Online Discriminative Tracking With Active Example Selection

Min Yang, Yuwei Wu, Mingtao Pei, Bo Ma, and Yunde Jia, *Member, IEEE*

***Abstract*— Most existing discriminative tracking algorithms use a sampling-and-labeling strategy to collect examples and treat the training example collection as a task that is independent of classifier learning. However, the examples collected directly by sampling are neither necessarily informative nor intended to be useful for classifier learning. Updating the classifier with these examples might introduce ambiguity to the tracker. In this paper, we present a novel online discriminative tracking framework that explicitly couples the objectives of example collection and classifier learning. Our method uses Laplacian regularized least squares (LapRLS) to learn a robust classifier that can sufficiently exploit unlabeled data and preserve the local geometrical structure of the feature space. To ensure the high classification confidence of the classifier, we propose an active example selection approach to automatically select the most informative examples for LapRLS. Part of the selected examples that satisfy strict constraints are labeled to enhance the adaptivity of our tracker, which actually provides robust supervisory information to guide semisupervised learning. With active example selection, we are able to avoid the ambiguity introduced by an independent example collection strategy and to alleviate the drift problem caused by misaligned examples. Comparison with the state-of-the-art trackers on the comprehensive benchmark demonstrates that our tracking algorithm is more effective and accurate.**

***Index Terms*— Active example selection, active learning, discriminative tracking, semisupervised learning.**

I. Introduction

VISUAL tracking aims to estimate the trajectory of an object automatically in a video sequence. Although the task is easily fulfilled by the human vision system, designing a robust online tracker remains a very challenging problem due to appearance variations caused by factors such as illumination changes, occlusion, background clutter, and object deformation.

Numerous tracking algorithms have been proposed to address appearance variations, and most of them fall into two categories: generative and discriminative methods. Generative methods represent an object in a particular feature space and then find the best candidate with maximal matching score. Some popular generative trackers include incremental visual tracking [1], visual tracking decomposition (VTD) [2], sparse representation-based tracking [3]–[7], and least soft-threshold squares tracking (LSST) [8]. Discriminative methods cast tracking as a binary classification problem that distinguishes the object from the background [9]–[15]. Benefiting from the explicit consideration of background information, discriminative trackers are usually more robust against appearance variations under complex environments. In this paper, we focus on learning an online classifier that is able to capture appearance changes adaptively for visual tracking.

The performance of discriminative trackers largely depends on the training examples used for classifier learning. Existing algorithms often collect training examples via a two-stage strategy [11]: sampling and labeling. The sampling process generates a set of examples around the current tracking result, and the labeling process estimates the labels of these examples using a heuristic approach that depends on the current tracking result (e.g., examples with small distance to the current track are labeled as positive, and examples far away from the current track are negative).

This example collection strategy raises several issues. First, the objective of the sampling process may not be consistent with the objective for the classifier, which makes the example collection strategy independent of classifier learning. The examples collected directly by sampling are neither necessarily informative nor intended to be useful for classifier learning, and might introduce ambiguity to the tracker. Second, assigning labels estimated by the current tracking result to unlabeled examples can cause drift [10], [11], [16]. Slight inaccuracy of tracking results can lead to incorrectly labeled examples and consequently degrades the classifier. State-of-the-art discriminative trackers mainly focus on learning a classifier that is robust to poorly labeled examples (e.g., semisupervised learning [16]–[19], P-N learning [20], multiple instance learning (MIL) [10], and self-paced learning [21]). However, the first issue is rarely mentioned in the literature of visual tracking.

In this paper, we frame the training example collection problem in discriminative tracking as one of active learning, and propose an active example selection approach to automatically select the most informative examples for classifier learning. Based on active example selection, we present a novel online discriminative tracking framework that explicitly couples the objectives of example collection and classifier

Manuscript received April 4, 2014; revised September 9, 2014 and December 3, 2014; accepted January 17, 2015. Date of publication July 2, 2015; date of current version July 7, 2016. This work was supported in part by the 973 Program of China under Grant 2012CB720000, in part by the Natural Science Foundation of China under Grant 61203291 and Grant 61375044, and in part by the Specialized Research Fund for the Doctoral Program of Chinese Higher Education under Grant 20121101110035. This paper was recommended by Associate Editor J.-M. Odobez.

The authors are with the Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081, China (e-mail: yangminbit@bit.edu.cn; wuyuwei@bit.edu.cn; peimt@bit.edu.cn; bma000@bit.edu.cn; jiayunde@bit.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2015.2395791

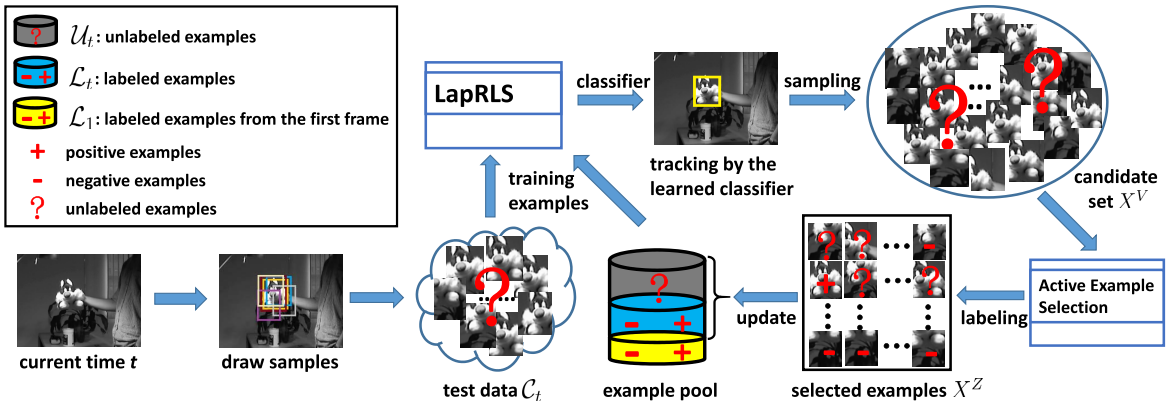


Fig. 1. Overview of our tracker. LapRLS is used to learn a robust classifier that is able to exploit both labeled and unlabeled data during tracking. An active example selection stage is introduced between sampling and labeling, which explicitly couples the objectives of example collection and classifier learning. The figure is best viewed in color.

learning, avoiding the ambiguity introduced by the two-stage example collection strategy.

The overview of our approach is shown in Fig. 1. A manifold regularized semisupervised learning method, i.e., Laplacian regularized least squares (LapRLS) [22], is employed to learn a robust classifier that is able to exploit both labeled and unlabeled data. LapRLS sufficiently utilizes the discriminative information contained in unlabeled data, and hence alleviate the drift problem caused by label noise. Using the formalism of active learning [23], [24], we introduce an *active example selection* stage between sampling and labeling to select the examples that are useful for LapRLS. The active example selection guarantees the consistency between example collection and classifier learning in a principled manner, and significantly improves the tracking performance, as our experiments demonstrated.

To make the classifier more adaptive to appearance changes, part of the selected examples that satisfy strict constraints are labeled, and the rest are considered as unlabeled data. According to the stability–plasticity dilemma [9], the additional labels provide reliable supervisory information to guide semisupervised learning during tracking, and hence increase the plasticity of the tracker. Our experiments suggest that this conservative labeling approach is crucial to handle appearance variations.

The rest of this paper is organized as follows. Section II reviews the related work. Section III introduces the details of classifier learning and active example selection. Detailed description of our tracking algorithm is provided in Section IV. We report and discuss the experimental results in Section V, and conclude this paper in Section VI.

II. Related Work

Much progress has been made in modeling appearance variations for visual tracking. A thorough review can be found in [25]. Our tracker incorporates ideas from prior work on semisupervised tracking, active learning, and stability–plasticity dilemma. We briefly review relevant literature on these three topics in the following.

A. Semisupervised Tracking

Semisupervised approaches have been previously used in tracking. Grabner *et al.* [16] proposed an online semisupervised boosting tracker to avoid self-learning as only the examples in the first frame are considered as labeled. Saffari *et al.* [17] proposed a multiview boosting algorithm that considers the given priors as a regularization component over the unlabeled data, and validated its robustness for object tracking. Kalal *et al.* [20] presented a P-N learning algorithm to bootstrap a prior classifier by iteratively labeling unlabeled examples via structural constraints. Gao *et al.* [19] employed the cluster assumption to exploit unlabeled data to encode most of the discriminant information of their tensor representation, and showed great improvement on tracking performance.

The semisupervised methods mentioned above actually determine the pseudolabel of the unlabeled data, and do not discover the intrinsic geometrical structure of the feature space. In contrast, the LapRLS algorithm employed in our method learns a classifier that predicts similar labels for similar data points by constructing a data adjacency graph. We show that it is crucial to consider the similarity in terms of label prediction during tracking. Bai and Tang [18] introduced a similar algorithm, i.e., Laplacian ranking SVM, for object tracking. Their method formulates the tracking process as a ranking problem and also incorporates the information of unlabeled examples via a manifold regularization. However, they adopt a handcrafted example collection strategy to obtain the labeled and unlabeled data, which limits the performance of their tracking method. Compared with the method in [18], our tracker is able to automatically select useful examples for classifier learning, which can efficiently exploit the discriminative information contained in the abundant unlabeled examples with a relatively small set of training examples. Our experiments demonstrate the superiority of our tracker over the traditional handcrafted example collection strategies.

B. Active Learning

Active learning, also referred to as experimental design in statistics, aims to determine which unlabeled examples would

be the most informative (i.e., improve the classifier the most if they were labeled and used as training data) [23], [24], and has been well applied in text categorization [26] and image retrieval [27], [28]. In this paper, we propose an active example selection approach in our example collection strategy using the framework of active learning, in which the task is to select the examples that improve the prediction accuracy of LapRLS the most.

We show that the active example selection approach introduces several advantages for visual tracking over existing methods. First, it guarantees the consistency between example collection and classifier learning in a principled way. That is, the selected examples are meaningful for LapRLS, which can improve the classification performance. Second, the active example selection tends to choose the representative examples, which reduces the amount of training data without performance loss. Third, assigning labels to the selected examples alleviates the drift problem caused by label noise. According to the theory of active learning, the examples that minimize the predictive variance when they are used for training will be selected. Thus, misaligned examples are intended to be rejected by the active example selection.

C. Stability–Plasticity Dilemma

We revisit the stability–plasticity dilemma to present more implication of our approach. If the classifier is trained only with the labeled examples from the first frame, it is the most *stable* description of the object appearance and can virtually not drift, but fails to track an object that undergoes appearance variations over time. On the other hand, an online classifier that bootstraps itself using the examples extracted from the current tracking result is a more *plastic* description. It is highly adaptive but easily drifts in the case of updating with mislabeled examples. Our tracker is designed to achieve a proper balance between stability and plasticity. To obtain a stable appearance model, we learn the classifier using LapRLS that is able to exploit unlabeled data effectively and avoid self-learning, and update the classifier using informative examples selected via active example selection to ensure the high classification confidence. Meanwhile, a relatively small number of examples are labeled using a conservative labeling approach, which increases the plasticity of the model as the labeled data contain additional supervisory information.

Several tracking algorithms are also designed from the aspect of stability–plasticity dilemma. Stalder *et al.* [29] extended the semisupervised boosting tracker [16] by integrating an adaptive prior, i.e., an supervised online classifier, to increase the plasticity of the model. Santner *et al.* [9] combined three components that have different adaptivity rates: template matching (stable), optical-flow-based mean-shift tracker (highly adaptive), and online random forest (moderately adaptive) to increase the stability and plasticity of the tracker at the same time. Gu *et al.* [30] proposed an online nearest neighbor classifier for efficient visual tracking, which involves a feature updating and pruning scheme to obtain a suitable tradeoff between plasticity and stability. Different from the trackers mentioned above, both the labeled and unlabeled examples used for online classifier update in our tracker are selected via active example selection, which can simultaneously choose the most informative examples and ensure the accuracy of classification.

III. Learning an Adaptive Classifier

Online discriminative trackers mainly focus on learning an adaptive classifier to separate the object from the background. Two stages, i.e., classifier learning and training example collection, are alternately performed to obtain the adaptive classifier during tracking. While most discriminative tracking algorithms consider them as two independent tasks, our method explicitly couples the objectives of example collection and classifier learning by introducing an active example selection approach. In this section, we describe how the classifier is learned with LapRLS and how the useful examples are selected for LapRLS via active example selection. To update the training example set, we also present a conservative labeling approach to estimate the labels of the selected examples.

A. Classifier Learning With LapRLS

Denote the feature space by $\mathcal{X}$, and labeled examples can be encoded as (x, y) pairs, where $x \in \mathcal{X}$ and $y \in \mathbb{R}$, and unlabeled examples are simply $x \in \mathcal{X}$. We focus on the binary classification problem and assume that positive examples are labeled with $+1$ and negative examples are labeled with -1.

At each time step during tracking, we denote the current labeled example set as $(X^L, \mathbf{y}) = \{(x_i, y_i)\}_{i=1}^{l}$ and the current unlabeled example set as $X^U = \{x_i\}_{i=l+1}^{l+u}$, where $\mathbf{y}$ is the label vector of X^L, and l and u are the numbers of labeled and unlabeled examples, respectively. The entire example set is denoted by $X^E = \{x_i\}_{i=1}^{l+u}$.

Given the training example set, our goal is to learn a label prediction function $f : \mathcal{X} \to \mathbb{R}$ that will generalize well on new examples. In this paper, we adopt the LapRLS algorithm to seek an optimal real-valued function that not only achieves low predictive error on labeled training data but also has high prediction confidence on unlabeled training data. Formally, the LapRLS algorithm solves the following optimization problem [22]:

$$f^* = \arg\min_{f \in \mathcal{H}_K} \sum_{x_i \in X^L} (y_i - f(x_i))^2 + \lambda_1 \|f\|_K^2 + \frac{\lambda_2}{2} \sum_{x_i, x_j \in X^E} (f(x_i) - f(x_j))^2 W_{ij} \quad (1)$$

where $\mathcal{H}_K$ is a reproducing kernel Hilbert space (RKHS) which is associated with a Mercer kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, $\|\cdot\|_K$ is the norm defined in $\mathcal{H}_K$, the function value $f(x_i)$ is the prediction of the label of example x_i given by f, and W is a $(l+u) \times (l+u)$ similarity matrix with entries W_{ij} indicating the adjacency weights between data points x_i and x_j.

The first term in (1) is a squared loss which makes the trained function have a high prediction accuracy on the labeled data. The second term in (1) penalizes the RKHS norm of the trained function, which imposes smoothness conditions on possible solutions and actually restricts the scale of the

optimization problem. The last term in (1) is an approximated manifold regularizer that preserves the local geometrical structure represented by a weighted adjacency graph with similarity matrix W. It actually respects a smoothness assumption, that is, data points closed to each other in a high-density region should share similar predictions given by the trained function. According to the spectral graph theory, this regularized term can be expressed as a compact form

$$\frac{1}{2}\sum_{x_i,x_j\in X^E}(f(x_i)-f(x_j))^2W_{ij}=\mathbf{f}^\top L\mathbf{f} \tag{2}$$

where $\mathbf{f}=[f(x_1),f(x_2),\ldots,f(x_{l+u})]^\top$, and L is the graph Laplacian given by $L=D-W$. Here, D is a diagonal matrix defined as $D_{ii}=\sum_{j=1}^{l+u}W_{ij}$. We adopt the local scaling method [31] to define the similarity matrix

$$W_{ij}=\begin{cases}\exp\left(-\frac{\|x_i-x_j\|_2^2}{\sigma_i\sigma_j}\right), & \text{if } i\in N_k^j \text{ or } j\in N_k^i\\ 0, & \text{otherwise}\end{cases} \tag{3}$$

where N_k^i indicates the index set of the k nearest neighbors of x_i in X^E, $\sigma_i=\|x_i-x_i^{(k)}\|_2$, and $x_i^{(k)}$ is the kth nearest neighbor of x_i in X^E. Note that the entries W_{ij} are nonnegative, which guarantees the positive semidefinite property of the graph Laplacian L and further ensures that the regularization term $\mathbf{f}^\top L\mathbf{f}$ is convex.

The key motivation for restricting f in an RKHS is the representer theorem (see the details in [22]) which shows that the solution of (1) is an expansion of kernel functions over both labeled and unlabeled data

$$f^*(x)=\sum_{x_i\in X^E}\omega_i^*K(x,x_i) \tag{4}$$

where $\omega_i^*\in\mathbb{R}$ is the coefficient of x_i. We use the following notations to denote the kernel functions:

$$\begin{aligned}(K_{x,E})_{1j}&=K(x,x_j),\quad x_j\in X^E\\ (K_{LE})_{ij}&=K(x_i,x_j),\quad x_i\in X^L,\ x_j\in X^E\\ (K_{EL})_{ij}&=K(x_i,x_j),\quad x_i\in X^E,\ x_j\in X^L\\ (K)_{ij}&=K(x_i,x_j),\quad x_i\in X^E,\ x_j\in X^E.\end{aligned} \tag{5}$$

Thus, (4) can be simplified as

$$f^*(x)=K_{x,E}\boldsymbol{\omega}^* \tag{6}$$

where $\boldsymbol{\omega}^*=[\omega_1^*,\ldots,\omega_{l+u}^*]^\top$. By substituting (6) into (1), we obtain a convex differentiable objective function of the $(l+u)$-dimensional vector $\boldsymbol{\omega}=[\omega_1,\ldots,\omega_{l+u}]^\top$

$$\boldsymbol{\omega}^*=\arg\min_{\boldsymbol{\omega}\in\mathbb{R}^{l+u}}\|\mathbf{y}-K_{LE}\boldsymbol{\omega}\|^2+\lambda_1\boldsymbol{\omega}^\top K\boldsymbol{\omega}+\lambda_2\boldsymbol{\omega}^\top KLK\boldsymbol{\omega} \tag{7}$$

where we use the fact $\|f\|_K=\boldsymbol{\omega}^\top K\boldsymbol{\omega}$, and $\mathbf{y}=[y_1,\ldots,y_l]^\top$ is the label vector of X^L.

The solution of (7) can be acquired by setting the gradient with respect to $\boldsymbol{\omega}$ to zero

$$\boldsymbol{\omega}^*=(K_{EL}K_{LE}+\lambda_1K+\lambda_2KLK)^{-1}K_{EL}\mathbf{y}. \tag{8}$$

Obviously, the optimal prediction function f^* can be efficiently obtained by solving a single system of linear equations described in (8), and then the predicted label of a test data x is given by the sign of $f^*(x)$ using (6).

B. Active Example Selection

With the prediction function learned by LapRLS, the tracked object can be located as described later in Section IV. In this section, we will show how the useful examples are selected by an active example selection approach to update the training example set of LapRLS.

1) Formulation: Given the object location at each frame, a large set of unlabeled examples is generated by randomly sampling around the object location, denoted by $X^V=\{x_i\}_{i=l+u+1}^{l+u+n}$, where n is the number of examples generated by sampling. We consider the example selection problem from the perspective of active learning, where the task is to automatically choose a set of m examples X^Z from X^V that together are maximally informative [23]. The informativeness of the selected examples X^Z is indicated by the performance of the classifier learned using X^Z as labeled data. Therefore, we seek a subset $X^Z\subset X^V$ that maximizes the prediction confidence of the classifier learned by LapRLS.

Suppose that we can observe the labels of the examples in X^Z by a measurement process $c_i=f(x_i)+\epsilon_i, x_i\in X^Z$, where c_i is the observed label of example x_i, f is the underlying label prediction function and $\epsilon_i\sim\mathcal{N}(0,\sigma^2)$ is the measurement noise. The observed labels c_i of different examples have measurement errors that are independent, but with equal variances σ^2. Using X^Z as labeled data and the rest in X^V as unlabeled data, the estimate of f, denoted by $\hat{f}$, can be obtained using LapRLS. For clarity, we recall (6) and (8) with different notations

$$\hat{f}(x)=K_{x,V}\hat{\boldsymbol{\omega}} \tag{9}$$

$$\hat{\boldsymbol{\omega}}=(K_{VZ}K_{ZV}+\lambda_1K+\lambda_2KLK)^{-1}K_{VZ}\mathbf{c} \tag{10}$$

where $\mathbf{c}=[c_1,\ldots,c_m]^\top$, and the kernel matrix $K_{x,V}$, K_{ZV}, K_{VZ} and K are defined similar to (5). Since the underlying function f is unknown, the labels c_i are actually invisible. Fortunately, we will show later that our active example selection approach is not dependent on the labels c_i.

Denote $H=K_{VZ}K_{ZV}+\lambda_1K+\lambda_2KLK$ and $\Delta=\lambda_1K+\lambda_2KLK$, and the covariance matrix of $\hat{\boldsymbol{\omega}}$ can be expressed as

$$\begin{aligned}\mathrm{Cov}(\hat{\boldsymbol{\omega}})&=\mathrm{Cov}(H^{-1}K_{VZ}\mathbf{c})\\ &=H^{-1}K_{VZ}\mathrm{Cov}(\mathbf{c})K_{ZV}H^{-1}\\ &=\sigma^2H^{-1}(H-\Delta)H^{-1}\\ &=\sigma^2(H^{-1}-H^{-1}\Delta H^{-1})\end{aligned} \tag{11}$$

where the third equation uses the assumption $\mathrm{Cov}(\mathbf{c})=\sigma^2I$. The covariance matrix $\mathrm{Cov}(\hat{\boldsymbol{\omega}})$ characterizes the confidence of the estimation, or the informativeness of the selected examples. The smaller the elements of the covariance matrix, the more stable the estimator $\hat{\boldsymbol{\omega}}$, which indicates that the selected examples are more informative. According to the theory of optimum experiment design [24], different criteria can be applied to the covariance matrix to obtain different active learning algorithms for LapRLS. He [28] used the D-optimality criterion that minimizes the determinant of $\mathrm{Cov}(\hat{\boldsymbol{\omega}})$ to design an active learning algorithm for image retrieval. However, this criterion does not directly consider the quality of predictions on test data.

Inspired by [26], we design the objective of active example selection in a transductive setting. Let $\mathbf{f}_V = [f(x_{l+u+1}), \ldots, f(x_{l+u+n})]^\top$ be the true labels of all examples in X^V given by the underlying label prediction function f, and $\hat{\mathbf{f}}_V = [\hat{f}(x_{l+u+1}), \ldots, \hat{f}(x_{l+u+n})]^\top$ be the predictions on X^V given by the estimator $\hat{f}$. Then the covariance matrix of the predictive error $\mathbf{f}_V - \hat{\mathbf{f}}_V$ is given by

$$\begin{aligned}\mathrm{Cov}(\mathbf{f}_V - \hat{\mathbf{f}}_V) &= \mathrm{Cov}(\hat{\mathbf{f}}_V) = \mathrm{Cov}(K\hat{\boldsymbol{\omega}}) \\ &= K\mathrm{Cov}(\hat{\boldsymbol{\omega}})K \\ &= \sigma^2 K(H^{-1} - H^{-1}\Delta H^{-1})K. \end{aligned} \quad (12)$$

Compared with $\mathrm{Cov}(\hat{\boldsymbol{\omega}})$, $\mathrm{Cov}(\mathbf{f}_V - \hat{\mathbf{f}}_V)$ directly characterizes the quality of predictions on the entire data X^V. We aim to select m examples X^Z from X^V such that the average predictive variance $1/n\mathrm{Tr}(\mathrm{Cov}(\mathbf{f}_V - \hat{\mathbf{f}}_V))$ is minimized, i.e., a high confidence of predictions on X^V is ensured. Since the regularization parameters (i.e., λ_1 and λ_2) are usually very small, we have

$$\mathrm{Tr}(K(H^{-1} - H^{-1}\Delta H^{-1})K) \approx \mathrm{Tr}(KH^{-1}K). \quad (13)$$

Therefore, the formulation of our active example selection approach can be expressed as

$$\begin{aligned}&\min_{X^Z} \quad \mathrm{Tr}(K(K_{VZ}K_{ZV} + \lambda_1 K + \lambda_2 KLK)^{-1}K) \\ &\text{s.t.} \quad X^Z \subset X^V, \quad |X^Z| = m. \end{aligned} \quad (14)$$

Note that the example selection itself is independent of the observed labels $\mathbf{c}$, despite the fact that we consider a semisupervised learning problem.

2) Sequential Optimization: The problem (14) is a combinatorial optimization problem which is NP-hard. Similar to [26], we use a sequential greedy optimization approach to efficiently solve (14). The sequential approach selects just one example in each iteration until m examples have been selected. Denote the selected examples in the previous iterations by $X^{Z'}$, and the task of each iteration is to seek a new example x by solving

$$\begin{aligned}&\min_{x} \quad \mathrm{Tr}(K(K_{VZ}K_{ZV} + \lambda_1 K + \lambda_2 KLK)^{-1}K) \\ &\text{s.t.} \quad X^Z = X^{Z'} \cup x, \quad x \in X^V - X^{Z'}. \end{aligned} \quad (15)$$

We have the following proposition.

Proposition 1: The problem (15) is equivalent to

$$\begin{aligned}&\max_{x} \quad \|M_{V,x}\|^2/(1 + M_{x,x}) \\ &\text{s.t.} \quad x \in X^V - X^{Z'} \end{aligned} \quad (16)$$

where $M = K(K_{VZ'}K_{Z'V} + \lambda_1 K + \lambda_2 KLK)^{-1}K$, $M_{V,x}$ and $M_{x,x}$ are x's column and diagonal entry in M, respectively.

Proof: Using the fact $K_{VZ}K_{ZV} = K_{VZ'}K_{Z'V} + K_{V,x}K_{x,V}$, we can define $\tilde{\Delta} = K_{VZ'}K_{Z'V} + \Delta$ and have

$$\begin{aligned}&K(K_{VZ}K_{ZV} + \lambda_1 K + \lambda_2 KLK)^{-1}K \\ &= K(K_{V,x}K_{x,V} + K_{VZ'}K_{Z'V} + \Delta)^{-1}K \\ &= K(K_{V,x}K_{x,V} + \tilde{\Delta})^{-1}K \\ &= K\tilde{\Delta}^{-1}K - K\tilde{\Delta}^{-1}K_{V,x}K_{x,V}\tilde{\Delta}^{-1}K/(1 + K_{x,V}\tilde{\Delta}^{-1}K_{V,x}) \end{aligned}$$

where the third equation uses the Woodbury matrix identity.

Algorithm 1 Sequential Active Example Selection

1: **Initialize:** $X^{Z'}$; $X^Z = \emptyset$
2: $M = K(K_{VZ'}K_{Z'V} + \lambda_1 K + \lambda_2 KLK)^{-1}K$
3: **while** $|X^Z| < m$ **do**
4: select x by solving (16);
5: $X^{Z'} = X^{Z'} \cup \{x\}$, $X^Z = X^Z \cup \{x\}$;
6: $M \leftarrow M - M_{V,x}M_{x,V}/(1 + M_{x,x})$;
7: **end while**
8: **return** X^Z

Let $M = K\tilde{\Delta}^{-1}K = K(K_{VZ'}K_{Z'V} + \lambda_1 K + \lambda_2 KLK)^{-1}K$. It can be easily validated that $K\tilde{\Delta}^{-1}K_{V,x}$ indicates x's column in M, $K_{x,V}\tilde{\Delta}^{-1}K$ indicates x's row in M, and $K_{x,V}\tilde{\Delta}^{-1}K_{V,x}$ indicates x's diagonal entry in M. Therefore, we have

$$K(K_{VZ}K_{ZV} + \lambda_1 K + \lambda_2 KLK)^{-1}K = \frac{M - M_{V,x}M_{x,V}}{(1 + M_{x,x})}.$$

Since M is independent of x, problem (15) is equivalent to looking for the x that maximizes $\mathrm{Tr}(M_{V,x}M_{x,V}/(1+M_{x,x})) = \|M_{V,x}\|^2/(1 + M_{x,x})$. This completes the proof. □

Equation (16) can be easily solved by selecting $x \in X^V - X^{Z'}$ with the highest $\|M_{V,x}\|^2/(1 + M_{x,x})$. Then, the matrix M is updated by $M = M - M_{V,x}M_{x,V}/(1+M_{x,x})$, as the set $X^{Z'}$ is augmented with the new example x.

Starting from a set $X^{Z'}$, m most informative examples can be sequentially selected by solving (16) iteratively. We summarize our active example selection approach in Algorithm 1. Note that we only need to calculate the Moore–Penrose inverse of $(K_{VZ'}K_{Z'V}+\lambda_1 K+\lambda_2 KLK)$ when M is initialized, and there is no need for matrix inverse at each iterative step.

3) Discussion: In the simplest situation, the previously selected example set $X^{Z'}$ is set to an empty set before example selection. It is clear that the examples are selected from X^V without considering the current training example set in this case. In fact, we found that the proposed sequential optimization algorithm allows us to incorporate the current training examples into the example selection problem in a very simple way, which further enhances the informativeness of the selected examples for classifier learning. Formally, we set $X^{Z'} = X^L$ and augment the candidate set as $X^V \cup X^L$ before we perform Algorithm 1 to select useful examples. Since the labeled examples contain the most discriminative information, we only incorporate the current labeled example set X^L into the algorithm in practice.

C. Labeling

Given the example set X^Z selected via active example selection, we estimate the labels of X^Z using a conservative labeling approach similar to [11]. Denote the region of the current tracking result by R_T and the region of selected example z_i by R_Z^i, the overlap rate between R_T and R_Z^i can be computed by

$$s(R_T, R_Z^i) = \frac{\mathrm{area}(R_T \cap R_Z^i)}{\mathrm{area}(R_T \cup R_Z^i)}. \quad (17)$$

Then, we estimate the labels of X^Z according to the following constraints: the examples with the overlap rate larger than a threshold δ are labeled as positive, the examples with the overlap rate less than a threshold ϵ are labeled as negative, and the rest examples are considered as unlabeled data. The labels estimated by the conservative labeling approach can provide reliable supervisory information to guide semisupervised learning and enhance the adaptivity of the tracker.

IV. Proposed Tracking Algorithm

In this paper, we cast visual tracking as a randomized search task. At every time step t, the goal of our algorithm is to find the optimal state $\mathbf{s}_t$ of an object with a given image frame $\mathbf{o}_t$. Given the previous object state $\mathbf{s}_{t-1}$ at time $(t-1)$, we first generate a set of N_s samples $\{\mathbf{s}_t^i\}_{i=1}^{N_s}$ by applying random sampling according to the motion model $p(\mathbf{s}_t|\mathbf{s}_{t-1})$. Then, each sample $\mathbf{s}_t^i$ is evaluated by the observation model $p(\mathbf{o}_t|\mathbf{s}_t^i)$, and the optimal state $\mathbf{s}_t$ is acquired using a greedy strategy

$$\mathbf{s}_t = \arg\max_{\mathbf{s}_t^i} p\big(\mathbf{o}_t|\mathbf{s}_t^i\big). \tag{18}$$

The adopted randomized search approach is similar to the popular grid search approach used by many discriminative trackers [10]–[12], but shows several advantages. First, the random search approach can easily incorporate sophisticated motion models. Second, the random search approach provides an efficient way to approximate the brute force search on the state space, which makes it more suitable for object tracking. In addition, we also note that the random search approach is closely related to the particle filter framework [32] where a distribution of the object state at every frame is maintained by a finite set of weighted particles. However, maintaining a set of particles actually increases the drift potential of the tracker. Instead, the random search approach is more robust against distracters.

We describe the motion model, the observation model, and the model update scheme below, and summarize our tracker in Algorithm 2.

A. Motion Model

We represent the object state at time t as $\mathbf{s}_t = (a_t, b_t, \sigma_t)$, where (a_t, b_t) denotes the image position of the object and σ_t denotes the scale of the object. The motion model is formulated as Brownian motion, i.e., $p(\mathbf{s}_t|\mathbf{s}_{t-1}) = \mathcal{N}(\mathbf{s}_t; \mathbf{s}_{t-1}, \mathbf{\Sigma})$, where $\mathbf{\Sigma}$ is a diagonal covariance matrix which indicates the variances of translation and scale.

B. Observation Model

We use $\mathcal{L}_t$ and $\mathcal{U}_t$, respectively, to denote the labeled and unlabeled examples collected from the first frame till the current time t. Specially, $\mathcal{L}_1$ only contains the labeled examples from the first frame, which can be considered as the training examples with the most accurate labels during tracking. Therefore, we construct the current labeled example set as $(X^L, \mathbf{y}) = \mathcal{L}_t \cup \mathcal{L}_1$ to make the learned classifier more stable.

For the tracking at time t, we first crop the image patches corresponding to the samples $\{\mathbf{s}_t^i\}_{i=1}^{N_s}$ from the observed image $\mathbf{o}_t$, and use the HOG feature [33] to describe the patches. After feature extraction, we obtain a set of test data, denoted by $\mathcal{C}_t = \{x_t^i\}_{i=1}^{N_s}$, where x_t^i is the feature vector of the sample $\mathbf{s}_t^i$. The current unlabeled example set is then constructed as $X^U = \mathcal{U}_t \cup \mathcal{C}_t$. It actually indicates that the classifier is trained in a transductive setting, which improves the prediction accuracy on the test data.

With the current training set X^L and X^U, an adaptive prediction function f_t can be learned with LapRLS, as described in Section III-A. For a test data x which belongs to the object class, the prediction value given by $f_t(x)$ should be as close to $+1$ as possible according to the objective function (1) of LapRLS. Therefore, we compute the observation likelihood of the sample $\mathbf{s}_t^i$ as

$$p\big(\mathbf{o}_t|\mathbf{s}_t^i\big) \propto \exp\big(-\|1 - f_t\big(x_t^i\big)\|^2\big). \tag{19}$$

Algorithm 2 The Proposed Tracking Algorithm

Input: image sequence $\mathbf{o}_1, \cdots, \mathbf{o}_T$; an initial bounding box at the first frame to indicate the tracked object; regularization parameters λ_1 and λ_2; kernel function K
Output: tracking results $\{\mathbf{s}_t\}_{t=1}^T$
1: **Initialize:** $\mathbf{s}_1$ is the initial object state; $\mathcal{L}_1$ and $\mathcal{U}_1$ is the labeled and unlabeled examples from the first frame, respectively;
2: **for** $t = 2, \cdots, T$ **do**
3: generate samples $\mathbf{s}_t^i$,
extract features x_t^i,
$\mathcal{C}_t = \{x_t^i\}_{i=1}^{N_s}$, the test data set;
4: labeled training set $(X^L, \mathbf{y}) \Longleftarrow \mathcal{L}_t \cup \mathcal{L}_1$,
unlabeled training set $X^U \Longleftarrow \mathcal{U}_t \cup \mathcal{C}_t$,
learn function f_t with LapRLS using (6) and (8);
5: select tracking result $\mathbf{s}_t$ using (18);
6: $X^V \Longleftarrow n$ unlabeled examples generated by sampling,
$X^{Z'} \Longleftarrow X^L$,
$X^V \Longleftarrow X^V \cup X^L$,
select m examples X^Z using Algorithm 1;
7: estimate the labels of X^Z as describe in Sec. III-C,
$\mathcal{L}_t, \mathcal{U}_t \Longleftarrow X^Z$, update the training example set;
8: **end for**

C. Model Update

Note that we retrain the classifier used for tracking at each time, therefore the training example set, i.e., $\mathcal{L}_t$ and $\mathcal{U}_t$, should be updated to account for appearance variations of the object. Once the object is located, we sample a large set of unlabeled examples X^V, and employ the active example selection to select a set of informative examples X^Z from X^V, as described in Section III-B. To make the trained classifier more adaptive to appearance changes, we assign labels to part of the set X^Z that satisfy strict constraints as described in Section III-C. Then, the examples X^Z are used to update the labeled example set $\mathcal{L}_t$ and the unlabeled example set $\mathcal{U}_t$, where random replacement happens once the number of examples in $\mathcal{L}_t$ or $\mathcal{U}_t$ reaches the example set capacity $|\mathcal{L}_t|$ or $|\mathcal{U}_t|$.

V. Experimental Results

We evaluate our tracker with 11 state-of-the-art methods on a recent benchmark [34], where each tracker is tested

on 51 challenging videos (more than 29 000 frames). The state-of-the-art trackers include the Tracking-Learning-Detection (TLD) tracker [20], tracking with MIL [10], VTD [2], the Struck method [11], the sparsity-based collaborative model (SCM) [4], Laplacian ranking support vector tracking (LRSVT) [18], compressive tracking (CT) [12], structural part-based tracking (SPT) [13], LSST [8], randomized ensemble tracking (RET) [15], and tracking with online nonnegative dictionary learning (ONNDL) [7]. We use the source codes publicly available on the benchmark (except that the source codes of LRSVT, SPT, LSST, RET, and ONNDL are provided by the authors) with the same initialization and their default parameters. Since the trackers involve randomness, we run them five times and report the average result for each sequence.

A. Implementation Details

We normalize the object region to 32×32 pixels, and extract nine overlapped 18×18 local patches within the region by sliding windows with seven pixels as step length. Each patch is represented as a 32D HOG feature, and these features are grouped into a 288D feature vector. For LapRLS and active example selection, we empirically set the regularization parameters λ_1 and λ_2 to be 0.001 and 0.1, respectively. The parameter k in (3) is chosen as 7 according to [31]. To avoid parameter tuning, we apply the linear kernel to LapRLS and active example selection in our experiments. Note that nonlinear kernels (e.g., Gaussian and polynomial kernels) can be used to handle more complex data. In the first frame, 20 positive examples, 80 negative examples, and 300 unlabeled examples are used to initialize the classifier. The example set capacity $|\mathcal{L}_t| = 200$ and $|\mathcal{U}_t| = 600$. Given the object location at the current frame, $n = 1200$ unlabeled examples are generated by applying uniform sampling within a search region around the current object location, and $m = 20$ informative examples are selected by active example selection. The scales of the examples are fixed to the same as the scale of the object, and the search region is set to twice the size of the object. We set the labeling constraint parameters $\delta = 0.8$ and $\epsilon = 0.2$. For randomized search, the number of samples $N_s = 600$, and the state transition matrix $\mathbf{\Sigma} = \text{diag}(8, 8, 0.01)$. Note that the parameters are fixed throughout the experiments in this section. Our tracker is implemented in MATLAB, which runs at 2 frames/s on an Intel Core i7 3.5-GHz PC with 16-GB memory. The MATLAB source code of our tracker, together with the experimental results of the competing trackers on the benchmark, is available at http://iitlab.bit.edu.cn/mcislab/~wuyuwei.

B. Quantitative Evaluation

1) Evaluation Criteria: The center location error as well as the overlap rate is used for quantitative evaluations. The center location error is the per frame distance (in pixels) between the center of the tracking result and that of ground truth. The overlap rate is defined as $s(R_T, R_G)$ using (17), where R_T is the region of the tracking result and R_G denotes the ground truth. We employ the precision plot and the success plot [34] to evaluate the robustness of trackers, rather than directly using the average center location error and the average overlap rate over all frames of one video sequence to indicate the overall performance. The precision plot indicates the percentage of frames whose center location error is within the given threshold. The result at error threshold of 20 pixels is selected as the representative precision score and used for ranking. The success plot shows the ratios of successful frames whose overlap rate is larger than the given threshold. The area under curve (AUC) of each success plot is used to evaluate and rank the trackers.

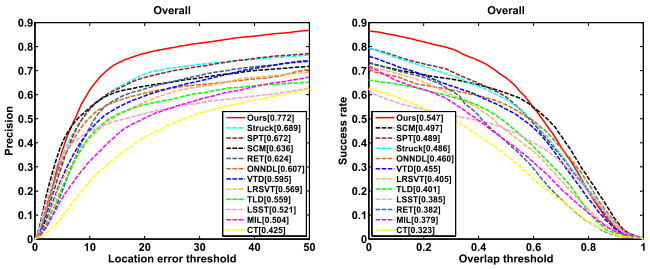


Fig. 2. Overall performances of the competing trackers on the 51 video sequences. The precision plot and the success plot are used, and the trackers are ordered according to their respective performance score in the legend.

2) Overall Performance: The overall performances of the competing trackers on the 51 sequences are illustrated by the precision plot and the success plot, as shown in Fig. 2. The trackers are ordered according to their respective performance score in the legend. From Fig. 2, we observe that both our tracker and the SCM, SPT, and Struck methods achieve good tracking performance. In the precision plot, our tracker performs 8.3% better than Struck, 10% better than SPT, and 13.6% better than the SCM. In the success plot, our tracker performs 5% better than the SCM, 5.8% better than SPT, and 6.1% better than Struck. We also observe that the SCM method provides higher precision and success rate when the error threshold is relatively small (e.g., five pixels in the precision plot, and 80% in the success rate), because of the fact that the SCM method exploits both holistic and local representation approaches based on sparse coding to handle appearance variations.

Overall, our tracker outperforms the state-of-the-art algorithms in terms of location accuracy and robustness. The reasons are explained as follows. First, LapRLS is effective for learning a robust classifier for visual tracking. It is crucial to utilize the discriminative information contained in the abundant unlabeled data, which can be easily collected during tracking, to refine the classifier. Second, the proposed active example selection method explicitly couples the objectives of example collection and classifier learning, and thus ensures the high classification confidence of the online classifier. Third, we use a conservative labeling approach to add reliable supervisory information for semisupervised learning, leading to a significant enhancement on the adaptivity of our tracker. Our experimental results validate these notions in the following sections.

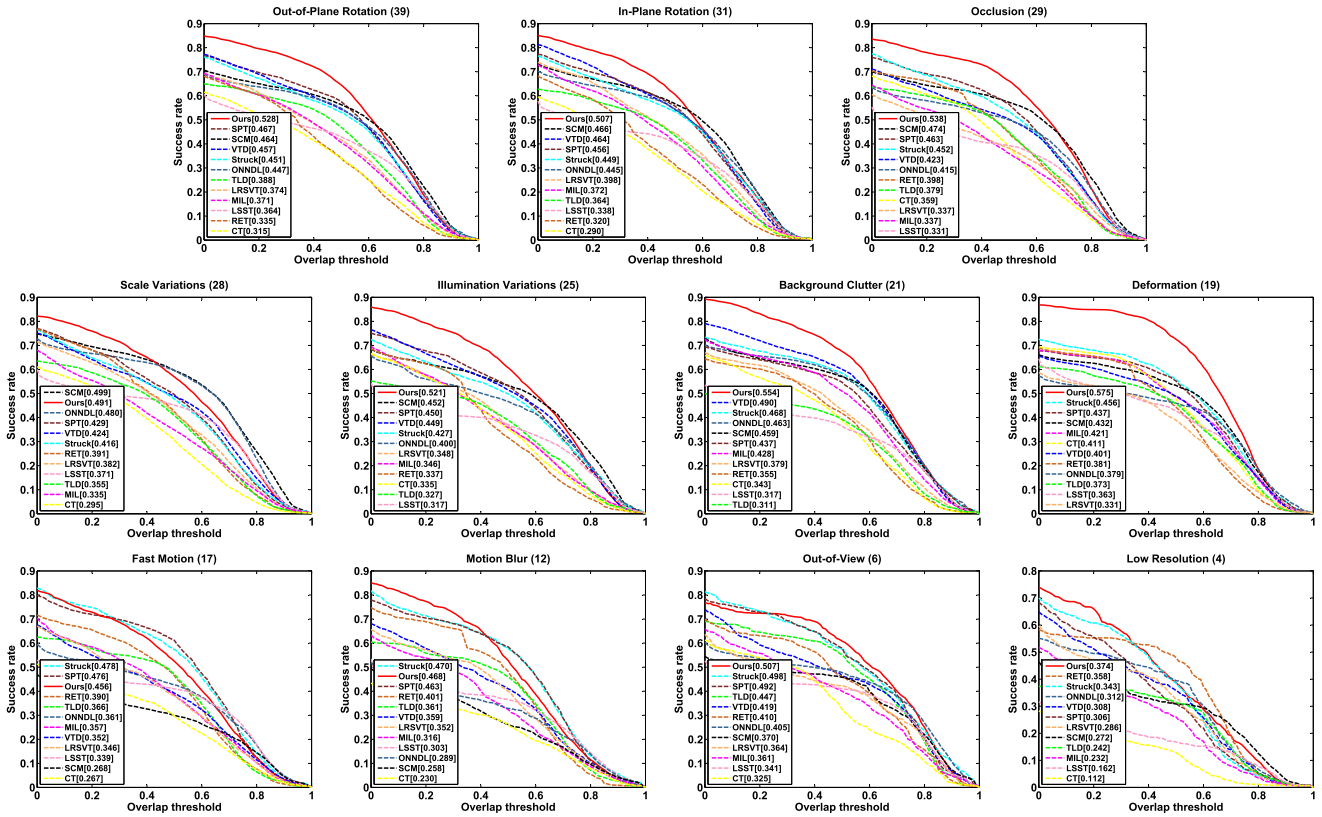


Fig. 3. Attribute-based performance analysis using success plot. The number of video sequences in each subset is shown in the title. Best viewed on high-resolution display.

3) Attribute-Based Performance Analysis: The video sequences used in the benchmark are annotated with 11 attributes which can be considered as different factors that may affect the tracking performance. One sequence can be annotated with several attributes. By putting the sequences that share a common attribute into a subset, we can analyze the performance of trackers to handle a specific challenging condition. In our experiments, we utilize the attribute-based performance analysis approach to demonstrate the robustness of our tracker. Figs. 3 and 4 show the success plots and the precision plots of the competing trackers for the 11 attributes (arranged in descending order of the number of video sequences in each subset), respectively. As shown in Fig. 3, our method shows the best tracking performance on 7 of the 11 video subsets in terms of success rate and also performs well in the other four subsets. For tracking precision, our method achieves the best performance on 9 of the 11 video subsets, as shown in Fig. 4. These results demonstrate that the proposed algorithm is robust to appearance variations caused by a set of factors. Due to space limitations, we mainly discuss the success plots for the top five attributes that occur more frequently than others.

On the occlusion subset, the SCM, SPT, and our method perform better than other trackers, which indicates that local representation methods are effective in dealing with occlusions. On the out-of-plane rotation and the in-plane rotation subsets, the SCM, SPT, VTD, Struck, ONNDL, and our method outperform others. On the scale variations subset, trackers with affine motion models (e.g., our method, SCM, and ONNDL) cope with scale variations better than others with translational motion (e.g., Struck, RET, and TLD). On the illumination variations subset, our method provides outstanding tracking results than others. It may benefit from the HOG feature used in our method that has been proved to be very robust against illumination changes. We also note that the performance of the SCM, ONNDL, and our method degrades on the fast motion and the motion blurring subsets, while grid search-based trackers (e.g., Struck, SPT, and RET) perform much better than others. The reason is that search regions of grid search-based trackers are large and the motion models of randomized search-based trackers should be carefully designed to handle fast motion.

4) Diagnostic Analysis: As previously mentioned, our tracking method chooses the most informative examples for classifier learning via active example selection, leading to significant improvement on tracking performance. In addition, we assign labels to part of the selected examples that satisfy strict constraints, which can increase the adaptivity of the classifier. To demonstrate the effectiveness of the active example selection approach and the conservative labeling strategy, we build three baseline algorithms to do validation and analyze various aspects of our method.

We begin with a naive tracker based on a classifier learned with LapRLS, denoted by BaseLine1. BaseLine1 only exploits the labeled examples from the first frame, and collects unlabeled examples using randomly sampling. We add the active

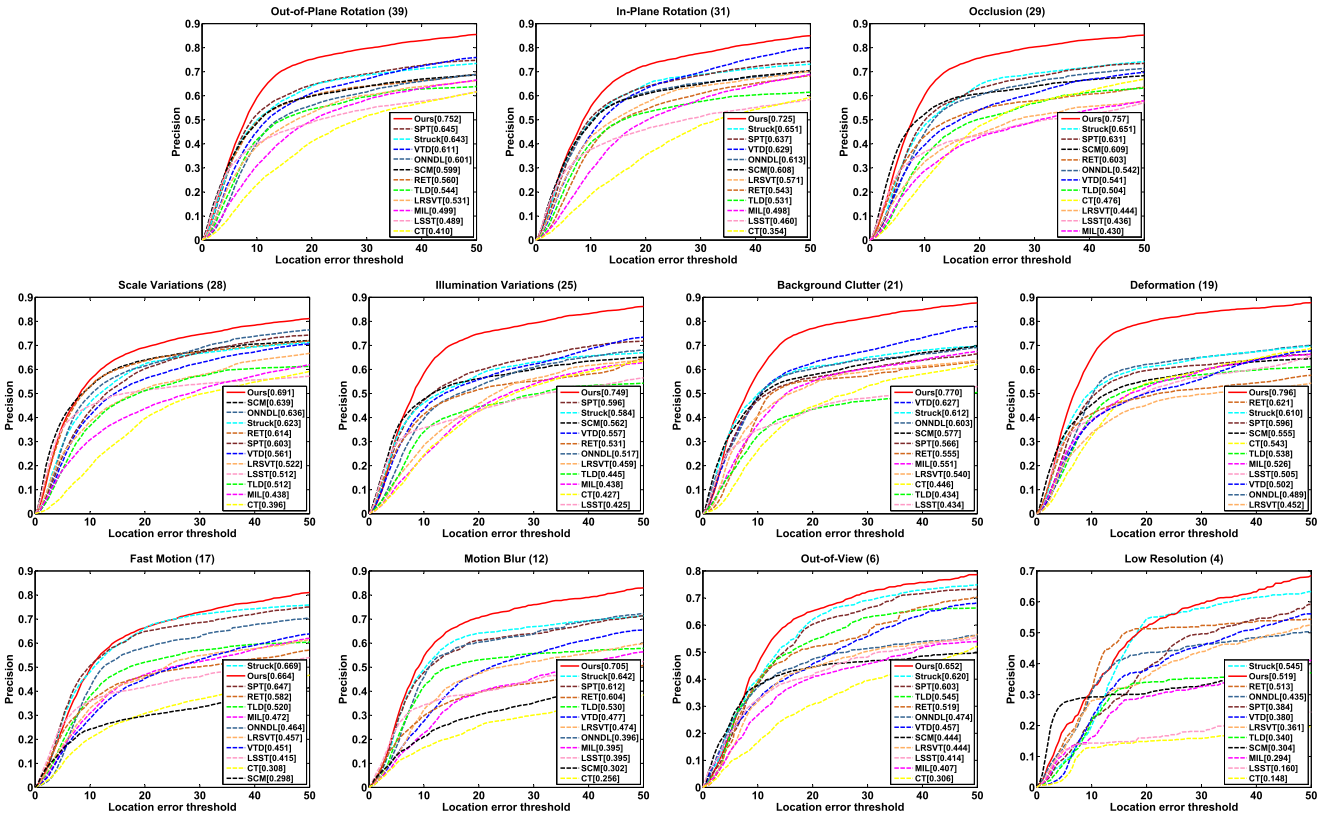


Fig. 4. Attribute-based performance analysis using precision plot. The number of video sequences in each subset is shown in the title. Best viewed on high-resolution display.

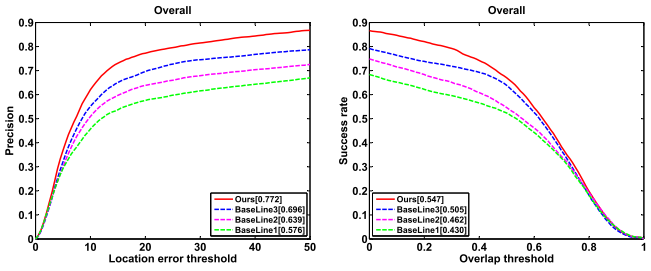
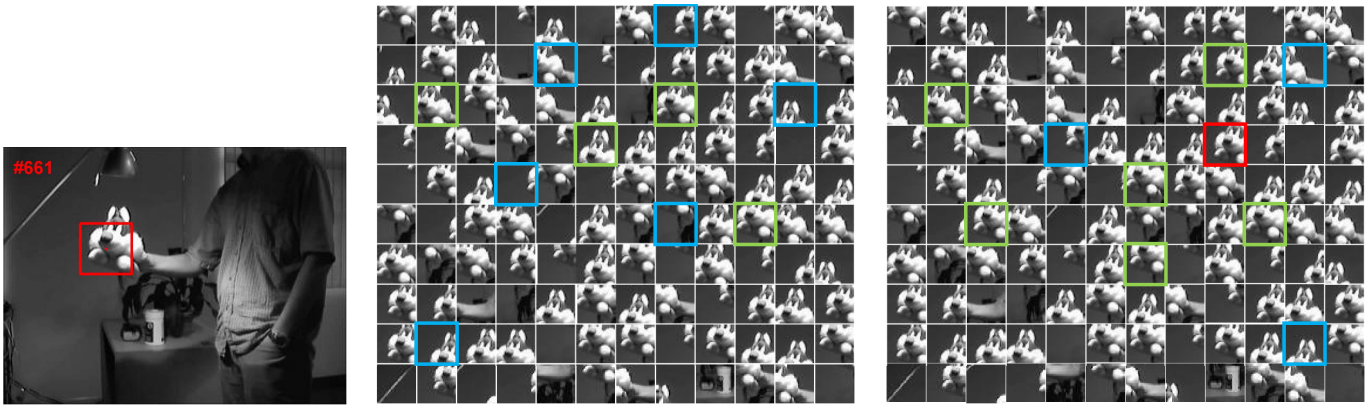


Fig. 5. Diagnostic Analysis. The overall performances of three baseline algorithms and our method on the 51 video sequences are presented for comparison in terms of precision and success rate.

example selection stage after the sampling process to select informative examples for LapRLS, leading to another baseline, denoted by BaseLine2. Both BaseLine1 and the BaseLine2 are stable versions, since no additional supervisory information is added during tracking, i.e., the training examples are collected without the labeling stage and the labeled examples come only from the first frame. We get BaseLine3 by allowing BaseLine1 to assign labels to part of the unlabeled examples using the conservative approach described in Section III-C. Note that adding supervisory information to BaseLine2 is the proposed tracking method.

The overall tracking performance of these baseline algorithms and our method is shown in Fig. 5. Surprisingly, even without additional example selection and labeling process, BaseLine1 produces good performance in terms of precision and robustness, outperforming the CT, MIL, LSST, and TLD trackers and being comparable with VTD. It demonstrates the effectiveness of LapRLS which can sufficiently exploit unlabeled data and preserve the local geometrical structure of the feature space. The performances of our method and Baseline3 are better than those of BaseLine1 and BaseLine2, which demonstrates that the additional supervisory information is significant for semisupervised learning. The conservative labeling strategy used in our tracking method achieves a suitable tradeoff between stability and plasticity in terms of capturing appearance variations. The performance of our method is significantly better than that of BaseLine3, and BaseLine2 outperforms BaseLine1. It validates the effectiveness of selecting informative examples for classifier learning. The active example selection guarantees the consistency between example collection and classifier learning, and thus improves the tracking performance. Furthermore, assigning labels to examples selected by active example selection alleviates the drift problem caused by label noise, since misaligned examples will be rejected to ensure the high prediction confidence of the classifier.

To further demonstrate the superiority of active example selection, we show an intuitive example in Fig. 6 to show the difference between our example collection strategy and the traditional sampling-and-labeling strategy that is used in most discriminative trackers (see [10], [13], [20]). We choose a specific frame (i.e., #661) in the *Sylvester* sequence where the current tracking result is slightly inaccurate,



(a) (b) (c)

Fig. 6. Intuitive example to illustrate the superiority of active example selection. Red, blue, and green borders in (b) and (c) indicate the positive, negative, and unlabeled examples, respectively. Best viewed on high-resolution display. (a) Current tracking result. (b) Examples selected by active example selection. (c) Examples selected by random selection.

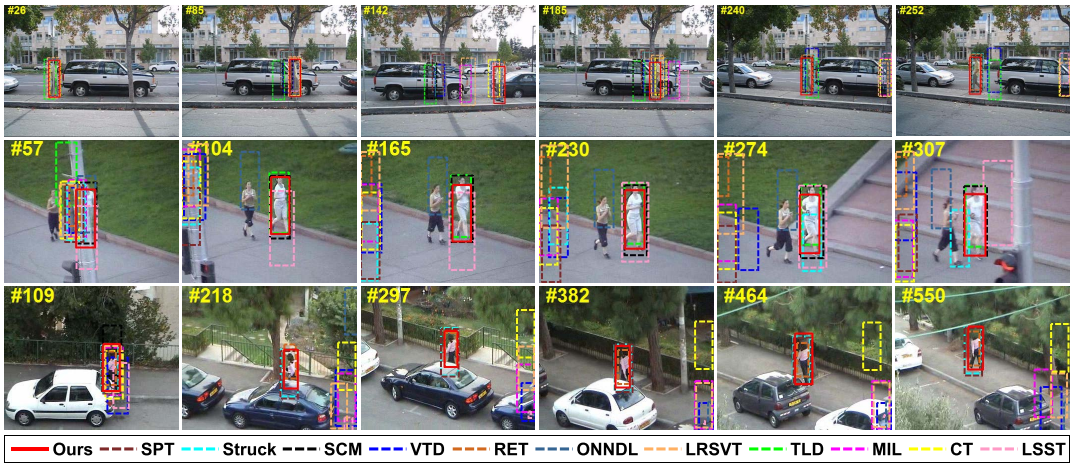


Fig. 7. From top to bottom, representative results of the competing trackers on sequences *David3*, *Jogging2*, and *Woman* (best viewed on high-resolution display). Objects are heavily occluded.

as shown in Fig. 6(b). Then 120 examples are generated by randomly sampling around the tracking result, from which we intend to select 10 training examples for classifier update. Our method employs active example selection to select the most informative examples and then estimates the labels of the selected examples, as shown in Fig. 6(b). In contrast, the traditional sampling-and-labeling strategy directly labels the 120 examples and randomly selects a fixed number of positive, negative and unlabeled examples to construct the training set. The numbers of positive, negative, and unlabeled examples are empirically preset parameters. We show in Fig. 6(c) that one positive, three negative, and six unlabeled examples are randomly selected. Note that positive, negative, and unlabeled examples have red, blue, and green borders, respectively. We can observe that the examples selected by the heuristic strategy are redundant (e.g., most unlabeled examples have similar appearances), and label noise will be introduced to the classifier (e.g., the positive example is misaligned). By contrast, the examples selected by active example selection are representative, and misaligned examples are intended to be rejected.

C. Qualitative Evaluation

We present a qualitative evaluation of the tracking results in this section. Twelve representative sequences are chosen from the subsets of four dominant attributes, i.e., occlusion, illumination variations, background clutter, and deformation. Several screenshots of the tracking results on these 12 sequences are shown in Figs. 7–10. Note that other challenges, e.g., out-of-plane rotation, in-plane rotation, and scale variations, are also included in the 12 sequences. We mainly discuss the four dominant challenges below.

1) Occlusion: Occlusion is one of the most general yet crucial problems in visual tracking. Fig. 7 shows tracking results on three challenging sequences (i.e., *David3*, *Jogging2* and *Woman*) with severe or long-term partial occlusions. In the *David3* sequence, the person suffers from partial occlusion as well as drastic pose variations. The TLD, VTD, Struck, SCM, and SPT methods fail to track the object after the person walks behind a tree (e.g., #85). The MIL, CT, LRSVT, LSST, and ONNDL methods lose the object after the person changes his direction (e.g., #185). Only RET and our method succeed in this sequence. In the *Jogging2* sequence, there is a short-term

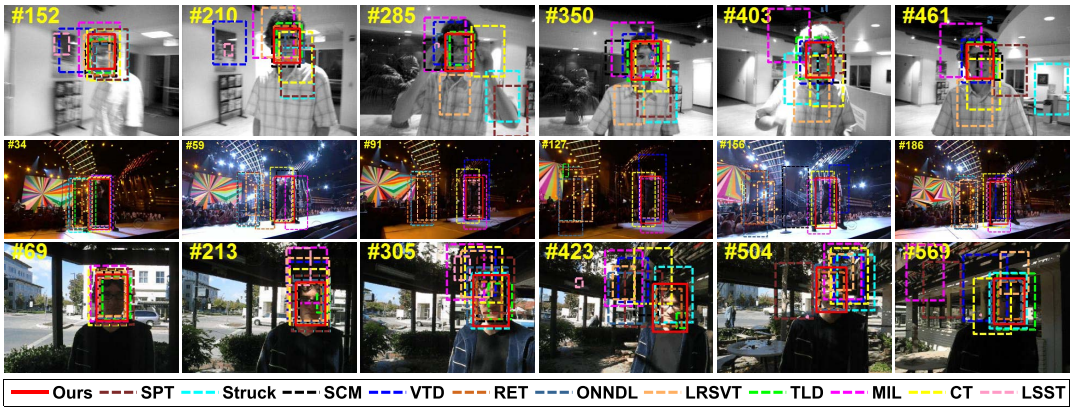


Fig. 8. From top to bottom, representative results of the competing trackers on sequences *David1*, *Singer2*, and *Trellis* (best viewed on high-resolution display). Objects undergo significant illumination variations.

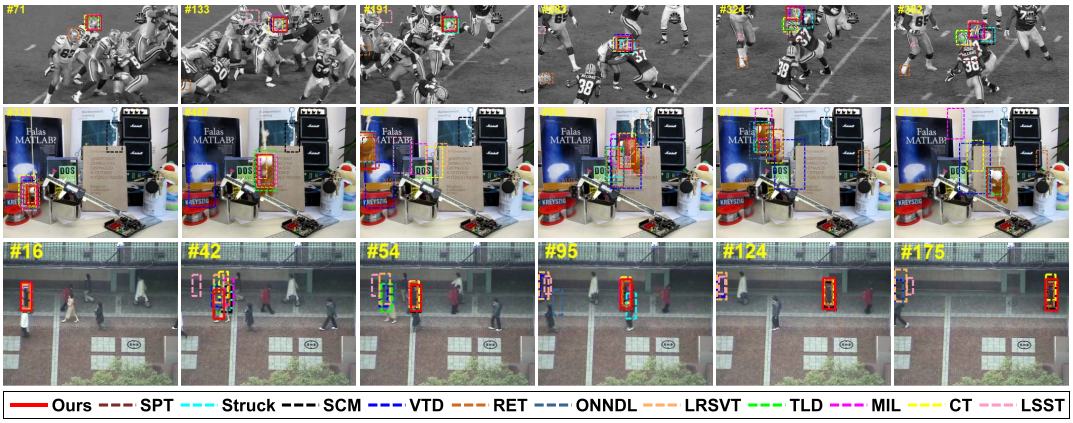


Fig. 9. From top to bottom, representative results of the competing trackers on sequences *Football*, *Lemming*, and *Subway* (best viewed on high-resolution display). Objects appear in background clutters.

complete occlusion for the tracked object (e.g., #57) as well as scale variations (e.g., #274). Most of the trackers lock on the obstacle after occlusion, while the TLD, SCM, and our method are able to reacquire the object and obtain satisfactory trajectories (e.g., #104). In the *Woman* sequence, the object undergoes long-term partial occlusions by cars with similar appearances, which confuse the online update in the TLD, MIL, VTD, and CT methods and cause the drift problem (e.g., #109). The SCM, LRSVT, LSST, and ONNDL methods fail gradually when long-term partial occlusion happens (e.g., #297). Only Struck, SPT, RET, and our method are able to keep the track on the object (e.g., #550). Our method selects informative examples for classifier learning via active example selection, and thus alleviates the drift problem caused by misaligned examples in handling occlusions.

2) Illumination Variations: As shown in Fig. 8, the tracked objects in the *David1*, *Singer2*, and *Trellis* sequences undergo significant illumination changes. In the *David1* sequence, there are drastic pose variations (e.g., #152) of the object in addition to illumination changes (e.g., #403). The VTD, Struck, CT, SPT, and LSST methods lose the object after the person changes his pose (e.g., #210). The MIL, SCM, LRSVT, RET, and ONNDL methods gradually drift away due to continuously illumination changes (e.g., #461). The TLD and our method obtain satisfying tracking results. In the *Singer2* sequence, the contrast between the foreground and the background is very low. The TLD, Struck, SPT, LRSVT, RET, and ONNDL methods fail to track the object at the beginning of the sequence (e.g., #34). The MIL, SCM, and CT methods drift away when there exist drastic illumination changes (e.g., #59), while the VTD and LSST methods perform slightly better. Only our method provides a stable and accurate trajectory in this sequence. In the *Trellis* sequence, the walking man suffers from large-scale illumination changes and the interference from shadows. The TLD, MIL, VTD, CT, SPT, LRSVT, and LSST methods drift away gradually (e.g., #305). In contrast, Struck, SCM, RET, ONNDL, and our method perform better. The robustness of our tracker against illumination variations comes from the fact that the adopted HOG feature is invariant to illumination changes.

3) Background Clutter: Fig. 9 shows tracking results on three challenging sequences (i.e., *Football*, *Lemming*, and *Subway*), where the objects appear in background clutters. In the *Football* sequence, the object undergoes pose variations as well as partial occlusions by the other players which have similar appearances. The LSST and RET methods lock on wrong targets due to the interference of similar appearances (e.g., #133). The TLD, Struck, SCM, LRSVT, CT,

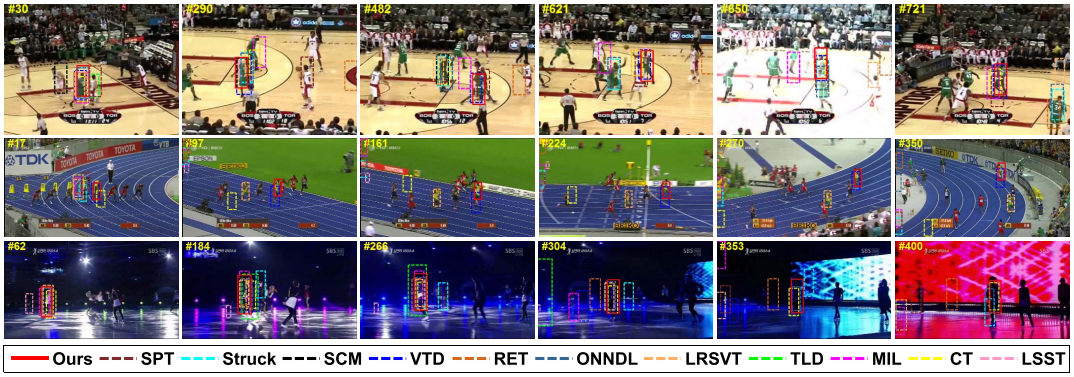


Fig. 10. From top to bottom, representative results of the competing trackers on sequences *Basketball*, *Bolt*, and *Skating1* (best viewed on high-resolution display). Object appearances change drastically due to object deformation, such as viewpoint changes and pose variations.

and SPT methods drift away when the object is occluded by another player (e.g., #324), while MIL, VTD, ONNDL, and our method perform well in this sequence. In the *Lemming* sequence, the SCM method locks on a similar target in the cluttered background at the beginning of the sequence (e.g., #333). The TLD, MIL, CT, LRSVT, RET, LSST, and ONNDL methods gradually drift away when the object changes its pose in the complex environment (e.g., #1145). In contrast, the Struck and SPT methods perform better, and our method shows the most accurate and robust track. In the *Subway* sequence, the TLD, VTD, LRSVT, LSST, and ONNDL methods are influenced by another walking person and fail to track the right object (e.g., #54). The CT, SPT, and RET methods succeed in this sequence but provide a relatively low overlap rate, while MIL, Struck, SCM, and our method achieve more accurate tracking results.

The reasons that our tracker performs well on these three sequences can be explained as follows. Our method learns an online classifier that considers the background information, and thus can achieve robust performance under complex environments. More importantly, an active example selection approach is adopted to select useful examples for classifier learning, which ensures the prediction accuracy of the online classifier during tracking.

4) Object Deformation: Fig. 10 shows tracking results on three challenging sequences (i.e., *Basketball*, *Bolt*, and *Skating1*) to evaluate whether our tracker is able to handle drastic appearance changes caused by nonrigid object deformation, such as viewpoint changes and pose variations. In the *Basketball* sequence, the person changes his pose frequently and often partially occluded by other players. The MIL, Struck, CT, LRSVT, LSST, RET, and ONNDL methods change their track to another player which has a very similar appearance (e.g., #482). In contrast, VTD and our method succeed in tracking the object in the entire sequence. In the *Bolt* sequence, there exist significant pose variations of the person, together with the viewpoint change. The trackers except VTD and our method fail when the viewpoint starts to change (e.g., #17). Our method achieves the best performance in terms of both overlap rate and tracking precision. In the *Skating1* sequence, the dancer continuously changes her pose on a stage with complex background as well as drastic illumination variations. The TLD, VTD, Struck, and LSST

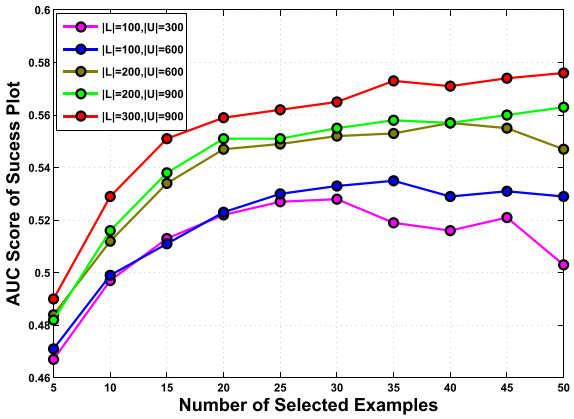


Fig. 11. AUC scores of the success plots of our tracker with different numbers of selected examples and different numbers of labeled and unlabeled examples.

methods gradually drift away when there are severe occlusion and large-scale change of the object (e.g., #184). The MIL, SCM, LRSVT, SPT, and RET methods lose the object as the dancer changes her orientation and appears in a dark background with very low contrast (e.g., #353). The CT method performs slightly better, and ONNDL and our method achieve more stable performance in the entire sequence. We show that our method adaptively copes with appearance variations through online update with the selected informative examples and thus achieves more accurate and consistent tracking results.

D. Parameters Analysis

Our algorithm has three important parameters: the labeled example set capacity $|\mathcal{L}_t|$, the unlabeled example set capacity $|\mathcal{U}_t|$, and the number of selected examples m. In this section, we further study the effect of these three parameters on the tracking performance. Fifty experiments are performed with $m = (5, 10, 15, 20, 25, 30, 35, 40, 45, 50)$ and $\{|\mathcal{L}_t|, |\mathcal{U}_t|\} = (\{100, 300\}, \{100, 600\}, \{200, 600\}, \{200, 900\}, \{300, 900\})$. Each experiment corresponds to a thorough evaluation of the 51 video sequences of the benchmark, and the AUC scores of the success plots with different m, $|\mathcal{L}_t|$, and $|\mathcal{U}_t|$ are shown in Fig. 11. As we can observe in Fig. 11, a larger set of informative examples

improves the tracking performance, and our tracker gets satisfying results when the number of selected examples m is set to 20. If the number of selected examples is too large, the performance degrades due to frequent updates of the classifier. In addition, the enlargement of the labeled example set apparently benefits the tracking process (e.g., the curve $\{|\mathcal{L}_t| = 100, |\mathcal{U}_t| = 600\}$ versus the curve $\{|\mathcal{L}_t| = 200, |\mathcal{U}_t| = 600\}$), while the improvement brought by a larger unlabeled example set is relatively small (e.g., the curve $\{|\mathcal{L}_t| = 200, |\mathcal{U}_t| = 600\}$ versus the curve $\{|\mathcal{L}_t| = 200, |\mathcal{U}_t| = 900\}$). Since the computation time of our tracker largely depends on the number of training examples, we set $|\mathcal{L}_t| = 200$ and $|\mathcal{U}_t| = 600$ to achieve a proper tradeoff between effectiveness and efficiency.

VI. Conclusion

In this paper, we have presented a novel online discriminative tracking framework that explicitly couples the objectives of training example collection and classifier learning in a principled manner. We have shown that selecting informative examples for classifier learning results in more robust tracking, and have proposed an active example selection approach using the formalism of active learning. We have also shown that assigning labels to part of the selected examples achieves a suitable tradeoff between stability and plasticity in terms of capturing appearance variations. The online classifier leaned by LapRLS using the automatically selected examples can not only utilize the discriminative information contained in the abundant unlabeled data, but also alleviate the drift problem caused by label noise. Both quantitative and qualitative evaluations compared with eleven state-of-the-art trackers on a comprehensive benchmark demonstrate the effectiveness and robustness of our tracker.

References

[1] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 125–141, May 2008.

[2] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 1269–1276.

[3] X. Mei and H. Ling, "Robust visual tracking using $\ell 1$ minimization," in *Proc. IEEE 12th Int. Conf. Comput. Vis. (ICCV)*, Sep./Oct. 2009, pp. 1436–1443.

[4] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparse collaborative appearance model," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2356–2368, May 2014.

[5] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1822–1829.

[6] Y. Wu, B. Ma, M. Yang, J. Zhang, and Y. Jia, "Metric learning based structural appearance model for robust visual tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 5, pp. 865–877, May 2014.

[7] N. Wang, J. Wang, and D.-Y. Yeung, "Online robust non-negative dictionary learning for visual tracking," in *Proc. IEEE Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 657–664.

[8] D. Wang, H. Lu, and M.-H. Yang, "Least soft-threshold squares tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 2371–2378.

[9] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, "PROST: Parallel robust online simple tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 723–730.

[10] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, Aug. 2011.

[11] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *Proc. IEEE Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 263–270.

[12] K. Zhang, L. Zhang, and M.-H. Yang, "Fast compressive tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 10, pp. 2002–2015, Oct. 2014.

[13] R. Yao, Q. Shi, C. Shen, Y. Zhang, and A. van den Hengel, "Part-based visual tracking with online latent structural learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 2363–2370.

[14] L. Zhang and L. J. P. van der Maaten, "Preserving structure in model-free tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 756–769, Apr. 2014.

[15] Q. Bai, Z. Wu, S. Sclaroff, M. Betke, and C. Monnier, "Randomized ensemble tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 2040–2047.

[16] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2008, pp. 234–247.

[17] A. Saffari, C. Leistner, M. Godec, and H. Bischof, "Robust multi-view boosting with priors," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2010, pp. 776–789.

[18] Y. Bai and M. Tang, "Robust tracking via weakly supervised ranking SVM," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1854–1861.

[19] J. Gao, J. Xing, W. Hu, and S. Maybank, "Discriminant tracking using tensor representation with semi-supervised improvement," in *Proc. IEEE Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 1569–1576.

[20] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: Bootstrapping binary classifiers by structural constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 49–56.

[21] J. S. Supancic, III, and D. Ramanan, "Self-paced learning for long-term tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 2379–2386.

[22] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, Nov. 2014.

[23] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *J. Artif. Intell. Res.*, vol. 4, pp. 129–145, Mar. 1996.

[24] A. C. Atkinson and A. N. Donev, *Optimum Experimental Designs*. London, U.K.: Oxford Univ. Press, 2002.

[25] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. Van Den Hengel, "A survey of appearance models in visual object tracking," *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 4, p. 58, Sep. 2013.

[26] K. Yu, J. Bi, and V. Tresp, "Active learning via transductive experimental design," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, 2006, pp. 1081–1088.

[27] X. He, W. Min, D. Cai, and K. Zhou, "Laplacian optimal design for image retrieval," in *Proc. 30th Annu. ACM SIGIR Conf.*, 2007, pp. 119–126.

[28] X. He, "Laplacian regularized D-optimal design for active learning and its application to image retrieval," *IEEE Trans. Image Process.*, vol. 19, no. 1, pp. 254–263, Jan. 2010.

[29] S. Stalder, H. Grabner, and L. Van Gool, "Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Sep./Oct. 2009, pp. 1409–1416.

[30] S. Gu, Y. Zheng, and C. Tomasi, "Efficient visual object tracking with online nearest neighbor classifier," in *Proc. Asian Conf. Comput. Vis. Workshops (ACCV)*, 2011, pp. 271–282.

[31] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2004, pp. 1601–1608.

[32] M. Isard and A. Blake, "CONDENSATION—Conditional density propagation for visual tracking," *Int. J. Comput. Vis.*, vol. 29, no. 1, pp. 5–28, Aug. 1998.

[33] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 886–893.

[34] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 2411–2418.

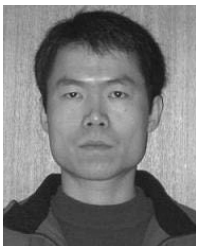
Min Yang received the B.S. degree from Beijing Institute of Technology, Beijjng, China, in 2010, where he is currently working toward the Ph.D. degree with the Beijing Laboratory of Intelligent Information Technology, School of Computer Science, under the supervision of Prof. Y. Jia.

His research interests include computer vision, pattern recognition, and machine learning.

Yuwei Wu received the Ph.D. degree in computer science from Beijing Institute of Technology, Beijing, China, in 2014.

He is a Research Fellow with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His research interests include computer vision, medical image processing, and object tracking.

Dr. Wu received the National Scholarship for Graduate Students and Academic Scholarship for Ph.D. candidates from the Ministry of Education, China, the Outstanding Ph.D. Thesis Award, the Xu Teli Excellent Scholarship from Beijing Institute of Technology, and the China Aerospace Science and Technology Corporation Scholarship from China Aerospace Science and Industry Corporation.



Mingtao Pei received the Ph.D. degree in computer science from Beijing Institute of Technology, Beijing, China, in 2004.

He was an Associate Professor with the School of Computer Science, Beijing Institute of Technology. He was a Visiting Scholar with the Center of Image and Vision Science, University of California at Los Angeles, Los Angeles, CA, USA, from 2009 to 2011. His research interests include computer vision with an emphasis on event recognition and machine learning.

Dr. Pei is a member of the China Computer Federation.

Bo Ma received the Ph.D. degree in computer science from Harbin Institute of Technology, Harbin, China, in 2003.

He was with the Department of Computer Science, The City University of Hong Kong, Hong Kong, from 2004 to 2006, where he was involved with research projects in computer vision and pattern recognition. He joined the Department of Computer Science, Beijing Institute of Technology, Beijing, China, in 2006, where he is currently an Associate Professor. His research interests include statistical pattern recognition, image object tracking, and information fusion.

Yunde Jia (M'11) received the B.S., M.S., and Ph.D. degrees in mechatronics from Beijing Institute of Technology (BIT), Beijing, China, in 1983, 1986, and 2000, respectively.

He was a Visiting Scientist with Carnegie Mellon University, Pittsburgh, PA, USA, from 1995 to 1997, and a Visiting Fellow with Australian National University, Canberra, ACT, Australia, in 2011. He is currently a Professor of Computer Science at BIT and is the Director of the Beijing Laboratory of Intelligent Information Technology. His research interests include computer vision, media computing, and intelligent systems.