Landmark-based Inductive Model for Robust Discriminative Tracking

Yuwei Wu, Mingtao Pei, Min Yang, Yang He, and Yunde Jia

Beijing Laboratory of Intelligent Information Technology,
School of Computer Science, Beijing Institute of Technology,
Beijing 10081, P.R. China
http://iitlab.bit.edu.cn/mcislab/~wuyuwei/

Abstract. The appearance of an object could be continuously changing during tracking, thereby being not independent identically distributed. A good discriminative tracker often needs a large number of training samples to fit the underlying data distribution, which is impractical for visual tracking. In this paper, we present a new discriminative tracker via the landmark-based inductive model (Lim) that is non-parametric and makes no specific assumption about the sample distribution. With an undirected graph representation of samples, the Lim locally approximates the soft label of each sample by a linear combination of labels on its nearby landmarks. It is able to effectively propagate a limited amount of initial labels to a large amount of unlabeled samples. To this end, we introduce a local landmarks approximation method to compute the cross-similarity matrix between the whole data and landmarks. And a soft label prediction function incorporating the graph Laplacian regularizer is used to diffuse the known labels to all the unlabeled vertices in the graph, which explicitly considers the local geometrical structure of all samples. Tracking is then carried out within a Bayesian inference framework where the soft label prediction value is used to construct the observation model. Both qualitative and quantitative evaluations on 65 challenging image sequences including the benchmark dataset and other public sequences demonstrate that the proposed algorithm outperforms the state-of-the-art methods.

1 Introduction

An appearance model is one of the most critical prerequisites for successful visual tracking. Designing an effective appearance model is still a challenging task due to appearance variations caused by background clutter, object deformation, partial occlusions, and illumination changes, *etc.* Numerous tracking algorithms have been proposed to address this issue [1], and existing tracking algorithms can be roughly categorized as either generative [2–7] or discriminative [8–15] approaches. Generative methods build an object representation, and then search for the region most similar to the object. However, generative models do not take into account background information. Discriminative methods train an online binary classifier to adaptively separate the object from the background, which are more robust against appearance variations of an object. In this paper, we focus on the discriminative tracking method.

In visual tracking applications, the samples obtained by the tracker are drawn from an unknown underlying data distribution. The appearance of an object could be continuously changing and thus it is impossible to be independent and identically distributed (*i.i.d*). A good discriminative tracker often needs a large number of labeled samples to adequately fit the real data distribution [16]. This is because if the dimensionality of the data is large compared to the number of the samples, then many statistical learning methods predict overfitting due to the "curse of dimensionality". However, precisely labeled samples only come from the first frame during tracking, *i.e.*, the number of labeled samples is very small. To acquire more labeled samples, in most existing discriminative tracking approaches, the current tracking result is used to extract positive samples and the surrounding regions are used to extract negative samples. Once the tracker location is not precise, the assigned labels may be noisy. Over time the accumulation of errors can degrade the classifier and cause drift. This situation makes us wonder: *with a very small number of labeled samples, whether we can design a new discriminative tracker which makes no specific assumption about the sample distribution.*

In this paper, we take full advantage of the geometric structure of the data and thus present a new discriminative tracking approach with the landmark-based inductive model (Lim). The Lim locally approximates the soft label of each sample by a linear combination of labels on its nearby landmarks. It is able to effectively propagate a limited amount of initial labels to a large amount of unlabeled samples, matching the needs of discriminative trackers. Under the graph representation of samples, the local landmarks approximation is employed to design a sparse and nonnegative adjacency matrix characterizing relationship among all samples. Based on the Nesterov's gradient projection algorithm, an efficient numerical algorithm is developed to solve the problem of the local landmarks approximation with guaranteed quadratic convergence. Furthermore, the object function of the label prediction provides a promising paradigm for modeling the geometrical structures of samples via Laplacian regularizer. Preserving the local manifold structure of samples can make our tracker have more discriminating power to handle appearance changes.

Specifically, the proposed method treats both labeled and unlabeled samples as vertices in a graph and builds edges which are weighted by the affinities (similarities) between the corresponding sample pairs. For each new frame, candidates predicted by the particle filter are considered as unlabeled samples and utilized to constitute a new graph representation together with the collected samples stored in the sample pool. A small number of landmarks obtained from the entire sample space enable nonparametric regression that calculates the soft label of each sample as a locally weighted average of labels on landmarks. Tracking is carried out within a Bayesian inference framework where the soft label prediction value is used to construct the observation model. A candidate with the highest classification score is considered as the tracking result. To alleviate the drift problem, once the tracked object is located, the labels of the newly collected samples are assigned according to the classification score of the current tracking results, in which no self-labeling is involved. The proposed tracker adapts to drastic appearance variations, as validated in our experiments.

1.1 Related work

Discriminative tracking has received wide attention for its adaptive ability to handle appearance changes. The essential component of discriminative trackers is the classifier updating. Straightforward appearance update with newly obtained results could result in incorrectly labeled training samples and degrade the models gradually with drifts. Grabner *et al.* [9] employed an online semi-supervised learning framework to train a classifier which is less susceptible to drift but not adaptive enough to handle fast appearance changes. Babenko *et al.* [11] integrated multiple instance learning (MIL) into online boosting algorithm to alleviate the drift problem. In the MIL tracking, the classifier is updated with positive and negative bags rather than individual labeled examples. Kalal *et al.* [13] developed a semi-supervised learning approach (*i.e.*, P-N learning) to train a binary classifier with structured unlabeled data. Zhang and Maaten [17] developed a structure-preserving object tracker that learns spatial constraints between objects using an online structured SVM algorithm to improve the performance of single-object or multi-object tracking. Wu *et al.* [18] addressed visual tracking by learning a suitable metric matrix in the feature space of local sparse codes to effectively capture appearance variations.

Different from the schemes of the classifier updating in [9, 11, 13, 18], in which candidates are not used to train the classifier, and therefore the class labels of them are assigned by the previous classifier. In our tracker, for each new frame, candidates are considered as unlabeled samples and utilized to constitute a new graph representation to update the current classifier. Explicitly taking into account the local manifold structure of labeled and unlabeled samples, we introduce a soft label propagation method defined over the graph, which has more discriminating power. In addition, once the tracked object is located, the discriminative appearance models are online updated in the manner of both supervised and unsupervised which makes our tracker more stable and adaptive to appearance changes. More details are discussed in Sect. 3.

Recently, researchers utilize the graph-based discriminative learning to construct the object appearance model for visual tracking. With the 2^{nd}-order tensor representation, Gao *et al.* [19] designed two graphs for characterizing the intrinsic local geometrical structure of the tensor space. Based on the least square support vector machine, Li *et al.* [20] exploited a hypergraph propagation method to capture the contextual information on samples, which further improves the tracking accuracy. Kumar and Vleeschouwer [21] constructed a number of distinct graphs (*i.e.*, spatiotemporal, appearance and exclusion) to capture the spatio-temporal and the appearance information. Then, they formulated the multi-object tracking as a consistent labeling problem in the associated graphs.

Our method differs from [19, 20] both in the graph construction and the label propagation method. Methods in [19, 20] construct the graph representation using kNN whose computational cost is expensive. In contrast, employing local landmarks approximation, we design a new form of the adjacency matrix characterizing relationship between all samples. The total time complexity scales linearly with the number of samples. More importantly, our method is an inductive model which can be used to infer the labels of unseen data (*i.e.*, candidates). Only a few samples are selected and used to learn a new discriminative model. The label of each sample can be interpreted as the weighted combination of the labels on landmarks. Graph Laplacian is incorporated into the object function of inductive learning as a regularizer to preserve the local geometrical structure of samples.

2 Landmark-based inductive model

2.1 Problem description

Suppose that we have l labeled samples $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{l}$ and u unlabeled samples $\{\boldsymbol{x}_i\}_{i=l+1}^{l+u}$, where $\boldsymbol{x}_i \in \mathbb{R}^d$, and $\boldsymbol{y}_i \in \mathbb{R}^c$ is the label vector. Denote $\boldsymbol{X} = \{x_1, x_2, \cdots, x_n\} \in \mathbb{R}^{d \times n}$ and $\boldsymbol{Y}_l = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_l\} \in \mathbb{R}^{l \times c}$, where $n = l + u$. If $\boldsymbol{x}_i$ belongs to the kth class ($1 \leq k \leq c$), the kth entry in $\boldsymbol{y}_i$ is 1 and all the other entries are 0's. In this paper, the data $\boldsymbol{X}$ is represented by the undirected graph $\mathcal{G} = \{\boldsymbol{X}, \boldsymbol{E}\}$, where the set of vertices is $\boldsymbol{X} = \{\boldsymbol{x}_i\}$ and the set of edges is $\boldsymbol{E} = \{e_{ij}\}$, where e_{ij} denotes the similarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Define a soft label prediction (*i.e.*, classification) function $f : \mathbb{R}^d \to \mathbb{R}^c$. A crucial component of our method is the estimation of a weighted graph $\mathcal{G}$ from $\boldsymbol{X}$. Then, the soft label of any sample can be inferred using $\mathcal{G}$ and known labels Y_l.

The time complexity of traditional graph-based semi-supervised learning methods is usually $O(n^3)$ with respect to the data size n, because $n \times n$ kernel matrix (*e.g.,* multiplication or inverse) is calculated in inferring the label prediction. Full-size label prediction is infeasible when n is large, the work of [22] inspired us to exploit the idea of landmark samples. To accomplish the soft label prediction, we employ an economical and practical prediction function expressed as

$$f(\boldsymbol{x}) = \sum_{k=1}^{m} K(\boldsymbol{x}, \boldsymbol{d}_k)\boldsymbol{a}_k. \tag{1}$$

The idea of this formulation is that the label of each sample can be interpreted as the locally weighted average of variables $\boldsymbol{a}_k$'s defined on m landmarks [22, 23]. As a trade-off between computational efficiency and effectiveness, in this paper, k-means algorithm is used to select the centers as the set of landmarks $\boldsymbol{D} = \{\boldsymbol{d}_k\}_{k=1}^{m} \in \mathbb{R}^{d \times m}$.

Eq. (1) is deemed as a inductive model, because it can diffuse the label of landmarks to all unlabeled samples, as discussed in Sect. 2.4. The above model can be written in a matrix form

$$\boldsymbol{f} = \boldsymbol{H}\boldsymbol{A}, \tag{2}$$

where $\boldsymbol{f} = [f(\boldsymbol{x}_1), f(\boldsymbol{x}_2), \cdots, f(\boldsymbol{x}_n)]^\top \in \mathbb{R}^{n\times c}$ is the landmark-based label prediction function on all samples. $\boldsymbol{A} = [f(\boldsymbol{d}_1), f(\boldsymbol{d}_2), \cdots, f(\boldsymbol{d}_m)]^\top = [\boldsymbol{A}_1, \boldsymbol{A}_2, \cdots, \boldsymbol{A}_c] \in \mathbb{R}^{m\times c}$ denotes the label of landmarks $\boldsymbol{d}_k$'s. $\boldsymbol{H} \in \mathbb{R}^{n\times m}$ is the cross-similarity matrix between the whole data $\boldsymbol{X}$ and landmarks $\boldsymbol{d}_k$,

$$\boldsymbol{H}_{ik} = K(\boldsymbol{x}_i, \boldsymbol{d}_k) > 0, 1 \leq i \leq n, 1 \leq k \leq m.$$

In what follows, we will elaborate how to effectively solve $\boldsymbol{A}$ and $\boldsymbol{H}$.

2.2 Solving optimal $\boldsymbol{H}$

Typically, we may employ Gaussian kernel or Epanechnikov quadratic kernel to compute $\boldsymbol{H}$. However, how to choose appropriate kernel bandwidths is difficult. Instead of adopting the predefined kernel, we learn an optimal $\boldsymbol{H}$ by considering the geometric structure information between labeled and unlabeled samples. We reconstruct $\boldsymbol{x}_i$ as a combination of its s closest landmarks in the feature space. In this work, we set $s = 10$. Similar to locality-constrained linear coding (LLC) [24], a local landmarks approximation method is proposed to optimize the coefficient vector $\boldsymbol{h}_i \in \mathbb{R}^s$:

$$\begin{aligned} \min_{\boldsymbol{h}_i\in\mathbb{R}^s} g(\boldsymbol{h}_i) = \frac{1}{2}\left\| \boldsymbol{x}_i - \sum_{j=1}^{s} \boldsymbol{d}_j \boldsymbol{h}_i \right\|^2, \\ s.t. \quad \mathbf{1}^\top \boldsymbol{h}_i = 1,\ \boldsymbol{h}_i \geq 0 \end{aligned} \tag{3}$$

where s entries of the vector $\boldsymbol{h}_i$ correspond to s coefficients contributed by s nearest landmarks. The constraint $\mathbf{1}^\top \boldsymbol{h}_i = 1$ follows the shift-invariant requirements. The main difference between LLC and our method is that we incorporate inequality constraints (*i.e.*, non-negative constraints) into the object function as we require the similarity measure to be a positive value. Therefore we need to develop a different optimization algorithm to solve Eq. (3). In this section, Nesterov's gradient projection (NGP) method [25], a first-order optimization procedure, is employed to solve the constrained optimization problem Eq. (3). A key step of NGP is how to efficiently project a vector $\boldsymbol{h}_i$ onto the corresponding constraint set C.

Denote $\mathcal{Q}_{\beta,\boldsymbol{v}}(\boldsymbol{h}_i) = g(\boldsymbol{v}) + \nabla g(\boldsymbol{v})^\top(\boldsymbol{h}_i - \boldsymbol{v}) + \frac{\beta}{2}\|\boldsymbol{h}_i - \boldsymbol{v}\|_2^2$, as the first-order Taylor expansion of $g(\boldsymbol{h}_i)$ at $\boldsymbol{v}$ with the squared Euclidean distance between $\boldsymbol{h}_i$ and $\boldsymbol{v}$ as a regularization term. Here $\nabla g(\boldsymbol{v})$ is the gradient of $g(\boldsymbol{h}_i)$ at $\boldsymbol{v}$. We can easily obtain

$$\min_{\boldsymbol{h}_i\in C} \mathcal{Q}_{\beta,\boldsymbol{v}}(\boldsymbol{h}_i) = \mathit{\Pi}_C\Big(\boldsymbol{v} - \frac{1}{\beta}\nabla g(\boldsymbol{v})\Big), \tag{4}$$

where $\mathit{\Pi}_C(\boldsymbol{v}) = \min_{\boldsymbol{v}'\in C}\|\boldsymbol{v} - \boldsymbol{v}'\|_2^2$ is the Euclidean projection of $\boldsymbol{v}$ onto C [26]. The projection operator $\mathit{\Pi}_C(\cdot)$ has been implemented efficiently in $O(s\log s)$.

From Eq. (4), the solution of Eq. (3) can be obtained by generating a sequence $\{\boldsymbol{h}_i^{(t)}\}$ at $\boldsymbol{v}^{(t)} = \boldsymbol{h}_i^{(t)} + \alpha_t\big(\boldsymbol{h}_i^{(t)} - \boldsymbol{h}_i^{(t-1)}\big)$, *i.e.*,

$$\boldsymbol{h}_i^{(t+1)} = \mathit{\Pi}_C\Big(\boldsymbol{v}^{(t)} - \frac{1}{\beta_t}\nabla g(\boldsymbol{v}^{(t)})\Big) = \min_{\boldsymbol{h}_i\in C} \mathcal{Q}_{\beta_t,\boldsymbol{v}^{(t)}}(\boldsymbol{h}_i). \tag{5}$$

In NGP, choosing proper parameters β_t and α_t is also significant for the convergence property. Similar to [25], we set $\alpha_t = (\delta_{t-1} - 1)/\delta_t$ with $\delta_t = \big(1 + \sqrt{1 + 4\delta_{t-1}^2}\big)/2$, $\delta_0 = 0$ and $\delta_1 = 1$. β_t is selected by finding the smallest nonnegative integer j such that $g(\boldsymbol{h}_i) \leq \mathcal{Q}_{\beta_t, \boldsymbol{v}^{(t)}}(\boldsymbol{h}_i)$ with $\beta_t = 2^j \beta_{t-1}$.

After getting the optimal weight vector $\boldsymbol{h}_i$, we set $\boldsymbol{H}_{ij'} = \boldsymbol{h}_i$, where j' is the indices corresponding to the s nearest landmarks and the cardinality $|j'| = s$. For the rest entries of $\boldsymbol{H}_i$, we set 0's. Apparently, $\boldsymbol{H}_{ij} = 0$ when landmark $\boldsymbol{d}_j$ is far away from $\boldsymbol{x}_i$ and $\boldsymbol{H}_{ij} \neq 0$ is only for the s closest landmarks of $\boldsymbol{x}_i$. In contrast to weights defined by kernel function (*e.g.*, Gaussian kernel), the local landmarks approximation method is able to provides optimized and sparser weights, as validated in our experiments.

2.3 Solving label prediction matrix $\boldsymbol{A}$

Note that the adjacency matrix $\boldsymbol{W} \in \mathbb{R}^{n \times n}$ between all samples encountered in practice usually have low numerical-rank compared with the matrix size [27]. We consider *whether we can construct a nonnegative and empirically sparse graph adjacency matrix $\boldsymbol{W}$ with the nonnegative and sparse $\boldsymbol{H} \in \mathbb{R}^{n \times m}$ introduced in Sect. 2.2.* Intuitively, we can design the adjacency matrix $\boldsymbol{W}$ to be a low-rank form

$$\boldsymbol{W} = \boldsymbol{H}\boldsymbol{H}^\top, \tag{6}$$

where the inner product is regarded as the metric to measure the adjacent weight between samples. Eq. (6) implies that if two samples are correlative (*i.e.*, $\boldsymbol{W}_{ij} > 0$), they share at least one landmark, otherwise $\boldsymbol{W}_{ij} = 0$. $\boldsymbol{W}$ defined in Eq. (6) naturally preserves some good properties (*e.g.*, sparseness and nonnegativeness).

To compute the label prediction matrix $\boldsymbol{A}$, we exploit the following optimization framework [22]:

$$\min L(\boldsymbol{f}_l, \boldsymbol{y}_l) + \eta \|\boldsymbol{f}\|_{\mathcal{G}}. \tag{7}$$

Here $L(\cdot,\cdot)$ is an empirical loss function, which requires that the prediction $\boldsymbol{f}$ should be consistent with the known class labels. η is a positive regularization parameter. $\boldsymbol{f}_l \in \mathbb{R}^{l \times c}$ is the sub-matrix corresponding to the labeled samples in $\boldsymbol{f} \in \mathbb{R}^{n \times c}$. Discriminative models take tracking as a binary classification task to separate the object from its surrounding background. In this case, $c = 2$. $\|f\|_{\mathcal{G}} = tr(\boldsymbol{f}^\top \boldsymbol{L} \boldsymbol{f})$ enforces the smoothness of $\boldsymbol{f}$ with regard to the manifold structure of the graph, where $\boldsymbol{L} \in \mathbb{R}^{n \times n}$ is the graph-based regularization matrix. Usually $\boldsymbol{L} = \boldsymbol{\Sigma} - \boldsymbol{W}$, where $\boldsymbol{\Sigma} = diag(\boldsymbol{W}\boldsymbol{1})$ is the vertex degree matrix of $\mathcal{G}$.

With the design of $\boldsymbol{W}$, Laplacian graph regularization can be approximated as

$$\boldsymbol{f}^\top \boldsymbol{L} \boldsymbol{f} = \boldsymbol{f}^\top (diag(\boldsymbol{H}\boldsymbol{H}^\top \boldsymbol{1}) - \boldsymbol{H}\boldsymbol{H}^\top)\boldsymbol{f}, \tag{8}$$

where nonnegative $\boldsymbol{W}$ guarantees the positive semi-definite (PSD) property of $\boldsymbol{L}$. Keeping PSD $\boldsymbol{L}$ is important to ensure that the graph regularizer $\boldsymbol{f}^\top \boldsymbol{L} \boldsymbol{f}$ is convex.

By plugging $\boldsymbol{f} = \boldsymbol{H}\boldsymbol{A}$ into Eq. (7) and choosing the loss function $L(\cdot,\cdot)$ as the $L2$-norm, the convex differentiable object function for solving label prediction matrix $\boldsymbol{A}$ can be formulated as

$$\min_{\boldsymbol{A}} \mathcal{L}(\boldsymbol{A}) = \frac{\eta}{2} tr\big((\boldsymbol{H}\boldsymbol{A})^\top \boldsymbol{L}(\boldsymbol{H}\boldsymbol{A})\big) + \|\boldsymbol{H}_l\boldsymbol{A} - \mathbf{Y}_l\|_F^2. \tag{9}$$

Here, $\boldsymbol{H}_l \in \mathbb{R}^{l\times m}$ is the rows in $\boldsymbol{H}$ that corresponds to the labeled samples, and $\boldsymbol{L}$ is defined in Eq. (8). By setting the derivative w.r.t. $\boldsymbol{A}$ to zero, we easily obtain the globally optimal solution to Eq. (9):

$$\boldsymbol{A}^* = \big(\boldsymbol{H}_l^\top \boldsymbol{H}_l + \eta \boldsymbol{H}^\top \boldsymbol{L}\boldsymbol{H}\big)^{-1} \boldsymbol{H}_l^\top \mathbf{Y}_l. \tag{10}$$

2.4 Soft label propagation

Through applying the inductive model Eq. (2), we are able to predict the soft label for any sample $\boldsymbol{x}_i$ (unlabeled training samples or novel test samples) as

$$\widehat{f}(\boldsymbol{x}_i) = \max_{k\in\{1,2\}} \frac{\boldsymbol{H}(\boldsymbol{x}_i)\ \boldsymbol{A}_k}{\mathbf{1}^\top \boldsymbol{H}\boldsymbol{A}_k}, \tag{11}$$

where $\{\boldsymbol{A}_k\}_{k=1}^c \in \mathbb{R}^{m\times 1}$ is the column vector of $\boldsymbol{A}$, and $\boldsymbol{H}(\boldsymbol{x}_i) \in \mathbb{R}^{1\times m}$ represents the weight between $\boldsymbol{x}$ and landmarks $\boldsymbol{d}_k$'s. Specifically, if $\boldsymbol{x}_i$ belongs to unlabeled training samples, $\boldsymbol{H}(\boldsymbol{x}_i) = \boldsymbol{H}_i$ where $\boldsymbol{H}_i$ denotes the i-th row of $\boldsymbol{H}$, $i = l+1, \cdots, n$. If $\boldsymbol{x}_i$ is a novel test sample, we need to compute the vector $\boldsymbol{H}_i$ as $\boldsymbol{H}(\boldsymbol{x}_i)$ described in Sect. 2.2, then update $\boldsymbol{H} \in \mathbb{R}^{(n+1)\times m}$, *i.e.*, $\boldsymbol{H} \leftarrow [\boldsymbol{H}; \boldsymbol{H}_i]$. After deriving the soft label prediction (*i.e.*, classification) of each sample, the classification score can be utilized as the similarity measure for tracking. In the next section, we will elaborate the application of the proposed landmark-based inductive model in tracking.

3 Lim tracker

In our tracking framework, the object is represented by five different image patches inside the object region. These five image patches correspond to the five parts of an object, respectively, as exemplified in Fig. 1. Therefore, image patches corresponding to the certain part of all samples are able to construct a sub-sample set $\boldsymbol{X}^{(\tau)}$, $\tau = 1, 2, \cdots, 5$. Each sub-sample set $\boldsymbol{X}^{(\tau)}$ is used to train a single classifier $\boldsymbol{f}^{(\tau)}$ using the inductive model predefined in Eq. (2). The final tracking result can be determined by the sum of the classification scores of the five image patches inside the object region:

$$SC = \sum_{\tau=1}^{5} \omega_\tau \boldsymbol{f}^{(\tau)}, \tag{12}$$

where ω_τ is the weight of τ-th image patch ($\sum_{\tau=1}^{5} \omega_\tau = 1$ and $\omega_\tau = 0.2$ in the experiments). This part-based scheme could potentially alleviate the drift caused by partial occlusions.

To initialize the classifier in the first frame, we draw positive and negative samples around the target location. Suppose the target is labeled manually, perturbation (e.g., shifting 1 or 2 pixels) around the object is performed for collecting N_p positive samples $\boldsymbol{X}_{N_p}$. Similarly, N_n negative samples $\boldsymbol{X}_{N_n}$ are

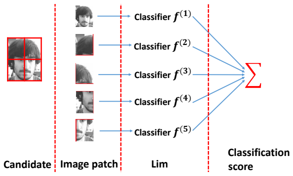

Fig. 1. Object representation using five different image patches. The candidate is normalized to the same size (24 × 24 in our experiment), each image patch is with 12 × 12.

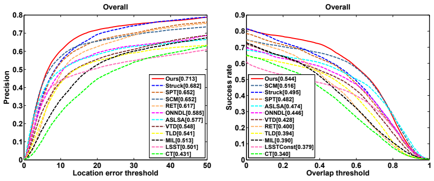

Fig. 2. Overall performance comparisons of precision plot and success rate. The performance s-core for each tracker is shown in the legend (best viewed on high-resolution display).

collected far away from the located object (e.g., within an annular region a few pixels away from the object). $\boldsymbol{X_1} = \boldsymbol{X}_{N_p} \bigcup \boldsymbol{X}_{N_n}$ is the initialized labeled sample set. K-means algorithm is exploited to select the centers as the set of landmarks $\boldsymbol{D}$. Using labeled samples and landmarks, we can train a prior classifier via the Lim.

For each new frame, candidates predicted by the particle filter are considered as unlabeled samples $\widehat{\boldsymbol{X}}$. According to Eq. (11), we can get the classification score of each candidate. A candidate with higher classification score indicates that it is more likely to be generated from the target class. The most likely candidate is considered as the tracking result for this frame. Then, perturbation (*i.e.*, the same scheme in the first frame) around the tracking result is performed for collecting sample set $\boldsymbol{X}_C$. If the classification score of the located object is higher than the predefined threshold ϵ (*i.e.*, the current tracking result is reliable), samples in $\boldsymbol{X}_C$ are regarded as labeled ones, otherwise regarded as unlabeled ones. That is, samples are collected in the manner of both supervised and unsupervised, and thus the stability and adaptivity in tracking objects of changing appearance are preserved.

3.1 Update the classifier

We construct a *sample pool* $\boldsymbol{X}_P$ and a *sample buffer pool* $\boldsymbol{X}'$. We only keep T collected sample set $\boldsymbol{X}_C$ to constitute the sample buffer pool. Every T frames, $\boldsymbol{X}'$ is utilized to update $\boldsymbol{X}_P$. After updating the sample pool, we will leave $\boldsymbol{X}'$ blank and then reconfigure it. In our experiment, we set the sample pool capacity as $\Theta(\boldsymbol{X}_P)$ which denotes the number of samples in the sample pool. If the total number of samples in the sample pool is larger than $\Theta(\boldsymbol{X}_P)$, samples in $\boldsymbol{X}_P$ will be randomly replaced with $\boldsymbol{X}'$. To reduce the risk of visual drift, we always retain the samples $\boldsymbol{X}_1$ obtained from the first frame in the sample pool. In other words, $\boldsymbol{X}_P = [\boldsymbol{X}_1; \boldsymbol{X}']$. Similarly, landmarks also should be updated using the sample pool every T frames. Specifically, we first implement k-means in the current sample pool $\boldsymbol{X}_P$ to obtain a new landmarks set. Then, the updated landmarks set is gained by carrying out the k-means algorithm again using the new landmarks set and the previous landmarks set, which is able to better characterize the samples distribution.

3.2 Bayesian state inference

Object tracking can be considered as a Bayesian inference task in a Markov model with hidden state variables. Given the observation set of the object $\mathcal{O}_{1:t} = \{\boldsymbol{o_1}, \boldsymbol{o_2}, \cdots, \boldsymbol{o_t}\}$, the optimal state $\boldsymbol{s_t}$ of the tracked object is obtained by the maximum a posteriori estimation $p\big(\boldsymbol{s}_t^i \big| \mathcal{O}_{1:t}\big)$, where $\boldsymbol{s}_t^i$ indicates the state of the i-th sample. The posterior probability $p\big(\boldsymbol{s}_t \big| \mathcal{O}_{1:t}\big)$ is formulated by Bayes theorem as $p\big(\boldsymbol{s}_t \big| \mathcal{O}_{1:t}\big) \propto p(\boldsymbol{o}_t|\boldsymbol{s}_t) \int p\big(\boldsymbol{s}_t|\boldsymbol{s}_{t-1}\big) p\big(\boldsymbol{s}_{t-1} \big| \mathcal{O}_{1:t-1}\big)\ d\boldsymbol{s}_{t-1}$. This inference is governed by the dynamic model $p\big(\boldsymbol{s}_t|\boldsymbol{s}_{t-1}\big)$ which models the temporal correlation of the tracking results in consecutive frames, and by the observation model $p(\boldsymbol{o}_t|\boldsymbol{s}_t)$ which estimates the likelihood of observing $\boldsymbol{o_t}$ at state $\boldsymbol{s_t}$.

We apply an affine image warp to model the object motion between two consecutive frames. The state transition distribution $p\big(\boldsymbol{s}_t|\boldsymbol{s}_{t-1}\big)$ is modeled by Brownian motion, *i.e.*, $p\big(\boldsymbol{s}_t|\boldsymbol{s}_{t-1}\big) = \mathcal{N}(\boldsymbol{s}_t; \boldsymbol{s}_{t-1}, \sum)$, where $\sum$ is a diagonal covariance matrix whose diagonal elements are the corresponding variances of respective parameters. The observation model $p(\boldsymbol{o}_t|\boldsymbol{s}_t)$ is defined as

$$p(\boldsymbol{o}_t|\boldsymbol{s}_t) \propto SC_t, \tag{13}$$

where $SC_t = \widehat{f}\big(\boldsymbol{x}^{(t)}\big)$ is the classification score at time t based on Eq. (11).

4 Experiments

We run our tracker on 65 challenging image sequences including the benchmark dataset [28] and 14 public sequences widely used in recent literatures. The total number of frames on the 65 sequences is more than 30000. We evaluate the proposed tracker against 11 state-of-the-art tracking algorithms including ONNDL [29], RET [30], CT [31], VTD [4], MIL [11], SCM [32], Struck [12], TLD [13], ASLSA [2], LSST [3] and SPT [14]. For fair comparisons, the source codes are provided by the benchmark with the same parameters except ONNDL, RET, LSST and SPT whose parameters of the particle filter are set as same as our tracker. As discussed in [28], we also annotate the attributes of 14 public sequences used in our paper. The proposed approach was implemented in MATLAB on a Intel Core2 2.5 GHz processor with 4GB RAM. Our tracker is about 2 frame/sec for all experiments. No code optimization is performed. The MATLAB source code and experimental results of 12 trackers are available at `http://iitlab.bit.edu.cn/mcislab/~wuyuwei/`.

4.1 Experimental setup

Note that we fix the parameters of our tracker for all sequences to demonstrate its robustness and stability. The number of particles is 400 and the state transition matrix is $[8, 8, 0.01, 0, 0.005, 0]$ in the particle filter. We resize the object image to 24×24 pixels. Gray scale information and HOG feature are extracted from each object region. In the first frame, $N_p = 20$ positive samples and $N_n = 100$ negative samples are used to initialize the classifier. The regularization parameter expressed in Eq. (10) is set to $\eta = 0.02$. The predefined threshold of classification score ϵ is set as 0.3. Given the object location at current frame, if $SC \geq \epsilon$, 2 positive samples and 50 negative samples are used for the supervised learning. If $SC < \epsilon$, the tracking result is treated as the unreliable one and 100 unlabeled sample are utilized for the unsupervised learning. The sample pool capacity $\Theta(\boldsymbol{X}_P)$ is set to 310, in which the number of positive, negative and unlabeled samples are 50, 160 and 100, respectively. The number of landmarks is set to 30 empirically. As a trade-off between computational efficiency and effectiveness, the landmarks set $\boldsymbol{D}$ is updated every $T = 10$ frames.

4.2 Quantitative comparisons

Evaluation criteria To measure the tracking performance, the *precision plot* [11] is adopted to measure the overall tracking performance. It shows the percentage of frames whose estimated location is within the given threshold distance of the ground truth. More accurate trackers have higher precision at lower thresholds. If a tracker loses the object, it is difficult to reach a higher precision.

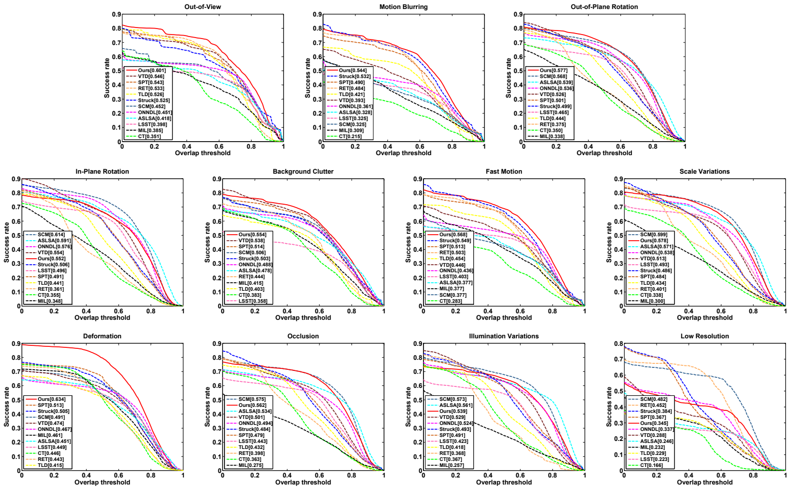

Fig. 3. Attribute-based performance analysis in success rate. The performance score of each tracker is shown in the legend (best viewed on high-resolution display).

The tracking overlap rate is also used for quantitative comparisons. It is defined by $score = \frac{area(ROI_T \bigcap ROI_G)}{area(ROI_T \bigcup ROI_G)}$, where ROI_T is the tracking bounding box and ROI_G is the ground truth. This can be used to evaluate the *success rate* of any tracking approach. The tracking result is considered as a success when the *score* is greater than the given threshold t_s. However, it may not be fair or representative for tracker evaluation using one success rate value at a specific threshold (*e.g.*, $t_s = 0.5$). Therefore, we count the number of successful frames at the thresholds varied from 0 to 1 and plot the *success rate* curve for our tracker and the compared trackers. The area under curve (AUC) of each success rate plot is employed to rank the tracking algorithms. More robust trackers have higher success rate at higher thresholds.

Overall performance The overall performance for 12 trackers is summarized by the precision plot and success rate on 65 sequence, as shown in Fig. 2. For precision plots, we use the results at error threshold of 20 for ranking these 12 trackers. The AUC score for each tracker is shown in the legend. In success rate, our tracker is 2.8% above the SCM, and outperforms the Struck by 3.1% in precision plot. SCM, ASLSA and LSST trackers also perform well in success rate, which suggests sparse representations are effective models to account for appearance change, especially for occlusion. Overall, our tracker outperforms other 11 trackers both in precision plot and success rate. Good performance of our method can be attributed to the fact that the classifier generalizes well on the new data from a limited number of training samples. That is, our method has excellent generalization ability. In addition, the local manifold structure of samples makes the classifier have more discriminating power.

Attribute-based performance Apart from summarizing the performance on the whole sequences, we also construct 11 subsets corresponding to different attributes to report specific challenging conditions. Fig. 3 shows the attribute-based performance analysis in success rate. Attributes OCC, IPR, OPR and SV occur more frequently than others on 65 sequences. Due to space limitations, in the following we mainly analyze the success rate and precision plot for these four attributes mentioned above and use other attributes as auxiliary.

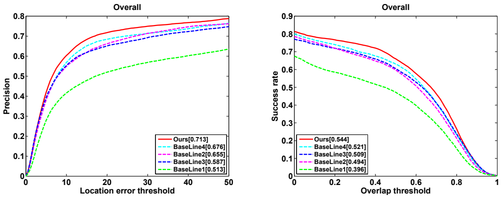

Fig. 4. The overall performance of two baseline algorithms and our method on 65 sequences is presented for comparison in terms of precision and success rate.

On the OCC subset, SCM, ASLSA, LSST and our method get better results than others. The results suggest that local image representations are more effective than holistic templates in dealing with occlusions. On the SV subset, we see that trackers with affine motion models (*e.g.*, our method, SCM, ASLSA and LSST) are able to cope with scale variation better than others that only consider translational motion (*e.g.*, Struck and MIL). On the OPR and IPR subsets, besides our tracker, the SCM and ASLSA trackers is also able to obtain the satisfactory results. The performance of SCM and ASLSA trackers can be attributed to the efficient spare representations of local image patches. Similarly, on the FM and MB subsets, Struck, SPT, TLD and our trackers perform favorably against other methods, which implies a good online learning algorithm facilitates trackers by updating the classifiers to adapt to appearance changes of the object.

Effectiveness of the optimal $\boldsymbol{H}$ To evaluate the contribution of the optimal $\boldsymbol{H}$ described in Sect. 2.2 to the overall performance of our tracker, we compute the Nadaraya-Watson kernel regression [33] for comparison. It assigns weights smoothly with $\boldsymbol{H}_{ik} = \frac{K_\sigma(\boldsymbol{x}_i,\boldsymbol{d}_k)}{\sum_{j=1}^{m} K_\sigma(\boldsymbol{x}_i,\boldsymbol{d}_j)}, 1 \leq i \leq n, 1 \leq j \leq m$. Two kernel functions are exploited in the Nadaraya-Watson kernel regression to measure the cross-similarity matrix between the whole data $\boldsymbol{X}$ and landmarks $\boldsymbol{d}_k$'s. We first adopt Gaussian kernel for the kernel regression and the corresponding tracking method is called as the *BaseLine1*. Epanechnikov quadratic kernel is also employed for the kernel regression, whose corresponding tracking method is referred to as the *BaseLine2* tracker. We use a more robust way to get σ which uses the nearest neighborhood size s of $\boldsymbol{x}_i$ to replace σ, *i.e.*, $\sigma(\boldsymbol{x}_i) = \|\boldsymbol{x}_i - \boldsymbol{d}_s\|^2$, where $\boldsymbol{d}_s$ is the sth closest landmarks of $\boldsymbol{x}_i$. The only difference between baseline algorithms and *Ours* is that baseline algorithms utilize the predefined kernel functions to solve cross-similarity matrix $\boldsymbol{H}$ while *Ours* takes advantage of local landmarks approximation method to optimize $\boldsymbol{H}$. The overall tracking performance of these baseline algorithms and our method on the 65 challenging sequences is presented in Fig. 4. On the whole, our method obtains more accurate tracking results than baseline algorithms.

Effectiveness of the prediction matrix $\boldsymbol{A}$ We design another two baseline algorithms to evaluate the effectiveness of the soft label prediction matrix $\boldsymbol{A}$ described in Sect. 2.3. In the *BaseLine3*, we do not consider the Laplacian graph regularizer in Eq. (9), *i.e.*, $\eta = 0$, and thus $\boldsymbol{A}$ becomes the least-squares solution. In the *BaseLine4*, we directly construct the adjacent matrix $\boldsymbol{W}$ using the kNN algorithm instead of $\boldsymbol{W} = \boldsymbol{H}\boldsymbol{H}^\top$. If $\boldsymbol{x}_i$ is among the k-neighbors of $\boldsymbol{x}_j$ or $\boldsymbol{x}_j$ is among the k-neighbors of $\boldsymbol{x}_i$, $\boldsymbol{W}_{ij} = 1$, otherwise, $\boldsymbol{W}_{ij} = 0$. The overall tracking performance on the benchmark is illustrated in Fig. 4. Surprisingly, even without Laplacian graph regularizer, the *BaseLine3* produces the precision score of 0.587 and the success score of 0.509, outperforming the ONNDL tracker, which implies that the success is due to the framework of the landmark-based inductive model. The overall performance can be further improved using our scheme of solving $\boldsymbol{A}$ described in Sect. 2.3.

4.3 Qualitative comparisons

Fig. 5 shows the qualitative tracking results of the 12 trackers over nine representative video sequences. In the *dragonbaby*, *Basketball* and *Freeman4* sequences are used to evaluate whether our method is able to handle significant pose changes. The *dragonbaby*, VTD, RET, ASLSA, SPT, SCM and TLD trackers are easy to drift at the beginning of the sequence when the object turns around (*e.g.*, $\sharp$28). The LSST tracker and our methods are able to track the object well although with some errors in some frames. SCM and ASLSA trackers

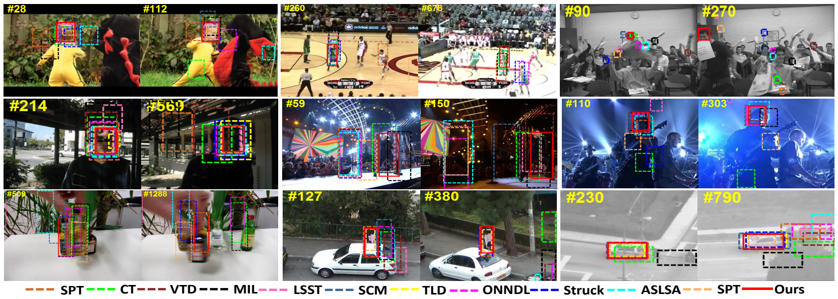

Fig. 5. Qualitative tracking results of the 12 trackers over 9 representative video sequences (*i.e.*, 'Dragonbaby", "Basketball", "Freeman4", "Trellis", "Singer2", "shaking","Liquor", "Woman" and "SUV") that are respectively aligned from left to right and from up to down (best viewed on high-resolution display).

do not perform well in this sequence as the drastic appearance changes due to shape information are not effectively accounted for the sparse representation. In the *Basketball* sequence, we see that SPT, CT, RET and SCM trackers are easy to drift at the beginning of the sequence (*e.g.,* ♯60). The TLD, ONNDL, Struck and MIL algorithms drift to another player as the appearance between players in the same team is very similar (*e.g.,* ♯473). VTD, ASLSA and our methods are able to track the whole sequence successfully. In the *Freeman4* sequence, all the trackers except our method perform poorly since the partial occlusions appear frequently. SMC method employs a fixed histogram intersection function to compute the similarity of histograms between the candidate and the template, thereby leading to lacking the ability to adapt to scene changes.

The *Woman*, *SUV* and *Liquor* sequences are utilized to test if our methods can tackle the occlusions. In the *Woman* sequence, the CT, SCM, MIL, VTD, TLD and ONNDL trackers fail to capture the object after the woman walks behind white car (*e.g.*, ♯127). The appearance model fuses more background interference due to an occlusion, which significantly influences the samples online update of the MIL, TLD, ASLSA and RET trackers. The LSST tracker fails gradually over time (*e.g.*, ♯380). In contrast, our method, SPT and Struck trackers achieve stable performance in the entire sequence. For the *SUV* sequence, most of the trackers drift when the long-term occlusion happens. In comparisons, our tracker and SCM have relatively lower center location errors and higher success rate. Although LSST and ASLSA trackers take partial occlusion into account, the results are not satisfied. The RET and TLD trackers are also achieve the satisfying results. In the *Liquor* sequence, the object suffers from background clutter besides heavy occlusions for many times. The CT, MIL, LSST and ASLSA trackers drift first when the occlusion occurs (*e.g.*, ♯361). Although the TLD, VTD, SPT, RET and Struck trackers obtain slightly better results than SCM and ONNDL trackers, they lose the object after several occlusions. Overall, our method achieves both the lowest tracking error and the highest overlap rate.

In the *Shaking*, *Singer2* and *Trellis* sequences, the objects undergo drastic illumination changes. From the *Shaking* sequence, we see that the Struck, LSST, TLD, CT and RET trackers drift from the object quickly when the spotlight blinks suddenly (*e.g.*, ♯110). SCM, VTD, ONNDL and our trackers can successfully track the surfer throughout the sequence with relatively accurate sizes of the bounding box. SPT, MIL, and ASLSA methods are also able to track the object in this sequence but with lower success rate than our method. In the *Singer2* sequence, the contrast between the foreground and the background is very low besides illumination change. Most trackers drift away at the beginning of the sequence when the stage light changes drastically (*e.g.*, ♯59). The VTD tracker performs slightly better as the edge feature is less sensitive to illumination change. In contrast, our method succeeds in tracking the object accurately. In *Trellis* sequence, a man walks under a trellis. Suffering from large changes in environmental illumination and head pose, the CT, TLD, MIL, SPT and LSST trackers drift gradually. In contrast, RET, ONNDL, ASLSA, SCM, Struck and our trackers obtain promising results.

5 Conclusion

In this paper, we have proposed a landmark-based inductive model for tracking. The idea of our method is that the label of each sample can be interpreted as the weighted combination of labels on landmarks. Through solving the cross-similarity matrix $\boldsymbol{H}$ and the label prediction matrix $\boldsymbol{A}$, our model is able to effectively propagate the landmarks' labels to all the unlabeled candidates. The Lim tracker is able to effectively fit the underlying data distribution to handle appearance changes. A candidate with the highest classification score is considered as the tracking result. In addition, explicitly considering the local geometrical structure of the samples, the graph-based regularizer is incorporated into the lim tracker, which makes our method have better discriminating power and thus is more adaptive to handle appearance changes. Compared with 11 state-of-the-art tracking methods on 65 challenging image sequences, the Lim tracker is more robust to illumination changes, pose variations and partial occlusions, *etc.* Experimental results have demonstrated the effectiveness and robustness of the proposed tracker.

Acknowledgement. This work was supported in part by the Natural Science Foundation of China (NSFC) under grant No. 61203291, the 973 Program of China under grant No. 2012CB720000, the Specialized Research Fund for the Doctoral Program of Higher Education of China under grant No.20121101120029, and the Specialized Fund for Joint Building Program of Beijing Municipal Education Commission.

References

1. Li, X., Hu, W., Shen, C., Zhang, Z., Dick, A., Hengel, A.V.D.: A survey of appearance models in visual object tracking. ACM Transactions on Intelligent Systems and Technology (TIST) **4** (2013) 58

2. Jia, X., Lu, H., Yang, M.: Visual tracking via adaptive structural local sparse appearance model. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. (2012) 1822–1829
3. Wang, D., Lu, H., Yang, M.H.: Least soft-thresold squares tracking. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. (2013) 2371–2378
4. Kwon, J., Lee, K.: Visual tracking decomposition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. (2010) 1269–1276
5. Mei, X., Ling, H.: Robust visual tracking using $\ell 1$ minimization. In: Proceedings of IEEE International Conference on Computer Vision. (2009) 1–8
6. Ross, D., Lim, J., Lin, R., Yang, M.: Incremental learning for robust visual tracking. International Journal of Computer Vision **77** (2008) 125–141
7. Wu, Y., Ma, B.: Learning distance metric for object contour tracking. Pattern Analysis and Applications **17** (2014) 265–277
8. Zhuang, B., Lu, H., Xiao, Z., Wang, D.: Visual tracking via discriminative sparse similarity map. Image Processing, IEEE Transactions on **23** (2014) 1872–1881
9. Grabner, H., Leistner, C., Bischof, H.: Semi-supervised on-line boosting for robust tracking. In: Proceedings of European Conference on Computer Vision. (2008) 234–247
10. Yang, M., Yuan, J., Wu, Y.: Spatial selection for attentional visual tracking. In: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, IEEE (2007) 1–8
11. Babenko, B., Yang, M., Belongie, S.: Robust object tracking with online multiple instance learning. Pattern Analysis and Machine Intelligence, IEEE Transactions on **33** (2011) 1619 –1632
12. Hare, S., Saffari, A., Torr, P.H.: Struck: Structured output tracking with kernels. In: Proceedings of the IEEE International Conference on Computer Vision. (2011) 263–270
13. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on **34** (2012) 1409–1422
14. Yao, R., Shi, Q., Shen, C., Zhang, Y., van den Hengel, A.: Part-based visual tracking with online latent structural learning. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. (2013) 2363–2370
15. Yang, M., Pei, M., Wu, Y., Jia, Y.: Learning online structural appearance model for robust object tracking. Science China Information Sciences **57** (2014) In press
16. Yu, Q., Dinh, T.B., Medioni, G.: Online tracking and reacquisition using co-trained generative and discriminative trackers. In: ECCV, Springer (2008) 678–691
17. Zhang, L., van der Maaten, L.: Preserving structure in model-free tracking. Pattern Analysis and Machine Intelligence, IEEE Transactions on **36** (2014) 756–769
18. Wu, Y., Ma, B., Yang, M., Jia, Y., Zhang, J.: Metric learning based structural appearance model for robust visual tracking. IEEE Transactions on Circuits and Systems for Video Technology **24** (2014) 865–877
19. Gao, J., Xing, J., Hu, W., Maybank, S.: Discriminant tracking using tensor representation with semi-supervised improvement. In: Proceedings of the IEEE International Conference on Computer Vision. (2013) 1569–1576
20. Li, X., Shen, C., Dick, A.R., van den Hengel, A.: Learning compact binary codes for visual tracking. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. (2013) 2419–2426
21. Kumar, K., Vleeschouwer, C.: Discriminative label propagation for multi-object tracking with sporadic appearance features. In: Proceedings of the IEEE International Conference on Computer Vision. (2013) 2000–2007

22. Zhang, K., Kwok, J.T., Parvin, B.: Prototype vector machine for large scale semi-supervised learning. In: Proceedings of the 26th Annual International Conference on Machine Learning, ACM (2009) 1233–1240
23. Liu, W., Wang, J., Chang, S.F.: Robust and scalable graph-based semisupervised learning. Proceedings of the IEEE **100** (2012) 2624–2638
24. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality-constrained linear coding for image classification. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2010) 3360–3367
25. Nesterov, Y., Nesterov, I.: Introductory lectures on convex optimization: A basic course. Volume 87. Springer (2004)
26. Duchi, J., Shalev-Shwartz, S., Singer, Y., Chandra, T.: Efficient projections onto the l1-ball for learning in high dimensions. In: Proceedings of the 25th international conference on Machine learning, ACM (2008) 272–279
27. Williams, C., Seeger, M.: Using the nyström method to speed up kernel machines. In: Advances in Neural Information Processing Systems, Citeseer (2001)
28. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2013) 2411–2418
29. Wang, N., Wang, J., Yeung, D.Y.: Online robust non-negative dictionary learning for visual tracking. In: Proceedings of IEEE International Conference on Computer Vision. (2013) 657–664
30. Bai, Q., Wu, Z., Sclaroff, S., Betke, M., Monnier, C.: Randomized ensemble tracking. In: Proceedings of IEEE International Conference on Computer Vision. (2013) 2040–2047
31. Zhang, K., Zhang, L., Yang, M.H.: Real-time compressive tracking. In: Proceedings of European Conference on Computer Vision, Springer (2012) 864–877
32. Zhong, W., Lu, H., Yang, M.: Robust object tracking via sparsity-based collaborative model. Image Processing, IEEE Transaction on **23** (2014) 2356–2368
33. Hastie, T., Tibshirani, R., Friedman, J.: The elements of statistical learning. Volume 2. (2009)