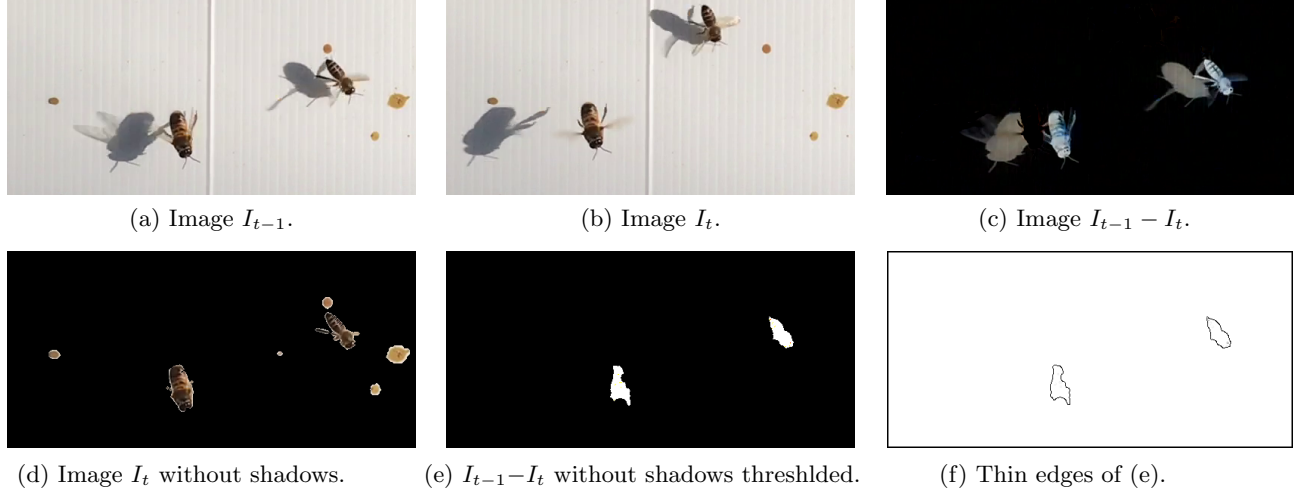


Figure 2. Example of tracked bees in front of the hive entrance.

bees in the next image. It works thanks to its positions on the two previous images. This assumed position is compared with all the bees on the image to determine which one fits a particular bee the best. To determine this assumed position, information about the bees in the previous images of the video is required, such as its coordinates and orientations tied to the corresponding ellipse issued of thresholded thin edges, as in Fig. 2 (f).

Considering one detected bee represented by an ellipse at time t : ϵ_t , the bee trajectory for the $t + 1$ frame can be estimated by knowing the coordinates of the two previous positions of the bee. Indeed, represented by the center and the orientation of an ellipse, at time $t - 1$: ϵ_{t-1} , the distance the bee travels for the frame at time $t + 1$ can be estimated, and, the direction it moves \mathcal{D}_A is estimated thanks to the orientation of ϵ_{t-1} and ϵ_t . Consequently, if Θ_{t-1} is the angle of ϵ_{t-1} , and Θ_t of ϵ_t , by computing the angle difference $\Delta\Theta$, we can assume that the ellipse ϵ_A at time $t + 1$ will shift in the direction \mathcal{D}_A . This direction corresponds to \mathcal{D}_t the movement direction between ϵ_{t-1} and ϵ_t , adjusted by $\Delta\Theta$:

$$\begin{cases} \Delta\Theta = \Theta_{t-1} - \Theta_t \\ \mathcal{D}_A = \mathcal{D}_t + \Delta\Theta \pmod{2\pi}. \end{cases} \quad (3)$$

Thereafter, with \mathcal{D}_A and the pixel distance between ϵ_{t-1} and ϵ_t , the assumed position of ϵ_A is estimated. Calling (X_{t-1}, Y_{t-1}) , and (X_t, Y_t) , respectively the coordinate of the ellipse centers at time t and $t - 1$, thus, (X_A, Y_A) , the assumed coordinate of ϵ_A are:

$$\begin{cases} X_A = X_t + (X_t - X_{t-1}) \cdot \sin(\mathcal{D}_A) \\ Y_A = Y_t + (Y_t - Y_{t-1}) \cdot \cos(\mathcal{D}_A). \end{cases} \quad (4)$$

In the following step, the assumed ellipse ϵ_A is compared with all the real ellipses of the image at time $t + 1$. For each real center of an ellipse, the distance between the center of ϵ_A is calculated. The detected ellipse ϵ_{t+1} at time $t + 1$ related to ϵ_t corresponds to which one has the minimal distance with the point at (X_A, Y_A) , that corresponds to a concordance \mathcal{C} :

$$\begin{cases} \mathcal{C} = \min \sqrt{(X_A - X_D)^2 + (Y_A - Y_D)^2} \\ (X_{t+1}, Y_{t+1}) = \arg \min_{(X_D, Y_D)} \sqrt{(X_A - X_D)^2 + (Y_A - Y_D)^2}, \end{cases} \quad (5)$$

where (X_D, Y_D) represents the coordinates of the center of the detected ellipse in the image at time $t + 1$ and (X_{t+1}, Y_{t+1}) the coordinates of the center of ϵ_{t+1} . This enable a tracking of bees with consistence.

Also, if ϵ_t has no ellipse corresponding at time $t + 1$, thus the bee is considered as moved out of the screen. When an ellipse at time $t + 1$ has no corresponding ellipse from time t , it is assimilated as a new bee entering the screen. Finally, the higher the frame rate is important the most accurate this method is.⁶

3. INTERPOLATION METHODS

For each bee, the trajectories have to be plotted from a list of points representing the bee positions $\{M(t), t \in T\}$ where $T = [t_0; t_1; \dots; t_n]$ is a list of floats representing the time of each position. Concretely, these positions corresponds to the centers of the ellipses. Actually, the list of point is separated into two lists of floats $\{x(t), t \in T\}$ and $\{y(t), t \in T\}$ where $x(t)$ and $y(t)$ represent respectively the abscissae and the ordinate of the bee at time t . Considering this abscissae and this ordinate, two functions f and g are calculated (by different ways depending on the interpolation method) such as $\forall t \in T, f(t) = x(t)$ and $g(t) = y(t)$. Then, the interpolated curve can be plotted from the new calculated list of points $(f(t), g(t))$ in order to estimate the trajectory.

Firstly, three interpolation methods are presented in an unidimensional context: considering a list of points $\{(x_0; y_0); (x_1; y_1); \dots; (x_n; y_n)\}$ which has to be interpolated by a function f . In other words, it is necessary that $\forall i \in \{0; 1; \dots; n - 1\}, f(x_i) = y_i$. There exists several methods to estimate such a function f .

3.1 Linear Interpolation

3.1.1 Principle

This method consists in linking points by segments. Indeed, let us note plotted segments $[(x_i; y_i); (x_{i+1}; y_{i+1})]$, with $i \in \{0; 1; 2; \dots; n - 1\}$.

3.1.2 Mathematical aspect

The interpolation function f is a piecewise affine function on each segment $[x_i; x_{i+1}]$ such that:

$$\forall i \in \{0; 1; \dots; n - 1\}, \quad \forall x \in [x_i; x_{i+1}], \quad f(x) = f(x_i) + \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}(x - x_i).$$

This represents a very simple method but, unfortunately, it cannot afford a natural looking to the curve, as illustrated in Fig. 3 . So this approach is not appropriate for the approximation of be trajectories.

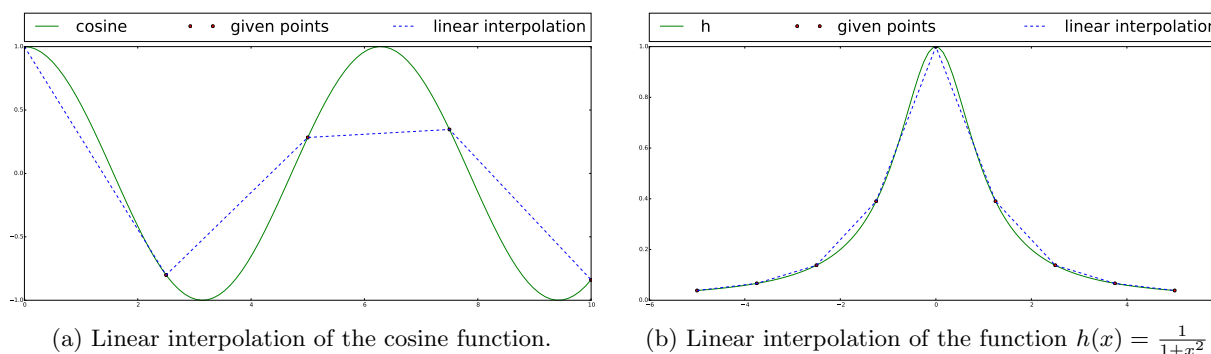


Figure 3. Linear interpolations of 1D functions.

3.2 Lagrange polynomial

3.2.1 Principle

This method estimates the unique polynomial of degree n which passes through $n + 1$ given discrete points.^{16,17} The main problem is that the approximated interpolation function may oscillate a lot between each point.

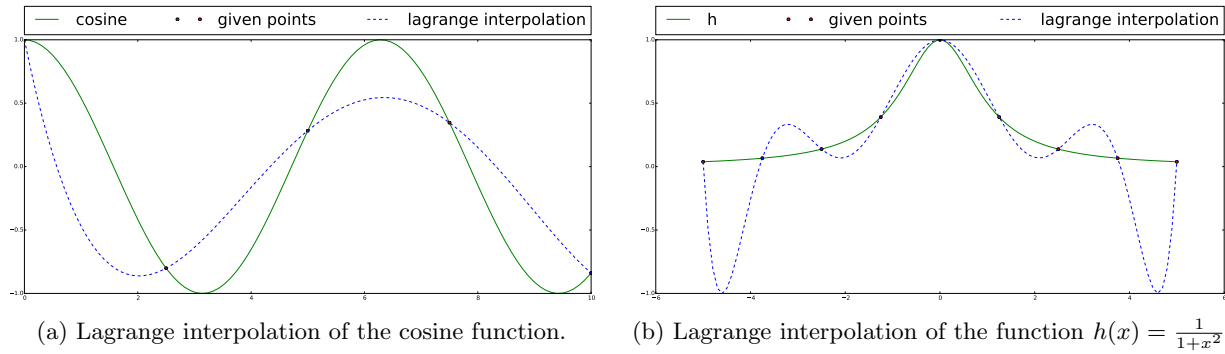


Figure 4. Lagrange interpolations of 1D functions.

3.2.2 Mathematical aspect

The interpolation function f represents a polynomial of degree n constructed with the following method:

$$\forall x \in [x_0, x_n], \quad f(x) = \sum_{i=0}^n y_i \cdot l(x),$$

where the l_i are Lagrange basis polynomials such that:

$$\forall i \in \{0, 1, \dots, n\}, \quad l_i(x) = \prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j}$$

Lagrange basis polynomials are constructed in order to tied 1 for one x_i and 0 on all the others given points:

$$\begin{cases} \forall i \in \{0, 1, \dots, n\}, & l_i(x_i) = 1 \\ \forall j \in \{0, 1, \dots, n\}, j \neq i, & l_i(x_j) = 0 \end{cases}$$

Unfortunately, as pointed out above, the Lagrange polynomial interpolation function may oscillate a lot between each point, as illustrated in Fig. 4.

3.3 Interpolation by cubic splines

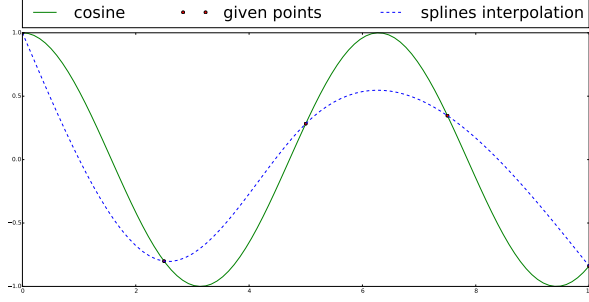
3.3.1 Principle

Considering this method, interpolated points are linked by polynomials of degree 3. So each point is linked between two of these polynomials.¹⁰ In order to get a more natural-looking curve, the first and second derivatives of each successive polynomials are linked. The first derivative represents the slope of the tangent at a given point of the curve whereas the second derivative is tied to the curve convexity. Consequently, these two derivatives creates continuities along the whole curve (thus avoiding discontinuities at level of detected/given points). This affects the looking of the curve in avoiding changes of tangent and convexity on the given points, as in Fig. 5.

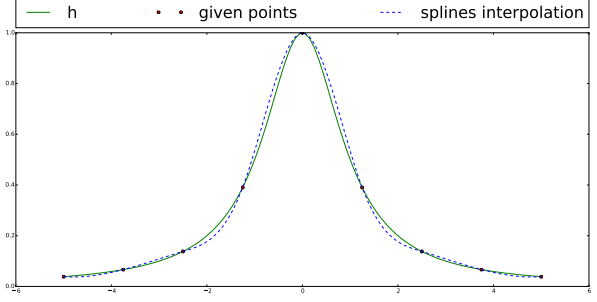
3.3.2 Mathematical aspect

The interpolation function f is constructed by pieces. Each successive point is linked by polynomials of degree 3. In order to get a function $f \in C^2$, the first and second derivatives are linked on each given point. To be able to plot the interpolation function, each of every 3rd order single polynomial $S_i(x)$ has to be determined, linking the points $(x_i, y_i), i \in \{0, 1, \dots, n-1\}$. For each $x \in \{x_0, x_1, \dots, x_{n-1}\}$, these polynomials are of the form:

$$S_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i$$



(a) Cubic splines interpolation of the cosine function.



(b) Cubic splines interpolation of the function $h(x) = \frac{1}{1+x^2}$

Figure 5. Cubic splines interpolations of 1D functions.

The global interpolation function can be piecewise defined as:

$$f(x) = \begin{cases} S_0(x - x_0), & \text{if } x \in [x_0, x_1] \\ S_1(x - x_1), & \text{if } x \in [x_1, x_2] \\ \dots \\ S_i(x - x_i), & \text{if } x \in [x_i, x_{i+1}] \\ \dots \\ S_{n-1}(x - x_{n-1}), & \text{if } x \in [x_{n-1}, x_n]. \end{cases}$$

So there are $4n$ unknown values to be determined:

$$\begin{cases} a_i \\ b_i \\ c_i \\ d_i \end{cases}, \quad \forall i \in \{0, 1, \dots, n-1\}$$

This process is the same for each $y \in \{y_0, y_1, \dots, y_{n-1}\}$. Each polynomial must link (x_i, y_i) to (x_{i+1}, y_{i+1}) . Consequently, $2n$ equations are given by the linking of points:

$$\forall i \in \{0, 1, \dots, n-1\}, \quad \begin{cases} S_i(0) = y_i \\ S_i(x_{i+1} - x_i) = y_{i+1} \end{cases}$$

$n-1$ equations given by the linking of the derivatives:

$$\forall i \in \{0, 1, \dots, n-2\}, \quad S'_i(x_{i+1} - x_i) = S'_{i+1}(0)$$

finally, $n-1$ equations given by the linking of the second derivatives :

$$\forall i \in \{0, 1, \dots, n-2\}, \quad S''_i(x_{i+1} - x_i) = S''_{i+1}(0)$$

Consequently, two equations are missing to get the $4n$ equations needed to determine the unknown values. These missing equations correspond to conditions on points at the limits, they are the values of the second derivative on the first and the last points ($S''_0(0)$ and $S''_{n-1}(x_n - x_{n-1})$). As the context does not require specific values for these conditions, natural cubic splines will be used. In other words, the splines are calculated such that: $S''_0(0) = 0$ and $S''_{n-1}(x_n - x_{n-1}) = 0$.

3.4 Implementation

In the presented experiments, the code is developed in Python language. In this context, although some functions exist in Python language to calculate an function of interpolation from a list of points using cubic splines, these Python functions do not use natural cubic splines (the second derivative of the points at the ends are set at

values different from 0). So, they do not fit the points as desired on the beginning and the end of the curves. In addition, mathematical equations of the plotted curves are necessary for the modeling and for the next step of the project; this cannot be given by the native Python functions. Therefore, a known algorithm is used in order to master conditions at the extremal points and retrieve the coefficients (so the equations) of the calculated cubic splines. Fig. 6 illustrates this phenomenon.

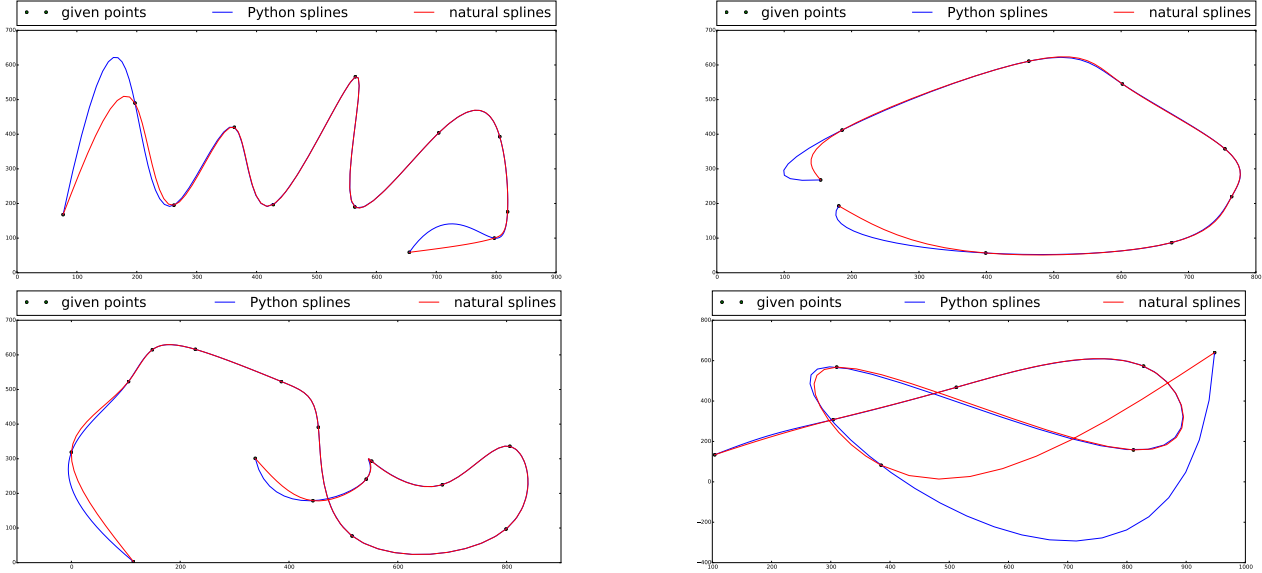


Figure 6. Comparisons between natural splines and native Python splines

As the previous images show, the Python splines do not create enough direct trajectories on the points at the ends which are less likely close to the original paths of the bees.

4. EXPERIMENTAL RESULTS ON A REAL VIDEO

Concerning the bee detection, the algorithm are coded in Python with the use the OpenCv library. The video run at 60 frames per seconds (fps) in 1080p, such a definition of 1920×1080 of pixels. The coordinates of the bees and the fitted curves are plotted together in Figs. 7, 8 and 9.

4.1 Bee 1

The first detected trajectory of a bee has been recorded between two points. That corresponds to a straight line, so the trajectory equation is tied to a line equation:

$$S_{x0}(t) = 0.0 \cdot t^3 - 1.3 \cdot t + 953.0$$

for the abscissae, and

$$S_{y0}(t) = 0.0 \cdot t^3 - 12.6 \cdot t + 890.0$$

for the ordinate. This corresponds to a polynomial of order 1 and is plotted in Fig. 7.

4.2 Bee 2

The second trajectory is composed of 18 detected coordinates in the image. Thus, the obtained polynomial corresponds in abscissae:

$$S_{x0}(t) = -13.1320651384 \cdot t^3 - 146.756032824 \cdot t + 1064.0$$

$$S_{x1}(t) = 26.3563256922 \cdot t^3 - 23.1742325973 \cdot t^2 - 160.387934352 \cdot t + 975.0$$

$$S_{x2}(t) = -13.6852376305 \cdot t^3 + 23.336930389 \cdot t^2 - 160.292229768 \cdot t + 878.0$$

$$S_{x3}(t) = 18.5586248298 \cdot t^3 - 0.813488958898 \cdot t^2 - 147.043146574 \cdot t + 789.0$$

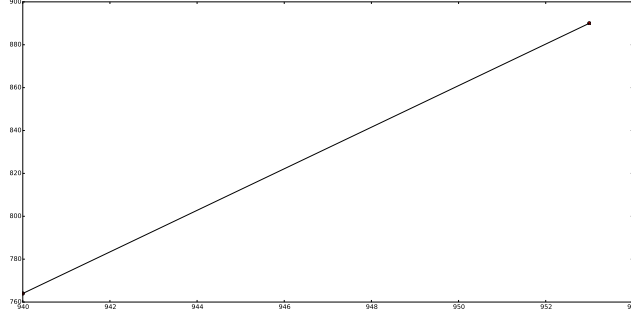


Figure 7. Example of a trajectory between 2 points (i.e., a line).

$$\begin{aligned}
S_{x4}(t) &= 27.8847383115 \cdot t^3 + 31.9370254466 \cdot t^2 - 128.735183934 \cdot t + 706.0 \\
S_{x5}(t) &= -31.8375780756 \cdot t^3 + 81.1453871727 \cdot t^2 - 62.2161176879 \cdot t + 647.0 \\
S_{x6}(t) &= -3.70742600889 \cdot t^3 + 24.9614258627 \cdot t^2 + 0.199654685906 \cdot t + 632.0 \\
S_{x7}(t) &= 12.2762821112 \cdot t^3 + 18.4189093764 \cdot t^2 + 25.7174989442 \cdot t + 640.0 \\
S_{x8}(t) &= -15.9197024359 \cdot t^3 + 40.0829366315 \cdot t^2 + 60.1303495371 \cdot t + 664.0 \\
S_{x9}(t) &= -12.4664723676 \cdot t^3 + 11.9893440976 \cdot t^2 + 90.7611029072 \cdot t + 710.0 \\
S_{x10}(t) &= 6.82959190626 \cdot t^3 - 10.0103130217 \cdot t^2 + 91.9252388341 \cdot t + 765.0 \\
S_{x11}(t) &= 4.80010474254 \cdot t^3 + 2.04190798933 \cdot t^2 + 87.2379417563 \cdot t + 817.0 \\
S_{x12}(t) &= -11.2910108764 \cdot t^3 + 10.5126810644 \cdot t^2 + 94.6229941408 \cdot t + 870.0 \\
S_{x13}(t) &= 1.05993876311 \cdot t^3 - 9.41263224692 \cdot t^2 + 95.2700816805 \cdot t + 927.0 \\
S_{x14}(t) &= 7.05125582396 \cdot t^3 - 7.54215207672 \cdot t^2 + 85.2966791372 \cdot t + 980.0 \\
S_{x15}(t) &= 5.12603794106 \cdot t^3 + 4.90124055379 \cdot t^2 + 83.7432017708 \cdot t + 1029.0 \\
S_{x16}(t) &= -7.90340758821 \cdot t^3 + 13.9471898616 \cdot t^2 + 94.8305137798 \cdot t + 1081.0
\end{aligned}$$

and in ordinate:

$$\begin{aligned}
S_{y0}(t) &= -17.3742070361 \cdot t^3 + 123.311836345 \cdot t + 69.0 \\
S_{y1}(t) &= 8.26303518045 \cdot t^3 - 30.6603653578 \cdot t^2 + 105.276327311 \cdot t + 138.0 \\
S_{y2}(t) &= 3.97406631428 \cdot t^3 - 16.0785385688 \cdot t^2 + 77.7828544127 \cdot t + 191.0 \\
S_{y3}(t) &= -4.50730043757 \cdot t^3 - 9.0654803671 \cdot t^2 + 62.9922550386 \cdot t + 232.0 \\
S_{y4}(t) &= 4.22913543601 \cdot t^3 - 17.0195399628 \cdot t^2 + 47.6481254328 \cdot t + 265.0 \\
S_{y5}(t) &= 7.24275869355 \cdot t^3 - 9.55635978163 \cdot t^2 + 32.0152432302 \cdot t + 288.0 \\
S_{y6}(t) &= -3.7221702102 \cdot t^3 + 3.22497908934 \cdot t^2 + 28.2909016465 \cdot t + 305.0 \\
S_{y7}(t) &= 7.64592214724 \cdot t^3 - 3.34355657572 \cdot t^2 + 28.2211501839 \cdot t + 322.0 \\
S_{y8}(t) &= -12.1225183788 \cdot t^3 + 10.1492472135 \cdot t^2 + 32.2244976179 \cdot t + 339.0 \\
S_{y9}(t) &= 11.3661513678 \cdot t^3 - 11.2434322784 \cdot t^2 + 31.5808593445 \cdot t + 359.0 \\
S_{y10}(t) &= 5.96191290755 \cdot t^3 + 8.81448190008 \cdot t^2 + 30.1520650043 \cdot t + 376.0 \\
S_{y11}(t) &= -10.648802998 \cdot t^3 + 19.3355046781 \cdot t^2 + 46.7108806385 \cdot t + 398.0 \\
S_{y12}(t) &= 7.15529908439 \cdot t^3 + 0.543499387535 \cdot t^2 + 58.4044124418 \cdot t + 430.0 \\
S_{y13}(t) &= -3.23339333959 \cdot t^3 + 13.1704977718 \cdot t^2 + 66.4714695943 \cdot t + 466.0 \\
S_{y14}(t) &= -8.96072572602 \cdot t^3 + 7.46450952542 \cdot t^2 + 78.6097091809 \cdot t + 509.0 \\
S_{y15}(t) &= -0.227703756309 \cdot t^3 - 8.34853587345 \cdot t^2 + 78.0896936821 \cdot t + 556.0 \\
S_{y16}(t) &= 4.95854075126 \cdot t^3 - 8.75036603164 \cdot t^2 + 68.0315160908 \cdot t + 599.0.
\end{aligned}$$

This corresponds to a polynomial of 3rd degree and is plotted in Fig. 8.

4.3 Bee 3

The trajectory trajectory is composed of 5 detected coordinates in the image. Thus, the obtained polynomial corresponds in abscissae:

$$S_{x0}(t) = 0.169142857143 \cdot t^3 - 19.0571428571 \cdot t + 736.0$$

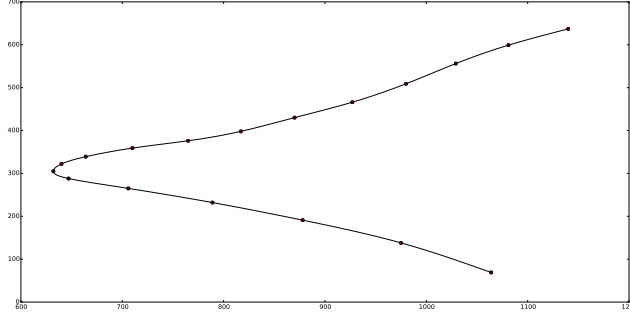


Figure 8. Example of a trajectory between 18 points.

$$\begin{aligned}
 S_{x_1}(t) &= 0.178285714286 \cdot t^3 + 1.26857142857 \cdot t^2 - 15.8857142857 \cdot t + 691.0 \\
 S_{x_2}(t) &= -0.178285714286 \cdot t^3 + 2.60571428571 \cdot t^2 - 6.2 \cdot t + 662.0 \\
 S_{x_3}(t) &= -0.169142857143 \cdot t^3 + 1.26857142857 \cdot t^2 + 3.48571428571 \cdot t + 660.0
 \end{aligned}$$

and in ordinate:

$$\begin{aligned}
 S_{y_0}(t) &= -3.66057142857 \cdot t^3 + 56.8785714286 \cdot t + 784.0 \\
 S_{y_1}(t) &= 4.86285714286 \cdot t^3 - 27.4542857143 \cdot t^2 - 11.7571428571 \cdot t + 869.0 \\
 S_{y_2}(t) &= -1.71085714286 \cdot t^3 + 9.01714285714 \cdot t^2 - 57.85 \cdot t + 744.0 \\
 S_{y_3}(t) &= 0.508571428571 \cdot t^3 - 3.81428571429 \cdot t^2 - 44.8428571429 \cdot t + 629.0
 \end{aligned}$$

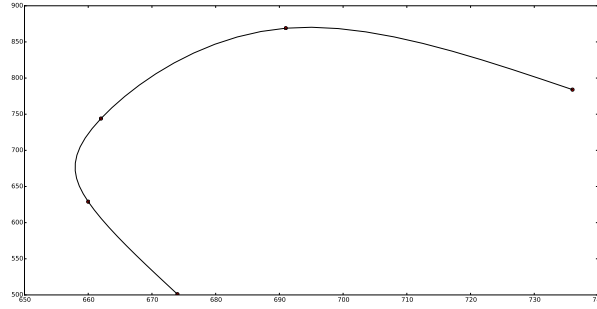


Figure 9. Example of a trajectory between 5 points.

These results underline that the cubic splines computation for the bee trajectories allows to construct polynomials and to plot fitted curves without stochastic behaviors at the extremal points.

4.4 Conclusion

In this paper, an interpolation method is proposed to model the flight trajectories of bees and then characterize the pace of the global activity in front of the hive to infer observations that will be consolidated and made available to apicultural data scientists. Thus, bees are detected and tracked using image and video processing methods, then the trajectories are modeled by interpolation using natural cubic splines. Consequently, the plotted curves show the consistency of the trajectories, especially at the extremal points (beginning and ending).

Several issues have been faced about the interpolation methods of the bees trajectories, and it was found that the cubic lines plotting method must be the best way to proceed. As a matter of fact, the curves created almost overlap the real trajectories and the loss of data is irrelevant considering the behavioral studies that follow. Indeed, either the collected polynomial or the shape of the plotted line thanks to semi-supervised learning or artificial neural network respectively will help machine learning algorithms give conclusions about the bees behavior.

REFERENCES

- [1] Storch, H. and Mosbeux, J., [*At the Hive Entrance: OBSERVATION HANDBOOK. How to know what happens inside the hive by observation on the outside*], European Apicultural Editions (1985).
- [2] Babic, Z., Pilipovic, R., Risojevic, V., and Mirjanic, G., “Pollen bearing honey bee detection in hive entrance video recorded by remote embedded system for pollination monitoring,” *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences* **3**(7) (2016).
- [3] Campbell, J., Mummert, L., and Sukthankar, R., “Video monitoring of honey bee colonies at the hive entrance,” *Visual observation & analysis of animal & insect behavior, ICPR* **8**, 1–4 (2008).
- [4] Kulyukin, V. A., “In situ omnidirectional vision-based bee counting using 1d haar wavelet spikes,” in [*Proceedings of the International MultiConference of Engineers and Computer Scientists*], **1** (2017).
- [5] Tu, G. J., Hansen, M. K., Kryger, P., and Ahrendt, P., “Automatic behaviour analysis system for honeybees using computer vision,” *Computers and Electronics in Agriculture* **122**, 10–18 (2016).
- [6] Magnier, B., Ekszterowicz, G., Laurent, J., Rival, M., and Pfister, F., “Bee hive traffic monitoring by tracking bee flight paths,” in [*13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, January 27-29, 2018, in Funchal, Madeira, Portugal*], 563–571 (2018).
- [7] Chiron, G., Gomez-Krämer, P., and Ménard, M., “Detecting and tracking honeybees in 3d at the beehive entrance using stereo vision,” *EURASIP Journal on Image and Video Processing* **2013**(1), 59 (2013).
- [8] Tashakkori, R. and Ghadiri, A., “Image processing for honey bee hive health monitoring,” in [*SoutheastCon*], 1–7, IEEE (2015).
- [9] Kimura, T., Ohashi, M., Crailsheim, K., Schmickl, T., Okada, R., Radspieler, G., and Ikeno, H., “Development of a new method to track multiple honey bees with complex behaviors on a flat laboratory arena,” *PloS one* **9**(1), e84656 (2014).
- [10] McKinley, S. and Levine, M., “Cubic spline interpolation,” *College of the Redwoods* **45**, 1049–1060 (1998).
- [11] Adeline, K. R. M., Chen, M., Briottet, X., Pang, S. K., and Paparoditis, N., “Shadow detection in very high spatial resolution aerial images: A comparative study,” *ISPRS Journal of Photogrammetry and Remote Sensing* **80**, 21–38 (2013).
- [12] Contours and Ellipses, *OpenCV function: Creating Bounding rotated boxes and ellipses for contours*. http://docs.opencv.org/2.4/doc/tutorials/imgproc/shapedescriptors/bounding_rotated_ellipses/bounding_rotated_ellipses.html.
- [13] Canny, J., “A computational approach to edge detection,” *IEEE Transactions on pattern analysis and machine intelligence* (6), 679–698 (1986).
- [14] Viola, P. and Jones, M., “Rapid object detection using a boosted cascade of simple features,” in [*IEEE Computer Vision and Pattern Recognition*], **1**, I–I, IEEE (2001).
- [15] Bradski, G. R., “Real time face and object tracking as a component of a perceptual user interface,” in [*Fourth IEEE Workshop on Applications of Computer Vision*], 214–219, IEEE (1998).
- [16] Werner, W., “Polynomial interpolation: Lagrange versus newton,” *Mathematics of computation* , 205–217 (1984).
- [17] Berrut, J.-P. and Trefethen, L. N., “Barycentric lagrange interpolation,” *SIAM review* **46**(3), 501–517 (2004).