Freie Universität Berlin
Fachbereich für Mathematik und Informatik
Institut für Informatik

Bachelorarbeit

Improvement of the Waggle Dance Detector

30. Juli 2015

Bearbeitet von:
Laura Mielke
Matrikelnr.: 4478532
Laura.Mielke@fu-berlin.de

Erstgutachter : Dr. Tim Landgraf
Zweitgutachter : Prof. Dr. Raúl Rojas

Abstract

Bees perform a waggle dance to advertise food locations to fellow bees. This dance encodes a lot of different information, including distance and direction from hive to food source. Decoding this dance has been the subject of study for longer than the past five decades. In order to study dance properties detection and decoding of a dance had to be done by hand. The Waggle Dance Detector offered the first automated real-time detection and decoding of dances. This thesis delivers an improved approach to dance orientation decoding for the Waggle Dance Detector.

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe. Diese Arbeit wurde keiner anderen Prüfungsbehörde in gleicher oder ähnlicher Form vorgelegt.

Berlin, den 30. Juli 2015

....

Laura Mielke

Acknowledgments

I would like to thank Tim Landgraf for the opportunity to write a thesis in this interesting field and for his support. Furthermore I want to thank Timo Hanisch, Sindy Hartmann, Felix Spielmann and Carsten Flöth for their support.

Contents

List of Figures

List of Tables

1 Introduction

Humans have always been interested in the behaviour of honey bees. In scientific research, both their unique bearing and their environmental importance have made them the center of detailed studies Weidenmüller and Seeley [13]. One point of interest in those studies is the waggle dance of honey bees. Waggle dancing allows bees to communicate the location of a promising food source to other bees through motion. This makes it a unique form of abstract information transfer Seeley [7]. First interpreted as information by Karl von Frisch [12], the knowledge about the dance and its encoded information has since then been deepened and detailed, often achieved through many hours of observations and manual analysis Landgraf et al. [2]. Although important, this manual work takes up the researchers time. To tackle this problem the Waggle Dance Detector was developed. It is a system for automated real-time detection and decoding of waggle dances Rau [6].

The goal of this thesis is to deliver improved solutions to the major problem of the Waggle Dance Detector, the waggle dance orientation decoding. First this thesis gives a brief introduction about the honey bee waggle dance. The main algorithms used in this thesis are presented followed by an overview of the Waggle Dance Detector as presented by Alexander Rau in 2014. Its goal and implementation are explained as well as problems. Following this several possible solutions to the waggle orientation decoding problem are suggested. An overview to the solution implementations is given. Afterwards the solutions are evaluated, each for themselves and against each other. In the end an outlook for possible future work and improvement is given.

2 State of the art

This chapter gives an overview on the waggle dance of honey bees. It is explained what information the waggle dance contains and how it is encoded. Then both the Principal Component Analysis and the Singular Value Decomposition are introduced with regard to their application in this thesis. Finally this chapter discusses the Waggle Dance Detector. A description is given and the implementation is illustrated. Finally problems and solutions known from the master thesis of Alexander Rau are discussed.

2.1 Waggle dance

The communication of food sources between bees has been observed and discussed for centuries. Already Aristotle mentioned that bees visiting promising food locations seem to attract more bees Aristotle et al. [1]. In the 18th century Spitzer described the habit of bees to dance Spitzner [9]. The first to decipher the information in the dance was Karl von Frisch. He suggested that a foraging bee advertises the food source to other bees through the dance. Following research of Frisch has shown that the dance is not only an advertisement for food. It is an encoding of the food sources' location von Frisch [12].

On returning from a promising food source a foraging bee can inform other bees about the location of this source through a certain dancing motion. She repeatedly walks in an approximate eight-figure on the hive surface, starting with the middle part as a forward motion, as can be seen in figure 2.1. During this forward motion she swings her abdomen from side to side with a frequency of about 13 Hz Landgraf et al. [2]. This is called the waggle run. She then returns in a semicircle to her starting position, alternating between a left and right turn for each repetition. Other bees will follow her dance and learn the information she communicates with her dance. Then they will fly out and visit the food source themselves.

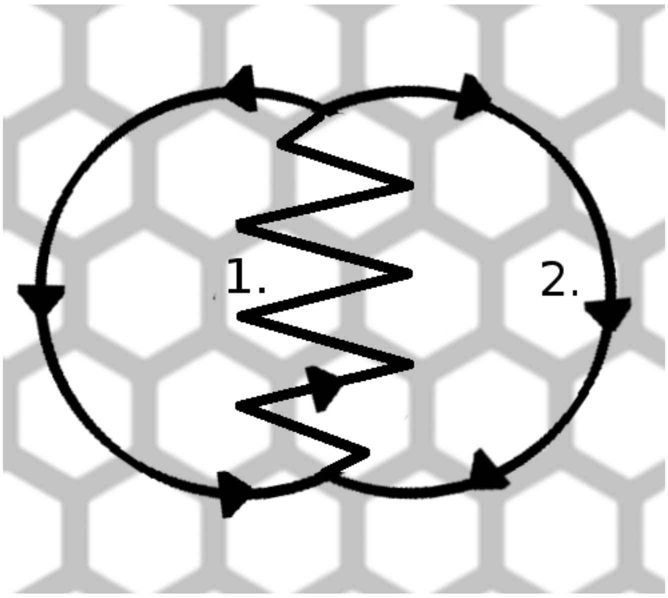


Figure 2.1: The approximate path of a waggle dance. The bee starts with her waggle run(1.). When finished, she turns back with a return run (2.). Here she alternates between left or right return runs after each waggle run.

As description of the food's location the bee communicates the distance and direction from the hive. For both, distance and direction, the waggle run is the most important part of the dance. Distance can be deduced from the length of a waggle run. A longer run means a longer distance, like a scaled version of the described way Seeley [7]. Typically a bee will only dance a waggle dance for a distance above 150m Tarpy and Service [11]. The direction can, too, be deduced from the run. When outside, bees relate directions to their angle to the sun's azimuth. That means flying in the direction of the sun would mean an angle of 0°. The bees differentiate between a left or right turn from the sun's azimuth, giving us degrees from -180° to 180°. Turning 45° to the left from the sun would mean a direction angle of -45°, turning right would mean 45°. In the dark, vertical hive bees use gravitation as their orientation. The aforementioned dancing direction of 0° would mean a movement directly opposed to gravity. A dance in direction of gravity would therefore mean a food direction opposed to the sun. The liveliness and number of repetition of a waggle run refers to the value of the source Seeley [7]. Other bees will follow the dance and learn its properties.

2.2 Underlying algorithms

2.2.1 Principal Component Analysis

First described by Karl Pearson in 1901, the Principal Component Analysis (PCA) has since then become an important tool in multivariate statistics. It is often used to identify underlying patterns in large multivariate datasets or to reduce their dimension in order to simplify studying them. Pearson himself described the PCA as a "complete analytical solution of the problem of drawing the best-fitting plane through n non-coplanar points" Pearson [5]. The goal of the PCA is to transform a set of possibly correlated variables into one of uncorrelated variables. Those new variables are called the Principal Components (PC). The PC is the direction of the dataset with the highest variance. Then the others follow subsequently with declining variances. Transformation onto the new set both maximizes variance and minimizes redundancy in the data.

Principal components for a dataset can be gained through the covariance matrix X_{cov}. To be precise, the eigenvectors of X_{cov} are the Principal Components of X. The eigenvector with the largest eigenvalue is the first Principal Component Pearson [5].

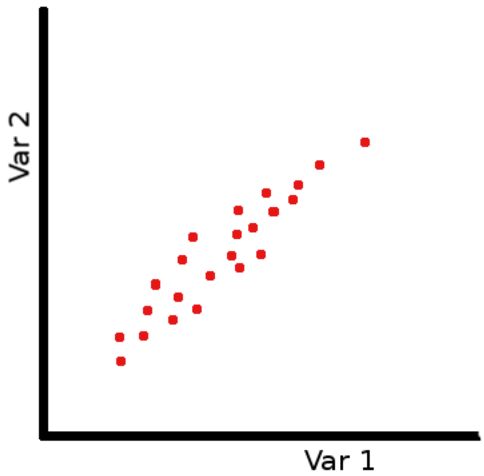


Figure 2.2: A two-dimensional dataset measured by variables $Var1$ and $Var2$. The data has a distinct direction, that none of the variables fits.

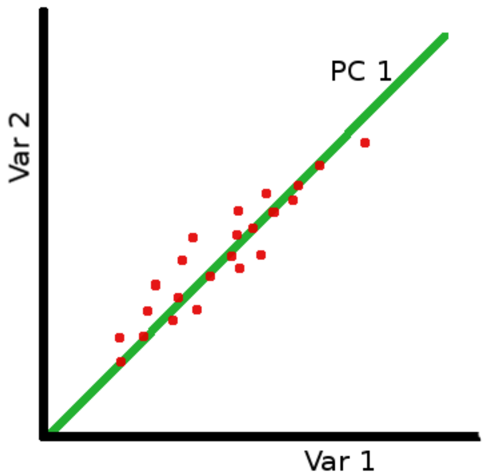


Figure 2.3: The first Principal Component $PC1$ approximates the data with a minimal error. Since it is a two-dimensional space the PC is a line.

Algorithm 1 Principal Component Analysis with eigen decomposition

function PCA(Matrix X)

  $Input : M \times N matrix$, M variables, N Samples

  $Output : X_{transformed}, PC, V$

  $[M, N] \leftarrow size(X)$

  $mean \leftarrow meanRows(X)$

  ▷ find mean for each variable

  $X \leftarrow X - mean$

  ▷ subtract mean for each variable

  $X_{cov} \leftarrow \frac{1}{N-1} * X^T X$

  ▷ Find covariance matrix

  $[PC, V] \leftarrow eig(X_{cov})$

  ▷ compute eigenvectors and diagonal matrix with eigenvalues

  $V \leftarrow diag(V)$

  ▷ extract eigenvalues from diagonal matrix

  $[V_{sorted}, indices] \leftarrow sort(V)$

  $V \leftarrow V_{sorted}$

  ▷ Sort eigen values in decreasing order and returns sorted values and array with former indices

  $PC \leftarrow PC(:, indices)$

  ▷ Sort eigenvectors by order of eigenvalues

  $X_{transformed} \leftarrow PC^T * X$

  ▷ If wished, transform input data onto principal components

2.2.2 Singular Value Decomposition

Another way to find a new orthogonal basis for a given data set is the Singular Value Decomposition of the data matrix. Here not the eigenvalues but the singular values determine the variance. A singular value is the square root of its eigenvalue counterpart. They can be achieved through a factorization of a $M \times N$ matrix X with rank r into 3 different matrices:

$$X = U \sum V^T \tag{2.1}$$

$$U = M \times M \text{ unitary matrix} \tag{2.2}$$

$$\sum = M \times N \text{ diagonal matrix} \tag{2.3}$$

$$V^T = N \times N \text{ unitary matrix} \tag{2.4}$$

$$\tag{2.5}$$

U are the left-singular vectors of X, on the diagonal of $\sum$ are the singular values and V are the right-singular vectors of X Stewart [10].

Algorithm 2 Principal Component Analysis with Singular Value Decomposition

function PCA_SVD(Matrix X)
  $Input : M \times N$matrix, M variables, N Samples
  $Output : X_{transformed}, V^T, \sum$
  $[M, N] \leftarrow size(X)$
  $mean \leftarrow meanRows(X)$
    ▷ find mean for each variable
  $X \leftarrow X - mean$
    ▷ subtract mean for each variable
  $[U, \sum, V^T] \leftarrow svd(X)$
    ▷ compute matrix with left singular vectors, diagonal matrix with sorted singular values and matrix with right singular values
  $\sum \leftarrow diag(\sum)$
    ▷ extract eigenvalues from diagonal matrix
  $X_{transformed} \leftarrow V^T * X$
    ▷ If wished, transform input data onto principal components

2.3 Waggle Dance Detector

In the field of honey bee research the techniques to study the honey bee waggle dance have always involved vast need for memory and countless hours of manual review Landgraf et al. [2]. Both are limited resources. To tackle those limitations Alexander Rau developed the Waggle Dance Detector. The Waggle Dance Detector is the first attempt to automatically detect and decode honey bee waggle dances in real time Rau [6].

2.3.1 Goal

The Waggle Dance Detector has the goal to detect and decode waggle dances in video recordings as a replacement for human work. There are major features which make it the preferable option. The first feature is to be automated, in order to reduce human supervision. It runs online and saves only relevant data, e.g. dance duration. This, as a second feature, reduces memory that is normally necessary to save the high resolution videos used in detection per hand. To deliver a proper replacement for detection and decoding through humans it has the implicit goal to achieve a low false negative rate and reliable decoding.

2.3.2 Implementation

As input, the Waggle Dance Detector (WDD) takes a frame from a camera. The frame gets translated into grayscale and is possibly subsampled. To process the frame, the WDD has three consecutive detection layers:

1. signal detection (layer 1)	Dot Detectors detect possible waggling
2. signal detection (layer 2)	cluster signals from layer 1
3. signal concatenation (layer 3)	connect layer 2 signals over time

Table 2.1: Layers for signal processing and their functions

The first layer matches a Dot Detector to each pixel. The Dot Detector is an observing system that checks for a frequency of 11-17 Hz in the color change of its pixel. The target frequency refers to the 13 Hz with which bees waggle. In order to achieve this, each Dot Detector has a buffer for the last N pixel values at its position. N is chosen, so that it can hold a whole waggle run, depending on the frame rate. Based on those values the Dot Detector makes a detection decision. According to the decision, the Dot Detector either emits a positive or negative detection signal to the second layer.

The second layer clusters the signals from layer 1 that belong to one honey bee. Corresponding to the scaled size of a honey bee a maximum distance is defined. The distance between two Dot Detectors and, consequently, their membership in a certain cluster is measured through the euclidean distance and should not exceed the honey bee size. For each found cluster a signal is then emitted to layer 3.

Layer 3 connects the clusters over time to create a timeline for each detected waggle run. The layer decides if a layer 2 detection belongs to an already existing waggle run or if it is a newly detected one. As in layer 2 here, too, the distance has to be taken into account. Waggle runs can experience gaps in detection. A maximum gap size, which was not specified in Raus thesis, is given. If a waggle run exceeds this gap size, it is closed and declared a finished run. With this, the detection is finished and the found waggle runs are passed to the extraction, or decoding layer.

The decoding layer extracts all relevant information from the found waggle dances and writes them to a file structure.

Position information	The coordinates of the bee in each frame. Position here means the center of the cluster. Coordinates for detection gaps are filled with $< -1, -1 >$.
Time Stamp	The time stamp marks the starting of the detection for this dance.
Duration	The numb er of frames that were clustered to this dance .
Direction	The dancing angle is determined through the first and last coordinate of the dance.
Images	From each frame of the dance an 20x20px image is cropped that shows the dancing bee. The center of the image is the starting position of the bee.

Table 2.2: Output values from layer 3 and their explanations

Finally the results are written into a CSV-file which is saved, along with the images, to a file structure that represents day, time, dance and run of the dance. An overview of the file structure can be found in Section **3.3**.

2.3.3 Problems

Although the Waggle Dance detector does fulfill most of its purposes, it still is only a prototype. Therefore several problems occurred during its evaluation, as mentioned by Alexander Rau in his thesis. The first was a high number of false positives. Since the results can be reevaluated by humans the false positive rate is not directly contradictory to the goal of the WDD. But they do diminish its value in respect of the reduction of human work. A part of these false positives came from a typing error in a formula of the WDD and has been corrected. As yet, the correction has been tested but not evaluated.

Another relevant problem is a time delay in duration. Since the Dot Detectors have a threshold function that works on a buffer, at least a few frames with waggles have to show up in the buffer. And as Rau himself states a bee starts her run before the first waggle, which makes a delay inevitable. The end of the run experiences the same problem the start does since enough waggle-free frames have to fill the buffer to stop recognition. The delay does influence the quality of the decoding of duration. At least the delays partly compensate each other. Measured by Rau, the duration has respectively $79ms$ and $139ms$ standard deviation for start and end. But it does influence other parts of decoding, such as waggle run orientation.

The most problematic part is the waggle run orientation decoding. According to Rau "first and last honey bee positions are used to decode waggle run orientation" Rau [6].

Rau does not specify the used algorithm, but it is clearly a rather naive approach. As his evaluation shows, it can deliver correct results, but there are a lot of possible traps. First, as he remarks, if a bee were to stop a run where it started, no orientation could be obtained. This is a rather unlikely case, but it does lead us to much more realistic problems. If the bee should turn in the last few frames from her path, the orientation Rau measures would use a wrong value. The same goes for the starting position. This is especially relevant in light of the time delay. After her run, the bee will turn right or left. Through the delay, the last frames of the run can easily be the first of the return part, as can be seen in figure 2.4. In general every deviation of one of both points results in deviations of the whole orientation. The shorter the run the higher the influence of a deviation. Although Rau remarks that the position decoding is well enough, with a standard deviation of $1.2mm$ along the y-Axis and $1.7mm$ along the x-Axis, a wrongly measured position would again have a high influence on the end result.

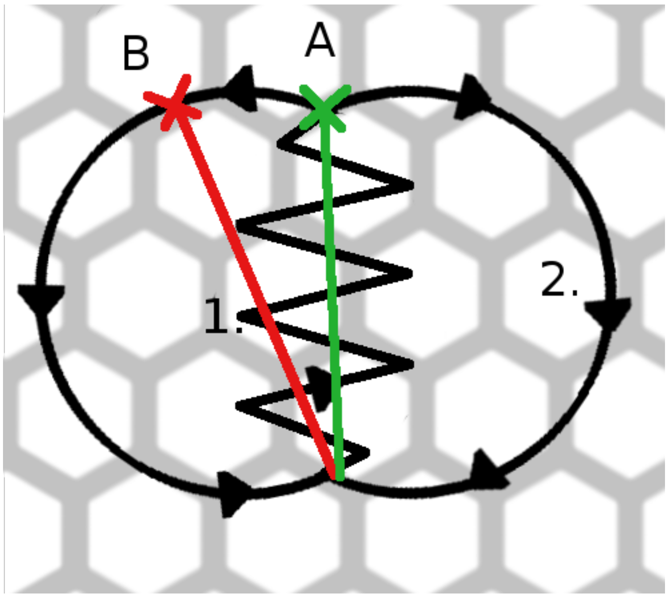


Figure 2.4: A shows the path from start to end position of the bee in a waggle dance. B shows the path between the start position of a bee and her position when measured with delay. Both paths diverge from each other.

2.3.4 Discussed solutions for orientation decoding

From all highlighted problems the most acute is the orientation decoding. It is one of the main values the WDD generates, since it is necessary to decode the location the dance advertises. But unlike the time delay, its error is not predictable. According to Rau, it has a mean error of about -1°, which is an acceptable value. But it has a standard deviation of 36° that means an unpredictable result. That still leaves the user to wonder if the computed angle is close to the original or if one of the afore-mentioned cases eventuated.

In his thesis Rau proposed and discussed several other possible solutions to this problem, next to the implemented one. The first was to include more positions in the fitting of the line. The second was a an image based approach. He mentioned an earlier implementation that considered the temporal difference between the images of a run, which failed during live tests. The last mentioned method was that of the RoboBee Landgraf et al. [2]. Here the orientation of the bees body in each frame was taken and averaged. However, Rau excluded this method due to missing access to necessary information in the WDD. Since the main problem of the WDD orientation decoding is its stability, this thesis will concentrate on the goal to find a more stable algorithm.

2.4 Summary

In this chapter the honey bees waggle dance has been introduced. Its has been explained what the dance means and how humans can decode its meaning. Important algorithms for this thesis have been discussed and an overview about the Waggle Dance Detector, its use and problems has been given. The main problem was explained and several possible solutions were introduced.

3 Approaches & Implementation Details

"From a physical point of view, the orientation extraction is definitely the most difficult part of waggle run decoding. " Rau [6]

The main purpose of this thesis is to test and compare different approaches for the decoding of waggle run orientation based on WDD output with regards to their stability. In Section **2.3.4** different possible ideas were suggested. Since none of them was declared with a definite algorithm, this section proposes suitable algorithms for them. Furthermore, new ideas will be considered. A final implementation for each mentioned idea will be offered in order to compare their usability and results.

3.1 Real-time vs Post Processing

When deciding which approaches are to be implemented, the first decision to be made was whether the orientation should be decoded while the WDD is running or afterwards. The original Waggle Dance Detector works in real-time. This concerns all steps, including orientation decoding. It is supposed to reduce runtime and memory. If an implementation of the orientation decoding works on the output data of the WDD it would not violate those goals. The output data is already reduced and therefore processing time for this data would also be minimized and no extra memory would be necessary. The Waggle Dance Detector has been in use for about one year now. There are at least two major sets of data on which it has been used, the Beehouse and the BeesBook data sets. For both sets data was recorded 24 hours per day. Beehouse was an 48-day, BeesBook a 31-day observation. Since they were treated with the prototype of the WDD, they suffer from the in Section **2.3.4** discussed issues. In order to improve their results it would be beneficial to have an algorithm that could compute improved orientations based on the results of the existing WDD. Such an implementation has the additional benefit that it can be tested on the same data sets as the Waggle Dance Detector. To receive those benefits, all following algorithms are implemented to use the output of the WDD instead of working in real-time as part of the WDD.

3.2 Ground Truth

In order to measure the functionality and efficiency of the tested algorithms, their results have to be tested against a ground truth, a set of data for which the correct results have been measured with a reliable method. Since the WDD is the first program of its kind, the data has to be measured by a human being, as it has been done before. In order to simplify this process, a measuring program for waggle dance orientations has been provided.

The input for the program consists of the sampled images for one waggle run. When starting, the first image is displayed for the user. Since the important information here is the forward motion of the bee over all images the images can be browsed per key. When keeping the key pressed this gives the impression of a flip book, where the movement of the bee becomes clear. To measure the angle of the bee a line is drawn on top of the current image. It has differently marked end points for the head and tail direction of the bee. This way the correct direction of the movement can be saved. Blue marks the head and red the tail direction of the bee. The user can drag the line across the image by the points to match the movement. The line can be stretched as wished. Different keys are assigned to useful functions, such as flagging if it is a dance, resetting the program, saving the angle and flag or skipping through the images. A listing of keys can be found in the appendix. Orientation is measured against the x-Axis. A framework for this program is also provided, which takes as input the output folder structure of the WDD and chooses random runs for the user to evaluate.

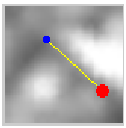


Figure 3.1: A caption of the program to create ground truth data can be seen. The image in the background is an output image of the WDD. On top the yellow line marks the bee orientation. The blue spot is the direction of the bees head. The red spot the direction of her tail.

3.3 Orientation framework

In order to use and compare all algorithms on a neutral basis, a framework has been implemented. As input it takes a WDD output folder structure and the algorithm to be used. Such a structure has the following form:

```
day
└─hh:mm:dancenumber in that minute
   └─run number in that dance
      ├─output CSV-file
      └─output images
```

Figure 3.2: Output file structure of the WDD

It traverses all waggle run folder in the structure. When a waggle run folder is reached, its output CSV-file is read. Such a file is build as follows:

1. line	Start X/Y-coordinates in absolute pixelvalues of the target frame and orientation (in rad)
2. line	Start time of the detection (HH:MM:SS:mmm) and number of frames
3. line	4 X /Y coordinates that represent the adjusted homography
4. line	Row of x and y coordinates, the bee positions in each frame

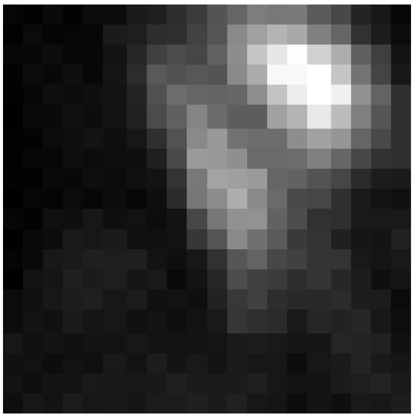
Table 3.1: Form of the output-file from the Waggle Dance Detector

The chosen algorithm is then called with necessary parameters from the output file. It returns an orientation which is written to a CSV-file in the waggle run folder, next to the WDD output-CSV. An additional version of this framework is provided that only reads runs that have a ground truth and compares the computed orientation to their ground truth. Comparison results are not written to a CSV-file but collected. In the end the standard deviation of the difference, the mean error and a plot are given for the whole file structure.

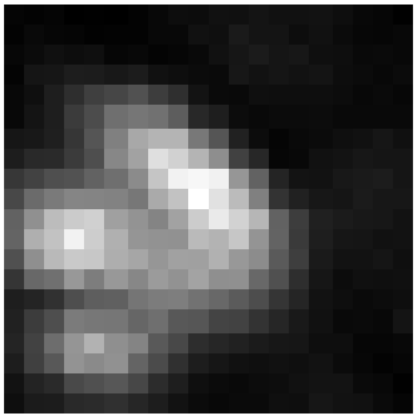
3.4 Image subtraction

Rau described his approach as "temporal differences of cropped image sequences and calculated properties for move fragments". This method is used as a possible preprocessing step for methods from Section **3.5**, Section **3.6** and Section **3.7**. It provides an alternative to the existing bee positions from the Waggle Dance Detector.

As input this method takes a path to the folder that holds the images of the run. The first step here is the temporal difference. All images cropped from frames for a run by the WDD show the same section of the frame. Consecutive frames from a video are a sequence in time. Hence those images only show a change over time, not over space. Image subtraction is often used as a means to detect a moving object in front of a static background Shapiro and Stockman [8]. In order to extract the bees movement one image $I1$ is subtracted from another image $I2$ as follows. Both images $I1$ and $I2$ have to have the same size. Then for each pixel $I1_{ij}$ from $I1$ the difference of its value to the value of its counterpart $I2_{ij}$ is computed. The result is a new image or matrix the size of $I1$ and $I2$ that contains the differences as entries. Movement here is equal to change. If nothing moved, the value of the new pixel is 0. This image subtraction is repeated for all images and their respective successors, if they have one. Then all results are added up to gain an image I_{sum} of the movement over the whole run. The higher the value in I_{sum} the more movement occurred. The movement of the bee is expected to have the highest velocity in the image.



(a)



(b)

Figure 3.3: The images show the difference images for the runs 20140823_1139_0_9 and 20140823_14_03_0_8. The bright fields show movement.

The second step is to extract the path of the bee from I_{sum}. The final implementation uses a threshold segmentation. A threshold $thresh_{gray}$ from the scale of intensity values in the image is chosen. All pixels with values $v < thresh_{gray}$ are assigned a value of 0 and all values with $v >= thresh_{gray}$ a value of 1. This describes the conversion of a grayscale image I_{gray} into a binary image I_{bin}. Ideally, in I_{bin} the bee path would become 1 and the background would become 0, depending on the choice of threshold. The basis to choose the threshold is the histogram of the pixel values in I_{gray}. This is a histogram showing the number of pixels in the image for each possible intensity value.

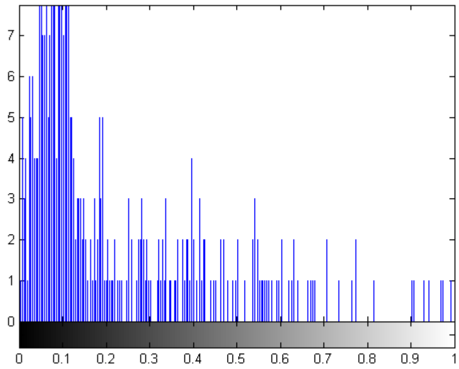
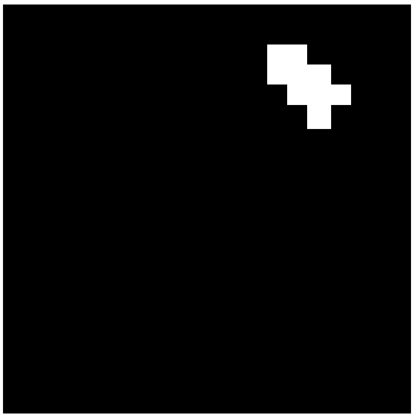
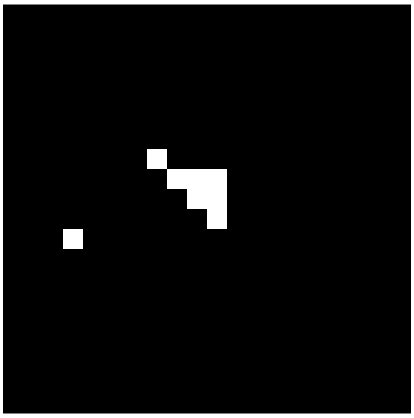


Figure 3.4: The histogram of the difference image from the waggle run 20140823_1139_0_9. The darker values have a much higher maximum, indicating little movement.

A common threshold is the low point between two peaks in the histogram Parker [4]. Here, Otsu's method was used. With Otsu's method a threshold is found that divides the values into two clusters that minimize the intraclass variance and maximize the interclass variance Otsu [3]. When looking at I_{sum} its obvious that the bee path is marked only by the brightest values. That means a single global threshold as defined by Otsu's method is not sufficient Shapiro and Stockman [8]. Finally, Otsus method for multiple thresholds was used. Here, Otsu's method is used for more than two clusters. The final number of clusters used was 10. After thresholding the positions of the pixels p_{ij} that contain a value of 1 is taken as a set of coordinates for bee movements. They then can be processed with one of the following algorithms.



(a) This image shows the differences from the images in waggle run 20140823_1139_0_9 after it was transformed into a binary image with a threshold from Otsus multithreshold method.



(b) The transformation of 20140823_14_03_0_8 to a binary image with Otsus multithreshold method.

3.5 First and last coordinates

This is the most naive approach. From the description in Rau [6] it could not be inferred which exact algorithm was used in the first implementation of the WDD. Rather the idea was given that the first and last coordinate of the bee positions was used. This implementation represents the thesis interpretation of this idea. The first and the last coordinates from the waggle run are taken as orientation points for a line. Then the angle of the line to the x-Axis is calculated with the inverse of the tangent function.

3.6 Mean direction

The algorithm in this approach is an interpretation of the algorithm that was used for the Beehouse data. Although it is unclear which exact algorithm was used there, this implementation creates nearly the same results. As input it takes an array with all bee positions from the detected frames. For each position, the difference of the position to its successor is computed. Then the mean of those differences is taken as a vector:

$$< meanX, meanY > \tag{3.1}$$

The angle of the vector is again computed with the inverse of the tangent function against the x-Axis. Compared to Section **3.5** it involves more coordinates and therefore might tackle the problem of vulnerability towards changes in coordinate values.

3.7 Principal Component Analysis

In this section both the implementation and options for improvement of the PCA are explained.

3.7.1 Basic implementation

The idea of Principal Component Analysis (PCA) has been described in Section **2.2.1** . PCA is here used with the singular value decomposition as in Algorithm **2** since it is numerically more stable than the by Pearson proposed eigen decomposition of the covariance matrix. Singular Value Decomposition (SVD) has been described in Section **2.2.2** .
Again, as input the list of bee positions is taken. They are transformed into a NxMmatrix X, where N is the number of positions and M is the number of coordinates, here 2. The data is then centered along the rows and SVD is used to decompose the matrix into $X = U \sum V^T$. The first Principal Component is taken from V. Since, as Pearson mentioned, the Principal Component gives us the direction closest to the data, the Principal Component is taken as the best approximation line to the points. Then, the orientation of the found Principal Component is computed with the help of the inverse of the tangent function. Since the Principal Component is a singular vector, it does not care for directions. The

direction of the data has to be found in an extra step. Section **3.6** was used, the mean difference of the coordinates. This might not be as specific as PCA, but it does take the direction into account. The PCA angle was then compared to the angle from Section **3.6**. A threshold distance of $\frac{\pi}{2}$ was assumed, above which the PCA-angle has to be negated. $\frac{\pi}{2}$ was chosen to compensate a certain error in the computation of both orientations.

3.7.2 Fine tuning

A few optimizations were made to the algorithm in order to catch data specific problems. The first was the removal from gaps in the frames. If, during detection, there was a detection gap, the position for those frames was given with the artificial coordinates:

$$< -1, -1 > \tag{3.2}$$

Those values do not represent a real bee position, but rather an outlier that would disturb the result. They are removed before computation of the angle.
The second improvement is a countermeasure to the delay of detection. As described in Section **2.3.3** one problem that occurs through delay is the adding of points that belong to the return run of the bee. This would shift the angle to the left or the right, depending on the direction the bee takes to return, as in figure 2.4. Removing the points from the delay removes the danger of such a shift. The first and last rows of the matrix get pruned before using PCA in order to achieve this.

3.8 Summary

In this chapter the different final attempts of orientation decoding were shown. All relate to the proposed and tried methods from the Waggle Dance Detector, in order to compare them and find the best one. In addition, programs with major contribution to the results of this thesis were introduced.

4 Evaluation and Discussion

In this section the different approaches will be discussed and evaluated. First the basis for evaluation, the ground truth, will be explained. Then for each approach the different tested algorithms are evaluated and the final decision explained. In the end, the final algorithms are evaluated against each other in order to find the best one.

4.1 Ground Truth

To evaluate the different methods against each other, a common ground truth has to be defined. For this, the Beehouse data was chosen. It was already used for parts of the Waggle Dance Detector evaluation, although not for the orientation evaluation. Nevertheless, here it is used because it was a realistic case in which the WDD was used. In conclusion, it will provide realistic circumstances for the evaluation. As mentioned before, the Beehouse data consists of 24h-recordings from 48 days, 15.07.2014 until 03.08.2014. From this time 220127 waggle runs were detected. As is known, the detection delivered too many false positives due to an error in the detection. Thus it has to be taken into account, that some decoded waggle runs might be false positives. For this thesis the ground truth angles were measured through human work. From the Beehouse data set two days, 23.08.2014 and 24.08.2014, were taken. Then random waggle runs were chosen and with the program from Section **3.2** orientations were measured by hand. They were written to a CSV-file in the folder of the measured run. The Beehouse data set was captured and processed with the WDD. As a consequence only output data from the WDD was available to define the ground truth could.

4.2 Image subtraction

To gain the orientation from the WDD-images, a row of different takes were tested. The first was the filtering of the background. Here from all images of a run the background was defined and then subtracted from each image. This approach was discarded. Not only the dancing bee moves in the images but a lot of bees around her. Accordingly this method did not deliver the path of the bee and was discarded.

The second approach was the opposite to the first: Image subtraction to retrieve the movement. Here the bees movement in the result was identifiable for the human eye in the resulting image I_{sum} . Different methods were tried to extract the movement through an algorithm. For all methods, the image I_{sum} was computed as explained in Section **3.4**. First, transformation into a binary image was applied. For this the grayscale image is transformed into a binary image by a given threshold. This threshold was defined through the histogram of the image. A threshold from a histogram is usually chosen as a minimum between two maxima in the histogram Parker [4]. Here, Otsu's method was used.

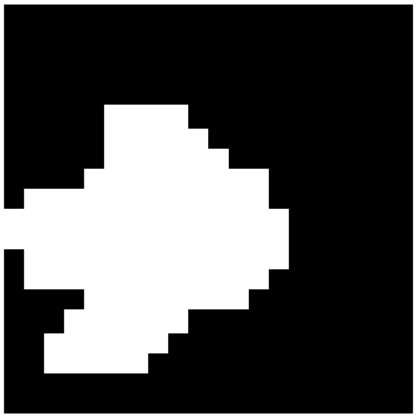


Figure 4.1: Thresholding of the difference image from 20140823_14_03_0_8 with Otsu's method.

The next approach was edge detection on the difference image, again with Otsu's method as a threshold. Different methods, e.g. Gaussian filter or Sobel approximation were used.

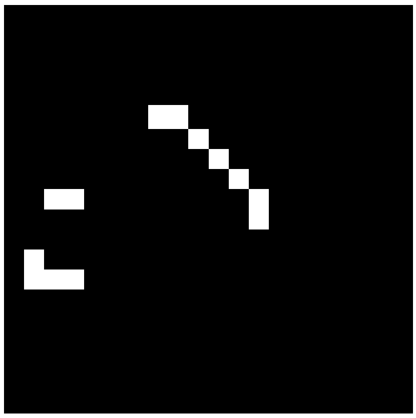


Figure 4.2: Edge detection on the difference image from 20140823_14_03_0_8 with Sobel

For all methods it became soon clear that not the method but the threshold choosing was the problem. A single threshold was not sufficient. The histogram histogram shows that it does not suffice to chose a threshold that equally minimizes the variance of two classes. The bee path only amounts to a small part of the image with a small variance in color values. The background has a comparably high amount of different values and takes in a huge part of the image. When looking at the histogram, the bee path can be found at one of the local maxima with the highest color values. To conquer this problem, Otsu's multithreshold method was chosen and, finally, used with the thresholding.

During the testing, two other problems became obvious. The first was, that there often was a lot of movement around the dancing bees. Bees that followed the dancing bee, but also bees that just moved independent from the dancer. This problem could be overcome with the threshold, since, over time, the main movement still comes from the dancing bee. The second problem was the bee's movement itself. The bee does not only move forward, she waggles. This waggle motion is orthogonal to the forward motion. The bee swings her body left and right while walking forward. This expands the room of interest into a new direction. A solution seemed to present itself, because the waggling does happen parallel to the bees body on both sides. While the difference between two frames is not that huge where the bee walks forward, it is far bigger where she waggles. That way, instead of the expected highlighted path of the bee, two parallel highlighted waggle paths were received, as can be seen in the image(a) of figure 3.3. Both were parallel to her far darker actual path and thus had the same angle. But at the bees start position most of the waggle paths are joined and could not be recognized as single objects, as can be seen in the image(b) of 3.3. Image processing was considered as an alternative to the coordinates by the WDD. But none of the above methods yielded coordinates that were satisfiable enough to perform orientation decoding on them.

4.3 First and last coordinates

Since this a basic approach, no variations were tested. When tested against the ground truth, it did perform better than expected. But, as mentioned, it does slip for some dances due to changes in one of both coordinates. This results in errors up to 54.4138 °. But despite the slips it did perform better than Rau's approach, with an standard deviation of 27.3759 °, compared to 36 ° from Rau's thesis.

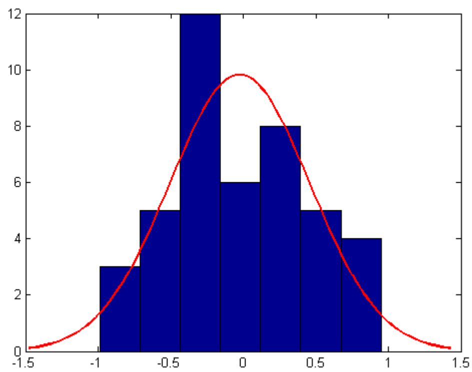


Figure 4.3: The errors of the angle from first to last coordinates against the ground truth. Angles are here measured in radians

4.4 Mean direction

As discussed, this approach does tackle at least some of the issues of the method Rau presented in his thesis. It takes more coordinates into account, which should make the result less vulnerable to errors in single coordinates. Surprisingly, it had a worse standard deviation than the approach from Section **3.5** , with 28.2697 ° . This might be explainable through the way the coordinates are incorporated. If the bee changes from her path, maybe because of another bee, the difference of the coordinates show this change in direction. The algorithm does not care if this fits the main direction of all coordinates, it incorporates the value just the same. Hence, there are more coordinates that can counter such an outlier but an outlier gets more influence on the result.

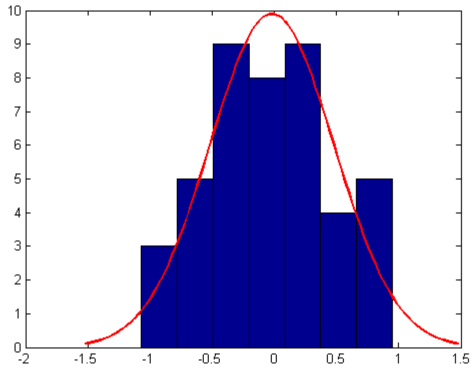


Figure 4.4: The errors of the angle from the mean coordinates vector against the ground truth. Angles are here measured in radians

4.5 Principal Component Analysis

When assuming that the positions of the bee mirrors her movement over time, this approach is the most similar to decoding by humans. As a human, one will try to draw a line through the points that approximates them as good as possible. One might, to say it simple, take a ruler and try to place it on the image so that it fits the points from our view. PCA is the statistical equivalent to that approach. It tries to find a line through the points that minimizes its distance to all points. Normally not a line but a plane would be received, but due to the number of variables the first PC can be directly taken as a line. The problem of PCA to deal with outliers could also be minimized due to sample cropping and the fact, that the bees movement is limited by time and space. PCA proved to have the best results in terms of standard deviation from all tested methods. It has a standard deviation of 23.1876 ° .

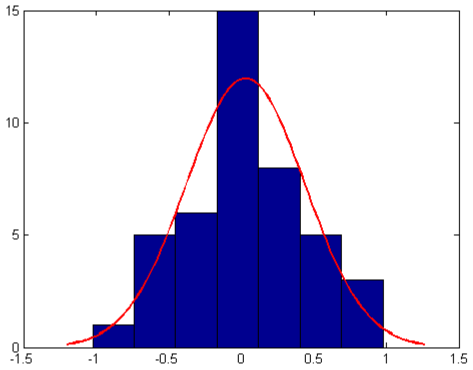


Figure 4.5: The errors of the angle from the Principal Component against the ground truth. Angles are here measured in radians

4.6 Comparison

All in this thesis presented methods have been tested on the same data set. For each, standard deviation and mean error were computed.

Line between coordinates	$27.3759\,^\circ$	$-1.4209\,^\circ$
Mean coordinates	$28.2697\,^\circ$	-1.1230°
PCA	$23.1876\,^\circ$	$1.7361\,^\circ$

Table 4.1: Results from evaluation with the Beehouse data

In terms of the mean error, all methods show a similar error. For the standard deviation the best approach is the Principal Component Analysis with Singular Value Decomposition. It shows a considerable improvement over the other approaches.

Line between coordinates	$27.3759\,^\circ$	$-1.4209\,^\circ$
Mean coordinates	$28.2697\,^\circ$	-1.1230°
PCA	$23.1876\,^\circ$	$1.7361\,^\circ$
Rau	$36\,^\circ$	$-1.06\,^\circ$

Table 4.2: Comparison of results with WDD prototype

As the presented methods, the implementation from the Waggle Dance Detector shows an error of about $-1\,^\circ$. Here no considerable differences occur. When comparing the standard deviation with the results from the evaluation of the WDD, the PCA shows an even greater improvement. It has an improvement of nearly $13\,^\circ$. This is not only an improvement in the detection itself, it also shows that PCA is more reliable. As predicted, it conquers the problems of single wrong measured coordinates as well as the delay. Since bees tend only to dance when the food source is at least 150 m away from the hive, a $13\,^\circ$ improvement means an error decrease of about 60m to the food source.

Although it has a higher mean error than the WDD-implementation it is preferable due to its reliability. One fact that specifically underlines the reliability of PCA is, that the WDD-implementation showed a number of false detection with up to 90° and an isolated amount of errors around 180°. Such high errors could be completely removed in all proposed methods. In PCA even worst case decoding delivered errors of not more than 60°.

4.7 Summary

This chapter evaluated each method for itself with regards to both its performance and the idea behind it. Then all methods were compared with each other and the evaluation results known from Raus thesis. It then is evaluated whether the new approach is a gain compared to the current state of the Waggle Dance detector.

5 Conclusion & Future work

The goal of this thesis to find a more stable algorithm was fulfilled. Since the by Rau implemented method did already have an acceptable mean error, here no improvement could be made. For future use, it would be interesting to calculate the mean error of the proposed methods against a bigger set of ground truth data. Since this is time consuming work such a set could not be provided in the means of this thesis.

In future work, it would be advisable to concentrate on image processing. Especially processing of real-time data from unscaled video footage to find the orientation might prove resourceful. It could deliver the missing details for recognition the WDD data does not provide. The current WDD output images have allow resolution. Higher resolution might enhance the results of edge detection. The problem of unwanted bee movement could be conquered with detection of the bee in the first frame and measurement of her body angle. This is currently not possible on the output data from the Waggle Dance Detector. For future improvement it would be another point of interest to remove the delay from detection with focus on the delayed stopping of recognition. Rau did mention, that the position recognition does work well, but he also mentioned that it was not sufficiently evaluated. Here, a proper evaluation would be of interest, in order to find probable issues or to confirm its correctness. For future application of the Waggle Dance Detector the proposed waggle orientation decoding could be written in C++ and integrated. That way post processing could be avoided.

Appendices

j	Flag it as a dance.
n	Flag it as no dance.
v	Flag that it is unsure if there is a dance.
r	Reset the program.
escape	Stop the program.
left and right arrows	Skip through the images.
F2	Save the current orientation of the line.
F1	Save flag and angle to a angle.csv file in the run-folder.

Table .1: Different keys for the program in Section **3.2**

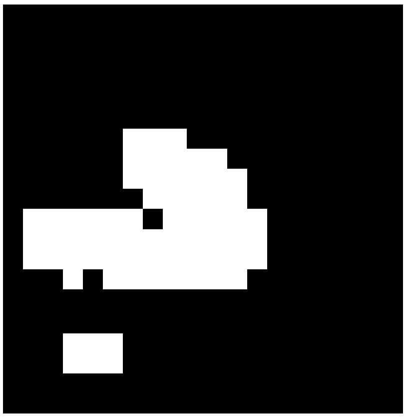


Figure .1: Edge detection on the difference image of 20140823_14_03_0_8 with laplace

Bibliography

[1] Aristotle, D.M. Balme, and A. Gotthelf. *Aristotle: 'Historia Animalium': Volume 1, Books I-X: Text*. Cambridge Classical Texts and Commentaries. Cambridge University Press, 2002. ISBN 9780521480024. URL `https://books.google.de/books?id=tpJE0isZCeUC`.

[2] Tim Landgraf, Raúl Rojas, Hai Nguyen, Fabian Kriegel, and Katja Stettin. Analysis of the waggle dance motion of honeybees for the design of a biomimetic honeybee robot. *PLoS ONE*, 6(8):e21354, 08 2011. doi: 10.1371/journal.pone.0021354. URL `http://dx.doi.org/10.1371%2Fjournal.pone.0021354`.

[3] Nobuyuki Otsu. A Threshold Selection Method from Gray-level Histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, January 1979. ISSN 0018-9472. doi: 10.1109/tsmc.1979.4310076. URL `http://dx.doi.org/10.1109/tsmc.1979.4310076`.

[4] J. R. Parker. *Algorithms for Image Processing and Computer Vision*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1996. ISBN 0471140562.

[5] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.

[6] Alexander Rau. Realtime Honey Bee Waggle Dance Decoding System. Master's thesis, Freie Universitä Berlin, Germany, 2014.

[7] T.D. Seeley. *THE WISDOM OF THE HIVE*. Harvard University Press, 1995. ISBN 9780674953765. URL `https://books.google.de/books?id=zjggAQAAMAAJ`.

[8] Linda G. Shapiro and George C. Stockman. *Computer vision*. Prentice Hall, Upper Saddle River, NJ, 2001. ISBN 0-13-030796-3. URL `http://opac.inria.fr/record=b1128947`.

[9] Johann Ernst Spitzner. *Ausführliche theoritische und praktische Beschreibung der Korbbienenzucht, nach ausgemachten Gründen der Naturlehre und langer eigener Erfahrung*. Friedrich Pohl, 1788.

[10] G. W. Stewart. On the early history of the singular value decomposition, 1992.

[11] D.R. Tarpy and North Carolina Cooperative Extension Service. *The Honey Bee Dance Language*. AG (Series). N.C. Cooperative Extension Service, 2004. URL `https://books.google.de/books?id=4PxTMQEACAAJ`.

[12] K. von Frisch. *Tanzsprache und Orientierung der Bienen*. Springer-Verlag, 1965. URL `https://books.google.de/books?id=7B82AAAAMAAJ`.

[13] Anja Weidenmüller and Thomas D. Seeley. Imprecision in waggle dances of the honeybee (apis mellifera) for nearby food sources: Error or adaptation? *Behavioral Ecology and Sociobiology*, 46(3):pp. 190–199, 1999. ISSN 03405443. URL `http://www.jstor.org/stable/4601661`.