

LabelBee: a web platform for large-scale semi-automated analysis of honeybee behavior from video

Rémi Mégret¹

Edgar Acuña³

Ivan F. Rodriguez²

Jose L. Agosto-Rivera⁴

Isada Claudio Ford⁴

Tugrul Giray⁴

¹Computer Science Dpt.
University of Puerto Rico,
Río Piedras campus
San Juan, Puerto Rico
remi.megret@upr.edu

²Dpt. of Mathematics
University of Puerto Rico,
Río Piedras campus
San Juan, Puerto Rico
ivan.rodriguez5@upr.edu

³Dpt of Mathematical Sciences
University of Puerto Rico,
Mayagüez campus
Mayagüez, Puerto Rico
edgar.acuna@upr.edu

⁴Biology Department
University of Puerto Rico,
Río Piedras campus
San Juan, Puerto Rico
{isada.claudio,jose.agosto1,
tugrul.giray}@upr.edu

ABSTRACT

The LabelBee system is a web application designed to facilitate the collection, annotation and analysis of large amounts of honeybee behavior data from video monitoring. It is developed as part of NSF BIGDATA project “Large-scale multi-parameter analysis of honeybee behavior in their natural habitat”, where we analyze continuous video of the entrance of bee colonies. Due to the large volume of data and its complexity, LabelBee provides advanced Artificial Intelligence and visualization capabilities to enable the construction of good quality datasets necessary for the discovery of complex behavior patterns. It integrates several levels of information: raw video, honeybee positions, decoded tags, individual trajectories and behavior events (entrance/exit, presence of pollen, fanning, etc.). This integration enables the combination of manual and automatic processing by the biologist end-users, who also share and correct their annotation through a centralized server. These annotations are used by the Computer Scientists to create new automatic models, and improve the quality of the automatic modules. The data constructed by this semi-automatized approach can then be exported for the analytic part, which is taking place on the same server using Jupyter notebooks for the extraction and exploration of behavior patterns.

CCS CONCEPTS

• Applied computing – Life and medical sciences – Computational biology • Information systems – Information systems applications – Multimedia information systems • Human-centered computing – Visualization – Visualization systems and tools

KEYWORDS

Video analysis, Behavior Monitoring, Data Quality, Human-in-the-loop

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
AIDR '19, May 13–15, 2019, Pittsburgh, PA, USA

© 2019 Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-7184-1/19/05...\$15.00
<https://doi.org/10.1145/3359115.3359120>

ACM Reference format:

Rémi Mégret, Ivan F. Rodriguez, Isada Claudio Ford, Edgar Acuña, Jose L. Agosto-Rivera, Tugrul Giray. 2019. LabelBee: a web platform for large-scale semi-automated analysis of honeybee behavior from video. In *Proceedings of Artificial Intelligence for Data Discovery and Reuse (AIDR '19)*. 4 pages.

1 Introduction

This paper presents the LabelBee system, composed of a graphical interface and back-end system, to support the analysis of videos of honeybees at the entrance of their colony (see Figure 1). Such a system is critical for the analysis of behavior of honeybees at large-scale to uncover new biology insights into task specialization and individual variations, and their effect on colony health and performance. Such biological studies require a combination of both the identification of individuals over long periods of time and a diversity of tasks. In this work, the focus was put on the foraging behavior (leaving the colony to collect resources), with an important value added by the video to classify the payload of returning foragers, as they bring pollen or other resources. Behaviors such as fanning (moving the wings to circulate the air in the hive) can also be observed. This data is typically not available through other means, such as RFID tags or bee counters, which focus on the entrance and exits but ignore other behavioral aspects visible in the videos.



Figure 1. Video collection system and sample image that capture the honeybee behavior at the entrance of their colony.

The originality of the LabelBee platform is to provide a unique set of functionalities for large-scale monitoring of large amount of individuals for a diverse set of behaviors. Being a hosted system

accessed through a simple web browser, it enables teams of biologists to analyze large sets of videos, and share their annotations, as well as train models from the annotations. These properties make the platform particularly extensible to the analysis of the behavior of other insects or animals.

2 Overview

The videos of behavior have been acquired using the setup described in more details in [1] and illustrated in Figure 1. The videos capture the entrance ramp of the colony, showing entrances and exits, as well as behaviors on the ramp such as fanning or guarding. When bees leave or enter the colony, they typically spend very little time on the ramp, crossing it in about 1 to 2 seconds. This amount in our current setup to 20 to 40 frames at 20 frames per second of video. In one hour, around 2000 such events may occur in total (see Figure 7), which prevents any analysis done purely visually.

These videos are uploaded to the system and can then be visualized and annotated in the LabelBee graphical user interface (GUI) shown in Figure 2. This interface is accessed through a Web browser. It allows the user to select a video and visualize it with random access to any frame. If the video was preprocessed with tag detection, a chronogram will show for each bee tagged with a unique barcode the intervals of time at which it is visible on the ramp.

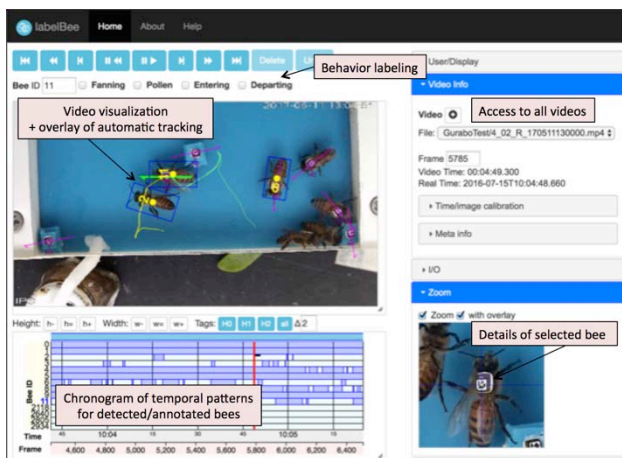


Figure 2. Overview of the LabelBee GUI.

Any bee can be selected, and annotated with behavioral labels: Leaving, Entering, bearing Pollen, Fanning, etc.. These annotations can then be saved and loaded locally or on the server. This feature enables a collaborative process where annotations done by one user can be reviewed and corrected by another user.

When selecting a bee, its detailed appearance can be displayed, which is useful to decide whether or not it bears pollen balls. This tool also allows the user to annotate body parts, which can be used to provide data to train Artificial Intelligence (AI) modules in the automatic detection and tracking of all bees, including the untagged ones. Additional analysis can be performed outside of

the graphical interface, which include tag bees detection, untagged bee detection and tracking, recognition of pollen bearing bees, recognition of fanning bees, identification of untagged bees. These tasks rely of ad-hoc Python scripts that are accessed through Jupyter notebooks, and interface with the annotation file formats of the GUI. (see Section 5)

3 Architecture

The general architecture of the currently deployed system is shown in Figure 3. The video data, acquired at the UPR Agricultural Research Station of Gurabo, is transferred to the UPR High Performance Computing facility (HPCf). This process is performed using external hard disks to account for the lack of Internet connection at the collection site. in the field. The raw video data, in AVI H264 format, is then organized by colony, date and time on the data storage server, and converted using FFMPEG to MP4 H264 format for processing and visualization. The conversion is uses FFMPEG option `-movflags faststart`, so that video headers are moved to the beginning of the file to be able to use these files directly in the browser.

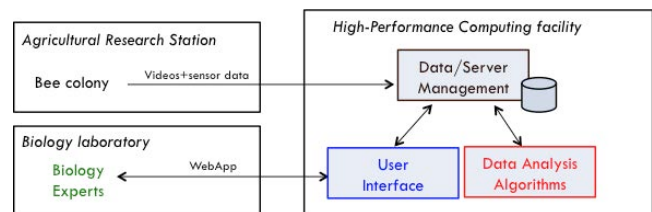


Figure 3. General architecture of the LabelBee system

The user interface for visualization and annotation of the video data is developed as a Web application hosted on a webserver virtual machine of the HPCf, configured to access the videos stored on a dedicated storage node through a NFS shared partition over 1Gb/s Ethernet interconnect. The WebApp itself is implemented using the Flask Python framework. It is served by an Apache HTTP Server, which provides site level password based access, and act as a reverse-proxy to the Flask app.

For performance issues, large static data such as video is not accessed through Flask, but served directly by Apache HTTP. This allows efficient handling of video transmission. For current setup, using a 5Mpix video at 20 fps, each individual 1h video takes around 3.6GB, which is not downloaded completely by the user's browser. We rely instead on the integrated caching of the browsers to perform "range requests" of only the part of the video currently being visualized, thus reducing the amount of data to be transmitted.

Data analysis algorithms are deployed on a separate node at HPCf, equipped with two NVIDIA GPU P100 co-processors. Currently, the annotation and analysis tasks are performed separately, although they may be integrated in a common graphical interface in the future. The output of automatic analysis can be loaded in the LabelBee GUI by exporting it to the Events

file format, which encapsulate the identity, position and behavior labels associated to each bee in a video. Some tasks require training data, which is provided in turn by manual annotation using the GUI, and exported in the same format.

This integration has defined the LabelBee system as a central part of the multi-disciplinary collaboration between Biologists and Computer Scientists which follows a iterative philosophy. As a first step, the interface enabled the Biologists to perform manual annotation of the videos at small to medium scale. These annotations were then leveraged by the Computer Scientists as training data to provide partial automation of the analysis, and scale-up the analysis to a much larger set of videos. The results obtained automatically can be visualized in their initial context by Biologists in case validation is necessary, or to perform part of the analysis that could not be automatized.

4 Event annotation

To facilitate annotations, bees are marked with tags printed on waterproof paper, including AprilTag [3]. In practice, 5x5 tags from the 25h5 family were used, which enable 3009 distinct ids. The detection uses a python wrapper around the apriltag binaries, to extract both position and id from images. Once these detections are performed in batch mode, they can be loaded in the LabelBee GUI to guide the user annotation. Users can visualize the tags (See Figure 4), and mark any incorrect detection, which was used to implement filters to remove false positives.

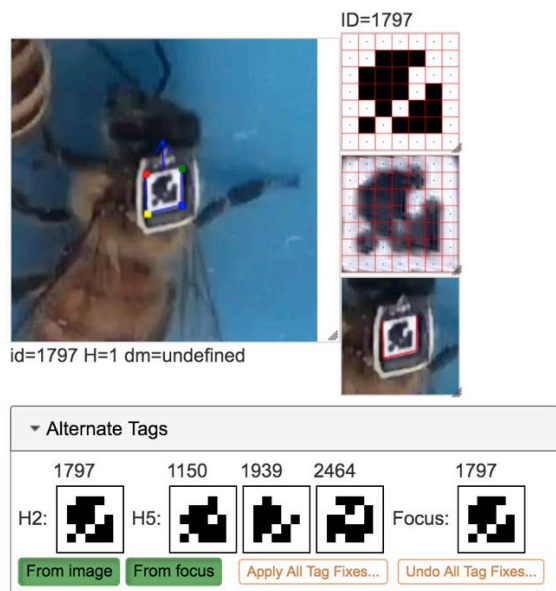


Figure 4. Interface for the visualization of individual tag detection for validation by user. Left: input image with tag corner overlaid. Right, from bottom to top: image detail, aligned tag extracted from the image, ideal tag for decoded id. Bottom: alternate tags proposed by the GUI based on visual similarity to fix incorrect identification.

The annotation of behavior is performed in the main GUI by creating events associated to one individual at a specific time and attaching labels from a list of behaviors (See Figure 5). To facilitate the annotation, the chronogram below the video show all events as colored circles to represent the different types of behaviors. Any tagged bee that appears in the video and that is not annotated will appear as blue rectangles, informing the user of the times in the video that require their attention.

Each event can be associated a time-span, to account for behaviors such as fanning, which can start and stop at any time while the bee is visible. This is in contrast to entrance and exits that are typically time-stamped at the time at which the bee passes an imaginary line at the limit of the ramp, and that are considered to be associated to the complete trajectory of the bee crossing the ramp. Curation and cleaning of the data is done following a multi-step process: a first annotation is performed by a volunteer, which is later cleaned by a second volunteer. The last version is then reviewed by an expert user to ensure uniform quality of annotations. Any overlapping events are highlighted by the system to facilitate this cleaning. This process is a key element to obtain good quality annotations, which is important to train automatic models that perform well.

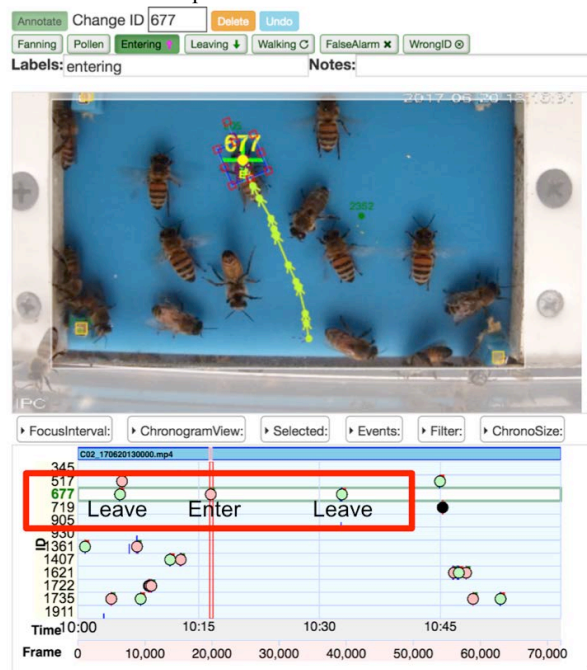


Figure 5. Example of foraging events shown in the GUI. Top: Labels associated to the selected trajectory in the video. Bottom: Timeline of bee #677 over 1h of video, with several entrance/exit events.

5 Data Analysis

The automatic analysis consists of a combination of modules that were trained on data provided by the manual annotation of the videos presented in previous section. It starts with the automatic

pose estimation of the individual bees to detect their body parts (head, thorax, abdomen, antennas) and connect them into individual skeletons as shown in Figure 6, using the method presented in [2] based on Deep Convolutional Networks and a Parts Affinity Fields approach.



Figure 6. Pose estimation of multiple honeybees using [2].

Tracking is then performed to produce trajectories. Classification of events is done using the geometry of the track (length, direction, starting point and ending point) into entrance and exit [4]. Detected bees are also classified into pollen and non-pollen bearing bees using a convolutional neural network [1] applied to geometrically compensated images obtained using the estimated pose. The complete pipeline runs at an average of 5 fps using one NVIDIA P100 GPU with 12GB RAM. Newer modules, such as fanning detection [5] and identification of untagged bees [6] are in the process of being integrated.

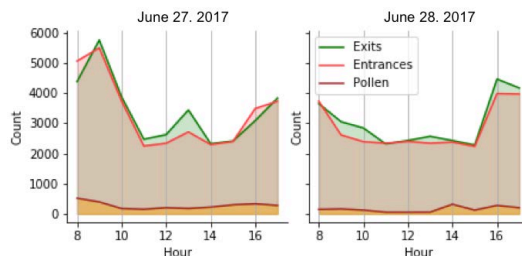


Figure 7. Actograms of foraging behavior of all bees generated by Jupyter notebooks, showing the evolution of counts of exits/entrances/entrances-with-pollen accumulated per hour, over 2 days (8am to 5pm).

The manual annotations and results of automatic analysis are used in a separate interface based on Jupyter Python notebooks, where the raw behavior data can be manipulated and visualized. Results of the automatic pipeline are shown in Figure 7, where the foraging behavior for each day is summarized by an hourly count of all entrances and exits. Predefined visualization functions can be called in Python to explore the behavior of individual tagged bees in various forms as illustrated in Figure 8. At this stage of the project, the use of Jupyter notebooks enables the flexibility of script-based exploration. Implementation of the most used workflows in a more integrated interface associated to the annotation system is considered for future work.

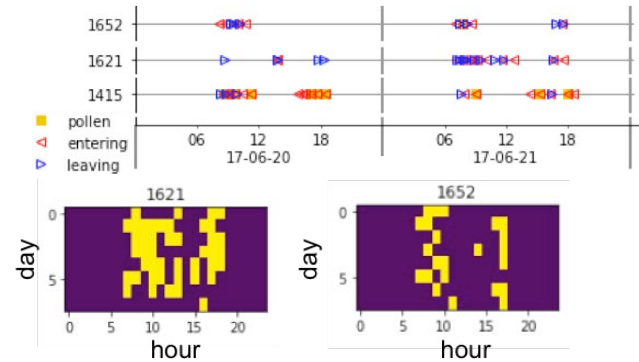


Figure 8. Example of visualization of foraging behavior of tagged bees generated in a Jupyter notebook from LabelBee annotations. Top: occurrences of the individual events for 3 bees (#1652, #1621, #1415) over 2 days of monitoring, showing exits (blue triangle), entrances (red triangle) and entrances with pollen (yellow square). Bottom: multiday actograms for two individual bees (#1621 and #1652), showing different behavior patterns for the entrance/exit events over 8 days.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1707355 and 1633184. UPR HPCf acknowledges the University of Puerto Rico and the Institutional Development Award (IDeA) INBRE grant P20 GM103475 from the National Institute for General Medical Sciences (NIGMS), a component of the National Institutes of Health (NIH) and the Bioinformatics Research Core of INBRE. Thanks to undergraduate students Shirley Serralles, Grace Rodriguez, Christian Esteves, Emmanuel Nieves for their involvement in software development, and to Janpierre Aleman and graduate student Stephanie Feliciano for their help in the set up of the data collection at the colony.

REFERENCES

- [1] I. F. Rodriguez, R. Megret, E. Acuna, J. L. Agosto-Rivera and T. Giray, "Recognition of Pollen-Bearing Bees from Video Using Convolutional Neural Network," *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Lake Tahoe, NV, 2018, pp. 314-322. doi:10.1109/WACV.2018.00041
- [2] I. F. Rodriguez, R. Megret, R. Egnor, K. Branson, J. L. Agosto, T. Giray and E. Acuña, 2018. Multiple Insect and Animal Tracking in Video using Part Affinity Fields. Visual observation and analysis of Vertebrate And Insect Behavior (VAIB), International Conference on Pattern Recognition (ICPR), Beijing, China, 20-24 August 2018.
- [3] J. Wang, and E. Olson. 2016. AprilTag 2: Efficient and Robust Fiducial Detection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4193-4198.
- [4] I. F. Rodriguez, K. Branson, E. Acuña, R. Mégret, J. L. Agosto-Rivera, T. Giray, "Automatic monitoring of the foraging behavior of tagged and untagged honey bees", *SIDIM 2019*, Humacao, March 1-2 2019.
- [5] E. Latorre, K. López, I. F. Rodríguez, M. Rosner Ortiz, R. Mégret, T. Giray, J. L. Agosto, "Recognition of Fanning Bees from Video using Convolutional Neural Networks", *SIDIM 2019*, Humacao, March 1-2 2019.
- [6] J. Chan, I. F. Rodriguez, R. Mégret, J. L. Agosto-Rivera, T. Giray, "Learning good features to discriminate untagged bees in video using non- supervised learning", *SIDIM 2019*, Humacao, March 1-2 2019.