Visual Tracking via Weakly Supervised Learning from Multiple Imperfect Oracles

Bineng Zhong[1], Hongxun Yao[1], Sheng Chen[1], Rongrong Ji[1], Xiaotong Yuan[2], Shaohui Liu[1], Wen Gao[1, 3]
[1]Harbin Institute of Technology, Heilongjiang Province, 150001, P. R. China
[2]Department of Electrical and Computer Engineering, National University of Singapore, Singapore
[3]Peking University, Beijing, 100871, P. R. China
{bnzhong,yhx,schen,rrji,shaohl}@vilab.hit.edu.cn; eleyuanx@nus.edu.sg; wgao@pku.edu.cn

Abstract

Long-term persistent tracking in ever-changing environments is a challenging task, which often requires addressing difficult object appearance update problems. To solve them, most top-performing methods rely on online learning-based algorithms. Unfortunately, one inherent problem of online learning-based trackers is drift, a gradual adaptation of the tracker to non-targets. To alleviate this problem, we consider visual tracking in a novel weakly supervised learning scenario where (possibly noisy) labels but no ground truth are provided by multiple imperfect oracles (i.e., trackers), some of which may be mediocre. A probabilistic approach is proposed to simultaneously infer the most likely object position and the accuracy of each tracker. Moreover, an online evaluation strategy of trackers and a heuristic training data selection scheme are adopted to make the inference more effective and fast. Consequently, the proposed method can avoid the pitfalls of purely single tracking approaches and get reliable labeled samples to incrementally update each tracker (if it is an appearance-adaptive tracker) to capture the appearance changes. Extensive comparing experiments on challenging video sequences demonstrate the robustness and effectiveness of the proposed method.

1. Introduction

Visual tracking has attracted significant attention due to its wide variety of applications, including intelligence video surveillance, human machine interfaces and robotics. Much progress has been made in the last two decades. However, designing robust visual tracking methods is still an open issue. Challenges in visual tracking problems include non-rigid shape and appearance variations of the object, occlusions, illumination changes, cluttered scenes, etc. To solve them, most top-performing methods rely on online learning-based algorithms [1-4] to adaptively update target appearance. In these methods, visual tracking is formulated as an online binary classification problem and the target appearance is updated adaptively using the images tracked from the previous frames. Compared with the approaches using fixed target models, such as [5], these adaptive approaches are more robust to appearance changes. However, the main drawback of these appearance-adaptive approaches is their sensitivity to drift, i.e., they may gradually adapt to non-targets.

Years of research in tracker drift avoidance have demonstrated that significant improvements on the final tracking results may be achieved by using notably more sophisticated feature selection or target representation procedures, more elaborate synergies between tracking and classification, segmentation or detection, and taking into account prior information on the scenes and tracked objects. One popular technique to avoid tracker drift is to make sure the current tracker doesn't stray too far from the initial appearance model. Matthews et al. [6] are among the first to utilize the technique and provide a partial solution for template trackers. In [7], discriminative attentional regions are chosen on-the-fly as those that best discriminate current object motion from background motion. Tracker drift is unlikely, since no on-line updates of attentional regions, and no new features are chosen after initialization in the first frame. Grabner et al. formulate tracking as an online semi-supervised learning problem [8]. Combining with a prior classifier, this method takes all the coming samples as unlabeled and uses them to update the tracker. Despite their success, these approaches are limited by the fact that they cannot accommodate very large changes in appearance. To balance between semi-supervised and the fully adaptive tracking, Stalder et al. [9] present an approach using object specific and adaptive priors. In [10], Babenko et al. propose to use a multiple instance learning based appearance model for object tracking. Instead of using a single positive image patch to update a traditional discriminative classifier, they use one positive bag consisting of several image patches to update a multiple instance learning classifier. In [11], co-training technique is applied to online multiple trackers learning with different features. The trackers collaboratively classify the new unlabeled samples and use these newly labeled samples with high confidence to update each other. However, independence among different features is required in co-tracking and this condition is too strong to be met in practice. Lu and Hager [12] propose model adaptation driven by feature matching and feature distinctiveness that may be robust to drift. Ren [13] and Yin [14] propose a paradigm of tracking by segmenting to alleviate the drift problem through accurate spatial support obtained in segmentation respectively. However, segmentation based methods only benefit from the situation when the foreground is in high contrast to the background, which is not always the case in natural scenes. Grabner et al. [15] and He et al. [16] use a tracker based on online learning for key-point matching. They perform tracker update only when motion consensus of local descriptors is verified. These two methods can alleviate the drift problem in some extent but they only work well for the tracking of texture-rich objects. A number of attempts [17, 18] have been made to utilize multiple observation models to improve the performance of a tracker. For a full review of tracking literature, please refer to [19].

The inspiration for this work comes from a brand new machine learning area in weak supervision, where the task is to jointly learn from multiple labeling sources [20-25]. This general task underlies several subfields receiving increasing interest from the machine learning community, such as data fusion, active learning, transfer learning, multitask learning, multiview learning, and learning under covariate shift. The problem of learning from multiple labeling sources is different from the unsupervised, supervised, semi-supervised or transductive learning problems, in which each training instance is given a set of candidate class labels provided by different labelers with varying accuracy and the ground truth label of each instance is unknown. In practice, a variety of real-world problems can be formalized as such a ‘multi-labelers’ problem. For example, there have been an increasing number of experiments using Amazon's Mechanical Turk [26] for annotation. In situations like these, the performance of different annotators can vary widely. Without the ground truth, how to learn classifiers, evaluate the annotators, infer the ground truth label of each data point, and estimate the difficulty of each data point are the main issues addressed by the task of learning from multiple labeling sources. Other examples of a ‘multi-labelers’ scenario involve reCAPTCHA [20], computer-aided diagnosis [23] and Search-engine optimizers [24].

To the best of authors’ knowledge, none of the earlier studies have viewed visual tracking as the problem of learning from multiple labeling sources. In the tracking literatures, for a given tracking scenario, we actually can get a lot of output via a number of tracking algorithms using different object representations and learning strategies. Since each kind of tracking algorithm has its strength and weakness and is particularly applicable for handling a certain type of variation, many methods often use sequential cascaded or parallel majority voting frameworks to fuse the output of a number of tracking algorithms. One of the main challenges dominating these two kinds of fusing schemes is that how would one measure the performance of a tracker when there is no ground truth available? The tracking approach, proposed in this paper, is conceptually different and explores a new strategy; in fact, instead of using sequential cascaded or parallel majority voting schemes, we consider visual tracking in a novel weakly supervised learning scenario where (possibly noisy) labels but no ground truth are provided by multiple imperfect oracles (i.e., trackers), some of which may be mediocre. A probabilistic approach is proposed to explore the possible alternative of fusing multiple imperfect oracles for visual tracking, and simultaneously infer the most likely object position and the accuracy of each imperfect oracle. Our method has the following advantages.

(1) We propose a natural way of fusing multiple imperfect oracles to get a final reliable and accurate tracking result. The imperfect oracles can be arbitrary tracking algorithms in the literature. This avoids the pitfalls of purely single tracking approaches.

(2) The proposed algorithm gives an estimate of the ground truth labeling of training data during tracking in a robust probabilistic inference manner and thus can alleviate the tracker drift problem.

(3) We can online evaluate tracking algorithms in the absence of ground truth, which is an important and challenging problem in visual tracking systems.

(4) The proposed approach can also handle missing labels (i.e., each tracker is not required to label all the image patches).

Therefore, we can greatly alleviate the tracker drift problem to robustly achieve long-term persistent tracking in ever-changing environments.

The rest of the paper is organized as follows. Section 2 introduces our weakly supervised learning formulation for visual tracking, and presents the probabilistic approach that jointly learns the most likely object position and each tracker’ accuracy. The detailed tracking algorithm is then described in Section 3. Experimental results are given in Section 4. Finally, we conclude this work in Section 5.

2. An weakly supervised learning view of visual tracking

2.1. A Chicken-and-egg problem

For online learning based visual tracking, a tracker observes samples (typically image patches) in each frame and predicts their labels to be either foreground or background. At the end of each frame, the adaptive tracker uses the newly obtained sample-label pairs to presumably improve its prediction rule for the frames to come. However, due to the challenges in natural scenes and accumulated tracking error, the tracker may gradually adapt to non-targets. The main reason behind tracker drift is that the tracker is updated using a self-learning policy in the absence of ground truth. Many demonstrations have shown that aggregating the judgments of a number of individuals (e.g., detectors, trackers and recognizers) enhances the tracking performance to some degree, a phenomenon that has come to be known as the “wisdom of crowds”. Thus, the performance of one tracker may be assessed by using majority voting scheme from other trackers, which is often utilized in tracking applications.

But how would one measure the performance of the other trackers when there is no ground truth available? This is an apparent chicken-and-egg problem. So the question is, despite the absence of ground truth, how to effectively address the apparent chicken-and-egg problem in order to learn the appearance changes of the target while alleviating the drift problem.

2.2. Optimal Integration of Labels from Labelers of Unknown Expertise

Recently, weakly supervised learning from multiple labeling sources has aroused the interests of many researchers. Before introducing our work, we briefly review the work of [25] as it forms the basis of our approach.

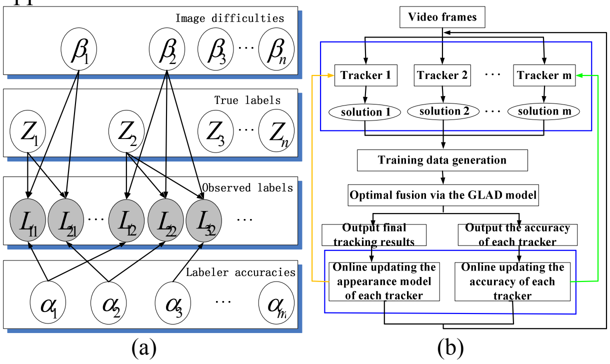


Fig. 1. (a) Graphical model [25] of image difficulties, true image labels, observed labels, and labeler accuracies. Only the shaded variables are observed. (b) Overview of the proposed tracking algorithm.

Consider a set of n images, each of which belongs to one of two possible categories of interest. We want to determine the class label $Z_j \in \{0,1\}$ of each image j by querying from m labelers. The observed labels depend on several causal factors: (1) the difficulty of the image; (2) the true label; and (3) the expertise of the labeler. Denote $1/\beta_j \in [0,+\infty)$ as the parameter of the difficulty of the image j, Z_j as the true label of the image j and $\alpha_i \in (-\infty,+\infty)$ as the parameter of the expertise of the labeler i. The labels given by labeler i to image j are denoted as L_{ij} and, under the model, are generated as follows:

$$p\left(L_{ij} = Z_j \middle| \alpha_i, \beta_j\right) = \frac{1}{1+e^{-\alpha_i \beta_j}} \qquad (1)$$

Thus, under the model, the log odds for the obtained labels being correct are a bilinear function of the difficulty of the label and the expertise of the labeler, i.e.,

$$\log \frac{p(L_{ij}=Z_j)}{1-p(L_{ij}=Z_j)} = \alpha_i \beta_j \qquad (2)$$

Fig.1 (a) shows the causal structure of the model. True image labels Z_j, labeler accuracy values α_i, and image difficulty values β_j are sampled from a known prior distribution. These determine the observed labels according to Equation 1. Given a set of observed labels $L = \{l_{ij}\}$, the task is to infer simultaneously the most likely values of $\mathbf{Z} = \{Z_j\}$ (the true image labels) as well as the labeler accuracies $\boldsymbol{\alpha} = \{\alpha_i\}$ and the image difficulty parameters $\boldsymbol{\beta} = \{\beta_j\}$. An Expectation-Maximization approach (EM) is derived for obtaining maximum likelihood estimates of the parameters of interest:

E step: Let the set of all given labels for an image j be denoted as $\mathbf{l}_j = \{l_{ij'} | j' = j\}$. Note that not every labeler must label every single image. In this case, the index variable i in $l_{ij'}$ refers only to those labelers who labeled image j. We need to compute the posterior probabilities of all $z_j \in \{0,1\}$ given the $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ values from the last M step and the observed labels:

$$\begin{aligned} p\left(z_j \middle| \mathbf{l}, \boldsymbol{\alpha}, \boldsymbol{\beta}\right) &= p\left(z_j \middle| \mathbf{l}_j, \boldsymbol{\alpha}, \beta_j\right) \\ &\propto p\left(z_j \middle| \boldsymbol{\alpha}, \beta_j\right) p\left(\mathbf{l}_j \middle| z_j, \boldsymbol{\alpha}, \beta_j\right) \\ &\propto P(z_j) \textstyle\prod_i p(l_{ij} | z_j, \alpha_i, \beta_j) \qquad (3) \end{aligned}$$

where we noted that $p\left(z_j \middle| \boldsymbol{\alpha}, \beta_j\right) = p(z_j)$ using the conditional independence assumptions from the graphical model.

M step: We maximize the standard auxiliary function Q, which is defined as the expectation of the joint log-likelihood of the observed and hidden variables $(\mathbf{l}, \mathbf{Z})$ given the parameters $(\boldsymbol{\alpha}, \boldsymbol{\beta})$, w.r.t. the posterior probabilities of the $\mathbf{Z}$ values computed during the last E step:

$$\begin{aligned} Q(\boldsymbol{\alpha}, \boldsymbol{\beta}) &= E[\ln p(\mathbf{l}, \mathbf{z} | \boldsymbol{\alpha}, \boldsymbol{\beta})] \\ &= E\left[\ln \textstyle\prod_j \left(p(z_j) \prod_i p\left(l_{ij} | z_j, \alpha_i, \beta_j\right)\right)\right] \\ &\quad \text{(since } l_{ij} \text{ are cond. indep. given } \mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\beta}\text{)} \\ &= \textstyle\sum_j E[\ln p(z_j)] + \sum_{ij} E[\ln p(l_{ij} | z_j, \alpha_i, \beta_j)] \qquad (4) \end{aligned}$$

where the expectation is taken over $\mathbf{z}$ given the old parameter values $\boldsymbol{\alpha}^{\text{old}}, \boldsymbol{\beta}^{\text{old}}$ as estimated during the last E-step. Using gradient ascent, we find values of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ that locally maximize Q.

To facilitate the description later on, we denote the above inference method as the GLAD (Generative model of Labels, Abilities, and Difficulties), which is also used in [25].

2.3. Visual tracking via learning from oracle set

A unified weakly supervised learning framework for visual tracking, in which multiple imperfect oracles cooperate on tracking objects, can be formulized as the following problem:

Consider that m imperfect oracles observe an input sequence $s_1, s_2, \ldots, s_T$, where $s_t = \{x_t^1, x_t^2, \ldots, x_t^{N_t}\}$ is a sample set obtained in frame t and x_t^j is the jth sample in s_t. Denote $L_t^j = \{l_t^{ij'} | j' = j\}$ as a set of candidate class labels for the sample x_t^j, where $l_t^{ij'} \in \{0,1\}$ is the label provided by ith oracle. Our weakly supervised learning-based tracker setting is depicted in Fig.2.

```
For $t = 1, 2, \ldots$
    1. Receive sample set $s_t = \{x_t^1, x_t^2, \ldots, x_t^{N_t}\}$.
    2. Predict $L_t^j = \{l_t^{ij'} | j' = j\}$ for each sample $x_t^j$ using multiple imperfect oracles.
    3. Jointly learn the ground truth label of each sample and the accuracy of each imperfect oracle from current observed sample set $s_t$.
    4. Update multiple imperfect oracles based on the robust tracking results.
```

Fig. 2. Weakly supervised learning setting for visual tracking.

3. Robust visual tracking

In this section, we develop our tracking algorithm inside the weakly supervised learning setting described above. The basic idea is to embed a heterogeneous set of trackers into our weakly supervised learning algorithm to form a robust tracking algorithm. The proposed tracking algorithm is schematically shown in Fig.1 (b).

Specifically, our tracking algorithm works as follows: For one incoming video frame, we first obtain a set of candidate solutions that are produced by a heterogeneous set of tracking algorithms. Then, training data, which is used for GLAD, is carefully selected according to a heuristic strategy. After the training data generation process, the GLAD model is utilized to infer simultaneously the most likely object position and the accuracy of each tracker. A testing sample with maximum probability belonging to positive sample is chosen to be the new object position, and also is retained as a positive training sample for further tracker update. An online evaluation strategy is developed to incrementally update the accuracy of each tracker. Meanwhile, target appearance model of each candidate tracker is updated if it is an appearance-adaptive tracker. The tracking procedure continues in this iterative fashion until the end of video. Below we give a detailed description about each component in this framework, and the algorithm is summarized finally.

3.1. A heterogeneous set of oracles

The key part of our algorithm proceeds by first computing many different tracking results that serve as proposal solutions, and then optimally fusing them using the GLAD model. The success of the method thus depends on the availability of good proposal solutions.

It is important to note that the proposal solutions need not to be good in the whole tracking process in order to be "useful". Instead, each solution may contribute to a particular time in the whole tracking process, if it contains reasonable tracking results for that time, no matter how poor it is in other times. This suggests the use of different tracking methods with different strengths and weaknesses for computing the proposals. In our experiments, we used six kinds of the proposal solutions (please see Table 1).

(1) Fragments-based Tracker [5]. The tracker uses static appearance models to obtain solutions. Due to using integral gray histograms and part based appearance model, such solutions are very efficient and robust to occlusions. However, the method tends to have difficulties tracking objects that exhibit significant appearance changes.

(2) Online Boosting Tracker [3]. The tracker uses online boosting method to obtain solutions. Due to the properties of the method, such solutions are able to adapt to appearance changes of the object, but unfortunately suffer from the drifting problem.

(3) Semi-Supervised Online Boosting Tracker [8]. The tracker uses semi-supervised online boosting method to obtain solutions. Such solutions can alleviate the drifting problem since the tracker cannot get too far away from the prior. But the prior might be too strong (i.e., limited appearance changes and partial occlusions) and generic (i.e., no discrimination between different objects from one class).

(4) Beyond Semi-Supervised Tracker [9]. The tracker uses beyond semi-supervised tracking method, which is balancing between semi-supervised and the fully adaptive tracking, to obtain solutions.

(5) Online Multiple Instance Learning-based Tracker [10]. The tracker uses online multiple instance learning (MIL) method, in which one positive bag consisting of several image patches is used to update a MIL classifier, to obtain solutions.

(6) SURF Tracker [16]. The tracker uses SURF descriptors to obtain solutions. Such solutions often contain good results for textured objects but are virtually useless for textureless objects.

Table 1. A rich set of complementary tracking approaches are used in our method (please refer to text for further explanation).

The reasons we choose such a rich set of proposal solutions are:

(1) The set consists of a variety of complementary tracking approaches, such as gray histogram-based patch matching [5], motion consensus of local descriptors [16] and online classification using haar feature [3, 8, 9, 10].

(2) The features used in all the six tracking methods can be extracted in a very efficient manner due to integral image data structure.

(3) Their source codes are publicly available. This makes the parameters tuning to achieve the results reported in original literature and the comparison of these approaches convenient and fair.

(4) Candidate solutions of each tracking method can be obtained independently, which allows for a high-speed parallel implementation if needed.

It is important to note that other (potentially more efficient and robust) approaches for obtaining proposal solutions may also be considered. The proposed approach provides a principled way of fusing proposal solutions from any tracking algorithms in the literature.

3.2. Heuristic selection of training data for optimal fusion

It can be seen from section 2.2 that the computational complexity of the E-Step is linear in the number of patches and the total number of labels. For the M-Step, the values of Q and ∇Q must be computed repeatedly until convergence. Computing each function is linear in the number of patches, number of labelers, and total number of image labels. To make our tracking algorithm computationally feasible in practice, we develop in this subsection a heuristic way to select training data for the GLAD model.

Consider that m imperfect oracles receive an image patch set $s_t = \{x_t^1, x_t^2, \ldots, x_t^{N_t}\}$ within current search window in frame t, where x_t^j is the jth image patch of interested. Denote $P_t = \{p_t^1, p_t^2, \dots, p_t^m\}$ as a set of proposal solution bounding boxes obtained via m imperfect trackers $T = \{T_1, T_2, \dots, T_m\}$ in frame t, $D(\cdot,\cdot)$ as the center location distance (pixels) between two image patch, and $l_t^{ij} \in \{0,1\}$ as the label of the image patch x_t^j provided by the tracker T_i. For each tracker T_i, denote Top_t^i as a set of image patches having the highest 10 confidences (estimated by the tracker T_i) belonging to the positive samples in frame t. The training data is heuristically selected as follow:

```
● For $j = 1, \dots, N_t$
    ■ If $x_t^j \in P_t = \{p_t^1, p_t^2, \dots, p_t^m\}$
        ◆ For $i = 1, \dots, m$
            ● If$(D(x_t^j, p_t^i) \le 5 \parallel x_t^j \in Top_t^i)$ $l_t^{ij} = 1$.
            ● Else $l_t^{ij} = 0$.
        ◆ End for
    ■ Else neglect the image patch $x_t^j$.
● End for
```

In addition, we initialize the parameter β_t^j of the difficulty of the image patch x_t^j by majority voting:

$$\beta_t^j = \exp\left(k_1 \times \left|\left(\frac{2}{m}\sum_{i=1}^m l_t^{ij}\right) - 1\right|\right) \tag{5}$$

where k_1 is a normalized factor and typically set as 1.6.

3.3. Online evaluation of trackers

Automatic evaluation of visual tracking algorithms in the absence of ground truth is a very challenging and important problem. As the accuracy α_i of each tracker can be inferred in each frame and the Q function in Equation (4) can be modified straightforwardly to handle a prior over each α_i by adding a log-prior term for each of these variables, we would like to online determine which of the trackers are most accurate. Heuristically, we are interested in the trackers which have the most supporting evidence over time. After the tracking of each frame, we first check the m proposal solutions of the imperfect trackers against the fusing results. A tracker is deemed to fail if the center location error (pixels) between its proposal bounding box and the final bounding box obtained by fusion is great than 5. This threshold can be perturbed with little effect on performance. Then, the prior accuracies of the m trackers at time t are adjusted as follows:

$$\alpha_i^t = (1 - w_i^t)\alpha_i^{t-1} + w_i^t M_i^t \tag{6}$$

where w_i^t is the learning rate and M_i^t is 1 for the tracker which succeeded and 0 for the remaining models.

In addition, in order to avoid false updating and maximize robustness, we adaptively adjust the learning rate w_i^t of the ith tracker as follows:

$$w_i^t = \begin{cases} \frac{k_2}{1+e^{-\alpha_i}}, & \text{if } M_i^t = 1 \\ \frac{k_2}{1+e^{\alpha_i}}, & \text{if } M_i^t = 0 \end{cases} \tag{7}$$

where k_2 (typically set as 0.1) is a normalized factor and α_i is the accuracy of the ith tracker, which is one of the output of the GLAD model. In the case that a tracker succeeded (i.e., $M_i^t = 1$), if the accuracy of the ith tracker α_i is large, the learning rate w_i^t is set to be large to adapt quickly. Otherwise, w_i^t is small.

3.4. Summary of the algorithm

A summary of our weakly supervised learning based tracking algorithm is described as follows.

```
Algorithm 1 Weakly Supervised Learning from Multiple Imperfect Oracles for Visual Tracking
Initialization:
    1. Acquire one manually labeled frame.
    2. Construct target appearance model for each candidate tracker.
    3. Initialize the accuracy of each candidate tracker .
for t = 2 to the end of the video
    1. Generate a set of proposal solutions via a heterogeneous set of trackers.
    2. Select training data for optimal fusion according to a heuristic strategy.
    3. Run the GLAD model.
    4. Locate the new target position as the sample with maximum probability belonging to positive sample.
    5. Update the accuracy of each candidate tracker.
    6. Update target appearance model of each candidate tracker if it is an appearance-adaptive tracker.
end for
```

4. Experiments

To evaluate the performance of our proposed tracking method, we apply it to several challenging video sequences and systematically compare our tracker with several representatives of state-of-the-art trackers.

4.1. Experiment setting

In our experiments, we chose to track only the location for simplicity and computational efficiency reasons. Thus, no scale and rotation adaption are implemented, which both however can be incorporated with slight modification of algorithm. We don't use any motion model to predict the new position. The centroid of the search window is the same as that of the previous target bounding box. The processing speed depends on the size of the search window which we have defined by enlarging the target region by half of its size in each direction. In addition, multiple imperfect oracles are parallel implemented. We have achieved the processing speed of 10 fps at the resolution of 320×240 pixels (the running time could be reduced substantially using multiple cores). The initial accuracy α_i^0 of each oracle is 1. Our algorithm is implemented using C++, on a machine with Intel Pentium Dual 2.0 GHz processor.

For performance evaluation, we compare our approach against several representatives of the current state-of-the-art in visual tracking – the Fragments-based Tracker [5], the Online Boosting Tracker [3], the

Semi-Supervised Online Boosting Tracker [8], the Beyond Semi-Supervised Tracker [9], the Online Multiple Instance Learning-based Tracker [10], and the SURF Tracker [16]. In the rest of our experiments, we refer to these six compared algorithms as **FT**, **OBT**, **SSOBT**, **BSST**, **OMILT** and **ST** respectively. For the first five methods, we use the same parameters as the authors have given on their websites [27, 28, 29] for all of our experiments. The **ST** is implemented by ourselves according to [16]. In addition, to more clearly illustrate that we can give proper labels to new samples for further tracker training, we also implement the variations of the six algorithms separately. More specifically, the variation of each algorithm updates itself with reliable labeled samples obtained by the finally fusing results. We refer to the variations of the six trackers as **FT_V**, **OBT_V**, **SSOBT_V**, **BSST_V**, **OMILT_V** and **ST_V** respectively. We have tested the tracking algorithms using dozens of challenging video sequences from the existing literatures as well as our own collections. Both qualitative and quantitative comparisons are done to evaluate the involved tracking algorithms. The quantitative performance is measured by the center location errors (pixels) in each frame and average center location errors in the whole sequences. Due to space limitation, we only show six challenging video sequences (please see Fig.3) in this paper. The *Bear* sequence on the row 1 is from the internet. It contains a running bear captured by a moving camera. This is a challenging sequence since it suffers from light & pose changes and dramatic figure/ground appearance pattern changes. The *Background_Clutter* sequence on the row 2 is taken from [28]. This sequence contains a 'Waldo' doll moving in front of very similar background. This is a very difficult visual tracking task. A tracking algorithm should neither track the needle to which the target object is attached nor track an object in the background. The *Airplane* sequence on the third row is a low quality surveillance video captured by a PTZ camera watching a runway. The *David_ Indoor* sequence on the fourth row are from an indoor sequence which contains a person moving from dark toward bright area with large lighting and pose changes. The *Person_Surf* sequence on the fifth row contains a surfer riding a wave. The turbulent wakes created by sweeping wave and the surfer create significant challenges for tracking algorithms. The *Motor* sequence on the last row contains a motor moving with large pose, lighting variation in a cluttered background.

4.2. Comparison with other trackers

To show the advantage of the proposed approach over the other trackers, we perform a number of experiments using a rich set of imperfect oracles consisting of **OBT_V**, **SSOBT_V**, **BSST_V**, and **OMILT_V** trackers and report the results in this subsection.

Fig.3 shows qualitative comparison results of the trackers, i.e., our method (in red), the **OBT** (in magenta), **SSOBT** (in green), **BSST** (in blue) and **OMILT** (in yellow), on the six challenging video sequences mentioned in subsection 4.1. Our method gives good results because it considers visual tracking in a weakly supervised learning scenario where (possibly noisy) labels are provided by multiple imperfect oracles. By simultaneously inferring the most likely object position and the accuracy of each imperfect oracle via the proposed probabilistic approach, we could obtain training data during tracking process in a robust manner for further tracker update. Therefore, the tracker drift problem is alleviated in the proposed method. This is further verified in the experiments on the four variations (i.e., **OBT_V**, **SSOBT_V**, **BSST_V** and **OMILT_V**) of the four trackers (i.e., **OBT**, **SSOBT**, **BSST** and **OMILT**). As shown in Fig.4, it is obviously to see that the four variations outperform their corresponding trackers.

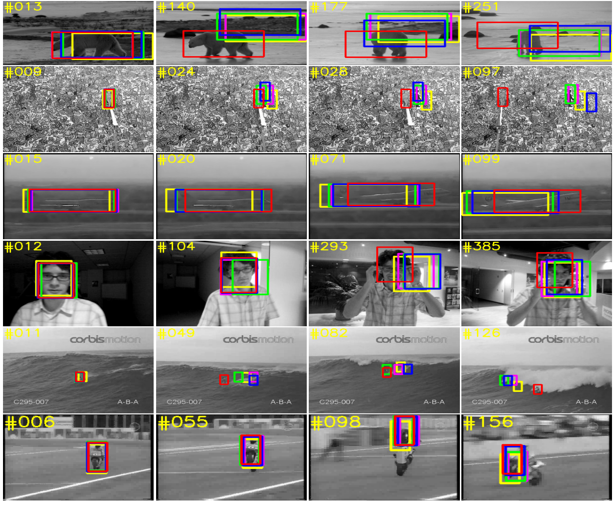


Fig. 3. Qualitative comparison results of our method (in red), the **OBT** (in magenta), **SSOBT** (in green), **BSST** (in blue) and **OMILT** (in yellow) on six challenging video sequences. Please see text for detailed description. This figure is best viewed in color.

The quantitative comparison results of the trackers are listed in Table 2 and Fig.5 respectively. Due to space limitation, we only plot the position error curves for the *Bear, Background_Clutter*, *Airplane* and *David_Indoor* sequence in Fig.5. As we can see, the continuously changing background and quick variation in foreground result in the failure of the single imperfect oracle, e.g., the **OBT**, **SSOBT**, **BSST** and **OMILT**; while our method can track the targets for almost the full length of all these sequences.

Image sequence	OBT	SSOBT	BSST	OMILT	Ours
Bear	76	75	63	85	18
Background_Clutter	89	94	111	101	3
Airplane	28	27	30	45	7

David_Indoor	16	24	15	18	11
Person_Surf	35	37	42	34	5
Motor	22	21	22	21	23

Table 2. Average center location errors (pixels). Quantitative comparison results on six challenging sequences by our method, the **OBT**, **SSOBT**, **BSST** and **OMILT** separately.

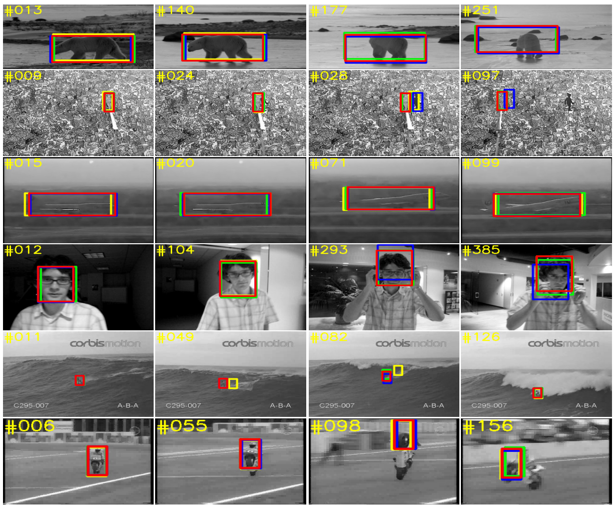


Fig. 4. Qualitative comparison results of our method (in red), the **OBT_V** (in magenta), **SSOBT_V** (in green), **BSST_V** (in blue) and **OMILT_V** (in yellow) on six challenging video sequences. Please see text for detailed description.

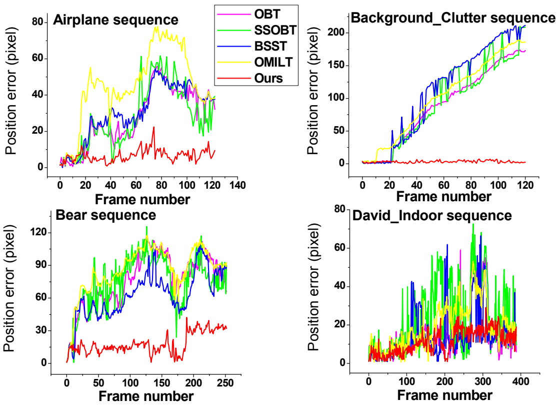
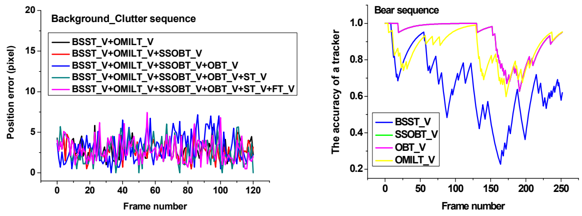


Fig. 5. Position error curves for four sequences we tested on.

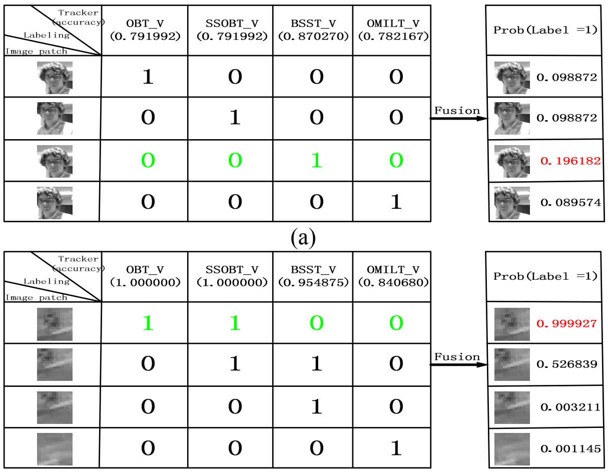
4.3. How many oracles?

Since the success of our method depends on the availability of good proposal solutions, there naturally arise the following questions: 1) How sensitive our method is to the number of oracles and the robustness of a single imperfect oracle? 2) How easy or difficult is it to obtain a good set of imperfect oracles? To answer these two questions, we calculate the tracker accuracies in different configuration of an oracle set. Instead of travel through all the possible combinations of trackers, we use a simple sequential heuristic, i.e., increase the oracles set by adding one oracle at a time. The measurements are made for several image sequences. The results for the *Background_Clutter* sequence are plotted in Fig.6 (a). Obviously, for different combinations of multiple imperfect oracles, a good combination can be chosen across a wide range of oracle sets. The same observation is identical for all the other test sequences. This property significantly eases the selection of a rich set of imperfect oracles. Furthermore, the experiments have shown that a good set of multiple oracles for a sequence usually performs well also for other sequences (please see Fig.3).



(a) (b)

Fig. 6. (a) Position error curves for different combinations of multiple imperfect oracles for the *Background_Clutter* sequence. The oracle set is increased in a sequential way. (b) Evolving curves of the accuracies for the **OBT_V**, **SSOBT_V**, **BSST_V** and **OMILT_V** in *Bear* sequence separately. Please note that the curve of **OBT_V** is the same as that of **SSOBT_V** in this example.



Tracker (accuracy) / Labeling / Image patch	OBT_V (0. 791992)	SSOBT_V (0. 791992)	BSST_V (0. 870270)	OMILT_V (0. 782167)	Prob(Label =1)
	1	0	0	0	0. 098872
	0	1	0	0	0. 098872
	0	0	1	0	0. 196182
	0	0	0	1	0. 089574

(a)

Tracker (accuracy) / Labeling / Image patch	OBT_V (1. 000000)	SSOBT_V (1. 000000)	BSST_V (0. 954875)	OMILT_V (0. 840680)	Prob(Label =1)
	1	1	0	0	0. 999927
	0	1	1	0	0. 526839
	0	0	1	0	0. 003211
	0	0	0	1	0. 001145

(b)

Fig.7. Illustration of the fusion process. Red indicates the highest probability belonging to positive sample. Please see text for details. (a) Frame#189 from *David_Indoor* sequence. (b) Frame #31 from *Person_Surf* sequence.

4.4. Multiple imperfect oracles: whom to trust?

One big advantage of multiple oracle based tracking lies in that the proposal solutions are not necessarily to be good in the whole tracking process in order to be "useful". To verify this advantage, we check the accuracies of the oracles over time in all the test sequences and give a typical example in Fig.6 (b). In our method, if one oracle gives correct tracking results in current frame, then its accuracy is incrementally updated to increase. Otherwise, its accuracy is incrementally updated to decrease. As shown in Fig.6 (b), it is obviously to see that each oracle may contribute to a particular time in the whole tracking process, if it contains reasonable tracking results for that time, no matter how poor it is in other times.

4.5. Illustration of the fusion process

In this subsection, we show through the two tracking examples the fusion process and how the training samples are reliably labeled to alleviate the tracker drift problem. In Fig.7, we visualize some representative image patches and their (possibly noisy) labels (i.e., 0 or 1) that are provided by multiple imperfect oracles from the frame#189 in *David_Indoor* sequence and the frame#31 in *Person_Surf* sequence respectively. As expected, though the positive/negative ratio of an image patch is 1:3 (the fourth row in Fig.7 (a)) or 2:2 (the second row in Fig.7 (b)), we can get the most likely target samples for further training. Meanwhile, these two case studies also show the proposed method is robust to noisy (or adversarial) labeling and the advantage of weak supervised learning scheme over majority voting scheme for tracking results fusion. For example, the image patch on the fourth row in Fig.7 (a) is labeled as 0 by **OBT_V, SSOBT_V** and **OMILT_V**. Only **BSST_V** labels it as 1. After fusion, we can still give the correct labeling (i.e., 1) to the image patch.

5. Conclusion and future work

We address in this paper the tracker drift problem and explore a novel visual tracking framework in the setting of weakly supervised learning where (possibly noisy) labels provided by multiple imperfect oracles can be efficiently used for inference. To instantiate the proposed weakly supervised tracking framework, we extend the GLAD model [25] to video processing mode and take advantage of sequential data for making real time and accurate inference. An online evaluation strategy is developed to incrementally update the accuracy of each tracker. Meanwhile, target appearance model of each candidate tracker is updated if it is an appearance-adaptive tracker. Extensive experimental results show that the proposed method can obtain more accurate data labels than single oracle and the majority vote heuristic, and it is robust to noisy labeling. In summary, we conclude with observations that advantages of multiple complementary oracles can be seamlessly combined to achieve robust tracking by simultaneously inferring the most likely object position and the accuracy of each oracle in the absence of ground truth, which is always the case in real-world tracking applications.

Acknowledgments

This work is supported by Natural Science Foundation of China (No. 60775024 and No. 60803147), National Basic Research Program of China (No.2009CB320906).

References

[1] J. Lim, D. Ross, R. Lin, and M. Yang. Incremental Learning for Visual Tracking. NIPS 2004.
[2] R. Collins, Y. Liu, and M. Leordeanu. Online Selection of Discriminative Tracking Features. TPAMI 2005.
[3] H. Grabner and H. Bischof. On-line Boosting and Vision. CVPR 2006.
[4] S. Avidan. Ensemble Tracking. TPAMI 2007.
[5] A. Adam, E. Rivlin, and I. Shimshoni. Robust Fragments-based Tracking using the Integral Histogram. CVPR 2006.
[6] I. Matthews, T. Ishikawa, and S. Baker. The Template Update Problem. TPAMI 2004.
[7] M. Yang, J. Yuan, and Y. Wu. Spatial Selection for Attentional Visual Tracking. CVPR 2007.
[8] H. Grabner, C. Leistner, and H. Bischof. Semi-Supervised On-line Boosting for Robust Tracking. ECCV 2008.
[9] S. Stalder, H. Grabner, and L. Van Gool. Beyond Semi-Supervised Tracking: Tracking Should Be as Simple as Detection, but not Simpler than Recognition. ICCV 2009 Workshop on On-line Learning for Computer Vision.
[10] B. Babenko, M. Yang, and S. Belongie. Visual Tracking with Online Multiple Instance Learning. CVPR 2009.
[11] F. Tang, S. Brennan, Q. Zhao, and H. Tao. Co-Tracking Using Semi-Supervised Support Vector Machines. ICCV 2007.
[12] L. Lu and G. Hager. A Nonparametric Treatment for Location/Segmentation Based Visual Tracking. CVPR 2007.
[13] X. Ren and J. Malik. Tracking as Repeated Figure/Ground Segmentation. CVPR 2007.
[14] Z. Yin and R. Collins. Shape Constrained Figure-Ground Segmentation and Tracking. CVPR 2009.
[15] M. Grabner, H. Grabner, and H. Bischof. Learning Features for Tracking. CVPR 2007.
[16] W. He, T. Yamashita, H. Lu, and S. Lao. SURF Tracking. ICCV 2009.
[17] B. Stenger, T. Woodley, and R. Cipolla. Learning to Track with Multiple Observers. CVPR 2009.
[18] M. Yang, F. Lv, W. Xu, and Y. Gong. Detection Driven Adaptive Multi-cue Integration for Multiple Human Tracking. ICCV 2009.
[19] A. Yilmaz, O. Javed, and M. Shah. Object Tracking: A Survey. ACM Computing Surveys 2006.
[20] L. Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum. reCAPTCHA: Human-Based Character Recognition via Web Security Measures. Science 2008.
[21] R. Snow, B. Connor, D. Jurafsky, and A. Ng. Cheap and Fast-But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks. EMNLP 2008.
[22] P. Donmez and J. Carbonell. Proactive Learning: Cost-Sensitive Active Learning with Multiple Imperfect Oracles. CIKM 2008.
[23] V. Raykar, S. Yu, L. Zhao, A. Jerebko, C. Florin, G. Valadez, L. Bogoni, and L. Moy. Supervised Learning from Multiple Experts: Whom to trust when everyone lies a bit. ICML 2009.
[24] O. Dekel and O. Shamir. Good Learners for Evil Teachers. ICML 2009.
[25] J. Whitehill, P. Ruvolo, J. Bergsma, T. Wu, and J. Movellan. Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise. NIPS 2009.
[26] Amazon. Mechanical turk. http://www.mturk.com.
[27] http://www.cs.technion.ac.il/~amita/fragtrack/fragtrack.htm
[28] http://www.vision.ee.ethz.ch/boostingTrackers/index.htm
[29] http://vision.ucsd.edu/~bbabenko/project_miltrack.shtml