Accurate detection and tracking of ants in indoor and outdoor environments

Meihong Wu[1¤], Xiaoyan Cao[1¤], Shihui Guo[1*¤],

1 School of Informatics, Xiamen University, Xiamen, Fujian, China

¤Current Address: School of Informatics, Xiamen University, Xiamen, Fujian, China
* guoshihui@xmu.edu.cn

Abstract

Monitoring social insects' activity is critical for biologists researching their group mechanism. Manually labelling individual insects in a video is labour-intensive. Automated tracking social insects is particularly challenging: (1) individuals are small and similar in appearance; (2) frequent interactions with each other cause severe and long-term occlusion. We propose a detection and tracking framework for ants by: (1) adopting a two-stage object detection framework using ResNet-50 as backbone and coding the position of regions of interest to locate ants accurately; (2) using the ResNet model to develop the appearance descriptors of ants; (3) constructing long-term appearance sequences and combining them with motion information to achieve online tracking. To validate our method, we build a video database of ant colony captured in both indoor and outdoor scenes. We achieve a state-of-the-art performance of 95.7% mMOTA and 81.1% mMOTP in indoor videos, 81.8% mMOTA and 81.9% mMOTP in outdoor videos. Our method runs 6-10 times faster than existing methods for insect tracking. The datasets and code are made publicly available, we aim to contribute to an automated tracking tool for biologists in relevant domains.

Author summary

The research on the group behavior of social insects is in great favor with biologists. But before analysis, each insect needs to be tracked separately in a video. Obviously, that is a time-consuming and labor-intensive work. In this manuscript, we introduce a detection and tracking framework that can automatically track the movement of ants in a video scene. The software first uses a residual network to detect the positions of ants, then learns the appearance descriptor of each ant as appearance information via another residual network. Furthermore, we obtain motion information of each ant by using the Kalman filter. Combining with appearance and motion information, we can accurately track every ant in the ant colony. We validate the performance of our framework using 4 indoor and 5 outdoor videos, including multiple ants. We invite interested readers to apply these methods using our freely available software.

Introduction

Insects can have both adverse and positive impacts on crop plants, affecting the agricultural production and ecological environment. This work is focused on a typical example of social insects - ants. They are reportedly to increase wheat yield by 36%

because ants can enhance soil water infiltration due to their tunnels and improved soil nitrogen [1]. Activity monitoring of insects, particularly social insects (such as ants in our case), proves to be remarkably challenging. Individuals in a colony of social insects are similar in appearance, and involve intensive interactions with each other, causing severe or long-term occlusion.

Manually following every ant in a colony is extremely tedious and time-consuming. Vision-based automatic detection and tracking methods can alleviate the process of manual labeling, allowing biologists to focus on behavior analysis. Traditional methods for multi-insect tracking methods are categorized as particle filtering (PF) and data-association-based tracking (DAT) ones. For tracking tasks in dense scenes, the PF method is computationally intensive [2]. Even with GPU acceleration, the processing speed is below the real-time frame rate [3]. Although the DAT method measures object similarity by modeling appearance, motion, and even other cues, it is difficult to maintain correct tracking during a long period of occlusion. Once trajectory drift occurs, the accumulated errors will result in tracking failure [4–6].

In recent years, with the popularity of computer vision, many advanced object detection and tracking methods have emerged.

Object detection

Existing methods in object detection are categorized as one-stage or two-stage, according to whether there is a separate stage of region proposal. One-stage frameworks (e.g., YOLO [7]) are fast, but their accuracy is typically slightly inferior compared with that of two-stage detection. The popularity of two-stage detection frameworks is enhanced by R-CNN [8], which proposes candidate regions via a selective search (SS) algorithm [9], thereby the detector focuses on these RoIs. However, using the SS algorithm [9] to generate region proposals is the main reason causing slow inference. Fast R-CNN [10] reduces the computational complexity of region proposals by downsampling the original image, while Faster R-CNN [11] proposes an RPN, which further improves the speed of training and inference.

Given the success of deep learning in general tasks of object detection, researchers also applied to detect specific groups of animals, such as a single mouse [12], fruit flies [13]. These methods are either limited to track a single object, or a fixed number of objects. General tools [14, 15] also offer the functionality to detect and track unmarked animals in the image. However, most of existing methods focus on the condition of ideal lab set-up and none of existing works reported the detection of ants in outdoor environments which contain diverse backgrounds and arbitrary terrains.

Our goal is to develop a framework for robust ant colony detection and tracking. Our work focuses on accurate detection and tracking ants in both indoor and outdoors scenes, and thus follows a two-stage detection framework as RPN-FCN [16]. Based on ResNet-50, we use position-sensitive score maps to encode the position information of the candidate bounding box proposed by RPN and then perform classification and regression, respectively.

Multi-object tracking (MOT)

In the last two decades, vision-based detection and tracking models have been widely used to study social insects [17, 18]. Appearance (particularly color) and motion information are the main metrics used in this category of method. Due to high similarity of ants' appearance, researchers either use the technique of pigmenting to create more distinct appearance features [19], or limit the observation to a laboratory setup [20, 21]. State-of-the-art methods, such as Ctrax [20] and idTracker [21], for insect tracking are tested in a laboratory setup and use background subtraction for foreground

segmentation. Notably, the operations of background modeling and foreground extraction are time-consuming.

The tracking-by-detection (TBD) paradigm is to match trajectories and detections in two consecutive frames, a process that requires metrics. The global nearest neighbor model measures motion state to achieve Drosophila tracking [22]. The global nearest neighbor model assumes that the motion state obeys the linear observation model, which commonly uses a constant velocity model - the Kalman filter (KF). However, changes in ants' speed and direction are difficult to predict, thus appearance information is integrated as a metric.

The DAT method is a mainstream method for ant colony tracking [4]. It allows a combination of multiple metrics, and uses Hungarian algorithm [23] to assign detections for trajectories. The PF method is suitable for solving nonlinear problems [2], but the growth in the number of particles leads to an exponential increase in the computational cost, preventing the effective multi-object tracking. Using Markov Chain Monte Carlo sampling can reduce computational complexity [24]. A GPU-accelerated semi-supervised framework can further improve tracking accuracy and performance [3].

When applying the methods above for tracking ant colonies, they are greatly disturbed by background noise and difficult to overcome the serious occlusion problem in dense scenes. Long short-term memory [25] and spatial-temporal attention mechanisms [26] have been developed to tackle the problem of long-term occlusion. A bilinear Long short-term memory structure that couples a linear predictor with input detection features, thereby modeling long-term appearance features [25]. The spatial-temporal attention mechanism is also suitable for the MOT task. The spatial attention module makes the network focus on the pattern of matching. Meanwhile the temporal attention module assigns different levels of attention to the sample sequence of the trajectory [26]. The TBD paradigm-based framework is dependent on detection results. Therefore, severe occlusion is likely to cause tracking failures. To prevent this situation, a detector with automatic bounding box repairing and adjustment is introduced by a cyclic structure classifier [27].

We propose a complete detection and tracking framework based on the TBD paradigm. We construct a gallery for each trajectory to store the sequence of historical appearance descriptors, which is used to online association metric. This strategy significantly mitigates the effects of long-term occlusion.

In this paper, we use a deep learning method to build a detection and tracking framework. Our method is based on the TBD paradigm and accomplishes the goal of online multi-ant tracking. To the best of authors' knowledge, this is the first work to achieve robust detection and tracking of ant colony in both indoor and outdoor environments (Fig 1). Our method is robust in tackling the challenge of visual similarity among colony individuals, handling diverse terrain backgrounds and achieving long-period of tracking. Our main contributions are as follows:

- We adopt a two-stage object detection framework, using ResNet-50 as the backbone and position sensitive score maps to encode regions of interest (RoIs). During the tracking stage, we use a ResNet network to obtain the appearance descriptors of ants and then combine them with motion information to achieve online association.

- Our method proves to be robust in both indoor and outdoor scenes. Furthermore, only a small amount of training data are required to achieve the goal in our pipeline, which are 50 images chosen for each scene in the detection framework and 50 labels randomly chosen for the tracking framework respectively.

- We construct an ant database with labeled image sequences, including five indoor videos (laboratory setup) and five outdoor videos, with 4983 frames and 115,433

labels in total. The labeled dataset are organized into the formats of PASCAL VOC and MOT Challenge for tasks of detection and tracking, respectively. The database will be made public for future research in this area.

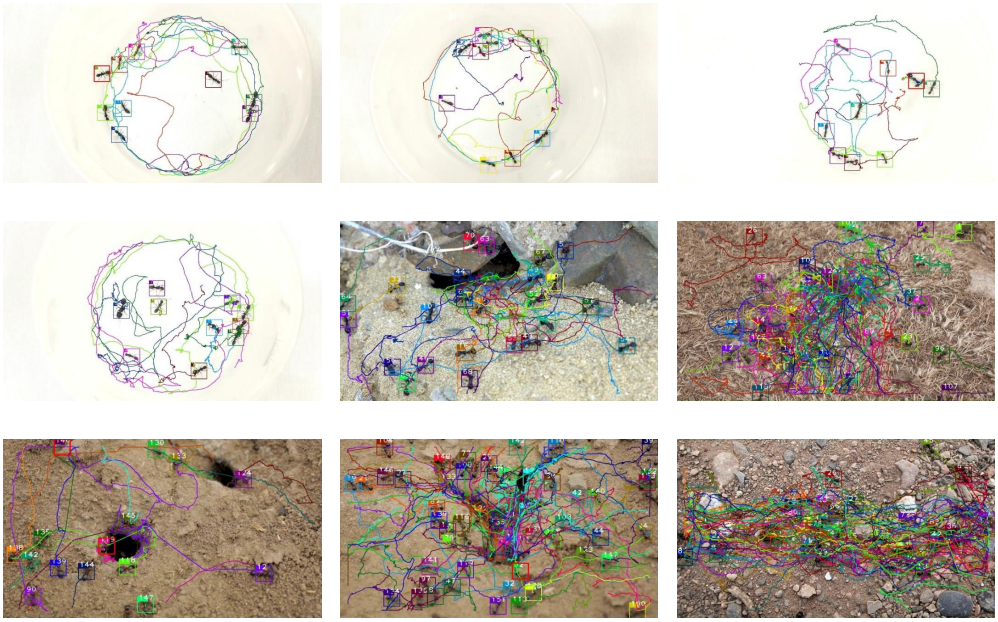


Fig 1. Tracking results by our method in both indoor and outdoor environments.

Results & Discussions

Ant colony database

We establish an video database of ant colony, which contains a total of 10 videos. Five videos are from an existing published work [28] and captured in the indoor (laboratory) environment. The remaining five outdoor videos are captured in different backgrounds and are obtained from the online website *DepositPhotos* (http://www.depositphotos.com). Table 1 shows detailed video information, where $\mathcal{I}$ represents an indoor video, $\mathcal{O}$ represents an outdoor video. The resolutions of indoor and outdoor videos are 1920*1080 and 1280*720, respectively.

Sequence	Frames	Objects	Sequence	Frames	Objects
$\mathcal{I}_1$	351	10	$\mathcal{O}_1$	600	18
$\mathcal{I}_2$	351	10	$\mathcal{O}_2$	677	37
$\mathcal{I}_3$	351	10	$\mathcal{O}_3$	557	19
$\mathcal{I}_4$	1001	10	$\mathcal{O}_4$	526	53
$\mathcal{I}_5$	351	10	$\mathcal{O}_5$	569	38

Table 1. Statistics of indoor $\mathcal{I}$ and outdoor $\mathcal{O}$ videos.

The videos in our database have a total of 4983 frames. There are 10 ants per frame in the indoor videos. The number of ants in each frame is 18-53 in the outdoor videos. The number of objects in this scenario is significant, considering the fact that the popular COCO benchmark dataset contains only on average 7.7 instances per image.

Some video characteristics present challenges for detection and tracking algorithms, for example over-exposure for indoor videos and diverse background for outdoor ones.

There are caves or rugged terrains in outdoor scenes, and ants may enter or leave the scene. Different from multi-human tracking, ants are visually similar and this causes significant challenges for tracking. We manually mark the video frame by frame. To facilitate training and reduce labeling cost, the aspect ratio of each bounding box is 1:1. Considering the posture and scale of ants, we set the size of the bounding box to 96*96 for indoor videos and 64*64 for outdoor videos. The database and code will be made publicly available.

Evaluation index

In this paper, the evaluation indicators of detection and tracking performance are as follows:

- Mean Average Precision (MAP): the weighted sum of the average precision of all videos. The weight value is the proportion of frames.
- False Positive (FP): the total number of false alarms.
- False Negative (FN): the total number of objects that do not match successfully.
- Identity Switch (IDS): the total number of identity switches during the tracking process.
- Fragments (FM): the total number of incidents where the tracking result interrupts the real trajectory.
- mean Multi-object Tracking Accuracy (mMOTA): the weighted sum of the average tracking accuracy of all videos. The equation to compute mMOTA is: mMOTA = 1 - (FP + FN + IDS)/NUM_LABELED_SAMPLES, where NUM_LABELED_SAMPLES is the total number of labeled samples.
- mean Multi-object Tracking Precision (mMOTP): the weighted sum of the average tracking precision of all videos. Tracking precision measures the intersection over union (IOU) between labeled and predicted bounding boxes.
- Frame Rate (FR): the number of frames being tracked per second.

Results of multi-ant detection

In our ant database, we set up five groups of training sets (Table 2) and compare their performance with that of the remaining datasets. The naming conventions are:

- $\mathcal{I}_5$+$\mathcal{O}_4$ represents a union of the 5^{th} indoor video and the 4^{th} outdoor video.
- $\mathcal{I}_{1-4}$ represents a union of indoor videos with their IDs of [1,2,3,4].
- $\mathcal{I}_5\,(50)$ represents the last 50 frames selected from the 5^{th} indoor video. This partition strategy ensures the frame continuity for the subsequent tracking task.
- $\mathcal{O}_{1-5}\,(-50)$ represent the union of 5 subsets, the last 50 frames de-selected from the outdoor videos with their IDs of [1,2,3,4,5].

In all scenarios, the detection accuracy of indoor videos is higher than that of outdoor videos, and MAP reaches over 90%. We also noticed that the test result for outdoor videos was only 49.7% on $\mathcal{I}_5$+$\mathcal{O}_4$. This is because we used only $\mathcal{O}_4$ as the

Training Data	Testing Data	Objects	MAP↑	FR↑
$\mathcal{I}_5$+$\mathcal{O}_4$	$\mathcal{I}_{1-4}$	10	90.4	**12.5**
	$\mathcal{O}_{1-3,5}$	28	49.7	16.0
$\mathcal{I}_5$ (50)+$\mathcal{O}_{1-5}$ (50)	$\mathcal{I}_{1-4}$	10	90.4	12.2
	$\mathcal{O}_{1-5}$ (−50)	33	81.9	**17.1**
$\mathcal{I}_5$+$\mathcal{O}_{1-5}$ (100)	$\mathcal{I}_{1-4}$	10	90.4	12.3
	$\mathcal{O}_{1-5}$ (−100)	33	82.4	16.6
$\mathcal{I}_5$+$\mathcal{O}_{1-5}$ (200)	$\mathcal{I}_{1-4}$	10	90.5	12.3
	$\mathcal{O}_{1-5}$ (−200)	33	85.1	16.6
$\mathcal{I}_5$+$\mathcal{O}_{1-5}$ (300)	$\mathcal{I}_{1-4}$	**10**	**90.5**	11.8
	$\mathcal{O}_{1-5}$ (−300)	**33**	**85.8**	16.2

Table 2. Detection results of different training sets.

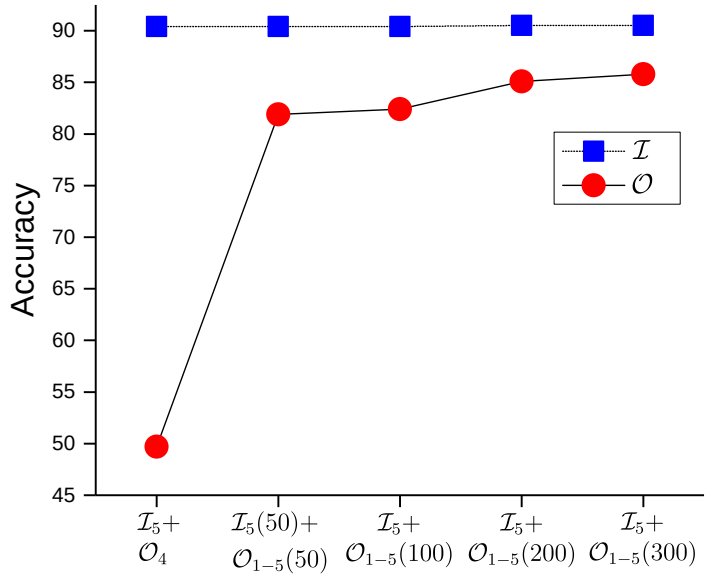


Fig 2. Detection accuracy of different training sets.

outdoor training set, which is insufficient to cover the wide range of diversity in terms of environmental backgrounds and ant appearances.

In the subsequent experiments, we integrate the images of all outdoor scenes into the outdoor training set and dramatically improve the accuracy of outdoor testing. Fig 2 clearly shows the effects of using different training sets. By further increasing in the number of images in outdoor videos, the detection accuracy of outdoor scenes improves slightly. For indoor environments, the detection accuracy is impervious to different training sets. Moreover, reducing the number of images to 50 ($\mathcal{I}_5$ has a total of 351 frames) does not reduce the detection accuracy. This shows that we need only a small number of training samples to achieve satisfactory results when the training and testing scenarios are the same.

The frame rate is around 12 FR for indoor videos and 16 FR for outdoor ones. The factor of different image resolution should be accountable for this performance gap. In practical applications, if accuracy is guaranteed, we tend to use smaller training sets to reduce labeling costs. Therefore, we use the model trained in "$\mathcal{I}_5$ (50)+$\mathcal{O}_{1-5}$ (50)" for comparison with the other methods in the comparative experiments.

Results of multi-ant tracking

Based on the TBD paradigm, we use detection results as the input to the tracking framework. For offline training, we randomly select 50 labeled samples from $\mathcal{I}_5$ as the training set. We visualized the tracking results in Fig 3.

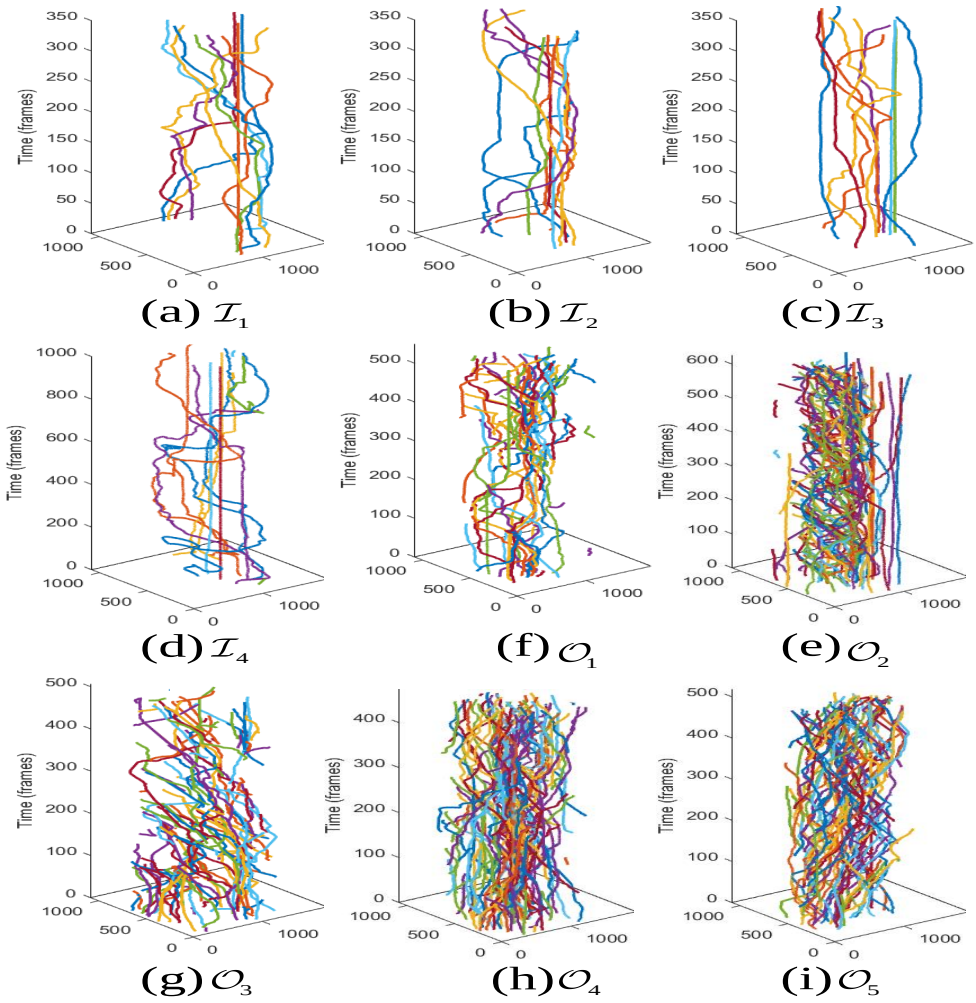


Fig 3. Tracking trajectories in test videos. Horizontal axes indicate the pixel coordinates in an image. (a-d) indoor scenes. (e-i) outdoor scenes.

Table 3 shows the performance of online tracking. After integrating the images of each outdoor video in the detection training set, our method gets 95% mMOTA for indoor videos and over 80% for outdoor videos. Additionally, mMOTP is around 80% for both indoor and outdoor videos. Notably, since the tracking performance depends on the detection result, the tracking task in $\mathcal{O}_{1-3,5}$ fails due to the low-quality detection (the second row in Table 3). Except for this failure case, the tracking performance is generally satisfactory considering that we only use 50 labeled samples from one indoor video.

The time cost of the tracking model is mainly incurred by generating 128-d feature vectors for each detection box. The average number of objects in outdoor videos is more than three times that in indoor videos. As for runtime time, FR reaches over 35 in indoor videos and more than 24 in outdoor videos.

Testing Data	FP↓	FN↓	IDS↓	mMOTA↑	mMOTP↑	FR↑
$\mathcal{I}_{1-4}$	236	621	21	89.9	79.9	36.5
$\mathcal{O}_{1-3,5}$	detection failure					
$\mathcal{I}_{1-4}$	239	628	22	**95.7**	81	**38.9**
$\mathcal{O}_{1-5}\,(-50)$	6078	7122	625	81.8	81.9	26.2
$\mathcal{I}_{1-4}$	260	**617**	**14**	95.7	81	38.7
$\mathcal{O}_{1-5}\,(-100)$	4867	6018	526	83.3	82.7	**28.0**
$\mathcal{I}_{1-4}$	228	709	17	95.4	81	36.3
$\mathcal{O}_{1-5}\,(-200)$	4289	3421	394	85.3	83	24.9
$\mathcal{I}_{1-4}$	**224**	820	24	94.8	**81.7**	35.4
$\mathcal{O}_{1-5}\,(-300)$	**2644**	**3007**	**266**	**85.8**	**83.3**	26.4
$\mathcal{I}_{1-4}$/GT	22	23	8	99.6	92.4	35.2
$\mathcal{O}_{1-5}\,(-50)$/GT	1697	458	1064	96.2	92.4	25.9

Table 3. Tracking performance evaluation. The last two rows indicate that we use the ground truth of detection for tracking, which leads to a boost in tracking performance.

We add a set of comparative experiments in the last two rows of Table 3. We directly use manually-labeled detection boxes for tracking and compare the detection results on the $\mathcal{I}_{1-4}$ and $\mathcal{O}_{1-3,5}$. Both mMOTA and mMOTP have been dramatically improved. This implies that an increase in detection accuracy could further boost the tracking performance of our framework.

Comparative experiments

There are two widely used insect tracking software: idTracker [21] and Ctrax [22]. idTracker needs to specify the number of objects before tracking, to create a reference image set for each object. Meanwhile, Ctrax assumes that objects will rarely enter and leave the arena. Thus, they are both not capable of tracking in outdoor scenes because of the variable number of ants. Therefore, we compare these two methods only in videos depicting indoor scenes. idTracker needs to specify the number of objects before tracking, in order to create a reference image set for each object. To compare them with our method, we convert their representations into square boxes as our ground truth. Table 4 shows the tracking results. In addition to a significant improvement of tracking accuracy, our method is 6 and 10 times faster than idTracker and Ctrax (see the column of FR).

Method	FP↓	FN↓	IDs↓	FM↓	mMOTA↑	mMOTP↑	FR↑
idTracker	881	8479	83	432	54	77.4	1.3
Ctrax	2832	5646	110	349	58.2	79.7	0.8
Ours	**239**	**628**	**22**	**189**	**95.7**	**81.1**	**8.7**

Table 4. Comparison of tracking results on videos $\mathcal{I}_{1-4}$.

idTracker uses the intensity and contrast of the foreground segmented area to extract appearance features and construct a reference image set for each individual. However, it can not track motionless individuals. Fig 4(a) shows that only a minority group of ants are successfully tracked over the period of video. Further, there are some trajectory fragments due to the limitations of the foreground segmentation model for multiple objects.

Compared to our results, the trajectories of Ctrax are incomplete. This indicates that there are more FN, as Fig 4(b) shows. Ctrax requires a sharp contrast between

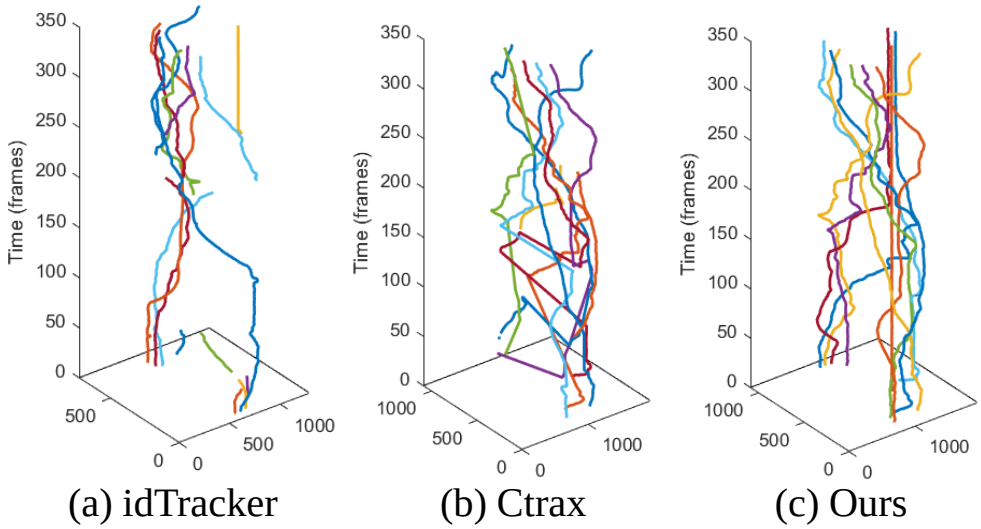


Fig 4. Comparison of tracking performance in spatial-temporal dimension ($\mathcal{I}_1$). Horizontal axes indicate the pixel coordinates in an image.

object and background. The ants passing through the overexposed areas in the scene will be ignored. Additionally, Ctrax assumes that the motion of the object obeys the linear distribution. However, the ants' movement is nonlinear, and their speed and direction might change abruptly, causing IDS in Ctrax.

Our method classifies and regresses twice to locate ants accurately. During the tracking stage, we use the historical appearance sequence as a reference and update it frame by frame. Compared with idTracker, our method effectively solves the long-term and short-term dependence of motion states, thereby reducing FM. Despite that we also assume the linear distribution of motion states, they are used only to filter impossible associations, and have nothing to do with association cost. We take the appearance distance between trajectories and detection boxes as association cost, thus the model is robust even when the ant movement is complicated. We take the appearance measure between trajectories and detection boxes as the association cost, thus the model is robust even when the ant movement is complicated.

We further compare the tracking accuracy of idTracker and Ctrax across different indoor videos, as Fig 5 shows. The large variance of idTracker's performance is affected by the number of static ants, which this method fails to track. Ctrax proves to be robust but with a lower accuracy compared with our method.

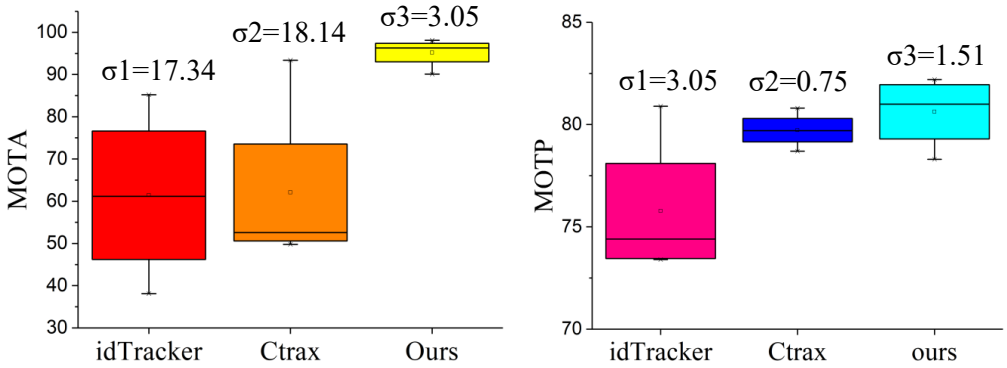
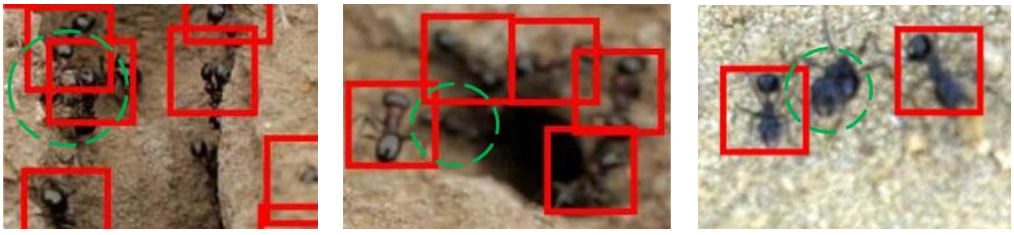


Fig 5. Comparison of tracking results for indoor scenes.

Failure cases

Limitations of detection framework

The number of ants in outdoor scenes is on average 33 per frame. It is also typical for ants to involve close body contact with each other for the purpose of information sharing. Naturally, their extremely-close interactions are highly likely to cause mis-detection (Fig 6(a)). Additionally, entrances and exits of ants in outdoor scenes are more prone to mis-detection (Fig 6(b)). Moreover, the dramatically non-rigid deformation of ants is also a factor causing the detection failure (Fig 6(c)). These three scenarios are all challenging cases that deserve our future efforts.



(a) Interaction (b) Enter the scene (c) Non-rigid warp

Fig 6. Examples of failed detection in outdoor scenes.

Limitations of tracking framework

According to Fig 7, Ant No.41 entered the scene at Frame No.88. Coincidentally Ant No.32 left the scene at an adjacent region, but its trajectory was not deleted. At Frame No.93, Ant No.41 drifted to Trajectory No.32. This defect is caused by insufficient appearance descriptors stored in the gallery of Ant No.41, and it moved near the exit location of another ant. This kind of mis-association occurs at the image boundary and accounts for the majority of IDS and FM in our experiments. However, when ants move inside the scope of both indoor and outdoor scenes, our method can accurately track multiple ants simultaneously for a long time, as Fig 1 shows.

Fig 7. Drift at the scene boundary. A newly-entered Ant No.41 is mis-associated with an existing Trajectory No.32.

Materials and methods

Overview

Following the TBD paradigm, we propose a uniform framework for detection and tracking to efficiently and accurately track the ant colony in both indoor and outdoor scenes (Fig 8). In the detection phase, we adopt a two-stage object detection framework, using ResNet-50 as the backbone, and encoding RoIs proposed by RPN via

position-sensitive score maps. Then we implement classification and regression through downsampling and voting mechanisms. (see details in Section Two-stage object detection). In the tracking stage, we first use ResNet to train the appearance descriptors of ants and measure the appearance similarity between two objects. Next, the tracking is accomplished by combining appearance and motion information for online association metric. (Section MOT framework).

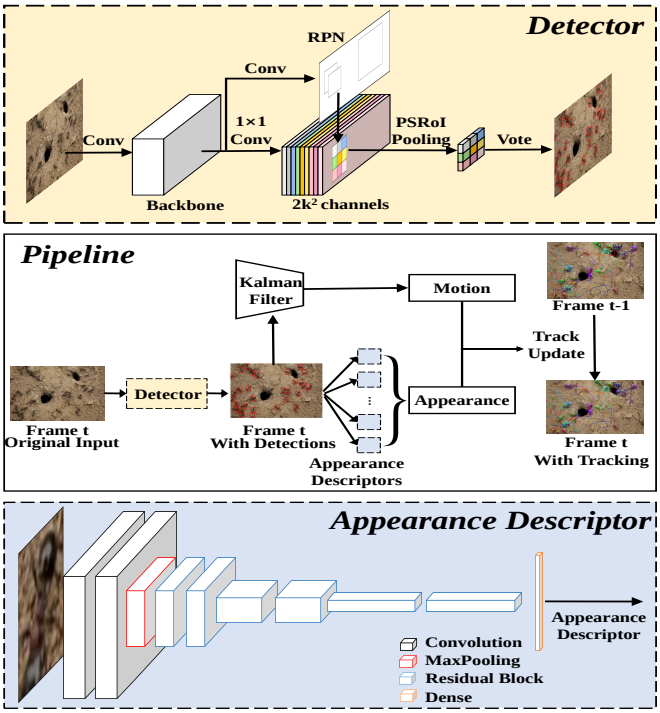


Fig 8. Architecture for detection and tracking.

Two-stage object detection

RPN

RPN is proposed in Faster R-CNN [11] to generate RoIs. Compared to SS [9], RPN is based on the CNN network structure and can connect the backbone with shared weight, significantly improving detection speed. We use ResNet-50 as the backbone and replace the fully connected layer with a 1*1 convolution to reduce the dimensions of feature maps. Considering that ResNet-50 conducts downsampling 32 times, we get 256-d feature maps via a 3*3 Atrous convolution to maintain translation variability. For each sliding position, we predict k region proposal boxes of different sizes and ratios; these boxes are called anchors. After the 256-d vector, we connect classification and regression branches through two parallel 1*1 convolution layers. The classification branch uses softmax to determine whether there is an object in anchor so that this branch has 2*k outputs. The regression branch will perform a regression on the 4D position parameters of anchors (i.e., center coordinates, width and height) so that there are 4*k outputs. RPN will propose k*w*h anchors with a w*h feature map, called RoIs. We use the Non-maximum suppression algorithm [29] to filter duplicate anchors and set the IOU threshold to 0.7.

PSRoI-based detection

On the basis of RoIs, the two-stage detection framework classifies and fine-tunes the location of bounding boxes. In Faster R-CNN, RoIs are scaled to the last feature maps and focusing on these areas through ROIPooling. Next, each RoI is classified and regressed through two fully connected layers, causing high computational complexity.

In order to reduce the number of parameters, we use RPN-FCN [16] to generate position-sensitive score maps via a convolutional layer, which is connected to the backbone. Both classification and regression tasks have independent position-sensitive score maps, forming three parallel branches with RPN.

For the classification task, since we only need to classify ants and background, we use k*k*2 convolution kernels to generate score maps. k*k indicates that each RoI is divided into k*k regions to encode position information. Each region is encoded by a specific feature map with two dimensions. Similarly, we use k*k*4 convolution kernels for fine-tuning the position of RoIs in the regression task.

To focus on RoIs, we perform average pooling on each region to get feature maps, called position sensitive region of interest (PSRoI) pooling, as the following formula shows:

$$r_c(i,j|\Theta) = \sum_{(x,y)\in \text{region}(i,j)} z_{i,j,c}\left(x+x_0, y+y_0|\Theta\right)/n. \tag{1}$$

$r_c(i,j|\Theta)$ is the result of downsampling in $(i,j)^{th}$ for c^{th} category, and $z_{i,j,c}$ is one score map in the k*k*2 position-sensitive score maps. $(x0, y0)$ represents the left-top corner of RoI. Θ is the set of parameters of the network, and n is the number of pixels in the region.

For the feature maps, we vote on k*k regions, getting the overall score of RoI on the classification or regression task, as the following formula shows:

$$r_c(\Theta) = \sum_{i,j} r_c(i,j|\Theta). \tag{2}$$

In the formula, $r_c(\Theta)$ represents the overall scores of all regions.

Next, we use softmax to implement binary classification, as the following formula shows:

$$s_c(\Theta) = e^{r_c(\Theta)} / \sum_{c=0}^{C} e^{r_c(\Theta)}. \tag{3}$$

Here, $s_c(\Theta)$ is the probability of c^{th} category. Finally, we use the Non-maximum suppression algorithm to filter the bounding box.

Since object detection includes classification and regression, we require a multitask loss function. In this paper, we weight the loss functions of the two tasks. Because softmax is used for the binary classification task, it is natural to adopt cross-entropy loss for the classification task. For the regression task, we calculate the matching degree between the four position parameters and ground truth:

$$L\left(s, t_{x,y,w,h}\right) = L_{cls}\left(s_{c^*}\right) + \lambda\left[c^* = 1\right] L_{reg}\left(t, t^*\right). \tag{4}$$

where c^* is the ground truth category label of RoI, and $c^* = 1$ represents ants. $L_{cls}\left(s_{c^*}\right)$ represents cross-entropy loss:

$$L_{cls}\left(s_c\right) = -\log\left(s_c\right). \tag{5}$$

$L_{reg}\left(t, t^*\right)$ represents the loss of the regression task, including 4 dimensions:

$$L_{reg}\left(t, t^*\right) = \Sigma_{i=1}^{4} g\left(t_i^* - t_i\right). \tag{6}$$

In the formula, t^* is the predicted position, and t is ground truth after translation and scaling.

MOT framework

Offline ResNet network architecture

We adopt a 15-layer ResNet network architecture to extract the appearance descriptors of objects, as Fig 8 shows. After downsampling eight times, the network will eventually obtain a 128-dimensional feature vector through a fully connected layer. The specific parameters are consistent with [28].

Cosine similarity metric classifier

We modify the parameters of softmax to get a cosine similarity measurement classifier, which can measure the similarity of the same category or different categories. First, the output of a fully connected layer is normalized by batch normalization, ensuring that it is expressed as a unit length $\|f_\Theta(x)\|_2 = 1, \forall x \in R^D$. Second, we normalize the weights, that is, $\varpi_k = \omega / \|\omega_k\|_2, \quad \forall k = 1, \ldots C$. Cosine similarity metric classifier is constructed as follows:

$$p\left(y_i = k | r_i\right) = \frac{\exp\left(\kappa \cdot \varpi_k^T r_i\right)}{\sum_{n=1}^{C} \exp\left(\kappa \cdot \varpi_n^T\right)}. \tag{7}$$

Here, κ is the free scaling parameter.

Because the cosine similarity classifier follows the structure of softmax, we use the cross-entropy loss for training:

$$L(D) = -\sum_{i=1}^{N}\sum_{k=1}^{C} \mathrm{gt}_{y_i - k} \cdot \log p\left(y_i = k | r_i\right). \tag{8}$$

Here, $L(D)$ represents the sum of the cross-entropy loss of N images, $p\left(y_i = k | r_i\right)$ is the prediction result of i^{th} image in k^{th} label, and $\mathrm{gt}_{y_i - k}$ is ground truth.

Motion matching

We use the KF model to predict the position of trajectories in the current frame. Then, we calculate the square of the Mahalanobis distance between the predicted position and the detected bounding box position by measuring the degree of motion matching [30] as follows:

$$d^{(1)}(i,j) = \left(d_j - y_i\right)^T S_i^{-1} \left(d_j - y_i\right). \tag{9}$$

Here, d_j is the position of the j^{th} detection box, y_i is the position of the i^{th} trajectory predicted by the KF, and S_i is the covariance matrix between the i^{th} trajectory and the detected bounding box.

We use a 0-1 variable to indicate whether trajectory and detection meet the association conditions. If the Mahalanobis distance meets $t^{(1)}$, (i,j) will be added to the association set. The formula can be expressed as:

$$b_{ij}^{(1)} = \begin{cases} 1, & d^{(1)}(i,j) < t^{(1)} \\ 0, & \text{otherwise} \end{cases}. \tag{10}$$

Here, $b_{ij}^{(1)}$ is the motion association signal.

Appearance matching

We use the appearance descriptors to measure the appearance similarity between ants. Furthermore, we create a gallery for each trajectory, and each gallery stores the latest 100 appearance descriptors. Then, we calculate the cosine distance of appearance descriptors between gallery and candidate bounding boxes. The smallest distance is used as an appearance matching degree as follows:

$$d^{(2)}(i,j) = \min\left\{1 - r_j^T r_k^{(i)} | r_k^{(i)} \in \mathrm{K}_i\right\}. \tag{11}$$

where r_j is the appearance descriptor of the j^{th} detection box, $r_k^{(i)}$ is the k^{th} appearance descriptor of the i^{th} trajectory, $d^{(2)}(i,j)$ represents the appearance matching degree between the i^{th} trajectory and the j^{th} bounding box.

Similarly, we introduce a 0-1 variable as an association signal. If the appearance matching degree from a pair of trajectory and detection boxes meets the threshold, we add it to the association set:

$$b_{ij}^{(2)} = \begin{cases} 1, & d^{(2)}(i,j) < t^{(2)} \\ 0, & \text{otherwise} \end{cases}. \tag{12}$$

where $b_{ij}^{(2)}$ represents the appearance association signal. In this paper, $t^{(2)}$ is set to 0.2.

Comprehensive matching

To combine motion and appearance information, we set a comprehensive association signal b_{ij}. Only when both motion and appearance matching degree meet the threshold, the (i,j) pair will be considered for matching. The formula expression is denoted as follows:

$$b_{ij} = \prod_{m=1}^{2} b_{i,j}^{(m)}. \tag{13}$$

However, the KF is scarcely possible to track accurately for long periods, because of the motion of ants is complicated. Therefore, we use the appearance matching degree (Section Appearance matching) as the association cost.

Track update

First, we use matching cascade to match in priority for the most recently associated trajectories, avoiding the trajectory drift caused by long-term occlusion [30]. During the matching, we use the Hungarian algorithm to find the minimum cost matches in the association cost matrix. For unmatched trajectories and detection boxes, we calculate the IOU. If they meet the threshold, they are associated.

After that, trajectories need to be updated. They have three states: unconfirmed, confirmed, and deleted. We assign a new trajectory for each unmatched detection box. Furthermore, if the duration of trajectory is less than three, it will be set to an unconfirmed state. The unconfirmed trajectories need to be successfully associated for three consecutive frames before being converted into confirmed state; otherwise, they will be deleted.

For the unmatched confirmed trajectories, if they are successfully matched in the previous frame, we will use the KF to estimate and update their motion state in the current frame; otherwise, we will suspend tracking. Moreover, if the number of consecutively lost frames of confirmed trajectories exceeds the threshold (Amax=30), they will be deleted.

Conclusion

We proposed a complete detection and tracking framework based on deep learning for ant colony tracking. In the detection stage, we adopted a two-stage object detection framework for the detection task. We also use a ResNet model to obtain ant appearance descriptors for online associations. Next, we combined appearance and motion information for the tracking task. The experimental results demonstrated that our method outperformed two mainstream insect tracking models in terms of accuracy, precision, and speed. Particularly, our work shows its advantage in robustly detecting and tracking ant colonies in outdoor scenes, which is rarely reported in existing literature. We believe our method could serve as an effective tool for high-throughput quantitative behavior analysis of ant colony for biologists.

In future research, we aim to achieve more robust detection. For example, by exploring additional information of ants' skeletal structure, we can potentially solve the aforementioned failure case of close interaction and nonrigid deformation problem. We also plan to improve the generalization ability of our detection and tracking frameworks so that it is applicable to a wide range of outdoor environments.

Acknowledgments

This work was supported by the Natural Science Foundation of Fujian Province of China (No. 2019J01002)

References

1. Evans TA, Dawes TZ, Ward PR, Lo N. Ants and termites increase crop yield in a dry climate. Nature communications. 2011;2(1):1–7.
2. Fasciano T, Nguyen H, Dornhaus A, Shin MC. Tracking multiple ants in a colony. In: 2013 IEEE Workshop on Applications of Computer Vision (WACV). IEEE; 2013. p. 534–540.
3. Poff C, Nguyen H, Kang T, Shin MC. Efficient tracking of ants in long video with GPU and interaction. In: 2012 IEEE Workshop on the Applications of Computer Vision (WACV). IEEE; 2012. p. 57–62.
4. Fasciano T, Dornhaus A, Shin MC. Ant tracking with occlusion tunnels. In: IEEE Winter Conference on Applications of Computer Vision. IEEE; 2014. p. 947–952.
5. Nguyen H, Fasciano T, Charbonneau D, Dornhaus A, Shin MC. Data association based ant tracking with interactive error correction. In: IEEE Winter Conference on Applications of Computer Vision. IEEE; 2014. p. 941–946.
6. Shen M, Li C, Huang W, Szyszka P, Shirahama K, Grzegorzek M, et al. Interactive tracking of insect posture. Pattern Recognition. 2015;48(11):3560–3571.
7. Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 779–788.
8. Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2014. p. 580–587.

9. Uijlings JR, Van De Sande KE, Gevers T, Smeulders AW. Selective search for object recognition. International journal of computer vision. 2013;104(2):154–171.

10. Girshick R. Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision; 2015. p. 1440–1448.

11. Ren S, He K, Girshick R, Sun J. Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems; 2015. p. 91–99.

12. Geuther BQ, Deats SP, Fox KJ, Murray SA, Braun RE, White JK, et al. Robust mouse tracking in complex environments using neural networks. Communications biology. 2019;2(1):1–11.

13. Murali N, Schneider J, Levine J, Taylor G. Classification and Re-Identification of Fruit Fly Individuals Across Days With Convolutional Neural Networks. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE; 2019. p. 570–578.

14. Romero-Ferrero F, Bergomi MG, Hinz RC, Heras FJ, de Polavieja GG. Idtracker. ai: tracking all individuals in small or large collectives of unmarked animals. Nature methods. 2019;16(2):179–182.

15. Sridhar VH, Roche DG, Gingins S. Tracktor: Image-based automated tracking of animal movement and behaviour. Methods in Ecology and Evolution. 2019;10(6):815–820.

16. Dai J, Li Y, He K, Sun J. R-fcn: Object detection via region-based fully convolutional networks. In: Advances in neural information processing systems; 2016. p. 379–387.

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

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

19. Fletcher M, Dornhaus A, Shin MC. Multiple ant tracking with global foreground maximization and variable target proposal distribution. In: 2011 IEEE Workshop on Applications of Computer Vision (WACV). IEEE; 2011. p. 570–576.

20. Branson K, Robie AA, Bender J, Perona P, Dickinson MH. High-throughput ethomics in large groups of Drosophila. Nature methods. 2009;6(6):451.

21. Pérez-Escudero A, Vicente-Page J, Hinz RC, Arganda S, De Polavieja GG. idTracker: tracking individuals in a group by automatic identification of unmarked animals. Nature methods. 2014;11(7):743.

22. Chiron G, Gomez-Krämer P, Ménard M. Detecting and tracking honeybees in 3D at the beehive entrance using stereo vision. EURASIP Journal on Image and Video Processing. 2013;2013(1):59.

23. Li Y, Huang C, Nevatia R. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE; 2009. p. 2953–2960.

24. Khan Z, Balch T, Dellaert F. MCMC data association and sparse factorization updating for real time multitarget tracking with merged and multiple measurements. IEEE transactions on pattern analysis and machine intelligence. 2006;28(12):1960–1972.

25. Kim C, Li F, Rehg JM. Multi-object tracking with neural gating using bilinear lstm. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. p. 200–215.

26. Zhu J, Yang H, Liu N, Kim M, Zhang W, Yang MH. Online multi-object tracking with dual matching attention networks. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. p. 366–382.

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

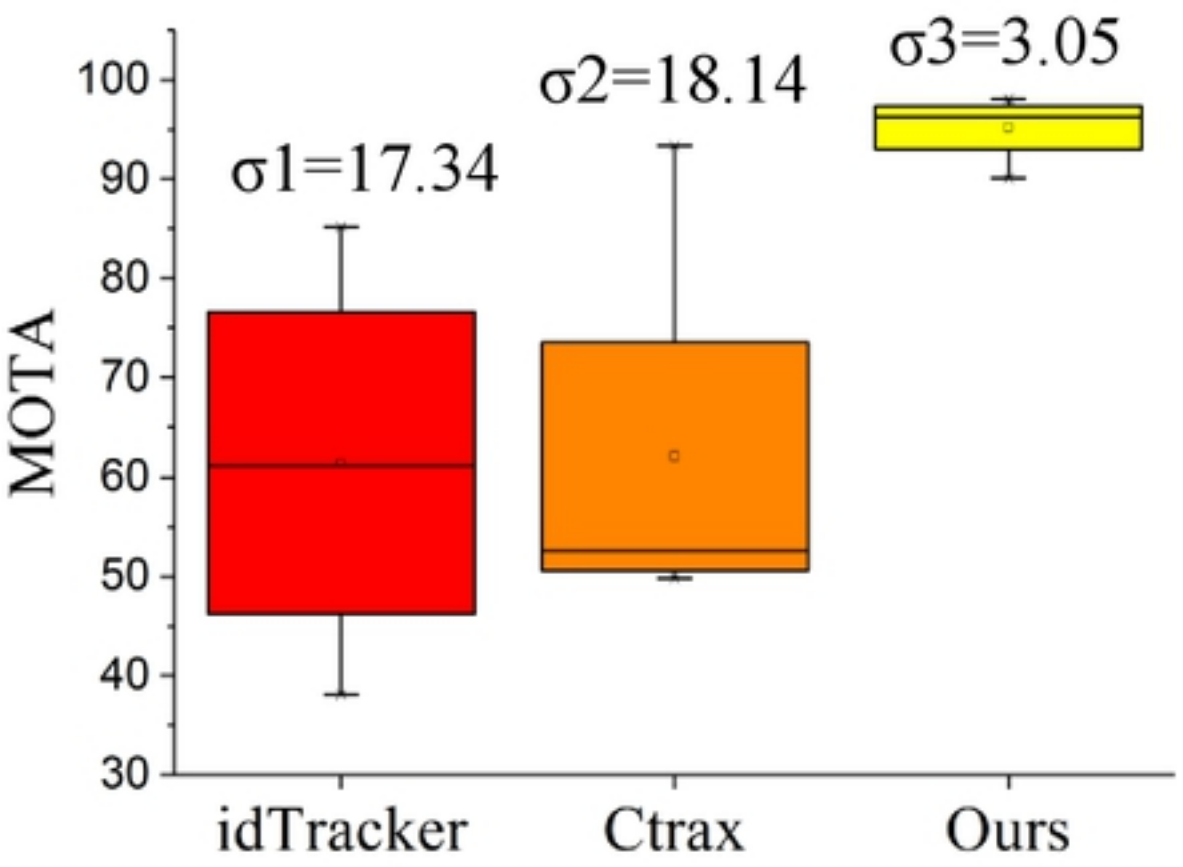
28. Cao X, Guo S, Lin J, Zhang W, Liao M. Online Tracking of Ants Based on Deep Association Metrics: Method, Dataset and Evaluation. Pattern Recognition. 2020;.

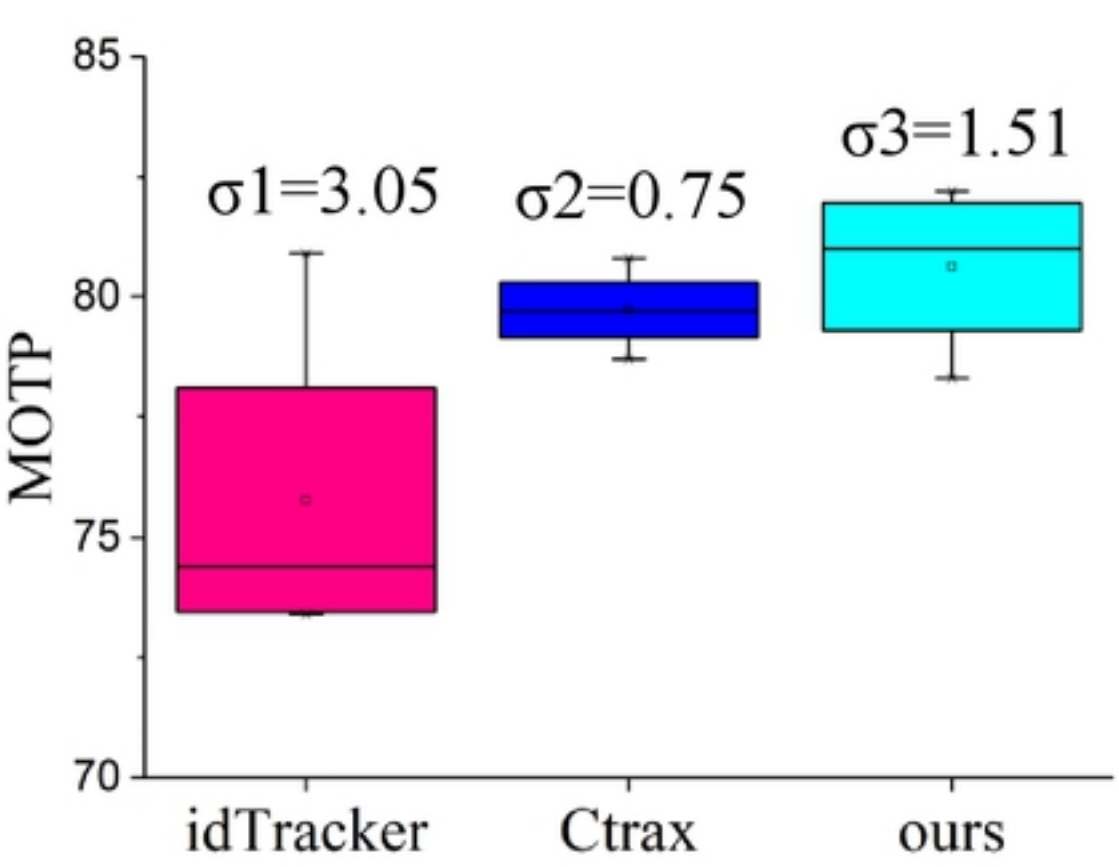
29. Neubeck A, Van Gool L. Efficient non-maximum suppression. In: 18th International Conference on Pattern Recognition (ICPR'06). vol. 3. IEEE; 2006. p. 850–855.

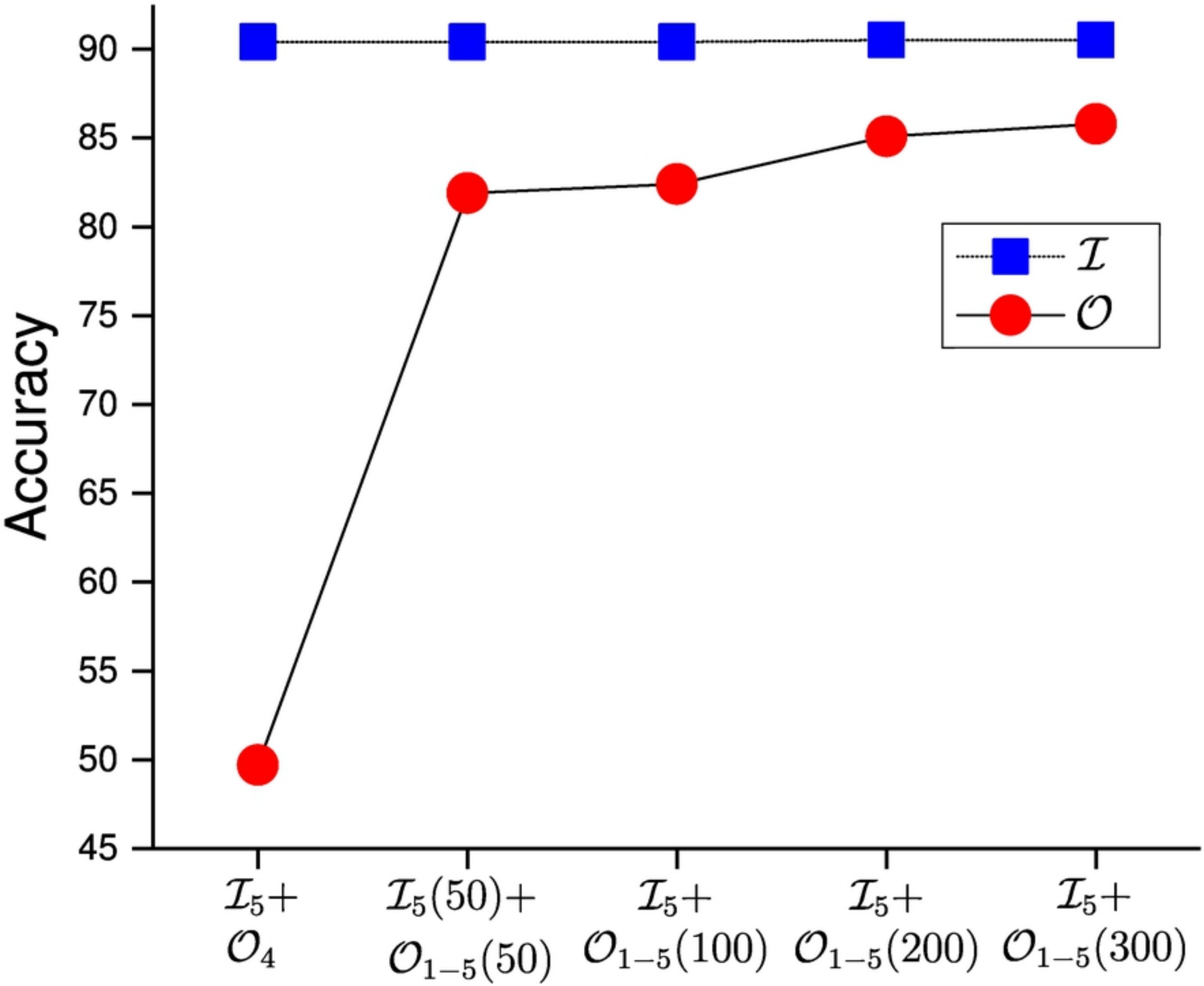
30. Wojke N, Bewley A, Paulus D. Simple online and realtime tracking with a deep association metric. In: 2017 IEEE International Conference on Image Processing (ICIP). IEEE; 2017. p. 3645–3649.







σ1=17.34
σ2=18.14
σ3=3.05
MOTA
100
90
80
70
60
50
40
30
idTracker
Ctrax
Ours


σ1=3.05
σ2=0.75
σ3=1.51
MOTP
85
80
75
70
idTracker
Ctrax
ours

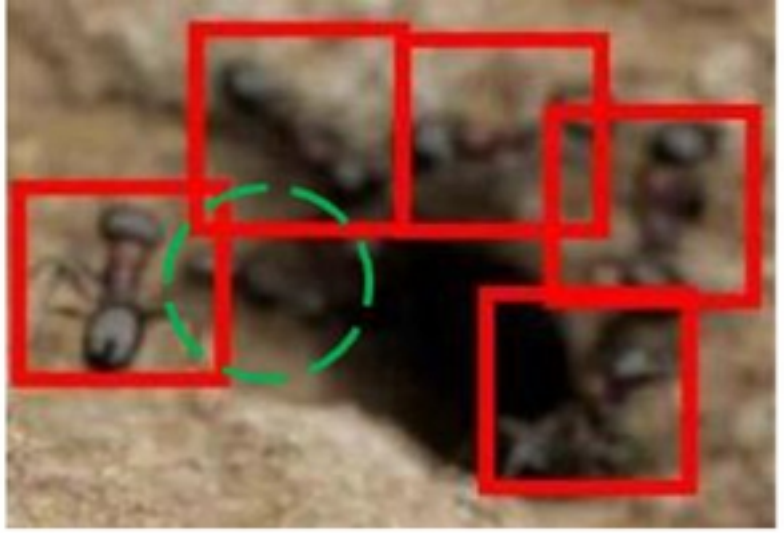

Accuracy
90
85
80
75
70
65
60
55
50
45
$\mathcal{I}$
$\mathcal{O}$
$\mathcal{I}_5+$ $\mathcal{O}_4$
$\mathcal{I}_5(50)+$ $\mathcal{O}_{1-5}(50)$
$\mathcal{I}_5+$ $\mathcal{O}_{1-5}(100)$
$\mathcal{I}_5+$ $\mathcal{O}_{1-5}(200)$
$\mathcal{I}_5+$ $\mathcal{O}_{1-5}(300)$


Outdoor5
#88
41

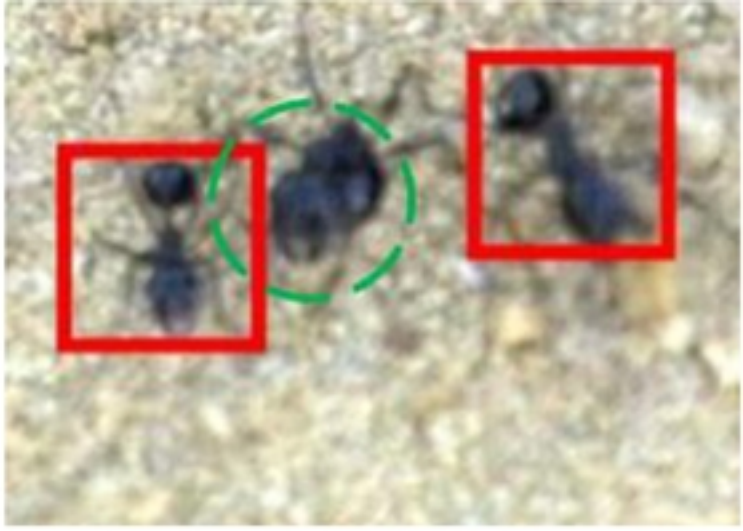
Outdoor5
#91
41

Outdoor5
#93
32

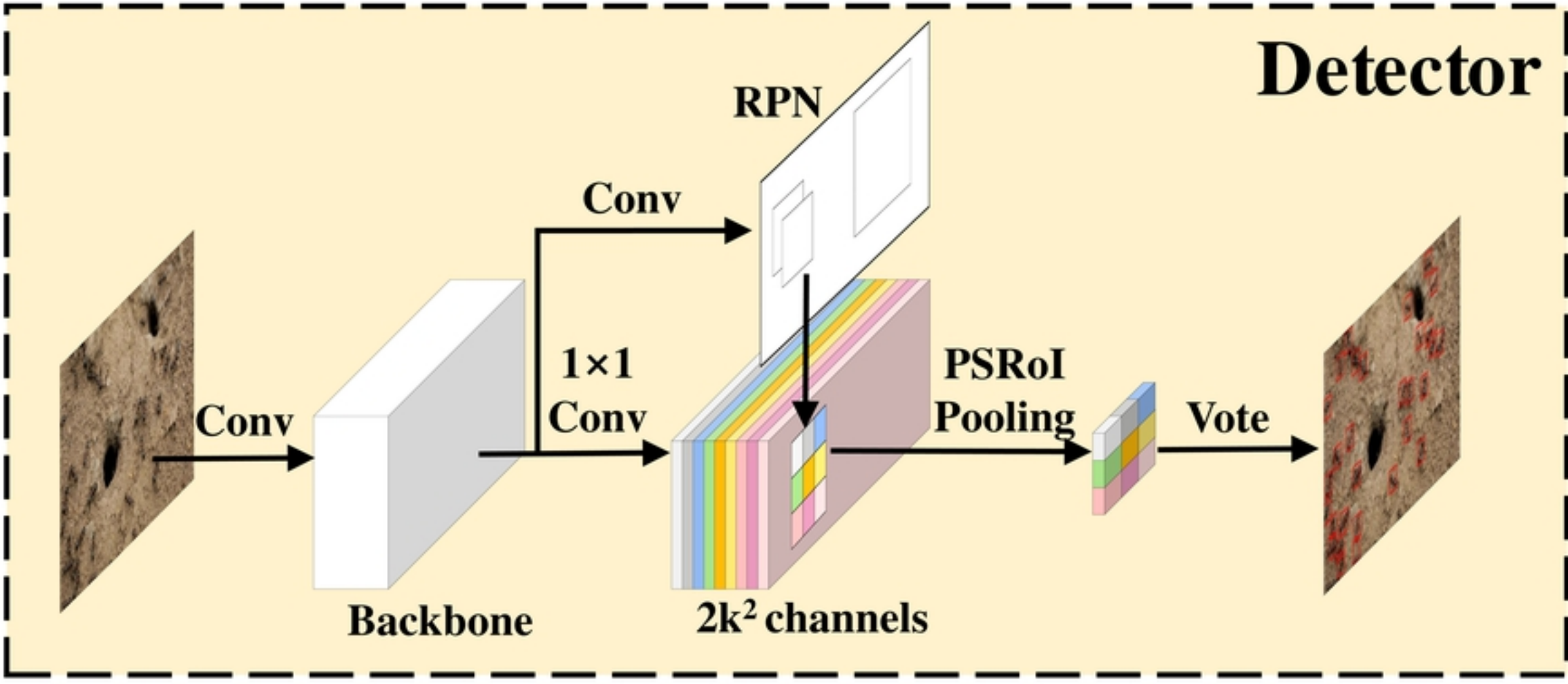
(a) Interaction



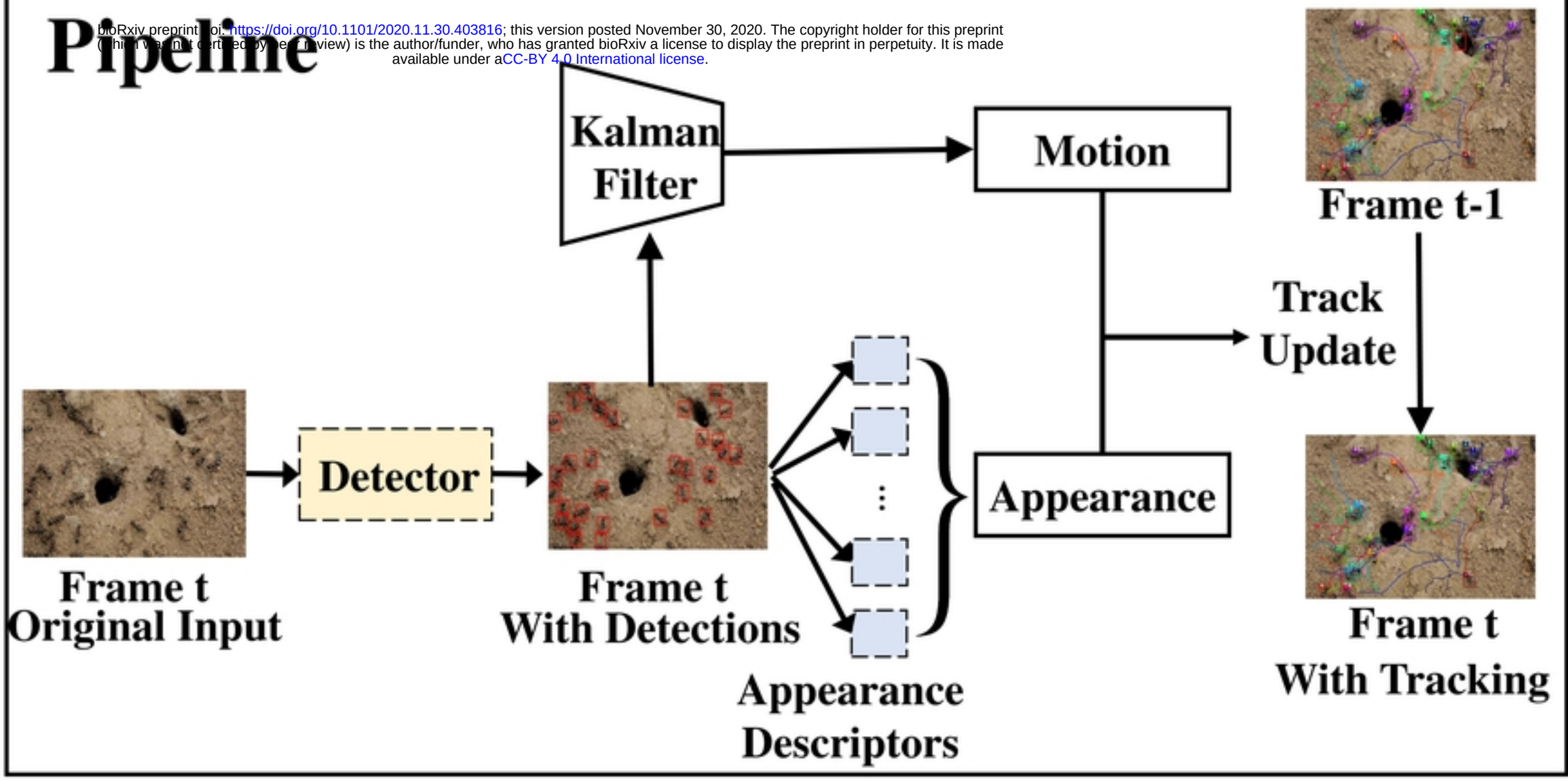
(b) Enter the scene

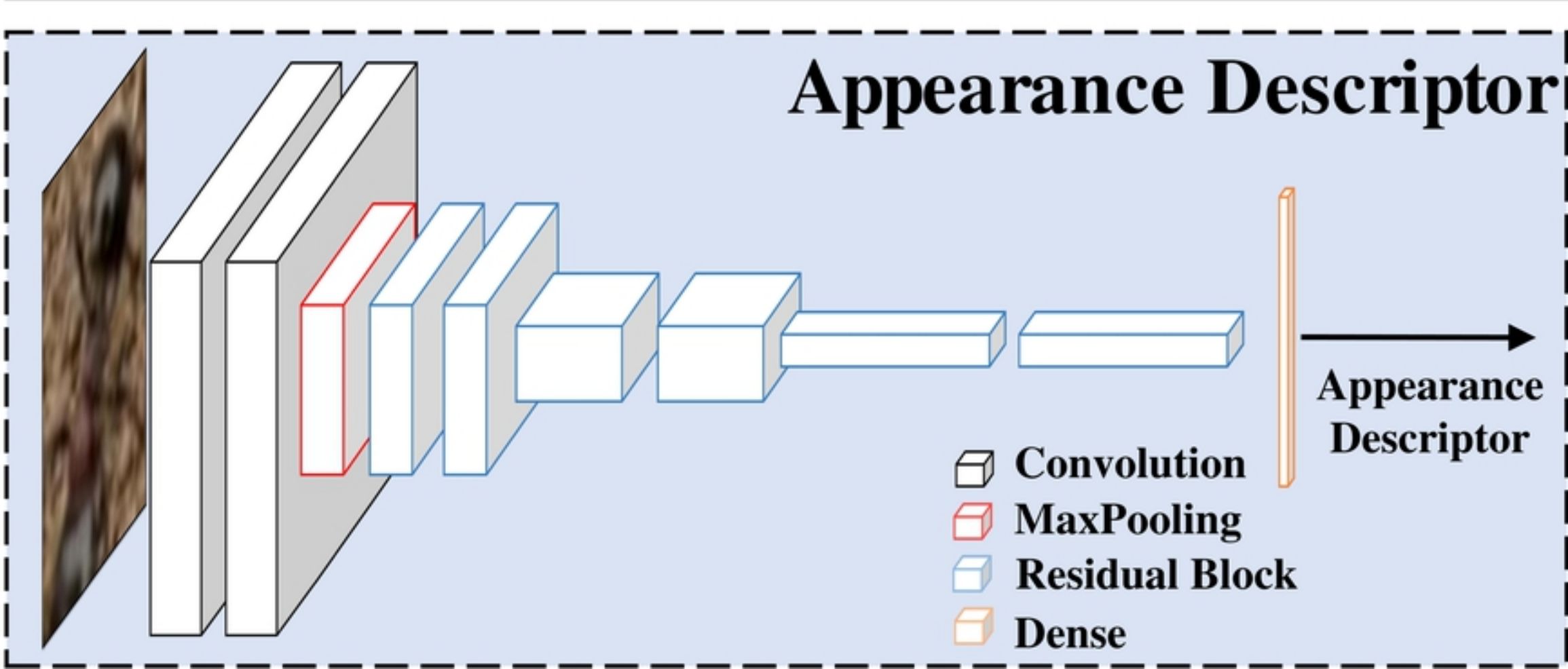


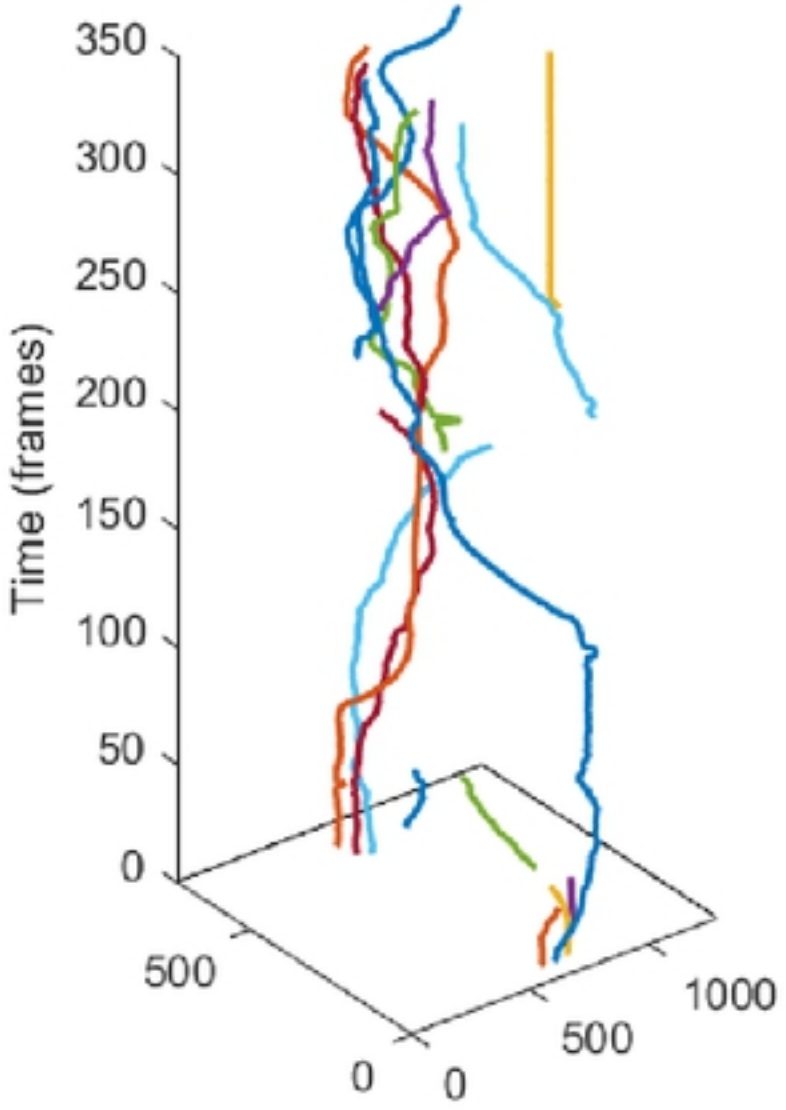
(c) Non-rigid warp



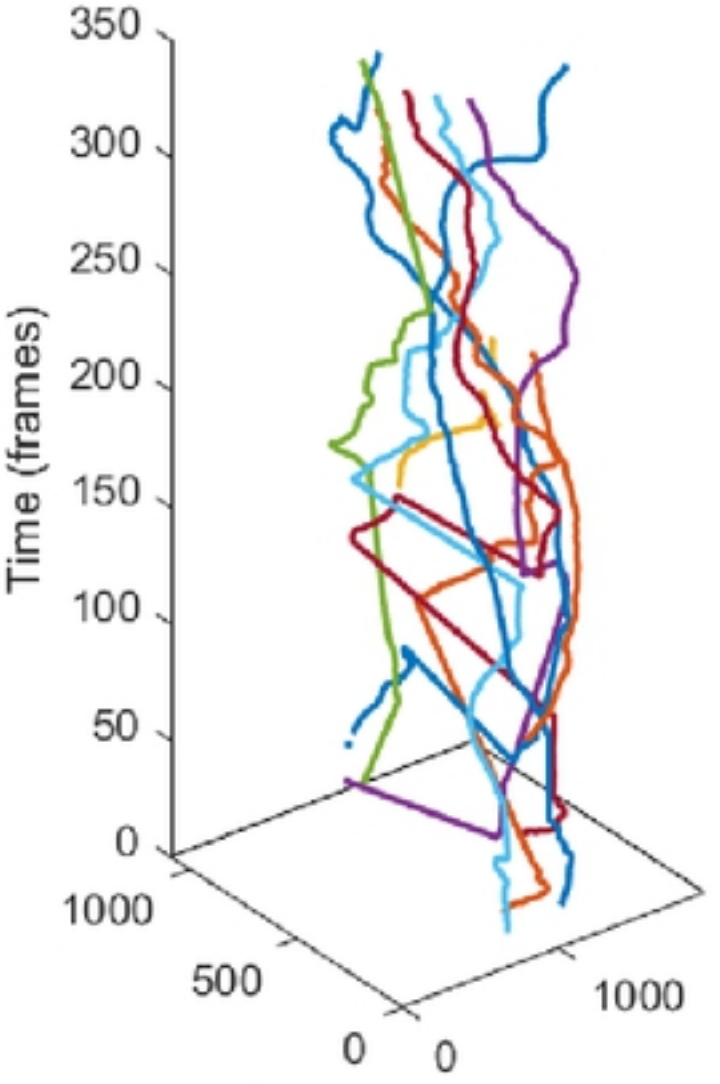
Pipeline



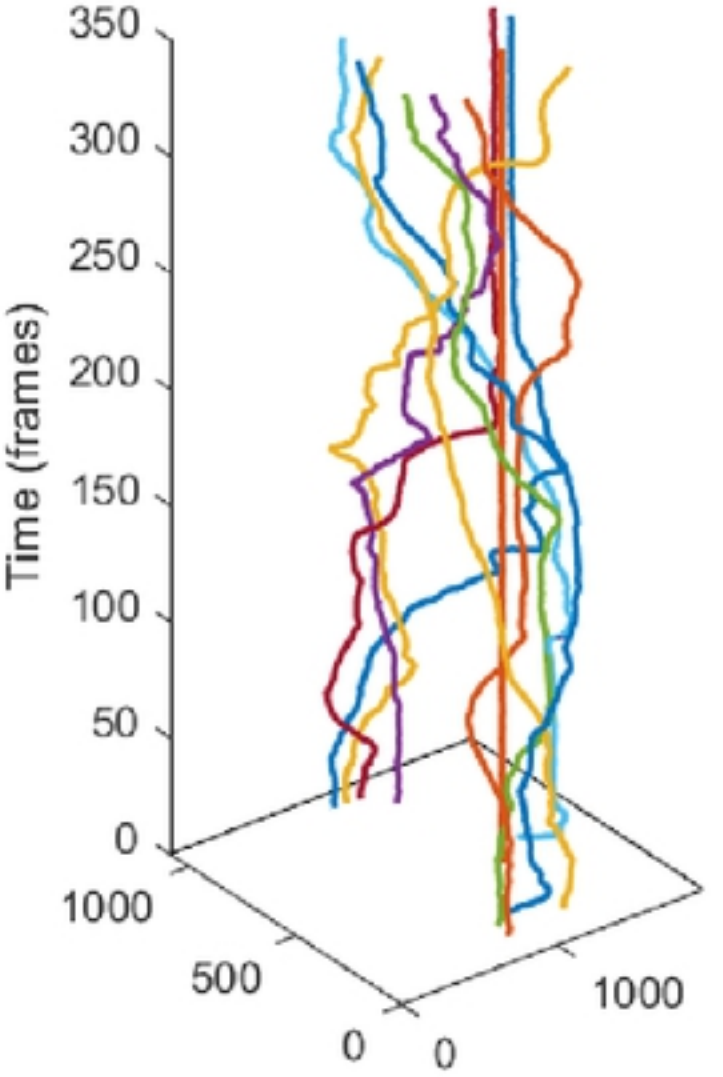




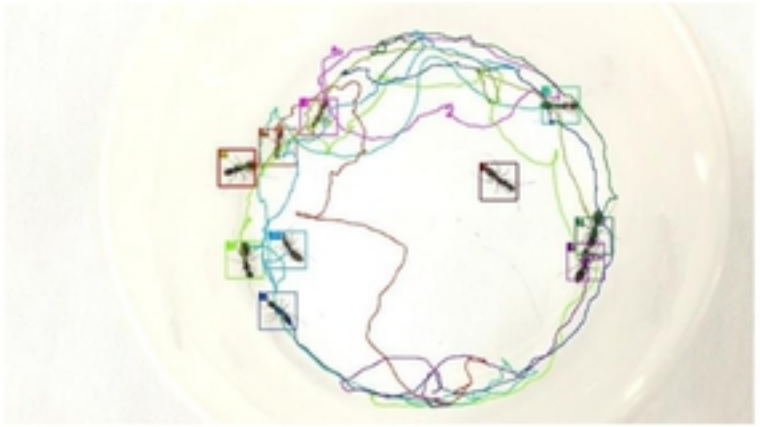
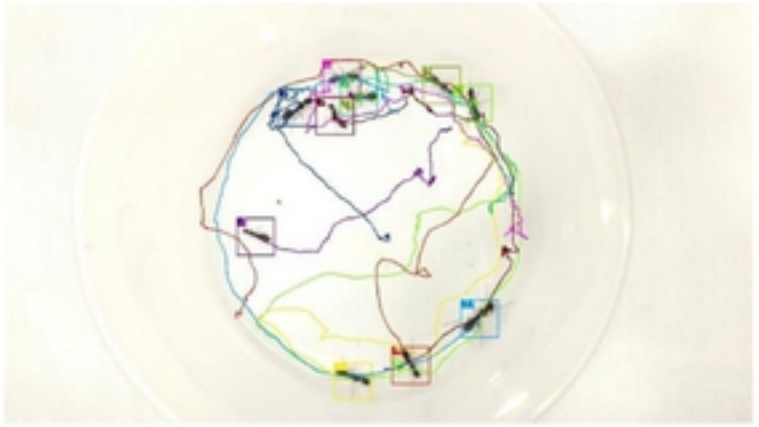
(a) idTracker

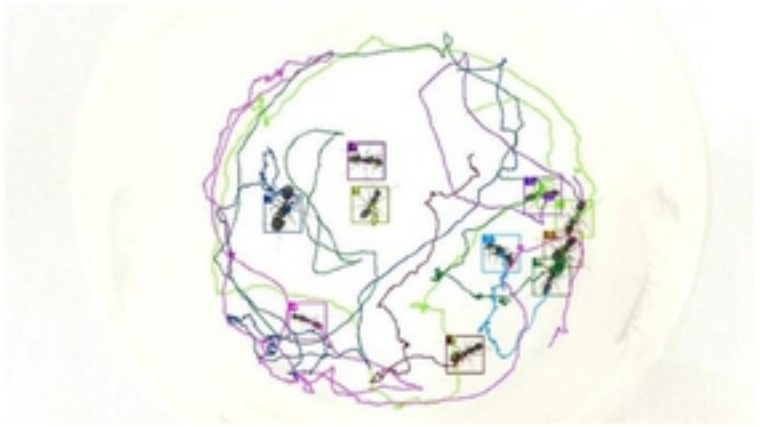


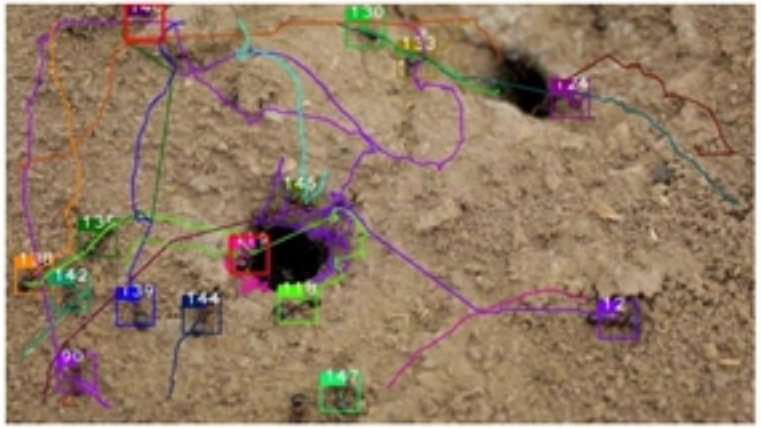
(b) Ctrax

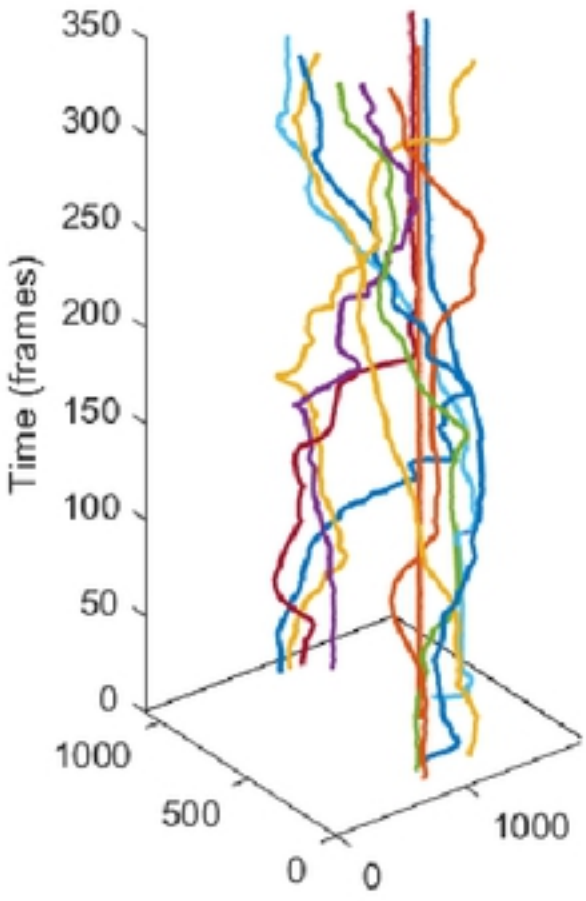


(c) Ours

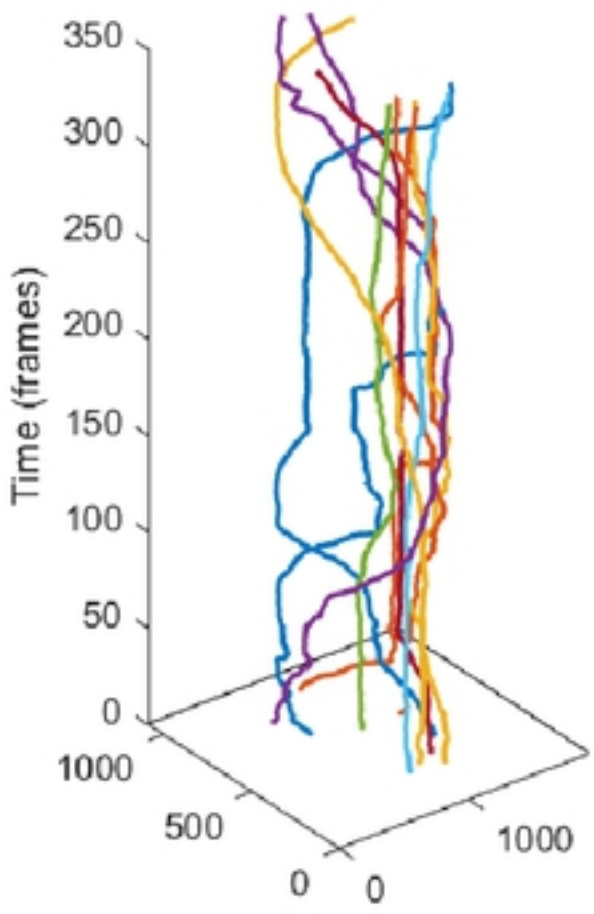





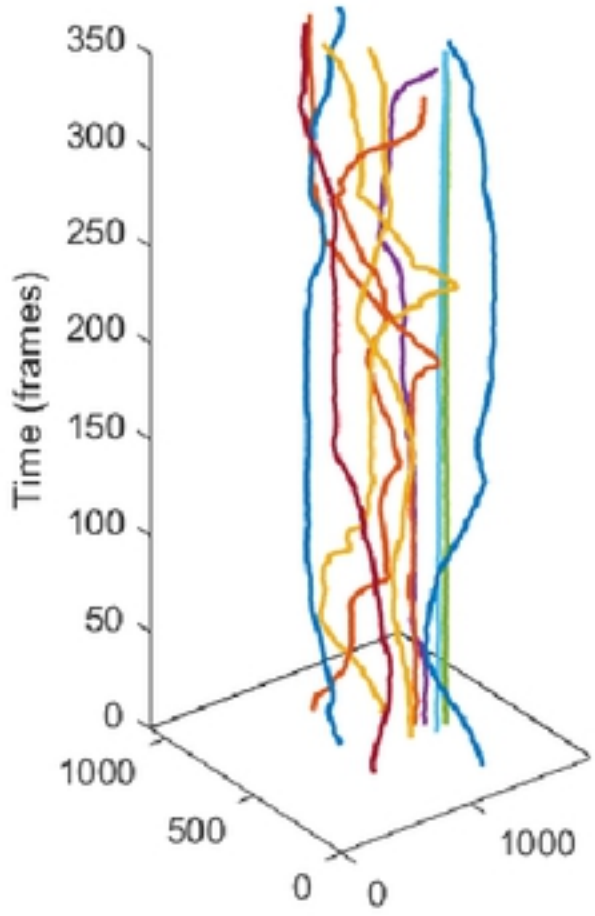





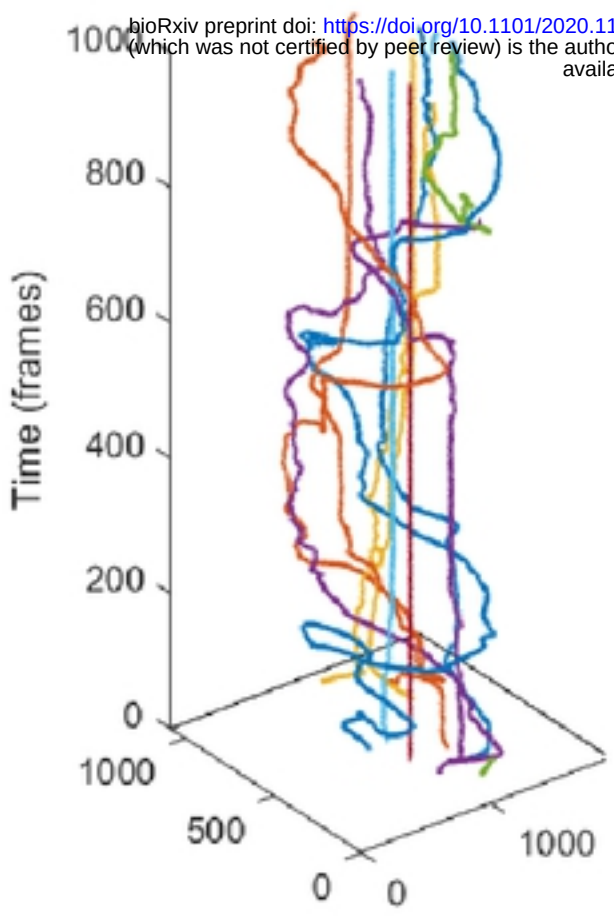
(a) $\mathcal{I}_1$



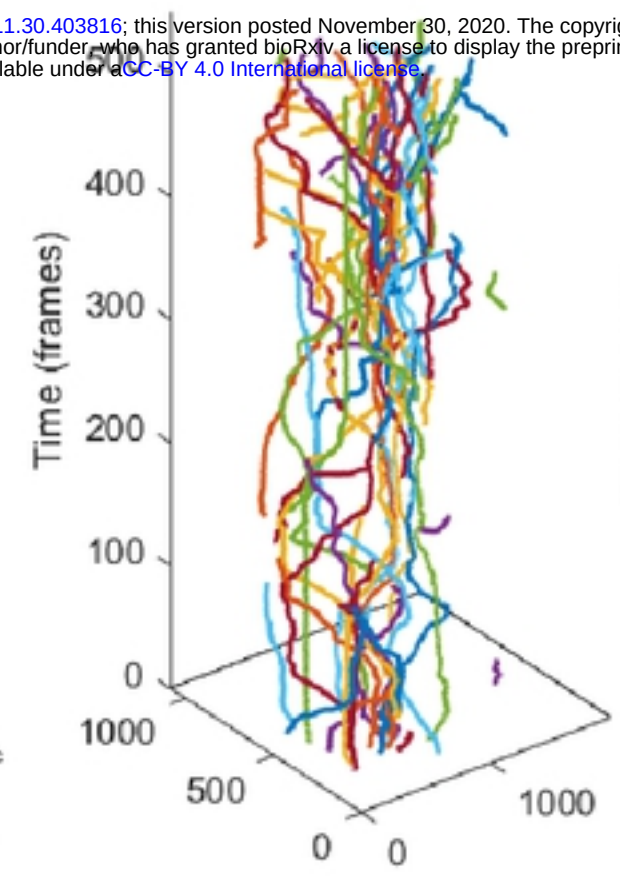
(b) $\mathcal{I}_2$



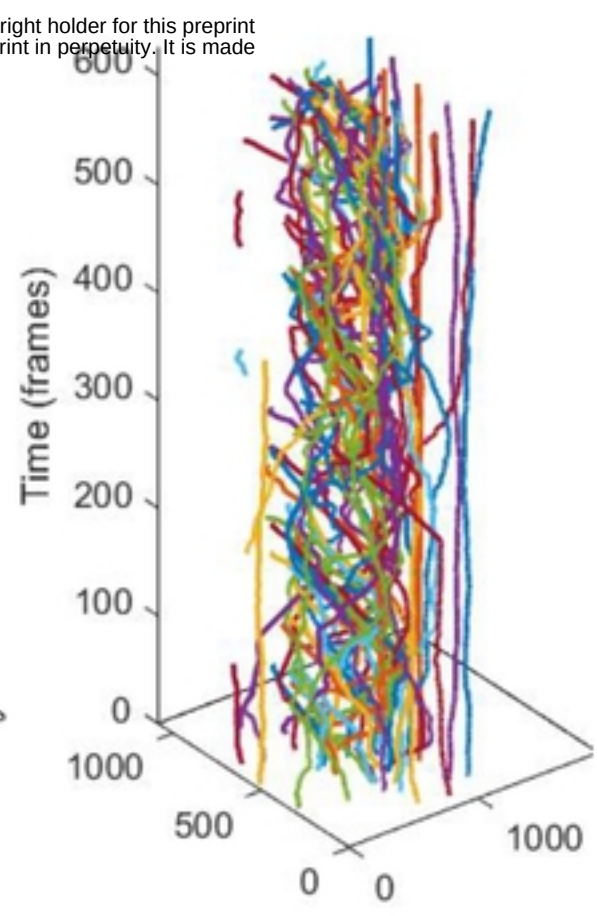
(c) $\mathcal{I}_3$



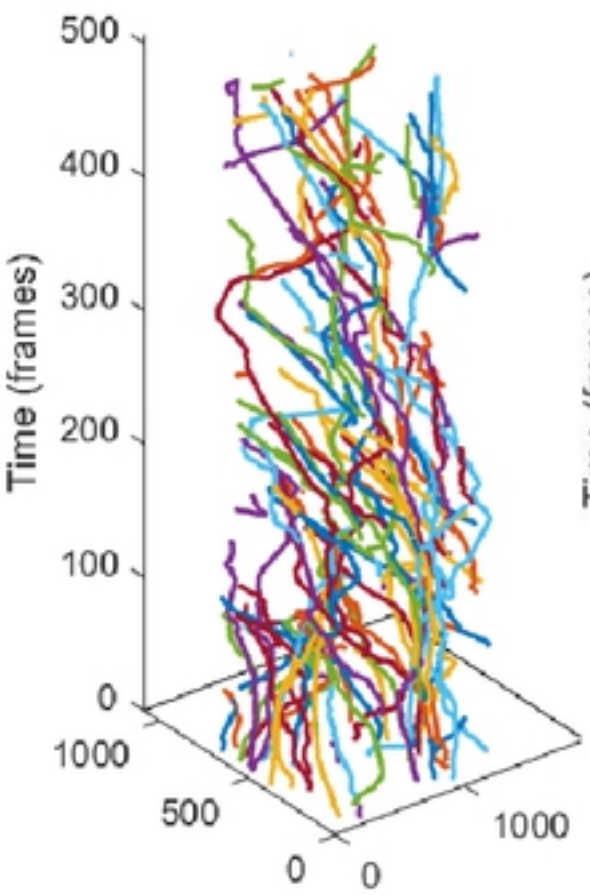
(d) $\mathcal{I}_4$



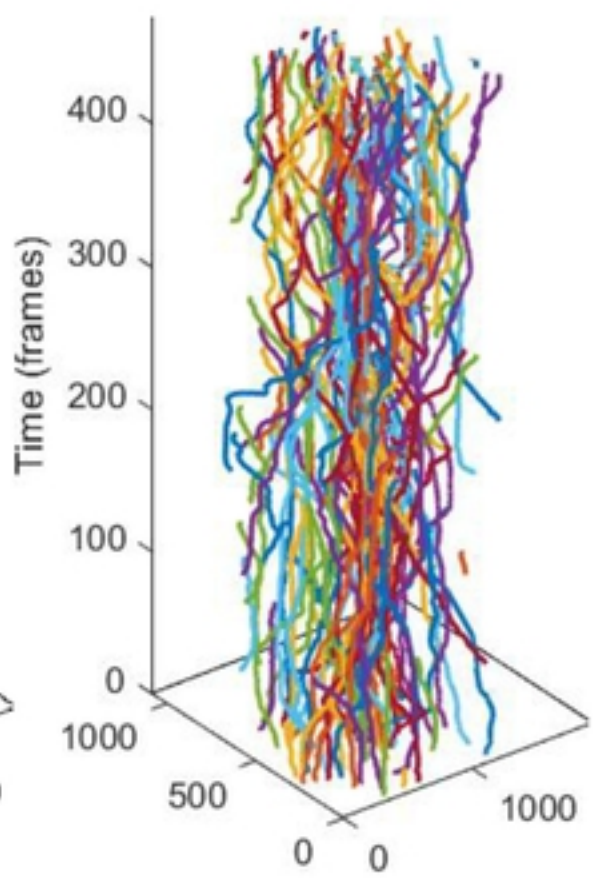
(f) $\mathcal{O}_1$



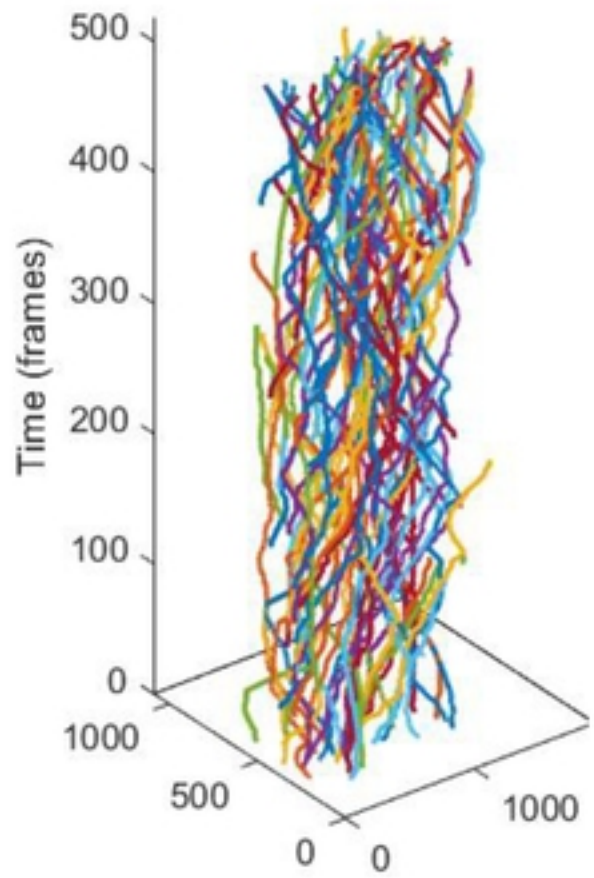
(e) $\mathcal{O}_2$



(g) $\mathcal{O}_3$



(h) $\mathcal{O}_4$



(i) $\mathcal{O}_5$