Online Tracking of Ants Based on Deep Association Metrics: Method, Dataset and Evaluation

Xiaoyan Cao[a], Shihui Guo[a,*], Juncong Lin[a,*], Wenshu Zhang[b], Minghong Liao[a]

[a]*School of Informatics, Xiamen University, Xiamen, Fujian, China*
[b]*School of Technologies, Cardiff Metropolitan University, U.K.*

Abstract

Tracking insect movement in a social group (such as ants) is challenging, because they are not only visually identical but also likely to perform intensive body contact and sudden movement adjustment (start/stop, direction turning). To address this challenge, we introduced an online multi-object tracking framework by combining both the motion and appearance information of ants. We obtained the appearance descriptors by using the ResNet model for offline training on a small (N=50) sample dataset. For online association, cosine similarity metric computes the matching degree between historical appearance sequences of the trajectory and the current detection. We validated our method in both indoor (lab-setup) and outdoor video sequences. The results show that the accuracy and precision of the model are 99.22%±0.37% and 91.93%±1.46% across 46041 testing samples, with real-time tracking performance. Additionally, we offered a public dataset of ant tracking with 46091 samples for future research in relevant domains.

Keywords: Ant Tracking, ResNet Model, Mahalanobis Distance, Appearance Descriptors

1. Introduction

Behavioral research on social insects (such as ants) enables us to understand their group division, task specialization and other types of distributed problems [1], which may potentially benefit modern applications, such as wireless communication and swarm robot control. Tracking individuals in an insect group for an

[*] Corresponding authors:guoshihui@xmu.edu.cn, jclin@xmu.edu.cn

extended period is a common approach to understand their behavior. However, manually labeling individual insects in a video is labor-intensive, especially for grouping organisms, and prone to human errors [2]. Automatic tracking methods by computer vision can significantly improve the processing speed and accuracy of vital information collection [3, 4, 5].

However, it is a non-trivial task to track such social insects. Their appearances are almost visually identical, and their grouping behavior leads to intensive interactions and occlusions between each other. Existing methods distinguish individuals by using RFID tags, QR codes, color paints [5]. However, RFID tags are only detectable within fixed distances; QR codes are usually too heavy for small insects; color paints tend to fall off over time. Marking insects by these techniques requires particular caution and is labor-intensive for professionals. Furthermore, these methods are not conducive to repeated research on insects.

This paper aims to resolve the challenges above and offers a solution to track unlabeled ant individuals automatically. Our method proves to be useful in both indoor and outdoor environments. The motivation of this work is to significantly improve the work efficiency of biological researchers in relevant tasks. The contributions of our work include:

- We introduce an online Multi-Object Tracking (MOT) framework to track ant individuals. This framework combines both motion and appearance matching, which effectively prevents trajectory fragments and ID switches from long-term occlusion caused by frequent interactions of ants, achieving efficient and high-precision tracking.

- We obtain ant appearance features based on the ResNet model with cosine similarity metric, to track unlabeled ants for a long time in a fixed position camera. The experiments show that our method is successful and robust with only a small size (N=50) of the training dataset, which makes it feasible to be applied in real applications with no need to construct a large training dataset.

- We construct a dataset of ant tracking with a total size of 46091 samples. We built the dataset following the standard MOT formulation. In contrast to an extensive collection of human tracking datasets, there are few datasets of ant tracking which are publicly accessible. We believe this dataset will benefit future works with relevant research objectives.

2. Related Work

2.1. Data Association Methods

In recent years, the task of object detection has made significant progress [6]. It leads to a majority number of the MOT frameworks, which adopt the tracking-by-detection paradigm. This paradigm first uses a pre-defined object detector to locate objects of the current frame, then associates detections with trajectories to update tracklets. Depending on the computational efficiency, these methods are categorized as offline and online.

The offline MOT methods are in general formulated as a global optimization problem, such as Network Flow [7], Multi-Cut [8, 9], Generalized Maximum Multi Clique Problem (GMMCP) [10] etc. Multi-Cut clusters detections in space and time for the task of re-identification [9]. The GMMCP is an ideal tracking method, which considers the pairwise relationship of all targets in a set of frames. It can be transformed into an optimal solution problem in integer programming [10]. However, searching for the optimal global solution limits the class of objective functions. Using a non-Markov method to extend the objective function can enhance the global consistency of trajectory [11].

In contrast, the online MOT methods match detections and trajectories frame-by-frame, and heavily rely on object detection results due to a short window of the matching process. The Global Nearest Neighbor algorithm (GNN) is widely used for data association [12, 13, 14], which calculates the metric of information matching between objects in two consecutive frames. The matching criteria are based on the information, including appearance, position, direction [13, 14]. In the past, Kalman filter [13, 14] and particle filter [15, 16] were mostly applied to data association process. The standard Kalman filter is used to construct a linear model of constant velocity to predict the object position in the next frame [13, 14], thereby narrowing the searching range of detected bounding boxes [17]. However, in the situation where the state transition and observation model of an object are non-linear, an extended Kalman filter [18] generates more accurate predictions. Particle filters are also used to deal with non-linear problems [15]. However, as the number of particles increases, the computational complexity increases exponentially. The Markov Chain Monte Carlo (MCMC) sampling, instead of traditional importance sampling, can reduce the computation complexity of particle filter [16]. Besides, the characteristics used for estimating the association probability are different for various types of objects. Therefore, researchers interactively adapt the association scoring function, thus improving tracking reliability [19].

In this paper, we adopt the online MOT framework and use the standard Kalman filter to predict the motion state of objects, to achieve real-time tracking. According to the prediction results, some objects are initially filtered and thus excluded from further matching process. We introduce the measurement of cosine appearance similarity to associate detected ants with the trajectories.

2.2. Appearance Features

Extracting appearance features is essential for solving the MOT problem. Features, such as color histogram [17], SIFT [12] and feature fingerprint [20], have been widely used in appearance modeling. In recent years, using deep networks to extract characteristic appearance features has become the mainstream [21]. Many tracking frameworks have been introduced for this purpose, such as Residual Network (ResNet) [22, 23], Long Short-Term Memory (LSTM) [22, 24], Siamese [25], and Quadruplet Convolutional Neural Network (Q-CNN) [26, 27].

Substantial or long-term occlusion brings a severe challenge for MOT. A loop-structured two-stage classifier with the kernel is proposed to solve this problem [28]. In this work, once the target object is severely occluded, an optimal classifier is selected to re-detect it according to the principle of entropy minimization. Other research works introduced the Spatial-Temporal Attention Mechanism, which adaptively assigns different weights of attention to calculate appearance features [29]. Similarly, the LSTM network also takes into account the tracking information of the prior period, then extracts the most reliable information for the current frame [22, 24], as well as repairs the previous wrong association [30].

Detection noise will seriously affect the effect of the appearance model, and some works have focused on handling this problem [31, 32, 33]. One solution is to learn the alternative tracking assumptions via CNN and automatically adjust the bounding box, thus getting a reasonable detection result for matching [31]. Other approaches integrate body joint detection [32] and posture information [33] to infer occlusion state and object direction, which can mitigate the effects of detection noise.

Inspired by research works mentioned above, we use the ResNet model for pre-training to obtain the appearance descriptor. Also, we store the last 100 frames for each trajectory as a gallery of the target, then use the historical appearance information for the cosine similarity metric.

2.3. Social Insects Tracking

Computer vision technology has benefited biologists' research on insects. Researchers have introduced MOT frameworks for tracking social insects, such as bees, fruit flies, and ants. For example, a tool, GuTomasi tracker [34], can efficiently track the movement of bees. For Drosophila, researchers cut the wings of Drosophila and let them move in a 2D culture dish. They proposed a tracking framework by integrating the planar geometric characteristics of Drosophila body and motion direction [13, 14]. Other methods use motion information; thus, the Kalman filter is used to estimate flight status in 3D space, and the Hungarian algorithm is used to match Drosophila under multi-viewpoints [35, 18]. In the ant colony, concurrently tracking multiple ants leads to high computational complexity. Using the MCMC method can significantly reduce the sampling time [16, 17], and some researchers developed a GPU-based semi-supervised framework to achieve efficient tracking [3].

In this paper, we not only verified the performance of our tracking framework in the lab environment but tested in an outdoor video. The results show high accuracy and precision. To the best of our knowledge, this is the first time to achieve ant colony tracking in a real-world scenario.

3. Method Overview

Our work provides an open dataset of detected & tracked ants and proposes a novel method for accurately tracking the ants. We captured moving ants in video sequences and prepared a dataset of detected & tracked ants as the benchmark (see details in Sec 4). Our method divides tracking into two stages: offline training and online tracking (Figure 1). For the offline training part, we adopt the ResNet model and obtain the appearance matching metric (Sec 5). We use a small dataset and effectively extract the hidden features that can be used to describe the appearance variations of a large number of individuals. For the online tracking part, the appearance and motion information are combined to associate trajectories with the detected ants (Sec 6). Our method requires no user interaction and dramatically reduces the time cost.

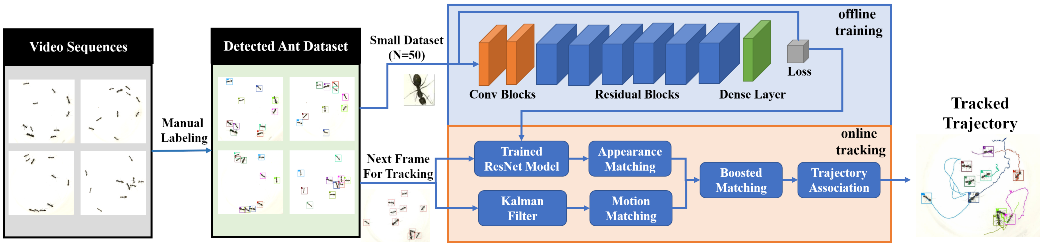


Figure 1: The pipeline of the proposed method.

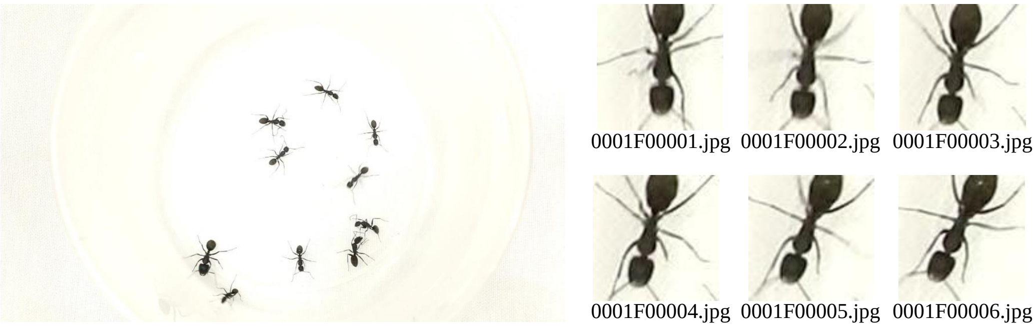
4. Dataset Preparation

4.1. Ant Collection in the Indoor Lab-setup

An ant colony is a complex biological system. According to reproductive ability, it can divide ants into queen and workers. Workers are further divided into major, medium, and minor ones based on their different forms of maturity. For captured video sequences in indoor lab-setup, we collected a group (N=70) of Japanese arched ants. In order to simulate the variations of ant individuals, we mix ants of different categories and body sizes: queen (17 mm) and workers (7.4-13.8 mm). We released all ants to the environment after the experiment.

4.2. Ant Video Capture in the Indoor Lab-setup

When capturing the indoor video, we used a transparent plastic container with a diameter of 10 cm to randomly mix the ants and load them into the container in batches (10 ants per batch). In the meantime, we applied anti-dusting powder in the inner wall of container, preventing ants from escaping from the container during the shooting. In our experiment, a high-resolution camera is used as the photography device and attached to a tripod. The captured video has a resolution of 1920x1080 with a frame rate of 25 fps in the format of H.264. Figure 2(a) shows a sample of captured images.



(a) a group of ants in a container

(b) captured sequences of ant No.1

Figure 2: Results from the task of data preparation in the setup of lab environment.

As can be seen from Figure 2, the visual appearance of the ant body in the video is difficult to differentiate due to overexposure. Besides, the low frame rate also causes the motion blur of ants' body, due to their high-speed movement. In contrast, experimental biologists typically use high-speed cameras (with a frame-rate of 500 [36]) to capture and analyze ant behaviors. These two factors pose significant challenges to the tracking problem.

4.3. Ant Video in the Outdoor Environment

For video sequences in the outdoor environment, we directly download the video from an online video website *DepositPhoto* [1]. We purchased the video with a one-time payment of 47.8 USD. The use of this video follows a royalty-free license. The video is 18 seconds in length, 30 frames per second, and 1280x720 pixels in resolution.

4.4. Dataset Processing

After the procedure of video collection, each ant in a video is marked frame by frame. The size of the bounding box is 96x96 and 64x64 for indoor and outdoor scenes respectively. We designed a MATLAB-based labeling software Visual-MarkData with three primary purposes: 1) to minimize the user's labeling time, 2) to reduce the difficulty of labeling, 3) to acquire the appropriate data arrangement for the training task. Please see the Appendix for the detailed explanations on the labeling tool. The dataset and source code of our work, including the labeling software, are provided as the supplementary materials and uploaded to an online public repository [2].

In order to quantitatively evaluate the tracking performance, we save the position information of each ant in the pixel coordinates in the original images. We formulate the data hierarchy following the standard format of MOT challenge. The manual labeling on the five indoor sequences and one outdoor sequence costs eight staff-days. In total, we collected 24050 and 22041 samples for indoor and outdoor environments, respectively. For detailed descriptions, please refer to our supplementary materials.

5. ResNet-based Appearance Descriptor

5.1. ResNet Network Architecture

We use the ResNet, a relatively shallow network architecture that can undertake online multi-objects tracking tasks [37]. The network consists of six residual blocks, two convolutional layers, and one pooling layer, for a total of 15 layers (Table 1). The training data are images of a detected ant (Figure 2(b)), and then the feature map is reduced to 12x12 through a series of convolutional layers. After

[1] http://www.depositphoto.com

[2] https://github.com/holmesww/multi-ants-tracking

that, the model extracted the feature vector of 128 dimensions from the fully connected layer. Finally, batch normalization (BN) is applied to obtain a normalized feature vector for the cosine similarity metric.

Name	Path Size/Stride	Output Size
Conv 1	3x3/1	32x96x96
Conv 2	3x3/1	32x96x96
Max Pool 3	3x3/2	32x48x48
Residual 4	3x3/1	32x48x48
Residual 5	3x3/1	32x48x48
Residual 6	3x3/2	64x24x24
Residual 7	3x3/1	64x24x24
Residual 8	3x3/2	128x12x12
Residual 9	3x3/1	128x12x12
Dense 10		128

Table 1: Hierarchy of the ResNet model in our work

5.2. Cosine Similarity Metric Classifier

We here train an appearance matching model with a cosine similarity metric. The training dataset is a paired data set $D = (x_i, y_i)_{i=1}^{N}$, where x is the input image, with the associated ant ID number $y_i = 1, \cdots, N$. Our method only requires a small dataset (N=50) and is capable of applying to a much larger dataset (N=24000 and 22041 for the indoor and outdoor environments, respectively).

Usually, CNN places a softmax classifier on top of the network for calculating the score of each class. The softmax classifier will select the class with the maximum probability as the output. The softmax classifier formula is as follows:

$$p(y = k|r) = \frac{\exp\left(w_k^T r + b_k\right)}{\sum_{n=1}^{C} \exp\left(w_n^T r + b_n\right)} \tag{1}$$

where r represents the underlying feature trained by the ResNet model, k represents the k^{th} class tag, w, b are parameters of the softmax classifier. Its loss function can be stated as:

$$L(D) = -\sum_{i=1}^{N} \sum_{k=1}^{C} \mathrm{gt}_{y_i=k} \cdot \log p\left(y_i = k|r_i\right) \tag{2}$$

where $L(D)$ is the sum of the cross entropy losses of the N images, $\log p(y_i = k|r_i)$ is the predicted result of the i^{th} image in the k^{th} label, and $\mathrm{gt}_{y_i=k}$ is the ground truth. The ResNet model backpropagates the fit error to adjust parameters during iteration.

The posterior probability of the softmax classifier is determined by the distance between the input and decision boundaries, which is valid for multi-classification tasks. However, our goal is to distinguish objects from the same class. For several images of the same type of objects, the posterior probability obtained by the softmax classifier cannot be directly used to calculate their similarities.

We modify the parameters of the softmax classifier in order to obtain a cosine similarity metric classifier that can measure the similarity of the same type objects [23]. First, in the previous ResNet network architecture, the fully connected layer has been normalized using BN to ensure that it is unit-length $\|f_\theta(x)\| = 1, \forall x \in R^D$. Second, weights are normalized, i.e. $\tilde{\omega} = \frac{\omega}{\|\omega_k\|_2}$ $\forall k = 1, \cdots, C$. The cosine similarity metric classifier can be expressed as:

$$p\left(y_i = k|r_i\right) = \frac{\exp\left(\kappa \cdot \tilde{\omega}_k^T r_i\right)}{\sum_{n=1}^{C} \exp\left(\kappa \cdot \tilde{\omega}_n^T\right)} \tag{3}$$

where κ represents the free scaling parameter.

Since the weight and offset are normalized in the training process, the distance effect between the posterior probability and the decision boundary is eliminated. The range of direction angles of each class at the decision boundary is the only factor to be adjusted to get the final classifier. Therefore, we can obtain the object similarity in different images by calculating the cosine distance between the posterior probabilities.

6. Online Multi-Ant Tracking Framework

6.1. Trajectory Association Model

In a MOT task, the motion information of the object is critical. However, motion paths of ants are irregular with frequent transitions of being static, straightthrough, U-turn. Such a significant motion uncertainty is likely to cause the problem of ID switches. Therefore, this paper uses the ResNet model to train the appearance features of ants offline, then combines the motion and appearance information to associate objects and trajectories in each frame.

6.1.1. Motion Matching

We calculate the square of the Mahalanobis distance between the object position predicted by the Kalman filter and the detected ant position to measure the degree of motion matching [38], which can be expressed as:

$$d^{(1)}(i,j) = \left(d_j - y_i\right)^T S_i^{-1} \left(d_j - y_i\right) \tag{4}$$

where d_j represents the position of the j^{th} detection box, y_i represents the position of the i^{th} trajectory predicted by Kalman filter, and S_i represents the covariance matrix between the i^{th} trajectory and the detected bounding box.

To further characterize the matching results between trajectories and objects, this paper introduces the motion association signal and calculates the 90% confidence interval of the Mahalanobis distance through the inverse chi-square distribution. If the squared Mahalanobis distance is less than the threshold, the association between the trajectory and the object is *potentially* successful. The formula can be expressed as a Bernoulli distribution:

$$b_{ij}^{(1)} = \begin{cases} 1, & d^{(1)}(i,j) < t^{(1)}, \\ 0, & otherwise. \end{cases} \tag{5}$$

where $b_{ij}^{(1)}$ is the association result between the i^{th} trajectory and the j^{th} detection box, which is a 0-1 binary variable. The fact of $b_{ij}^{(1)} = 1$ indicates that the j^{th} detection box is *potentially* associated with the i^{th} trajectory, given the metric of motion matching. The squared Mahalanobis distance threshold $t^{(1)}$ is set to 15.507 in our experiment.

6.1.2. Appearance Matching

In this paper, the ResNet model utilizes the metric of cosine similarity to perform the offline training of ant's appearance features. We obtain a 128-dimensional feature vector as the appearance descriptor r [38], which characterizes an ant's appearance. In the online tracking process, the i^{th} trajectory is designated with a set $\mathbf{K}_i$ of feature vectors (N=100) from the most recent successful associations. The formula expression is denoted as follows:

$$d^{(2)}(i,j) = \min_k \left\{1 - r_j r_k^{(i)}\right\}, \text{for } r_k^{(i)} \in \mathbf{K}_i \tag{6}$$

where r_j denotes the eigenvector value of the j^{th} detection box under the cosine similarity association model, $r_k^{(i)}$ denotes the k^{th} appearance descriptor of the i^{th} trajectory, and $d^{(2)}(i, j)$ denotes the matching degree between the appearance of the j^{th} detection box and the i^{th} trajectory. The above equation computes the minimal distance between the current detection box and all trajectories. This metric is used to filter the tracking trajectories and associate them with the detection box in the following step.

This paper introduces an appearance-associated signal $b_{ij}^{(2)}$. An object is *potentially* associated with one trajectory if its appearance matching degree satisfies the following condition:

$$b_{ij}^{(2)} = \begin{cases} 1, & d^{(2)}(i, j) < t^{(2)}, \\ 0, & otherwise. \end{cases} \tag{7}$$

In the formula, $b_{ij}^{(2)}$ follows the Bernoulli distribution. The threshold $t^{(2)}$ is set to 0.2, based on the observed outputs from the ResNet model. The fact of $b_{ij}^{(2)} = 1$ indicates that the j^{th} detection box is *potentially* associated with the i^{th} trajectory, given the metric of appearance matching.

6.1.3. Comprehensive Matching

Our method obtains a small set of candidate trajectories **CT** for each detection box, considering both the motion and appearance information. The i^{th} trajectory is added to the candidate set of the j^{th} detection box, if the condition of $b_{ij} = 1$ is satisfied:

$$b_{ij} = \prod_{m=1}^{2} b_{i,j}^{(m)} \tag{8}$$

This equation indicates that only if both $b_{i,j}^{(1)}, b_{i,j}^{(2)}$ are true, the i^{th} trajectory is associated with the j^{th} detection box. Among all trajectory candidates in **CT** of the j^{th} detection box, we associate the detection box with the trajectory with the maximal value of the appearance similarity.

6.2. Matching Cascade Model

When the object is blocked for a long duration, the uncertainty of the object position predicted by the Kalman filter increases as the occlusion time passes by. It indicates that the covariance matrix increases, resulting in a decrease in the Mahalanobis distance. If another object moves to this point at this moment and competes with the object for the detection box, the trajectory with a longer occlusion time is more likely to be associated. It causes the problem of ID switch and the tracking discontinuity. In order to solve this problem, this paper introduces the concept of matching cascade [38], which preferentially matches the trajectory closest to the time of the last association. In other words, give priority to allocation the trajectory with the same occlusion time, in order to avoid ID switches.

6.3. Track Update

After cascade matching of the current frame, the trajectories need to be updated. There are three operations to update a trajectory: set to be a tentative trajectory, set as a confirmed trajectory, delete a trajectory. For detections that cannot be associated with existing trajectories, a new trajectory will be created and considered to be tentative in the first two associations. We require three consecutively-associated frames before converting this tentative trajectory into a confirmed one; otherwise, delete it [38].

If an unmatched trajectory has been matched successfully in the current frame, we compute the mean moving speed of the object to estimate its position in the next frame. Otherwise, suspend the tracking of this trajectory. If the lost frames of a confirmed trajectory exceed the predefined maximum number of A_{max}, the trajectory will be deleted.

7. Results and Discussions

7.1. Hardware, Software and Parameters

The hardware configuration is Intel (R) Core (TM) i7-9700K GPU GeForce RTX 2080 Ti, and the memory is 16.0GB, and we implemented in Python environment. All source code and dataset are attached as supplementary files. In the training experiment, the batch size of the ResNet model is set to 16, the learning rate is set to 1e-4, and the number of steps is 100000. In this model, there are 2800,864 network parameters. For the online tracking framework, this article sets $A_{max} = 30$ and the minimum cosine distance to be 0.2.

7.2. Training ResNet

The training dataset contains 50 images of detected ants. To construct this dataset, we randomly select six ants from all detected and labeled ants, and randomly select 8 to 10 images for each of them. A larger dataset can be used to train ResNet. In our pilot experiment, we use 19849 images for training and achieve the performance of MOTA: 99.8%, MOTP: 94.0%. Although our current implementation uses only 50 samples, this does not cause a significant deterioration of our tracking performance. See detailed discussions on the statistics of our method in the following section. The use of a small training dataset is one of the advantages of our method since this eliminates the demand of manually constructing a large training dataset. This is similar to existing works [28, 39] which can track targets at high accuracy and effectively reduce the labeling demand. However, these two methods focus on the task of single-target tracking, which is different from our focus. The training time based on the above hardware platform is about 30 minutes. In addition, a set of manually-labeled tracking results is used as ground truth to evaluate the performance.

Using the above model and parameter settings, after 100,000 iterations, the total training loss and accuracy convergence of the 128-dimensional sample features set are shown in Figure 3. The blue curve represents the decline in total training loss, and the red curve represents the increase in accuracy. The training loss and accuracy of the model both had a stable convergence performance. After nearly 40,000 iterations, the total loss of the model converges to about 1.1, and the accuracy tends to be stable.

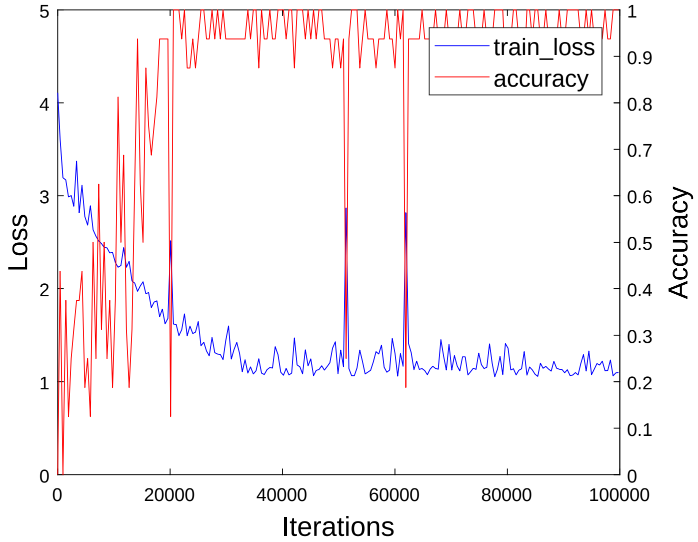
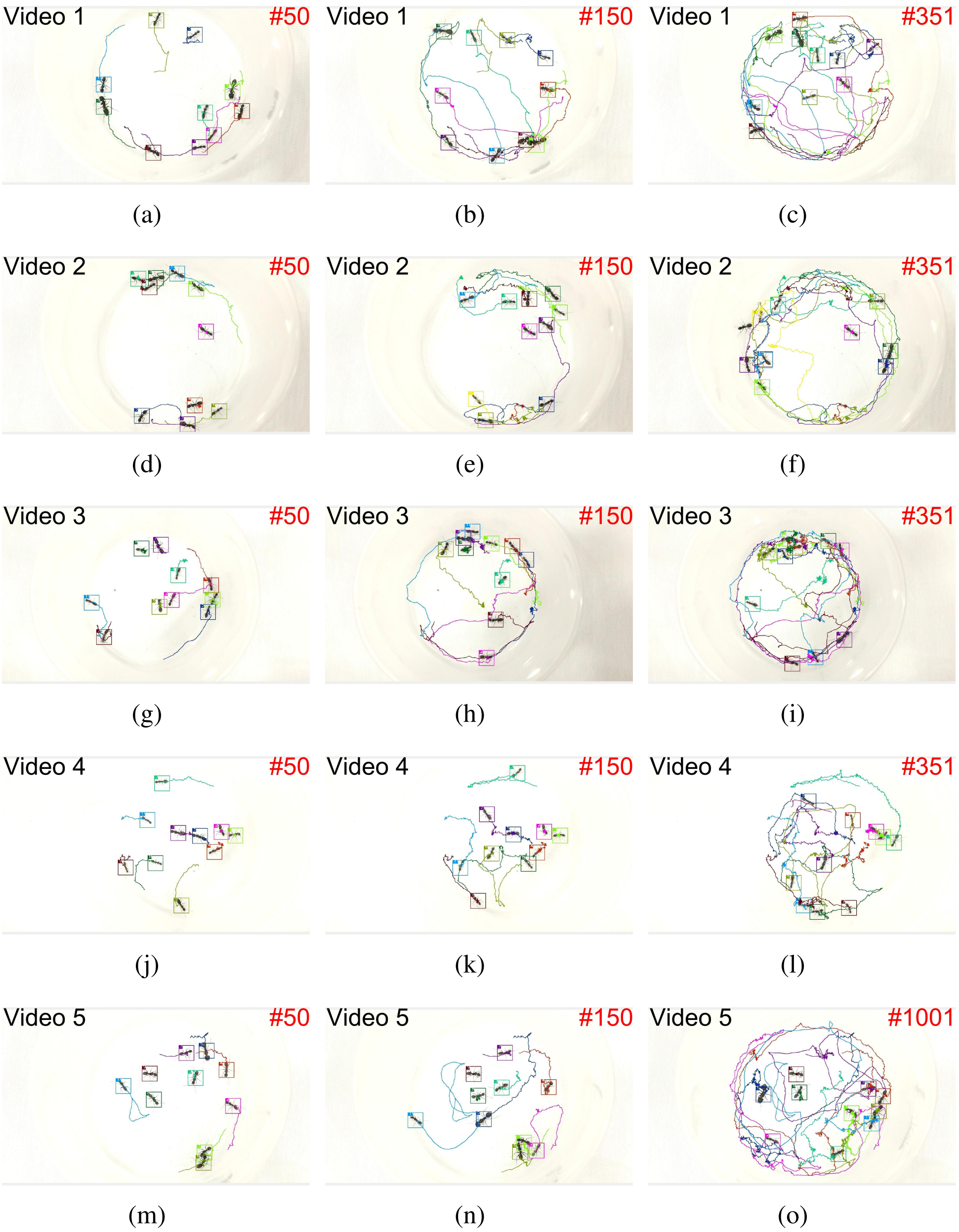


Figure 3: the ResNet model training loss and accuracy convergence curves.

7.3. *Tracking Performance in Indoor Lab-setup*

The results of multi-ants tracking video (indoor lab-setup) obtained in the above experiment are shown in Figure 4. There are 10 ants in each video. Each ant



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

Figure 4: Tracking result in an indoor lab-setup environment.

is positioned in a square bounding box of different colors. The ant's ID number is indicated in the upper left corner of the bounding box, to observe the tracking accuracy. In order to quantify the performance of the model, we use the following eight indicators of MOT to evaluate tracking performance [40]:

- False Positive (FP): the total number of false alarms.
- False Negative (FN): the total number of objects that did not match successfully.
- Identity Switch (IDS): the total number of object switches during the tracking process.
- Fragments (FM): the total number of incidents that the tracking result interrupts the real trajectory.
- Mostly Tracked (MT): the proportion of predicted trajectories that hits successfully in real trajectories, over 80%.
- Most Lost (Mostly Lost, ML): the proportion of predicted trajectories that hits successfully in the real trajectory, no more than 20%.
- Multi-object Tracking Accuracy (MOTA): tracking accuracy of IDS considering false positives and missed objects.
- Multi-object Tracking Precision (MOTP): tracking consistency between labeled and predicted bounding boxes.
- Frame Rate (FR): the number of frames being tracked per second.

Table 2 demonstrates the tracking results in the indoor lab-setup environment (Figure 4). We found that the MOTA from Video 1 to Video 5 is close to full, MT value is 10, and IDS rarely happens. This indicates that all ants are accurately tracked in each video, with few tracking drifts. At the same time, the average MOTP value is as high as 91.74%, which means that we can track ants' positions precisely. A small amount of FN and FM indicates that most of the matches are successful, signifying that our model can robustly track multi-ants simultaneously in a global scope. Besides, due to the GPU-based online tracking process, the frame rate is around 35 fps, which is well above the standard video rate (24 fps).

In order to visualize the tracking performance in the whole video process, this paper compares tracking results with the ground truth for Video 5 and draws the tracking error graph (Figure 5). The left side of this figure shows the complete tracking of the video, where the abscissa represents the frame number, the ordinate

	FP	FN	IDS	FM	MT	ML	MOTA	MOTP	FR
Video 1	0	0	0	0	10	0	99.4	92.1	36.11
Video 2	7	10	4	7	10	0	98.8	91	34.58
Video 3	6	6	0	6	10	0	99.1	89.8	35.21
Video 4	8	8	4	6	10	0	98.9	91.8	35.45
Video 5	1	2	0	0	10	0	99.8	94	35.33
Outdoor	66	55	34	7	99	2	99.3	92.9	35.24

Table 2: Tracking performance evaluation.

represents the ant number, and the color represents the error value. The maximum value of the tracking error (FN or IDS) is 50. In the left image, false tracking is difficult to identify with human observation, and the overall rendering shows an excellent tracking effect. These indicate that the proposed model can track the ants in the laboratory environment accurately for a long time.

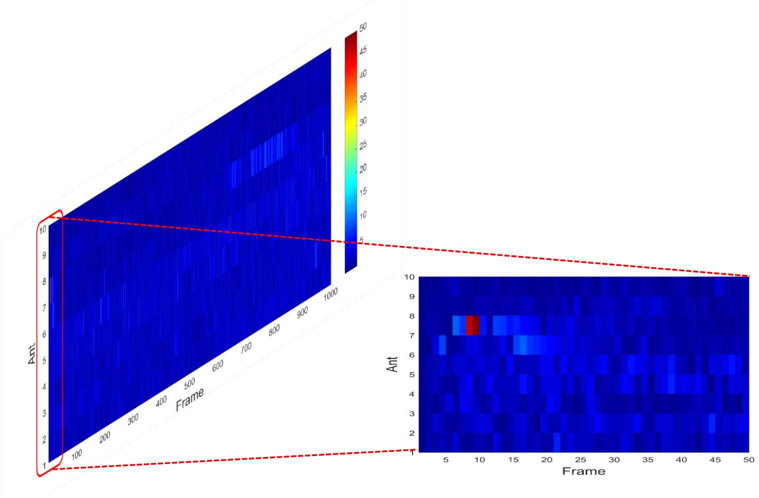


Figure 5: Tracking error graph over time in the case of indoor lab-setup environment.

7.4. Outdoor Environment

Results in the previous section showed that our method achieved satisfactory tracking results in the lab set-up. We hope to further evaluate the effectiveness of our tracking framework in real-world environments, to obtain more meaningful and persuasive evaluation-data. At the same time, in real-world ant swarms, their interactions and complex environments may expose the limitations of our model, which will provide reliable guidance for our future efforts. We selected a 569-frame (18 seconds) outdoor ant video for testing. In addition to the complex environment in the video, 101 ants appeared in the entire video, with an average of 40 ants per frame. Moreover, this outdoor test is far more challenging than our previous experiments because most ants are fast-moving. We directly used the pre-trained model from the ideal lab images, instead of re-training with outdoor images. Figure 6 shows the tracking results.

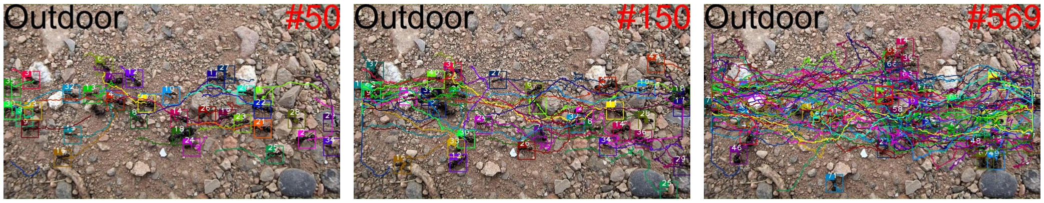


Figure 6: Tracking performance in an outdoor environment.

According to Table 2 (the last row for the case of outdoor environment), MOTA value is 99.3%, indicating that our model can still accurately track each ant in a complex real-world environment. The precision index - MOTP value - even exceeds the average of tracking results in previously ideal environments, consequently indicating that our model is robust. Besides, the value of FM is maintained at the same level compared to the ideal environment. The result shows that this metric is not affected by the concentration of ant colonies and the complexity of the environment, thus showing that our model can alleviate the trajectory fragmentation problem.

We observed that the metric of IDS increased to 34. An in-depth analysis showed that this is caused by the newly-entered ants which are associated with the existing trajectories. When constructing the ground truth, we assign a new trajectory to an ant entering into the video. The number of created trajectories by our method is 77, but 101 ants appeared in the video. The difference between these two numbers is the same as the IDS, which indicates that 34 ants who should have their new trajectories were incorrectly assigned to the existing trajectories of other ants, resulting in trajectory drift. However, it is worth noting that since all IDS occur due to the entrance of new ants, it also illustrates that our tracking model is accurate while ants are moving inside the video scene.

7.5. Occlusion Handling

The strategy of combining both motion and appearance matching leads to a boost of our capability in tracking ants when severe occlusion happens between individuals. Figure 7 shows an example. Ants No. 4 and 6 cross each other from Frame 109 to 136, and the occlusion lasts around one second. During this interval, these two ants demonstrate close body contact with each other, and the bounding boxes almost overlap (Frame 124). Our algorithm can still accurately identify and track both individuals after they depart from each other. The success of our method builds upon the capability to predict the motion state of ants.

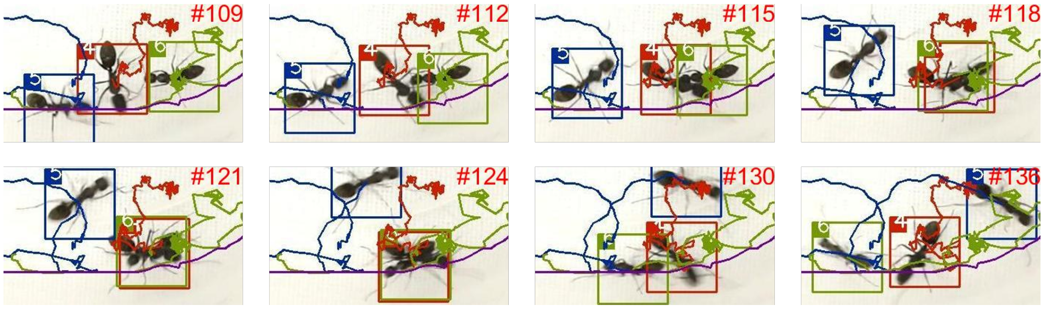
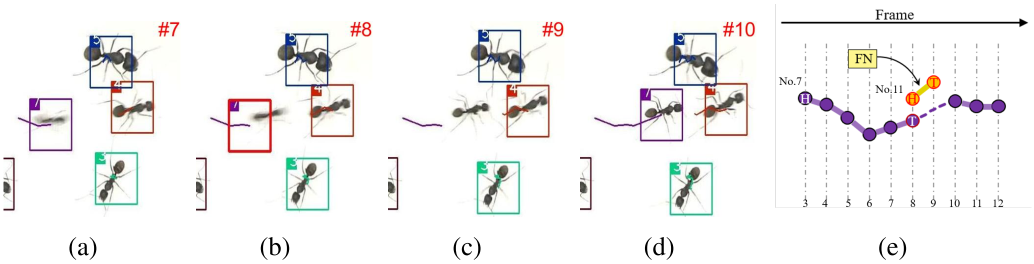


Figure 7: Tracking in the scenario involving severe occlusion.

7.6. Limitation and Failure Case

In order to further analyze the failure case of our method, we conduct an in-depth analysis of the tracking results from Frame 1 to 50 (Figure 5). The tracking plot of ant No.7 in Frame 9 is dark red, indicating an occurrence of either FN or IDS. However, the other ants do not report the corresponding tracking errors of IDS, which informs that it is a false negative on ant No.7, rather than an incident of ID switch. It is worth pointing out that the subsequent trajectory tracking is still correct, indicating that the model can re-identify the same ant after the trajectory is temporarily miss-associated.

To further analyze this FN problem, we intercept the video sequence from Frame 7 to 10, with a specific focus on ant No.7 (Figure 8). The results in Figure 8 show that in Frame 7 and 8, the body of the ant No.7 is blurred because of



(a) (b) (c) (d) (e)

Figure 8: Analysis of the FN of ant No. 7.

its fast movement. It indicates that the frame rate of our captured video is not sufficient and increases the tracking difficulty. In Frame 8, since the speed exceeds the threshold, the framework fails to associate the labeled detected bounding box with the trajectory. In this scenario, we create a new bounding box at the predicted position by the Kalman filter. Therefore, the trajectory lost its association in Frame 8 and 9. In other words, FN occurs twice, as shown in Figure 8(e).

Following the flow of trajectory update, a new trajectory, numbered as 11, is generated and set in a tentative state, as shown in Figure 8(e). However, in Frame 10, since the displacement of the ant No.7 is reduced, the motion matching with the Trajectory 7 satisfies the threshold, while the Trajectory 11 does not. So the detection of the ant No.7 is re-associated with the original Trajectory 7, and Trajectory 11 is deleted because it matches less than three frames. The analysis results show that the model can accurately recognize the ant after the omission and is capable of self-correction.

8. Conclusion

Tracking individuals in a group of social insects enables biologists to effectively and accurately understand their collective intelligence in decision making and task division [17]. This work combines both the motion and appearance information, and can successfully track unmarked ants in real-time. Our method can significantly reduce the cost of research and increase the speed of information collection. We use the ResNet model for offline training on a small sample data set of 50 images to describe the appearance of the ants. The experimental results show that the accuracy and precision of the model are 99.22% and 91.93% (average across 46041 testing samples), effectively alleviating ID switches or fragments caused by severe long-term occlusion. This confirms that an appearance descriptor from a small training dataset can effectively apply to an extensive testing dataset.

Our method can successfully handle the scenarios of indoor lab-setup and outdoor environment. To address one of our limitations (discussed in Sec. 7.6), we will explore alternative motion models to solve miss matching caused by the abrupt change in speed. For the problem of IDS in the outdoor scene, we plan to introduce additional mechanism to identify the new entrants in the scene and create separate trajectories for association.

For future work, we consider tackling the problem of detection to identify ants in images, thus building a complete detecting-tracking framework. Our current implementation uses the manually-labeled detection results as the baseline. However, we acknowledge that although the detection problem is independent of the track, the detection quality does significantly affect the tracking accuracy and precision. Therefore, a robust detection method is a critical component as part of the complete solution of ant behavior analysis. In addition, we will extensively test and improve our model using data sets with more complex occlusions and illumination changes, thereby enabling real-time tracking of ants in the real-world. Although the pilot experiment of an outdoor environment (Figure 6) preliminarily confirms the effectiveness of our method, a rigorous analysis is required for a wide range of outdoor scenarios. Further, drawing on the idea of transfer learning, we intend to extend our model to other kinds of ants, even other insects (such as bees). Investigating the differences in appearance descriptors across different ant and insect species may reveal exciting findings, in correlation with the biological research studies.

Acknowledgement

This work is supported by National Natural Science Foundation of China (61702433, 61661146002) and the Fundamental Research Funds for the Central Universities.

References

[1] T. D. Seeley, Honeybee democracy, Princeton University Press, 2010.

[2] T. Balch, Z. Khan, M. Veloso, Automatically tracking and analyzing the behavior of live insect colonies, in: Proceedings of the fifth international conference on Autonomous agents, ACM, 2001, pp. 521–528.

[3] C. Poff, H. Nguyen, T. Kang, M. C. Shin, Efficient tracking of ants in long video with gpu and interaction, in: 2012 IEEE Workshop on the Applications of Computer Vision (WACV), IEEE, 2012, pp. 57–62.

[4] M. Shen, C. Li, W. Huang, P. Szyszka, K. Shirahama, M. Grzegorzek, D. Merhof, O. Deussen, Interactive tracking of insect posture, Pattern Recognition 48 (11) (2015) 3560–3571.

[5] D. P. Mersch, A. Crespi, L. Keller, Tracking individuals shows spatial fidelity is a key regulator of ant social organization, Science 340 (6136) (2013) 1090–1093.

[6] W. Wang, J. Shen, X. Dong, A. Borji, R. Yang, Inferring salient objects from human fixations, IEEE Transactions on Pattern Analysis and Machine Intelligence PP (99) (2019) 1–1.

[7] Z. Lijuan, Z. Zhiping, Multiple target tracking using hierarchical data association based on network flows, Journal of Computer-Aided Design & Computer Graphics 30 (9) (2018) 1670–1677.

[8] S. Tang, B. Andres, M. Andriluka, B. Schiele, Multi-person tracking by multicut and deep matching, in: European Conference on Computer Vision, Springer, 2016, pp. 100–111.

[9] S. Tang, B. Andres, M. Andriluka, B. Schiele, Subgraph decomposition for multi-target tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5033–5041.

[10] A. Dehghan, S. Modiri Assari, M. Shah, Gmmcp tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4091–4099.

[11] A. Maksai, X. Wang, F. Fleuret, P. Fua, Non-markovian globally consistent multi-object tracking, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2544–2554.

[12] B. Miranda, J. Salas, P. Vera, Bumblebees detection and tracking, in: Workshop Vis. Observation Anal. Anim. Insect Behav. ICPR, 2012, pp. 1–4.

[13] P. Sirigrivatanawong, K. Hashimoto, Multiple drosophila tracking with behavior classification, in: 2017 IEEE International Conference on Mechatronics and Automation (ICMA), IEEE, 2017, pp. 1041–1046.

[14] P. Sirigrivatanawong, K. Hashimoto, Multiple drosophila tracking with heading direction in crossover and touching scenarios, in: 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO), IEEE, 2016, pp. 1954–1959.

[15] A. Veeraraghavan, R. Chellappa, M. Srinivasan, Shape-and-behavior encoded tracking of bee dances, IEEE transactions on pattern analysis and machine intelligence 30 (3) (2008) 463–476.

[16] Z. Khan, T. Balch, F. Dellaert, Mcmc-based particle filtering for tracking a variable number of interacting targets, IEEE transactions on pattern analysis and machine intelligence 27 (11) (2005) 1805–1819.

[17] M. Fletcher, A. Dornhaus, M. C. Shin, Multiple ant tracking with global foreground maximization and variable target proposal distribution, in: 2011 IEEE Workshop on Applications of Computer Vision (WACV), IEEE, 2011, pp. 570–576.

[18] D. Grover, J. Tower, S. Tavaré, O fly, where art thou?, Journal of The Royal Society Interface 5 (27) (2008) 1181–1191.

[19] H. Nguyen, T. Fasciano, D. Charbonneau, A. Dornhaus, M. C. Shin, Data association based ant tracking with interactive error correction, in: IEEE Winter Conference on Applications of Computer Vision, IEEE, 2014, pp. 941–946.

[20] A. Pérez-Escudero, J. Vicente-Page, R. C. Hinz, S. Arganda, G. G. De Polavieja, idtracker: tracking individuals in a group by automatic identification of unmarked animals, Nature methods 11 (7) (2014) 743.

[21] P. Li, D. Wang, L. Wang, H. Lu, Deep visual tracking: Review and experimental comparison, Pattern Recognition 76 (2018) 323–338.

[22] C. Kim, F. Li, J. M. Rehg, Multi-object tracking with neural gating using bilinear lstm, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 200–215.

[23] N. Wojke, A. Bewley, Deep cosine metric learning for person re-identification, in: 2018 IEEE winter conference on applications of computer vision (WACV), IEEE, 2018, pp. 748–756.

[24] A. Sadeghian, A. Alahi, S. Savarese, Tracking the untrackable: Learning to track multiple cues with long-term dependencies, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 300–311.

[25] X. Dong, J. Shen, Triplet loss in siamese network for object tracking, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 459–474.

[26] J. Son, M. Baek, M. Cho, B. Han, Multi-object tracking with quadruplet convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5620–5629.

[27] X. Dong, J. Shen, D. Wu, K. Guo, X. Jin, F. Porikli, Quadruplet network with one-shot learning for fast visual object tracking, IEEE Transactions on Image Processing 28 (7) (2019) 3516–3527.

[28] X. Dong, J. Shen, D. Yu, W. Wang, J. Liu, H. Huang, Occlusion-aware real-time object tracking, IEEE Transactions on Multimedia 19 (4) (2016) 763–771.

[29] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, M.-H. Yang, Online multi-object tracking with dual matching attention networks, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 366–382.

[30] N. Narayan, N. Sankaran, S. Setlur, V. Govindaraju, Learning deep features for online person tracking using non-overlapping cameras: A survey, Image and Vision Computing 89 (2019) 222–235.

[31] J. Shen, D. Yu, L. Deng, X. Dong, Fast online tracking with detection refinement, IEEE Transactions on Intelligent Transportation Systems 19 (1) (2017) 162–173.

[32] R. Henschel, Y. Zou, B. Rosenhahn, Multiple people tracking using body and joint detections, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2019, pp. 0–0.

[33] P. Li, J. Zhang, Z. Zhu, Y. Li, L. Jiang, G. Huang, State-aware re-identification feature for multi-target multi-camera tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2019, pp. 0–0.

[34] S. Gu, Y. Zheng, C. Tomasi, Efficient visual object tracking with online nearest neighbor classifier, in: Asian Conference on Computer Vision, Springer, 2010, pp. 271–282.

[35] R. Ardekani, A. Biyani, J. E. Dalton, J. B. Saltz, M. N. Arbeitman, J. Tower, S. Nuzhdin, S. Tavaré, Three-dimensional tracking and behaviour monitoring of multiple fruit flies, Journal of The Royal Society Interface 10 (78) (2013) 20120547.

[36] S. Guo, J. Lin, T. Wöhrl, M. Liao, A neuro-musculo-skeletal model for insects with data-driven optimization, Scientific reports 8 (1) (2018) 2129.

[37] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[38] N. Wojke, A. Bewley, D. Paulus, Simple online and realtime tracking with a deep association metric, in: 2017 IEEE International Conference on Image Processing (ICIP), IEEE, 2017, pp. 3645–3649.

[39] X. Lu, C. Ma, B. Ni, X. Yang, I. Reid, M.-H. Yang, Deep regression tracking with shrinkage loss, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 353–369.

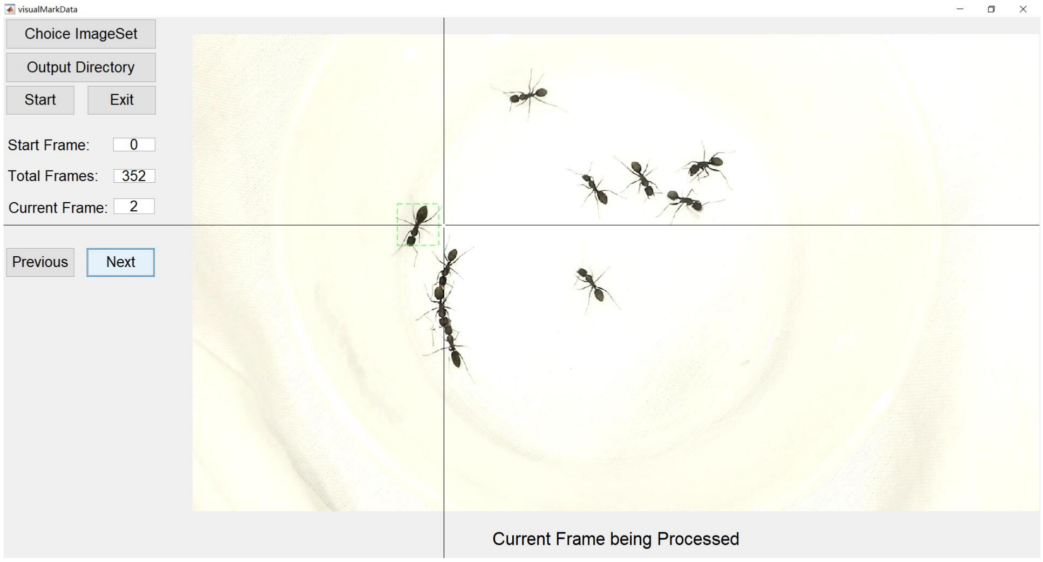
[40] K. Bernardin, R. Stiefelhagen, Evaluating multiple object tracking performance: the clear mot metrics, Journal on Image and Video Processing 2008 (2008) 1.

Appendix

We developed a labeling software VisualMarkData in this paper to collect the data used for detecting and tracking ants. Figure 9 shows the interface of software.

The main operations are as follows.

Prepare for labeling: Before labeling, the user clicks ”Choose Image Set” to select an image set, and ”Output Directory” to select the storage path of labeling results. The filename of the image set is defined in the format of ”AntXImageY”, where X is the number of ants in the first frame and Y is the size of the bounding box. For example, the image set, named as ”Ant10Image96”, indicates that this



Start Frame: 0

Total Frames: 352

Current Frame: 2

Figure 9: The interface of VisualMarkData.

image set contains 10 ants in the first frame and each ant will be marked with a bounding box with its size of 96x96.

Label: The user clicks on the body center of ant, and the software can automatically save the position information of ant in the current frame. Moreover, it can automatically intercept a square patch centered on the labeling point as the training image. It should be emphasized that the user does not label all the ants in one image simultaneously, but only labels the same ant until he/she finishes the entire image set. After that, the user labels another ant from the first frame. This way helps reduce the difficulty of labeling.

Next: The user clicks ”Next” to update the interface with the image on the next frame. The labeled position on the previous frame is displayed with a green-dotted frame, which can help the user quickly locate the target ant.

Previous: If the labeled position of the previous frame is incorrect, the user can click ”Previous” to return to the previous frame and re-label.

Check and modify: After the user finishes labeling the entire image set, he/she needs to check the quality of labeled results. In this case, the user can enter the frame to be modified.