# LocusTracker: Markerless Multi-Agent Tracking In Various Environments

Roee M. Francos[1†], Moshe Guershon[2,3], Amir Ayali[2,4], Tali Hatuka[5] and Alfred M. Bruckstein[1]

[1]Computer Science Faculty, Technion- Israel Institute of Technology, Haifa, Israel
(Tel:+972-4-8294361; E-mail: roee.francos@cs.technion.ac.il, freddy@cs.technion.ac.il)
[2]School of Zoology, Tel Aviv University, Tel Aviv, Israel
(Tel: +972-3-640-9820; E-mail: mois@tauex.tau.ac.il, ayali@tauex.tau.ac.il)
[3]The Steinhardt Museum of Natural History, Tel Aviv University, Tel Aviv, Israel
[4]Sagol School of Neuroscience, Tel Aviv University, Tel Aviv, Israel
[5]Porter School of the Environment and Earth Sciences, Tel Aviv University, Tel Aviv, Israel
(Tel: +972-3-6405162; E-mail: hatuka@tauex.tau.ac.il)

**Abstract:** In biological behavioral experiments precise location and tracking of living organisms is of great importance. In particular, reliable and accurate video/vision based detection and tracking of mobile agents (animals) in a swarm can provide valuable information on movement responses, on interaction between individuals, and the influence of the environment's geometry and other factors on the emergence of collective behaviour. In this work we present a computer-vision software tool called the LocusTracker, developed for research on the swarming behaviours of locusts. The tool enables tracking individual locusts in a group, and was deployed in various environments for several hours, without disturbances that would be inherent when using physical markers on the moving insects. The tool is an advanced computer-vision-based multiple blob detection and tracking system that can serve as a useful platform for research on biological and robotic multi-agent collective behaviours, and hence is released as an open-source software tool for the scientific community.

**Keywords:** Biological Swarm Tracking and Analysis, Multi-Agent Trajectory Tracking, Video-based Detection and Tracking, Bio-Inspired Robotics Tracking, Locust Trajectory Analysis

## 1. INTRODUCTION

Insect tracking is a critical step in understanding and modelling motions of natural swarms. In this work a new computer-vision software tool, called LocusTracker is used for locust trajectory prediction. The tool is based on the principles of tracking by detection and association and enables tracking of individual locusts in various laboratory environments and can be used for tracking and analysis of other insect species and bio-inspired robots as well.

The focus of this tracker is to track locusts in dense and cluttered environments that simulates the close proximity of locusts in nature. We relied on physical models of urban environments developed in [1] as examples of challenging and intricate physical environments that admit locusts to cluster in very close proximity to one another.

Since the locusts are small and visually indistinguishable, even state-of-the-art deep learning detectors and trackers do not provide sufficiently accurate results. Hence, the challenge in developing this tracker was in tracking similarly appearing locusts for very long time periods of several hours without the usage of physical markers on individual locusts.

Using physical markers, such as QR codes, to accurately track insects in controlled laboratory environments is a standard approach in the biological community, however this method requires usage of very high resolution cameras that observe a relatively small environment, in order to have sufficient pixel resolution to read and decipher the markers that are attached to locusts. Furthermore, since locusts tend to form dense clusters and sometimes even climb on top of each other, these markers can become occluded, leading to the inability to identify and track insects. Additionally, usage of such markers does not allow tracking of locusts and other animals in the wild, while the developed LocusTracker software does not have these limitations and hence can be used in other environments to track different animals that perform collective motions as well.

LocusTracker is used to detect locusts in 2 dimensional environments including free-space environments of various geometries and shapes as well as in environments which imitate models of cities, while the locusts are still on the ground. However, the provided software can be used to track and analyze motions locusts and other animals in 3 dimensional space as well. Fig. 1 displays a typical locust used in the experiment, one of the environments in which locusts are tracked and a resulting trajectory of one of the locusts.

By analyzing the resulting trajectories which are generated without the placement of any physical markers on the locusts we hope to gain a deeper understanding of their behavior, based on the characteristics of the environment in which the locusts are placed. We publish the LocusTracker code as an open-source framework [1] for the benefit of the community. We present the design criteria the locust detection and tracking algorithm is based upon.

† Roee M. Francos is the presenter of this paper.

[1]https://github.com/RoeeFrancos1990/LocusTracker.git

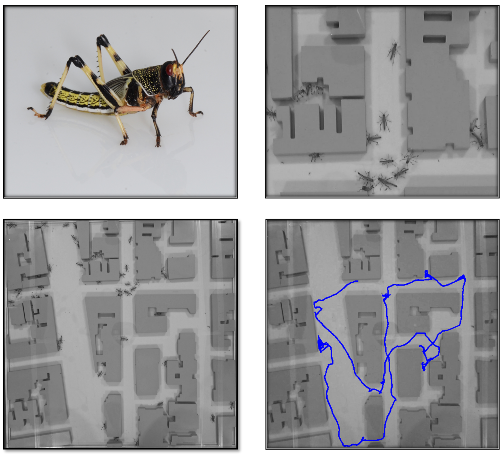

Fig. 1 Top left- typical locust used in the experiment, top right- zoom in on locusts inside a section of the maze, bottom left- full view of all locusts in one of the maze environments, bottom right- an example of a locust trajectory in the environment.

## 2. OVERVIEW AND COMPARISON TO RELATED RESEARCH

The problem of analysis of collective motion of locusts/ants and miniature robot-swarms in various environments requires several capabilities: location of the individual mobile agents, tracking and recording their trajectories, resolving collisions and occlusions, and reverse engineering the local inter-agent interactions.

Previous approaches mainly use markers affixed to the agents for location identification and tracking in conjunction with video capture based methods involving detection and tracking of markers. Marker based methods are used since they offer an easy solution for re-identification following occlusions and aggregations.

Their drawbacks are that they inherently cause disturbances to the observed animals. Additionally, markers can become occluded in case the animals are in close proximity to one another, or when they climb on each other as is often seen in the locust experiments we conducted. Furthermore, cameras with very high resolution are required in order to decipher the markers readings.

The solution we propose uses video based markerless methods that employ standard cameras along with sophisticated computer-vision detection, tracking and re-identification methods. This is the approach we describe below.

There were markerless systems deployed before, however they suffer from several drawbacks that led to the majority of the studies to be conducted using systems that rely on markers. Drawbacks of previous markerless systems included problems with aggregation and poor long-time tracking performance, and none of these systems met all our design goals which included simultaneously tracking a large number of insects for very long time horizons of several hours along with a deployment in complex obstacle filled environments with significant animal encounters, aggregations and occlusions.

The technical problems overcome by our method were perfectly identifying moving locusts using motion-based difference images, accurate background subtraction that enabled detection of locusts that manifest themselves in a very small number of pixels, and development of complex logic that resolves occlusions and trajectory association conflicts when locusts aggregate.

We describe below the works closely related to our LocusTracker. These works may be categorized as being marker-based or markerless methods, single or multiple multi-agent trackers, classical or deep-learning based methods, simple vs. complex environments and short vs. long time horizon tracking.

During the last decades there have been attempts to automate, measure and quantify experimental data on emergent behaviour in several types of insect swarms. In conjunction with these efforts in data gathering, the interdisciplinary multi-agent community developed various local interaction models drawing inspiration from biological swarms.

In [2], a pursuit model that aims to explain why ant trails starting from an anthill to a food source are straight, even though ants do not have any sense of geometry was developed. In [3], an early simple model was introduced to investigate the emergence of coherent motion in systems of particles with biologically motivated interactions. Several early reviews investigated observation-based collective motion, emphasizing the basic laws that may underly various factors influencing collective motion [4]. Others considered design and analysis methods in swarm robotics that are inspired by collective motion from an engineering perspective [5].

Towards developing an understanding of insect-swarm emergent behaviour and dynamics in laboratory conditions, studies such as [6] were performed, offering quantitative data that can be used as a benchmark for comparing the characteristics of animal aggregation models. Several recent studies focused specifically on studying and modelling of locust collective motion. The study in [7] reviews advances in locust collective motion and its modeling from both biological and statistical physics perspectives towards the goal of analyzing and predicting swarm dynamics of natural locust swarms as well as behaviours of locust swarms in laboratory experiments. In [8], collective motion and walking kinematics of locusts are inferred based on the characterization and adaptations of the behavioral state of the individual locust before, during, and after a specimen is a part of a swarm. Authors found that participation in collective motion induced in the individual locust unique behavioral kinematics, imply the existence of a distinct behavioral mode referred to as a “collective-motion-state” that is long lasting and occurs only for locusts that participate in collective motion experiments.

In [1], integration of urban design and animal science has led to the development of methods for examining the connections between urban layout and physical movement. This approach involves the creation of a structured process to assess mobility efficiency across different urban environments by leveraging experimental data on dynamic behavior of living organisms, such as locusts, within miniature city models. By utilizing these natural-biological agents as indicators of flow, the research offers valuable insights into the complex interplay of flows within urban landscapes. Within the context of this work, LocusTracker was developed.

Several of these works studied interesting problems concerned with developing proven mathematical models of locust interactions in laboratory conditions as they were observed through visual recordings of locust swarms. In [9], a simple model investigating collective marching of locusts in a ring environment was developed with the goal of explaining emergence of collective clockwise or counterclockwise movements of the locust swarm. In [10], a locust-inspired random pairwise interaction model is developed, proving that local interactions that happen at encounters can account for interesting emergent global phenomena.

In [11], a vision-based system for social insect tracking is presented. The method uses markers to track the location of a honeybee queen inside a bee colony. The markers are attached only to the queen bee allowing the researchers to reason about queen and worker bee interactions. The focus is on tracking a single queen bee for long periods of time.

The authors use a special marker called WhyCode [12] that allows them to use a camera with slightly lower resolution compared to usage of Aruco [13] markers. Using this marker type allows to track markers when they occupy 25 pixels or more, compared to 50 using Aruco. As stated earlier, marker based tracking of interacting insects results in marker occlusions and detection loss at meetings. The authors indeed experience this and propose a method to fix the detection drift once the marker is detected again.

As in our setting, honeybee colonies present challenges for detection and tracking due to the high density of specimens that constantly move and occlude each other. A major difference between the previous work and LocusTracker is that we aim to simultaneously track large numbers of similarly appearing insects rather than tracking a single more important insect that is indistinguishable from the rest of the swarm.

In [14], an early work concerning automatic markerless tracking and analysis of live insect colonies by usage of computer vision algorithms is developed. With a similar goal as in LocusTracker, this work aims to track hundreds of small insects simultaneously and afterwards analyze ensuing colony behaviours. Tracking of ants is performed using combined color-based and movement-based tracking. Before this work was published, insect paths were mainly recorded manually through careful observation by specialists. Although the goal of this work was to track hundreds of animals, only experiments with several ants in free space recorded for several minutes are reported.

In [15], individual pause-and-go motion is shown to be instrumental in the formation and maintenance of swarms of marching locust nymphs. The principal interactions leading to the emergence of order in the observed swarms are studied both experimentally, using small groups of marching locusts in the lab, and through computer simulations. A custom-built markerless multiple locusts tracking system is developed to analyze motions of locust nymphs in a homogenous circular arena. The tracking system developed consists of three main components: foreground-background analysis, connectivity for cohesive region creation, and trajectory association using Voronoi partitioning.

While both our tracker and the one developed in [15] share the common objective of tracking locusts, there are several notable differences between the two methods. In LocusTracker we employ a background subtraction method tailored to the more complex environment we use in our experiments. Additionally, the trajectory association module uses a different set of rules, and we have incorporated a unique locust cluster analysis module into LocusTracker.

In [16], a multiple object tracking system is developed for insect behavior research, enabling long-term tracking without usage of invasive markers. The method identifies pixels belonging to roaches by relying on different color statistics between roaches and the background. Similar to our approach, the system uses prior knowledge on the number of tracked insects to maintain long trajectories. In case tracked insects are close to each other, pixels belonging to the bounding boxes that surround them are discarded so that each bounding box contains a single insect, which is different than the cluster analysis we perform in LocusTracker. An additional limitation of [16], is that it assumes distinct insect centroids, which may not hold for swarming animals. Furthermore, the arena analyzed is simplistic and does not contain obstacles or occlusions and there is no special logic for tracking large numbers of insects as in LocusTracker. In [17], an additional markerless multiple animal tracker is proposed. This software suite enables the user to choose between several visual trackers, provides an easy to use graphical user interface as well as a manual tracking error correction system.

Markerless tracking of an entire colony of honeybees is investigated in [18]. The output trajectories provide data for quantitative studies of collective bee behaviour such as comb-cell activities and waggle dances. As in our setting, authors aim to track hundreds of similarly looking honeybees in dense environments and achieve this by training a deep convolutional network that performs pixel level foreground-background segmentation.

Interestingly, the authors develop a training mechanism that enables them to utilize visual features of bees that appear identical to the human eye to improve tracking performance. The aim is to train a deep network to generate a distinguishable visual signature for each bee that manifests in a different feature vector embedding for different bees. This embedding is used as an additional input to the logical matching process between detections of bees in subsequent frames along with position and motion-based matching.

In [19], a computer-vision based markerless outdoor insect counting and tracking system that performs behaviour analysis for pollination purposes is developed. The goal of this work is automatically gather information on insect distributions to predict pollination efficacy and assist in precision pollination. The authors focus on classifying various species of insects and on gathering information across large spatial extents.

Similarly to our setting, the authors argue that while tagging insects with markers assists in tracking them across large areas, placing such markers is cumbersome, can affect insect behaviour and is not practical for gathering information across large areas [20].

However, the major challenge [20] focuses on is tracking insects across varying illumination effects that change foreground and background statistics. Since in our case we were primarily interested in tracking identical animals for long time horizons with simple cameras we relax the requirement on being adaptive to varying illumination conditions. Despite the remarkable outdoor performance, the authors did not need to handle crowded and congested locations where many insects are located in close proximity to one another as with LocusTracker.

In [21], BugTracker, an open-source software suite for tracking and measuring arthropod activity is developed. Like our tracker, BugTracker is a markerless tracker that aims to be less sensitive to illumination changes in the environment where arthropods are being tracked. BugTracker relies on the well known CSRT tracker [22] (discriminative correlation filter with channel and spatial reliability). The authors examined several computer vision technologies suitable for tracking of insects in laboratory environments.

Although related to our work, BugTracker solves a simpler problem than LocusTracker since it tracks only a single insect in a simple environment that does not contain obstacles, and hence does not need to handle occlusions, shadows from insects near obstacles and aggregation of insects whose individual trajectories need to be resolved using conflict resolution algorithms and cluster analysis methods. Furthermore, results reported considered tracking of an insect for few tens of seconds while in LocusTracker we can track dozens of insects for hours. Additionally, relying on the CSRT tracker will not assist in tracking identically looking insects like locusts and will result in identity switches.

In [23], a markerless multitracking algorithm that extracts a characterizing intensity and distance based descriptor from each zebra fish that is observed in an experiment is developed. The obtained descriptors are then used to identify the individual fish and track their motions while attempting to preserve their identity. In [24], a more advanced version of the tracking algorithm proposed in [23] is proposed. Instead of using intensity and distance based descriptors to identify tracked animals, the authors train a classification network that predicts the identity of a detected animal, and later use these descriptors as a signature in order to identify the same animals the deep network was trained upon, thus assisting with managing ID conflicts in the assignment stage and the trajectory generation stage. While these approaches provide animal re-identification capabilities, it is not straightforward to apply them in the scenario we discuss due to the similar appearance of tracked locusts, their proximity, and the amount of locust interactions that causes large clusters of locusts to form.

## 3. CONTRIBUTIONS

The proposed LocusTracker software tool offers several practical contributions and advantages over existing biological and robotic multi-agent tracking methods and is particularly suited for tracking agents in dense environments in which agents are in very close proximity to one another, and exhibit collective and swarming behaviours.

- The tool enables tracking of individual locusts in complicated and obstacle strewn environments for very long time horizons of several hours without the usage of physical markers on the agents. This implies that agents of all sizes can be tracked in large environments without requiring strict resolution constraints and without suffering from misdetections due to marker occlusions.
- The tool is designed to address tracking of identically looking biological agents exhibiting collective behaviour in laboratory environments.
- The tool allows to fully determine and control the outputs of the tracker that can later be used for careful statistical analysis of the parameters extracted. The constraints that influence the tracking algorithm can also be controlled and new functionalities can be added based on the particular needs of the user.
- One click open source Python software. The user only needs to click on the corners of the environment in which agents are to be tracked. This alleviates constraints of carefully positioning the camera over the area of interest.

## 4. ALGORITHMIC OVERVIEW OF VISUAL TRACKING OF LOCUSTS IN LABORATORY ENVIRONMENTS

In this section we present the main algorithmic components of LocusTracker and its operation principles. LocusTracker consists of six main modules. The first is a background subtraction module that is used to detect the locations of locusts based on the difference between the current frame and a reference image of an empty environment that does not contain locusts. This process makes LocusTracker less sensitive to illumination changes that occur in different environments and under different lighting conditions.

The second module, the blob detection module, aims to detect and identify locusts or cluster of locusts. The input to this module is the foreground image generated by the background subtraction module. Locusts are detected based on various properties locusts exhibit such as their shape and size.

The third module is a locust trajectory association module. This module gets as input a set of candidate points that represent the center of mass of each detected locust in the current frame as well as information on the locations of locusts that were detected in previous frames.

The goal of this model is to determine how to associate the newly detected potential locations of locusts to the previous trajectories of other locusts. Since the locusts are visually indistinguishable from each other, the matching between candidate points and trajectories is based on a set of logical constraints that were inferred from observations on the movement patterns of locusts in various environment models recorded in the Ayali lab at Tel Aviv university. These constraints will be elaborated upon in the next sections.

The fourth module is responsible for locust cluster analysis and aims at identifying whether detected objects are part of a locust cluster and if so, how many are in the detected cluster. Furthermore, this module is also responsible to manage IDs of locusts that form, enter and exit existing clusters in order to continue their individual tracking once they leave the cluster. When locusts are part of a locust cluster, all locusts in the cluster are formally assumed to be located at the same point at the cluster's center.

The fifth module's task is to extract meaningful parameters for the statistical analysis and the locomotion evaluation subsequently performed. This information includes many parameters such as locations of locusts, speed, angle of movement, whether they are stationary or moving, if they are part of a cluster, close to a detected corner in the environment and more. A detailed description on the extracted parameters and how they are calculated is provided in a dedicated section.

The sixth module's goal is to visualize the extracted results in a number of ways to better understand the movements and behaviours of locusts in the environment. Among the outputs of this module are a heat map showing the magnitude of the concentration of locusts in different parts of the environment as a function of time, visualizations of the trajectories of individual locusts during the experiment as well a video file showing the movements of locusts across time.

Fig. 2 shows a block diagram description of the algorithmic components of LocusTracker. Fig. 3 shows tracking of locusts in a particular snapshot of time at several analyzed city models.

### 4.1. Background Subtraction Module

The purpose of this module is obtain a difference image between the current frame and a frame of an empty city model prior to the introduction of locusts into it. The goal is to detect potential areas in which locusts are present in order to later track them. Since we are using a controlled laboratory environment we can guarantee that the light properties of the scene remain almost fixed throughout the experiment and therefore we can use an image of the empty environment to detect changes in the

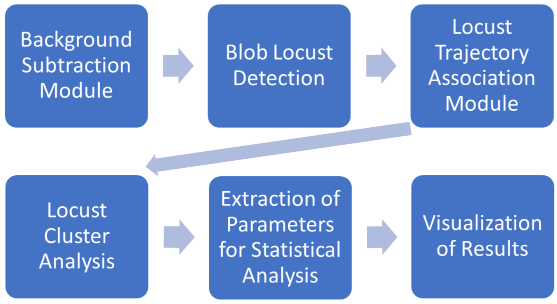


Fig. 2 Block diagram of the algorithmic components used in LocusTracker.

images that occur only due to the presence of locusts.

It is worth noting that to ensure that locusts do not hop outside of the arena, it is covered by an almost transparent fiberglass cover. Since the cover must be as transparent as possible in order to allow the obtained recorded image to be of good quality, we must also handle reflections arising from the light sources in the environment. Using the background subtraction model allows us to control the lightning in the arena, attempt to make all parts of it evenly lit and still be able to remove reflections from the recorded images.

The left image of Fig. 4 presents an environment model before locusts are introduced into it. The middle image displays a typical frame in the analyzed videos showing locusts inside the arena. The right image shows the absolute difference image obtained between the middle image and the left image, mainly highlighting pixels in which locusts are located.

### 4.2. Blob Locust Detection Module

The second module, the blob locust detection module, aims to detect individual locusts or locust clusters from the foreground image obtained from the background subtraction module. Since all tracked locusts are about the same size, we can utilize this knowledge to determine the number of locusts that belong to a detected locust blob.

In this module several filters are applied to the image in order to obtain only blobs that have the typical size of locusts we are looking for. These filters allow to further remove reflections and other sources of noise from the image. Each detected locust blob is represented using its center of mass which is computed after the blob's contour is extracted. The calculated center of mass serves as the point associated with the locust in the current frame.

In case the area of a detected blob is larger than the typical area of a locust (this occurs when locusts are very close to each other or one on top of the other), we can determine the number of locusts in a cluster based on the blob's area. Since we are operating in a multi-agent paradigm at which all agents are identical, all of them have the same size. This is also true in the biological scenario investigated since all the locusts have exactly the same characteristics in terms of size.

The output of this module (in every frame) is a list of the locusts blob's centers, the number of locusts in each

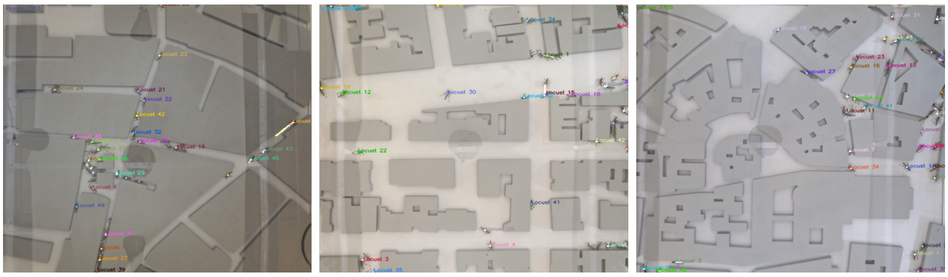

Fig. 3 Each picture shows the tracking of locusts in a particular snapshot of time. The left image shows locusts being tracked in the Cairo city model, the center image shows locusts tracked in the New York city model, the left image shows locusts tracked in the Rome city model.

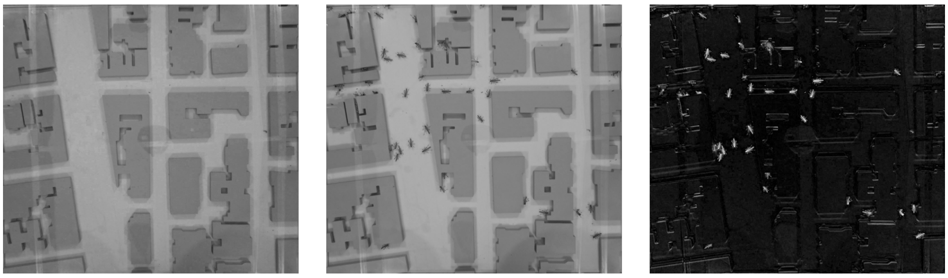

Fig. 4 Background Subtraction Module. The left image displays the empty arena before locusts are introduced into it. The middle image shows locusts inside the arena. The right image shows the resulting thresholded absolute difference image, where high intensity pixels mainly indicate locations of locusts in the current frame.

blob, and whether a detected blob is a cluster containing more than a single locust. Additionally, we use the minimal Euclidean distance between detected locusts, a distance corresponding to a locust size, as a filtering measure to ensure that only a single candidate point is detected for each locust. Fig. 5 displays the detected blob centers as white circles.

Based on the distance of the camera and the arena the average locust blob has a radius of 21 pixels. If animals having different sizes are being observed, or the arena is photographed at other distances than the average size of the observed animal needs to be assessed and provided to the model in order for it to accurately assess locations of observed individuals and density of entities inside each detected cluster.

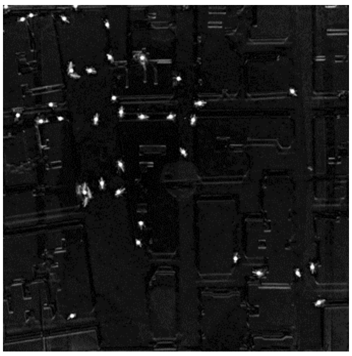

Fig. 5 An obtained difference image with white circles indicating detected locusts or locust clusters.

### 4.3. Locust Trajectory Association Module

The goal of this module is to associate detected points in the current frame to previously detected points in order to form long locust trajectories. Many experiments and trials on the required logical conditions were performed in order to determine the most suitable constraints necessary to match locust candidate locations detected in previous frames to candidate locations detected in the current frame.

Among the logical conditions used in the association module are: number of detected locusts, locust orientation, distance marched, a locust's maximal speed (including its possibility to hop), the time (frame) at which locusts are detected, whether a candidate locust point belongs to a cluster, the maximal number of locusts in the environment, the current and previous locations of detect locusts, and the maximal time a locust can remain stationary. Throughout the detection and locust trajectory generation process there are 3 possible cases:

1. The number of existing locust trajectories is smaller than the number of detected locusts in the current frame.
2. The number of existing locust trajectories is equal to the number of detected locusts in the current frame.
3. The number of existing locust trajectories is greater than the number of detected locusts in the current frame.

After each locust blob's center is determined we must match the newly detected locust locations to the previous trajectories. The requirement is to match a single candidate point to a single trajectory. At first, we sort the distances from each detected point in the current frame to the last point in all previous locust trajectories. We choose the trajectory that has the closest point in any of the previous trajectories to the candidate point as the potential trajectory we wish to add the new candidate point into. In case the distance between the candidate point to the previous latest point in the chosen locust trajectory is smaller than the maximal distance a locust can travel, and the time that passed since this last point was added to the trajectory list is below the time limit a locust can remain stationary we add the new candidate point to the locust trajectory.

If a detected locust point is at a greater distance than allowed (based on the maximal possible locust speed and the time that passed since the last time this locust was detected), a new trajectory is formed starting from the current locust point. This logic is useful since sometimes locusts tend to cluster together and therefore we cannot always start the tracking of locusts with an individual trajectory for every locust. Only after locusts move away from each other throughout the experiment we can assign them an identity that is separated from the cluster they belonged to and start aggregating their trajectory.

Once a locust departs from a cluster, the location of the center point of the cluster is used as the particular locust's position at the time intervals it belonged to the cluster. Therefore, each trajectory of a certain locust is composed of time intervals at which it belonged to a cluster and time intervals at which it is considered as an individual locust. Once a locust point is matched to a certain trajectory it is removed from the list of current points and the next candidate points are matched to the remaining trajectories that locusts were not matched to in the current frame.

In case a locust leaves a locust cluster and the maximal number of locusts were already detected, its location is added to the trajectory of the locust whose last trajectory point is closest to the newly detected point. This logic is used in order to allow the generated locust trajectories to be long and extend to the duration of the recorded video. Therefore, we use prior knowledge on the number of locusts in the environment. This procedure is continued until all new detected locusts locations are added to the previous locust trajectories or start forming new trajectories. In order to provide the association module a parameter that filters motions of locusts based on their heading angle, a maximal turning angle of 3 degrees is used to connect current and past locations of detected locusts.

### 4.4. Locust Cluster Analysis

The goal of this module is to determine whether a detected locust blob contains more than a single locust based on the typical size of the observed locusts. If this is indeed the case, then the number of locusts in the cluster is determined based on its area. Furthermore, this module is also responsible to manage IDs of locusts that form, enter and exit existing clusters to continue their individual tracking once they leave the cluster. When locusts are part of a cluster, all locusts in the cluster are assumed to be located at the same point at the center of the cluster. In order to manage the IDs of the locusts inside each cluster we use several logical conditions. This is performed in order to accommodate several possible scenarios:

1. Forming of a new cluster- this occurs when two or more locusts that were previously apart from each other move to be in close proximity to one another.
2. Exiting of a locust from a cluster- this occurs when a certain locusts moves away from a cluster it was previously a part of.
3. Merging of clusters- this happens when two clusters become closer to one another and form a larger merged cluster.
4. Separation of clusters- this happens when a larger cluster breaks up into two separate smaller clusters.
5. Breaking of a cluster- this occurs when the locusts that comprise a cluster move away from each other and separate into individual entities.

In the locust cluster analysis module, based on the number of detected locusts inside each cluster, we add the location of the cluster's center to each of the locusts trajectories that comprise it. It is possible that the IDs of locusts within a cluster may be shuffled with those of other locusts in the same cluster. However, this is not a limitation in our study.

### 4.5. Extraction of Parameters for Statistical Analysis

Several parameters are extracted during the analysis to later be used for statistical analysis. We list several of those parameters that require explanation. The full list of extracted parameters is provided in the section detailing the content of the output files generated by LocusTracker.

#### 4.5.1. Movement Tracking

Based on the centroid of each detected blob we can determine the movement of each locust. Locusts' speed is calculated in units of (cm/sec). If the maximum movement threshold is exceeded, then the prediction for the candidate point corresponding to a detected locust is not associated to it. If the prediction exceeds the threshold for all locusts, the detected point is discarded. The chosen value to discard a candidate point for associating it to a particular trajectory in the experiments is a movement of more than $5$ cm/sec.

Additionally, to classify whether a locust is moving or stationary in every frame, we use a predetermined movement threshold in units of (cm/sec). If exceeded then the locust is considered to be moving. The chosen value in the conducted experiments is $0.25$ cm/sec. We can calculate the mentioned parameters in metric units since we know the size of the environment and hence can convert pixel measurements to real-world units.

#### 4.5.2. Speed Calculation

Locusts' speeds are calculated in each frame they are detected by the Euclidean distance between the current point a locust is located at and its previous position. Since locusts are not guaranteed to be detected in every frame, we divide the Euclidean distance by the time that passed between two detections. In order to get the time between two detections we use the frame numbers at which locust were detected and the camera recording speed which is 30 frames per second (FPS).

## 5. POSSIBLE ADD-ON FEATURES

Since the entire software suite is open source, the community can easily add further modules to the tracking software, depending on the desired analysis. In this section we list several such add on analysis modules we added to LocusTracker.

5.0.1. Corner Detection Module

Corner detection is a classic computer vision algorithm that facilitates extraction of points of interest within an image. Within our framework, our objective is to identify corners within the arena to assess whether the presence of a locust near a corner influences its decision-making process regarding movement. Corner detection operates on the principle of identifying points where two dominant edge directions exist within a localized area. A point qualifies as a corner if both Eigen values of its second moment matrix surpass a predetermined threshold. By employing the Harris corner detector and adjusting the selected threshold, we can effectively pinpoint the locations of corners within each urban model.

Corner information is extracted from the reference image prior to introducing locusts to the environment. A distance threshold measured from the center of the blob representing the locust is used to determine if a locust is near a corner. The chosen value in the experiments to be considered near a corner is a distance less than 1 cm.

5.0.2. Active Area Calculation

The goal of this module is to compute the percentage of the environment that is walkable for the locusts, i.e., the parts of the environment that do not contain obstacles and are free space.

5.0.3. Detection of Marching Intervals

We aim to examine an additional parameter concerning the marching intervals of locusts. Locusts are deemed to be marching together if, on average, they are in close proximity to each other and if throughout this time they are mostly moving. To meet this criterion, locusts must remain in proximity for a duration exceeding 5 seconds, with proximity defined as a distance of less than 5 cm apart. It is important to note that the marching together condition distinguishes between locusts marching together and those forming a cluster, thereby separating the analysis of marching intervals from cluster analysis.

### 5.1. Visualization of Results and Saved Parameters

The extracted results are presented through various visualizations to facilitate a deeper comprehension of the movements and behaviors of locusts within the environment. These visual outputs comprise:

- A heatmap depicting the intensity of locust concentration across distinct areas of the environment over time.
- Visual representations illustrating the trajectories followed by individual locusts throughout the duration of the experiment.
- A video showing the temporal evolution of locust movements throughout the experiment.

Fig. 6 depicts a visualization of the results from two of the conducted experiments. Each one of the trajectory images at the top of Fig. 6 represents a chosen trajectory of a single locust out of the 50 locusts that participated in each experiment. The bottom images show the aggregated heat map of all locusts that participated in the corresponding experiment as the image above it, indicating locations in the environment where locusts tend to aggregate.

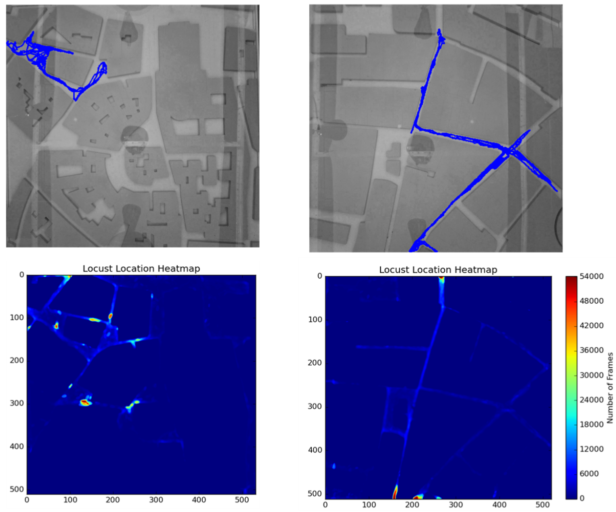


Fig. 6 Top- example trajectory of a locust in the (left) Rome (right) Cairo maze environment, Bottom- accumulated heatmap of locust locations in a half hour experiment inside the (left) Rome (right) Cairo maze environment. Bright (redder) locations indicate locations where locusts tend to aggregate and darker (bluer) areas indicate locations where locusts are less commonly found.

## 6. LOCUSTRACKER TECHNICAL DETAILS, PARAMETERS AND OUTPUTS

The output from the LocusTracker software is presented in tabular format, and comprises the computed parameters essential for the statistical analysis. Additionally, various visual outputs aid in the evaluation of the obtained results. This section provides details on the extracted parameters utilized in the statistical analysis, accompanied by explanations regarding the methodologies employed for their computation.

6.0.1. Operation of LocusTracker

By using the physical dimensions of the arena which are $120 \times 120$ cm, we are able to convert the pixel-based measurements obtained from the videos to real-world metric quantities. Hence, to use the provided software in other settings, the only information that needs to be provided to the software are the physical dimensions of the model. In order to allow more flexibility in the footage step of recording the locusts movements, before the analysis starts the user is asked to click on the 4 corners of the observed environment. This step removes from the user the requirement to perfectly align the field of view of the camera with the arena, prior to the recording of the video.

6.0.2. Output Files Format

The movement files for each locust are stored within a designated folder, with each Excel file corresponding to the movement data of an individual locust. These files encompass information including:

- Frame number: indicates the frame in which the locust was detected.
- Locust ID number: a unique identifier assigned to each locust.
- Coordinates: the position of the center of the locust's blob in that frame.
- Heading angle: the direction in which the locust is moving, based on a coordinate system centered at the locust's blob.
- Movement flag: indicates whether the locust is considered to be moving in the current frame.
- Speed: the speed of the locust in cm/sec.
- Cluster flag: a Boolean indicating if the locust is part of a locust cluster in the current frame.
- Number of locusts: specifies the number of locusts corresponding to the observed locust in the current frame. If this number is 1 than the locust is not a part of a locust cluster; if it is more than 1 than the number of locusts in the cluster is calculated based on the ratio between the cluster's area and the typical area of a locust.
- Corner flag: indicates if the locust is near a corner in the current frame.
- Area type: the type of area in which the locust is currently located.
- Distance advanced: the distance the locust has moved up to the current frame.
- Stationary intervals: time intervals during which the locust is stationary.
- Moving intervals: time intervals during which the locust is moving.
- Marching partners: IDs of other locusts that the locust is marching with in the current frame.

Furthermore, an aggregated Excel file is generated, consolidating the information from all locusts. This integrated file simplifies the subsequent analysis process.

## 7. TOWARDS BIOLOGICAL SWARM TRACKING IN THE WILD

The principals and the algorithmic pipeline of LocusTracker can be utilized to track locusts, other insects and various types of animals that exhibit swarming behaviours in the wild. However, there are a few modifications that should be performed in order to make LocusTracker robust for tracking of insects in the wild.

Since our model was developed for behaviour analysis in laboratory conditions which usually have somewhat controlled lighting settings we did not need to handle significant illumination changes in the environment and could focus on other challenges such as developing logical rules for association of previous and current locust locations, investigating algorithms for forming of long trajectories and resolving conflicts in aggregated locust cluster situations.

To extend the performance of LocusTracker to natural outdoor environments, the background subtraction component's logic must be adapted for such conditions. The background subtraction component relies on the assumption that in laboratory environments the background is almost stationary (up to reflections and slight illumination variations). This assumption allows us to detect locusts extremely fast using difference images of foreground and background images without the need to rely on more sophisticated detection techniques that must attend to background and illumination changes. We believe that possible solutions can leverage recent progress in development of deep panoptic segmentation neural networks.

The aim of panoptic segmentation techniques is to jointly perform semantic and instance segmentation, implying that each pixel is classified as belonging to a particular predefined class such as locust, background, etc. and that different instances, i.e., different locusts have different labels.

Using such methods may enable to segment locations of locusts as well as locust clusters from the provided images. After obtaining locust locations we will be able to proceed with the existing tracking pipeline using the obtained locust blobs. Possible examples for such networks are [25, 26], however careful attention should be put in developing panoptic segmentation networks that are suitable for segmentation of small objects in the image such as insects. We leave the development of such networks for future work.

An additional potential avenue for adapting LocusTracker to operation in natural environments is to draw inspiration from a recent work that develops a recurrent neural network for tracking of tiny insects in cluttered natural environments [27]. This work addresses some of the limitations of deep-learning based small object detection and detection of occluded objects. This work can serve as a starting point for future research and can be integrated with our locust trajectory association and locust cluster analysis modules when attempting to track locust swarms in natural environments.

## 8. CONCLUSIONS & FUTURE WORK

In this work we present LocusTracker, a markerless visual object tracking framework for moving locusts and other insects. LocusTracker consists of 6 main modules and relies on the principles of tracking by detection. LocusTracker puts special emphasis on long-time tracking of large numbers of insects that are located in close proximity to each other. The tracking algorithm developed is tested in a variety of scenes that contained moving locusts and provided promising results for tracking of locusts in a controlled laboratory environment models. Future improvements of LocusTracker are mainly to extend its capabilities and allow its deployment in natural real-world environments for tracking of locusts, insects and other types of natural or robotic swarms.

## REFERENCES

[1] Moshe Guershon, Roee Mordechai Francos, Amir Ayali, and Tali Hatuka. Locust behavior and city topology: A biodynamic approach for assessing urban flows. *Iscience*, 27(6), 2024.

[2] Alfred M Bruckstein. Why the ant trails look so

straight and nice. *The Mathematical Intelligencer*, 15(2):59–62, 1993.

[3] Tamás Vicsek, András Czirók, Eshel Ben-Jacob, Inon Cohen, and Ofer Shochet. Novel type of phase transition in a system of self-driven particles. *Physical review letters*, 75(6):1226, 1995.

[4] Tamás Vicsek and Anna Zafeiris. Collective motion. *Physics reports*, 517(3-4):71–140, 2012.

[5] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7:1–41, 2013.

[6] Douglas H Kelley and Nicholas T Ouellette. Emergent dynamics of laboratory insect swarms. *Scientific reports*, 3(1):1073, 2013.

[7] Gil Ariel and Amir Ayali. Locust collective motion and its modeling. *PLOS computational Biology*, 11(12):e1004522, 2015.

[8] Daniel Knebel, Ciona Sha-Ked, Noa Agmon, Gil Ariel, and Amir Ayali. Collective motion as a distinct behavioral state of the individual. *Iscience*, 24(4), 2021.

[9] Michael Amir, Noa Agmon, and Alfred M Bruckstein. A locust-inspired model of collective marching on rings. *Entropy*, 24(7):918, 2022.

[10] Thomas Dagès and Alfred M Bruckstein. Doubly stochastic pairwise interactions for agreement and alignment. *SIAM Journal on Applied Mathematics*, 82(4):1246–1266, 2022.

[11] Kristi Žampach, Jiří Ulrich, Tomáš Rouček, Martin Stefanec, Dominik Dvořáček, Laurenz Fedotoff, Daniel Nicolas Hofstadler, Fatemeh Rekabi-Bana, George Broughton, Farshad Arvin, et al. A vision-based system for social insect tracking. In *2022 2nd International Conference on Robotics, Automation and Artificial Intelligence (RAAI)*, pages 277–283. IEEE, 2022.

[12] Jiří Ulrich, Ahmad Alsayed, Farshad Arvin, and Tomáš Krajník. Towards fast fiducial marker with full 6 dof pose estimation. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, pages 723–730, 2022.

[13] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014.

[14] Tucker Balch, Zia Khan, and Manuela Veloso. Automatically tracking and analyzing the behavior of live insect colonies. In *Proceedings of the fifth international conference on Autonomous agents*, pages 521–528, 2001.

[15] Gil Ariel, Yotam Ophir, Sagi Levi, Eshel Ben-Jacob, and Amir Ayali. Individual pause-and-go motion is instrumental to the formation and maintenance of swarms of marching locust nymphs. *PloS one*, 9(7):e101636, 2014.

[16] Diego Marcovecchio, Natalia Stefanazzi, Claudio Delrieux, Ana Gabriela Maguitman, and Adriana Ferrero. A multiple object tracking system applied to insect behavior. In *XVIII Congreso Argentino de Ciencias de la Computación*, 2013.

[17] Osamu Yamanaka and Rito Takeuchi. Umatracker: an intuitive image-based tracking platform. *Journal of Experimental Biology*, 221(16):jeb182469, 2018.

[18] Katarzyna Bozek, Laetitia Hebert, Yoann Portugal, and Greg J Stephens. Markerless tracking of an entire insect colony. *BioRxiv*, pages 2020–03, 2020.

[19] Malika Nisal Ratnayake, Don Chathurika Amarathunga, Asaduz Zaman, Adrian G Dyer, and Alan Dorin. Spatial monitoring and insect behavioural analysis using computer vision for precision pollination. *International Journal of Computer Vision*, 131(3):591–606, 2023.

[20] Femke Batsleer, Dries Bonte, Daan Dekeukeleire, Steven Goossens, Ward Poelmans, Eliane Van der Cruyssen, Dirk Maes, and Martijn L Vandegehuchte. The neglected impact of tracking devices on terrestrial arthropods. *Methods in Ecology and Evolution*, 11(3):350–361, 2020.

[21] Hajnalka Málik-Roffa, Dávid Tőzsér, Béla Tóthmérész, and Tibor Magura. Bugtracker: Software for tracking and measuring arthropod activity. *Diversity*, 15(7):846, 2023.

[22] Alan Lukezic, Tomas Vojir, Luka ˇCehovin Zajc, Jiri Matas, and Matej Kristan. Discriminative correlation filter with channel and spatial reliability. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6309–6318, 2017.

[23] Alfonso Pérez-Escudero, Julián Vicente-Page, Robert C Hinz, Sara Arganda, and Gonzalo G De Polavieja. idtracker: tracking individuals in a group by automatic identification of unmarked animals. *Nature methods*, 11(7):743–748, 2014.

[24] Tristan Walter and Iain D Couzin. Trex, a fast multi-animal tracking system with markerless identification, and 2d estimation of posture and visual fields. *Elife*, 10:e64000, 2021.

[25] Rohit Mohan and Abhinav Valada. Efficientps: Efficient panoptic segmentation. *International Journal of Computer Vision*, 129(5):1551–1579, 2021.

[26] Yanwei Li, Xinze Chen, Zheng Zhu, Lingxi Xie, Guan Huang, Dalong Du, and Xingang Wang. Attention-guided unified network for panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7026–7035, 2019.

[27] Lars Haalck, Sebastian Thiele, and Benjamin Risse. Tracking tiny insects in cluttered natural environments using refinable recurrent neural networks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 7126–7135, 2024.