Disagreement-Based Multi-System Tracking

Quannan Li[1], Xinggang Wang[2], Wei Wang[3], Yuan Jiang[3], Zhi-Hua Zhou[3], Zhuowen Tu[1]

[1]Lab of Neuro Imaging, University of California, Los Angeles
[2]Huazhong University of Science and Technology
[3]National Key Laboratory for Novel Software Technology, Nanjing University

Abstract. In this paper, we tackle the tracking problem from a fusion angle and propose a disagreement-based approach. While most existing fusion-based tracking algorithms work on different features or parts, our approach can be built on top of nearly any existing tracking systems by exploiting their disagreements. In contrast to assuming multi-view features or different training samples, we utilize existing well-developed tracking algorithms, which themselves demonstrate intrinsic variations due to their design differences. We present encouraging experimental results as well as theoretical justification of our approach. On a set of benchmark videos, large improvements (20% $\sim$ 40%) over the state-of-the-art techniques have been observed.

1 Introduction

Object tracking has been a long standing problem in vision. Once a tracker gets initialized, it starts to track the target in a video by performing two steps: (1) making a prediction about the location of the target, and (2) updating its object model (location, appearance, and shape) based on the prediction. This is in spirit very similar to the bootstrapping and learning procedure in a learning algorithm. With the recent success in detection-based tracking approaches, an increasing amount of work has treated the tracking problem as a semi-supervised learning problem [1–5]. Picking a target to track at the beginning provides supervised data; the remaining of the frames for the tracker to explore do not contain label information and thus is unsupervised. Due to the errors introduced in both the prediction and model updating stage, nearly any tracker will eventually fail with the errors being accumulated over the time.

Disagreement-based semi-supervised learning approaches [6], such as co-training or tri-training [7, 8], provide a mechanism to allow classifiers trained on different views or data samples to exploit unlabeled data. The learning process is a type of ensemble learning [9–11]. It involves multiple classifiers which label the unlabeled data to update and improve each other [12]. From a different angle, the use of multiple classifiers can be viewed as a fusion problem and it has been shown that fusing complementary features in a tracking system often leads to enhanced performances [13–15]. However, less efforts have been made in learning to fuse well-developed existing algorithms through semi-supervised learning; we will see

later (in both theory and experiments) that a disagreement-based fusion significantly improves the performance over direct combination of features/systems [14, 16, 17].

In this disagreement-based multi-system tracking approach, we seek a balance between the current tracker and the level of agreements among other trackers. Our intuition is to find the location where the current tracker is confident but disagrees with other trackers, while other trackers reach a high degree of agreement. We provide both theoretical and experimental evidence to our approach and show much improved results over the state-of-the-art techniques on benchmark videos.

2 Related Work

A number of tracking methods have been proposed to perform fusion[13, 14, 18, 19, 16, 15, 17]. Different from [13, 17] where multiple parts were tracked and correlated, we deal with a single target. In [14, 16] multiple trackers were fused but these trackers represent different features and they were directly combined. In [18] the tracking approach was combined via the weighted combination of the PDFs. Different from [18], our method does not perform direct multiplication but seeks a balance between the PDF of one tracker and the degree of agreement by the other trackers; also, in our method, each tracker performs prediction separately maintaining certain independence and patches at the agreed positions can be recommended to update the other trackers. In [20], the tracking combination method is trained for specific scenarios. Different from [20], our method is based on the disagreement-based semi-supervised learning and do not require an off-line training process; also, it can be applied to general videos, and performs very well on a fairly large benchmark dataset. In [21], mutual information was used for the fusion. Here, the proposed fusion approach is based on the disagreements among the trackers. The most related work to our approach is [2], where the co-training idea was used to retrain classification-based trackers. However, [2] followed the standard co-training implementation using one specific type of classifier, SVM. In [19] several tracking algorithms were combined in a Bayesian framework whereas we here emphasize disagreement-based fusion through semi-supervised learning.

In disagreement-based semi-supervised learning, much of the work has been focused on using multi-view features [7] or different data samples [8]. The spirit of all such kind of approaches [7, 22, 8, 23] is to train multiple classifiers with disagreements, and then label the unlabeled instances for each other to update/improve the model. [24] provided PAC bounds with multi-view features, while [12] provided a sufficient condition for multi-view as well as single-view features. Recently, a sufficient and necessary condition was proved for disagreement-based semi-supervised learning, by establishing a connection between disagreement-based and graph-based approaches [25].

In this paper, we emphasize taking advantages of having various well-developed tracking algorithms. In the democratic co-learning framework [26], different al-

gorithms are also used; however, their approach is for classification and a direct voting of all the methods is used. A main difference between tracking and classification is that there is no labeling information provided once the tracking process starts (it is a dynamic system), whereas most disagreement-based semi-supervised learning algorithms can still use labeled data in retraining. Notice that the existence of large disagreements among the classifiers is a premise for the learning or tracking process to continue [12], while the prediction is made by seeking the agreements among the classifiers. For example, in [22, 23], classifiers are learned so that they not only fit the supervised data well, but also themselves reach a high degree of agreement; the tri-training algorithm [8] uses confident and agreed data from two classifiers to help the third classifier. Our agreement is used in the prediction stage like the bootstrapping stage in [26], and we further emphasize the consistency with the information provided by the current tracker. Our work is also related to the active learning literature [27] but we do not have humans in the loop.

3 Disagreement-Based Tracking

The problem of making predictions in a tracking system has its own unique characteristic, and directly applying the standard co-training formulation [2] may not necessarily yield a good solution. Instead, we take advantages of having well-developed existing algorithms, *experts*, and combine them by exploiting the disagreements among the experts. The differences in the intrinsic design of the existing systems will naturally lead to a certain amount of biases/variations, a property the disagreement-based approaches requires [12].

3.1 Prediction of Single Tracker

In this section, we first clarify our notations for a single tracker. A tracker can be viewed as a learner denoted by $h^t = (A, f^t, X^t)$ since it always updates itself. Here, A is a specific tracking method e.g. mean-shift tracker [28], or particle filtering [29]; f^t is the underlying appearance model about the target at time t which can be represented by a discriminative model [5], generative model [4], or template matching [28]; X^t is the position of the target at time stamp t. Given a new image I^{t+1} at time stamp $t+1$, tracker h^t makes a prediction, X^{t+1}, about the position of the target and updates its underlying appearance model to f^{t+1}. We can view tracking a target of a tracker A as computing

$$\begin{aligned} q_A^{t+1}(x) \;&\equiv\; p_A(y_x = +1 | I^{t+1}(x), f^t) \cdot p(X^{t+1} = x, X^t) \\ &\text{and} \sum_x q_A^{t+1}(x) = 1 \end{aligned} \tag{1}$$

Here $y_x = +1$ indicates the occurrence of target at location x and $I^{t+1}(x)$ is an image patch centered at x. Motion coherence is assumed that the prediction on the time stamp $t+1$ is smooth w.r.t. to the prediction on the time stamp t,

for example, $p(X^{t+1} = x, X^t)$ can be a constant within a neighborhood of X_t and zero outside. This corresponds to the local search strategy adopted by most of the trackers.

Now that $q_A^{t+1}(x) \in [0, 1]$ indicates how likely x_{t+1} is the correct position for the target. For an existing tracking algorithm, it may not strictly follow the formulation as in Eq. (1), but we still can use it so long as it outputs a probability map for the prediction.

3.2 Disagreement of Trackers

Suppose we have a set of existing trackers (experts) for making a prediction in a tracking system $S = \{h_i, i = 1..n\}$ with $n \geq 3$ being the number of trackers and each h_i is a tracker trained by tracking algorithm A_i. Given an input I^{t+1} at a time stamp $t + 1$, each tracker h_i computes a $q_i^{t+1}(x)$ to make a prediction of random variable X. Let $p^{t+1}(x)$ denote the ground probability map which indicates how likely x is the correct position, our general objective is to combine the probability maps by different trackers to obtain high probability modes in the "ground-truth" $p^{t+1}(x)$.

A direct way to fuse the multiple trackers is by linearly combining the probability maps together [15]. Here, we call it direct tracker fusion (DTF), which serves as a baseline algorithm:

$$\bar{q}^{t+1}(x) = \frac{1}{n} \sum_{i=1}^{n} q_i^{t+1}(x) \tag{2}$$

with the hope that $\bar{q}^{t+1}(x) \rightarrow p^{t+1}(x)$ as each tracker being unbiased and independent. Algorithms like [15] perform in this way with an adaption in the weighting parameters. The target location is retrieved by $\check{x}^{t+1} = \arg\max_x \bar{q}^{t+1}(x)$. In DTF, at each time, all trackers use the same prediction, $\check{x}^{t+1}$, and each tracker updates its appearance model to f_i^{t+1} based on $\check{x}^{t+1}$ separately and continues the tracking process. Fusing trackers leads to improvement over the original ones (see Section 3.2 and Section 4 for theoretical and empirical justification respectively).

However, predicting the position for the target w.r.t. Eq. (2) has a big drawback, i.e., the average performance $\overline{q}^{t+1}(x)$ of the n trackers may be degenerated by one bad tracker in the group. Here we give an example to illustrate this: suppose there are four trackers f_1, f_2, f_3, f_4 and two candidate positions x^* and $x^{'}$ at $t+1$, where x^* is the correct position for the target. The outputs of the four trackers on the two candidate positions are $q_1(x^{'}) = 0.9$, $q_2(x^{'}) = 0.4$, $q_3(x^{'}) = 0.4$, $q_4(x^{'}) = 0.4$ and $q_1(x^*) = 0.1$, $q_2(x^*) = 0.6$, $q_3(x^*) = 0.6$, $q_4(x^*) = 0.6$. The tracker f_1 is very confident but disagrees with other trackers and makes a wrong prediction. To some extent, this kind of tracker which is confident but disagrees with other trackers can be thought of as a outlier tracker. If we fuse the four trackers with direct tracker fusion (DTF), the position $x^{'}$ will be predicted as the position for the target according to $x = \arg\max_x \overline{q}^{t+1}(x)$. Unfortunately,

we get a wrong position $x^{'}$ due to the outlier tracker f_1 at $t+1$ although three trackers make correct prediction with confidence larger than 0.5.

Let ζ_{t+1} denote the probability mass on such an event that the average performance $\overline{q}^{t+1}(x)$ is degenerated by some outlier tracker f_i, i.e., the other $n-1$ trackers agree with each other and predict the correct position with high confidence while the DTF in Eq. (2) predict the position wrongly due to the outlier tracker f_i at $t+1$. Next, we give the formulation for combining the multiple trackers based on their disagreement to avoid this kind of event for the purpose of robustness. Given n trackers, we still let each tracker perform prediction separately. If the current tracker is confident but disagrees with other trackers while other trackers reach a high degree of agreement, the current tracker is prone to be drifted to the agreed position of other trackers to reach more robust predictions. Our intuition is that we seek a balance between the generated distribution $q_i^{t+1}(x)$ of the current tracker and the degree of agreement by the other trackers as

$$Q_i^{t+1}(x) = (1-\alpha)q_i^{t+1}(x) + \frac{\alpha}{n-1}[\sum_{j=1,j\neq i}^{n} q_j^{t+1}(x)] \cdot \delta(\forall j \neq i, q_j^{t+1}(x) \geq TH) \tag{3}$$

and the specific location by the i-th tracker is $\tilde{x}_i^{t+1} = \arg\max_x Q_i^{t+1}(x)$. TH is a threshold corresponding to a confidence zone. α balances the importance of each tracker's own prediction and the influence from other trackers. The derivation of TH and α will be given in Section 3.2.

Note that the second term is non-zero only when all the other trackers have a high-degree agreement; this is different from the traditional fusion-based tracking [15] where weighted sum is performed; in addition, we emphasize that Eq. (3) focuses mainly on the places with high probability and it is not necessary to fit $p^{t+1}(x)$ at all xs as in the general statistical learning; our disagreement formulation in Eq. (3) can take advantage of this property.

Eq. (3) can be understood as the following: if the current tracker disagree with the other trackers while the other trackers are confident and agree with each other, the prediction of the current tracker will be influenced towards the agreed location (depending upon the overall probability map); otherwise, tracker h_i gives out a prediction as if there were no other trackers. In such a way, the trackers can keep relative independence and also enable *confident* interactions between each other. This makes our approach robust to outlier trackers. In addition, using the agreement of other trackers gives the overall system an ability to be self-aware of when the system starts to drift. This happens when all trackers have high entropy of $q_i^{t+1}(x)$ with large disagreement.

The overall output is then given by $x^{t+1\,*} = \arg\max_x \mathcal{Q}^{t+1}(x)$ and

$$\mathcal{Q}^{t+1}(x) = \frac{1}{n}\sum_{i=1}^{n} Q_i^{t+1}(x) \tag{4}$$

Note that $x^{t+1\,*}$ is the output of the overall system but it does not participate in the retraining of the individual trackers. The pseudo code of disagreement-based tracking is shown in Fig. 1. Tracking based on the disagreements among the trackers shows advantage over using a direct combination and we justify this point both theoretically and empirically in the following sections.

Given n trackers $\{h_i, i = 1..n\}$, each tracker $h_i = (A_i, f_i^t, X_i^t)$ adopts a specific tracking method A_i. At the time stamp $t = 0$, a target is manually identified located at X^0. All trackers start with the same X^0 and obtain their appearance model f_i^0. Given a new image I^{t+1} at time stamp $t + 1$,

- Each tracker h_i searches a local neighborhood around X_i^t and generate a probability map q_i^{t+1} using Eq. (1).
- Find modes of $\tilde{x}_i^{t+1}$ for $Q_i^{t+1}(x)$ as in Eq. (3). $\tilde{x}_i^{t+1}$ keeps a balance between the estimation of the current tracker and the level of agreements among other trackers.
- Assign $X_i^{t+1} = \tilde{x}_i^{t+1}$, sample patches around X_i^{t+1} and update the appearance model of each tracker to f_i^{t+1} using the embedded model updating/learning rule in A_i
- Based on $x^{t+1\,*} = \arg\max_x \sum_i Q_i^{t+1}(x)$, report the $x^{t+1\,*}$ as the tracking result for disagreement-based tracking (DBT).

Fig. 1. Pseudo code of disagreement-based tracking.

Theoretical Justification We first show that a linear combination of multiple trackers as in Eq. (2), direct tracker fusion (DTF), gains improvement over the individual systems. Let $p^{t+1}(x)$ denote the ground truth which indicates how likely x is the correct position, and let $q_i^{t+1}(x) \in [0, 1]$ be the output of algorithm A_i.

Lemma 1. *If we take an average of the predictions from all the experts:* $\overline{q}^{t+1}(x) = \frac{1}{n}\sum_{i=1}^{n} q_i^{t+1}(x)$ *as in Eq. (2), then the average is bounded in a PAC sense. We suppose that the* n *trackers are independent and unbiased: then* $\overline{q}^{t+1}(x) \to p^{t+1}(x)$ *as* $n \to +\infty$.

Proof. For any small $\epsilon > 0$, with Hoeffding inequality, we get that $P(|\overline{q}^{t+1}(x) - p^{t+1}(x)| \geq \epsilon) \leq 2\exp(-2n\epsilon^2)$. □

Lemma 1 shows that $\overline{q}^{t+1}(x)$ can converge to the ground truth $p^{t+1}(x)$ exponentially. Let $error_i^{t+1}$ denote the error rate of the tracker f_i at $t + 1$, i.e., the probability that f_i predicts a wrong position for the target at $t + 1$, $error_{min}^{t+1} = \min_i\{error_i^{t+1}\}$ and $error_{max}^{t+1} = \max_i\{error_i^{t+1}\}$, we give the following theorem to show that fusing the multiple trackers according to Eq. (3) and Eq. (4) will improve the performance at least $\zeta_{t+1} - n(error_{max}^{t+1})^{n-1}$, contrasting to the direct tracker fusion (ζ_{t+1} was defined in the previous section).

Theorem 1. *If we fuse the multiple trackers according to Eq. (3) and Eq. (4) where* $\alpha \geq \frac{2}{3}$ *and* $TH \geq \frac{1}{2}$*, contrasting to the direct tracker fusion in Eq. (2), the performance at* $t+1$ *can be improved at least* $\zeta_{t+1} - n(error_{max}^{t+1})^{n-1}$*, where* ζ_{t+1} *is the probability of the event that the average performance* $\overline{q}^{t+1}(x)$ *is degenerated by some outlier tracker.*

Proof. Let $\mathcal{X}^{t+1}$ denote the set of candidate positions at $t+1$. If there is some $x^* \in \mathcal{X}^{t+1}$, at which $q_i^{t+1}(x^*) \geq TH$ for all $i \in \{1, \ldots, n\}$, it is easy to find that such x^* is unique, since $TH \geq 0.5$ (Here we neglect the probability mass on the event that at $t+1$ there are two positions x and $x^{'}$ at which $q_i^{t+1}(x) = q_i^{t+1}(x^{'}) = \frac{1}{2}$). Considering Eq. (3) we get $\mathcal{Q}^{t+1}(x^*) = \frac{1}{n}\sum_{k=1}^{n} q_k^{t+1}(x^*)$, and x^* will be selected as the tracking result, no matter whether $\arg\max \mathcal{Q}^{t+1}(x)$ or $\arg\max \overline{q}^{t+1}$ is used. If for any $x \in \mathcal{X}^{t+1}$ there are less than $n-1$ trackers with $q_i^{t+1}(x) \geq TH$, then the second term of Eq. (3) is zero. So for any $x \in \mathcal{X}^{t+1}$, $\mathcal{Q}^{t+1}(x) = \frac{1-\alpha}{n}\sum_{k=1}^{n} q_k^{t+1}(x)$. Predicting the tracing result according to $\arg\max \mathcal{Q}^{t+1}(x)$ is equal to predicting according to $\arg\max \overline{q}^{t+1}$. Next we analyze the situation when there is some $\widehat{x} \in \mathcal{X}^{t+1}$, at which $q_i^{t+1}(\widehat{x}) < TH$ and $q_j^{t+1}(\widehat{x}) \geq TH$ for all $j \neq i$. Obviously, such $\widehat{x}$ is also unique, since $TH \geq 0.5$ and $n \geq 3$.

Case 1: $\widehat{x}$ is the correct position for the target at $t+1$. We will show that even if $q_i^{t+1}(\widehat{x})$ is very close to 0, i.e., tracker f_i is an outlier at $t+1$, it will not degenerate the fusion of the multiple trackers due to Eq. (3). We obtain

$$\begin{aligned} &Q_i^{t+1}(\widehat{x}) = (1-\alpha)q_i^{t+1}(\widehat{x}) + \\ &\frac{\alpha}{n-1}\sum_{j\neq i} q_j^{t+1}(\widehat{x}) \geq (1-\alpha)q_i^{t+1}(\widehat{x}) + \alpha \cdot TH \end{aligned} \tag{5}$$

$$Q_{j,j\neq i}^{t+1}(\widehat{x}) = (1-\alpha)q_j^{t+1}(\widehat{x}) \geq (1-\alpha)\cdot TH$$

Thus,

$$\begin{aligned} \sum_{k=1}^{n} Q_k^{t+1}(\widehat{x}) \geq\ & (1-\alpha)q_i^{t+1}(\widehat{x}) + \alpha \cdot TH + \\ & (1-\alpha)(n-1)TH \end{aligned} \tag{6}$$

For an incorrect position $x' \neq \widehat{x}$, since $\sum_{x\in\mathcal{X}^{t+1}} q_k^{t+1}(x) = 1$, it is easy to see that

$$\begin{aligned} Q_i^{t+1}(x') &= (1-\alpha)q_i^{t+1}(x') \leq (1-\alpha)(1-q_i^{t+1}(\widehat{x})) \\ Q_{j,j\neq i}^{t+1}(x') &= (1-\alpha)q_j^{t+1}(x') \leq (1-\alpha)(1-q_i^{t+1}(\widehat{x})) \\ &\leq (1-\alpha)(1-TH) \end{aligned} \tag{7}$$

Therefore,

$$\begin{aligned} \sum_{k=1}^{n} Q_k^{t+1}(x^{'}) \leq\ & (1-\alpha)(1-q_i^{t+1}(\widehat{x})) \\ & + (1-\alpha)(n-1)(1-TH) \end{aligned} \tag{8}$$

We see that in general $\sum_{k=1}^{n} Q_k^{t+1}(\widehat{x}) \geq \sum_{k=1}^{n} Q_k^{t+1}(x^{'})$ for $\alpha \geq \frac{2}{3}$ and $TH \geq \frac{1}{2}$. This makes the correct position more robust, i.e., the prediction of the disagreement-based tracking will never be influenced even if f_i is a outlier tracker. So the improvement is at least ζ_{t+1}.

Case 2: $\widehat{x}$ is not the correct position for the target at $t+1$. Since $q_j^{t+1}(\widehat{x}) \geq TH$ for all $j \neq i$ and $TH \geq \frac{1}{2}$, $n-1$ trackers predict the wrong position $\widehat{x}$ as the tracking result at $t+1$. Now we bound the probability of such event. Since $error_j^{t+1} \leq error_{max}^{t+1}$ and the multiple trackers are assumed to be independent, the probability mass on the event that $n-1$ trackers predict the position mistakenly is at most $n(error_{max}^{t+1})^{n-1}$. The worst situation is that the fusion according to Eq. (3) performs worse than the direct tracker fusion in **case 2** completely. We get Theorem 1 proved. □

From Theorem 1 we know that the fusion will get benefit from Eq. (3) under the situation that one bad tracker degenerates the direct tracker fusion. When n (the number of the trackers) is large, it would be difficult for the remaining $n-1$ trackers to achieve some agreement (See the experiment "Non-Relax" in Section 4). In practice, we can relax this constraint, e.g., when two or more trackers achieve agreement, the agreement term would take effect.

Note that the performance of Eq. (2) and Eq. (3) depends on the correlation between the trackers. The correlation depends on two factors: (1) the intrinsic design of the trackers; (2) the training samples used to train the trackers. Two trackers with the same design trained on the same set of samples are highly correlated, and two different types of trackers trained on the same set of samples are more correlated than those trained on different set of samples. If the n trackers are the same, then using Eq. (3) shows no advantage over Eq. (2).

In summary, Lemma 1 suggests that fusing the multiple experts directly might gain exponential improvement, contrasting to the single tracker; Theorem 1 shows that our disagreement-based fusion method can provide more robustness to the tracking system, which motivates the use of Eq. (3) by keeping a balance between the current expert f_i and the agreement from the other experts.

4 Experiments

In the experiments, we make a comprehensive comparison between the performance of disagreement-based tracking, direct tracker fusion, and the individual trackers. Four trackers are used and the experiment is conducted on 11 commonly tested videos (listed in Table 1). The trackers used are MilTracker [5], the semi-supervised on-line boosting tracker (semiBoost)[3], Incremental Visual Tracker (IVT)[4], and Incremental Visual Tracker using edge information (IVTE).

Since the individual trackers perform prediction separately, the computational complexity of the proposed method only adds slight overhead over the individual ones with the multi-core processor and parallel computing. Compared with the large performance gain, this computational overhead is tolerable.

From the experiments, we observe that, statistically, each individual tracker gets significantly improved by using Eq. (3): the average center location error

has been reduced by more than 12 pixels and the success rate has increased by $20\% \sim 40\%$. The result of DBT (Disagreement-Based Tracking) also outperforms the system by directly combining the original trackers, i.e., DTF, with 3.3 pixels reduction in center location error and 4.4% improvement in success rate. DBT also significantly outperforms PROST [30].

4.1 Implementation details

While a wealthy body of tracking papers/systems have been reported, we found a few systems (with available source code) having decent performance on general videos. Here, we provide brief descriptions for these trackers we used with necessary changes made to them.

MilTracker adopts an online multiple instance learning algorithm to train a discriminative classifier. In order to handle the ambiguity of sampled patches, a bag of potentially positive image patches are extracted. MilTracker maintains a pool of Haar features and the online boosting mechanism is adopted.

SemiBoostTracker also adopts an online boosting mechanism and it formulates the update process in a semi-supervised fashion combined with a given prior. This helps to alleviate the drifting problem.

The IVT incrementally learns a low dimensional eigenspace representation to model the appearance changes of the object. In IVT model, the target is represented as a vector of gray-scale value, and the motion is modeled by an affine image warping. To propagate sample distributions over time, a particle filter framework is adopted. Since both MILTracker and SemiBoostTracker do not support affine transformation, we disabled the scaling and rotating ability of IVT. IVTE is similar to IVT, except that it uses level set as the feature.

The forms of the 4 tracking systems' outputs are rather different. MILTracker and SemiBoostTracker produce scores on local search regions; IVT and IVTE propagate probabilities via particles. In the experiment, we map the scores of MILTracker and SemiBoostTracker to the range $[0, 1]$ to produce probability maps q_{MIL} and q_{SBT} (The probability maps are normalized to make sure that $\sum_x q_A^{t+1}(x) = 1$). For IVT and IVTE, we keep the position entries (a_i, b_i) of the particle and use a parzen window approach to estimate the probability for prediction. For a point $x = (a, b)$ on the image, its probability is calculated as $q_{IVT}(x) = \sum_{i=1}^{M} w_i * \max\{0, 1 - \sqrt{(a-a_i)^2 + (b-b_i)^2}/L\}$. In our experiment, L is set to 15. A map q_{IVTE} is produced similarly as q_{IVT} for IVTE.

Based on q_{MIL}, q_{SBT}, q_{IVT}, and q_{IVTE} we respectively compute the corresponding Q_{MIL}, Q_{SBT}, Q_{IVT}, and Q_{IVTE} using Eq. (3) (Since 4 trackers are used, we relaxed Eq. (3) that when 2 trackers achieve confident agreement, the agreement term will take effect) and thus, each tracker makes its own prediction separately. As we have mentioned before, $\tilde{x}_i^{t+1}$ found by mean shift algorithm [28] can represent multiple points (modes). For each tracker, e.g., MILTracker, the one mode with the maximum value is reported as its prediction, significant modes found are used to retrain the tracker and as the seeds for further search at the next time stamp. For the results reported in our experiment, α is set to be 0.67 as suggested by the theoretical section. The threshold TH in Eq. (3) is

set as $0.8/(\Lambda/3)$ (Λ is the size of the search window). For each tracker, 2 modes are kept.

4.2 Quantitative Results

The average center location error The average center location error is a commonly used metric to measure the performance of tracking and is defined as the average error between the predicted locations to the ground truth. In Table 1, we summarize the results of the average center location error on all the 11 videos. It's clear that our disagreement based tracker outperforms the individual trackers, Direct tracker fusion, and the co-training scheme. For non-relax (using Eq. (3) directly without relaxation), it's less possible for the trackers to achieve some agreements, and the chance of interaction between trackers is reduced. Still the result is better than the individual trackers.

Table 1. Comparison of Average Center Location Error. Non-Relax indicates to use Eq. (3) directly without relaxation;Co-Training stands for the results using co-training method.

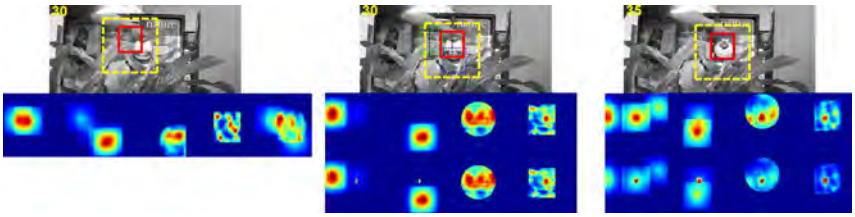
videos	MilT	IVT	IVTE	SBT	DTF	Co-Training	Non-Relax	DBT
Girl	31.9	25.2	18.1	19.3	20.6	39.8	23.3	**13.4**
CokeCan	20.5	55.3	11.0	14.9	9.3	49.0	7.9	**6.6**
Tiger1	**15.9**	71.9	56.6	20.9	37.7	64.1	49.0	31.2
Sylv	10.9	44.0	19.5	16	19.5	31.7	**7.3**	10.8
StatOcc	27.8	3.3	4.8	74.4	**2.5**	41.2	26.2	3.0
David	22.9	4.9	16.9	26.4	7.0	9.2	7.9	**4.1**
Cliffbar	12.0	31.4	78.6	29.9	27.1	45.6	16.7	**8.5**
Surfer	9.2	6.7	23.9	67.6	5.1	**4.7**	5.1	4.8
faceocc2	20.1	14.2	9.1	17	6.5	12.4	12.0	**6.1**
Indoor	17.2	30.2	193.5	116.5	4.7	61.9	10.7	**4.5**
faceocc	27.1	11.8	11.3	**6.8**	8.9	23.3	**7.6**	9.7
In all	20.8	21.8	22.3	37.3	11.5	32.7	15.8	**8.2**

The success rate If the location error on one frame is less than a pre-specified threshold, the prediction is regarded as a successful prediction. The success rate is defined as the ratio of successful predictions over all the predictions. To compute the success rate, we set T as certain ratio of the average width of the target, i.e., $T = \beta * (w + h)/4$, where, w and h are the width and height of the target respectively. Conceptually, this is similar to the overlap score evaluation used in [30] and thus, success rate is conceptually similar to the tracked percentile in [30]. We observe that, the trackers can not track the target precisely at all times, if there is no overlap but the prediction of the tracker is not far from the target or within the search area of the tracker, it's often possible for the tracking process to recover. Our evaluation measurement can reflect such a phenomenon. We compare the success rate in Table 2 and from Table 2, the DBT achieves the highest success rate.

Table 2. Comparison of success rate when $\beta = 0.5$.

MilT	IVT	IVTE	SBT	DTF	Non-Relax	DBT
0.596	0.733	0.753	0.592	0.862	0.84	0.900

Comparison of probability maps In Fig. 2, we show how the probability maps are generated in Eq. (2) and Eq. (3) on a testing video, Tiger1. DTF drifts from the 30th frame; on this frame, the predictions of the trackers are rather diverse. In DTF, however, the individual tracker's prediction is not fully respected and it has to comply with the voted prediction; this is the primary reason for drifting and getting trapped; using Eq. (3) however leads to a more robust prediction. As we can see from the second figure, on the 30th frame, MILTracker and Semiboost tracker achieves certain agreement, but IVT and IVTE still complies with its own prediction since the agreement is weak. On the 35th frame, when MILTracker, IVT and Semiboost tracker achieve confident agreement, IVTE is pulled back to the agreed position and the four trackers merge again. The benefit of our disagreement-based tracking is obvious: the trackers then keep their relatively different traces and the risk of getting trapped is reduced.



(a) (b) (c)

Fig. 2. Illustration of the probability maps where four trackers (experts) are adopted (the figures have been scaled for visualization). (a) shows the results by DTF. (b) and (c) display the probability maps generated by disagreement-based tracking. The probability maps inside the dashed yellow rectangle are shown below the screen shots. Underneath each figure, from left to right, the probability maps are IVT, IVTE, MILTracker, Semiboost tracker respectively (see the discussions about these trackers in the experiments). For DTF in (a), the fifth probability map is the combined map. For disagreement-based tracking in (b) and (c), the first rows shows the original probability maps, and the second row shows the Q_i^{t+1} computed by Eq. (3).

Comparison with other methods PROST [30] is another fusion based tracking algorithm that adopts 3 trackers (a template model, an optical-flow based mean-shift tracker and an online random forest tracker). Table 3 and Table 4 compare the average location errors and the tracked percentage (computed using the overlap score in [30]) with PROST. From the two tables, we can observe that, our disagreement-based tracking outperforms PROST and achieves better tracked percentages on most of the videos.

The best experimental performance of Democratic Integration in [15] was achieved by using uniform qualities, which assigned equal weights to all the

Table 3. Comparison of average center location error with [30]

Method	Girl	tiger1	sylv	David	faceocc	faceocc2
[30]	19.0	**7.2**	**10.6**	15.3	**7.0**	17.2
Ours	**13.4**	31.2	10.8	**4.1**	9.7	**6.1**

Table 4. Comparison of tracked percentage with [30]

Method	Girl	tiger1	sylv	David	faceocc	faceocc2
[30]	89	**79**	73	80	**100**	82
Ours	**97**	30	**83**	**100**	**100**	**100**

clues, and corresponded directly with DTF. In addition, we implemented the quality measure of normalized saliency and the performance was not as good as DBT: their average center location error is 13.8 with success rate 0.87.

We did not get the implementation of [2]. Nevertheless, we did experiment on some videos used in [2] and the results of DBT are better than [2] qualitatively (skipped here due to page limit). Moreover, we indeed implemented co-training and reported the result in Table 1 (average center location error 32.7), which is much worse than DBT.

Table 5. Performance by varying α and TH ($TH = R/(\Lambda/3)$)

R/α	0.8/0.3	0.8/0.67	0.8/0.85	0.7/0.67	0.9/0.67
Average Error	10.0	8.2	9.8	9.95	9.8
Success Rate	0.875	0.90	0.895	0.87	0.89

Robustness by varying the parameters In table 5, we summarize the performance of disagreement-based tracking by varying the parameters α and TH, which are the two key parameters in Eq. (3). We can see from this table, by varying TH and α, the results (especially the success rate) do not change too much. This demonstrates the robustness of disagreement-based tracking. The average center location error has relatively larger change because on portions of the videos Tiger1 and Indoor, disagreement-based tracking gets distracted to positions distant from the targets. In such cases, the success rate does not vary too much, but the center location error is increased.

Screenshots of the results In Fig. 3, we compare the tracking results on the video Girl. This video undergoes several challenges: fast appearance change and occlusion. Although both MILTracker and IVTE can track the face of the girl successfully, the tracking process is not very stable. IVT drifts from the face at the 20th frame. From the 391th frame, direct tracker fusion also drifts and get trapped at the background of the images. As can be seen from both the screen shots and the error plot on the right of Fig. 3, we find that disagreement-based tracking tracks most robustly and accurately. In Fig. 3, we can also find

Fig. 3. Comparison of tracking results on the video Girl. The first row on the left shows the results of disagreement-based tracking; the second row on the left shows the results of 4 individual trackers and direct tracker fusion; the right plot shows the comparison of center location error (the number of the x axis denotes the number of predictions). The coloring scheme is: dotted green: IVT, dotted black: MILTracker, dotted blue: IVTE, dotted yellow: Semiboost tracker, solid Red: disagreement-based tracking, and solid magenta: Direct tracker fusion.

a very nice property of democratic tracking that, the traces of the four trackers are similar but not exactly the same, thus, they can explore different spaces, recommend confident samples to other trackers, and thus avoid to be trapped at an incorrect position.

5 Conclusion

In this paper, we have introduced a disagreement-based tracking method which fuses multiple existing tracking systems in the following way that seeks a balance between the coherence of the current tracker and the degree of agreements among other trackers. In such a way, it enables the interaction between trackers and keeps the appealing characteristics of the trackers at the same time. As illustrated in the experiments, the balance complies with the characteristic of tracking. Disagreement-based tracking can be built on top of various existing well-developed tracking systems utilizing their intrinsic biases. Adopting several state-of-the-art tracking algorithms, our approach is able to improve each of them by a large margin on widely used benchmark videos in the literature

Acknowledgement. Zhuowen Tu and Quannan Li are supported by Office of Naval Research Award N000140910099 and NSF CAREER award IIS-0844566; Wei Wang, Yuan Jiang and Zhi-Hua Zhou are supported by National Natural Science Fund of China 60975043.

References

1. Avidan, S.: Ensemble tracking. In: CVPR. (2005)
2. Tang, F., Brennan, S., Zhao, Q., Tao, H.: Co-tracking using semi-supervised support vector machines. In: ICCV. (2007)
3. Grabner, H., Leistner, C., Bischof, H.: Semi-supervised on-line boosting for robust tracking. In: ECCV. (2008)
4. Lim, J., Ross, D., Lin, R.S., Yang, M.H.: Incremental learning for visual tracking. Int'l J. of Comp. Vis. (2008)

5. Babenko, B., Yang, M.H., Belongie, S.: Visual tracking with online multiple instance learning. In: CVPR. (2009)
6. Zhou, Z.H., Li, M.: Semi-supervised learning by disagreement. Knowledge and Information Systems **24** (2010) 415–439
7. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: COLT. (1998) 92–100
8. Zhou, Z.H., Li, M.: Tri-training: Exploiting unlabeled data using three classifiers. IEEE Trans. on Know. and Data Eng. (2005)
9. Dietterich, T.G.: Ensemble methods in machine learning. In: INTERNATIONAL WORKSHOP ON MULTIPLE CLASSIFIER SYSTEMS, Springer-Verlag (2000) 1–15
10. Kuncheva, L.I.: A theoretical study on six classifier fusion strategies. IEEE Tran. on PAMI **24** (2002) 281–286
11. Bennett, K.P., Demiriz, A., Maclin, R.: Exploiting Unlabeled Data in Ensemble Methods. In: ACM SIGKDD '02 Edmonton, Alberta CA. (2002)
12. Wang, W., Zhou, Z.H.: Analyzing co-training style algorithms. In: ECML. (2007)
13. Wu, Y., Huang, T.: Robust visual tracking by integrating multiple cues based on co-inference learning. In'l J. of Comp. Vis (2004)
14. Siebel, N., Maybank, S.: Fusion of multiple tracking algorithms for robust people tracking. In: ECCV. (2002)
15. Triesch, J., von der Naksvyrg, C.: Democratic integration: Self-organized integration of adaptive visual cues. In: Neuroal Computation. (2001)
16. Spengler, M., Schiele, B.: Towards robust multi-cue integration of visual tracking. In: MVA. (2003)
17. Kwon, J., Lee, K.M.: Visual tracking decomposition. In: CVPR. (2010)
18. I Leichter, M Lindenbaum, E.R.: A general framework for combining visual trackersthe 'black boxes' approach. It'l J. of Comp. Vis (2006)
19. Zhong, B., Yao, H., Chen, S., Ji, R.R., Yuan, X., Liu, S., Gao, W.: Visual tracking via weakly supervised learning from multiple imperfect oracles. In: CVPR. (2010)
20. Stenger, B., Woodley, T., Cipolla, R.: Learning to track with multiple observers. In: CVPR. (2009)
21. Mundy, J.L., Chang, C.F.: Fusion of intensity, texture, and color in video tracking based on mutual information. In: AIPR. (2010)
22. Collins, M., Singer, Y.: Unsupervised models for named entity classification. In: EMNLP. (1999)
23. Leskes, B., Torenvliet, L.: The value of agreement a new boosting algorithm. J. of Comp. and Sys. Sci. (2008)
24. Dasgupta, S., Littman, M.L., McAllester, D.: Pac generalization bounds for co-training. In: NIPS. (1999)
25. Wang, W., Zhou, Z.H.: A new analysis of co-training. In: ICML, Haifa, Israel (2010) 1135–1142
26. Zhou, Y., Goldman, S.: Democratic co-learning. In: Inte'l Conf. on Tools with Art. Intell. (2004)
27. Dasgupta, S.: Coarse sample complexity bounds for active learning. In: NIPS. (2005)
28. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: CVPR. (2000)
29. Isard, M., Blake, A.: Condensation - conditional density propagation for visual tracking. Int'l J. on Comp. Vis. (1998)
30. Santner, J., Leistner, C., Saffari, A., Pock, T., Bischof, H.: Prost: Parallel robust online simple tracking. In: CVPR. (2010)