Behavioral clues for forager transitions in a fully tracked honeybee colony

Janek Szynal

Supervisor: Dr. Tim Landgraf

Biorobotics Lab
Free University of Berlin

This dissertation is submitted for the degree of
Bachelor of Computer Science

April 2019

I would like to dedicate this thesis to my loving parents ...

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Janek Szynal
April 2019

Acknowledgements

And I would like to acknowledge ...

Abstract

This is where you write your abstract ...

Table of contents

List of figures

List of tables

Chapter 1

Introduction - 95%

1.1 The honeybee and the division of labor

A honeybee (*Apis mellifera*) colony manifests multiple fascinating examples of complex adaptive behaviour. Localized cues exchanged between individuals amount to emergent directional signals for the entire colony in ways heavily investigated, but often still not completely understood.

One of the most notable and well-researched adaptive mechanisms of a colony is its division of labor (DOL). During the winter (a season of low activity for the bees), the colony focuses on survival and its workers are generalists, performing sets of tasks not easily distinguishable from those of other workers.

For the spring-summer season, however, the hive's goals change and along with them, the patterns of labor division. Hive growth and resource accumulation take priority, and specialization eventuates amongst workers. They begin to fill distinct roles, the allocation of which highly correlates with age (an effect known as temporal polyethism) - but is also grounded in the colony's current needs and in environmental factors affecting it (adaptive behaviour) [6]. Groups of workers that can be categorized as performing the same set of tasks are commonly referred to as castes. It is common to recognize four of them in the temporal caste system that the worker bees exhibit in the summer: *cell cleaners, nurses, middle-aged bees (MABs),* and *foragers*. This work concentrates on the transition between *MABs* and *foragers*, possibly the most distinguishable and important in the lifecycle of a bee.

1.2 The foraging phase

The foraging phase is the last one in a bee's life and comes with an increased risk of death. It is often proposed that this has to do with the extreme strain foraging puts on their bodies - essentially causing them to work themselves to death. This is supported for example by [10], who have shown honeybee flight to cause extremely high metabolic rates and induce oxidative stress, likely significantly accelerating the ageing process and causing early deaths. On the other hand, the results of [7] suggest that foragers' deaths are usually caused not by senescence, but rather by the heightened risks of outside life that come with their function (such as the risk of predation). According to their findings, forager mortality rates are constant with respect to age - and not accelerating, as would have been suggested by the body strain hypothesis.

Regardless of the reasons behind it, foraging nearly always ends in the death of a bee. That fact, combined with an estimate of the length of the foraging phase (mean of 7.7 days ± 0.75 days, median of 7 days, and range of 2 to 17 day according to [7], should allow us to get a very simple estimate of the foraging period, which we can then use to validate the results we produce with more involved methods.

1.3 The *BeesBook* project

This work operates on data from the 2016 iteration of the BeesBook project [9]. A hard- and software framework is set up, allowing for high-confidence tracking of all individuals over the entire lifespan of a honeybee colony. It is the first dataset ever collected (to our knowledge) that is extensive enough to provide a comprehensive view of a hive's life, maintaining the spatial, temporal and social context of the information it stores.

The data it collects can be put in 3 categories:

- a list of detections (each assigned to a bee, a point in time and a point in the hive space)
- a collection of individuals' paths (added in [X])
- and a collection of bee waggle dance occurrences (added in [X]).

The original *BeesBook* paper put significant focus on waggle dance research, but the dataset and collection method is meant to serve a very general purpose. The system is described to be "conceived as a budget-priced framework for the incremental development of software and hardware components", capable of supporting a wide range of investigations into invertebrates, the waggle dance, honeybee division of labor, collective intelligence and other related fields.

1.4 This work's goals

This work's primary goal is to add building blocks to the *BeesBook* project by deriving new abstractions that can be used in future analyses and to show how the process of deriving such abstractions could look like.

A secondary goal is to provide an example of how analysis could be undertaken in the future, given the data, the set of abstractions and the process for creating them.

Chapter 2

Related work

I present related work in three categories. The first one is an introduction to invertebrate tracking and automated observation, along with an overview of the state of the art.

The second focuses on the division of labor in honeybees. It's meant to set foundation for the analysis that we undertake in this work, as well as for determining what other kinds of approaches should be accessible given the BeesBook dataset and the building blocks that this work adds to it.

Finally, the last one collects works similar to this contribution - ones that use the *BeesBook* dataset to perform some analysis of a honeybee colony's life and/or add their own functionalities or improvements to the system.

2.1 Invertebrate observation and tracking

Observing invertebrates at scale, before a certain degree of automation was possible, required a lot of careful manual work and some creative approaches. A fascinating example of how experiments were conducted back then can be found in [6]. The authors mark a hundred bees out of a colony of 21 thousand, using a brush with pigment mixed with shellac (following the example set by Karl Von Frisch [8]). They then pick for observation small subsections of the hive (quadrants), employing the help of a Texas Instruments calculator to generate randomness for their choices. Inferences about the entire population are made using the samples, but even to observe the samples, 8 hours of continuous work per day, for over 20 days, was necessary. To create maps of activity, authors used glass sheets that they put markings on and exchanged every day. They then photographed the sheets and projected the photographs against a single sheet of paper, one by one, thereby aggregating the one-day information sets into a single map. They also used a number of other physical and numerical tricks to be able to produce quality data.

One step toward an automated process is to take video recordings and analyze them manually [4], also using markings, sometimes such that identifying a single individual from its marker was possible. This is less error-prone, but analysis of the film requires no less time than real-time observation (and often much more).

Compared to both those methods, modern approaches that analyze video footage automatically save a tremendous amount of effort. Researchers tend to take one of two paths: tracking the animals based on their previous positions and body features or using specialized markers (tags) that are put on individuals for identification.

A prominent and oft-cited example in unmarked tracking is [3], where good results were achieved in tracking unmarked honeybees over short periods of time. They system employed was based on vector quantization and temporal contextual information. It was able to distinguish between individuals solely based on their body size and movement, keeping track of 50% of the hive (350 bees) over 10 seconds.

A recent notable work [1] reports maintaining 71% tracks for over 2 min, around 46% for 5mins. Such approaches are impressive technologically and of great importance by virtue of being generalizable (authors cite cells in tissues and human crowds as examples of potential usage); yet on the problem of invertebrate tracking they do not yet achieve the results of marker-based systems in terms of effectiveness and scale.

They therefore do not allow for analysis that can draw conclusions relating to entire lifespans of individuals, or the colony (which is what I am attempting in this work). For anything related to foraging, systems based on trajectory and body features are particularly unreliable, as they would inevitably lose track of individuals identities whenever those would leave the hive and come back.

TODO: Add Mersch (2013).

A recent experiment with a strong focus on trophallaxis [2] has a lot of similarities to ours in terms of tracking method. It uses tag markers and tracks multiple bees across a longer period of time. Being focused on a more specific problem allowed for a more specialized setup that made tracking easier, but was more intrusive in terms of how it influenced the bees' lifes. The hive was single-sided and designed in a way that prevented the bees from obscuring each others' barcodes (presumably by keeping the space between hive surface and glass ceiling small). The hive was also backlit by infrared light, making the surface easier to distinguish from the insects. These factors (hopefully) did not have a strong adverse effect on the relevant data, but would not work for a holistic observation attempted in BeesBook.

An additional compilation of related work can be found in the attachment section of [5]. It collects 14 contributions in a table format, comparing them with respect to parameters such as species, colony size, duration of study, and tracking method (manual or automatic).

The focus of that comparison is interaction networks - they do, however, still fulfil the more general criteria of relevance to invertebrate traching and to the BeesBook project.

2.2 TODO: drop or write down, DOL reasearch

2.3 Works related to Beesbook

TODO: table-based refactor or text-based refactor

BeesBook is a long-running project with work subsequent to the original paper contributing improvements in the soft- and hardware of the detection stack (Sixt et al., 2016, Wild et al., 2018, Boenisch et al., 2018), as well as various applications of its data in collective behaviour and honeybee research (Adrian* et al, 2018).

RenderGAN Boenisch 2018 Automatic detection and decoding of honey bee waggle dances (Wario 2017) DCCN on markers, Wild 2018

This work is meant to add building blocks to the BeesBook system and derive new abstractions that can be used in future analyses, particularly those pertaining to the forager transition and the foraging phase of a bee's life. While no previous work seems to state that as an explicit goal, a number of them have made significant contributions as a side effect of their examinations.

Chapter 3

Implementation and methods

3.1 Building a structured library of abstractions

BeesBook is well-suited for testing a broad range of hypotheses on the data it collects. This work attempts to use BeesBook's data to pinpoint the moment in which workers transition to foraging, and potentially answer some questions about the circumstances of that transition.

I propose an approach of defining abstractions layered on top of the original form of the data (which is a list of detections). Each abstraction is conceived as a step in a funnel that starts with detections and is meant to produce the status of being a forager (or not).

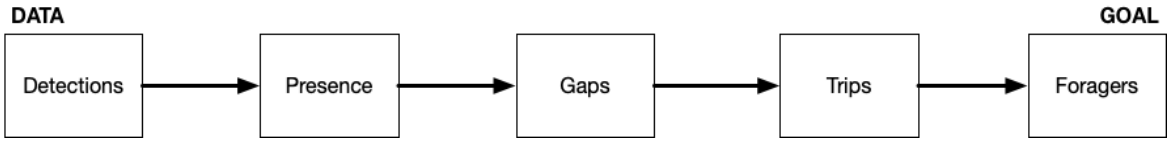

Fig. 3.1 Visualization of the abstraction steps

On top of Detections, we define Presence as a numerical score equal to the number of Detections that were registered for a given bee in a time window of $t = 30$ seconds. On top of Presence, we define Gaps as periods of time where bees had a Presence score of zero or close. On top of that, we define Trips as Gaps that seem to be caused by the bee leaving the hive (as opposed to occlusion, entering brood cells etc.). We finally use Trips as a metric that lets us determine when a given bee has started to forage. A more detailed description of the process, along with a list of potential problems, is presented below.

The first key idea is the ability to reuse such abstractions for work with other research goals. For example, (Marcus, 2019) examines the deaths of bees information below). For research into other parts of the DOL or related in-hive behavior, the first three abstractions

could be reused (e.g. Gaps for an investigation into behavior that relates to cleaning brood cells).

The second key idea is to apply the same methodology for all new research goals. If existing abstractions cannot be reused, defining new ones in a preservable or reproducible way ensures future analyses will be much easier to perform.

3.2 Detections to Presence

A Detection in the BeesBook database is the result of a single bee tag being recognized in a single frame of video. If that tag is also visible in the next frame, this is registered as a new detection. Each detection has information on where the tag was detected (x/y coordinates and orientation), when (exact timestamp), with what confidence and by which of the four cameras.

TODO: TABLE: an example detection

Calculating Presence scores from raw detections is tallying up the number of times a given bee was detected in a 30-sec interval window. The interval size was chosen arbitrarily, albeit with some experimentation. The code was written in a way that allows for easy re-generation of Presence scores with a different interval window size. Another important hyperparameter is the confidence requirement. Every detection has a confidence value assigned to it, representing the likelihood that the tag was decoded properly (i.e. belongs to the bee it was assigned to, and not another one).

Determining a good confidence requirement could conceivably have great influence on the next steps. A very high threshold could mean that too many valid detections are filtered out and artificial troughs in presence score are created in consequence. A low one could mean including detections that did not really occur (should not have occured) and artificial peaks in presence score are created. The way the process is currently designed, it has more robustness towards including fake detections than it has toward discarding real ones. It is therefore reasonable to use a fairly low confidence rate. I use C, which filters out N% detections (based on three randomly chosen sample days. The interval window has less significance and is easier to get right. If it's small, it could lead to unnecessary calculation, too large and it could lead to short artificial gaps in presence. [Mention the framing of this as robustness against false positives vs false negatives]

TODO: FIG: distribution of confidence values in detections from e.g. one given day).

Each camera records with a speed of 3FPS. With an interval size of 30 seconds, that gives a max expected Presence score of 90 (with exceptions - see the Detection/presence problems

section below). The score is then thresholded at X, mapping it to a binary function. That function is smoothed out by morphological closing with a parameter of X.

TODO: FIG: a day of presence, as score and as a binarized function

TODO: stub: A big part of the purpose (advantage of PRES over DET) is reduced size/greater speed. Det file for 1h is 400mb, presence for a day is 12mb.

From the binarized and smoothed out form of presence, Gaps are generated by taking the difference between Presence values for intervals i +1 and i, for all intervals. This identifies exits (whenever the difference is -1) and entries (whenever the difference is 1).

This makes up a long list of all situations, where the video cameras lost track of a bee.

For a sample of those situations, I generate videos for manual analysis.

Foraging trips are the single most defining aspect of the forager's life and a natural starting point for the investigation. No other bee caste is expected to disappear from the cameras' field of vision with such frequency and regularity. We therefore decide on looking closer into trips as our primary method of identifying foragers. We start with a raw list of detections as offered by the BeesBook database (See the format Fig X / Appendix I). In order to convincingly determine an absence period's beginning and end, we divide the time of the entire experiment into intervals of n seconds. For every such interval, we go through all possible bee ids and mark the cell for [given interval, given bee_id) with a 1 if the database contains at least one detection for given bee with a timestamp within the bounds of the given time interval. This produces a matrix of binary values representing every bee's history of hive presence.

Fig. X - Illustration of the presence dataframe, dark cell background meaning present, light meaning absent.

Given the presence matrix, we interpret a bee's absence and reappearance as a trip. In order to prevent misclassified detections (TODO: cite BeesBook's misclassification rate) from artificially increasing the number of trips, we smooth out the presence matrix with a rolling median (window_size = r). This means, that any period of presence or absence shorter than floor(r/2) will be ignored.

We then summarize the amount of trips a given bee has taken on a given day. This hopefully makes foragers distinct from non-foragers

It is important to note that that when requesting data from the database, we pass a 'desired confidence' value c. This value, together with the size of the interval in seconds (n) and the size of the rolling window that smoothes out the detections (r) are three very important hyperparameters that contribute to the accuracy of the resulting trip data and to the credibility of conclusions we might draw from them. <to be continued>

Ideas: talk about false positives and false negatives with a presence map for all bees, just take some statistical hints - (dis)appearence proximity to exit

References

[1] Bozek, K., Hebert, L., Mikheyev, A. S., and Stephens, G. J. (2018). Pixel personality for dense object tracking in a 2d honeybee hive. *arXiv:1812.11797 [cs, q-bio, stat]*. 00000 arXiv: 1812.11797.

[2] Gernat, T., Rao, V. D., Middendorf, M., Dankowicz, H., Goldenfeld, N., and Robinson, G. E. (2018). Automated monitoring of behavior reveals bursty interaction patterns and rapid spreading dynamics in honeybee social networks. *Proceedings of the National Academy of Sciences*, 115(7):1433–1438.

[3] Kimura, T., Ohashi, M., Okada, R., and Ikeno, H. (2011). A new approach for the simultaneous tracking of multiple honeybees for analysis of hive behavior. *Apidologie*, 42(5):607–617.

[4] Naug, D. (2008). Structure of the social network and its influence on transmission dynamics in a honeybee colony. *Behavioral Ecology and Sociobiology*, 62(11):1719–1725.

[5] Schlegel, A. (2017). Temporal Analysis of Honey Bee Interaction Networks Based on Spatial Proximity. Master's thesis, Freie Universität Berlin. 00000.

[6] Seeley, T. D. (1982). Adaptive significance of the age polyethism schedule in honeybee colonies. *Behavioral Ecology and Sociobiology*, 11(4):287–293.

[7] Visscher, P. and Dukas, R. (1997). Survivorship of foraging honey bees. *Insectes Sociaux*, 44(1):1–5. 00096.

[8] von Frisch, K. (1965). *Tanzsprache und Orientierung der Bienen*. Springer.

[9] Wario, F., Wild, B., Couvillon, M. J., Rojas, R., and Landgraf, T. (2015). Automatic methods for long-term tracking and the detection and decoding of communication dances in honeybees. *Behavioral and Evolutionary Ecology*, page 103.

[10] Williams, J. B., Roberts, S. P., and Elekonich, M. M. (2008). Age and natural metabolically-intensive behavior affect oxidative stress and antioxidant mechanisms. *Experimental Gerontology*, 43(6):538–549. 00096.

Appendix A

How to install LaTeX

Windows OS

TeXLive package - full version

1. Download the TeXLive ISO (2.2GB) from
   https://www.tug.org/texlive/
2. Download WinCDEmu (if you don't have a virtual drive) from
   http://wincdemu.sysprogs.org/download/
3. To install Windows CD Emulator follow the instructions at
   http://wincdemu.sysprogs.org/tutorials/install/
4. Right click the iso and mount it using the WinCDEmu as shown in
   http://wincdemu.sysprogs.org/tutorials/mount/
5. Open your virtual drive and run setup.pl

or

Basic MikTeX - TeX distribution

1. Download Basic-MiKTeX(32bit or 64bit) from
   http://miktex.org/download
2. Run the installer
3. To add a new package go to Start » All Programs » MikTex » Maintenance (Admin)
   and choose Package Manager

4. Select or search for packages to install

TexStudio - TeX editor

1. Download TexStudio from
   http://texstudio.sourceforge.net/#downloads

2. Run the installer

Mac OS X

MacTeX - TeX distribution

1. Download the file from
   https://www.tug.org/mactex/

2. Extract and double click to run the installer. It does the entire configuration, sit back and relax.

TexStudio - TeX editor

1. Download TexStudio from
   http://texstudio.sourceforge.net/#downloads

2. Extract and Start

Unix/Linux

TeXLive - TeX distribution

Getting the distribution:

1. TexLive can be downloaded from
   http://www.tug.org/texlive/acquire-netinstall.html.

2. TexLive is provided by most operating system you can use (rpm,apt-get or yum) to get TexLive distributions

Installation

1. Mount the ISO file in the mnt directory

   ```
   mount -t iso9660 -o ro,loop,noauto /your/texlive####.iso /mnt
   ```

2. Install wget on your OS (use rpm, apt-get or yum install)

3. Run the installer script install-tl.

   ```
   cd /your/download/directory
   ./install-tl
   ```

4. Enter command 'i' for installation

5. Post-Installation configuration:
   http://www.tug.org/texlive/doc/texlive-en/texlive-en.html#x1-320003.4.1

6. Set the path for the directory of TexLive binaries in your .bashrc file

For 32bit OS

For Bourne-compatible shells such as bash, and using Intel x86 GNU/Linux and a default directory setup as an example, the file to edit might be

```
edit $~/.bashrc file and add following lines
PATH=/usr/local/texlive/2011/bin/i386-linux:$PATH;
export PATH
MANPATH=/usr/local/texlive/2011/texmf/doc/man:$MANPATH;
export MANPATH
INFOPATH=/usr/local/texlive/2011/texmf/doc/info:$INFOPATH;
export INFOPATH
```

For 64bit OS

```
edit $~/.bashrc file and add following lines
PATH=/usr/local/texlive/2011/bin/x86_64-linux:$PATH;
export PATH
MANPATH=/usr/local/texlive/2011/texmf/doc/man:$MANPATH;
export MANPATH
```

```
INFOPATH=/usr/local/texlive/2011/texmf/doc/info:$INFOPATH;
export INFOPATH
```

Fedora/RedHat/CentOS:

```
sudo yum install texlive
sudo yum install psutils
```

SUSE:

```
sudo zypper install texlive
```

Debian/Ubuntu:

```
sudo apt-get install texlive texlive-latex-extra
sudo apt-get install psutils
```

Appendix B

Installing the CUED class file

LaTeX.cls files can be accessed system-wide when they are placed in the <texmf>/tex/latex directory, where <texmf> is the root directory of the user's TeXinstallation. On systems that have a local texmf tree (<texmflocal>), which may be named "texmf-local" or "localtexmf", it may be advisable to install packages in <texmflocal>, rather than <texmf> as the contents of the former, unlike that of the latter, are preserved after the LaTeXsystem is reinstalled and/or upgraded.

It is recommended that the user create a subdirectory <texmf>/tex/latex/CUED for all CUED related LaTeXclass and package files. On some LaTeXsystems, the directory look-up tables will need to be refreshed after making additions or deletions to the system files. For TeXLive systems this is accomplished via executing "texhash" as root. MIKTeXusers can run "initexmf -u" to accomplish the same thing.

Users not willing or able to install the files system-wide can install them in their personal directories, but will then have to provide the path (full or relative) in addition to the filename when referring to them in LaTeX.