Interactive Tracking of Insect Posture

Minmin Shen[1], Chen Li[2], Wei Huang[3], Paul Szyszka[4], Kimiaki Shirahama[2], Marcin Grzegorzek[2], Dorit Merhof[5], and Oliver Duessen[1]

Abstract

In this paper, we present an association based tracking approach to track multiple insect body parts in a set of low frame-rate videos. The association is formulated as a MAP problem and solved by the Hungarian algorithm. Different from traditional *track-and-then-rectification* scheme, this framework refines the tracking hypotheses in an interactive fashion: it integrates a key frame selection approach to minimize the number of frames for user correction while optimizing the final hypotheses. Given user correction, it takes user inputs to rectify the incorrect hypotheses on the other frames. Thus, the framework improves the tracking accuracy by introducing active key frame selection and interactive components, enabling a flexible strategy to achieve a trade-off between human effort and tracking precision. Given the refined tracks at bounding box (BB) level, the tip of each body part is estimated, and multiple body parts in a BB are further differentiated. The efficiency and effectiveness of the framework is verified on challenging video datasets for insect behavioral experiments.

Keywords: Multiple object tracking, Active key frame selection, Interactive user correction and tracks refinement, Insect tracking

[1]M. Shen and O. Duessen are with the Dept. of Computer and Information Science and the INCIDE center, University of Konstanz, Konstanz, D-78464, Konstanz, Germany. M. Shen is also with the School of Software Engineering, South China University of Technology, Guangzhou, China 510640. E-mail: minmin.shen@uni-konstanz.de.
[2]C. Li, K. Shirahama and M. Grzegorzek are with the Research Group for Pattern Recognition, Institute for Vision and Graphics, University of Siegen, Siegen, D-57076, Germany.
[3]W. Huang is with the Dept. of Computer Science, Nanchang University, Nanchang, China 330031.
[4]P. Szyszka is with the Institute of Neurobiology, University of Konstanz.
[5]D. Merhof is with the Institute of Imaging & Computer Vision, RWTH Aachen University, Aachen, Germany.

1. Introduction

The movements of body parts of harnessed insects, such as antennae or mouthparts, provide information about internal states [1], sensory processing [2] and learning [3, 4, 5, 6]. Although there is some research reported in animal tracking, estimating the center of body mass (position) is much simpler than detecting the detailed body posture and position of appendages (pose) [7]. To the best of our knowledge, our work is the first research about tracking multiple insect body parts that are of different types. Insect posture is estimated as the tip of each body part (e.g. a bee's antennae or tongue as shown in Figure 2).

Although the application scenario of our tracking framework addresses a particular task, the challenges to be addressed, however, characterize a generic tracking problem resulting from: 1) varying number of targets, 2) incoherent motion, 3) occlusion and merges, 4) all targets have dark appearance, similar shape and no texture and 5) long tracking gaps. Most tracking frameworks assume a coherent motion, i.e. all the elementary targets move with similar average velocity over extended periods of time. However, this assumption does not hold here. A pictorial illustration is shown in Figure 1, where a set of object detections as unordered bounding boxes (BBs) are produced by a standard moving object detector. Different colors are used here to denote the expected label for better visualization. It can be seen that a merged (see Figure 1g) or false negative (FN) BB (see (b,i)) produces a tracking gap, which makes it unsuitable for frame-by-frame tracking approaches such as particle filter based algorithms [8]. As the mandibles (i.e. label 2 and 4) do not provide much information for biologists, we do not track them in the case where they are merged or occluded.

The different occlusion and merge conditions are illustrated as in Figure 2. We already attempted to address partly these issues in our previous work [9], but the targets are difficult to differentiate at BB level under merge conditions

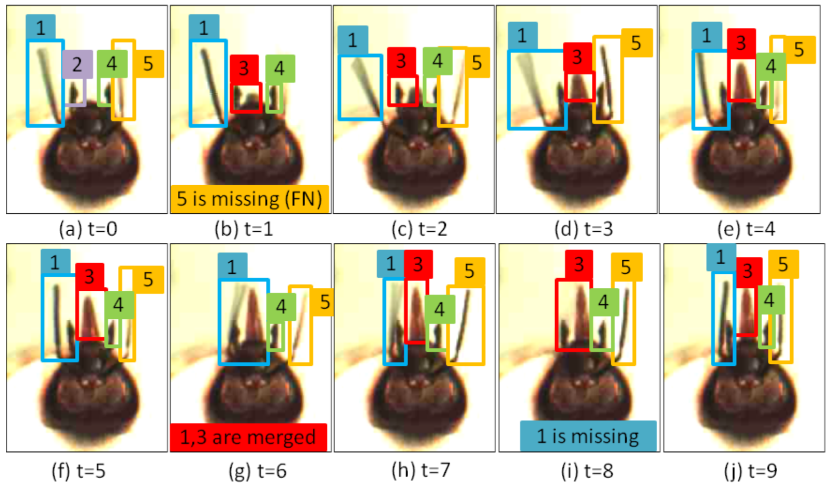

Figure 1: Object detections at 10 consecutive frames including merged and false negative BBs. Identification of each BB, shown in a different color, is a challenging task. The label for each body part is denoted as 1:left antenna; 2:left mandible; 3:proboscis; 4:right mandible; 5:right antenna.

(see Figure 2a,e). In this application, we denote *occlusion* as the cases where target a is occluded by target b, and *merge* where targets a and b are merged at the same BB. For occlusion conditions, estimating the position of an occluded target a if it is not visible makes little sense, though maintaining its identity when it appears again is challenging. For merge conditions, we propose a new algorithm to differentiate targets at pixel precision by estimating the tip of each target (shown as the small solid circle in Figure 2a,e).

The tracking problem of this paper is formulated as follows. The inputs to our tracking framework are a set of detection responses at BB level, thus only provide rough estimation of the targets' positions. We denote the detection responses by $\mathbf{Z}_{1:N} = \{\mathbf{z}_{i,t} | 1 \leq i \leq n_t, 1 \leq t \leq N\}$, where n_t is the number of detection responses at time t. Our objective is to estimate the trajectories of the tips of n targets. In the case of a honey bee, $n = 5$, i.e. 1: right antenna; 2: right mandible; 3: proboscis; 4: left mandible; 5: left antenna. The trajectories are denoted as $\mathbf{T} = \{T^i_{t_{i1},t_{i2}} | 1 \leq i \leq n\}$, where $T^i_{t_{i1},t_{i2}}$ is the track of the i^{th} target existing from time t_{i1} to t_{i2}.

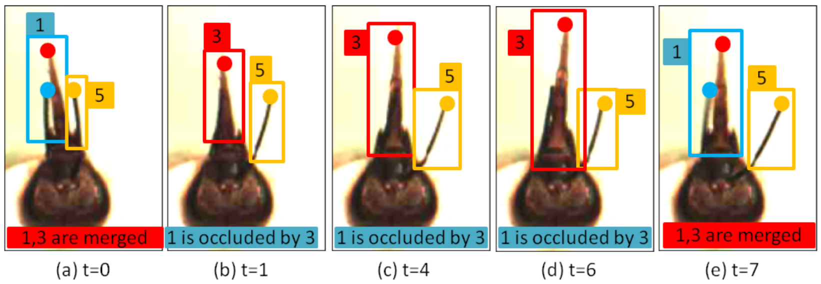

Figure 2: Sample frame of (a,e) merge or (b,c,d) occlusion. Merged targets are difficult to be differentiated at BB level, thus we propose to estimate the position of the tip of each target, which is denoted as a solid circle in the corresponding color.

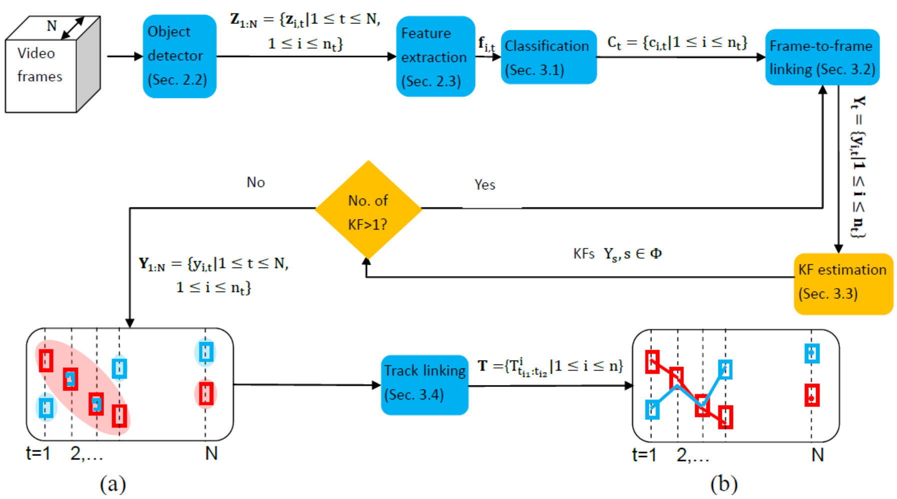

Figure 3: The flowchart of the overall tracking framework: the yellow blocks highlight the interactive part, while the blue blocks denote the automated computation part.

In this paper, we propose an interactive framework for insect tracking integrating a frame query approach, instead of the traditional track-and-then-rectification scheme. As shown in Figure 3, the overall framework includes six stages: (1) moving object detection, (2) feature extraction, (3) classification of moving objects, (4) constrained frame-to-frame linking, (5) key frame (KF) estimation and annotation query and (6) track linking through merge conditions. The yellow blocks highlight the interactive part, while the blue blocks indicate the automated computation part. We will address our tracking problem by fulfilling two sub-tasks. The first sub-task is to assign a label $y_{i,t}$ to the corresponding BB $\mathbf{z}_{i,t}$, and construct tracks at BB level $\mathbf{Y}_{1:N} = \{y_{i,t}|1 \leq i \leq n_t, 1 \leq t \leq N\}$. Given the input $\mathbf{z}_{i,t}$, a feature vector $\mathbf{f}_{i,t}$ is extracted to represent the information about its position, motion and shape. The initial label $y_{i,t}$ is estimated by classification (Section 4.1) and constraint frame-to-frame linking (Section 4.2). This framework queries users to rectify the incorrect labels only for certain frames (i.e. $Y_s|s \in \Phi$, where Φ is the set of KFs), which are estimated in Section 4.3, and the framework takes them as prior information to compute the labels of BBs on the other frames. The tracks are iteratively refined until user query is no longer required. As a result, reliable tracks $\mathbf{Y}_{1:N}$ are constructed, which is indicated with a pink shaded ellipse in Figure 3a. The second sub-task is to find the position of the tip (i.e. the endpoint, shown as colored solid circles in Figure 2) of each target $\mathbf{x}_t^i$ and construct complete tracks $\mathbf{T} = \{T_{t_{i1},t_{i2}}^i|1 \leq i \leq n\}$ through merge or occlusion conditions, which are indicated as solid colored lines in Figure 3b. We propose an algorithm in Section 4.4 to link the gaps between the tracks to compute automatically the final trajectories $\mathbf{T}$.

The rest of this paper is organized as follows. The related work to this paper is summarized in Section 2. The preliminaries are introduced in Section 3, including object detection and preprocessing (Section 3.1) and an anatomical model of insect body parts (Section 3.2). The proposed tracking framework is elaborated in Section 4. Its practicability and accuracy is validated by experimental results in Section 5. Section 6 concludes this paper.

2. Related Work

The multiple object tracking (MOT) approaches could be classified into two categories [10]: Association based tracking approaches and Category free tracking. The former category usually first localizes objects in each frame and then links these object hypotheses into trajectories without any initial labeling. The latter one, also referred as online object tracking [11], requires the initialization of a fixed number of objects in the first frame (in the form of BBs or other shape configurations), then localizes these fixed number of objects in the subsequent frames. As we aim to track varying number of objects, we adopt the association based tracking approach. The success of most existing association based tracking algorithms comes from discriminative appearance model (using the cues of color or texture) [12, 13], or constant velocity motion of targets [14, 12, 15, 13]. There are a few published studies that address problems in tracking animals. They include algorithms for tracking freely moving animals (e.g. bee dance [16, 17], ants [18, 19]) and freely moving body parts of harnessed animals (e.g. bees' antennae [20], mouse whiskers [21]). A more detailed review could be found in [7]. We summarize the related work to this paper and their main characteristics in Table 1, including the tracking framework, appearance model and type or number of target(s). We also list the assumptions of these works according to the authors, which may make them inapplicable for our case. In this paper, we take an association based approach, designing an MAP framework that maps the difficult MOT problem into a simpler object classification problem with regularization via temporal correlations. This method is able to address the challenges here such as incoherent motion and merge or occlusion conditions.

To overcome the bottleneck of the automatic tracking performance by introducing user input, some interactive algorithms have been reported [25, 26, 27, 28]. But some of them either requires users to view the whole video [26, 25], or not to focus on frame query techniques [27]. The most conceptually similar work to ours is proposed in [28], which extends the tracker in [29] by estimating

Table 1: Related work and their main characteristics.

	Tracking framework	Appearance model	Type/number of target(s)	Remarks
[14]	Hungarian	Foreground response	Multiple generic objects	Assume coherent motion
[12]	Hungarian	Color histogram	Multiple human pedestrians	Assume coherent motion
[15]	Hungarian	Foreground response	Multiple cars	Assume coherent motion
[13]	Hungarian	Color histogram	Multiple human pedestrians	Assume coherent motion
[16]	Particle filter	Geometrical features	Single bee	Incorporate specific behavior model
[17]	Particle filter	Optical flow	Single bee	Assume coherent motion
[8]	Particle filter	Foreground response	Multiple mice and larvae	Assume coherent motion
[18]	Simple data association technique	Foreground response	Multiple ants	Does not tackle occlusion and merges
[19]	Particle filter	Foreground response	Multiple ants	Assume coherent motion
[22]	Not specified	Specific warm-like Insect features	Multiple Drosophila larvae	Does not resolve collisions involving more than two animals
[23]	Hungarian	Area of connected components	Multiple Drosophila adults	Assume coherent motion
[24]	Graph based framework	Combined features that capture local spatiotemporal structure	Multiple Drosophila larvae	Training samples of encounters of two larvae required
[20]	Antennae identified by two largest clusters	None	Two bee antennae	Does not tackle MOT problems including merge, occlusion, etc
[21]	Probabilistic framework	Splines	Mouse whiskers	Does not tackle MOT problems including merge, occlusion, etc
[7]	Probabilistic framework	Intensity map	Constant number of animals	Assume two blobs of the same object overlap in some consecutive frames, etc

more KFs for user annotation to improve the tracking accuracy. However, since the KF estimation scheme in [28] punishes significant label change, it is not applicable in our task, where different objects could be detected in turns at the same position (see Figure 1(f,g)).

3. Preliminaries

When controlled stimulus conditions are needed, insects are often restrained and their behavior is monitored as movements of body parts such as their antenna or mouthparts. The proboscis is the mouthpart of the insect, and hungry bees extend their proboscises reflexively when stimulated with food or with a previously conditioned odorant (Figure 4).

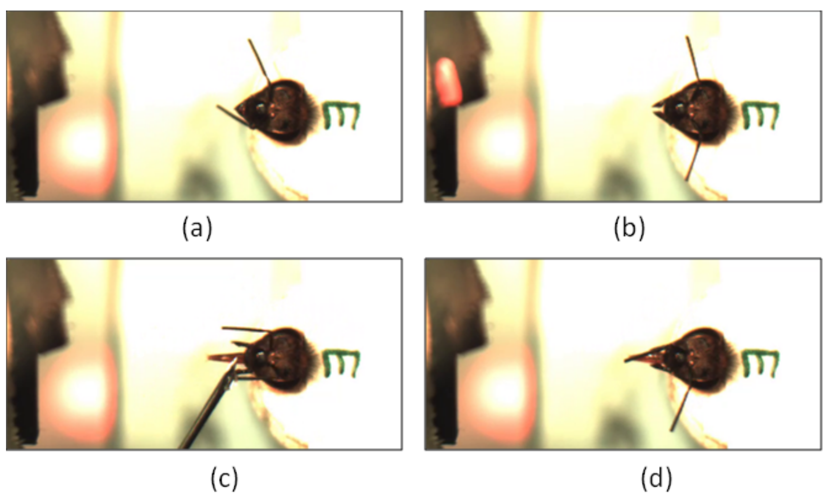

Figure 4: Associative odor-reward learning paradigm in honey bees. A bee that has learned the association between an odorant and a food-reward extends its proboscis when stimulated with the learned odorant: (a) before odorant stimulation, (b) odorant released indicated by the LED, (c) sugar rewarding, (d) during odorant stimulation.

3.1. Object Detection and Preprocessing

As our interests focus on tracking the antennae and mouthparts when they are moving, it is preferred to detect the moving part rather than segmenting the body part on a single frame basis. Thus, Gaussian Mixture Model (GMM) background modeling [30] is used. A more advanced background subtraction method based on a dynamic background model [31] may reduce false detections, but a standard moving object detector is used here as we focus on the tracking part. The object detector generates an unordered set of bounding boxes (BBs) including false positives (e.g. shadows, reflection and the insect's legs), false negatives (e.g. motion blurred antennae), missing objects (e.g. the antenna above the insect's head, or the proboscis not extended), merged detections (one bounding box including two or three objects) and occluded detections, which make the following tracking task difficult. Therefore, pre-processing operations include shadow removal [30], exclusion of undesired objects by incorporating position information, and segmentation of merged measurements.

These pre-processing operations greatly reduce the undesired detection measurement, but some false, missing, merged measurements may still remain. Thus, a subsequent tracking algorithm is required to tackle this problem.

3.2. Anatomical Model of Insect Body Parts

Modeling the anatomy of an insect's head is important for accurate tracking, due to the physical limitations of the moving objects' relative positions. The positions of insect body parts (e.g. antennae and mouthparts) are ordered in a certain sequence, which is rather similar among various insects. Figure 5 shows an image of an ant's head. These body parts are symmetric, thus they could be classified according to their types, and then further identified (tracked) by exploiting the temporal correlation between neighboring frames. Our framework incorporates an anatomical model of insects' heads as a priori, which is elaborated in Section 4.2.

We use a feature vector $\mathbf{f}_{i,t}$ to represent each BB in terms of its position, motion and shape. We follow our previous work [32, 33] to extract the information of position and motion. A challenge in our tracking task results from the similarity of the objects of interest, all of which have dark appearance, similar shape, and no texture. Therefore, some widely used features (such as color his-

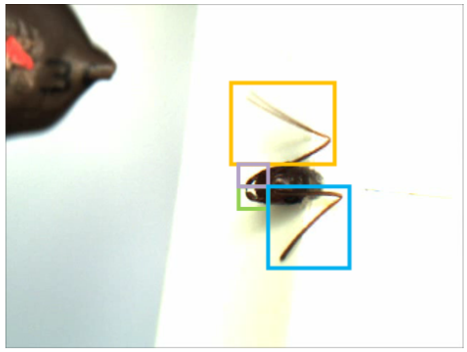

Figure 5: The closed up sample video frame of an ant. In the case of an ant, two antennae (yellow and blue BB) and its mouthparts (purple and green BB) need to be tracked (i.e. $n = 4$).

togram [34], image patch [27] and Haar-like features [35]) are not good choices for discriminative representation here. For example, the advantage of point based features originates from the discriminative local appearance at interest points [36, 37, 38], which is distinct from surroundings (or other targets) and remains consistent over time. However, the local features at interest points of our targets vary dramatically over time, as they tend to move incoherently. It is illustrated in Figure 6 where the Kanade-Lucas-Tomasi (KLT) feature tracker [39] fails to track the left antenna. The initial interest points are detected by a corner detector [40].

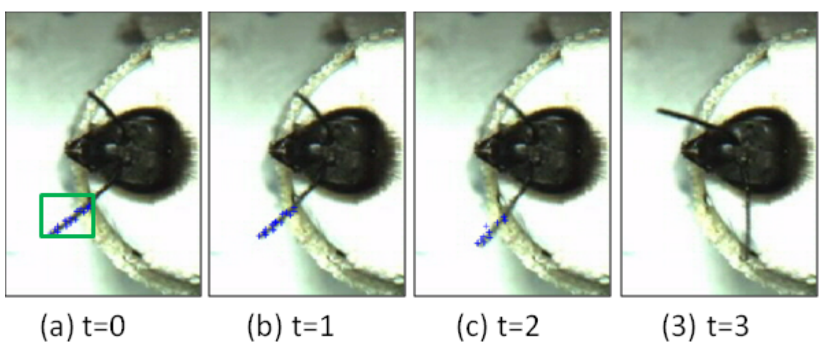

Figure 6: The initial interest points in (a) (denoted by blue stars) are detected by corner detector within the green bounding box. The number of successfully tracked points reduces dramatically over time (b) ten (c) six (d) zero.

To characterize the shape of each object, an appropriate shape descriptor should be used to model its appearance. The appearance model based on shape context has been successfully used in many machine vision tasks such as frontal face recognition [41], smooth object retrieval [42]. We used the top-hat filter as a line detector in our previous work [32] to differentiate a bee's antenna from other objects, as a bee's antenna is line-shaped. But this is not applicable for the other insects such as ants. Popular shape descriptors include the *Edge Histogram Descriptor* (EHD) [43], the *Isomerous Edge Histogram Descriptor* (IEHD) [44], the *Geometrical Feature* (GF) [45] (including the object perimeter, area, etc), the *Shape Signature Histogram* (SSH) [46], the *Fourier Descriptor* (FD) [47] and the *Internal Structure Histogram* (ISH) [44]. These six feature extraction methods describe shapes from different perspectives.

To select an effective shape feature for representing the insect body parts, two characteristics should be considered here: first, all body parts have small sizes in the frames (about 200$\sim$600 pixels); second, all body parts have simple bendability, i.e. few local boundary information such as curvature and junctions are present. Some examples of detected body parts are illustrated in Figure 7. The first characteristic makes the discrete points on the edges of the body parts limited (about 50$\sim$200 points), so that the boundary-based shape descriptors are not able to obtain enough good sample points. The FD also suffers from this fact. The second characteristic weakens the descriptive power of GF. We found that insect body parts' shapes embody good linear edges in different orientations. This indicates that edges are important low-level features in image description, thus we choose EHD as the shape descriptor. It is verified by the comparison of the six shape discriptors in Section 5.2.3. EHD is one of the most popular edge-based features, and able to describe both local and global features. In our work, EHD is used to describe the global shape features of insect body parts by two steps. First, the regional edge histograms are extracted based on five categories of edge directions: $45°$, $90°$, $135°$, $180°$ and any other degrees. Second, a global edge histogram is calculated as the mean value of the extracted histograms.

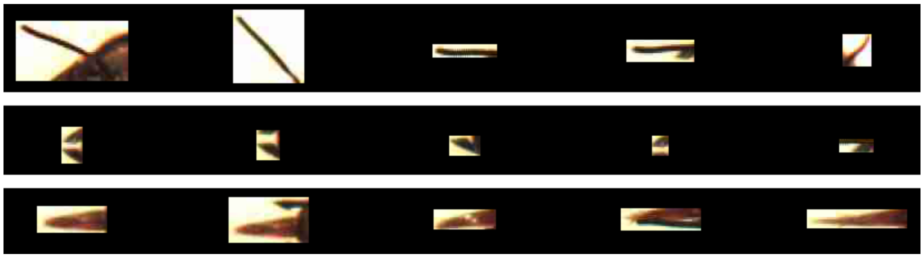

Figure 7: The first row are detected antennae, the second row are detected mandibles, and the third row are detected proboscis.

4. Proposed Interactive Framework

Similar to many association based approaches (e.g. [48]), we define the association as a MAP problem. Our objective is to determine correspondence of multiple BBs through N frames. Under the MAP framework, a global optimum $\widehat{\mathbf{Y}}_{1:N}$ is found by maximizing the posterior probability $P(\mathbf{Y}_{1:N}|\mathbf{Z}_{1:N})$:

$$\widehat{\mathbf{Y}}_{1:N} = \arg\max_{\mathbf{Y}_{1:N}} P(\mathbf{Y}_{1:N}|\mathbf{Z}_{1:N}) = \arg\max_{\mathbf{Y}_{1:N}} \prod_{t=1}^{N} P(Z_t|Y_t)P(\mathbf{Y}_{1:N}) \tag{1}$$

where Z_t, Y_t are ordered collections of BBs $\mathbf{z}_{i,t}$ and its label $y_{i,t}$ at time t: $Z_t = \{\mathbf{z}_{i,t}|1 \leq i \leq n_t\}, Y_t = \{y_{i,t}|1 \leq i \leq n_t\}$. $P(Z_t|Y_t)$ is the likelihood that the collection of BBs Z_t is generated from the sequence of labels Y_t. We assume that Y_t is temporal independent of each other. $P(\mathbf{Y}_{1:N})$ is the a priori probability of a labeling sequences $\mathbf{Y}_{1:N}$. The labels are initially estimated at frame level (Section 4.1), and then temporal correlation is considered for refinement by data association (Section 4.2).

4.1. Object Classification

Due to the symmetry of insect's appearance, a detection response $\mathbf{z}_{i,t}$ is first classified as one of m classes $c_{i,t}$, where $c_{i,t} \in \{$1:antenna; 2:mandible; 3:proboscis$\}$. Its label $y_{i,t}$ is estimated by differentiating the details (either on the left hand side or the right hand side) in the following tracking step.

In this paper, we select the Support Vector Machine (SVM) as a classifier. It improves the performance of our previous work in [9] due to its advantage of dealing with high-dimensional data. Probability-based classifiers (Naïve Bayes) need a large number of training examples to appropriately estimate probabilistic distributions in high-dimensional feature spaces [9, 49]. Similarity-based classifiers (e.g. k-Nearest Neighbour) fail to appropriately measure similarities in high-dimensional feature spaces, because of many irrelevant dimensions. In this work, we adopt a multi-class Support Vector Machine (mSVM) using the one-against-one (1vs1) strategy. Each class is determined by computing pair-wise votes using two-class SVMs. In the case of K classes, $K(K-1)/2$ two-class classifiers are trained. The final classification result is determined by counting which class the object has been assigned to most frequently.

The object classification generates a class label $c_{i,t}$ and the corresponding class probability $P(c_{i,t}|\mathbf{z}_{i,t})$ for each BB. Given the output of this classification step, however, two challenges remain in the following tracking task: 1) incorrect classification hypotheses, 2) identity swapping due to the interaction of moving objects.

4.2. Constrained Frame-to-Frame Linking

Based on the output of object classification $c_{i,t}$, we exploit the appearance information of an insect, i.e. position and ordering of $\mathbf{z}_{i,t}$, to assign the label $y_{i,t}$. As we assume that the likelihood $P(Z_t|Y_t)$ is temporally independent, the label $y_{i,t}$ is determined by the class label $c_{i,t}$ and the relative position of $\mathbf{z}_{i,t}$ to the origin (left or right).

Incorporating prior knowledge of the appearance model: The likelihood $P(Z_t|Y_t)$ is estimated following the constraint that Z_t should be ordered in an ascending manner, as insect body parts are assumed to be ordered in a certain sequence. The label sequences Y_t that violate this assumption will be considered as incorrect hypotheses (i.e. $P(Z_t|Y_t) = 0$). For other Y_t, the likelihood $P(Z_t|Y_t)$ is computed considering the rule of combination without repetition, as n_j BBs are detected out of n objects.

$$P(Z_t|Y_t) = \begin{cases} 0 & \text{if } \hat{m}_1 > m_1 \text{ or } \hat{m}_2 > m_2 \\ & \text{or } \hat{m}_3 > m_3 \text{ or } \exists \mathbf{z}_{i,t} > \mathbf{z}_{k,t}, \forall k < i \\ \binom{n}{n_j} & \text{otherwise} \end{cases} \tag{2}$$

where m_k is the number of $\{C_t|c_{i,t} = k\}$. This is considered as a priori knowledge incorporating the characteristics of insects' appearance. It is easily adapted to other insects by setting the value of m_k and n.

Estimation of benchmark frames: The frames with the highest posteriori probabilities are assumed to be correct hypotheses. Among these frames, we define a set of frames Ψ as the *benchmark frames*: Y_b, where $b \in \Psi : P(Z_t|Y_t) = 1$ & $P(Z_{t\pm1}|Y_{t\pm1}) \neq 1$.

We define $P(\mathbf{Y}_{1:N})$ in Equation (1) to guarantee that only the benchmark frames are used to help rectify the potentially incorrect hypotheses on their neighboring frames by data association:

$$P(\mathbf{Y}_{1:N}) = \prod_{b \in \Psi} P(Y_{b\pm1}|Y_b) \tag{3}$$

The conditional probability $P(Y_{b\pm1}|Y_b)$ is defined as a function of the pairwise linking cost between Y_b and $Y_{b\pm1}$:

$$P(Y_{b\pm1}|Y_b) = \prod_{i,k} P(y_{i,b} \mapsto y_{k,b\pm1}) \tag{4}$$

where the sign "$\mapsto$" denotes correspondence. The frame-to-frame linking between Y_b and $Y_{b\pm1}$ is found by forming a $n_t \times n_t$ cost matrix $\mathbf{M} = \{M_{i,k}\}$ with

$$M_{i,k} = -logP(y_{i,b} \mapsto y_{k,b\pm1}) = \|\mathbf{z}_{i,b} - \mathbf{z}_{k,b\pm1}\| \tag{5}$$

where $n_t = \max(\mathrm{n_b}, \mathrm{n_{b\pm1}})$ and the sign "$\mapsto$" denotes correspondence. As an association optimization algorithm, Hungarian algorithm [50] is applied to find the optimal linking by minimizing the linking cost.

The likelihood of frames $Y_{b\pm1}$ (i.e. those frames that are rectified with Y_b) is recomputed as

$$P(Z_{b\pm1}|Y_{b\pm1}) = \begin{cases} 0 & \text{if } \hat{m}_1 > m_1 \text{ or } \hat{m}_2 > m_2 \\ & \text{or } \hat{m}_3 > m_3 \text{ or } \exists \mathbf{z}_{i,b\pm1} > \mathbf{z}_{k,b\pm1},\ \forall k < i \\ 1 & \text{otherwise} \end{cases} \tag{6}$$

New benchmark frames are estimated and frame-to-frame linking is performed iteratively.

4.3. KF Estimation and Annotation Query

According to Equations (1) and (3), $\widehat{\mathbf{Y}}_{1:N}$ is the current optimal estimation for the labels given a set of benchmark frames in $\{Y_b, b \in \Psi\}$ estimated in Section 4.2. The success of frame-to-frame linking lies in the estimation of benchmark frames. We use prior knowledge in Equation (2) to initially estimate the set of benchmark frames Ψ, but the constraints in Equation (2) do not always hold, and some frames could not be rectified with the given benchmark frames.

To refine further $\widehat{\mathbf{Y}}_{1:N}$, it is required to determine new benchmark frames $\{Y_b, b \in \Psi\}$ in Equation (3) to form a new set Ψ^* by introducing human effort. With the new benchmark frames, the constraint in Equation (2) is relaxed. To minimize user effort, we propose an approach to minimize the number of KFs while optimizing the final hypothesis. The intuitive concept is that only the potential benchmark frames should be rectified, so that corrections on the rectified KFs could propagate to their neighboring frames in the subsequent frame-to-frame linking. Given the new set Ψ^* with added KFs obtained from the user annotation, we combine Equations (1) and (3) and define a new cost function

$$\widehat{\mathbf{Y}^*}_{1:N} = \arg\max_{\mathbf{Y}^*_{1:N}} \prod_{t=1}^{N} P(Z_t|Y_t) \prod_{b\in\Psi^*} P(Y_{b\pm1}|Y_b) \tag{7}$$

The refined labels $\widehat{\mathbf{Y}^*}_{1:N}$ are found by solving Equation (7).

As illustrated in Figure 3, we refine the incorrect hypotheses in $\widehat{\mathbf{Y}}_{1:N}$ by interactively 1) requesting user correction on estimated KFs; 2) taking corrected

KFs and rectifying their neighboring frames by frame-to-frame linking and 3) updating KFs. We define the annotation cost of each frame to indicate the degree of "usefulness" of user annotation, in order to estimate which frame should be the potential benchmark frame and added to form a new set of benchmark frames Ψ^*. The higher the annotation cost is, the more erroneous Y_t tends to be. Naturally, the annotation cost is related to the probability of incorrect hypothesis. Here we consider two conditions of frames $\widehat{\mathbf{Y}}_{1:N}$, i.e. $Y_{b\pm1}$ and the others. For $Y_{b\pm1}$, we should also take their association with Y_b into consideration. Therefore, the annotation cost is defined as

$$A(Y_t) = P_\epsilon \begin{cases} 1 - P(Z_t|Y_t)\prod_{i,k} P(y_{i,t} \mapsto y_{k,t\pm1}) & t = b \pm 1 \\ 1 - P(Z_t|Y_t) & \text{otherwise} \end{cases} \tag{8}$$

As $A(Y_t)$ interprets the probability that $y_{i,t}$ could be an incorrect hypothesis, it provides a flexible strategy for users to set the threshold τ, for which one could choose KFs from the frames $A(Y_t) \geq \tau$ considering the trade-off between tracking accuracy and human effort. The KFs Y_s are defined as $s \in \Phi$: $P(X_{s-1}|Y_{s-1}) = 1$ & $A(Y_s) \geq \tau$. Users are queried to rectify the KFs Y_s, which are subsequently used to form a new set of benchmark frames as $\Psi^* = \Psi \cup \Phi$.

4.4. Track Linking Through Merge Conditions

Given reliable tracklets $\widehat{\mathbf{Y}^*}_{1:N}$ as benchmarks, we treat them as rough approximation of the tips. To extract further the positions of the tips of each object at pixel level $\mathbf{x}_t^i$ through merge conditions, we propose an approach to link the tracklets by interpolating the missing tracklets on the in-between frames. Let us denote the track of the i^{th} target as a set of tracklet association $T^i_{t^p_{i1},t^p_{i2}} = \{\mathbf{x}_t^i | t^p_{i1} \leq t \leq t^p_{i2}\}$, where t^p_{i1}, t^p_{i2} indicate the tail and head of the pth tracklet of $T^i_{t^p_{i1},t^p_{i2}}$, respectively.

For the merge condition where tips of targets a and b are merged (i.e. they are bounded within the same BB labeled $y_{a,t}$), we define $P_m(\mathbf{x}_t^a, \mathbf{x}_t^b \xrightarrow{m} y_{a,t})$ to indicate the probability of merge. It is defined as the product of the independent appearance component $P_{a,m}(\mathbf{x}_t^a, \mathbf{x}_t^b \xrightarrow{m} y_{a,t})$ and the temporal component

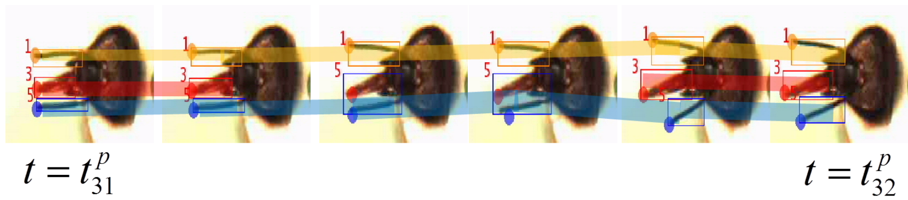

Figure 8: An example of linking tracks through merge condition: the shaded lines indicate the tracks at BB level, and the circles indicate the tips. t_{31}^{p}, t_{32}^{p} indicate the tail and head of the pth tracklet of the proboscis (i.e. label 3) $T^{3}_{t_{31}^{p}, t_{32}^{p}}$, respectively.

$P_{t,m}(\mathbf{x}_t^a, \mathbf{x}_t^b \stackrel{m}{\rightarrow} y_{a,t})$, respectively.

$$P_m(\mathbf{x}_t^a, \mathbf{x}_t^b \stackrel{m}{\rightarrow} y_{a,t}) = P_{a,m}(\mathbf{x}_t^a, \mathbf{x}_t^b \stackrel{m}{\rightarrow} y_{a,t}) P_{t,m}(\mathbf{x}_t^a, \mathbf{x}_t^b \stackrel{m}{\rightarrow} y_{a,t}) \tag{9}$$

where

$$P_{a,m}(\mathbf{x}_t^a, \mathbf{x}_t^b \stackrel{m}{\rightarrow} y_{a,t}) = \begin{cases} 1 & \text{if } \mathbf{f}_{a,t} \in \Xi \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

$$P_{t,m}(\mathbf{x}_t^a, \mathbf{x}_t^b \stackrel{m}{\rightarrow} y_{a,t}) = \begin{cases} 1 & \text{if } t_{b1}^{p} < t < t_{b2}^{p} \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

Here, Ξ is a set of $\mathbf{f}_{a,t}$ that constrains the position and size of the target. The starting and ending time indices t_{i1}^p, t_{i2}^p of the pth track $T^{i}_{t_{i1}^{p}, t_{i2}^{p}}$ are empirically set by defining the gap between its temporal neighboring tracks larger than a threshold α, i.e. $t_{i2}^{p} < t_{i1}^{p+1} - \alpha$.

We initialize the estimated tracks as the set of confident tracklets $\mathbf{T}^0 = \{\mathbf{x}_t^a | P_m(\mathbf{x}_t^a, \mathbf{x}_t^b \stackrel{m}{\rightarrow} y_{a,t}) = 0\}$. The tip $\mathbf{x}_t^a$ is determined by applying Morphological operations: the object is firstly thinned to lines, and the furthest end point to the centroid of the insect's head is estimated as the tip.

To fill the frame gap under merge condition, we use Harris corner detector to find M candidate pixel positions $\mathbf{x}_t^i, 1 \leq i \leq M$ to interpolate detection responses for estimating $\mathbf{x}_t^a$ and $\mathbf{x}_t^b$. We denote the set of candidate points as $\{\mathbf{x}_t^i \in \Omega\}$. The estimated tracks are constructed with new added points that are selected from Ω, which have the least pairwise linking costs to their temporal nearest neigbors in $\mathbf{T}^0$.

In summary, an overview of the algorithm is shown in Algorithm 1 and 2.

5. Experiments

5.1. Experimental Setup

Each individual insect was imaged using a CCD camera ("FMVU-03MTM/C" point gray), in order to record the head with appended body parts (e.g. proboscis, mandibles and antennae). Stimulus delivery (odor) is monitored by lighting an LED within the field of view of the camera, so that data analysis can be done relatively to stimulus delivery (see Figure 4, 5). Individual bees are harnessed on a platform, with their heads in fixed positions, but able to move antennae and mouthparts freely. The camera is focused statically on the top of an individual bee. Although it would be possible to record with a high speed camera, we aim at developing a framework that uses affordable cameras such as web-cam or consumer level cameras, which keeps the data volume low.

We developed a system *LocoTracker* to implement our algorithm in C++, using OpenCV library version 2.4.8 (http://www.opencv.org) and tested on an Intel Core2 CPU, 3.00 GHz, with 8 GB RAM. We constructed a Qt-based (http://qt-project.org/) graphical user interface (GUI) to display KF and take user annotation in order to implement user interaction in Section 4.3. For determining the KFs, the threshold of annotation cost is set as $\tau = 1$. The GUI displays each KF and the initial hypotheses, so that the user is able to recognize the errors and correct the mismatches, false negatives and false positives. Figure 9 shows two snapshots of the GUI, illustrating how this system facilitates user interactions.

We test LocoTracker on recorded videos of two types of insects, i.e. ten videos of a bee and one video of an ant. The anatomical model is trained on 10 manually annotated objects for each type. The characteristics of tested videos

Algorithm 1 Summary of the proposed algorithm (Sub-task 1).

Assign $y_{i,t}$ for each $\mathbf{z}_{i,t}$

Input: $\{\mathbf{z}_{i,t}\}$, n, m_k

1. Initialization: For each frame Z_t, compute $P(Z_t|Y_t)$ following Equation (2).

2. Updating:

while $\exists\ Y_t\ \overline{updated}$ **do**

end while

for $t = 1, \ldots, N$ **do**

end for

- Find the benchmark frames $\{Y_b\}$, where $b \in \Psi : P(Z_t|Y_t) = 1\ \&\ P(Z_{t\pm1}|Y_{t\pm1}) \neq 1$.
- Apply pair-wise linking only on $\{Y_b, b \in \Psi\}$ and their temporal neighbors $Y_{b\pm1}$, update labels $Y_{b\pm1}$.
- Mark $Y_b, Y_{b\pm1}$ *updated*.

3. KF estimation and annotation query:

- Query user correction and receive correction $Y_s, s \in \Phi : P(Z_{s-1}|Y_{s-1}) = 1\ \&\ A(Y_s) \geq \tau$.
- Form a new set of benchmark frames as $\Psi^* = \Psi \cup \Phi$.
- Update $P(Z_s|Y_s) = 1, \forall s \in \Phi$.

4.

if $\Phi \neq \oslash$ **then**

repeat step 2-3

end if

Output: $\widehat{\mathbf{Y}^*}_{1:N}, A(Y_t)$

Algorithm 2 Summary of the proposed algorithm (Sub-task 2).

Find the tip position $\mathbf{x}_t^i$ and link tracks through merge conditions

1. Initialization:

   - Construct initial tracks $\mathbf{T}^0 = \{\mathbf{x}_t^i \in \Omega\}$.
   - Estimate t_{i1}^p, t_{i1}^p as the tail and head of the pth track $T_{t_{i1}}^p$ by empirically setting a threshold of the gap between neighboring tracks α, i.e. $t_{i2}^p < t_{i1}^{p+1} - \alpha$.

2. Updating:

for $t = t_{i1}^p, \ldots, t_{i2}^p$ **do**

    if $\exists \mathbf{x}_t^i \in \Omega$ at time t **then**

        for $\epsilon = -1, +1, \ldots, -\alpha, +\alpha$ **do**

            if $\exists \mathbf{x}_{t+\epsilon}^a$ or $\mathbf{x}_{t+\epsilon}^b \in \mathbf{T}^0$ at time $t + \epsilon$ **then**

- Set $\mathbf{x}_{t+\epsilon}^a, \mathbf{x}_{t+\epsilon}^b$ as the nearest temporal neighbors.
- Apply pair-wise linking only on $\{\mathbf{x}_t^i \in \Omega\}$ and their nearest temporal neighbors $\mathbf{x}_{t+\epsilon_a}^a, \mathbf{x}_{t+\epsilon_b}^b \in \mathbf{T}^0$, determine $\mathbf{x}_t^a, \mathbf{x}_t^b$.
- Update current tracks $\mathbf{T}$ by $\mathbf{T} = \mathbf{T}^0 \bigcup \{\mathbf{x}_t^a, \mathbf{x}_t^b\}$.

            end if

        end for

    end if

end for

Output: $\widehat{\mathbf{T}}$.

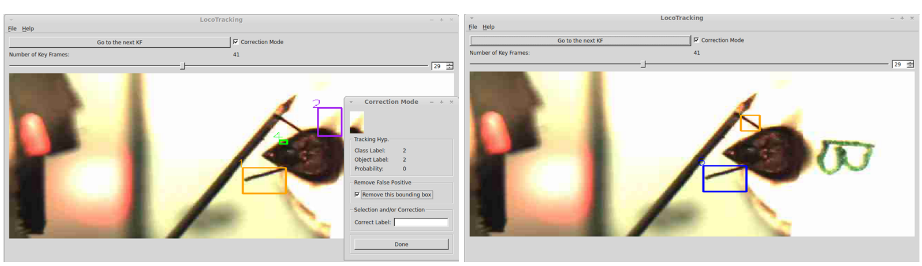

Figure 9: Two snapshots of the GUI: initial tracking hypotheses on a KF (left) and user corrected labels (right). *LocoTracker* enables users to correct tracking errors including mismatches, false positives and false negatives.

are listed in Table 2, including length (number of tested frames), imaging resolution (pixels per μm), frame-rate (frames per second), **GT** (number of ground truth tracks) and **UO** (unobserved objects). Particularly, the ratio of **UO** is measured as $\frac{no.\ of\ frames\ that\ contain\ unobserved\ objects}{no.\ of\ frames}$ to indicate the tracking gaps due to complicated motion patterns of body parts (e.g. the antenna above the insect head, or the proboscis not extended). The higher the value is, the more tracking gaps the video presents.

Table 2: The characteristics of tested videos

Insect	Length	Imaging Res.(pix/μm)	Framerate (f/s)	**GT**	**UO**
Bee	8222	39	60	5	0.50±0.11
Ant	430	22	30	4	0.13

5.2. Experimental Results

LocoTracker is tested in terms of practicability and accuracy. We measure the practicability in two ways: 1) processing time of automated computation and user correction, and 2) the trade-off between human effort and tracking accuracy. Regarding accuracy, results of our algorithm are compared with some state-of-the-art tracking methods as well as ground truth. Ground truth is manually annotated by a student in our group.

5.2.1. Practicability

The complexity of the algorithm is measured by processing time. We record the average running time for automated computation parts (Section 3.1, 3.2, 4.1, 4.2) and the user correction time. For the running time, it takes about 0.1 seconds per frame. For recording the user correction time, the other student tested LocoTracker and it takes about 8 seconds to correct all object labels on each KF. The average of user correction time over the whole video is about 0.8 seconds per frame, thus the additional human labor is tolerant. At each iteration given the user correction for requested KFs, computing Equation (7) takes less than 0.1 second. Therefore, the response time of the software between two consecutive user corrections is trivial. For comparison, we tested the established software Zootracer [51], which also provides user correction, on bee videos. It is designed based on the prior that the displacement between adjacent frames is small and the appearance gradually changes [27]. It is a single target tracker, which takes about 6 seconds per object per frame, as user correction is required for most of the video frames.

The trade-off between human intervention and tracking accuracy is tested on bee videos. Figure 10a shows the convergence of the iterative KF estimation and annotation query (Section 4.3). The KF ratio ($KF\ ratio = \frac{no.\ of\ KFs}{no.\ of\ frames}$) depends on the difficulty of tracking: more KFs are estimated for more challenging videos. For all tested videos, the main workload concentrates in the first 5 iterations. Figure 10b shows the accuracy improvement versus the average annotation time at the 0th (before user correction), 1st, 3rd and final iteration. The accuracy is measured as the ratio of tracking errors **TE** (i.e. the number of incorrectly labeled frames) defined in [52]. The **TE** for all bee videos drops below 0.05 at the final interation, while additional annotation time is about 1 second on each frame. In summary, the **TE** is 0.02 ± 0.01 for all tested videos, with the user correction only at the KF ratio as 0.14 ± 0.02 and additional annotation time.

5.2.2. Accuracy

Sub-task 1:

We compare our tracking method with several state-of-the-art association based tracking and category free tracking methods.

First, our method is compared with the established software Ctrax [23] and our base tracker [33] that estimates assignment automatically. Ctrax is designed for tracking multiple Drosophila adults, but cannot tackle the situations when the number of target is not constant and if occlusions are too complex [53]. Identity switch errors occur in the cases of false detection, presence of proboscis and occlusions or merges. We tested three different methods on one of the bee videos and an ant video for comparison. Ctrax is not applicable for tracking

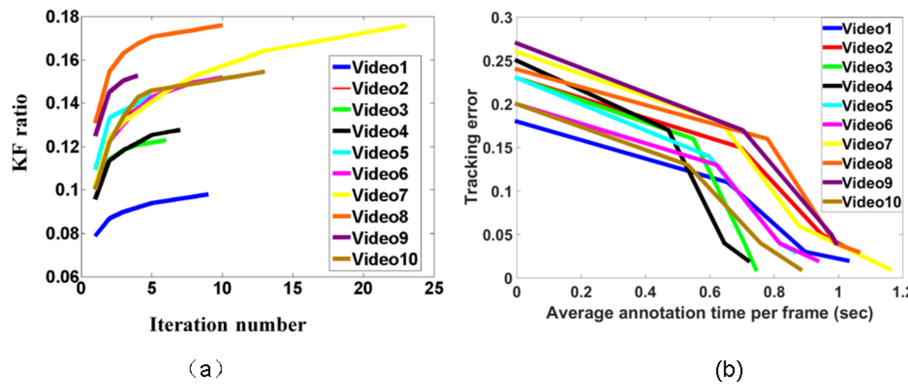

Figure 10: (a) KF ratio vs. Iteration number of ten tested videos: the user query stops at a KF ratio of 0.1 $\sim$ 0.18, and the KF ratio drops dramatically within five iterations. (b) Tracking error vs. annotation time: The **TE** of all bee videos drops below 0.05 at the final interation, while additional annotation time is about 1 second on each frame.

ant's antennae, as they do not fit the shape prior of Ctrax. The output of the bee video by Ctrax contains only the tracks of two antennae, and assumes errors in tracking other body parts, thus only these two targets are taken into account in Table 3.

Table 3: **TE** of three methods on different insect videos

	Ctrax [23]	Base tracker [33]	Ours
Bee	0.73	0.10	0.02
Ant	\	0.14	0.02

Second, we tested the state-of-the-art category free tracking methods (CT [35], MTT [54], SPOT [55] and TLD [56]) and ours on the same video. The tested codes are provided by the authors. With the intitial annotated right (orange colored) and left antenna (blue colored), the tracking results at frames $\{3, 11, 43\}$ for first three methods and ours are shown in Figure 11. The different tracking methods are denoted using different line types. All the compared methods start to drift from the right antenna at frame 3, and lose both antennae at frame 43, even when no interaction of targets presents. TLD fails to track the right antenna at the second frame, because the number of valid feature points drop from 100 to 8. Besides, the median of forward-backward error is too large (70 pixels). Its detector outputs two BBs with similar confidence, so it terminates both tracking and detection in the following frames. This indicates that category free tracking methods are not applicable for tracking insect body parts from a low frame rate video, as temporal correlation is too weak to predict the position of target at the current frame given the previous frame.

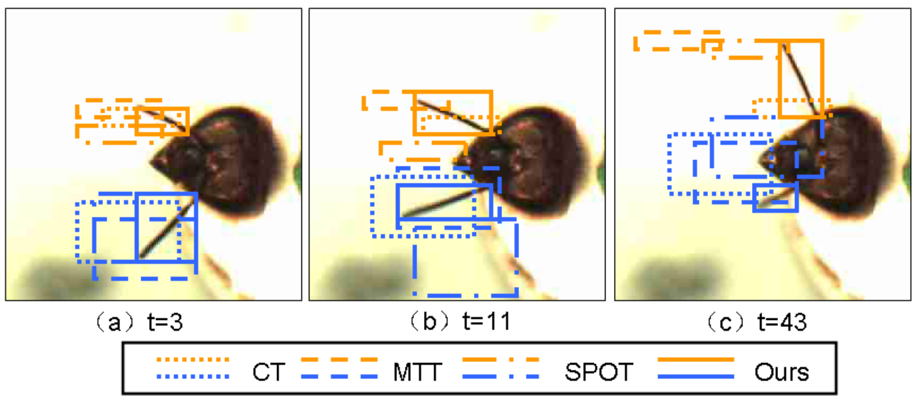

Figure 11: Results of four tracking methods.

Sub-task 2:

The final tracking results are the position of the tip of each object. Table 4 shows results for various flavors of our algorithm. To further show the robustness of our method, we list the ratio of detection errors after preprocessing described in Section 3.1, including *merged detections*, *occluded detections*, *false negatives* (FN) and *false positives* (FP). It is seen that the estimated positions of tips by our approach are very close to the ground truth. The mean of position error of all objects is merely 5 to 8 pixels, which are small compared to the average size of the bee's head (the size of Figure 2a is 180×280 pixels). The exact position of the tip of the ant's antennae is ambiguous, because the motion blur is more severe (see the right antenna in Figure 5a) due to a lower frame-rate.

To show how well the tracks are linked, we follow [15] to use the track completeness factor **TCF** as measurement. A **TCF** of 1 is the ideal indication that the final tracks completely overlap with the ground truth. The **TCF** for most objects are above 0.93, except for the proboscis, as it has the highest occluded detection ratio. If an object is occluded, it does not make sense to estimate its position. This indicates the advantage of the proposed approach in linking tracks in merged conditions, which produces the tracks comparable to manual "point and click" results.

To show the advantage of our method in fulfilling two sub-tasks, we illustrate ten consecutive sample frames of the final tracking results in Figure 12. This is an extreme case of merge condition. As the result of sub-task 1, the label of each BB is estimated. The correct labels are given in (e) with the help of user correction, even though they do not follow the ascending order we assumed. Given the reliably labeled BBs, the positions of proboscis tips in (a)-(i) in merged BBs are estimated with an acceptable precision by our track linking approach. As the final outputs, three trajectories of tips are drawn on one of the video frames for visualization, as shown in Figure 13.

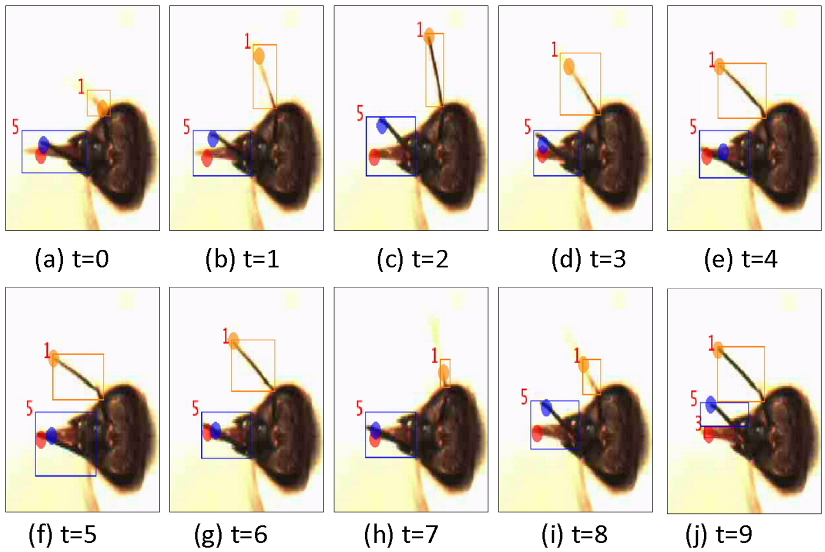

Figure 12: Ten consecutive sample frames of the final tracking results under merge condition.

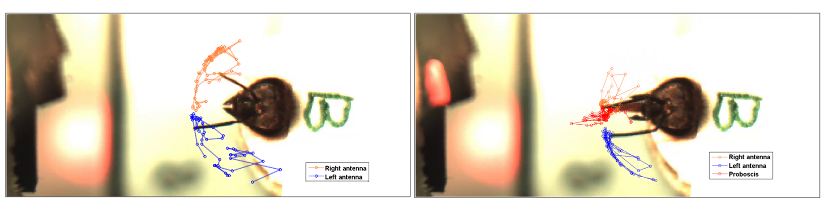

Figure 13: Three trajectories of tips of 100 frame in the videos shown in Fig. 4 are drawn on one of the video frames: The orange dots denote the tips of right antenna, red for proboscis and blue for left antenna.

Table 4: Comparison of our method with ground truth.

Object name	R-Antenna	Proboscis	L-Antenna
Average position error (pixels)	5.3±0.5	8.3±3.2	6.4±0.73
TCF	0.93±0.03	0.58±0.16	0.95±0.03
Merged (%)	0.57±0.47	13±5.1	0.93±1.0
Occluded (%)	0.95±0.93	14±2.4	2.5±2.2
FN (%)	5.5±3.2	2.0±2.2	5.80±2.8
FP (%)	0.16±0.19	0.00±0.00	1.4±0.90

5.2.3. Anatomical Model

To validate the analysis about the anatomical model of insect body parts in Section 3.2, we list the classification results of Section 4.1 for six shape descriptors in Table 5 tested on Bee videos. Similar to most biomedical data, the class distribution is skewed. For example, the number of mandibles is much smaller than antennae. The unbalanced data problem causes that the minority class is more likely to be misclassified than the majority class. Taking the unbalanced classes into account, the mean and variance are calculated by treating each class with equal weight. As shown in Table 5, EHD produces the highest mean value of classification results and the lowest variance, thus is selected for our work.

Table 5: Classification results in Section 4.1 for six shape descriptors

	Antenna	Mandible	Proboscis	Mean	Variance
EHD [43]	0.96	0.66	0.55	0.72	0.05
IEHD [44]	0.16	0.43	0.95	0.51	0.16
GF [45]	0.99	0.34	0.67	0.67	0.11
SSH [46]	0.99	0.44	0.43	0.62	0.10
FD [47]	0.44	0.32	1.00	0.59	0.13
ISH [44]	0.99	0.33	0.47	0.60	0.12

6. Conclusion

In this paper, we proposed a method aiming at achieving high precision of tracking multiple targets by minimizing additional human effort for correction. Our method integrates a frame query approach, enabling users to correct the erroneous tracking hypotheses and making full use of the user input to optimize the final results. This is a preferable approach to traditional track-and-then-rectification scheme, as it does not require an additional round of manual evaluation and correction while guaranteeing a high precision of the tracking results. Particularly, an important aspect of this system is its ability to estimate the trajectories of insect body parts at pixel precision even in merge conditions. The practicability and tracking performance of this system is validated on challenging video datasets for insect behavioral experiments.

Acknowledgments:

The authors would like to thank Ci Wang for the helpful discussion on the paper, Le Duan, Manuel Wildner and Deepika Banakar for help with software development, testing and evaluation, Oliver Kühn and Christopher Dieter Reinkemeier for the video acquisition, Cathrin Warnke for her proof reading. This work was funded by Bundesministerium für Bildung und Forschung (01GQ0931 to PS and CGG), with partial support also from the National Natural Science Foundation of China under Grants 61302121, 61403182, and 61363046.

[1] S. Sauer and M. Kinkelin and E. Herrmann and W. Kaiser, The dynamics of sleep-like behaviour in honey bees, Journal of Comparative Physiology A 189 (8) (2003) 599–607.

[2] J. Erber and B. Pribbenow and A. Bauer and P. Kloppenburg, Antennal reflexes in the honeybee: Tools for studying the nervous system, Apidologie 24 (1993) 238–296.

[3] V. Rehder, Quantification of the honeybee's proboscis reflex by electromyographic recordings, Journal of Insect Physiology 33 (7) (1987) 501–507.

[4] B. H. Smith, C. I. Abramson, T. R. Tobin, Conditional withholding of proboscis extension in honeybees (apis mellifera) during discriminative punishment, Journal of Comparative Psychology 105 (4) (1991) 345–356.

[5] M. A. Chabaud, J. M. Devaud, M. H. Pham-Delègue, T. Preat, L. Kaiser, Olfactory conditioning of proboscis activity in drosophila melanogaster, Journal of Comparative Physiology A 192 (12) (2006) 1335–1348.

[6] F. J. Guerrieri and P. d'Ettorre, Associative learning in ants: Conditioning of the maxilla-labium extension response in camponotus aethiops, Journal of Insect Physiology 56 (1) (2010) 88–92.

[7] A. Pérez-Escudero, J. Vicente-Page, R. C. Hinz, S. Arganda, G. G. D. Polavieja, Idtracker: Tracking individuals in a group by automatic identification of unmarked animals, Nature Methods 11 (7) (2014) 743–748.

[8] H. Pistori, V. V. V. A. Odakura, J. B. O. Monteiro, W. N. Gonçalves, A. R. Roel, J. D. A. Silva, B. B. Machado, Mice and larvae tracking using a particle filter with an auto-adjustable observation model, Pattern Recognition Letters 31 (4) (2010) 337–346.

[9] M. Shen, W. Huang, P. Szyszka, C. G. Galizia, D. Merhof, Interactive framework for insect tracking with active learning, in: International Conference on Pattern Recognition, IEEE, 2014, pp. 2733–2738.

[10] W. Luo, X. Zhao, T. K. Kim, Multiple object tracking: A review, arXiv preprint arXiv:1409.7618.

[11] Y. Wu, J. Lim, M. H. Yang, Online object tracking: A benchmark, in: Conference on Computer Vision and Pattern Recognition, IEEE, 2013, pp. 2411–2418.

[12] Z. Qin, C. R. Shelton, Improving multi-target tracking via social grouping, in: Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 1972–1978.

[13] C. Huang, B. Wu, R. Nevatia, Robust object tracking by hierarchical association of detection responses, in: European Conference on Computer Vision, Springer, 2008, pp. 788–801.

[14] B. Bose, X. Wang, E. Grimson, Multi-class object tracking algorithm that handles fragmentation and grouping, in: Conference on Computer Vision and Pattern Recognition, IEEE, 2007, pp. 1–8.

[15] A. G. A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, W. Hu, Multi-object tracking through simultaneous long occlusions and split-merge conditions, in: Conference on Computer Vision and Pattern Recognition, Vol. 1, IEEE, 2006, pp. 666–673.

[16] A. Veeraraghavan, R. Chellappa, M. Srinivasan, Shape-and-behavior encoded tracking of bee dances, Pattern Analysis and Machine Intelligence 30 (3) (2008) 463–476.

[17] T. Landgraf, R. Rojas, Tracking honey bee dances from sparse optical flow fields, FB Mathematik und Informatik FU (2007) 1–37.

[18] T. Balch, Z. Khan, M. Veloso, Automatically tracking and analyzing the behavior of live insect colonies, in: International Conference on Autonomous Agents, ACM, 2001, pp. 521–528.

[19] F. Ying, Visual ants tracking, Ph.D. thesis, University of Bristol (2004).

[20] S. Mujagić, S. M. Würth, S. Hellbach, V. Dürr, Tactile conditioning and movement analysis of antennal sampling strategies in honey bees (apis mellifera l.), Journal of Visualized Experiments (70) (2011) e50179–e50179.

[21] J. Voigts, B. Sakmann, T. Celikel, Unsupervised whisker tracking in unrestrained behaving animals, Journal of Neurophysiology 100 (1) (2008) 504–515.

[22] B. Risse, S. Thomas, N. Otto, T. Löpmeier, D. Valkov, X. Jiang, C. Klämbt, Fim, a novel ftir-based imaging method for high throughput locomotion analysis, PloS one 8 (1) (2013) e53963.

[23] K. Branson, A. A. Robie, J. Bender, P. Perona, M. H. Dickinson, High-throughput ethomics in large groups of drosophila, Nature Methods 6 (6) (2009) 451–457.

[24] L. Fiaschi, F. Diego, K. Gregor, M. Schiegg, U. Koethe, M. Zlatic, F. A. Hamprecht, Tracking indistinguishable translucent objects over time using weakly supervised structured learning, in: Conference on Computer Vision and Pattern Recognition, IEEE, 2014, pp. 2736–2743.

[25] A. Yao, J. Gall, C. Leistner, L. V. Gool, Interactive object detection, in: Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 3242–3249.

[26] J. Yuen, B. Russell, C. Liu, A. Torralba, Labelme video: Building a video database with human annotations, in: International Conference on Computer Vision, IEEE, 2009, pp. 1451–1458.

[27] A. Buchanan, A. Fitzgibbon, Interactive feature tracking using kd trees and dynamic programming, in: Conference on Computer Vision and Pattern Recognition, Vol. 1, IEEE, 2006, pp. 626–633.

[28] C. Vondrick, D. Ramanan, Video annotation and tracking with active learning, in: Neural Information Processing Systems, 2011, pp. 28–36.

[29] C. Vondrick, D. Ramanan, D. Patterson, Efficiently scaling up video annotation with crowdsourced marketplaces, in: European Conference on Computer Vision, Springer, 2010, pp. 610–623.

[30] P. Kaewtrakulpong, R. Bowden, An improved adaptive background mixture model for realtime tracking with shadow detection, in: European Workshop on Advanced Video Based Surveillance Systems, Springer, 2001, pp. 135–144.

[31] D. S. Pham, O. Arandjelović, V. Svetha, Detection of dynamic background due to swaying movements from motion features, Image Processing 24 (1) (2015) 332–344.

[32] M. Shen, P. Szyszka, C. G. Galizia, D. Merhof, Automatic framework for tracking honeybee's antennae and mouthparts from low framerate video, in: International Conference on Image Processing, IEEE, 2013, pp. 4112–4116.

[33] M. Shen, P. Szyszka, O. Deussen, C. G. Galizia, D. Merhof, Automated tracking and analysis of behavior in restrained insects, Journal of Neuroscience Methods 239 (2015) 194–205.

[34] Y. Huang, I. Essa, Tracking multiple objects through occlusions, in: Conference on Computer Vision and Pattern Recognition, IEEE, 2005, pp. 1051–1058.

[35] K. Zhang, L. Zhang, M. H. Yang, Real-time compressive tracking, in: European Conference on Computer Vision, Springer, 2012, pp. 864–877.

[36] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.

[37] R. Martin, O. Arandjelović, Multiple-object tracking in cluttered and crowded public spaces, in: Advances in Visual Computing, Springer, 2010, pp. 89–98.

[38] O. Arandjelović, Contextually learnt detection of unusual motion-based behaviour in crowded public spaces, in: Computer and Information Sciences II, Springer, 2012, pp. 403–410.

[39] B. D. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, in: International Joint Conference on Artificial Intelligence, Vol. 2, 1981, pp. 674–679.

[40] C. Harris, M. Stephens, A combined corner and edge detector, in: Alvey Vision Conference, Manchester, UK, 1988, pp. 147–151.

[41] R. Arandjelović, A. Zisserman, Smooth object retrieval using a bag of boundaries, in: International Conference on Computer Vision, IEEE, 2011, pp. 375–382.

[42] O. Arandjelović, Gradient edge map features for frontal face recognition under extreme illumination changes, in: British Machine Vision Association Conference, BMVA Press, 2012, pp. 1–11.

[43] H. Frigui, P. Gader, Detection and Discrimination of Land Mines in Ground-Penetrating Radar Based on Edge Histogram Descriptors and A Possibilistic K-Nearest Neighbor Classifier, Fuzzy Systems 17 (1) (2011) 185–199.

[44] C. Li, K. Shirahama, M. Grzegorzek, Application of Content-Based Image Analysis to Environmental Microorganism Classification, International Journal on Biocybernetics and Biomedical Engineering 35 (1) (2015) 10–21.

[45] S. Z. Li, Shape matching based on invariants, in: Shape Analysis, Progress in Neural Networks, Ablex, Norwood, NJ, 1999, pp. 203–228.

[46] D. Zhang, G. Liu, Review of Shape Representation and Description Techniques, Pattern Recognition 37 (1) (2004) 1–19.

[47] D. Zhang, G. Lu, A Comparative Study of Curvature Scale Space and Fourier Descriptors for Shape-Based Image Retrieval, Journal of Visual Communication and Image Representation 14 (1) (2003) 39–57.

[48] L. Zhang, Y. Li, R. Nevatia, Global data association for multi-object tracking using network flows, in: Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.

[49] G. Guo, C. R. Dyer, Learning from examples in the small sample case: Face expression recognition, Systems, Man, and Cybernetics, Part B: Cybernetics 35 (3) (2005) 477–488.

[50] J. Munkres, Algorithms for assignment and transportation problems, Journal of the Society for Industrial and Applied Mathematics 5 (1) (1957) 32–38.

[51] A. Buchanan, A. Fitzgibbon, Zoo tracer (2014).
URL http://research.microsoft.com/zootracer

[52] C. R. del Bianco, F. Jaureguizar, N. Garcia, Bayesian visual surveillance: A model for detecting and tracking a variable number of moving objects, in: International Conference on Image Processing, IEEE, 2011, pp. 1437–1440.

[53] A. I. Dell, J. A. Bender, K. Branson, I. D. Couzin, G. G. D. Polavieja, L. P. J. J. Noldus, A. Pérez-Escudero, P. Perona, A. D. Straw, M. Wikelski, Automated image-based tracking and its application in ecology, Trends in Ecology & Evolution 29 (7) (2014) 417–428.

[54] T. Zhang, B. Ghanem, S. Liu, N. Ahuja, Robust visual tracking via multi-task sparse learning, in: Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 2042–2049.

[55] L. Zhang, L. V. D. Maaten, Structure preserving object tracking, in: Conference on Computer Vision and Pattern Recognition, IEEE, 2013, pp. 1838–1845.

[56] Z. Kalal, J. Matas, K. Mikolajczyk, Pn learning: Bootstrapping binary classifiers by structural constraints, in: Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 49–56.