Linearization to Nonlinear Learning for Visual Tracking

Bo Ma [1], Hongwei Hu [1], Jianbing Shen* [1], Yuping Zhang [1], Fatih Porikli [2]

[1]Beijing Lab of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, China
[2]Research School of Engineering, Australian National University, and NICTA Australia

Abstract

Due to unavoidable appearance variations caused by occlusion, deformation, and other factors, classifiers for visual tracking are nonlinear as a necessity. Building on the theory of globally linear approximations to nonlinear functions, we introduce an elegant method that jointly learns a nonlinear classifier and a visual dictionary for tracking objects in a semi-supervised sparse coding fashion. This establishes an obvious distinction from conventional sparse coding based discriminative tracking algorithms that usually maintain two-stage learning strategies, i.e., learning a dictionary in an unsupervised way then followed by training a classifier. However, the treating dictionary learning and classifier training as separate stages may not produce both descriptive and discriminative models for objects. By contrast, our method is capable of constructing a dictionary that not only fully reflects the intrinsic manifold structure of the data, but also possesses discriminative power. This paper presents an optimization method to obtain such an optimal dictionary, associated sparse coding, and a classifier in an iterative process. Our experiments on a benchmark show our tracker attains outstanding performance compared with the state-of-the-art algorithms.

1. Introduction

Visual object tracking is a fundamental task in computer vision, and various tracking algorithms [23, 14, 19, 16, 28, 25, 7, 4] have been proposed in the past. Nevertheless, it still remains as a challenging problem for typical real-world scenarios where many uncontrolled factors such as illumination variation, occlusion, deformation, in-plane rotation, background clutters, etc. encountered during tracking cause considerable complications.

Among methods addressing these difficulties, sparse coding based methods have attracted much attention because of their robust performance in similar classification [10] and recognition [29] tasks. After the first trial of Mei and Ling [19] in tracking based on l_1 minimization. Liu *et al*. [16] proposed a local sparse appearance model with K-selection to represent target appearance. To model local appearance, Wang *et al*. [28] learned visual prior from generic real-world images and transferred it into local sparse representation. Wang *et al*. [25, 24] supposed noise term in sparse representation was Gaussian-Laplacian distributed and introduced a least soft-thresold squares algorithm. However, the dictionary in most of these tracking methods based on sparse representation often results from unsupervised clustering algorithms, and may not be the most advantageous to visual tracking. Most sparse coding approaches do not take the spatial structure between original sample space and coding space into consideration. When encoding a sample, it is argued that those items in dictionary close to this sample to be reconstructed should be activated. Local Coordinate Coding (LCC) [34, 33] learns a set of anchor points (also called dictionary) to reconstruct original samples while keeping locality. It has recently shown impressive performance on nonlinear learning. An approximation method of LCC was proposed by wang *et al*. [26], which demonstrated the good performance of locality constrain in sparse coding for image classification. Lin *et al*. [15] have already introduced it into large-scale image classification, which showed good performance and high classification accuracy. But they formed the dictionary by k-means only. Xie *et al*. [31] proposed a large-scale dictionary learning for LCC in object recognition experiments. Hu *et al*. [8] introduced nonlinear learning using LCC into visual tracking. But their method took still a two-stage learning strategy, and the treatment of separating dictionary learning from classifier learning may not produce optimal dictionary for tracking.

Dictionary learning and updating are crucial steps of

*Corresponding author: Jianbing Shen (shenjianbing@bit.edu.cn).
This work was supported in part by the National Natural Science Foundation of China (61472036 and 61272359), and the National Basic Research Program of China (973 Program) (2013CB328805). Specialized Fund for Joint Building Program of Beijing Municipal Education Commission.

handling and adapting to appearance changes during the tracking process. Therefore, how to choose a suitable dictionary carries great importance. Mei and Ling [19] employed the idea of holistic target templates by a set of trivial positive and negative templates as the dictionary to encode the candidate target, and proposed a pioneering tracking algorithm using sparse representation. To handle occlusion more efficiently, Jia *et al*. [9] extended dictionary learning to overlapped local patches that are cropped from holistic target templates. This method considered only reconstruction error of the target while ignoring the discriminative information contained in training samples. Accounting for drastic appearance changes, Zhong *et al*. [36] exploited both holistic templates and local representation, and proposed a sparsity-based collaborative model by combing a generative model and a discriminative one. Most sparse coding based discriminative tracking algorithms [27, 17] have separate dictionary learning and classifier training mechanisms. The dictionary usually acquired from unsupervised clustering algorithms, and may not suit necessarily to tracking objective. Yang *et al*. [32] proposed an online manner for visual tracking. But this method gives no consideration to locality of sparse codes, and fails to exploit the underlying manifold geometry since no unlabeled samples are considered for dictionary learning.

A suitable dictionary is significantly important to reconstruct samples accurately. Thus, much work has been done to establish a discriminative dictionary which exploits the discriminative information of samples. Zhang *et al*. [35] learned a discriminative dictionary by K-SVD and applied it in face recognition. Mairal *et al*. [18] learned discriminative dictionaries for local image analysis. Jiang *et al*. [10] introduced label consistent K-SVD to learn a discriminative dictionary for sparse coding in image classification. Pham *et al*. [21] proposed a framework of joint representation and classification, which was applied in pattern recognition. In order to achieve sparse representation, Aharon *et al*. [1] presented a method for designing overcomplete dictionaries. Kong and Wang [12] proposed an image classification approach using block-coordinate descent method based on online discriminative dictionary learning.

Due to appearance changes in real-life tracking scenarios, a classifier function for tracking is bounded to be nonlinear in essence. Appearance variation can be reminiscent of the curse of dimensionality problem. But in real tracking scenarios, we seldom suffer from this, and even a small number of templates can be sufficient to obtain good tracking performance. This is due to the fact that typically higher dimensional visual data often lie on embedding manifold of lower dimensionality. With this, the nonlinear theory [34] shows that the learning complexity of a nonlinear function depends on the inherent dimensionality of underlying input sample space under some Lipschitz continuity assumption. The nonlinear theory of using LCC also provides theoretical foundation to advocate discriminative tracking using sparse coding.

Inspired by the nonlinear learning theory, we introduce an elegant method to jointly learn a visual dictionary and classifier for visual tracking in a semi-supervised manner. The learned dictionary not only approximates the underlying manifold with both labeled and unlabeled samples considered, but also possesses favorable discriminative capacity. Thus, several shortcomings confronted by existing tracking methods mentioned above can be overcome efficiently. In addition, our method builds on a solid theoretical foundation, providing guidance for discriminative tracking by localizing sparse coding. An iterative optimization method to compute the discriminative dictionary, the sparse codes and the classifier is also presented. Based on the joint learning approach, we utilize particle filter framework as an update strategy. The proposed tracking method has been tested on over fifty video sequences, and reported very promising results. Our source code will be publicly available online [1].

2. Linearization to Nonlinear Learning

2.1. Problem Formulation

Given a set of labeled samples $X_l = \{\mathbf{x}_i \in \mathcal{R}^d\}_{i=1}^n$ with their labels $Y = \{y_i\}_{i=1}^n$ and a group of unlabeled sample $X_u = \{\mathbf{x}_i\}_{i=n+1}^{n+u}$, we aim to learn a discriminative dictionary to represent samples, a nonlinear classifier to distinguish positive and negative samples and the sparse coefficients of each sample under the learned dictionary. It has been proved that a (β, δ, p)-Lipschitz smooth nonlinear classification function $f(\mathbf{x})$ can be approximated by a linear function in regard to local coordinate coefficients of a sample [34], LCC is just the upper bound of the approximation which is computed as

$$\min_{\mathbf{D},\boldsymbol{\alpha}_i} \|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|^2 + \mu \sum_{j=1}^{m} |\alpha_i^j| \|\mathbf{d}_j - \mathbf{x}_i\|^2, \quad (1)$$
$$\text{s.t.} \quad \mathbf{1}^T \boldsymbol{\alpha}_i = 1,$$

where μ is a constant factor, $\mathbf{D} = [\mathbf{d}_1, \cdots, \mathbf{d}_m] \in \mathbf{R}^{d \times m}$ is the dictionary, and α_i^j is the j-th element of $\boldsymbol{\alpha}_i$, and $\mathbf{1}$ is a vector with all ones.

Considering the spatial relationships of different samples, we argue that samples close to each should possess same labels. Like many coding methods, LCC seeks a linear combination of bases to reconstruct sample. An approximated way to calculate the sparse code of a sample is to find sample's k nearest neighbors in bases and then solve a constrained least squares problem with these k bases ([15]).

[1] *http://github.com/shenjianbing/LCCtracking*

Algorithm 1 Linearization to Nonlinear Learning

Input: $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, $\{\mathbf{x}_i\}_{i=n+1}^{n+u}$, μ, λ_1 and λ_2.
Output: $\mathbf{D}$, $\mathbf{A}$ and $\mathbf{w}$.

1: Initialization: $\mathbf{D}$ is obtained by k-means, $\mathbf{A}_l = (\mathbf{D}^T\mathbf{D})^{-1}(\mathbf{D}^T\mathbf{X})$, and $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_{n+u}]$.
2: $t = 1$.
3: **while** $t < T$ **do**
4: Classifier learning: Solve $\mathbf{w}$ with fixed $\mathbf{D}$ and $\mathbf{A}$ using Eq. (4);
5: Coding: Solve $\mathbf{A}$ with fixed $\mathbf{D}$ and $\mathbf{w}$ by Algorithm 2;
6: Dictionary learning: Learn $\mathbf{D}$ with fixed $\mathbf{A}$ and $\mathbf{w}$ by Eq. (14);
7: $t = t + 1$.
8: **end while**

Hence, similar samples have similar sparse codes, and it is reasonable to expect similar sparse codes correspond to the same labels. Moreover, the globally linear approximation to the nonlinear function $f(\mathbf{x}_i)$ can be denoted as $f(\mathbf{x}_i) \approx \boldsymbol{\alpha}_i^T\mathbf{w}$ where $\mathbf{w}$ is a weighted vector under the nonlinear learning theory using LCC. To learn a discriminative dictionary, the labeled samples must also be considered.

Consequently, we formulate our joint semi-supervised discriminative dictionary learning, sparse codes and classifier learning using both labeled and unlabeled samples as

$$\begin{aligned}\min_{\mathbf{D},\mathbf{A},\mathbf{w}} &\sum_{i=1}^{u+n}\left(\|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|^2 + \mu\sum_{j=1}^{m}|\alpha_i^j|||\mathbf{d}_j - \mathbf{x}_i||^2\right)\\ &+ \lambda_1||\mathbf{A}_l^T\mathbf{w} - \mathbf{y}||^2\\ &+ \lambda_2\sum_{i=1}^{n+u}\sum_{j=1}^{n+u}\|\boldsymbol{\alpha}_i^T\mathbf{w} - \boldsymbol{\alpha}_j^T\mathbf{w}\|^2 B_{ij}, \qquad (2)\\ \text{s.t.}\quad &\mathbf{1}^T\boldsymbol{\alpha}_i = 1, \quad i = 1, \cdots, n+u.\end{aligned}$$

The first item in the above objective function considers both labeled and unlabeled samples, and the second one is the discriminative item where $\mathbf{A}_l = [\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_n]$ is the code matrix corresponding to samples with labels, and the last one is the Laplacian constrain with $B_{ij} = \boldsymbol{\alpha}_i^T\boldsymbol{\alpha}_j$, and λ_1, λ_2 are the parameters that adjust the influence of discriminative power and the Laplacian regularization item respectively. In fact, the sign of each element in the coefficient vector $\mathbf{w}$ can be considered as the label of corresponding dictionary item as discussed in Sec. 2.2, while each code of sample is the weight vector with respect to dictionary items. It is discovered that the locality of local coordinate coding leads to sparsity, but not vice versa.

2.2. Optimization

The optimization problem in Eq. (2) is not jointly convex over $\mathbf{D}$, $\mathbf{w}$, and $\mathbf{A}$, which makes it difficult to solve directly. Whereas, it can be solved over one variable with fixed two other ones. Thus, we decompose the optimization problem into three sub-problems.

Classifier Learning. With fixed dictionary $\mathbf{D}$ and sparse code matrix $\mathbf{A}$, the classifier can be learned with the following optimization problem,

$$\begin{aligned}\min_{\mathbf{w}} &\lambda_1||\mathbf{A}_l^T\mathbf{w} - \mathbf{y}||^2 + \lambda_2\sum_{i=1}^{n+u}\sum_{j=1}^{n+u}\|\boldsymbol{\alpha}_i^T\mathbf{w} - \boldsymbol{\alpha}_j^T\mathbf{w}\|^2 B_{ij}, \qquad (3)\\ \text{s.t.}\quad &\mathbf{1}^T\boldsymbol{\alpha}_i = 1, \quad i = 1, \cdots, n.\end{aligned}$$

The minimization problem has a closed-form solution which can be received by setting the derivative of Eq. (3) to zero. The expression is written as

$$\mathbf{w} = \left(\lambda_1\mathbf{A}_l\mathbf{A}_l^T + \lambda_2\mathbf{A}(\boldsymbol{\Delta} - \mathbf{A}^T\mathbf{A})\mathbf{A}\right)^{-1}(\lambda_1\mathbf{A}_l\mathbf{y}), \quad (4)$$

where $\boldsymbol{\Delta} = \text{diag}(\Delta_1, \Delta_2, \cdots, \Delta_{u+n})$ with $\Delta_i = \sum_{j=1}^{u+n} B_{ij}$.

Coding. With fixed $\mathbf{D}$ and $\mathbf{w}$, the objective function is the same as Eq. (2). But the objective function is non-differential with respect to $\mathbf{A}$, so we introduce an approximation formulation inspired by locality-constrained linear coding [26] which can be derived analytically. What's more, the Laplacian regularization term is neglected due to its small impact on sparse codes in our experiments. Denote $\mathbf{c}_i = [c_i^1, \cdots, c_i^m]^T$ where $c_i^j = \|\mathbf{x}_i - \mathbf{d}_j\|$ represents the metric between sample $\mathbf{x}_i$ and dictionary item $\mathbf{d}_j$. The approximation objective function is written as

$$\begin{aligned}\min_{\mathbf{A}} &\sum_{i=1}^{u+n}\left(\|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|^2 + \mu\|\mathbf{c}_i \odot \boldsymbol{\alpha}_i\|^2\right) + \lambda_1||\mathbf{A}_l^T\mathbf{w} - \mathbf{y}||^2\\ \text{s.t.}\quad &\mathbf{1}^T\boldsymbol{\alpha}_i = 1, \quad i = 1, \cdots, n+u, \qquad (5)\end{aligned}$$

where $\odot$ indicates the Hadamard product. To optimize the minimization problem, we present to solve one column of $\mathbf{A}$ with fixed all the other columns, and iterate this procedure until convergence. The closed-form solution for each $\boldsymbol{\alpha}_i$ can be calculated analytically as

$$\boldsymbol{\alpha}_i = \mathbf{P}^{-1}\left(\mathbf{Q} - \frac{\mathbf{1}^T\mathbf{P}^{-1}\mathbf{Q}\mathbf{1} - 1}{\mathbf{1}^T\mathbf{P}^{-1}\mathbf{1}}\mathbf{1}\right). \qquad (6)$$

For the $\boldsymbol{\alpha}_i$s corresponding to sample $\mathbf{x}_i$s with labels,

$$\mathbf{P} = \mathbf{D}^T\mathbf{D} + \mu\text{diag}(\mathbf{c}_i^2) + \lambda_1\mathbf{w}\mathbf{w}^T, \qquad (7)$$

$$\mathbf{Q} = \mathbf{D}^T\mathbf{x}_i + \lambda_1\mathbf{w}y_i, \qquad (8)$$

Algorithm 2 Coding Algorithm

Input: $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, $\{\mathbf{x}_i\}_{i=n+1}^{n+u}$, μ, λ_1, $\mathbf{D}$ and $\mathbf{w}$.
Output: $\mathbf{A}$

$t = 1$;
while $t < T$ **do**
  for $i = 1 : n + u$ **do**
    $\mathbf{P} = \mathbf{D}^T\mathbf{D} + \mu\text{diag}(\mathbf{c}_i^2)$;
    $\mathbf{Q} = \mathbf{D}^T\mathbf{x}_i$;
    if $i \leq n$ **then**
      $\mathbf{P} = \mathbf{P} + \lambda_1\mathbf{w}\mathbf{w}^T$;
      $\mathbf{Q} = \mathbf{Q} + \lambda_1\mathbf{w}y_i$;
    end if
    $\boldsymbol{\alpha}_i = \mathbf{P}^{-1}\left(\mathbf{Q} - \frac{\mathbf{1}^T\mathbf{P}^{-1}\mathbf{Q}\mathbf{1}-1}{\mathbf{1}^T\mathbf{P}^{-1}\mathbf{1}}\mathbf{1}\right)$;
  end for
  $t = t + 1$.
end while

where $\text{diag}(\mathbf{c}_i^2)$ denotes a diagonal matrix with its j-th element $(c_i^j)^2$. For those $\boldsymbol{\alpha}_i$s correspond to samples without labels,

$$\mathbf{P} = \mathbf{D}^T\mathbf{D} + \mu\text{diag}(\mathbf{c}_i^2), \tag{9}$$

$$\mathbf{Q} = \mathbf{D}^T\mathbf{x}_i. \tag{10}$$

The coding algorithm is summarized in Algorithm 2.

Dictionary Learning. Given $\mathbf{A}$ and $\mathbf{w}$, the dictionary $\mathbf{D}$ can be obtained by solving the following minimization problem

$$\min_{\mathbf{D}} \sum_{i=1}^{u+n} \left(\|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|^2 + \mu \sum_{j=1}^{m} |\alpha_i^j| \|\mathbf{d}_j - \mathbf{x}_i\|^2 \right).$$

After derivation (more details please refer to [31]), the above equation is equivalent to minimizing

$$\min_{\mathbf{D}} tr(\mathbf{D}^T\mathbf{D}\mathbf{G}) - 2tr(\mathbf{D}^T\mathbf{S}), \tag{11}$$

with

$$\mathbf{G} = \sum_{i=1}^{n+u} \left(\boldsymbol{\alpha}_i\boldsymbol{\alpha}_i^T + \mu\text{diag}(|\boldsymbol{\alpha}_i|)\right), \tag{12}$$

$$\mathbf{S} = \sum_{i=1}^{n+u} \left(\mathbf{x}_i\boldsymbol{\alpha}_i^T + \mu\mathbf{x}_i|\boldsymbol{\alpha}_i|^T\right), \tag{13}$$

where $tr(\cdot)$ denotes the trace operator on a square matrix. The optimal dictionary can be obtained using block-coordinate descent as in [31], but a closed-form solution exists, which is derived as

$$\mathbf{D} = \mathbf{S}\mathbf{G}^{-1}. \tag{14}$$

The proposed algorithm is summarized in Algorithm 1.

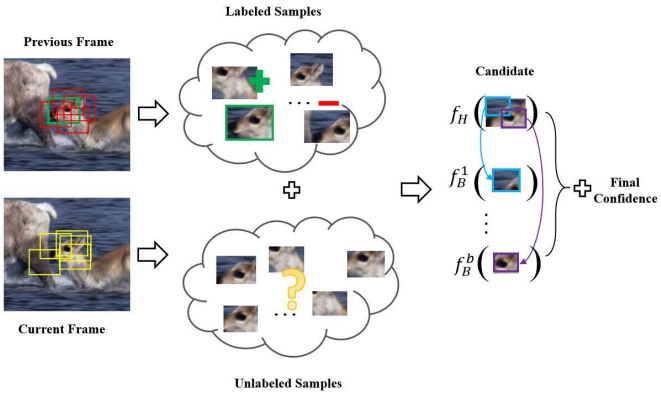

Figure 1. Confidence calculation. The holistic classifier f_H and block classifiers $f_B^1, \cdots, f_B^b$ are trained with labeled and unlabeled samples. Given a candidate, classifiers are applied to classify their corresponding blocks. The final confidence of a candidate is a weighted sum of holistic and block classifiers.

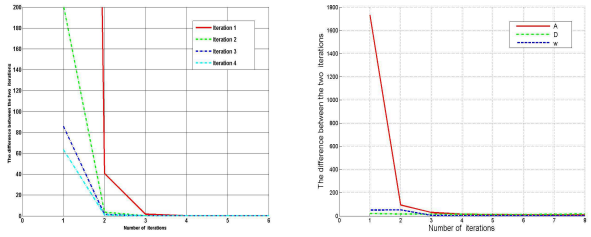

Figure 2. Convergence of the proposed coding algorithm (left) and the proposed joint discriminative dictionary, sparse codes and classifier learning algorithm (right).

3. Tracking Approach

3.1. Samples Acquisition

In most tracking approaches, the state of a target in the first frame is annotated manually, and we are no exception. To acquire the labeled samples for the proposed algorithm, we first sample a set of holistic templates $\{\mathbf{x}_i\}_{i=1}^{n}$ around target region randomly under a normal distribution. Numerous tracking algorithms treat tracking as a binary classification problem where the labels of samples are either positive or negative. In our experiments, we assign the samples with continuous labels in $[0, 1]$. For a sample $\mathbf{x}_i$ cropped around target region, its label is calculated as $y_i = (\text{Ar}_s \cap \text{Ar}_t)/(\text{Ar}_s \cup \text{Ar}_t)$, where Ar_t denotes the area of target region, Ar_s the area of sample.

3.2. Confidence Calculation

To train our model, we need unlabeled samples as well. Instead of collecting unlabeled samples off-line, we treat the candidates $\{\mathbf{x}_i\}_{i=n+1}^{n+u}$ in current frame as unlabeled samples in an on-line manner as shown in Figure 1. Candidates

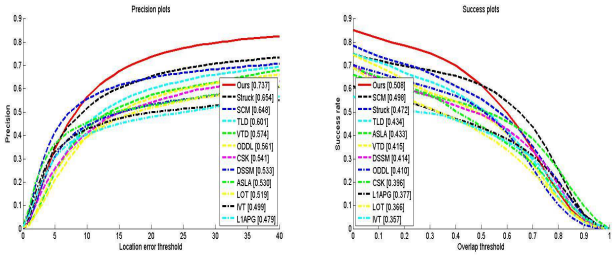

Figure 3. Overall performance comparisons of precision plot (left side) and success rate (right side) for these trackers. The overall performance score at 20 pixel is presented in the legend.

are sampled in current frame around the center of pervious target location. Once we obtain the labeled and unlabeled training samples, a dictionary $\mathbf{D}$, local coordinate code matrix $\mathbf{A}$ for all samples and weight coefficients $\mathbf{w}$ of the linear classifier could be calculated by Eq. (2) with the proposed algorithm. For a candidate $\mathbf{x}_i$, we can compute its label by $f(\mathbf{x}_i) = \boldsymbol{\alpha}_i^T \mathbf{w}$, where $\boldsymbol{\alpha}_i$ is the local coordinate codes of $\mathbf{x}_i$. The label here could be seen as a confidence that determines the similarity of candidate to target. Therefore, all confidences of candidates can be calculated with their local coordinate codes and weight coefficients obtained.

Considering only holistic templates may not be very effective to handle partial appearance changes especially caused by partial occlusion. Therefore, we divide a target region into several blocks, and collect different set of samples for different blocks. Labels of samples belong to block are assigned by the same way as holistic ones. Candidates in current frame are divided into blocks as the unlabeled block samples. Let $f_H(\mathbf{x}_i)$ denotes the classification confidence for the holistic template of candidate $\mathbf{x}_i$ and $\{f_B^j(\mathbf{x}_i^j)\}_{j=1}^b$ indicate the confidences for its blocks. The final confidence of $\mathbf{x}_i$ is measured as

$$f(\mathbf{x}_i) = \nu f_H(\mathbf{x}_i) + (1-\nu)\frac{1}{b}\sum_{j=1}^{b} f_B^j(\mathbf{x}_i^j), \qquad (15)$$

where $\mathbf{x}_i^j$ is the j-th block of sample $\mathbf{x}_i$. For computational efficiency, we train these classifier every few frames. To obtain the local codes of samples, we execute the local coordinate coding in Eq. (1) with the learned dictionary.

The proposed tracking algorithm is implemented under the particle filter framework. We employ a Gaussian distribution to model the state motion distribution $p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ with six independent parameters of the affine transformation, and the candidates are extracted from an importance function which equals to the motion model. Thus, the weights of each particle become the observation likelihood $p(y_i|\mathbf{x}_i)$ of candidate $\mathbf{x}_i$, which is proportional to the final confidence of $\mathbf{x}_i$ in Eq (15). To determine current target state, we choose the candidate with the highest probability.

3.3. Updating

The appearance of target during tracking changes continuously caused by illumination variations, occlusions, and deformation etc., so the manifold geometric of sample space will change as well. Therefore, the dictionary should be recalculated to adapt the new manifold structure of samples, and the classifiers and sparse codes should also be updated. In our implementation, we retain two sample pools during tracking: one is used to store labeled samples and the other one is for unlabeled samples. When the classification confidence of the current target is larger than a preset threshold θ, we collect a set of labeled samples based on current target state, and put these samples into the former pool. If the confidence of current target is smaller than the threshold θ, the candidates in current frames are regarded as unlabeled samples and will be placed in the latter pool. After every several frames during tracking, we select a certain number of labeled and unlabeled samples from these two pools randomly, and recalculate the discriminative dictionary and classifier using Algorithm 1. We carry out the updating for both holistic samples and block ones.

4. Experiments

The proposed tracking method is tested on a tracking benchmark [30] with 51 sequences which contain different kinds of difficulties encountered in visual tracking to verify the performance. The Testing video sequences and ground truth are accessed from `http://visual-tracking.net/`. We compare our method with 9 of the most popular tracking approaches in this benchmark which are ASLA [9], CSK [6], IVT [22], L1APG [2], LOT [20], SCM [36], Struck [5], TLD [11] and VTD [13]), and two other tracking algorithms (DSSM [37] and ODDL [32]) which are very related to our works. To evaluate the performance of our tracker quantitatively, the tracking results are estimated with distance precision (DP) and overlap precision (OP) as in [30] by one-pass evaluation (OPE) as in the benchmark.

4.1. Experimental Results

The feature in our experiments for each template is a vector combined by the intensity vector stretched by row of target and the histograms of oriented gradients (HOG) feature [3] of samples. Our tracking approach is carried out under the framework of particle filters with the affine parameters $[10, 10, 0.04, 0, 0.001, 0]$, and particle number is set to 600 which is the same as the number of tracking algorithms to be compared. To implement the proposed joint dictionary, sparse codes and classifier learning algorithm, we set the number of positive samples to 20 and that of negative samples to 200, and for unlabeled samples, the number is 200. The size of holistic templates are normalized by 24×24, and

Table 1. The performance scores for these trackers.

	ASLA	DSSM	Struck	IVT	L1APG	LOT	SCM	TLD	VTD	CSK	ODDL	WDL	Ours
DP	0.530	0.533	0.654	0.499	0.479	0.519	0.648	0.601	0.574	0.541	0.561	0.689	**0.737**
OP	0.433	0.414	0.472	0.357	0.377	0.366	0.498	0.434	0.415	0.396	0.410	0.493	**0.508**

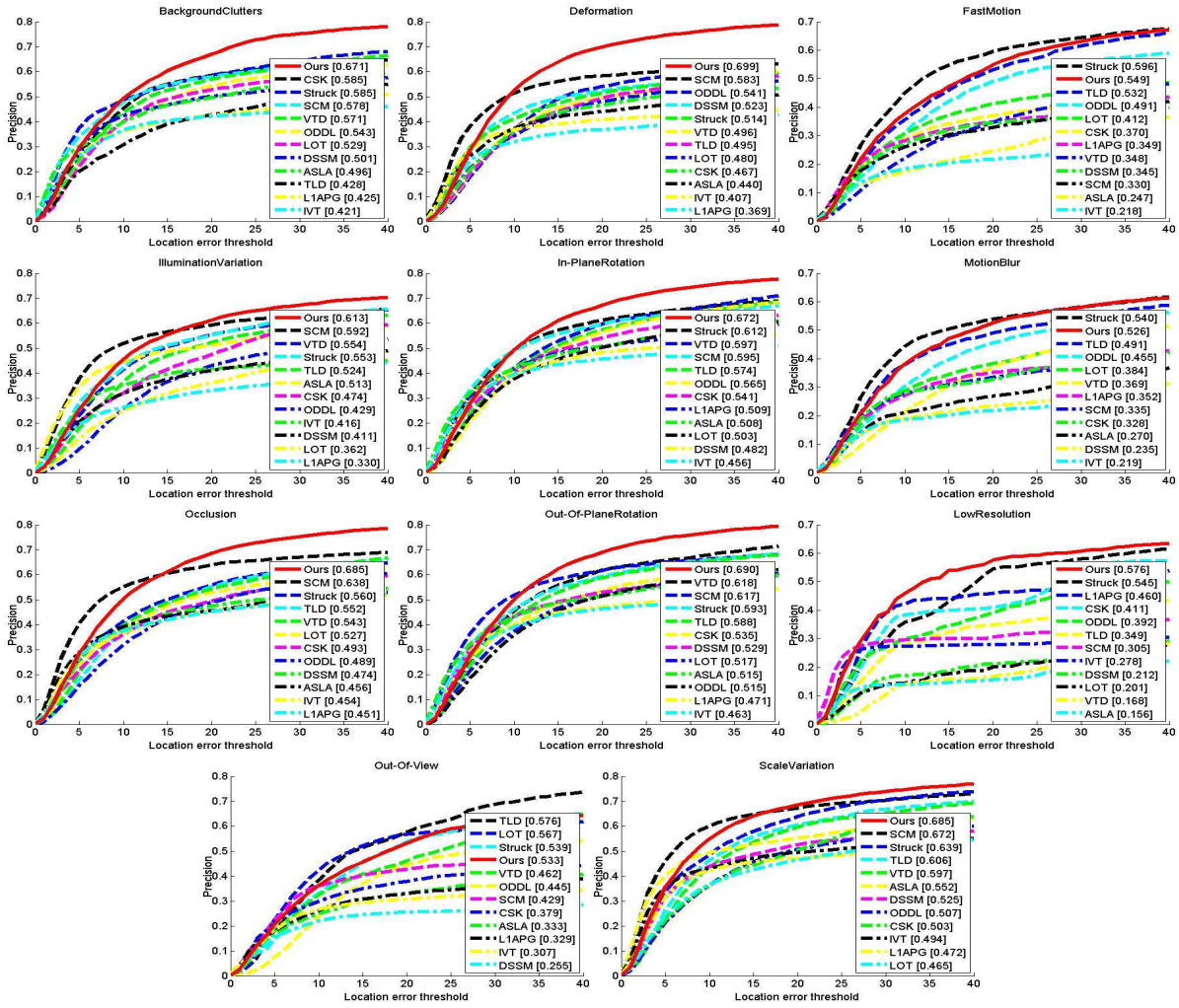

Figure 4. Attribute-based estimation comparisons of precision plots for these trackers.

the block size is 12×12 with step size $[12\ 12]$. We learn 20 dictionary items for holistic samples. To balance the influence of the holistic classifier and the block classifiers, we set the balance factor to 0.8 manually. The threshold θ used to update sample pools is set to 0.65. These parameters are fixed for all test sequences during the experiments. All these parameters are determined by cross validation.

As is shown in the left of Figure 2, we test the proposed coding algorithm on our experimental data, and demonstrate that the difference between two iterations converges with increasing of the iterations numbers. The proposed coding algorithm converges quickly, and less than 4 iterations are needed in our experiments. The proposed joint discriminative dictionary, sparse codes and classifier learning algorithm converges fast as shown in right side one of Figure 2. The differences between two iterations of dictionary $\mathbf{D}$, coding $\mathbf{A}$ and classifier $\mathbf{w}$ converge fast, which needs 8 iterations at most.

We compare our tracking results with the other trackers', and draw the precision plots and success plots in Figure 3. The center location errors of all trackers on all sequences are collected as the holistic evaluation, and the performance scores at 20 pixel are regarded as the criterion for ranking. We show the performance scores in the legend of precision

Figure 5. Visual comparisons. The names of these sequences are 'Boy', 'Car4', 'CarDark', 'Jumping', 'Crossing', 'David3', 'Deer', 'Lemming', 'Faceocc2', 'Doll', 'Faceocc1', 'MotorRolling', 'Dog1', 'Fish', 'Jogging-1', 'MountainBike', 'Subway', 'Singer1', 'Trellis' and 'Walking2' from left to right and top to bottom.

plots. The performance scores of overlap rate are calculated as the areas under curves in the success plots. Among these trackers, SCM, DSSM, ASLA, ODDL, and L1APG are sparse coding based tracking algorithms, which proves that sparse representation works in visual tracking community. SCM performs the best among the four sparse representation based tracking methods, which gets the second best performance score in success plots. Struck also works well on these video sequences, and it obtains the best performance in precision plots except against other popular trackers. But they fail to respect the underlying manifold geometry of sample space. We note that ODDL also shows comparable tracking results on these video sequences, which demonstrates the dictionary learning contributes to tracking performance. Our tracker performs pretty well on these sequences, and it improves the location error performance score of Struck from 0.654 to 0.737, and the overlap performance score from 0.498 to 0.508. For clarity, we list all performance scores of these trackers in the first two rows of Table 1. It demonstrates that the introduced joint discriminative dictionary and classifier learning algorithm is effective in visual tracking.

To further verify the validity of the proposed joint dictionary, sparse codes and classifier learning algorithm, we compare the proposed tracking method with a version that without dictionary learning (denoted as WDL). The performance scores of WDL are shown in Table 1. From it, we find that the performance of the proposed tracking method is superior to the version without dictionary learning.

Sequences are annotated with different attributes considering common challenges encountered during tracking. To verify the performance of trackers under different challenging situations, we evaluate the performance score of these trackers based on 8 attributes: background clutters, deformation, fast motion, illumination variation, in-plane rotation, motion blur, occlusion, and out-of-plane rotation. As are shown in Figure 4, we present the precision plots of all trackers under these challenges. Our tracker performs best on 6 attributes in precision plots.

We show several frames of part of the tracking results obtained by the proposed approach and other trackers in Figure 5. Our tracker is able to handle illumination variations ('Car4' and 'CarDark' etc.), occlusions ('David3', 'Faceocc1', 'Faceocc2' etc.), scale changes ('Dog1', 'Singer1' etc.) well as shown in these figures.

5. Conclusions

In this paper, we give a principled method to jointly learn an nonlinear classifier, sparse codes and a discriminative dictionary for visual tracking. An optimization algorithm is presented to obtain optimal dictionary, sparse codes and classifier simultaneously. Based on the optimization, a visual tracker is developed under particle filter framework

with online updating strategies adopted to adapt to changing appearance. A group of experiments on challenging video sequences demonstrate the superior performance of the proposed method.

References

[1] M. Aharon, M. Elad, and A. Bruckstein. k-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. on Signal Processing*, 54(11):4311–4322, 2006.

[2] C. Bao, Y. Wu, H. Ling, and H. Ji. Real time robust l1 tracker using accelerated proximal gradient approach. In *IEEE CVPR*, pages 1830–1837. IEEE, 2012.

[3] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE TPAMI*, 32(9):1627–1645, 2010.

[4] J. Gao, H. Ling, W. Hu, and J. Xing. Transfer learning based visual tracking with gaussian process regression. In *ECCV*, pages 188–203. Springer, 2014.

[5] S. Hare, A. Saffari, and P. H. Torr. Struck: Structured output tracking with kernels. In *IEEE ICCV*, pages 263–270. IEEE, 2011.

[6] J. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *ECCV*, pages 702–715. Springer, 2012.

[7] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE TPAMI*, 2015.

[8] H. Hongwei, M. Bo, X. Tao, and P. Junbiao. Nonlinear learning using lcc for online visual tracking. In *IEEE ICME*, pages 1–6. IEEE, 2014.

[9] X. Jia, H. Lu, and M.-H. Yang. Visual tracking via adaptive structural local sparse appearance model. In *IEEE CVPR*, pages 1822–1829. IEEE, 2012.

[10] Z. Jiang, Z. Lin, and L. S. Davis. Learning a discriminative dictionary for sparse coding via label consistent k-svd. In *IEEE CVPR*, pages 1697–1704. IEEE, 2011.

[11] Z. Kalal, J. Matas, and K. Mikolajczyk. P-n learning: Bootstrapping binary classifiers by structural constraints. In *IEEE CVPR*, pages 49–56. IEEE, 2010.

[12] S. Kong and D. Wang. Online discriminative dictionary learning for image classification based on block-coordinate descent method. *Computing Research Repository*, abs/1203.0856, 2012.

[13] J. Kwon and K. M. Lee. Visual tracking decomposition. In *IEEE CVPR*, pages 1269–1276. IEEE, 2010.

[14] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. V. D. Hengel. A survey of appearance models in visual object tracking. *ACM Trans. on Intelligent Systems and Technology*, 4(4):58:1–58:48, 2013.

[15] Y. Lin, F. Lv, S. Zhu, M. Yang, T. Cour, K. Yu, L. Cao, and T. Huang. Large-scale image classification: fast feature extraction and svm training. In *IEEE CVPR*, pages 1689–1696. IEEE, 2011.

[16] B. Liu, J. Huang, L. Yang, and C. Kulikowsk. Robust tracking using local sparse appearance model and k-selection. In *IEEE CVPR*, pages 1313–1320. IEEE, 2011.

[17] X. Lu, Y. Yuan, and P. Yan. Robust visual tracking with discriminative sparse learning. *Pattern Recognition*, 46(7):1762–1771, 2013.

[18] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *IEEE CVPR*, pages 1–8. IEEE, 2008.

[19] X. Mei and H. Ling. Robust visual tracking using l1 minimization. In *IEEE ICCV*, pages 1436–1443. IEEE, 2009.

[20] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan. Locally orderless tracking. In *IEEE CVPR*, pages 1940–1947. IEEE, 2012.

[21] D.-S. Pham and S. Venkatesh. Joint learning and dictionary construction for pattern recognition. In *IEEE CVPR*, pages 1–8. IEEE, 2008.

[22] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *IJCV*, 77(1-3):125–141, 2008.

[23] S. Salti, A. Cavallaro, and L. Di Stefano. Adaptive appearance modeling for video tracking: survey and evaluation. *IEEE TIP*, 21(10):4334–4348, 2012.

[24] D. Wang, H. Lu, and M. Yang. Robust visual tracking via least soft-threshold squares. *IEE TCSVT*, 2015.

[25] D. Wang, H. Lu, and M.-H. Yang. Least soft-thresold squares tracking. In *IEEE CVPR*, pages 2371–2378. IEEE, 2013.

[26] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *IEEE CVPR*, pages 3360–3367. IEEE, 2010.

[27] Q. Wang, F. Chen, W. Xu, and M.-H. Yang. Online discriminative object tracking with local sparse representation. In *IEEE WACV*, pages 425–432. IEEE, 2012.

[28] Q. Wang, F. Chen, J. Yang, W. Xu, and M.-H. Yang. Transferring visual prior for online object tracking. *IEEE TIP*, 21(7):3296–3305, 2012.

[29] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE TPAMI*, 31(2):210–227, 2009.

[30] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *IEEE CVPR*, pages 2411–2418. IEEE, 2013.

[31] B. Xie, M. Song, and D. Tao. Large-scale dictionary learning for local coordinate coding. In *BMVC*, pages 1–9, 2010.

[32] F. Yang, Z. Jiang, and L. S. Davis. Online discriminative dictionary learning for visual tracking. In *IEEE WACV*, pages 854–861. IEEE, 2014.

[33] K. Yu and T. Zhang. Improved local coordinate coding using local tangents. In *ICML*, pages 1215–1222, 2010.

[34] K. Yu, T. Zhang, and Y. Gong. Nonlinear learning using local coordinate coding. In *Advances in NIPS*, volume 22, pages 2223–2231, 2009.

[35] Q. Zhang and B. Li. Discriminative k-svd for dictionary learning in face recognition. In *IEEE CVPR*, pages 2691–2698. IEEE, 2010.

[36] W. Zhong, H. Lu, and M.-H. Yang. Robust object tracking via sparsity-based collaborative model. In *IEEE CVPR*, pages 1838–1845. IEEE, 2012.

[37] B. Zhuang, H. Lu, Z. Xiao, and D. Wang. Visual tracking via discriminative sparse similarity map. *IEEE TIP*, 23(4):1872–1881, 2014.