Master Thesis, Institute of Computer Science, Freie Universität Berlin

Biorobotics Lab, Intelligent Systems and Robotics

Detection of Honey Bee Dancers and Followers with Model-Based Machine Learning

Kadir Tugan
Matriculation number: 429 5229
kadir@tugan.net

Supervisor: Prof. Dr. Tim Landgraf, Freie Universität Berlin
Second Supervisor: Prof. Dr. Raúl Rojas, Freie Universität Berlin

Berlin, 04 May 2017

Abstract

The waggle dance patterns of the western honey bee are well understood and researched. New methods in computer vision systems allow long-term tracking of individual bees to capture the spatiotemporal position of each bee in the hive. Previous research on the detection of the waggle dance primarily focuses on videos with a high frame rate. The few currently known methods on spatiotemporal data require a high temporal resolution of at least 13 Hz to capture the waggle behavior part. Contrary to the dancer, research on the detection of waggle dance followers is not existent.

This thesis introduces a new model to detect waggle-dancers and their followers in spatiotemporal data with a low temporal resolution by using domain knowledge to engineer specific features that match these behaviors. We describe the model of the waggle-dance and their followers, the patterns behind it and the process to utilize this knowledge into specific features.

The proposed model allows the discovery of not only the waggle-dancers but its followers; the combination of long-term tracking and the detection enables further research into the relation between each bee in the colony.

Eidesstattliche Erklärung

Ich versichere hiermit an Eides Statt, dass diese Arbeit von niemand anderem als meiner Person verfasst worden ist. Alle verwendeten Hilfsmittel wie Berichte, Bücher, Internetseiten oder ähnliches sind im Literaturverzeichnis angegeben. Zitate aus fremden Arbeiten sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Berlin, den 4. Mai 2017

Kadir Tugan

Contents

1 Introduction

A honey bee colony has little-known command structure; each bee acts autonomously while working together on complex tasks otherwise not feasible alone. Nest construction, brood control, defense and division of labor are just a few examples of behaviors where organization and communication are paramount.

The waggle dance or the round-dance are examples of such a communication behavior, the bee performs either dance to convey the location information about a food source [1]. A bee discovering a food source returns to the hive and communicates the location by encoding it in the dance. Any bee following the dance, subsequently called followers, can decode the location and is able to find the advertised site. This thesis focuses its efforts on the waggle dance.

Our understanding of the honey bee is limited; there is very little knowledge about behaviors and interaction over a long time frame. The BeesBook Project[1] focuses on these gaps of knowledge and the discovery of short and long term behaviors.

The research requires the tracking of each bee to extract repeating patterns and thus behaviors. Traditional tracking approaches were composed of manual labeling over a very short time frame or on a very limited number of bees; a colony containing just a few thousand bees renders prolonged tracking nearly impossible. The BeesBook project solved this issue by developing a computer vision system to record the ID and the spatiotemporal position of each bee within a hive, at a frequency of 3 Hz over a period of nine weeks [2]. The computer vision system uses a probabilistic approach; the ID is not always perfectly decodable, a probability indicates the confidence. Further work transformed the probabilistic nature into an explicit trajectory for each bee [3–6]. The trajectory contains data about the position, orientation and ID of the bee at each time.

The low temporal resolution of the source material is a challenge when detecting behaviors; some parts of behaviors contain fast movements which are not covered by 3 Hz data. Two examples of these behaviors are the waggle dance with its waggle part where the bee moves its body from side to side at a frequency of about 13 Hz [7] and the stop signal which lasts about a tenth of a second and exhibits a vibration of about 320 Hz [8].

Previous research on detection on waggle dancers exclusively use this fast moving feature [2, 9–11]. This thesis presents another way to detect waggle dancers and their followers with a different set of defining features. The described approach uses intricate domain knowledge about the *dance* and *follow* behavior to characterize patterns. The patterns are subsequently used and combined to create a classification model. The model is evaluated

[1] A project by the BioRobotics lab of the Freie Universität Berlin.

against a manually labeled ground truth samples to determine the accuracy and error.
The proposed method allows the detection of waggle dancers and followers in low-frequency spatiotemporal data. Previous detection methods focused on the dancers; there are no known publications about automatic detection of followers. The automatic detection of dancers and followers, coupled with long term tracking, enables the research into dance-to-follower communication patterns over a long time frame.

2 Related Work

The following sections cover the related work from different aspects: the behavioral models of the *dance* and *follow* behavior, the detection of waggle dances on spatiotemporal and video data and the necessity of this work from a software perspective. There are to date no publications about the automatic detection of waggle-dance followers in either videos or trajectories.

2.1 Behavioral Model

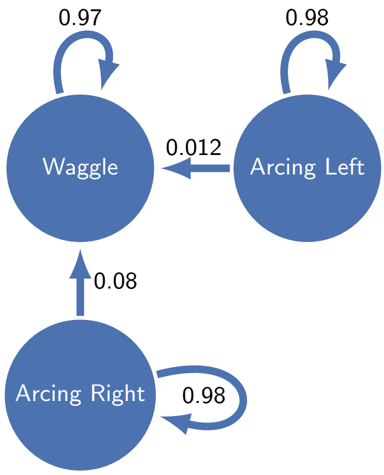
Karl von Frisch was the first to publish extensive research on the waggle dance. He describes in his publications “Aus dem Leben der Bienen” [12] and “The dance language and orientation of bees”[1] the spatiotemporal pattern of the waggle dance.
While there is a lot of excellent research continuing the work by Frisch, this thesis will focus on its most relevant research. The waggle dance pattern is further described by Landgraf et al. in “Analysis of the Waggle Dance Motion of Honeybees for the Design of a Biomimetic Honeybee Robot” [13] and explores different intrinsic properties. Different velocities, durations and frequencies are just a few of those properties; this thesis uses some of these properties to accurately model the dance and their followers.

The follower is further described by Božič and Valentinčič [14], and Grüter and Farina [15]. They describe the spatiotemporal pattern of the follower and the orientation and position of followers in relation to the dancer.

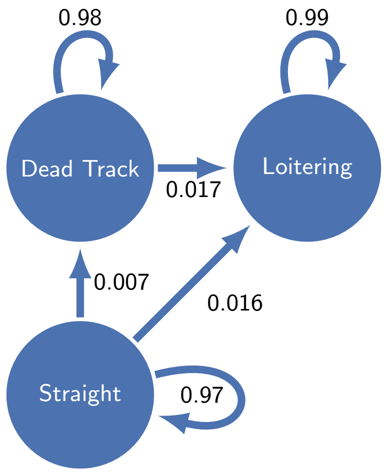
2.2 Waggle Dance Detection

There are two publications which relate to the detection of waggle dancers on spatiotemporal data and one publication which detects dancers on videos. The first and most similar work on spatiotemporal detection is by Feldman and Balch from 2003 published in “Automatic identification of bee movement” [9]. The second is by Kimura et al. in their publication “A new approach for the simultaneous tracking of multiple honeybees for analysis of hive behavior” [10].
The final result of the work by Feldman and Balch is the ability to differentiate between trajectories of dances and other behaviors. The main difference to this thesis is the higher resolution data of 30 Hz. They start by transforming the trajectory into a sequence of labels. Each label represents a different type of movement and is specifically feature engineered; the most prominent are the “arcing left”, “arcing right”, “waggle” and “straight” label. The sequence of labels is put into a hidden Markov model and the resulting transition matrix was split into two transition graphs by removing transitions with a probability of less than 0.005.

The first transition graph (see Figure 1a) could determine the probability of a dance given a sequence of labels; the second graph (see Figure 1b) did not represent any other behaviors.



(a) First transition graph. Shows the approximate state transition of a typical waggle-dance behavior.



(b) Second transition graph. Shows no apparent behavior.

Figure 1: Transition graphs from "Automatic identification of bee movement" by Feldman and Balch [9]. Extracted by removing all transitions with a prob. of 0.005 or lower from the hidden Markov model transition matrix.

The described approach is not applicable to this thesis; the resolution of 3 Hz prevents the extraction of aforementioned engineered features such as the waggle which oscillates at approximately 13 Hz or the arcing movement which in fast dances look like a line. The use of a hidden Markov model in our thesis is also not necessary and would add needless complexity.

The second publication by Kimura et al. [10] tracks bees in 30 fps videos and claims to detect waggle dances from these trajectories. The publication does not convey any information about the success or the approach of the dance detection. The method cannot be reproduced due to missing explanation in the methodology.

The detection on videos has been solved by Wario et al. [2]. They developed and published a system which is divided into two parts. Methods for automatic tracking in honey bees and the decoding of dances. The first part describes a system to detect the identity and position of each bee within three fps videos and produces the data for this thesis. The second part describes the detection of waggle dances and their locations within 120 fps videos without the ability to detect the identity of the dancer.

2.3 Existing Software Solutions

JAABA [16] is a general purpose tool to annotate animal behavior. It advertises their learning feature in which the tool learns from a few samples and their video and trajectories. The relation to this work is that choosing a working solution over something new is preferable. JAABA learns through human labeling with the initial learning attempt on a video, trajectory and set of labeled behaviors. It tries to learn when a specific input of frames defines a behavior. The result is applied either to the same video to find other unlabeled occurrences or to another video to present the results in a user interface. The researcher can then correct or add labels to improve the system. The learning phases are of iterating nature, the system predicts, the researcher corrects.
JAABA requires both the source video and the trajectories and is not optimized for large amounts of data in the order of hundreds of terabyte. The software is written in Matlab; integration into any pipeline would add large technical debt and is an issue. A goal of this thesis is the detection on trajectories alone to reduce the amount of data processed by a factor of thousand.

3 Implementation

This implementation section describes the process of modeling features to detect dancer and follower by using domain specific knowledge. The process is described in a linear fashion and explains each step and the decision behind it.
Section 3.1 gives an overview of the provided data and highlights its characteristics and challenges. The following section 3.2 illustrates the behavioral model and the interaction between a waggle-dancer and its followers. Section 3.3 describes the extraction of egocentric velocities from the spatiotemporal data to explain in section 3.4 the intrinsic patterns of each iteration of the dance and their followers.
Section 3.5 uses the described patterns to build a rank correlation function for a single point in time associated with the middle of the iteration. A *dancing* and *following* bee performs in most cases multiple iterations; section 3.6 describes the use of this knowledge to improve the correlation function by summarizing over a time window. The last section 3.7 finds the optimal time in the correlation function and designs a classifier from it.

3.1 Data

The data which this thesis relies on is a product of a series of contributions. A complete honey bee colony was observed for nine weeks. Each individual bee was tagged with an unique ID; high-resolution cameras recorded the honeycomb with 3 fps. A computer vision system processes the videos and decodes the ID and the location of each bee in a video-frame into probabilistic bits. The setup is described by Wario et al. [2] in “Automatic methods for long-term tracking and the detection and decoding of communication dances in honeybees”.
Further work improves the decoding of the ID with a mix of neural networks [4]. The tracking [5, 6] finalizes the process by combining the probabilistic IDs and their spatiotemporal position into trajectories of a single bee. The produced data is the foundation for this thesis.
The data is in a table format style from a trajectory/track perspective. Each track consists of several rows, in which each row conveys location information about a bee, see Table 1.

Each row contains the following information:

- **track_id**
  Unique ID indicating the track this position belongs to.
- **timestamp**
  Time the location was recorded as a Unix timestamp.

track_id	timestamp	x	y	orientation	bee_id
11	1471258925.79	1150	2287	-1.57	[0.1, 0.0, ... , 0.0, 0.0]
11	1471258926.13	1164	2287	-0.52	[0.0, 0.1, ... , 0.0, 0.0]
11	1471258925.46	1150	2288	-0.83	[0.9, 0.0, ... , 0.1, 0.0]
11	1471258924.13	1226	2237	-0.72	[0.0, 0.0, ... , 0.0, 0.0]
29	1471258924.13	2114	1275	-0.38	[0.0, 0.0, ... , 0.0, 1.0]
29	1471258924.47	2126	1276	-0.64	[1.0, 0.0, ... , 0.0, 0.0]
29	1471258924.80	2126	1289	-0.27	[0.0, 0.0, ... , 0.0, 1.0]
29	1471258925.13	2125	1289	-0.71	[0.1, 0.0, ... , 0.0, 0.0]
29	1471258925.46	2115	1325	-0.01	[0.0, 0.0, ... , 0.0, 1.0]
41	1471258927.45	2726	2515	1.66	[0.4, 0.4, ... , 0.3, 0.4]
41	1471258927.12	2714	2527	2.14	[0.5, 0.7, ... , 0.2, 0.6]
41	1471258924.80	2627	2577	2.31	[1.0, 0.7, ... , 0.0, 0.8]
41	1471258924.47	2563	2575	2.75	[1.0, 0.0, ... , 0.0, 1.0]
41	1471258924.13	2538	2562	2.71	[1.0, 0.0, ... , 0.0, 1.0]
50	1471258924.13	2375	1901	1.65	[0.8, 0.8, ... , 0.7, 0.4]

Table 1: Excerpt of the data

- **x and y**
  The location in pixels.
- **orientation**
  The direction the bee faces as an angle measured in radians.
- **bee_id**
  Twelve confidence probabilities representing each bit in the decoded ID of the bee. E.g. [0.7, 0.3, ...] → the first bit has a confidence of 0.7 to be a 1, the second a confidence of 0.3 and so on.

The detection on the data poses different challenges through errors or gaps in the trajectory. The following problems in the data either occur very often or pose a severe challenge in the detection.

- **Gaps in tracks**
  The track allows gaps of up to five seconds. The location of the bee can be unknown for this duration; a new track is generated if the bee is unknown for a more extensive time. The gaps allow for stabler and longer tracks where bees can be missing for a short time and occurs when the camera is not able to capture the tag of the bee. The gaps are an advantage in most cases; without it the smallest obstruction of the tag would lead to fragmentation of the track. The obstruction can either be by another bee covering the tag or when tracked bee moves into a cell or leaves the recording area. Gaps occur very often.

- **Tracking error**
  A track should only embody the trajectory of a single bee. Low confidence values in the ID can lead to false concatenation of observations into a track. The trajectory jumps from one bee to another. The error occurs rarely.

- **Jumps in the orientation**
  Some circumstances lead to erroneous jumps in the orientation by up to 180 degrees. The error occurs rarely.

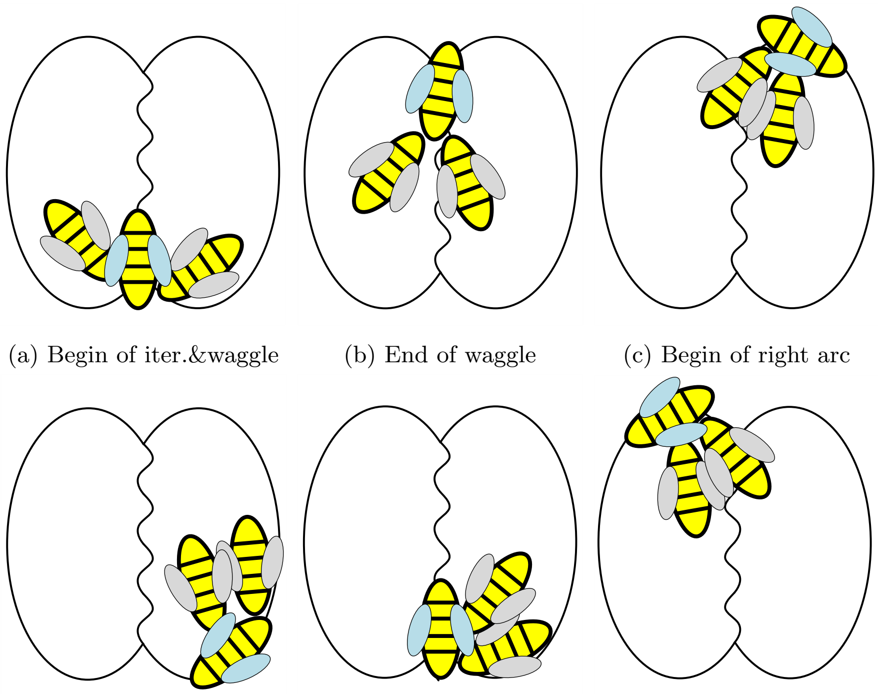
Ground Truth Data

Evaluating any model requires ground truth data to calculate the false positive and the false negative error. Evaluating just the positive classified (dancers and followers) would neglect the false negative error; we would have no knowledge about dancers and followers which we did not find with our model. Measuring each type of error requires full knowledge in a limited time frame.

We labeled the behavior of each bee in a time frame of ten minutes into the three classes: *dancer*, *follower* or *other*. The labeling process consisted of two steps. Generating a video for each track within the ten minutes and the review thereof to determine the behavior of the bee in the track. Labeling ten minutes of ground truth data equaled to a review of over 29 hours of video.
Each label is for simplification purposes track based, a track is labeled either *dancer* or *follower*. The label is only given if the track is error free and with a full iteration of the behavior present.

3.2 Model of a Dancer and Follower

A bee discovering a food source can convey the location information to its fellow bees. Bees interested and motivated in learning the location participate in the dance and receive the information by closely following the movement of the dancer. The dance acts as a medium to communicate. There are currently two known types of dances, the waggle-dance and the round-dance. We focus in this thesis only on the waggle-dance.
The dancer and follower have each its own movement patterns which are relative to each other. Frisch was the first to extensively research and describe the waggle-dance in his book from 1927 "Aus dem Leben der Bienen"[12]. Research about the follower bees are less extensive. Božič and Valentinčič well describe the patterns of the followers in their publication "Attendants and followers of honey bee waggle dances" [14].
This section will summarizes their research by explaining and illustrating the sequence of patterns in each behavior.



(a) Begin of iter.&waggle (b) End of waggle (c) Begin of right arc

(d) End of right arc (e) Middle of iteration (f) Begin of left arc

Figure 2: The typical behavioral pattern of a waggle dancer and their followers. The blue winged bee is a dancing bee, the gray winged are follower bees.

The dance consists of several iterations. Each iteration can be split into four rough steps and together form a shape similar to an *eight*. Figure 2 illustrates the pattern in more detailed steps. The patterns of the followers are explained in conjunction with the dancer patterns since they relate to each other.

1. **The waggle**
   The dancer bee swings her abdomen from side to side with a frequency of about 13 Hz while moving forward with a small velocity. A follower bee joining the dance can be on either side of the dancer facing the abdomen (see Figure 2a) and starts to trail the dancer. The followers continue to face the abdomen at the end of the waggle but are positioned more behind than to the side. The transition from Figure 2a to 2b illustrates this.

2. **Arc movement in either direction**
   The dancer rotates into either a right or a left direction and returns in an arcing movement, similar to a half circle, approximately to the beginning of the iteration. Figure 2c, 2d and 2e illustrates this arcing movement. All followers, if previously not, are now on the same side

of the dancer while continuously facing the flank of the dancer. This is achieved by moving sidewards while slightly rotating.

3. **The waggle**
   This step is the same as step one. Each follower is now on the same side of the dancer.

4. **Arc movement into opposite direction**.
   The dancer makes an arcing movement into the direction opposite to step two (see Figure 2f). Everything else is same as step two.

A dancer performs typically multiple iterations of the above mentioned four steps. The number of iterations for a follower has a higher variance, sometimes performing for just a single or half iteration.

3.3 Extraction of Trajectories and Egocentric Movement

The patterns mentioned in the previous section describes the dancing or following from an egocentric perspective; simple coordinates or the vector between is not enough. The extraction of egocentric movements is required. The data in its source form yields only the position and the orientation of the bee at time t; this spatiotemporal information can be transformed into the following three velocities from an egocentric perspective (see Figure 3).

1. **Forward velocity**
   The bee moves forward/straight.

2. **Sideward velocity**
   The bee moves sideways.

3. **Turn velocity**
   The bee rotates/turns.

Turn Velocity

The turn velocity, also called angular velocity, is calculated by subtracting the orientation at time o_t from the previous o_{t-1}:

$$o'_t = o_t - o_{t-1}.$$

The values range in [-2π,2π] and are not in a comparable order due to the nature of angles. A simple calculation further explains this nature. The following two angular movements are the same, a 40° clockwise movement and still differ in result: $90° - 50° = 40°$ and $10° - 330° = -320°$. The comparability issue is resolved using the arctan2 in combination with sin and cos and results in the turn velocity:

$$\text{turnvelocity}_t = \text{arctan2}(sin(o'_t), cos(o'_t)).$$

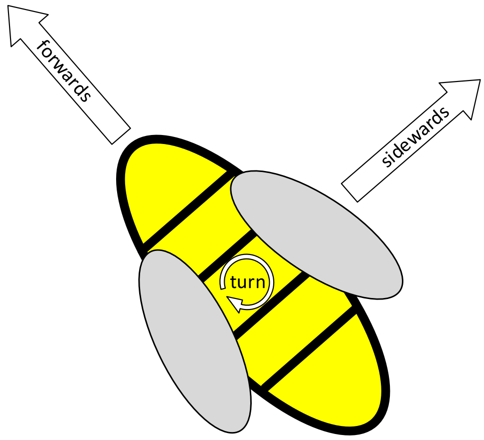


Figure 3: Forward, sideward and turn velocity

Forward and Sideward Velocity

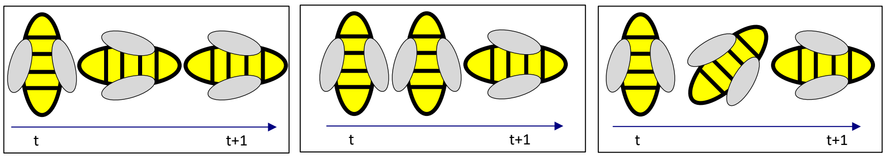
The calculation of the forward and sideward velocity requires trajectories which we can obtain by subtracting the position at time x_t and y_t from the previous position at time x_{t-1} and y_{t-1}:

$$x'_t = x_t - x_{t-1}$$
$$y'_t = y_t - y_{t-1}$$

The vectors x'_i and y'_i are not from an egocentric perspective and need to be rotated by the orientation o_i. Each x' and y' at time t form together a vector and can be rotated resulting in the forward and sideward velocity.

$$\text{forwardvelocity}_t = x'_t * cos(o_t) - y'_t * sin(o_t)$$
$$\text{sidewardvelocity}_t = x'_t * sin(o_t) + y'_t * sin(o_t)$$

It is important to note that these calculations are just an approximation. The rotation of the trajectory is based on the orientation at time t, but it is not known when the rotation occurred. Does the bee rotate first and then move (Fig. 4a), move and then rotate (Fig. 4b) or a mixture of both (Fig. 4c). Each version has different forward and sideward velocities.
The most reasonable assumption is a mixture of both where the angular velocity happens in a linear fashion, see Figure 4c. This can be achieved by rotating the vector with the averaged angle between o_t and o_{t-1}.
This thesis uses the second mentioned variant (see Figure 4b) of egocentric velocity calculation to keep it simple. Any of the mentioned variants approximate for our purposes good enough. The chosen implementation yields sufficient results to detect dancers and followers.



(a) Major forward movement (b) Major sideward movement (c) Mix of sideward and forward movement

Figure 4: Three different variants of egocentric velocity approximation.

3.4 Patterns

The waggle part of the dance is the most defining pattern to detect or distinguish it from other behaviors. It consists of a slight forwards movement while throwing its body from side to side in a fast fashion and looks to the human eye like a vibration. This *vibration* part is the defining feature and is not captured by the 3 Hz data used in this thesis.
The follower behavior does not exhibit such an outstanding feature, but there are several other patterns appearing in either behavior:

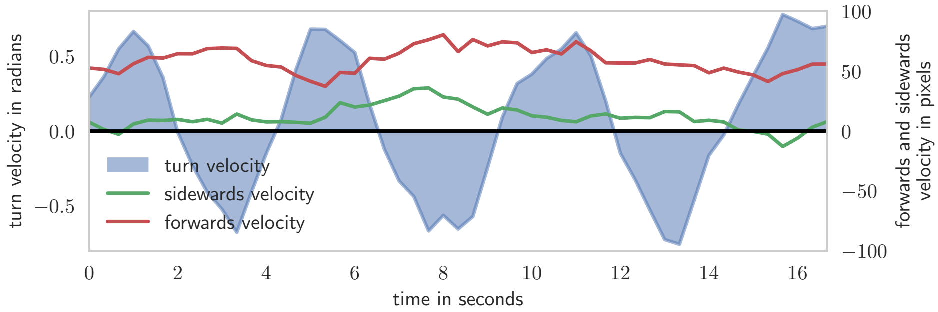
High Velocity

An indicator of both the *dancing* and the *following* behavior is the increased velocity, most bees in the hive move very slowly with a median velocity of 2.3mm/s while dancers and followers move at a median velocity of 10.5mm/s. The same also applies to the turn velocity with an overall average of 34°/s, or 213°/s for dancers and followers.

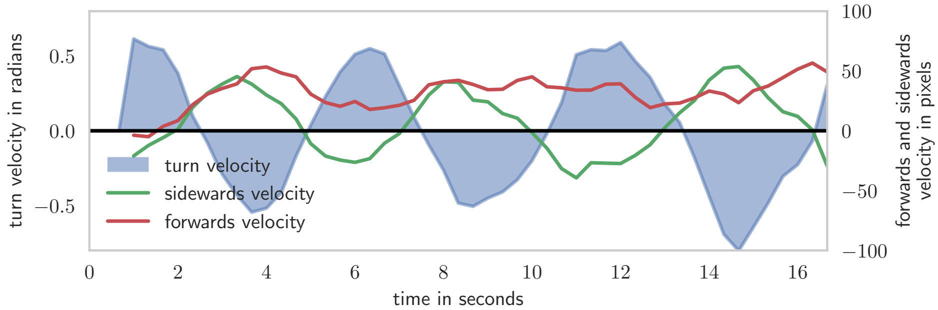
Lemniscate Shape

The trajectory of both behaviors matches a lemniscate shaped pattern, which resembles the infinity symbol, also shown in Figure 2. The shape can be in some iterations skewed or different depending on how crowded the hive is; other bees may block or alter the path. A commonly observed deviation is that one arc movement has a smaller circumference. Another less common observation is a repeated single arc movement into the same direction as before with the subsequent return to their regular lemniscate shape. Our data does only support fragile detection for this pattern. The dancer bee moves in some cases too fast, resulting in a triangle shape oberserved trajectory. The data also contains gaps which make the shape even less recognizable.
Detecting the shape is complicated, it requires the correct window size and starttime of the iteration. The low frequency and missing quality of the data only increase the complication.

(a) Egocentric velocities for a *dancing* bee



(b) Egocentric velocities for a *following* bee



(c) Egocentric velocities for a random bee

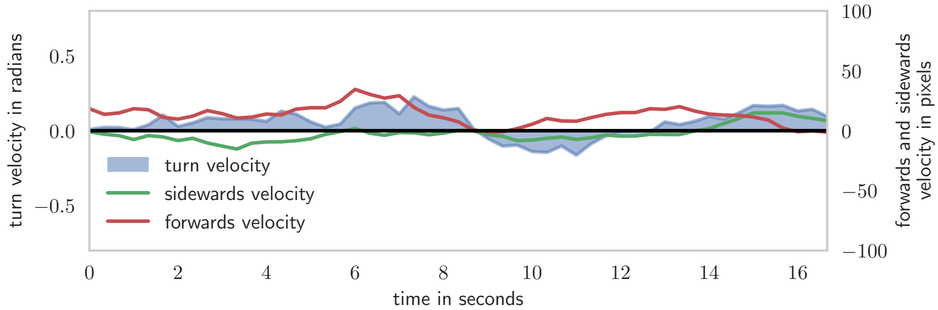


Figure 5: Egocentric velocities in comparison to each other for an exemplary (a) dancer, (b) follower and a (c) random bee. Each velocity has been smoothed over with a moving average of 3 seconds to remove noise. The turn velocity is drawn as an area plot to reduce clutter and effortlessly recognize the pattern but has no further meaning behind it.

Alternating Left and Right Turn

The turn velocity in each iteration for both the dancer and the follower contain two continuous 360 degree rotations. One 360° right rotation for the right arc movement and one for the left arc movement. The pattern is apparent in our model of the dance, see Figure 2. The velocity plots in Figure 5a and 5b shows this pattern of alternating turn velocity in both the dancer and the follower.

Alternating Left and Right Sideward Movement

The follower decodes the information by sticking to the dancer. She achieves this by moving sidewards while the dancer executes the arc movement. This pattern is similar to the alternating turn velocity described above with the exception that it uses the sidewards velocity and occurs more strongly in the follower. The pattern is also visible in Figure 5b.

3.5 Detecting Single Iterations

Combining the patterns described in the previos section yield better results compared to using each pattern in isolation. The lemniscate pattern is excluded in this thesis due to the reason mentioned above to avoid complexity, overfit and a worse detection. The two alternating patterns are of the same nature and can be reduced to a single implementation. This implementation of the *alternating velocity* can be combined with the *high velocity* pattern and further simplified by restating the goal. We constrain the requirements to find single points in time where a switch in the velocity direction occurs, achieved by observing the last and next n points. Such a switch highly correlates with the middle of a *dancing* or *following* iteration. The correlation also occurs at the beginning of an iteration if there was an iteration before or the end if there is one after. The constraint reduces the detection to single iterations while simplifying the implementation.
A new function *velocity_direction_switch* (vds in future) is introduced which correlates with the magnitude of such a switch in direction. The function can be applied for both the velocity_turn and the velocity_side. The functionality is explained in a top to bottom approach by first presenting the result and then explaining each part.

$$\begin{aligned}
\mathrm{M1}(X) &= \mathrm{mean}(X_{t-n} \ldots X_t) \\
\mathrm{M2}(X) &= \mathrm{mean}(X_t \ldots X_{t+n}) \\
\mathrm{vds}_t(X) &= \mathrm{M1}(X) * -\mathrm{M2}(X) * \frac{\min(|\mathrm{M1}(X)|, |M2(X)|)}{\max(|\mathrm{M1}(X)|, |\mathrm{M2}(X)|)} \\
\mathrm{vds_turn}_t &= \mathrm{vds}_t(\mathrm{velocity_turn}) \\
\mathrm{vds_side}_t &= \mathrm{vds}_t(\mathrm{velocity_side})
\end{aligned}$$

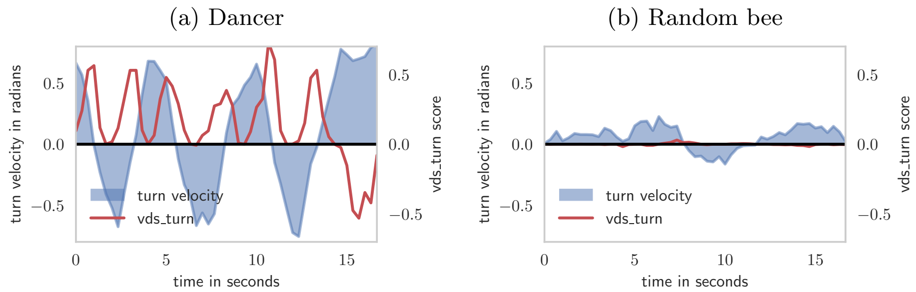


Figure 6: The correlation between the vds_turn and the turn_velocity for a (a) dancer and a (b) random bee. The vds_turn shows a high value when the turn_velocity switches its direction.

A high vds at time t indicates a switch of direction in velocity, the correlation between the velocity and its vds is shown in Figure 6. The idea is to find the middle of a dance/follow iteration. We do for each point in time:

1. **Quantify left and right side**
   Take n left and n right points, average each and call them M1 and M2. The averaging reduces noise and errors and allows for comparison between the left and right side.

2. **Multiply M1 with -M2**
   We expect different signs on M1 and M2 when in the middle of the iteration. The result is positive if a switch exists. Figure 7a shows the correlation between M1 and M2. Our previous *high velocity* pattern is included in this step; the result is greater the larger the switch in direction is.

3. **Add a penalizing factor**
   The turn and side velocity will be relatively constant, a sudden change of speed in *dancing* or *following* is rare, e.g. the bee will not do a fast right arc followed by a slow left arc. We add a factor to penalize a mismatch in proportion. Figure 7b visualizes the correlation between M1 and M2 with the penalization.

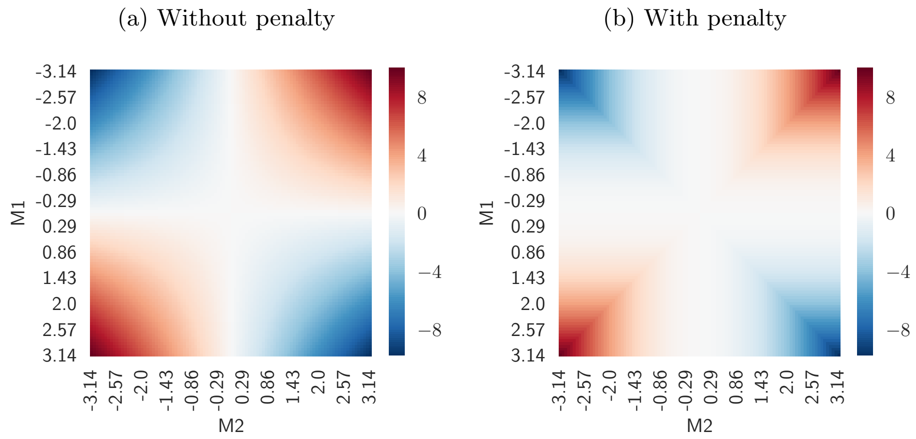


Figure 7: Visualization of the vds_turn as a heatmap (red for high values, blue for low values) for each combination of M1 and M2. A visualization for vds_side would show the same pattern. (a) shows the correlation without a penalty and (b) with the penalty.

Evaluation of the Function

We applied the vds function with both the turn and the sidewards velocity on a one hour dataset not related to our ground truth data. We chose $n = 7$ as the constant by means of intuition which equals to a window size of about 4.66 seconds; the constant was kept after promising results. The 200 largest values in both vds_turn and vds_side were evaluated by labeling the behavior at time of the large values. Possible labels were *dancer*, *follower* and *other*. The result of the evaluation shown in Table 2 shows moderate success in the detection.

	vds_turn	vds_side
dancer	121	28
follower	25	120
other	54	52
	200	200
unique dancers	17	4
unique followers	8	31

Table 2: Evaluation of top 200 values in vds_turn and vds_side. Each column shows the distribution of the behaviors in their 200 largest vds scores (vds_turn/vds_side). The table contains also the number of unique bees in the relevant behavioral classes.

The error rate for each is about 25%. The dancers and followers appear in average multiple times per high vds, this repetitive pattern is used to improve the detection in the next section. The evaluation method in this section is just a quick way to grasp the capability of the vds score. It cannot assess the amount of missed dancers and followers.

3.6 Detecting Multiple Iterations

The evaluation shows that each track for a dancer or follower, in general, contains several points with a high vds score. The first approach to use this observation consisted of a simple moving average; the correlation improved slightly however still contained too much noise. The nature of the vds function includes a fast decline when not in the middle of a velocity switch. A window over multiple iterations would contain several of these low vds values and would pull the average down. The fast decline between each velocity switch is apparent in the previous figure 6a.
The second approach counted each occurrence in the track with a score above a threshold. This solution favors long tracks where noise may randomly exceed the threshold; the bias poses a problem due to the variance in the track lengths, with some tracks being only 0.3 seconds long and others over an hour. The approach also leads to loss of information about the time where the threshold is exceeded.
The final approach adjusts the previous bias by counting the occurrences over a threshold in a sliding window instead of the whole track. Each *dance* or *following* has typically multiple iterations where a switch in velocity happens; the vds score will exceed the threshold frequently in a short time frame.

$$\text{sliding\_threshold\_count}_t = \sum_{i=t-n}^{t+n} \begin{cases} 1, & \text{if } \text{vds}_i > \text{threshold} \\ 0, & \text{otherwise} \end{cases}$$

The sliding_threshold_count (in future stc or stc_turn/stc_side) function has two constants which determine the level of correlation. The first is the window size $2 * n$ and should ideally encapture two iterations. A sliding window size with less than two iterations defeats the purpose of detecting multiple iterations while a large window size favors long tracks. Each waggle-dance iteration varies in time depending on the information the dancer wants to convey [1]. We set the window size for the stc to 10 seconds to capture even very short tracks. The second constant, the threshold, is more sensitive to variation. We obtained good results after the initial setting to the 99.9 percentile of their respective vds score (e.g. the threshold for stc_turn is the 99.9 percentile of all vds_turn values). We kept this initial setting due to good intermediate results.
Figure 8 shows the sliding_threshold_count for the turn velocity (stc_turn) in correlation with the velocity_direction_switch (vds_turn). The stc_turn is highest when the window of 10 seconds contains multiple iterations of a dance.

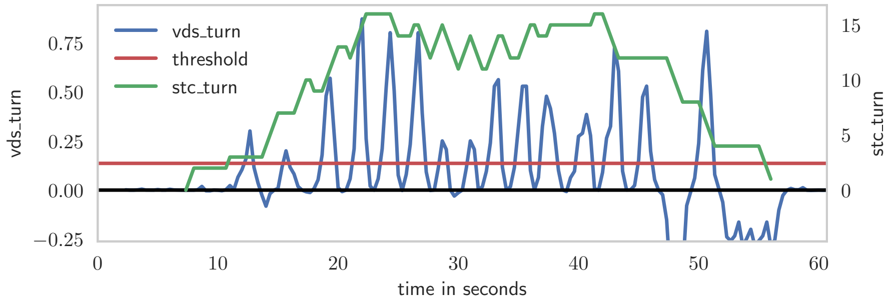


Figure 8: The correlation between the vds_turn function and the stc_turn function with a window size of 10 seconds. The subject is a dancer. The accumulating effect is apparent; multiple occurrences of a vds_turn over the threshold within a short time results in a higher stc_turn.

3.7 Classification

The sliding_threshold_count is a correlation function, a bigger value indicates higher probability of a dance or follow behavior occurrence. The stc function has no classification capability on its own. This section builds a classification algorithm by selecting the optimal time t for the stc functions and uses these values to build a decision boundary.
This optimal time t is where the information about the behavior is largest and ideally where the stc_turn and stc_side is highest. The maximum stc value of a track with a dance or follow behavior is very likely greater than one without these behaviors, while in comparison the lowest value will be in both cases about the same. Observations and previous evaluations have shown that a dancer has in general a higher stc_turn with a bit stc_side and the follower the other way around. Selecting a time t with the maximum value in either stc_turn or stc_side introduces a bias. The addition of a new variable stc_sum, a sum of stc_turn and stc_side, resolves this.

$$\text{stc\_sum}_t = \text{stc\_turn}_t + \text{stc\_side}_t$$

The selected time t is where stc_sum is greatest. We define new variable names for these values.

$$\begin{aligned} i &= \text{argmax}(\text{stc\_sum}_t)\ \forall t \in T \\ \text{stc\_turn\_max} &= \text{stc\_turn}_i \\ \text{stc\_side\_max} &= \text{stc\_side}_i \\ \text{stc\_max} &= \text{stc\_sum}_i \end{aligned}$$

The decision boundary can be calculated with any standard classification algorithm using the features stc_turn_max, stc_side_max and the label (dancer, follower, other) of each track. This thesis decided on the logistic regression to avoid overfitting on the dancer and follower samples with out of the box settings.

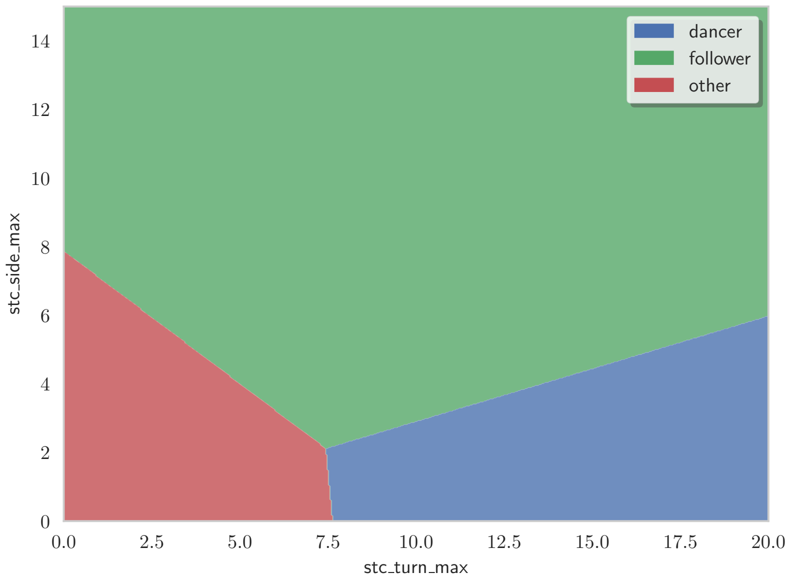


Figure 9: The decision boundaries of the logistic regression.

4 Evaluation

The evaluation described in this section is under some limitations. The first limitation is the evaluation on a per track basis; each track has a single class. The classification in time series containing multiple classes can be complex, detecting the right start and end time for the class is a challenge. This challenge also applies to the manual labeling to acquire ground truth data and is the reason for this decision. Acquiring the ground truth data is a labor-intensive process as described in section 3.1. The simplification to a per track basis also simplifies the labeling and classification process. The drawback occurs when a long track contains multiple dances, followings or a mix of both but does not apply to our situation. No sample with this conflict was found during the labeling of the ground truth data.

behavior	no. of samples	percentage
dancer	6	0.4
follower	22	1.6
other	1308	97.9

Table 3: The number of samples per behavioral class. The table shows the magnitude of the class imbalance.

The second limitation is that the evaluation used the same ground truth data as the logistic regression. The cross-validation of the samples is not possible, only 2% of the samples are dancer or follower. The magnitude of this class imbalance is shown in Table 3. The drawback is the possibility of an overfit in the logistic regression but can not be further evaluated without more ground truth data.

		predicted			
		dancer	follower	other	accuracy
	dancer	5	1	0	0.833
truth	follower	1	18	3	0.818
	other	0	1	1307	0.999
	accuracy	0.833	0.9	0.998	0.996

Table 4: Confusion matrix of the model.

The model performed well under the given limitations with a total accuracy of 0.996. The class imbalance skews this number, the confusion matrix in Table 4 details each type of error in our model. The F1-score is another more suitable metric to explain the amount of error. The model shows success with an F1-score of 0.83 and 0.86 for a dancer and follower. A more detailed report is shown in Table 5.

	precision	recall	f1-score	support
dancer	0.833	0.833	0.833	6
follower	0.900	0.818	0.857	22
other	0.998	0.999	0.998	1308
avg / total	0.995	0.996	0.995	1336

Table 5: Precision, recall, F1-score and the support for each class.

A detailed investigation of the misclassifications shows different types of error and is divided into following groups:

- **High gap**
  A high gap in the data may either lead to unrecognized dancers or followers or the misclassification between them. Missing data points either distort or obstruct the pattern.

- **Single iteration**
  The variance in the number of iterations for a follower is high. A follower sometimes follows for just a half iteration or single iteration. The requirements for a dancer or follower label is the execution of at least a single full iteration. The sliding_threshold_count on the other hand is designed to detect multiple iterations and may fail to correlate with a single iteration.

- **Overcrowded hive**
  An overcrowded hive may lead to deviations in the trajectory and variation in the patterns.

- **Attendees**
  Honey bee attendees, first described by Božič and Valentinčič [14], may exhibit patterns similar to the follower. Motivated attendees may sway left and right with the dancer while not participating in the dance. The swaying matches the pattern of the *alternating side velocity* and can in few cases lead to misclassification.

5 Future Work

This thesis presented a model to automatically detect waggle dancers and their followers on low temporal-resolution spatiotemporal data. The detection of dance-followers is novel; previous work focused only on detecting dancers using the waggle part of the dance as the only feature.
The presented model uses domain knowledge to identify and implement multiple waggle-dance patterns for the detection. The implementation is divided into four steps. The first step described the typical behavioral model of the dancer and follower and subsequently identifying the patterns in it.
The second step selected a subset of these patterns and implemented a rank correlation function associated with the middle or beginning of a single dance iteration. The selection of the patterns focuses on the capability to detect the behaviors in a low temporal-resolution space. The data used in this thesis is recorded with 3 Hz and does not permit the detection of fast movements like the previously mentioned waggle part. The third step improved the robustness of the correlation function with a sliding window to extend it to multiple dance iterations. The last step of the implementation finds the optimal point in time in the correlation function and builds a classifier from it.
The evaluation showed the success of the model with an F1-score of about 0.85 for dancers and followers even under conditions of strong class imbalance. The number of dancer and follower samples comprise respectively 0.4% and 1.6% of the population. Typical machine learning models handle a class imbalance poorly when working with small datasets. The use of domain knowledge mitigated such a problem by constructing discriminitive features with a significant decision boundary.

While the thesis provided a different and novel approach to detect waggle dancers and followers, parts of it lacked methodology. Future work could improve the theoretical foundation of the model. Following three instances detail the gaps in the methodology and present a solution to each.
The first instance of lacking methodology is the window size for the velocity_direction_switch. The window size was chosen by intuition but could have been determined with research about the range of duration a single dance iteration can take. The work by Landgraf et al. in 2011 [13] analyses waggle dances metrics and can be used in the future to improve the theoretical foundation.
The second instance is the use of the 99.9 percentile as the threshold. The correct approach would have been an analysis of the sliding_threshold_count distribution for dancers and followers. The analysis would require more ground truth data than currently available.
The evaluation is the third instance of lacking methodology by reusing data. The same data to learn the logistic regression is used in the evaluation. No

cross-validation techniques were applied, this can lead to overfitting and reduces the confidence in the evaluation results. The scarcity of ground truth data is the cause for the lack of cross-validation. Splitting the few dancer and follower samples would not yield any significant result.

A future improvement unrelated to the scientific methods is the use of weighted classification algorithms. The current state does not address the class imbalance and slight overestimation of the *other* behavior. Discarding samples of the majority class or the implementation of different learning constants per class can negate some of the impacts of the class imbalance. An increase in ground truth data would, similar to the other problems, benefit and increase leeway.

This section shows that the prerequisite for most of the future work lies in the generation of more ground truth data. More data would improve the significance of the validation and enable to test more variations without fear of overfitting. While generating more ground truth data is no challenge itself, the necessary time effort poses a limitation within the frame of this thesis.

6 References

References

[1] K. von Frisch, *The dance language and orientation of bees*. Cambridge, MA, US: Harvard University Press, 1967.

[2] F. Wario, T. Landgraf, *et al.*, "Automatic methods for long-term tracking and the detection and decoding of communication dances in honeybees", 2015. DOI: 10.3389/fevo.2015.00103. [Online]. Available: http://www.diss.fu-berlin.de/docs/receive/FUDOCS_document_000000023377 (visited on 03/11/2017).

[3] J. Mischek, "Probabilistisches tracking von bienenpfaden", unpublished Master Thesis, Freie Universität Berlin, Sep. 9, 2016.

[4] L. Sixt, "Rendergan: Generating realistic labeled data – with an application on decoding bee tags", unpublished Bachelor Thesis, Freie Universität Berlin, Aug. 21, 2016.

[5] B. M. Rosemann, "Ein erweiterbares system für experimente mit multitarget tracking von markierten bienen", unpublished Master Thesis, Freie Universität Berlin, 2017, 55 pp.

[6] F. Boenisch, "Feature engineering and probabilistic tracking on honey bee trajectories", unpublished Bachelor Thesis, Freie Universität Berlin, Feb. 21, 2017, 52 pp.

[7] T. Landgraf, "Robobee: A biomimetic honeybee robot for the analysis of the dance communication system", PhD thesis, Freie Universität Berlin, Freie Universität Berlin, Germany, 2013. [Online]. Available: http://www.diss.fu-berlin.de/diss/receive/FUDISS_thesis_000000094818 (visited on 03/30/2017).

[8] J. C. Nieh, "The stop signal of honey bees: Reconsidering its message", *Behavioral Ecology and Sociobiology*, vol. 33, no. 1, pp. 51–56, Jul. 1, 1993, ISSN: 0340-5443, 1432-0762. DOI: 10.1007/BF00164346. [Online]. Available: https://link.springer.com/article/10.1007/BF00164346 (visited on 04/22/2017).

[9] A. Feldman and T. Balch, "Automatic identification of bee movement", 2003.

[10] T. Kimura, M. Ohashi, *et al.*, "A new approach for the simultaneous tracking of multiple honeybees for analysis of hive behavior", *Apidologie*, vol. 42, no. 5, p. 607, Sep. 1, 2011, ISSN: 0044-8435, 1297-9678. DOI: 10.1007/s13592-011-0060-6. [Online]. Available: https://link.springer.com/article/10.1007/s13592-011-0060-6 (visited on 03/17/2017).

[11] T. Landgraf, R. Rojas, *et al.*, "Tracking honey bee dances from sparse optical flow fields", 2007. [Online]. Available: http://www.diss.fu-berlin.de/docs/servlets/MCRFileNodeServlet/FUDOCS_derivate_000000000829/2007_11.pdf (visited on 03/16/2017).

[12] K. v. Frisch, "Aus dem leben der bienen", 1927.

[13] T. Landgraf, R. Rojas, *et al.*, "Analysis of the waggle dance motion of honeybees for the design of a biomimetic honeybee robot", *PLOS ONE*, vol. 6, no. 8, e21354, Aug. 3, 2011, ISSN: 1932-6203. DOI: 10.1371/journal.pone.0021354. [Online]. Available: http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0021354 (visited on 03/13/2017).

[14] J. Božič and T. Valentinčič, "Attendants and followers of honey bee waggle dances", *Journal of Apicultural Research*, vol. 30, no. 3, pp. 125–131, Jan. 1, 1991, ISSN: 0021-8839. DOI: 10.1080/00218839.1991.11101246. [Online]. Available: http://dx.doi.org/10.1080/00218839.1991.11101246 (visited on 03/16/2017).

[15] C. Grüter and W. M. Farina, "The honeybee waggle dance: Can we follow the steps?", *Trends in Ecology & Evolution*, vol. 24, no. 5, pp. 242–247, May 2009, ISSN: 0169-5347. DOI: 10.1016/j.tree.2008.12.007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169534709000779 (visited on 04/28/2017).

[16] M. Kabra, A. A. Robie, *et al.*, "Jaaba: Interactive machine learning for automatic annotation of animal behavior", *Nature Methods*, vol. 10, no. 1, pp. 64–67, Jan. 2013, ISSN: 1548-7091. DOI: 10.1038/nmeth.2281. [Online]. Available: http://www.nature.com/nmeth/journal/v10/n1/abs/nmeth.2281.html (visited on 03/15/2017).