2014 22nd International Conference on Pattern Recognition

Interactive Framework for Insect Tracking with Active Learning

Minmin Shen
INCIDE center
University of Konstanz
Konstanz, Germany 78464
School of Software Engineering
South China University of Technology
Guangzhou, China 510640
minmin.shen@uni-konstanz.de

Wei Huang
Dept of Computer Science
Nanchang University
Nanchang, China 330031
huangwei@ncu.edu.cn

Paul Szyszka
and C. Giovanni Galizia
Institute of Neurobiology
University of Konstanz
paul.szyszka@uni-konstanz.de
giovanni.galizia@uni-konstanz.de

Dorit Merhof
Institute of Imaging
& Computer Vision
RWTH Aachen University
Aachen, Germany
Dorit.Merhof@lfb.rwth-aachen.de

***Abstract*—Extracting motion trajectories of insects is an important prerequisite in many behavioral studies. Despite great efforts to design efficient automatic tracking algorithms, tracking errors are unavoidable. In this paper, we propose general principles that help to minimize the human effort required for accurate multi-target tracking in the form of applications that can track the antennae and mouthparts of a honey bee based on a set of low frame rate videos. This interactive framework estimates which key frames will require user correction, i.e. those that are used for user correction, which are used for 1) incrementally learning an object classifier and 2) data association based tracking. To this framework we apply a standard classification algorithm (i.e. naive Bayesian classification) and an association optimization algorithm (i.e. Hungarian algorithm). The precision of tracking results by our framework on real-world video data is above 98%.**

***Keywords*-multi-object tracking; insect tracking;**

I. Introduction

Behavior analysis of insects has gained attraction in recent years. Many research efforts in biomimicry have been focused on applying biological models of insect behavior to many fields ranging from information technology and electronic engineering to social science. In order to study insect behavior, hours of video data of insects is recorded. As a preliminary step, the motion of insects is inferred as annotated bounding boxes (BBs) at each frame of the video. However, the task of manually labeling insect motion calls for operators who have undergone intensive and time-consuming training, but who can easily introduce bias into the analyzed data. Therefore, in order to perform fine-grained analysis, reliable and accurate motion tracking is required.

This is generally studied as the tracking problem in the field of computer vision [1]. Many efficient semi-supervised tracking algorithms have been reported for multi-target tracking such as the well known Multi-Hypothesis Tracking [2] and Joint Probabilistic Data Association Filters [3], which propose inference over multiple objects by tracking over a longer period of time in contrast to frame-by-frame tracking. However, this category of approaches suffers from an exponentially growing search space with the number of frames. Data association based tracking (DAT) algorithms have been proposed [4–10] as a means to overcome the drawback of long tracking gaps. In this category of algorithms, frame-to-frame linking is firstly applied to generate reliable tracklets, which are then linked by data association techniques to generate optimal tracks. In spite of intensive efforts to improve the accuracy by exploiting different association optimization approaches, such as Hungarian algorithm [8], Linear Programming [5], and cost-flow network [7], tracking errors are unavoidable in the outputs of automatic tracking approaches. In practice, given the output of a tracking algorithm, human effort is required to rectify the annotated videos manually. In the following, this is referred to as a "track-and-then-rectification" approach. The "track-and-then-rectification" approach requires the user to go through all the tested videos, which is unrealistic for actual behavioral studies.

In insect behavioral studies, many videos of the same type of experiment are used for analysis. For example, in our paper, dozens of bees are used for each experiment, and from each video their motion has to be analyzed. Generally, manual labeling is required for training samples of each video due to the varying feature characteristics, which is a cumbersome and undesirable process. Moreover, the number of each class is unbalanced in our case - a common occurence in biological or medical data. Thus the selection of training samples is a further challenge to be met. We wish to make the most use of labeled data, and propose an active samples selection scheme allowing users to pick the most representative samples without viewing through the whole video. We consider this problem as an instance of active learning, which aims to minimize human annotation by requesting labels for only a portion of the training samples [11, 12]. A number of active learning approaches have been proposed that address the classification problem by training a classifier with maximum accuracy [11, 13, 14], given a fixed annotation budget. However, these approaches are not applicable for our annotation task, where all the labels of fixed video data are required.

The objective of our framework is to provide all of the correct labels of a set of insect videos with minimal human effort. Given the detection responses, the proposed framework includes the following two stages: (1) classification of moving objects, (2) tracking and key frame (KF) query, as shown in Figure 1. At the beginning, a small set of training samples is manually labeled on one of the videos and used to train

1051-4651/14 $31.00 © 2014 IEEE
DOI 10.1109/ICPR.2014.471

the initial classifier. Then the proposed system interactively estimates tracks and queries the user to annotate only on the frames with high uncertainty (i.e. KFs). The user annotation is used as (1) resource of new training samples and (2) for updating the DAT tracker. The newly added training samples are actively selected from those user corrected labels that are considered to be the most informative. The classifier is incrementally learned from the new set of training samples, and will be used for the next video. The tracks of the current video are iteratively refined until no more user queries are required. This framework for insect tracking can also be generalized for other applications of motion tracking such as visual surveillance [15].

We highlight our contribution in this paper as follows:

- We suggest interactive tracking instead of a "track-and-then-rectification" approach for acquiring accurate video annotations for further analysis. We show how this framework works with data association techniques to fulfil multi-target tracking, which can be extended by exploiting more advanced techniques. The principle of interactive tracking and user annotation presented in this paper is applicable for other DAT algorithms.
- We construct an approach for measuring annotation cost and estimate which KFs require user annotation in order to correct the tracking hypothesis. It defines the annotation cost as a function of uncertainty, which enables users to achieve a trade-off between performance and human effort.
- Specifically, we apply this framework using standard classification and data association techniques to track individual bee's antennae and mouthparts. This is an appropriate example application, as we have to handle the problems of long tracking gaps, similar appearance of the targets, and identity switching of targets.

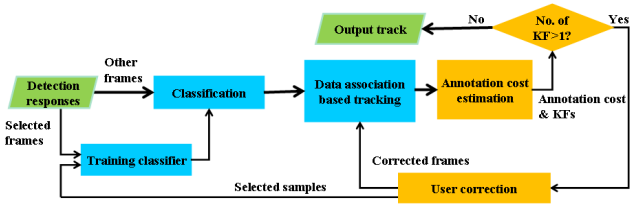

Figure 1. Flowchart of the proposed tracking framework

II. Related Work

To minimize the human effort required to select the most informative samples as training samples, incremental learning [12, 16] has been proposed. A re-sampling approach is exploited using the Naive Bayesian Classifier to select the samples that produce high error scores by the current classifier in [16]. This approach improves the classification performance from unbalanced biomedical data, but an additional step of ranking the samples is required. In [12], the detector is trained using initial training samples and generates hypotheses for the test image. The incorrect hypotheses are then rectified by the user and used as new samples for training a new detector. This idea is conceptually similar to our work, but it focuses on detection on single images, and may require the user to examine the entire video for a complete labeling.

There has been some work on interactive tracking, but this either requires users to view the whole video [17], or to refrain from focusing on frame query techniques [18]. The work most conceptually similar to ours is proposed in [19]. It extends the tracker in [20] by estimating more KFs for user annotation in order to improve the tracking accuracy. However, since the KF estimation scheme in [19] 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 2).

III. Statement of Problem

Although the application scenario of our framework addresses a particular task, the challenges to be addressed characterize a generic tracking problem. Moreover, new challenges are encountered in this application. We summarize the challenges of tracking the movements of mouthparts and antennae of individual bees from low frame-rate videos as follows: 1) long tracking gaps, 2) varying appearance of an object, 3) the fact that different objects with similar appearance and position appear in turns and 4) detection errors are produced by standard moving object detectors such as false, missing, splitted or merged BBs. We define $X_j = \{x_{i,j}\}$ to be a set of BBs at the jth frame at pixel position i, which is generated by a standard moving object detector. The number of frames in a video is denoted as T, i.e. $1 \leq j \leq T$. Our goal is to assign each $x_{i,j}$ a label $y_{i,j}$, where $y_{i,j} \in \{$1:right antenna; 2:right mandible; 3:proboscis; 4:left mandible; 5:left antenna; 6:false positive$\}$.

An example is shown in Figure 2, where a set of detection responses as unordered BBs are generated by subtracting the background, which uses a Gaussian Mixture Model (GMM). We use different colors to denote the expected label for better visualization. It can be seen that the proboscis (i.e. label 3) varies its appearance (see Figure (e)(i)(j)), and looks similar to mandibles in (e)(f)(g). Particularly, the ambiguous identity of a merged BB (a BB including multiple objects) needs to be determined by a user according to the biological experimental setup. An occluded, missing or merged BB produces a tracking gap, which renders it unsuitable for frame-by-frame tracking approaches such as particular filter based algorithms. Moreover, the sudden change of labels of a proboscis on (g), (h), (i) is unlikely to be accurately identified by state-of-the-art DAT approaches, as it is neither an occluded target nor a missing one. These issues make the tracking problem rather challenging.

IV. Data Association based Tracking

Our choice of base tracker is the DAT algorithm proposed in [21], which is briefly summarized in this section. In theory, any DAT approach is applicable in our framework. We will extend it in Section V to construct an efficient active learning algorithm. The linking between Section IV (blue blocks) and Section V (yellow blocks) is illustrated in Figure 1.

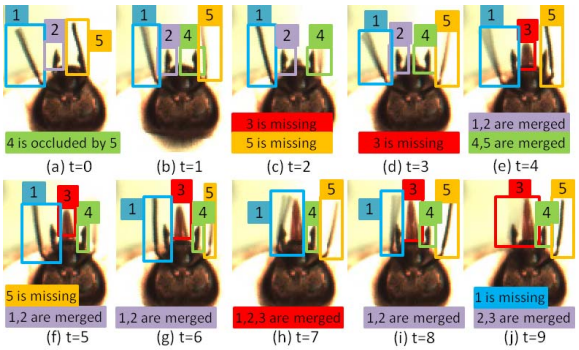

Figure 2. The detection responses at 8 consecutive frames including occluded, missing or merged BBs. Identification of each BB, shown in a different color, is a challenging process.

Determining the correspondence of multiple BBs through T frames is rather difficult, given occluded, missing or false detection. In addition, merged or splitted detection makes our tracking problem more challenging. Generally, a global optimization $\widehat{\mathbf{Y}} = \{\widehat{Y_j}, j = 1, \ldots, T\}$ is found by minimizing the following cost function

$$\begin{aligned} \widehat{\mathbf{Y}} &= \arg\max_{\mathbf{Y}} P(\mathbf{Y}|\mathbf{X}) \\ &= \arg\max_{\mathbf{Y}} \prod_{j=1}^{T} P(X_j|Y_j)P(\mathbf{Y}) \end{aligned} \tag{1}$$

where $Y_j = \{y_{1,j}, \ldots, y_{i,j}, \ldots, y_{n_j}\}$ is an ordered collection of the assigned labels, and n_j is the number of the detected objects in the jth frame. The observations and their estimations are $\mathbf{X} = \{X_j, j = 1, \ldots, T\}$ and $\mathbf{Y} = \{Y_j, j = 1, \ldots, T\}$, respectively. Here we assume that the likelihood $P(X_j|Y_j)$ is temporal independent. The labels are initially estimated at frame level (Section IV-A and Section IV-B), and then temporal correlation is considered for refinement (Section IV-C).

A. Object Classfication

Generally, the first task in tracking is object classification, e.g. classifying different moving objects as human, vehicles or other by way of visual surveillance. In our case, $x_{i,j}$ is assigned a class label $c_{i,j}$, where $c_{i,j} \in \{1$:antenna; 2:mandible; 3:proboscis$\}$. For antenna and mandible, their details (either on the left hand side or the right hand side) are further differentiated in the tracking step.

In order to build an appearance model, we use a feature vector $\mathbf{z}_{i,j}$ to represent each object in terms of its location, shape, texture and speed. Seven features are extracted from a BB, including (1) distance between the nearest vertex and mandible, (2) distance between the furthest vertex and the tongue line, (3) area of the object, (4-5) motion vector, (6) area of top-hat filtered output, and (7) a logical variable indicating whether the mandible is within the BB.

Object classification generates label $c_{i,j}$ and the corresponding class probability $P(c_{i,j}|\mathbf{z}_{i,j})$ for each BB. Similar to most biomedical data, the class distribution of $\mathbf{z}_{i,j}$ is skewed. For example, the number of mandibles is much smaller than antennae. The unbalanced data problem means that the minority class is more likely to be misclassified than the majority class. We will solve this problem by incremental learning with a Naive Bayesian classifier in Section V-A. In spite of its naive design and simple assumptions, the Naive Bayesian classifier performs well in our framework.

B. Estimation of Benchmark Frames

Based on the output of object classification, we exploit the appearance information of a bee, i.e. position and ordering of $x_{i,j}$, to assign the label $y_{i,j}$. The objects are assumed to be ordered in a certain sequence, so the likelihood $P(X_j|Y_j)$ is estimated following the assumption that Y_j should be ordered in an ascending manner, i.e.

$$P(X_j|Y_j) = \begin{cases} 0 & \text{if } m_{1,j} > 2 \text{ or } m_{2,j} > 2 \text{ or } m_{3,j} > 1 \\ & \text{or } \exists y_{i,j} > y_{k,j}, \forall k < i \\ 1 & \text{if } Y_j = \{1, 5\} \text{ or } Y_j = \{2, 4\} \\ \binom{5}{n_j} & \text{otherwise} \end{cases} \tag{2}$$

where $m_{k,j}$ is the number of $\{c_{i,j}|c_{i,j} = k\}$ at the jth frame. This is considered a priori knowledge to determine the *benchmark frames*, i.e. the frames at which no user rectification is required.

Although this a priori knowledge may be particular in the task of tracking the mouthparts and antennae of a bee, the selection of benchmark frames is a general principle that can be used as part of a general tracking framework. For example, frames initially annotated by a user [19, 20] can be used as benchmark frames.

The frames with likelihood $P(X_j|Y_j) = 1$ are assumed to be correctly labeled. Particularly, we define a certain portion of them as the benchmark frames Y_b, where $b \in \Psi : P(X_j|Y_j) = 1 \ \& \ P(X_{j\pm1}|Y_{j\pm1}) \neq 1$.

C. Constrained Frame-to-Frame Linking

Generally, $P(Y_j)$ in Equation (1) is modelled as a Markov chain, and the probability of estimating the labels at the current frame Y_j depends on the previous frame Y_{j-1}. However, the errors in a previous frame will propagate to the following frames. For this reason, we redefine $P(Y_j)$ to guarantee that only the benchmark frames will help to rectify the labels of their neighboring frames:

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

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

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

The frame-to-frame linking between X_b and $X_{b\pm1}$ is found by forming a $n \times n$ cost matrix $\mathbf{M} = \{M_{i,k}\}$ with

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

where $n = \max(\mathrm{n_b}, \mathrm{n_{b\pm1}})$ and the sign "$\mapsto$" denotes a correspondence. The Hungarian algorithm [22] is applied to find optimal linking by minimizing the linking cost.

V. User Interactive Annotation

A. Incremental Learning for Classification

Practically, manual annotation for a small portion of each video is used to provide training samples in classification for behavioral studies [23]. A typical procedure is to train the classifier using selected training samples, while annotation of the rest of the video is completed automatically. Thus, the appropriate size of the training samples is determined empirically.

In this paper, all videos originated from the same type of experiment, but featuring different bees. Instead of selecting training samples for each video, a new strategy is exploited here: we start with an insufficient size of training samples from one of the videos, and train the initial Naive Bayesian Classifier NB_{init}. Then the KFs are estimated by the proposed tracking framework in Figure 1. The informative samples are actively selected from the KFs when the user is asked to rectify the incorrect hypotheses. These are subsequently used to update NB_{init} and generate a new classifier NB_{incr}. Before the user annotates the next video, the classifier NB_{incr} is applied to generate tracking hypotheses. With more videos being annotated interactively, using this framework, the training samples are collected incrementally until the predefined maximum number is achieved. The prediction performance will be improved regardless of the initial training samples, as only incorrect hypotheses are added into the training samples.

B. Constrained Tracking with Active Learning

According to Equation (1) and (3), $\widehat{\mathbf{Y}}$ is the current optimal estimation for the labels given a set of benchmark frames in $\{Y_b, b \in \Psi\}$ estimated in Section IV-B. The estimated tracks are refined by adding more benchmark frames Y_b, which are obtained via query requests for user annotation, i.e. KFs. We wish to minimize the number of KFs to optimize the final estimation. Let us redefine Equation (1) as

$$\begin{aligned}\widehat{\mathbf{Y}^*} &= \arg\max_{\mathbf{Y},\Psi} P(\mathbf{Y}|\mathbf{X}) \\ &= \arg\max_{\mathbf{Y},\Psi} \prod_{j=1}^{T} P(X_j|Y_j) \prod_{b\in\Psi} P(Y_{b\pm1}|Y_b)\end{aligned} \tag{6}$$

We solve Equation (6) by an Expection Maximization (EM) algorithm as follows:

E-step In the E-step, the new set of benchmark frames Ψ^* is determined by adding annotated KFs. We define the annotation cost of each frame to indicate the degree of "usefulness" of user annotation, in order to compute which frames should be added to Ψ^*. The higher the annotation cost at the jth frame, the more erroneous the label Y_j tends to be. It is natural to define the annotation cost as the probability of incorrect labeling

$$A(Y_j) = P_\epsilon = 1 - P(Y_j|X_j) \tag{7}$$

where

$$\begin{aligned}&P(Y_j|X_j) \\ &= \begin{cases} P(X_j|Y_j)\prod_{i,k} P(y_{i,j} \mapsto y_{k,j}) & j = b \pm 1 \\ P(X_j|Y_j) & \text{otherwise} \end{cases}\end{aligned} \tag{8}$$

As $A(Y_j)$ interprets the probability that Y_j is incorrectly labeled, it is straightforward for the users to set the threshold τ for choosing KFs with $A(Y_j) \geq \tau$ considering the trade-off between tracking accuracy and human effort. The KFs Y_s are defined as $s \in \Phi : P(X_{s-1}|Y_{s-1}) = 1 \;\&\; A(Y_s) \geq \tau$, which request user annotation by a graphical user interface (GUI) shown in Figure 4. Given the annotated frames in Φ and the updated probability $P(Y_s) = 1, \forall s \in \Phi$, the new set of benchmark frames Ψ^* is determined according to Section IV-B.

By way of example, "difficult" video frames (frame 370 - 530) belonging to a classical conditioning bee video are tested. As shown in Figure 3, the red bars indicate the tracking errors (TE) in $\widehat{\mathbf{Y}}$ in Equation (1), and the blue squares indicate the annotation cost. Here, it is possible to see the estimated annotation cost $A(Y_s)$ in all the frames where TE is greater than 0.5. This validates our active learning scheme. The set of KFs Φ is chosen by setting $\tau = 0.5$, which request user annotations (green stars). It is seen that the frames with TE all follow the KFs Φ, which help to rectify the TE in the following M-step.

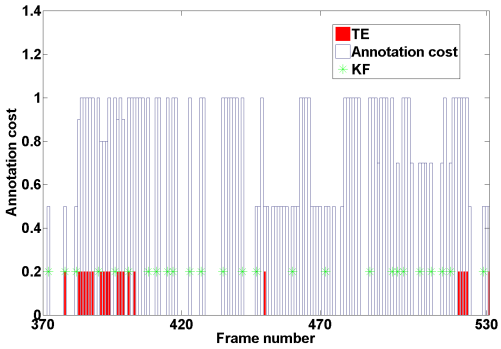

Figure 3. Frames with actual TE correspond to the ones with high annotation costs, which will be recified with the estimated KFs.

M-step Given the new set Ψ^* obtained from the E-step, let us rewrite Equation (6) as

$$\widehat{\mathbf{Y}^*} = \arg\max_{\mathbf{Y}^*} \prod_{i=1}^{T} P(X_j|Y_j) \prod_{b\in\Psi^*} P(Y_{b\pm1}|Y_b) \tag{9}$$

In the M-step, the current optimal labels $\widehat{\mathbf{Y}^*}$ are updated by solving Equation (9). The processing steps of our tracking framework are summarized as follows:

---

1) Object Classification: Train object classifier NB_{incr} with the incrementally learned training sample, and generate class labels $c_{i,j}$.

2) Estimation of Benchmark Frames: Given the class labels $c_{i,j}$ and a priori knowledge, assign the labels $\mathbf{Y}^0$. Compute the likelihood $P(X_j^0|Y_j^0)$ and determine the set of benchmark frames Ψ^0.
3) Constrained Frame-to-Frame Linking: Apply pair-wise linking only on the benchmark frames $Y_b^0, b \in \Psi^0$ and their temporal neighbors $Y_{b\pm1}^0$. Update the labels on $Y_{b\pm1}^0$ for $\mathbf{Y}^t$ and compute $P(Y_{b\pm1}^t|Y_b^t)$.
4) KF estimation and annotation query:
   - Compute $A(Y_j)$ given $P(X_j|Y_j)$ and $P(Y_{b\pm1}|Y_b)$ and estimate KFs $Y_s, s \in \Phi$.
   - Request for user annotations at the KFs. Obtain the updated set of benchmark frames Ψ^{t+1}, probability $P(X_j^{t+1}|Y_j^{t+1})$, $P(Y_{b\pm1}^{t+1}|Y_b^{t+1})$, and labels $\mathbf{Y}^{t+1}$.
   - If $\exists P(X_j^{t+1}|Y_j^{t+1}) < 1$, go to step 3; otherwise, output $\mathbf{Y}^{t+1}$.

VI. Experiments

We test our tracking framework on a set of challenging videos of a classical conditioning experiment, where an odorant is paired with a sugar reward to to study the associative learning of bees [24]. The sugar stick appearing in the video disturbs the modeling of the background of the GMM, resulting in longer tracking gaps and more detection errors than in the scenario shown in [21]. Thus, the tracking problem here is more challenging. In this experiment, six videos of "video 1∼6" are tested, each of which has a total number of frames $T = 800$, with a frame rate of 60 fps and a frame size of 480×720.

The incrementally learned Naive Bayesian Classifier is tested on "video 1", "video 2" and "video 3", while some objects $x_{i,j}$ are selected as training samples from "video 4". The classification procedure is applied in accordance with Section V-A. Only 5 samples for each class are selected to train the initial classifier, then another 55 samples are collected from incorrect hypotheses on KFs to train the new classifier NB_{incr}. Table I shows the improvement of NB_{incr} over NB_{init} on three videos, in term of the precision of assigning class labels $c_{i,j}$.

Table I
Classification precision of NB_{init} and NB_{incr}

Video	1	2	3
NB_{init}	0.75	0.77	0.75
NB_{incr}	0.95	0.88	0.83

We test the practicality and performance of the interactive annotation and tracking in Section V-B on all six videos. The complexity is measured by processing time. The proposed algorithm is run using Matlab on an Intel Core i7-2600K CPU, 3.4 GHz, with 16 GB RAM. We construct a GUI for user interaction, as shown in Figure 4. At each iteration given the user correction, computing Equation (9) takes less than 0.1 second. To show the convergence of the EM steps, Figure 5a illustrates that the user query stops at a KF ratio of $30\% \sim 57\%$ (number of KF vs. number of total frames). The KF ratio depends on the difficulty of tracking: more KFs are estimated for more challenging videos. The main workload concentrates on the first 5 iterations.

To show the actual annotation cost (annotation time) and the corresponding improvement, the precision of the output tracks at 0th (before user interaction), 1st, 5th and final iteration is shown in Figure 5b. The precision is defined as (number of correctly labeled frames)/(total number of frames). The precision is as high as $90\% \sim 98\%$ after 5 iterations, and above 98% for all the tested videos at the end of KF query, which can be considered as highly accurate for most applications. Table II shows that the improvement of precision over [21] is up to 24%.

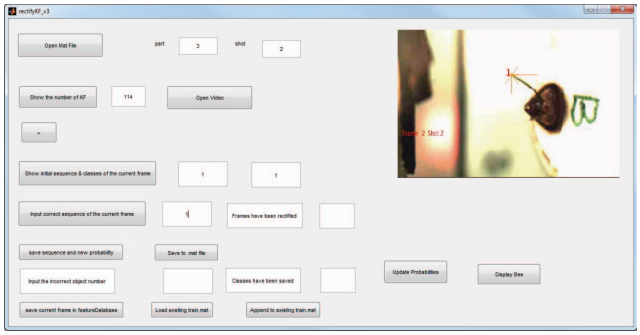

Figure 4. GUI for user interaction.

Table II
Tracking precision on all tested videos

Video	1	2	3	4	5	6
[21]	0.90	0.94	0.93	0.80	0.74	0.83
Proposed framework	0.99	0.99	0.99	0.98	0.98	0.98

Some example KFs are shown in Figure 6, which include missing or merged detection responses, or false positives. These frames are estimated as KFs by the framework for user correction, otherwise they are not likely to be correctly labeled by state-of-the-art automatic tracking algorithms.

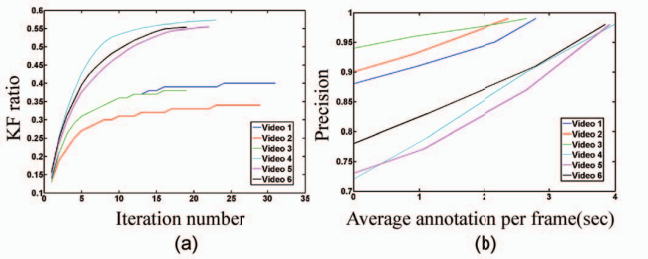

Figure 5. Tracking performance of the proposed framework in terms of (a) KF ratio vs. iteration number and (b) precision vs. averge annotation time per frame (second).

VII. Conclusion

Our motivation is to design an interactive framework for highly accurate object tracking while at the same time minimizing the human effort required. Some human effort will always be required as tracking errors are unavoidable in state-of-the-art tracking algorithms. We propose an interactive

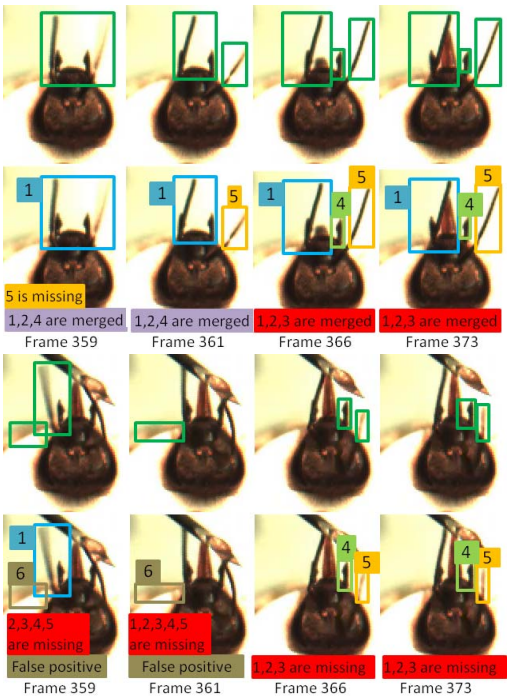

Figure 6. The first and third rows are detection responses, while the second and fourth rows are final ouputs of the proposed framework.

framework, with a new annotation cost function for measuring the difficulty of correct labeling by automatic approaches. This framework enables users to rectify incorrect hypothese based only on the KFs by estimated annotation cost rather than the entire video. It further makes use of user annotation to rectify the other frames and to optimize the final tracks with active learning. We apply this framework exploiting a Naive Bayesian classification and a DAT approach on tracking a bee's antennae and mouthparts from a set of low frame rate videos. The experiments verify the practicality and efficiency of this framework even in challenging cases.

ACKNOWLEDGMENT

The authors would like to thank Manuel Wildner for help with the software evaluation. This work was funded by the INCIDE center and the Zukunftskolleg, University of Konstanz, Germany, with partial support also from the Natural Science Foundation of China under Grants 61302121 and 61363046.

REFERENCES

[1] A. Veeraraghavan, R. Chellappa, and M. Srinivasan, "Shape-and-behavior encoded tracking of bee dances," in *IEEE PAMI*, vol. 3, 2008, pp. 463–476.

[2] D. Reid, "An algorithm for tracking multiple targets," *Automatic Control, IEEE Trans. on*, vol. 24, no. 6, pp. 843–854, 1979.

[3] T. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *Oceanic Engineering, IEEE J. of*, vol. 8, no. 3, pp. 173–184, 1983.

[4] J. Berclaz, F. Fleuret, and P. Fua, "Robust people tracking with global trajectory optimization," in *CVPR*, vol. 1. IEEE, 2006, pp. 744–750.

[5] H. Jiang, S. Fels, and J. J. Little, "A linear programming approach for multiple object tracking," in *CVPR*. IEEE, 2007, pp. 1–8.

[6] B. Leibe, K. Schindler, and L. Van Gool, "Coupled detection and trajectory estimation for multi-object tracking," in *ICCV*. IEEE, 2007, pp. 1–8.

[7] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *CVPR*. IEEE, 2008, pp. 1–8.

[8] A. A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and W. Hu, "Multi-object tracking through simultaneous long occlusions and split-merge conditions," in *CVPR*, vol. 1. IEEE, 2006, pp. 666–673.

[9] C. Huang, B. Wu, and R. Nevatia, "Robust object tracking by hierarchical association of detection responses," in *ECCV*. Springer, 2008, pp. 788–801.

[10] Y. Li, C. Huang, and R. Nevatia, "Learning to associate: Hybridboosted multi-target tracker for crowded scene," in *CVPR*. IEEE, 2009, pp. 2953–2960.

[11] P. Jain and A. Kapoor, "Active learning for large multi-class problems," in *CVPR*. IEEE, 2009, pp. 762–769.

[12] A. Yao, J. Gall, C. Leistner, and L. Van Gool, "Interactive object detection," in *CVPR*. IEEE, 2012, pp. 3242–3249.

[13] A. J. Joshi, F. Porikli, and N. Papanikolopoulos, "Breaking the interactive bottleneck in multi-class classification with active selection and binary feedback," in *CVPR*. IEEE, 2010, pp. 2995–3002.

[14] S. Vijayanarasimhan and K. Grauman, "Large-scale live active learning: Training object detectors with crawled data and crowds," in *CVPR*. IEEE, 2011, pp. 1449–1456.

[15] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *Sys., Man, and Cyber., Part C: App. and Reviews, IEEE Trans. on*, vol. 34, no. 3, pp. 334–352, 2004.

[16] M. S. Lee, J.-K. Rhee, B.-H. Kim, and B.-T. Zhang, "Aesnb: active example selection with naïve bayes classifier for learning from imbalanced biomedical data," in *Bioinformatics and Bio Engineering*. IEEE, 2009, pp. 15–21.

[17] J. Yuen, B. Russell, C. Liu, and A. Torralba, "Labelme video: Building a video database with human annotations," in *ICCV*. IEEE, 2009, pp. 1451–1458.

[18] A. Buchanan and A. Fitzgibbon, "Interactive feature tracking using kd trees and dynamic programming," in *CVPR*, vol. 1. IEEE, 2006, pp. 626–633.

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

[20] C. Vondrick, D. Ramanan, and D. Patterson, "Efficiently scaling up video annotation with crowdsourced marketplaces," in *ECCV*. Springer, 2010, pp. 610–623.

[21] M. Shen, P. Szyszka, C. G. Galizia, and D. Merhof, "Automatic framework for tracking honeybees antennae and mouthparts from low framerate video," in *ICIP*. IEEE, 2013, pp. 4112–4116.

[22] J. Munkres, "Algorithms for assignment and transportation problems," in *J. SIAM*, vol. 5, 1957, pp. 32–38.

[23] H. Jhuang, E. Garrote, X. Yu, V. Khilnani, T. Poggio, A. D. Steele, and T. Serre, "Automated home-cage behavioural phenotyping of mice," *Nature communications*, vol. 1, p. 68, 2010.

[24] Y. Matsumoto, R. Menzel, J. Sandoze, and M. Giurfa, "Revisiting olfactory classical conditioning of the proboscis extension response in honey bees: A step toward standardized procedures," in *J. Neuroscience Methods*, 2012, pp. 159–167.