Chapter 6

Active Collaboration of Classifiers for Visual Tracking

Kourosh Meshgi and Shigeyuki Oba

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.74199

Abstract

Recently, discriminative visual trackers obtain state-of-the-art performance, yet they suffer in the presence of different real-world challenges such as target motion and appearance changes. In a discriminative tracker, one or more classifiers are employed to obtain the target/nontarget label for the samples, which in turn determine the target's location. To cope with variations of the target shape and appearance, the classifier(s) are updated online with different samples of the target and the background. Sample selection, labeling, and updating the classifier are prone to various sources of errors that drift the tracker. In this study, we motivate, conceptualize, realize, and formalize a novel active co-tracking framework, step by step to demonstrate the challenges and generic solutions for them. In this framework, not only classifiers cooperate in labeling the samples but also exchange their information to robustify the labeling, improve the sampling, and realize efficient yet effective updating. The proposed framework is evaluated against state-of-the-art trackers on public dataset and showed promising results.

Keywords: visual tracking, active learning, active co-tracking, uncertainty sampling

1. Introduction

Visual tracking is one of the building blocks of human-robot interaction. Implicit or explicit, this task is embedded in many high-level complicated tasks of the robot: automating industrial workcells [1], attending the speaker in a multimodal spoken dialog system [2], following the target [3] and vision-based robot navigation [4], aerial visual servoing [5], imitating the behavior of a human [6], extracting tacit information of an interaction [7], sign-language interpretation [8], and autonomous driving as well as simpler tasks such as human-robot cooperation [9], obstacle avoidance [10], first-person view action recognition, [11] and human-computer interfaces [12].

The most general type of tracking is single-object model-free online tracking, in which the object is annotated in the first frame and tracked in the subsequent frames with no prior knowledge about the target's appearance, its motions, the background, the configurations of the camera, and other conditions of the scene. Visual tracking is still considered as a challenging problem despite numerous efforts made to address abrupt appearance changes of the target [13], complex transformations [14] and deformations [15, 16], background clutter [17], occlusion [18], and motion artifacts [19].

Generative trackers attempt to construct a robust object appearance model or to learn it on the fly using advanced machine learning techniques such as subspace learning [20], hash learning [21], dictionary learning [22], and sparse code learning [13]. General object tracking is the task of tracking arbitrary objects through one-shot learning, typically with no *a priori* knowledge about the target's geometry, category, or appearance. Called model-free tracking, the task is to learn the target appearance and update it by adjusting to target's changes on the fly. To this end, discriminative models focus on target/background separation using correlation filters [23–25] or dedicated classifiers [26], which assist them to dominate the visual tracking benchmarks [27–29]. Using tracking-by-detection approaches is a popular trend in recent years, due to significant breakthroughs in object detection domain (deep residual neural networks [30], for instance), yielding strong discriminating power with offline training. Adopted for visual tracking, many of such trackers are adjusted for online training and accumulate knowledge about a target with each successful detection (e.g., [26, 31–33]).

Tracking-by-detection methods primarily treat tracking as a detection problem to avoid having model object dynamics especially in the case of sudden motion changes, extreme deformations, and occlusions [34, 35]. However, there is a multitude of drawbacks in the tracking-by-detection setting:

1. *Label noise*: inaccurate labels confuse the classifier [15] and degrade the classification accuracy [34]. The labeler is typically built upon heuristics and intuitions, rather than using the accumulated knowledge about the target.

2. *Self-learning loop*: the classifier is retrained by their own output from earlier frames, thus accumulating error over time [35].

3. *Uniform treatment of samples*: equal weight for all samples in evaluating the target [36] and training the classifier [37], despite the uneven contextual information in different samples. The classifier is trained using all the examples with equal weights, meaning that negative examples which overlap very little with the target bounding box are treated equally as those negative examples with significant overlaps.

4. *Stationarity assumption*: assuming a stationary distribution of the target appearance does not hold for most of the real-world scenarios with drastic target appearance changes [35]. In the context of visual tracking, the non-stationarity means that the appearance of an object may change so significantly that a negative sample in the current frame looks more similar to a positive example in the previous frames.

5. *Model update difficulties*: adaptive trackers inherently suffer from the drifting problem. Noisy model update [38] and the mismatch between model update frequency and target

	T0	T1	T2	T3	T4	T5	T6
Online update		✓	✓	✓	✓	✓	✓
Co-tracking				✓	✓	✓	✓
Active learning					✓	✓	✓
Dual memory						✓	✓
Ensemble							✓

Table 1. Trackers introduced in this chapter: **T0**, a part-based tracker without model update; **T1**, the part-based tracker with model update; **T2**, a KNN-based tracker with color and HOG features; **T3**, co-tracking of KNN-based classifier T2 and part-based detector T1; **T4**, active co-tracking of T1 and T2 with online update; **T5**, active asymmetric co-tracking of short-memory T1 and long-memory T2 (modified from [40]); and **T6**, active ensemble co-tracking of bagging-induced ensemble and long-memory T2 (modified from [41]).

evolution rate [39] are two major challenges of the model update. If the update rate is small, the changes of the target are not reflected into target's template, whereas rapid update of the tracker renders it vulnerable to data noise and small target localization errors. This phenomenon is also known as *stability plasticity dilemma*.

In this study we motivate, conceptualize, realize, and formalize a novel co-tracking framework. First, the importance of such system is demonstrated by a recent and comprehensive literature review. Then a discriminative tracking framework is formalized to be evolved to a co-tracking by explaining all the steps, mathematically and intuitively. We then construct various instances of the proposed co-tracking framework (**Table 1**), to demonstrate how different topologies of the system can be realized, how the information exchange is optimized, and how different challenges of tracking (e.g., abrupt motions, deformations, clutter) can be handled in the proposed framework. Active learning will be explored in the context of labeling and information exchange of this co-tracking framework to speed up the tracker's convergence while updating the tracker's classifiers effectively. Dual memory is also proposed in the co-tracking framework to handle various tracking scenarios ranging from camera motions to temporal appearance changes of the target and occlusions.

It should be noted that preliminary results of this research were published in [40, 41]; however, the results presented here are slightly different because of using different feature-based auxiliary classifier, different target estimations, and ROI-detection scheme (that was omitted here to conserve the flow of the progressive system design).

2. Tracking by detection

Typically tracking-by-detection method consists of five major steps: **SAMPLING**, **CLASSIFYING**, **LABELING**, **ESTIMATING**, **UPDATING**.

SAMPLING: To obtain the positive sample(s) and negative samples (the target and the background, respectively), dense or sparse (stochastic) sampling is performed either around last known target position (using Gaussian distributions, particle filters, or various motion models) or around the saliencies or key points in the current frame [21]. Adaptive weights for the samples based on their appearance similarity to the target [42], occlusion state [18], and spatial distance to previous target location [43] have been considered; especially in the context of tracking by detection, boosting [44] has been extensively investigated [45–47].

CLASSIFYING: The classification module of tracking-by-detection schemes utilizes offline-trained classifiers or online supervised learning methods to classify the target from its background (e.g., [48]). To robustify this module especially against label noise, supervised learning with robust loss functions [46, 49] and semi-supervised [39, 50] and multi-instance [47, 51, 52] learning approaches are considered. Efficient sparse sampling [53], leveraging context information [17, 54], considering sample information content for the classifier [55], and landmark-based label propagation [43] are among other proposed approaches to address this issue. Another interesting approach is to reformulate to couple the labeling and updating process to bridge the gap between the objectives of these two steps, as labeling aims for predicting binary sample labels, whereas updating typically tries to estimate object location [15]. The label noise problem amplifies when the tracker does not have a forgetting mechanism or a way to obtain external scaffolds (i.e., self-learning loop). This inspired the use of co-tracking [34], ensemble tracking [56, 57], or label verification schemes [58] to break the self-learning loop using auxiliary classifiers.

LABELING: The result of classification process provides the target/background label for each sample, a process which can be enhanced by employing an ensemble of classifiers [56, 57], exchanging information between collaborative classifiers [34], and verifying labels by auxiliary classifiers [58] or landmarks [43].

ESTIMATING: The state of the target, i.e., the location and scale of the target usually described with a bounding box, is then determined by selecting the sample with the highest classification score [15], calculating the expectation of target state [41], or performing an estimated bounding box regression [59].

UPDATING: Updating the classifier is another challenge of the tracking-by-detection schemes. Updating the classifier, with the data labeled by itself previously in a closed-loop (known as self-learning loop), is susceptible to drift from the original data distribution because a tiny error or a small noise can be amplified. Therefore along with many types of research to revalidate the data labels (such as [58]), the importance of having a "teacher" to guide the classifier during training is discussed in literature [39]. Cooperative classifiers in frameworks such as ensembles of homogeneous or heterogeneous classifiers [60], co-learning [34], and hybrids of generative and discriminative models [61] are some of the approaches to provide this guidance through cooperation. Furthermore, feature selection based on its discrimination ability [45], replacing the weakest classifier of an ensemble [45] or the oldest one [60], or applying a budget on the sample pool (hence, keeping only some prototypical samples) [15, 43] is proposed to improve the performance of such solutions.

On top of that, the frequency of update is another important role player in tracker's performance [39]. Higher update rates capture the rapid target changes but is prone to occlusions, whereas slower update paces provide a long memory for the tracker to handle temporal target variations but lack the flexibility to accommodate permanent target changes. To this end, researchers try to combine long- and short-term memories [62] and role-back improper updates [57] or utilize different temporal snapshots of the classifier to overcome non-stationary distribution of the target's appearance [63]. This pipeline, however, was altered in some studies to introduce desired properties, e.g., to avoid label noise by merging sampling and labeling steps [15].

2.1. Formalization

Online visual tracking is the task to update the state vector $\mathbf{p}_t$ involving location, size, and shape of the bounding box, at each observation of video frame $t = 1, \dots, T$. The update process is sometimes written with transformation $\mathbf{y}_t$ that transforms the previous state vector $\mathbf{p}_{t-1}$ to the current state $\mathbf{p}_t = \mathbf{p}_{t-1} \circ \mathbf{y}_t$.

In tracking-by-discrimination framework, we utilize a classifier θ_t that discriminates an image patch $\mathbf{x}$ into either target or background, where the classifier is denoted as a real valued discriminant function $h(\mathbf{x}|\theta_t) \in \mathbb{R}$ and the function value $s = h(\mathbf{x}|\theta_t)$ is called a discrimination score or, in short, score. The patch $\mathbf{x}$ (i.e., the area of the image bounded by the bounding box $\mathbf{p}_t$) is labeled as target if $s > \tau$ with a threshold τ and as background if $x < \tau$. A typical procedure of the tracking-by-discrimination is written as follows.

SAMPLING: The samples are defined using these transformations, and their corresponding image patches $\mathbf{x}_t^j \in \mathcal{X}_t$ are selected from image. We obtain N samples of state $p_t^j, j = 1, \dots, N$ by drawing random transformations $\mathbf{y}_t^j \in \mathcal{Y}_t$ using dense or sparse sampling strategy, transforming the previous state p_{t-1} with a transformations $\mathbf{y}_t^j$ as $\mathbf{p}_t^j = \mathbf{p}_{t-1} \circ \mathbf{y}_t^j \in \mathcal{P}_t$.

CLASSIFYING: We calculate the score s_t^j of the image patches $\mathbf{x}_t^{\mathbf{p}_t^j}$ corresponding to all samples, or bounding boxes, using the current classifier θ_t ($h : \mathcal{X} \to \mathbb{R}$):

$$s_t^j = h\left(\mathbf{x}_t^{\mathbf{p}_{t-1} \circ \mathbf{y}_t^j} | \theta_t\right) \tag{1}$$

LABELING: We determine label l_t^j of each sample j using the score of the sample. If the score is above a threshold τ, the sample is likely to be target match:

$$l_t^j = \text{sign}\left(s_t^j - \tau\right) \tag{2}$$

ESTIMATING: We determine the next target state $\mathbf{p}_t$ typically by selecting the best $\mathbf{p}_t^j$ that corresponds to the maximum score s_t^j, $\mathbf{p}_t = \mathbf{p}_{t-1} \circ \mathbf{y}_t^{j^*}$ s.t. $j^* = \text{argmax}_{j \in \{1, \dots, N\}} s_t^j$.

UPDATING: Finally, we update the classifier by its own labeled data:

$$\theta_{t+1} = u(\theta_t, \mathcal{X}_t, \mathcal{L}_t) \tag{3}$$

in which $u(l)$ is the update function (e.g., budgeted SVM update [15]) and $\mathcal{X}_t$, $\mathcal{L}_t$ are the set of input patches and output labels used as the training set of the discriminator.

2.2. Baseline system implementation

To develop a baseline tracking-by-detection algorithm for this study, we use a robust part-based detector for the **CLASSIFYING** process. This detector employs strong low-level features based on histograms of oriented gradients (HOG) and uses a latent SVM to perform efficient matching for deformable part-based models (pictorial structures) [64]. From each frame, we draw N samples from a Gaussian distribution whose mean is the target's bounding box in the last frame (including its location and size). The selected detector then outputs the classification score for each sample, which is thresholded to obtain the sample's label. The highest classification score is considered as the current target location (**Figure 1**).

In the first frame, we generate $\alpha_1 N$-positive samples by perturbing the first annotated target patch by few pixels in location and size, select $\alpha_2 N$-negative samples from local neighborhood of the target, and select $\alpha_3 N$-negative samples from global background of the object in a regular grid ($\alpha_1 + \alpha_2 + \alpha_3 = 1$). These samples are used to train the SVM detector in the first frame. From the next frames, the labels are obtained by the detector itself, and the classifier is batch-trained with all of the samples collected so far.

There are several parameters in the system such as the parameters of sampling step (number of samples N, effective search radius Σ_{search}). These parameters were tuned using a simulated annealing optimization on a cross validation set. The part-base detector dictionary, and the thresholds τ_l, τ_u, and the rest of abovementioned parameters have been adjusted using cross validation. With $N = 1000$, $\tau = 0.34$ T1 achieved the speed of 47.29 fps on a Pentium IV PC @ 3.5 GHz and a Matlab/C++ implementation on a CPU.

2.3. Method of evaluation

The experiments are conducted on 100 challenging video sequences, OTB-100 [65], which involves many visual tracking challenges such as target appearance, pose and geometry changes, environment lighting and camera position changes, target movement artifacts such

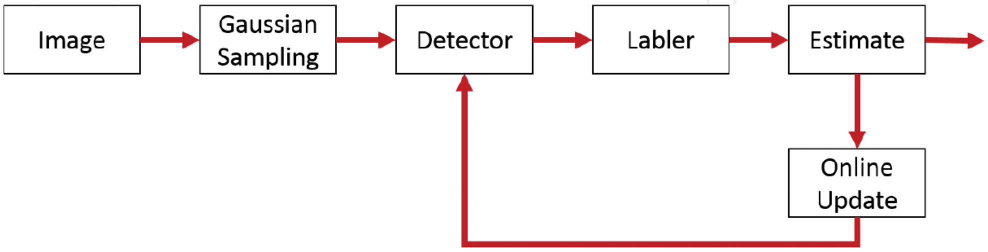

Figure 1. A simple tracking-by-detection pipeline. After gathering some samples from the current frame, the tracker employs its detector to label the samples as positive (target) or negative (background). The target position is estimated using these labeled samples. The labels, in turn, are used to update the classifier for the next frame.

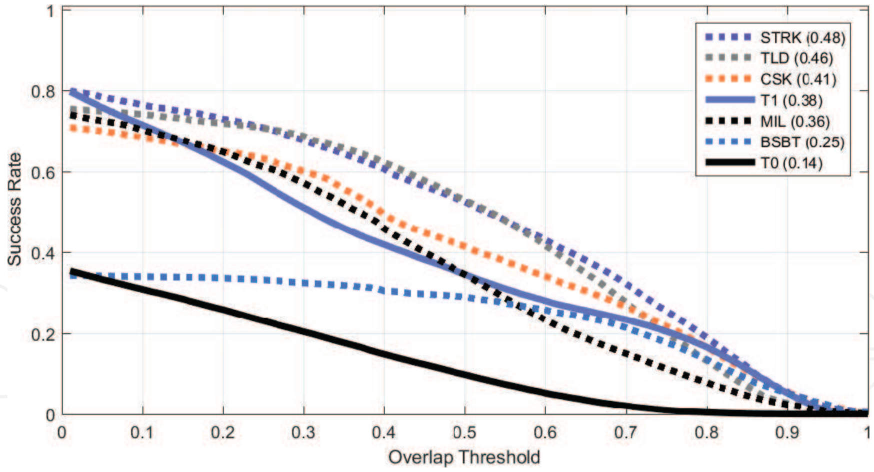

Figure 2. Quantitative performance comparison of the baseline tracker (T1), its variant without model update (T0), and the state-of-the-art trackers using success plot.

as blur and trajectory variations, and low imaging resolution and noise and background objects which may cause occlusions, clutter, or target identity confusion. The performance of the trackers is compared with the area under the curve of success plots and precision plots, on all of the sequences, or a subset of them with the given attribute.

Success plot indicates the reliability of the tracker and its overall performance, while precision plot reflects the accuracy of the localization. The area under the surface of this plot (AUC) counts the number of successes of tracker over time $t \in \{1, \dots, T\}$, i.e., when the overlap of the tracker target estimation $\mathbf{p}_t$ with the ground truth $\mathbf{p}_t^*$ exceeds the threshold τ_{ov}. Success plot graphs the success of the tracker against different values of the threshold τ_{ov}, and its AUC is calculated as

$$AUC = \frac{1}{T} \int_0^1 \sum_{t=1}^{T} 1\left(\frac{|\mathbf{p}_t \cap \mathbf{p}_t^*|}{|\mathbf{p}_t \cup \mathbf{p}_t^*|} > \tau_{ov} \right) d_{\tau_{ov}}, \tag{4}$$

where T is the length of sequence; |.| denotes the area of the region; ∩ and ∪ stand for intersection and union of the regions, respectively; and $1(.)$ denotes the step function that returns 1 iff its argument is positive and 0 otherwise. This plot provides an overall performance of the tracker, reflecting target loss, scale mismatches, and localization accuracy.

To establish a fair comparison with the state of the art of tracking-by-detection algorithms, TLD [58] and STRUCK [15] are selected based on the results of [27], BSBT [66] and MIL [47] are selected based on popularity, and CSK [36] was selected as one of the latest algorithms in the category. Since our trackers contain random elements (in sampling and resampling), the results reported here are the average of five independent runs.

2.4. Results

Figure 2 presents the success and precision plots of T1 along with other competitive trackers for all sequences. We also included a fixed version of T1 tracker (a detector without model update) as T0 to emphasize the role of updating. The figure demonstrates that without the model update, the detector cannot reflect the changes in target appearance and lose the target rapidly in most of the scenarios (comparing T0 and T1). However, it is also evident that having a single tracker is not robust against all of the target's variations (in line with [60]) and the performance of T1 is still low.

3. Co-tracking

A single detector may have difficulties in distinguishing the target from the background in certain scenarios. In those cases, it is beneficial to consult another detector with higher robustness. These second detector may have complimentary characteristics to the first one or simply may be a more sophisticated detector that trades computational complexity with speed.

Collaborative discriminative trackers utilize classifiers that exchange their information, to achieve more robust tracking. These information exchanges are in the form of queries that one classifier sends to another. The purpose of this information exchange is to bridge across long-term and short-term memories [62]; accommodate multi-memory dictionaries [67], mixture of deep and shallow models [68]; facilitate multi-view on the data [34]; and enable learning from mistakes [58].

3.1. Formalization

Built on co-training principle [69], collaborative tracking (co-tracking) provides a framework in which two classifiers exchange their information to promote tracking results and break self-learning loop (**Figure 3**). In this two-classifier framework [34], the challenging samples for one classifier are labeled by the other one, i.e., if a classifier finds a sample difficult to label, it relies on the other classifier to label it for this frame and similar samples in the future. In this case, we calculate the discrimination score s_t^j as a weighted sum of the two discriminant functions, $s_t^j = \sum_{c=1}^{2} \alpha_t^{(c)} h\left(\mathbf{x}_t^j | \theta_t^{(c)}\right)$ where $\alpha_t^{(c)}$ denotes the weight of each discriminator $\theta_t^{(c)}$, $c = 1, 2$. At the **CLASSIFYING** step, the corresponding sample $\mathbf{x}_t^j$ is considered as a challenging sample for the cth discriminator when $\tau_l < h\left(\mathbf{x}_t^j | \theta_t^{(c)}\right) < \tau_u$ holds because it locates close to the corresponding discrimination boundary. When one of the two discriminators answered it challenging, the score of the sample is calculated with using the other score:

$$s_t^j = \begin{cases} \alpha_t^{(2)} h\left(\mathbf{x}_t^j | \theta_t^{(2)}\right) & , h\left(\mathbf{x}_t^j | \theta_t^{(1)}\right) \in (\tau_l, \tau_u) \text{ and } h\left(\mathbf{x}_t^j | \theta_t^{(2)}\right) \notin (\tau_l, \tau_u) \\ \alpha_t^{(1)} h\left(\mathbf{x}_t^j | \theta_t^{(1)}\right) & , h\left(\mathbf{x}_t^j | \theta_t^{(2)}\right) \in (\tau_l, \tau_u) \text{ and } h\left(\mathbf{x}_t^j | \theta_t^{(1)}\right) \notin (\tau_l, \tau_u) \\ \sum_{c=1}^{2} \alpha_t^{(c)} h\left(\mathbf{x}_t^j | \theta_t^{(c)}\right) & , \text{otherwise} \end{cases} \tag{5}$$

At the **UPDATING** step, the weight $\alpha_t^{(c)}$ of the discriminator c is adjusted according to the degree of contradiction to the provisional answers that are determined at the **ESTIMATION**

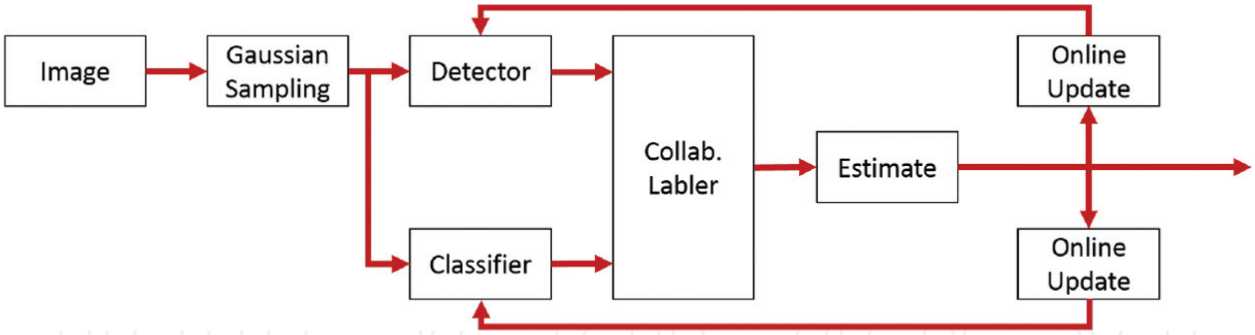

Figure 3. Collaborative tracking. A detector and an auxiliary classifier trust each other to handle the sample difficult for them to classify.

step by an integration of all the information. Finally, the classifiers are updated using only the samples that they successfully labeled in the previous frame to reflect the latest target changes.

3.2. Evaluation

For this experiment, we selected a naive classifier with complementary properties to the main classifier in the previous section. This classifier is a KNN classifier using HOC and HOG features, trained on the samples trained from the first frame and updated with all the labeled samples by the collaboration of the classifiers. Not being pre-trained, the performance of this auxiliary classifier is poor in the beginning but gradually gets better. The quick classification of the KNN (owning to its kd-tree implementations and lightweight features) and lack of pre-training grant it high speed and generalization which is in contrast to the main detector. However, it should be noted that without being supervised by the main SVM-based detector, this classifier cannot perform well in isolation for tracking task. **Figure 5** presents the performance of this auxiliary tracker as T2. As observed in the figure, the performance of the obtained co-tracker (T3) is better than the main detector (T1) and the auxiliary classifier (T2) as a result of co-labeling, data exchange, and co-learning.

4. Active co-tracking

The co-tracking framework provides a means for classifiers to exchange information. This framework utilizes a utility measure (e.g., the classification confidence in [34]) to select the data for which one of the collaborators fails to classify with high confidence and then trains the other classifier on those data. This approach has two main shortcomings: (1) the redundant labeling of all samples for both classifiers and (2) training the collaborator with “all” of the uncertain samples. While the former increases the complexity of the system, the latter is not the optimal solution for tracking a target with non-stationary appearance distributions [35].

In this view, a principled ordering of samples for training [70] and selecting a subset of them based on criteria [37] can reduce the cost of labeling leading to faster performance increase as a function of the amount of data available. It is found that detectors trained with an effective, noise-free, and outlier-free subset of the training data may achieve higher performance than those trained with the full set [71, 72].

Robust learning algorithms provide an alternative way of differentially treating training examples, by assigning different weights to different training examples or by learning to ignore outliers [73]. Learning first from easy examples [74], pruning adversarial examples[1] [75], and sorting the samples based on their training value [37] are some of the approaches explored in the literature. However, the most common setting is active learning, whereby most of the data is unlabeled and an algorithm selects which training examples to label at each step, for the highest gains in performance. Thus, some active learning approaches focus on learning the hardest examples first (those closest to the decision boundary). Some approaches focus on learning the hardest examples first (e.g., those closest to the decision boundary), whereas some others gauge the information contained in the sample and select the most informative ones first. For example, Lewis and Gale [76] utilized the uncertainty of the classifier for a sample as an index of its usefulness for training.

4.1. The idea

Active learning has been used in visual tracking to consider the uncertainty caused by bags of samples [55], to reduce the number of necessary labeled samples [77], to unify sample learning and feature selection procedure [78], and to reduce the sampling bias by controlling the variance [79].

In this study, we utilized the sampling uncertainty that can bind the active learning and co-tracking. As mentioned earlier, the baseline classifier, despite being accurate, has low generalization on new samples, slow classification speed, and computationally expensive retraining. On the other hand, the auxiliary classifier is agile and learns rapidly, with negligible retraining time. To combine the merits of these two classifiers, to cancel out their demerits with one another, and to address the aforementioned issues of co-tracking (redundant labeling and excessive samples), we incorporate an active learning module to select the most informative data, i.e., those for which the naive classifier is most uncertain, and query their labels from the part-based detector. This architecture (**Figure 4**, here called T4) mainly uses naive classifier for labeling the data and only asks the label of hard samples from the slower detector and, therefore, limits the redundancy and unleashes the speed of the agile classifier. In addition, by training the naive classifier only on hard samples, the generalization of this classifier is preserved while increasing its accuracy.

To further increase the accuracy of the tracker and make it more robust against occlusions and drastic temporal changes of the target, it is possible to update the detector less frequently. This asymmetric version of the active co-tracker (T5), by introducing long-term memory to the tracker, benefits from combining the long- and short-term collaboration (as in [62]) and reduces the frequency of the expensive updates of the tracker (Algorithm 1).

[1] Images with tiny, imperceptible perturbations that fool a classifier into predicting the wrong labels with high confidence

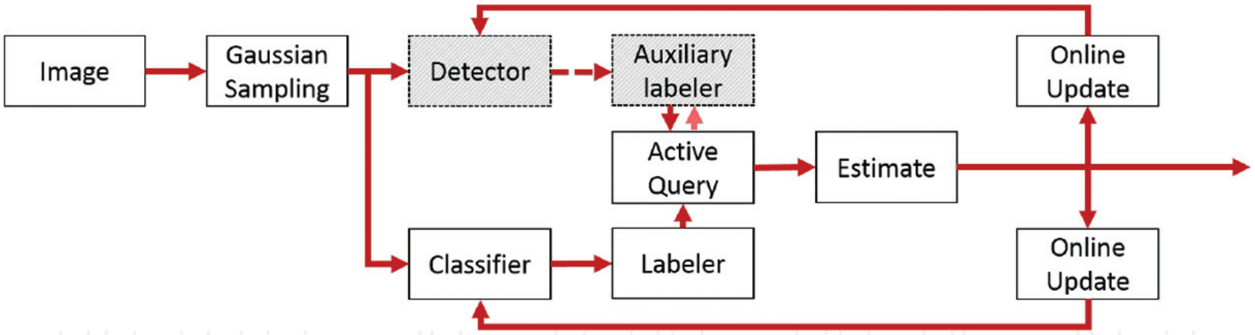

Figure 4. Active co-tracker, a collaborative tracker that utilizes an active query mechanism to query the most informative samples from the main detector and feeds them to the lightweight classifier to learn.

```
Algorithm 1: Active co-tracking (ACT)
Input: Target position in last frame $\mathbf{p}_{t-1}$
Output: Target position in current frame $\mathbf{p}_t$
for $j \leftarrow 1$ to $n$ do
    Generate a sample $\mathbf{p}_t^j \sim \mathcal{N}(\mathbf{p}_{t-1}, \Sigma_{search})$
    Calculate $s_t^j \leftarrow h\left(\mathbf{x}_t^{\mathbf{p}_t^j} | \theta_t^{(1)}\right)$ (Eq.(6))
Determine uncertain samples $\mathcal{U}_t$ (Eq.(7))
if $\mathbf{p}_t^j \in \mathcal{U}_t$ then $\theta_t^{(1)}$ is uncertain
    Query $\theta_t^{(2)}: l_t^j \leftarrow Sign\left(h\left(\mathbf{x}_t^{\mathbf{p}_t^j} | \theta_t^{(2)}\right)\right)$
else
    Label using $\theta_t^{(1)}: l_t^j \leftarrow Sign\left(s_t^j\right)$
$\mathcal{D}_t \leftarrow \mathcal{D}_t \cup \left\langle \mathbf{x}_t^{\mathbf{p}_t^j}, l_t^j \right\rangle$
Update $\theta_t^{(2)}$ with $\mathcal{D}_{t-\Delta, .., t}$ every $\Delta$ frames ($\Delta = 1$ for T4)
if $\sum_{j=1}^{n} 1\left(l_t^j > 0\right) > \tau_p$ and $\sum_{j=1}^{n} \pi_t^j > \tau_a$ then
    Approximate target state $\widehat{\mathbf{p}}_t$ (Eq.(9))
    Update $\theta_t^{(1)}$ with $\mathcal{U}_t$
else target occluded
    $\widehat{\mathbf{p}}_t \leftarrow \mathbf{p}_{t-1}$
```

4.2. Formalization

In the proposed active co-tracking framework, a main classifier attempts to label the sample, and it queries the label from the other classifier if the main classifier emits uncertain results. This is in contrast with using a linear combination of both classifiers based on their classification accuracy as adopted in T3. At the **CLASSIFYING** step, the proposed tracker can score each sample based on the classifier confidence, i.e., for sample $\mathbf{p}_t^j$ we calculate score s_t^j:

$$s_t^j = h\left(\mathbf{x}_t^{\mathbf{p}_t^j} | \theta_t^{(1)}\right). \tag{6}$$

Based on uncertainty sampling [76], the samples for which the classification score is more uncertain (i.e., $s_t^j \to 0$) contain more information for the classifier if they are labeled by the other classifier. Therefore, the scores of all samples are sorted, and m samples with the closest values to 0 are selected to be queried from $\theta_t^{(2)}$. To handle the situations for which the number of highly uncertain samples are more than m, a range of scores are determined by lower and higher thresholds (τ_l and τ_u), and all the samples in this range are considered highly uncertain:

$$\mathcal{U}_t = \left\{\mathbf{p}_t^i | \tau_l < s_t^i < \tau_u \quad \text{or} \quad |\left\{\exists j \neq i | s_t^j \leq s_t^i\right\}| < m\right\} \tag{7}$$

in which $\mathcal{U}_t$ is the list of uncertain samples. The label of the samples $l_t^j \in \mathcal{L}_t, j = 1, \ldots, N$ is then determined by

$$l_t^j = \begin{cases} \text{sign}\left(h\left(\mathbf{x}_t^{\mathbf{p}_t^j} | \theta_t^{(1)}\right)\right) & , \mathbf{p}_t^j \in \mathcal{U}_t \\ \text{sign}\left(h\left(\mathbf{x}_t^{\mathbf{p}_t^j} | \theta_t^{(2)}\right)\right) & , \mathbf{p}_t^j \notin \mathcal{U}_t \end{cases} \tag{8}$$

and all image patches $\mathbf{x}_t^{\mathbf{p}_t^j}$ and labels l_t^j are stored in $\mathcal{D}_t$.

At the **ESTIMATION** step, we follow the importance sampling mechanism originally employed by particle filter trackers:

$$\widehat{\mathbf{p}}_t = \frac{\sum_{j=1}^{n} \pi_t^j \mathbf{p}_t^j}{\sum_{j=1}^{1} \pi_t^j}. \tag{9}$$

where $\pi_t^j = s_t^j 1\left(l_t^j > 0\right)$ and $1(.)$ are the indicator function, 1 if true, zero otherwise. This mechanism approximates the state of the target, based on the effect of positive samples, in which samples with higher scores gravitate the final results more toward themselves. Upon the events such as massive occlusion or target loss, this sampling mechanism degenerates [13]. In such cases, the number of positive samples and their corresponding weights shrinks significantly, and the importance sampling is prone to outliers, distractors, and occluded patches. To

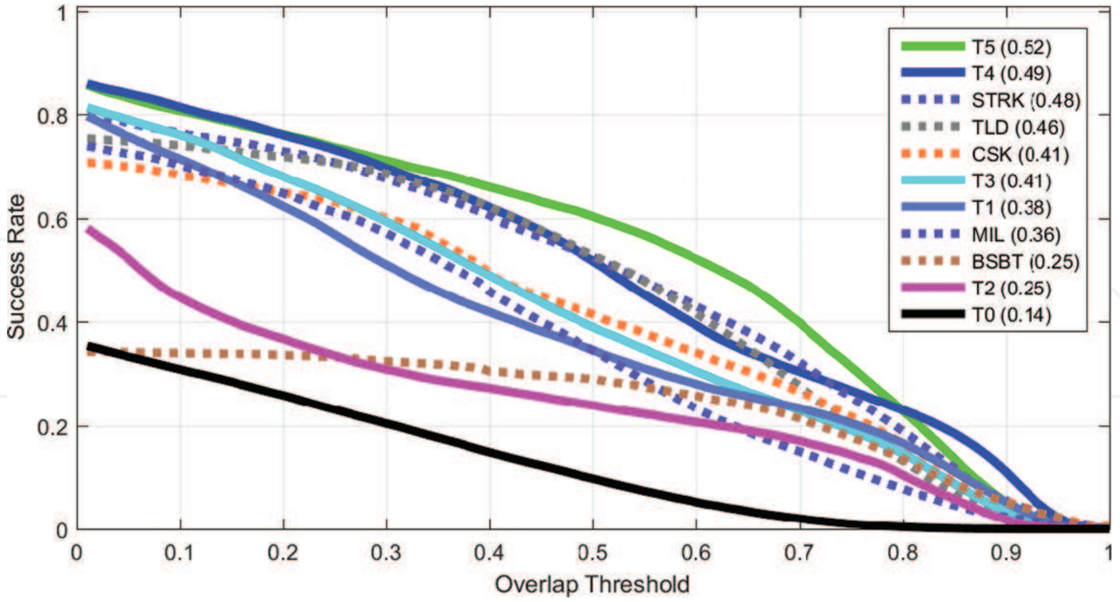

Figure 5. Quantitative performance comparison of the asymmetric active co-tracker (T5), active co-tracker (T4), the ordinary co-tracker (T3), and their individual trackers (T1 and T2).

address this issue, if the number of positive samples is less than τ_p, and their score average is less than τ_a, the target is deemed occluded to avoid tracker degeneracy.

4.3. Evaluation

Figure 5 illustrates the effectiveness of the proposed trackers against their baselines. The active query mechanism in T4 improves the efficiency and effectiveness of co-tracking (T3). Especially in the asymmetric co-tracker (T5), the mixture of long-term and short-term memory classifiers using this method is to key to automatically balance the stability-plasticity equilibrium. It is also prudent for the tracker to adapt to the temporal distribution of the target appearance, before its redistribution by illumination changes, etc.

In summary, the advantages of the proposed trackers especially the asymmetric ones (T5) compared to the conventional co-tracking (T3) are as follows: (1) the classifiers do not exchange all the data they have problems in labeling; instead, the most informative samples are selected by uncertainty sampling and exchanged; (2) the update rate of classifiers is different to realize a short- and long-term memory mixture; (3) the samples that are labeled for the target localization can be reused for training, and the need for an extra round of sampling and labeling is revoked; and (4) since in the proposed asymmetric co-tracking, one of the classifiers scaffolds the other one instead of participating in every labeling process, a more sophisticated classifier with higher computational complexity can be used.

5. Active ensemble co-tracking

Ensemble discriminative tracking utilizes a committee of classifiers, to label data samples, which are in turn used for retraining the tracker to localize the target using the collective knowledge of the committee. In such frameworks the labeling process is performed by leveraging a group of classifiers with different views [45, 56, 80], subsets of training data [57, 81], or memories [57, 82].

In ensemble tracking [45, 47, 56, 57, 60, 83–85], the self-learning loop is broken, and the labeling process is performed by eliciting the belief of a group of classifiers. However, this framework typically does not address some of the demands of tracking-by-detection approaches like a proper model update to avoid model drift or non-stationary of the target sample distribution. Besides, ensemble classifiers do not exchange information, and collaborative classifiers entirely trust the other classifier to label the challenging samples for them and are susceptible to label noise.

Traditionally, ensemble trackers were used to providing a multi-view classification of the target, realized by using different features to construct weak classifiers. In this view, different classifiers represent different hypotheses in the version space, to accurately model the target appearance. Such hypotheses are highly overlapping; therefore an ensemble of them overfits the target. The desired committee, however, consists of competing hypotheses, all consistent with the training data, but each of the specialized in certain aspect. In this view, the most informative data samples are those about which the hypotheses disagree the most, and by labeling them, the version space is minimized leading to quick convergence yet accurate classification [86]. Motivated by this, we proposed a tracker that employs a randomized ensemble of classifiers and selects the most informative data samples to be labeled.

5.1. The idea

To create ensembles of classifiers, researchers typically make different classifiers by altering the features [45], using a pool of appearance and dynamics models [87], utilizing different memory horizons [82], and employing previous snapshots of a classifier in different times [57], but creating a collaborative mechanism in the ensemble, where classifiers exchange information is hardly addressed in the visual tracking literature. This data exchange can be in the form of query passing between ensemble members, in which the queries can be the samples for which a classifier is uncertain or even the ensemble is most uncertain.

Selecting such queries is addressed in different machine learning domains such as curriculum learning [74] and active learning. Query-by-Committee (QBC) algorithm [86, 88] is an active learning approach for ensembles that selects the most informative query to pass within a committee of models which are all trained on the current labeled set but represent competing hypotheses. The label of the queried sample is then decided by the vote of the ensemble members, and the samples for which the ensemble has more diverse ideas are selected as the next query to ask from the teacher (here, the auxiliary classifier). In this case, where the task is a binary classification, the most disputed sample (i.e., with close positive and negative votes) is the most informative since learning its label would maximally train the ensemble. Training with the external label for this sample, shrinks the version space (i.e., the space of all consistent hypotheses with the training data) such that it remains consistent with the hypotheses of all classifiers, but rejects more potential incorrect ones.

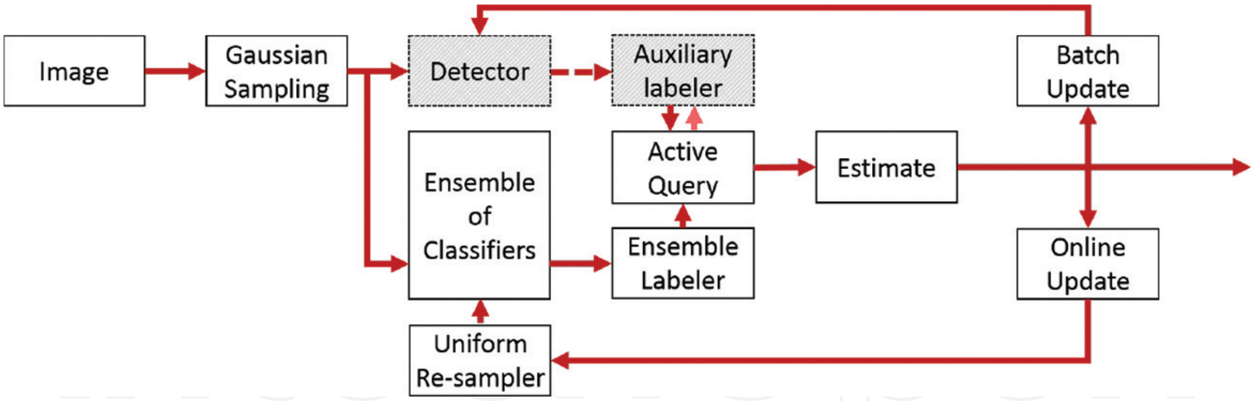

Figure 6. Active ensemble co-tracker. The bagging-induced ensemble labels the input samples and only queries the most disputed ones from the slow part-based classifier.

QBC was originally designed to work with stochastic learning algorithms, which pose limitations to use it with non-probabilistic or deterministic models. To alleviate this problem, Abe and Mamitsuka [89] enable deterministic classifiers to work with random subsets of training data to create different variations of the same learning model. By creating temporary ensemble using this "bagging" procedure [90], they realized Query-by-Bagging (QBag) to enhance the learning speed and generalization of the base learning algorithm.

We propose the adjustment of the QBag algorithm for online training to solve the label noise problem in T6. Similar to T5, the drift problem is handled using dual-memory strategy: the committee rapidly adapts to target changes, whereas the main classifier possesses a longer memory to promote the stability of the target template (**Figure 6**).

5.2. Formalization

An ensemble discriminative tracker employs a set of classifiers instead of one. These classifiers, hereafter called *committee*, are represented by $\mathcal{C} = \left\{\theta_t^{(1)}, \dots, \theta_t^{(C)}\right\}$ and are typically homogeneous and independent (e.g., [56, 85]). Popular ensemble trackers utilize the majority voting of the committee as their utility function:

$$\mathbf{s}_t^j = \sum_{c=1}^{C} \text{sign}\left(h\left(\mathbf{x}_t^{\mathbf{p}_{t-1}\circ\mathbf{y}_t^j} | \theta_t^{(c)}\right)\right). \tag{10}$$

And Eq. (8) is used to label the samples. Finally, the model is updated for each classifier independently, meaning that each of the committee members is trained with a random subset of the uncertain set. $\theta_{t+1}^{(c)} = u\left(\theta_t^{(c)}, \Gamma_t^{(c)} \sim \mathcal{U}_t\right)$ where $u(\theta, \mathcal{X})$ is the updating the model θ with samples $\mathcal{X}$. The uncertain set $\mathcal{U}_t$ contains all of the samples for which the ensemble disagrees and was sent to the auxiliary classifier for labeling. The detector $\theta_t^{(o)}$ is also updated with all recent data $\mathcal{D}_{t-\Delta,..,t}$ every Δ frames.

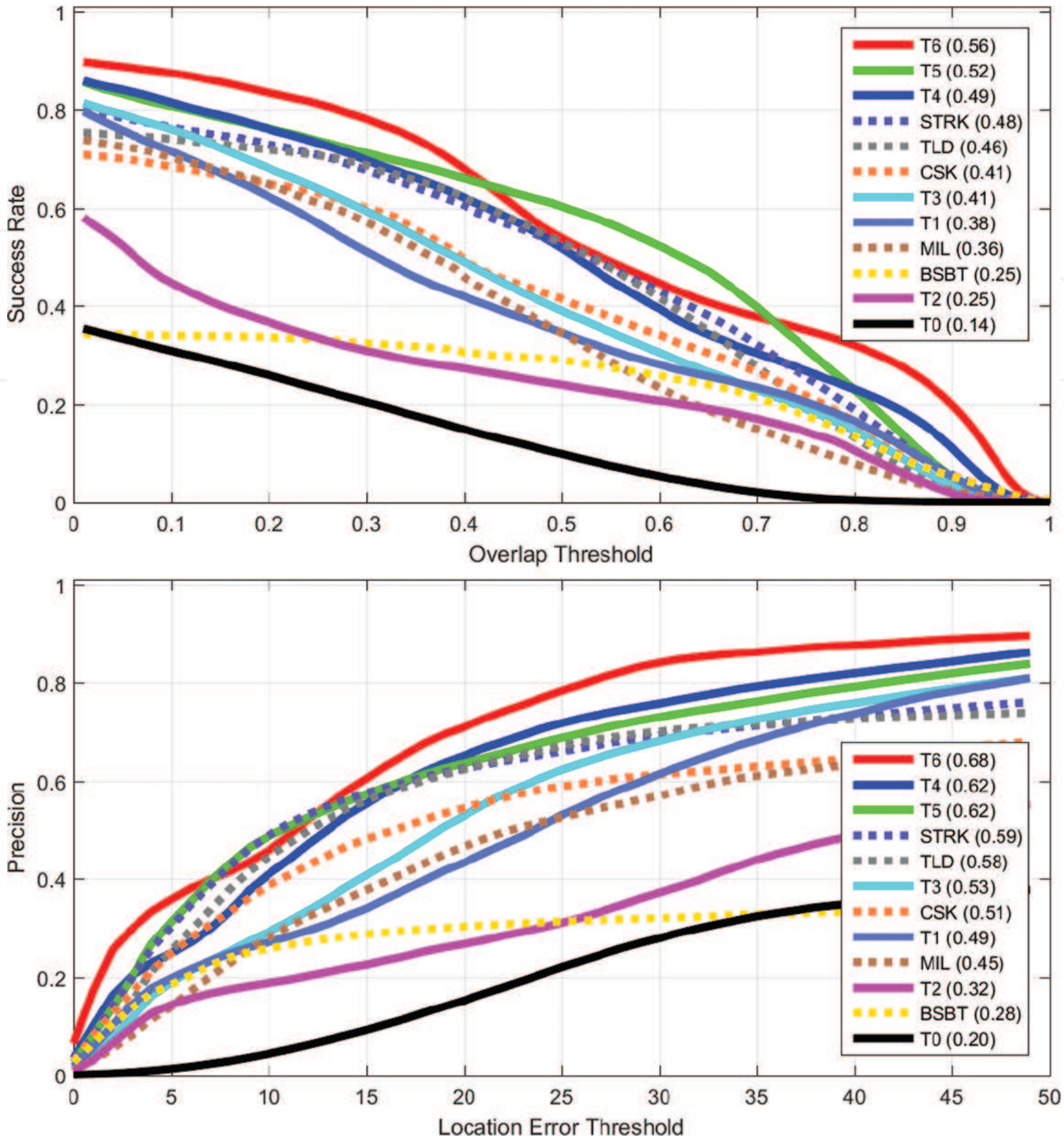

Figure 7. Quantitative performance comparison of the active ensemble co-tracker (T6) with its predecessors.

5.3. Evaluation

Figure 7 depicts the overall performance of the proposed tracker against other benchmarked algorithms on all sequences of the dataset. The plots show that T6 has a superior performance over T5 and its predecessors. The steep slope between $0.9 \geq \tau_{ov} > 1$ indicates the high quality of the predictions (i.e., more predictions have higher overlap with the ground truth, rather than being partially correct), and the other slope around $\tau_{ov} \approx 0.4$ along with high success rate near $\tau_{ov} \rightarrow 0$ indicates that the algorithm was successful in continue tracking, despite all the tracking challenges.

6. Discussion

The instances of the proposed framework are evaluated against state-of-the-art trackers on public sequences that become the de facto standards of benchmarking the trackers. The trackers are compared with popular metrics such as success plot and precision plot to establish a fair benchmark. In addition, the performance of the proposed trackers is investigated for videos with a distinguished tracking challenge, and the results are compared with state of the art and discussed. Additionally, the effect of the information exchanged will be examined thoroughly to illustrate the dynamics of the system. The preliminary results of the proposed framework demonstrate a superior performance for the proposed trackers when applied on all the sequences and most of the subsets of the test dataset with distinguished challenges. Finally, the future research direction is discussed, and the opened research avenues are introduced to the field.

As **Figure 7** and **Table 2** demonstrate, T6 has the best overall performance among investigated trackers on this dataset. While this algorithm has a clear edge in handling many challenges, its performance is comparable with T5 in the case of occlusions and z-rotations. It is also evident that T6 is troubled with fast deformations since neither of the ensemble members is specialized in handling a specific type of deformations and the collective decision of the ensemble may involve mistakes with high confidence. On the other hand, T5 utilizes a dual-memory scheme, and a single classifier can handle extreme temporal deformations better than the ensemble in

	IV	DEF	OCC	SV	IPR	OPR	OV	LR	BC	FM	MB	ALL
T0	12	12	13	12	13	13	14	5	12	15	18	14
T1	37	29	3	36	42	39	43	30	33	39	36	38
T2	23	19	23	23	28	25	25	22	23	24	20	25
T3	41	32	39	40	44	42	43	30	36	43	39	41
T4	50	39	47	48	53	49	48	37	44	50	45	49
T5	52	47	53	51	59	56	52	38	41	53	46	52
T6	57	40	51	53	61	55	63	46	53	60	58	56
TLD	49	32	42	44	50	43	45	37	40	45	42	46
STRK	46	41	44	43	51	48	44	39	39	52	48	48
CSK	40	36	36	34	43	39	32	29	42	39	32	41
MIL	35	35	38	35	41	39	40	32	31	35	28	36
BSBT	23	18	23	21	27	24	32	23	23	26	24	25

The first, second, and third best methods are shown in color. The challenges are illumination variation (IV), scale variation (SV), occlusions (OCC), deformations (DEF), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-play rotation (OPR), out-of-view problem (OV), background clutter (BC), and low resolution (LR)

Table 2. Quantitative evaluation of state of the art under different visual tracking challenges using AUC of success plot (%).

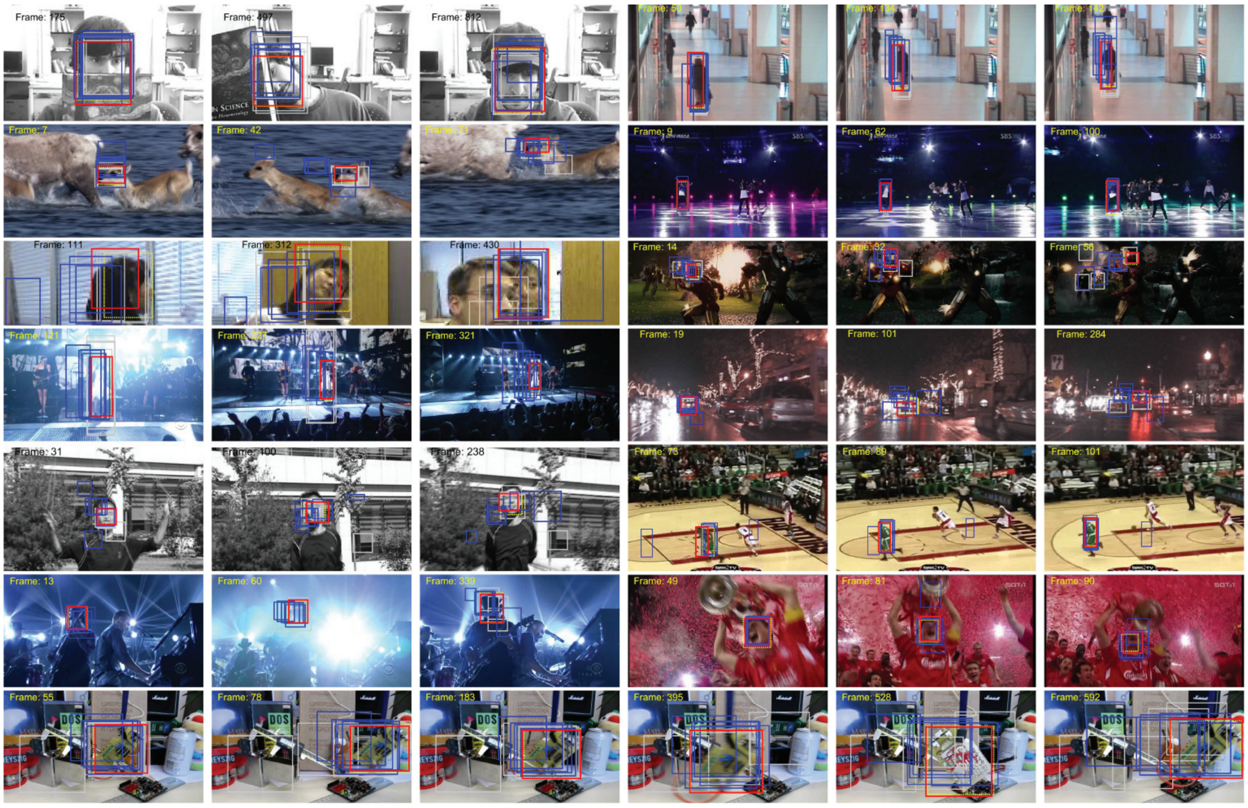

Figure 8. Qualitative results of T6 in red against other trackers (T0–T5 in blue and TLD, STRK, CSK, MIL, and BSBT in gray) on challenging video scenarios of OTB-100 [65]. The sequences are (from top to bottom, left to right) FaceOcc2 and Walking2 with severe occlusion, Deer and Skating1 with abrupt motions, Firl and Ironman with drastic rotations, Singer1 and CarDark and Shaking with poor lighting, Jumping and Basketball with nonrigid deformations, and Shaking,Soccer with drastic lighting, pose, and noise level changes and Board with intensive background clutter. The ground truth is illustrated with yellow dashed box. The results are available in http://ishiilab.jp/member/meshgi-k/act.html.

T6. Interestingly, it is observed that in most of the subcategories that T6 is clearly better than the other trackers, the success plot of T6 starts with a plateau and later has a sharp drop around $\tau_{ov} = 0.8$. This means that T6 provides high-quality localization (i.e., bigger overlaps with the ground truth). Similarly, from precision plot, it is evident that T6 shows a graceful degradation in different scenarios, and although it does not provide a good scale adaptation for targets, it is able to localize them better than the competing trackers (**Figure 8**).

7. Conclusions and future works

This chapter provides a step-by-step tutorial for creating an accurate and high-performance tracking-by-detection algorithm out of ordinary detectors, by eliciting an effective collaboration among them. The use of active learning in junction with co-learning enables the creation of a battery of tracker that strives to minimize the uncertainty of one classifier by the help of another. The progressive design leads to use a committee of classifiers that use online bagging to keep up with the latest target appearance changes while improving the accuracy and generalization of the base tracker (a feature-based KNN). Inspired by the query-by-bagging algorithm, this algorithm selects the most informative samples to learn from the long-term memory auxiliary detector, which realizes a gradually decreasing dependence on this slow and likely overfit detector yet robust against fluctuations in target appearance and occlusions. Furthermore, using an expectation of the bounding boxes compensates for overreliance of the tracker on the classifiers' confidence function. The balance in stability-plasticity equilibrium is achieved by the combination of several short-term classifiers with a long-term classifier and managing their interaction with an active learning mechanism.

The trail of proposed trackers led to T6, which incorporates ensemble tracking, active learning, and co-learning in a discriminative tracking framework and outperform state-of-the-art discriminative and generative trackers on a large video dataset with various types of challenges such as appearance changes and occlusions.

The future direction of this study involves other detectors to care for context, to have accurate physical models for known categories, to use deep features to improve discrimination, and to examine different methods of building the ensemble and detecting most informative samples or exchanging.

Acknowledgements

This article is based on results obtained from a project commissioned by the Japan NEDO and was supported by Post-K application development for exploratory challenges from the Japan MEXT.

Author details

Kourosh Meshgi* and Shigeyuki Oba

*Address all correspondence to: meshgi-k@sys.i.kyoto-u.ac.jp

Graduate School of Informatics, Kyoto University, Kyoto, Japan

References

[1] Borangiu T. "Visual conveyor tracking in high-speed robotics tasks," in Industrial Robotics: Theory, Modelling and Control. InTech, Rijeka, Croatia 2006

[2] Cech J, Mittal R, Deleforge A, Horaud R. Active-speaker detection and localization with mic and cameras embedded into a robotic head. In: Humanoids'13; 2013

[3] Cosgun A, Florencio DA, Christensen HI. Autonomous person following for telepresence robots. In: ICRA'13; IEEE; 2013. pp. 4335-4342

[4] Andersen NA, Andersen JC, Bayramoglu E, Ravn O. Visual navigation for mobile robots. In: Robot Vision. Rijeka, Croatia: InTech; 2010

[5] Campoy P, Mondragón IF, Olivares-Méndez MA, Martínez C. Visual servoing for UAVs. In: Visual Servoing. Rijeka, Croatia: InTech; 2010

[6] Moeslund TB, Hilton A, Krüger V. A survey of advances in vision-based human motion capture and analysis. CVIU. 2006;**104**(2):90-126

[7] Störring M, Moeslund TB, Liu Y, Granum E. Computer vision-based gesture recognition for an augmented reality interface. In: VIIP'04. Vol. 3; 2004. pp. 766-771

[8] Koller O, Zargaran O, Ney H, Bowden R. Deep sign: Hybrid CNN-HMM for continuous sign language recognition. In: BMVC'16; 2016

[9] Wang L, Schmidt B, Nee AY. Vision-guided active collision avoidance for human-robot collaborations. Manufacturing Letters. 2013;**1**(1):5-8

[10] Ess A, Leibe B, Schindler K, Van Gool L. Moving obstacle detection in highly dynamic scenes. In: ICRA'09; IEEE; 2009. pp. 56-63

[11] Xia L, Gori I, Aggarwal JK, Ryoo MS. Robot-centric activity recognition from first-person RGB-D videos. In: WACV'15; IEEE; 2015. pp. 357-364

[12] Rautaray SS, Agrawal A. Vision based hand gesture recognition for human computer interaction: A survey. AI Review. 2015;**43**(1):1-54

[13] Bao C, Wu Y, Ling H, Ji H. Real time robust l1 tracker using accelerated proximal gradient approach. In: CVPR'12; 2012

[14] Kwon J, Lee KM. Tracking by sampling trackers. In: ICCV'11; IEEE; 2011. pp. 1195-1202

[15] Hare S, Saffari A, Torr PH. Struck: Structured output tracking with kernels. In: ICCV'11; 2011

[16] Hilsmann A, Schneider DC, Eisert P. Image-based tracking of deformable surfaces. In: Object Tracking. Rijeka, Croatia: InTech; 2011

[17] Dinh TB, Vo N, Medioni G. Context tracker: Exploring supporters and distracters in unconstrained environments. In: CVPR'11; 2011

[18] Meshgi K, Maeda S-I, Oba S, Ishii S. Data-driven probabilistic occlusion mask to promote visual tracking. In: CRV'16; IEEE; 2016. pp. 178-185

[19] Wu Y, Ling H, Yu J, Li F, Mei X, Cheng E. Blurred target tracking by blur-driven tracker. In: ICCV'2011; 2011

[20] Ross DA, Lim J, Lin R-S, Yang M-H. Incremental Learning for Robust Visual Tracking. International Journal of Computer Vision. Springer; 2008;**77**(1-3):125-141

[21] Fang J, Xu H, Wang Q, Wu T. Online Hash Tracking with Spatio-Temporal Saliency Auxiliary. Computer Vision and Image Understanding. Elsevier; 2017;**160**:57-72

[22] Taalimi A, Qi H, Khorsandi R. Online multi-modal task-driven dictionary learning and robust joint sparse representation for visual tracking. In: AVSS'15; 2015

[23] Kiani H, Sim T, Lucey S. Correlation filters with limited boundaries. In: CVPR'15; 2015

[24] Danelljan M, Hager G, Shahbaz Khan F, Felsberg M. Learning spatially regularized correlation filters for visual tracking. In: ICCV'15; 2015. pp. 4310-4318

[25] Danelljan M, Robinson A, Khan FS, Felsberg M. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In: ECCV'16; 2016

[26] Nam H, Han B. Learning multi-domain convolutional neural networks for visual tracking. In: CVPR'16; 2016

[27] Wu Y, Lim J, Yang M-H. Online object tracking: A benchmark. In: CVPR'13; IEEE; 2013. pp. 2411-2418

[28] Kristan M, Matas J, Leonardis A, Felsberg M. The visual object tracking vot2015 challenge results. In: ICCVw'15; 2015

[29] Li A, Lin M, Wu Y, Yang M-H, Yan S. NUS-PRO: A new visual tracking challenge. IEEE Transactions on Pattern Analysis and Machine Intelligence. IEEE; 2016;**38**(2):335-349

[30] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: CVPR'16; 2016. pp. 770-778

[31] Wang N, Yeung D-Y. Learning a deep compact image representation for visual tracking. In: NIPS'13; 2013. pp. 809-817

[32] Li H, Li Y, Porikli F, et al. Deeptrack: Learning discriminative feature representations by convolutional neural networks for visual tracking. In: BMVC. Vol. 2014; 2014

[33] Hong S, You T, Kwak S, Han B. Online tracking by learning discriminative saliency map with convolutional neural network. In: ICML'15; 2015. pp. 597-606

[34] Tang F, Brennan S, Zhao Q, Tao H. Co-tracking using semi-supervised support vector machines. In: ICCV'07; 2007

[35] Bai Q, Wu Z, Sclaroff S, Betke M, Monnier C. Randomized ensemble tracking. In: ICCV'13; 2013

[36] Henriques JF, Caseiro R, Martins P, Batista J. Exploiting the circulant structure of tracking-by-detection with kernels. In: ECCV'12; Springer; 2012. pp. 702-715

[37] Lapedriza A, Pirsiavash H, Bylinskii Z, Torralba A. Are all Training Examples Equally Valuable? arXiv. 2013

[38] Matthews I, Ishikawa T, Baker S. The template update problem. IEEE Transactions on Pattern Analysis and Machine Intelligence. IEEE; 2004;**26**(6):810-815

[39] Grabner H, Leistner C, Bischof H. Semi-supervised on-line boosting for robust tracking. In: ECCV'08; 2008

[40] Meshgi K, Mirzaei MS, Oba S, Ishii S. Efficient asymmetric co-tracking using uncertainty sampling. In: ICSIPA'17; 2017

[41] Meshgi K, Oba S, Ishii S. Robust discriminative tracking via query-by-committee. In: AVSS'16; 2016

[42] Pérez P, Hue C, Vermaak J, Gangnet M. Color-based probabilistic tracking. In: ECCV'02; 2002

[43] Wu Y, Pei M, Yang M, Jia Y. Robust Discriminative Tracking Via Landmark-Based Label Propagation. IEEE Transactions on Image Processing. IEEE; 2015;**24**(5):1510-1523

[44] Oza NC, Russell S. Online ensemble learning. In: AAAI'00, 2000

[45] Grabner H, Grabner M, Bischof H. Real-time tracking via on-line boosting. In: BMVC'06; 2006

[46] Leistner C, Saffari A, Roth P, Bischof H. On robustness of on-line boosting: a competitive study. In: ICCVw'09; 2009

[47] Babenko B, Yang M-H, Belongie S. Visual tracking with online multiple instance learning. In: CVPR'09; 2009

[48] Avidan S. Support vector tracking. PAMI. 2004;**26**(8):1064-1072

[49] Masnadi-Shirazi H, Mahadevan V, Vasconcelos N. On the design of robust classifiers for computer vision. In: CVPR'10; 2010

[50] Leistner C, Saffari A, Santner J, Bischof H. Semi-supervised random forests. In: ICCV'09; 2009

[51] Zeisl B, Leistner C, Saffari A, Bischof H. On-line semi-supervised multipleinstance boosting. In: CVPR'10; 2010

[52] Zhang K, Song H. Real-Time Visual Tracking via Online Weighted Multiple Instance Learning. Pattern Recognition. Elsevier; 2013;**46**(1):397-411

[53] Henriques JF, Caseiro R, Martins P, Batista J. High-speed tracking with kernelized correlation filters. PAMI. 2015;**37**(3):583-596

[54] Grabner H, Matas J, Van Gool L, Cattin P. Tracking the invisible: Learning where the object might be. In: CVPR'10; 2010

[55] Zhang K, Zhang L, Yang M-H, Hu Q. Robust Object Tracking via Active Feature Selection. IEEE Transactions on Circuits and Systems for Video Technology. IEEE; 2013;**23**(11): 1957-1967

[56] Saffari A, Leistner C, Santner J, Godec M, Bischof H. On-line random forests. In: ICCVw'09; 2009

[57] Zhang J, Ma S, Sclaroff S. MEEM: Robust tracking via multiple experts using entropy minimization. In: ECCV'14; 2014

[58] Kalal Z, Mikolajczyk K, Matas J. Tracking-learning-detection. PAMI. 2012;**34**(7):1409-1422

[59] Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR'14; 2014. pp. 580-587

[60] Avidan S. Ensemble tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence. IEEE; 2007;**29**(2):261-271

[61] Woodley T, Stenger B, Cipolla R. Tracking using online feature selection and a local generative model. In: BMVC'07; 2007

[62] Hong Z, Chen Z, Wang C, Prokhorov D, Tao D. Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking. In: CVPR'15; 2015

[63] Li J, Hong Z, Zhao B. Robust visual tracking by exploiting the historical tracker snapshots. In: ICCVW'15; 2015. pp. 41-49

[64] Felzenszwalb PF, Girshick RB, McAllester D, Ramanan D. Object detection with discriminatively trained part-based models. PAMI. 2010;**32**(9):1627-1645

[65] Wu Y, Lim J, Yang M-H. Object tracking benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence. IEEE Transactions on Pattern Analysis and Machine Intelligence. IEEE; 2015;**37**(9):1834-1848

[66] Stalder S, Grabner H, Van Gool L. Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition. In: ICCVw'09; 2009

[67] Xing J, Gao J, Li B, Hu W, Yan S. Robust object tracking with online multi-lifespan dictionary learning. In: ICCV'13; 2013. pp. 665-672

[68] Zhuang B, Wang L, Lu H. Visual tracking via shallow and deep collaborative model. Neurocomputing. 2016;**218**:61-71

[69] Blum A, Mitchell T. Combining labeled and unlabeled data with co-training. In: COLT'98; 1998

[70] Vijayanarasimhan S, Grauman K. Cost-Sensitive Active Visual Category Learning. International Journal of Computer Vision. Springer; 2011;**91**(1):24-44

[71] Razavi N, Gall J, Kohli P, Van Gool L. Latent Hough transform for object detection. In: ECCV'12; 2012

[72] Zhu X, Vondrick C, Ramanan D, Fowlkes CC. Do we need more training data or better models for object detection? In: BMVC'12; 2012

[73] De la Torre F, Black MJ. Robust principal component analysis for computer vision. In: ICCV'01; 2001

[74] Bengio Y, Louradour J, Collobert R, Weston J. Curriculum learning. In: ICML'09; 2009

[75] Lu J, Issaranon T, Forsyth D. Safetynet: Detecting and Rejecting Adversarial Examples Robustly. arXiv. 2017

[76] Lewis DD, Gale WA. A sequential algorithm for training text classifiers. In: ACM SIGIR'94; 1994. pp. 3-12

[77] Lampert CH, Peters J. Active structured learning for high-speed object detection. In: PR; Springer; 2009. pp. 221-231

[78] Li C, Wang X, Dong W, Yan J, Liu Q, Zha H. Active sample learning and feature selection: A unified approach. arXiv. 2015

[79] Beygelzimer A, Dasgupta S, Langford J. Importance weighted active learning. In: ICML'09; ACM; 2009. pp. 49-56

[80] Han B, Sim J, Adam H. Branchout: Regularization for online ensemble tracking with convolutional neural networks. In: ICCV'17; 2017. pp. 2217-2224

[81] Meshgi K, Oba S, Ishii S. Efficient version-space reduction for visual tracking. In: CRV'17; 2017

[82] Meshgi K, Oba S, Ishii S. Active discriminative tracking using collective memory. In: MVA'17; 2017

[83] Oza NC. Online bagging and boosting. In: SMC'05; 2005

[84] Saffari A, Leistner C, Godec M, Bischof H. Robust multi-view boosting with priors. In: ECCV'10; 2010

[85] Leistner C, Saffari A, Bischof H. Miforests: Multiple-instance learning with randomized trees. In: ECCV'10; 2010

[86] Seung S, Opper M, Sompolinsky H. Query by committee. In: COLT'92; 1992

[87] Kwon J, Lee KM. Visual tracking decomposition. In: CVPR'10; 2010

[88] Settles B. Active Learning. Morgan & Claypool Publishers; 2012

[89] Abe N, Mamitsuka H. Query learning strategies using boosting and bagging. In: ICML'98; 1998

[90] Breiman L. Bagging predictors. Machine Learning. Springer; 1996;**24**(2):123-140