Tracking individual honeybees among wildflower clusters with computer vision-facilitated pollinator monitoring

Malika Nisal Ratnayake, Adrian G. Dyer, Alan Dorin

Abstract

Monitoring animals in their natural habitat is essential for advancement of animal beha**vioural studies**, especially in pollination studies. **Non-invasive techniques** are preferred for these purposes as they reduce opportunities for research apparatus to interfere with behaviour. The ability to track unmarked insect pollinators in this way will help researchers **better understand pollination ecology**.

We present a **novel hybrid detection and tracking algorithm** to monitor unmarked insects outdoors. Our software can detect an insect, identify when a tracked insect becomes **occluded** from view and when it **reemerges**, determine when an insect **exits** the **camera field of view**, and our software assembles a series of **insect locations into a coherent trajectory**.

Insect detecting: **background subtraction and deep learning-based detection**

It tracked honeybees at a rate of 86.6% on our dataset, 43% higher than the computationally more expensive, standalone deep learning model YOLOv2.

Introduction

Since pollination is an ongoing requirement of crops and wildflowers alike, it would be ideal to establish field stations that can provide ongoing data on pollinator behaviours. To be practical, such a solution would need to be cheap to assemble and install.

Invasive methods for example mark insects with electronic tags such as Passive Integrated Transponders (PIT) [9–12] or tags facilitating image-based tracking [13]. Although, these methods can **track insects over expansive areas** and thus provide important **larger scale information** [14], the spatiotemporal resolution of collected data is lower than that of image-based tracking [5].

But their application is often confined to laboratories offering **constant backgrounds** and **illumination** needed for accurate tracking [17–21] or **require human intervention** [22].

Behavioural research on animals shows that **environmental factors** such as **wind, temperature, humidity, sky exposure, may affect behaviour and interactions** [23,24], and these are exactly the kinds of factors that field monitoring must explore.

Background subtraction is efficient where background and illumination are constant, and significant background/object contrast exists [5]. This method has also been used to count and track honeybees [29–37] and bumblebees [1]

For rare species, or for **species not previously tracked**, a requirement for large training datasets increases the difficulty in implementing a tracking algorithm. Together, these factors currently limit the use of deep learning for generalised animal tracking, and for its application in remote devices for ecological research extracting movement and behavioural data from high-resolution data.

Previous tracking approaches have used convolutional neural networks (CNNs) to **estimate honeybee posture [25], distinguish between pollen-bearing and non-bearing honeybees [40],** monitor interactions of honeybees in a hive [13] and monitor hive entry/exits [41].

Ideally, it is desirable to **detect an insect** and **identify its position** in all of these scenarios to enable accurate census of pollinators, and **what flowers** they visit.

To maintain the identity of the insect and terminate tracking if necessary, it is important for accurate recognition of insects as they move through a complex environment.

In this paper, we present a novel **Hybrid Detection and Tracking (HyDaT)** algorithm to monitor foraging insects outdoors. Hence, our implementation **tracks one insect** at a time from its entry to its exit from view, or from the start of a video sequence to the conclusion. In order to **extract multiple** plant-pollinator interaction sequences (actually, sequences of interactions between a unique pollinator and a set of flowers) we **re-run the software on each insect** detected in a region/clip in turn. We compare the efficiency and effectiveness of our algorithm against human ground observations and previously described methods, and apply our approach to track foraging on flower carpets in a new dataset (78 minutes of outdoor video).

Materials and methods

Our Hybrid Detection and Tracking (HyDaT) algorithm has four main components (Fig 2): A hybrid detection algorithm begins at the start of the video and moves through the footage until it first detects and identifies an as yet untracked insect.

If this insect is not detected within a subset of subsequent frames, the algorithm uses novel methods to predict if it is occluded or has exited the view. Positional data collected from the algorithm is then linked to synthesise coherent insect trajectories. Finally, this information is **analysed to obtain movement and behavioural data** (e.g. heat-maps, speed or turn-angle distributions).

The hybrid detection algorithm

We use a hybrid algorithm consisting of background subtraction and deep learning-based detection to locate an insect.

As discussed in the introduction, **background subtraction** can **detect movements in the foreground without prior training** and works efficiently where the **background is mainly stationary**. In contrast, **deep learning**-based detection can **detect** and **identify** an insect irrespective of changes in the background, but it requires training with a dataset prior to use.

1. The algorithm begins using the **trained deep learning model** to **initialise the detection** process **by locating the insect's first appearance** in a video. This ensures identification of an insect with a low probability of false positives, even if the background is moving.
2. After initial identification, the technique used for insect detection is determined by the number of regions of inter-frame change within a calculated radius MDT_{DL} of the predicted position of the insect in the next frame (Data association and tracking, Eq 4)
    a. If there is a **single** region of significant change identified between frames, the **background subtraction** technique is used to locate the insect.
    b. If a **small number** of regions of change are detected within the predicted radius of the insect, then the **region closest to the predicted position is recorded as the insect's position.**
    c. However, sometimes the region within the radius around the insect's predicted position is **too full** of movement to be sure which is the insect. In this case, background subtraction is unusable, or perhaps insufficiently inaccurate, so the hybrid algorithm switches to **deep learning**.
    d. In **addition**, whenever the **background** subtraction technique **fails** to detect movement likely to indicate the insect's position, **deep leaning** is used.

- **Deep learning-based detection**: convolutional neural network (CNN)-based YOLO (You Only Look Once)

- **Background subtraction-based detection**: We use K-nearest neighbour (KNN)-based background/foreground segmentation [48] (OpenCV 3.4.1 [50]) to detect foreground changes in the video.
    - The KNN background subtractor works by updating parameters of a Gaussian mixture model for better kernel density estimation
    - The resulting binary image includes changes of the foreground assuming a constant background. A **median filter** and an **erosion-based morphological** filter are applied to the segmented image to remove noise.
    - The **resulting image contains changes in the foreground** caused by insects and moving objects.
    - Next, contours of the foreground detections (blobs) are extracted from the binary image and filtered based on their enclosing area to remove areas of movement less than a predetermined minimum pixel count covered by the focal insect. The position of the insect is designated by the centroid of this filtered blob (Fig 3).

Identifying occlusions

In the event that the focal insect is undetected, our algorithm analyses the variation in insect body area before its disappearance to identify a possible occlusion. **Background subtraction is used** to measure this change from the video.

Identifying an insect exiting the field of view

If an insect is re-detected near the point of disappearance of the original focal insect before a new insect appears, the algorithm resumes tracking it, assuming this to be the original focal insect

Data association and tracking

For applications discussed above, our algorithm **tracks one insect** at a time from its first appearance until its exit from view, before it is **re-applied to track subsequent insects** in footage.

In a set of **three successive frames**, the predicted insect position in the **third is calculated from the detected positions** in the first two frames, assuming constant insect velocity over the three frames [35,52].

When an insect is **first detected**, the **predicted position for the next frame is assumed to be the same** as its current position (as there are no preceding frames). In the case of **occlusions** or frames in which **no insect is detected, the predicted position is carried forward** until the insect is re-detected.

Assign the predicted position of the focal insect to an individual detection within the frame. This is done using a process derived from the Hungarian method [53] which minimises the distance between assigned detections and predictions.

Experiments and results

We selected a patch of Scaevola (Scaevola hookeri) groundcover as the experimental site to evaluate our methods because of the species' tendency to grow in two dimensional carpets and to flower in high floral densities.

A Samsung Galaxy S8 phone camera (12 MP CMOS sensor, f/1.7, 1920 × 1080 pix, **60 fps**) mounted on a tripod was set 600 mm above the groundcover to record videos (Fig 5). A ruler placed in the recorded video frame was later used to convert pixel values to spatial scale (millimetres). Recorded videos covered an area of 600 mm ×332 mm with a density of 10.24 pixels/mm−2. Average area covered by a honeybee was 1465 ± 531 pixels (e.g. see Fig 3a).

YOLOv2 object detection model [54] was used as the deep learning-based detection model in HyDaT

Detection rate is our measure to evaluate the number of frames where the position of the insect is accurately recorded with respect to human observations. If the algorithm recorded the position of the honeybee in an area that was in fact covered by the body of the bee, this was considered as a **successful detection**. The time taken by the algorithm to process the video was recorded as the **tracking time**.

We also compared the detection rate and tracking time of HyDaT to the stand-alone deep learning-based YOLOv2 [49] model after using the same training dataset for each.

To benchmark our results further, we also processed the seven honeybee videos using Ctrax [18], current state-of-the-art insect tracking software.

HyDaT processed the seven videos totalling 6 minutes 11 seconds (22260 frames at 60 fps) of footage in 3:39:16 hours, a reduction in tracking time of 52% over YOLOv2. This improvement in speed is possible because 91% of detections by HyDaT were made with background subtraction which requires much lower computational resources than purely deep learning based models.

turn-angle distributions for Lamb's-ear since a single camera setup cannot accurately measure these attributes for three-dimensional motion, a limitation we discuss below.

Benötigen wir eine Echtzeit Analyse?

Videos sinnvoll? Dann gleiche Kameraausrichtung.

Später stationäre Aufnahme? Dann solche Videos und solche Pflanzen, hier schon Probleme beim Wechseln der Pflanzenart.