SIViP (2017) 11:881–888
DOI 10.1007/s11760-016-1035-x

ORIGINAL PAPER

Visual object tracking with online sample selection via lasso regularization

Qiao Liu[1] · Xiao Ma[2] · Weihua Ou[1] · Quan Zhou[3]

Received: 27 July 2016 / Revised: 27 October 2016 / Accepted: 25 November 2016 / Published online: 11 January 2017

Abstract In the past years, discriminative methods are popular in visual tracking. The main idea of the discriminative method is to learn a classifier to distinguish the target from the background. The key step is the update of the classifier. Usually, the tracked results are chosen as the positive samples to update the classifier, which results in the failure of the updating of the classifier when the tracked results are not accurate. After that the tracker will drift away from the target. Additionally, a large number of training samples would hinder the online updating of the classifier without an appropriate sample selection strategy. To address the drift problem, we propose a score function to predict the optimal candidate directly instead of learning a classifier. Furthermore, to solve the problem of a large number of training samples, we design a sparsity-constrained sample selection strategy to choose some representative support samples from the large number of training samples on the updating stage. To evaluate the effectiveness and robustness of the proposed method, we implement experiments on the object tracking benchmark and 12 challenging sequences. The experiment results demonstrate that our approach achieves promising performance.

Keywords Discriminative method · Object drift · Score function · Samples selection · Sparse constraint

1 Introduction

Visual object tracking is an important computer vision problem in real applications, such as surveillance, human computer interaction, vehicle navigation. Several approaches have been proposed in the past years, which can be classified into generative methods [15,19,23,24] and discriminative methods [3,6,10,13,16,18,31]. Generative methods focus on modeling the appearance of the object which might be varied in a different frame. Discriminative methods cast object tracking as a classification problem that distinguishes the tracked target from the background.

Discriminative methods become more popular in the computer vision, i.e., face recognition [12,20,21,29], object tracking [2,8,30], mainly because they do not need to construct a complex appearance model. Some representative discriminative methods have received much attention in recent years. For instances, Kalal et al. [13] proposed a novel tracking framework (TLD) that decomposes the long-term tracking task into tracking, learning and detection. Zhu et al. [31] presented a collaborative correlation tracker (CCT) to deal with the scale variation and the drift problem. Gao and Ling et al. [4] proposed a new transfer

Electronic supplementary material The online version of this article (doi:10.1007/s11760-016-1035-x) contains supplementary material, which is available to authorized users.

✉ Weihua Ou
ouweihuahust@gmail.com

Qiao Liu
liuqiao.hit@gmail.com

Xiao Ma
turingki@yeah.net

Quan Zhou
quan.zhou@njupt.edu.cn

[1] School of Big Data and Computer Science, Guizhou Normal University, Guiyang, China

[2] School of Computer Science, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China

[3] Key Lab of Ministry of Education for Broad Band Communication and Sensor Network Technology, Nanjing University of Posts and Telecommunications, Nanjing, China

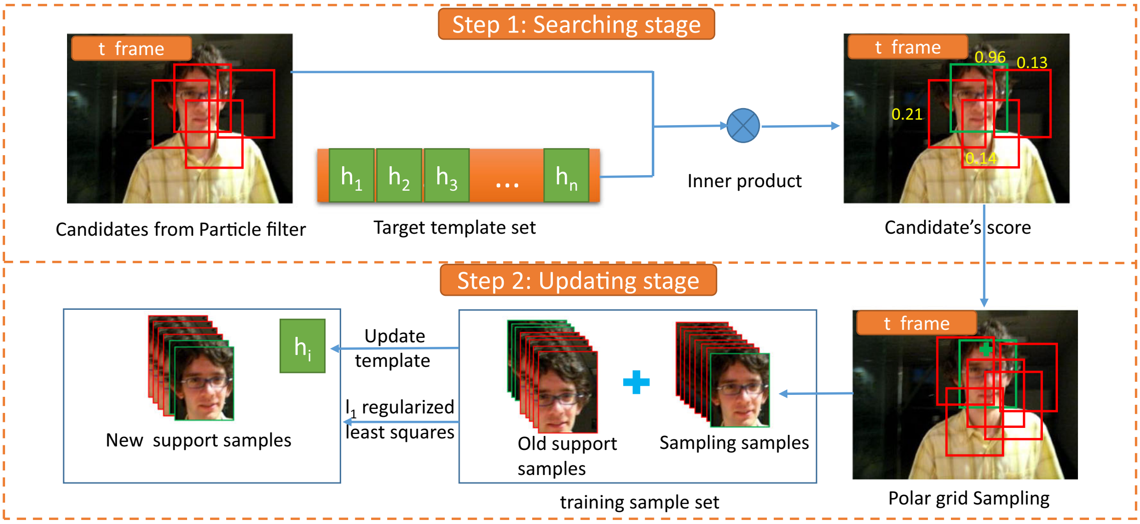


Fig. 1 Illustration of the proposed method. *Searching stage* using the score function to calculate every candidate's score and choosing the maximum one as the optimal candidate. *Updating stage* extracting a set of training samples using polar grid sampling around the currently tracked target; appending these samples into the old representative sample set to form a support sample set; exploiting ℓ_1-regularized least squares to obtain some representative samples and updating the target template. The *green* and *red bounding box* denote the positive sample and negative sample, respectively

learning-based visual tracker to alleviate drift using gaussian processes regression (TGPR). Danelljan et al. [3] proposed a novel approach (DSST) by learning discriminative correlation filters based on a scale pyramid representation in the tracking-by-detection framework. Henriques et al. [10] presented a high-speed kernelized correlation filter (KCF) by using a circulant matrix. All of these methods achieved satisfied performance on the OTB [28] and received much attention from the researchers in recent years.

Generally, discriminative method trains a classifier to identify the object, which heavily depends on the selection of the positive and negative training samples. Most existing discriminative methods regard the currently tracked target as the positive sample and select samples from the neighborhood around the currently tracked target as the negative samples to update the classifier. The classifier will be updated with a sub-optimal positive sample when the currently tracked target is not accurate. After that the tracker would drift in a long time. Additionally, a large number of training samples will hinder the classifier to be updated online in real time. Therefore, it is necessary to design a sample selection strategy for the classifier updating.

Different from the most existing discriminative methods, in this paper, we propose a score function instead of learning a classifier to predict the optimal candidate directly. As shown in step 1 of Fig. 1, we use the similarity of between the candidate which is generated from particle filter and the target template set as our score function and exploit the inner product to measure the similarity. Because our approach does not to update the classifier with a sub-optimal positive sample, thus, it can avoid the drift problem. To address the problem of a large number of training samples, we propose an online sample selection strategy based on ℓ_1-regularized least squares, as shown in step 2 of Fig. 1. We construct a training sample set and calculate the ground truth of it. Then, we minimize the errors of the score function and ground truth to choose some representative support samples for the update of the target template set.

The main contributions of this paper are summarized as follows:

- A simple score function is proposed to predict the optimal candidate directly instead of learning a classifier, which can address the drift problem.
- A sparsity-constrained sample selection method is proposed, through which the representative support samples are chosen to construct the templates.

The rest of this paper is organized as follows. We briefly review the related works in Sect. 2 and describe the proposed approach in Sect. 3. Then, we show the experimental details and results in Sect. 4 and conclude this work in Sect. 5.

2 Related works

In this section, we review the particle filter framework for tracking firstly, because our approach is based on this framework. Then, we briefly introduce sparse representation model for tracking, because the sparse constraint is applied on our approach to solve the tracking problem.

2.1 Particle filter framework

Particle filter [22] is Bayesian sequential importance sampling technique. It provides a general framework for estimating and propagating the posterior probability density function of state variables. During the last years, a large number of popular trackers [1,5,7,11,19] based on this framework are proposed. Our approach also uses the particle filter as motion model (see the candidates from particle filter of Fig. 1).

Given $t-1$ observed patches $\mathbf{I}_{1:t-1} = \{\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_{t-1}\}$ from the first frame to the $t-1$ frame. Let $\mathbf{b}_t = [x, y, w, h] \in \mathbb{R}^4$ be the state variables in t^{th} frame, where (x, y) are the coordinates of the center point of bounding box and w, h are the width and height of the bounding box, respectively. The state variables $\mathbf{b}_t$ can be formulated by the following predicting distribution:

$$p(\mathbf{b}_t|\mathbf{I}_{1:t-1}) = \int p(\mathbf{b}_t|\mathbf{b}_{t-1})p(\mathbf{b}_{t-1}|\mathbf{I}_{1:t-1})d\mathbf{b}_{t-1}. \tag{1}$$

Given the observed patch $\mathbf{I}_t$ in frame t, the state variables $\mathbf{b}_t$ can be updated by the following formulation:

$$p(\mathbf{b}_t|\mathbf{I}_{1:t}) = \frac{p(\mathbf{I}_t|\mathbf{b}_t)p(\mathbf{b}_t|\mathbf{I}_{1:t-1})}{p(\mathbf{I}_t|\mathbf{I}_{1:t-1})}, \tag{2}$$

where $p(\mathbf{I}_t|\mathbf{b}_t)$ denotes the observation model.

The observation model $p(\mathbf{I}_t|\mathbf{b}_t)$ represents the similarity between a target candidate and the target template. For an observed patch $\mathbf{I}_t$, we use $\mathbf{x}_t$ to represent the features extracted from $\mathbf{I}_t$. We introduce a score function of $\mathbf{x}_t$ to approximate $p(\mathbf{I}_t|\mathbf{b}_t)$:

$$p(\mathbf{I}_t|\mathbf{b}_t) \propto F(\mathbf{x}_t). \tag{3}$$

This function is defined as a simple inner product between the candidate and the target template (see Sect. 3.1). The optimal candidate state is the one with the biggest score value.

2.2 Sparse representation-based tracking

Sparse representation has been applied to visual tracking [1,7,11,17,19,25,27] to find the target with the minimum reconstruction error from the target template subspace. These methods can be classified as two categories: holistic sparse representation [1,19,25,26] and local sparse representation [7,11,17]. In the first class, Mei et al. [19] cast the tracking problem as finding a sparse approximation in a template subspace. They adopt the holistic representation of the object as the appearance model and, then, track the object by solving the ℓ_1 minimization problem (ℓ_1 tracker). To address the bottleneck of the computational cost of the ℓ_1 tracker, Bao et al. [1] proposed a new ℓ_1 norm-related minimization model based on the accelerated proximal gradient approach (ℓ_1-APG) which can run in real time. This category of methods can handle the partial occlusion and slight deformation effectively.

In contrast to the holistic sparse representation, the local sparse representation encodes the each local patch of a target sparsely with an over-complete dictionary and, then, aggregate the corresponding sparse codes. For instances, Jia et al. [11] proposed a structural local sparse appearance model which exploits both partial information and spatial information of the target based on a novel alignment-pooling method. Liu et al. [17] also presented a robust tracking algorithm using a local sparse appearance model, which used a static sparse dictionary and a dynamically online-updated basis distribution to model the target appearance. Because the local sparse representation can exploit the structural information of the object, it can better deal with the occlusion and deformation. However, it is more complicated and has the higher computational expense. In this paper, we impose a sparse constraint on the score function and we apply holistic sparse representation to solve the tracking problem.

3 Proposed approach

In this section, we give the details of the proposed approach which includes four parts primarily. Specifically, we present the score function in Sect. 3.1 and propose the online sample selection in Sect. 3.2. Then, we give the template updating strategy in Sect. 3.3. Finally, the whole algorithm is summarized in Sect. 3.4.

3.1 Score function

The aim of score function is to predict which candidate is the optimal one. Given a target template set $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n\} \in \mathbb{R}^{d\times n}$ and a target candidate $\mathbf{x}$ in the tth frame, where $\mathbf{x} \in \mathbb{R}^d$ is the HOG feature vector extracting from the target candidate . We use a simple inner product to measure the similarity between the candidate and the target template as a part of the score function:

$$f_i(\mathbf{x}) = \langle \mathbf{x}, \mathbf{h}_i \rangle, \tag{4}$$

where $\mathbf{x}$ and $\mathbf{h}_i$ are normalized, i.e., $\|\mathbf{x}\|_2 = 1$, $\|\mathbf{h}_i\|_2 = 1$. For a candidate $\mathbf{x}$, the larger the value of score function $f(\mathbf{x})$ is, the higher the similarity between the candidate and the target template has. However, we explore the target template set $\mathbf{H}$ which is made up of several templates, rather than a single target template. Therefore, the average score of the similarity between the candidate and each target template in $\mathbf{H}$ is adopted as the final score function $F(\mathbf{x})$.

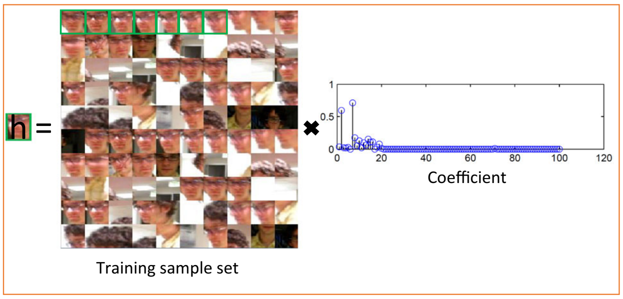


Fig. 2 Illustration of the observation. The *bounding box* with *green line* denotes the positive sample in the training sample set and the other are negatives

$$F(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}). \tag{5}$$

For the given target template $\mathbf{h}_i$ in the tth frame, let $\mathbf{A}_i = [\mathbf{x}_i^1, \mathbf{x}_i^2, \ldots, \mathbf{x}_i^m] \in \mathbb{R}^{d\times m}$ be the corresponding training sample set which consists of the support sample set $\mathbf{A}_{t-1}$ and updating sample set $\mathbf{X}_t$, where $\mathbf{x}_i^j$, $j = 1, 2, \ldots, m$ indicates the jth training sample of the $\mathbf{A}_i$, $\mathbf{A}_{t-1}$ is the old support sample set in the $(t-1)$th frame and $\mathbf{X}_t$ is updating sample set which is sampled from the currently tracked target in the tth frame. As we know, each template can be linearly represented by the training sample set, i.e.,

$$\mathbf{h}_i = \mathbf{A}_i \boldsymbol{\omega}_i, \tag{6}$$

where $\boldsymbol{\omega}_i$ is the coefficient vector of the $\mathbf{A}_i$.

In a real application, the appearance of the object target is very similar to the certain adjacent frames. Therefore, the target template of these object targets can be sparsely represented by a few positive and negative support samples in these adjacent frames, as shown in Fig. 2. Based on this fact, for a target template, just a few representative support samples are needed to represent it. Therefore, we impose a sparsity constraint on the coefficient vector $\boldsymbol{\omega}_i$ and reformulate Eq. (6) below:

$$\mathbf{h}_i \propto \mathbf{A}_i \boldsymbol{\omega}_i \quad s.t. \quad \|\boldsymbol{\omega}_i\|_0 \leq \alpha, \tag{7}$$

where α is a threshold value.

Then, substituting Eq. (7) into Eq. (4), we obtain the following score function:

$$\begin{aligned} f_i(\mathbf{x}) &= \langle \mathbf{x}, \mathbf{A}_i \boldsymbol{\omega}_i \rangle \\ &= \sum_{j=1}^{m} \omega_i^j \langle \mathbf{x}, \mathbf{x}_i^j \rangle \qquad s.t. \quad \|\boldsymbol{\omega}_i\|_0 \leq \alpha, \end{aligned} \tag{8}$$

where $\boldsymbol{\omega}_i = [\omega_i^1, \omega_i^2, \ldots, \omega_i^m]^{\mathrm{T}} \in \mathbb{R}^m$ $(\omega_i^j, j = 1, 2, \ldots, m)$ denotes the coefficient of jth inner product.

3.2 Online sample selection via ℓ_1-regularized least squares

Given a target in the tth frame, we exploit the polar grid sampling to obtain an updating sample set $\mathbf{X}_t = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k] \in \mathbb{R}^{d\times k}$, where $\mathbf{x}_r \in \mathbf{R}^d, r = 1, 2, \ldots, k$ denotes the rth training sample of $\mathbf{X}_t$. For each training sample $\mathbf{x}_r$, we define a function to calculate its ground truth,

$$g(\mathbf{x}_r) = \frac{\text{overlap}(b, \text{box}(\mathbf{x}_r))}{b}, \tag{9}$$

where $g(\mathbf{x}_r)$ is normalized 0 to 1, b represents the bounding box area of the currently tracked target, $\text{box}(\mathbf{x}_r)$ indicates the bounding box area of the training sample $\mathbf{x}_r$, and overlap(,) calculates the overlap area of the two bounding boxes. Therefore, for the training sample set $\mathbf{X}_t$, we can obtain its ground truth $\mathbf{y}$ using Eq. (9), where $\mathbf{y} = [y_1, y_2, \ldots, y_k]^{\mathrm{T}} = [g(\mathbf{x}_1), g(\mathbf{x}_2), \ldots, g(\mathbf{x}_k)]^T \in \mathbb{R}^k$.

For the old support sample set $\mathbf{A}_{t-1} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_u] \in \mathbb{R}^{d\times u}$ in the $(t-1)$th frame, we also can get its corresponding ground truth $\mathbf{s} = [s_1, s_2, \ldots, s_u]^{\mathrm{T}} \in \mathbb{R}^m$ using Eq. (9). Combining the old support sample set $\mathbf{A}_{t-1}$ and updating sample set $\mathbf{X}_t$, we get the training sample set $\mathbf{A}_i = [\mathbf{A}_{t-1}, \mathbf{X}_t] \in \mathbb{R}^{d\times(k+u)}$ and the associated ground truth $\mathbf{q} = [\mathbf{y}, \mathbf{s}] \in \mathbb{R}^{(k+u)}$, we call the training sample set $\mathbf{A}_i \in \mathbb{R}^{d\times m}$ is the candidate support sample set , where $m = k + u$.

Suppose that the candidate support sample in $\mathbf{A}_i$ is normalized. we get coefficient $\boldsymbol{\omega}_i$ in the tth frame by minimizing following objective function:

$$\min_{\boldsymbol{\omega}_i} \sum_{\ell=1}^{m} \left(q_\ell - \sum_{j=1}^{m} \omega_i^j \langle \mathbf{x}, \mathbf{x}_i^j \rangle \right)^2 + \lambda \|\boldsymbol{\omega}_i\|_1, \tag{10}$$

where $q_\ell \in \mathbb{R}$ denotes the ℓth ground truth of the $\mathbf{q}$ and λ is regularization parameter. Utilizing matrix notation, Eq. (10) can be reformulated as following:

$$\min_{\boldsymbol{\omega}_i} \left\| \mathbf{q} - (\mathbf{A}_i^T \mathbf{A}_i \boldsymbol{\omega}_i) \right\|_2^2 + \lambda \|\boldsymbol{\omega}_i\|_1, \tag{11}$$

Let $\mathbf{D} = \mathbf{A}_i^T \mathbf{A}_i$, then Eq. (11) can be simplified as below:

$$\min_{\boldsymbol{\omega}_i} \|\mathbf{q} - \mathbf{D}\boldsymbol{\omega}_i\|_2^2 + \lambda \|\boldsymbol{\omega}_i\|_1. \tag{12}$$

Equation (12) can be solved by ℓ_1-regularized least squares [14]. Then, we choose the corresponding samples which the coefficient are greater than the predefined threshold σ as new support samples. The new support sample set $\mathbf{A}_t$ in the tth frame is constituted by all these support samples.

3.3 Template updating

Template updating is a very important step in updating stage of visual tracking. If the template set $\mathbf{H}$ is fixed, the tracker will be failed because the target's appearance changes dynamically. However, if the template set $\mathbf{H}$ is updated too frequently, the errors would be accumulated and the tracker would drift away from the target.

In our method, we adopt a discriminative strategy to update the template set. For a given target, if its score is greater than the predefined threshold θ, we can obtain the vector of coefficient $\boldsymbol{\omega}_i$ and new support sample set $\mathbf{A}_t$ by solving the problem (10). Based on the assumption the target template can be represented by the linear combination of some representative support samples, the template $\mathbf{h}_i$ can be updated by Eq. (6). If the template number in the template set $\mathbf{H}$ is below the given threshold η, we put this new template $\mathbf{h}_i$ into the template set $\mathbf{H}$. Otherwise, this new template will be appended into $\mathbf{H}$ and the oldest template will be discarded.

3.4 Algorithm

The proposed method is described in Algorithm 1 and the details of the algorithm implementing will be given in Sect. 4.1. The overall algorithm includes searching stage and updating stage. In searching stage, the optimal candidate is obtained using the score function. In updating stage, some representative samples are chosen by ℓ_1-regularized least squares and the target template set is updated using the selected samples.

Algorithm 1 Visual Tracking with Online Sample Selection via ℓ_1 regularization

```
Inputs: Testing sequence $\psi = \{I_0, I_1, ..., I_F\}$ and initial state $b_0$.
Outputs: The predicted optimal states $\{b_1, b_2, ..., b_F\}$.
Predefine template set $\mathbf{H}$ and threshold $\theta$.
for $t = 1$ to $F$ do
    Searching stage:
    Generate $M$ candidate samples by exploiting particle filter.
    for $i = 1$ to $M$ do
        Calculate the score of every candidate by score function $F(\mathbf{x})$.
        Get $Max(F(\mathbf{x}))$ as an optimal candidate.
    end for
    Updating stage:
    if $Max(F(\mathbf{x})) > \theta$ then
        Save the currently tracked target as the positive sample and
        draw the negative samples by polar grid sampling.
        Get the corresponding coefficient vector $\boldsymbol{\omega}_i$ by solving Eq. (10).
        Construct the new target template $\mathbf{h}_i$ by Eq. (6) and append $\mathbf{h}_i$
        to $\mathbf{H}$.
    end if
end for
```

4 Experiments

In this section, we first introduce the experimental implementation details in Sect. 4.1. it includes the parameter setting, datasets, comparison tracker and evaluation. Then, we give the experiment results and analyze in Sect. 4.2.

4.1 Experiment details

Parameter setting: All the methods are carried out in MATLAB 2014a on a PC with an Intel 3.7 GHz Dual Core CPU and 8 GB RAM. The image patches are resized to 32 × 32 pixels during the tracking process. For the HOG feature, the cell size and the number of orientation bins are set to 4 and 9, respectively. In the updating stage, the radius of the polar grid and the number of angular division are set to 5 and 16, respectively. The other mentioned parameters of the paper are listed as follows.

Parameter name	M	k	λ	σ	θ	η
Parameter value	750	81	0.3	0.01	0.32	160

M denotes the number of the candidates generated from the particle filter, which is set to 750. k is the number of the training samples that sampling by the polar grid and is set to 81. λ is a regularization parameter which represents the sparsity degree of the coefficient vector $\boldsymbol{\omega}_i$ and is set to 0.3. σ denotes the threshold of the coefficient vector and is set to 0.01. We choose the corresponding samples as representative samples of which the coefficient is greater than the threshold σ. θ is a threshold of the score for the update of support sample set and target template, which is set to 0.32. η is the maximum number of the template in target template set and is set to 160.

Datasets Our experiments are carried out on the OTB [28] that contains 50 image sequences. These image sequences have 11 attributes (illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out-of-view, background clutters and low resolution), which represents the challenging aspects of visual tracking. We also choose 12 challenging sequences in these 50 image sequences to qualitatively evaluate our approach. They are Dudek, jogging-1, jogging-2, Suv, FleetFace, Freeman3, Freeman4, Lemining, Sylvester, Tiger2, woman and Walking2.

Comparison tracker In order to examine the performance of the proposed approach, 8-state-of-the-art trackers which have a superior performance on the OTB are chosen to compare with ours. They are CCT [31], DSST [3], TGPR [4], KCF [10], Struck [6], SVM [27], RR [27] and TLD [13].

Evaluation criterion Two criteria are used to evaluate the performance of our approach. One of the widely used criteria is center location error (CLE), which is the average Euclidean distance between the center locations of the tracked targets and the manually labeled ground truths. We use the precision [9] to measure the overall tracking performance, which is defined as the percentage of frames whose estimated location is within the given threshold distance of the ground truth. Usually, this threshold distance is set to 20 pixels.

Another evaluation criterion is the Pascal VOC overlap ratio (VOR) [28], which is defined as $S = |r_t \cap r_a|/|r_t \cup r_a|$, where r_t and r_a represent the bounding box of the tracked target and the ground truth, respectively; $\cap$ and $\cup$ represent the intersection and union of two regions, respectively; $|\cdot|$ denotes the number of the pixel in the region. In order to measure the overall performance on a given image sequence, we count the number of successful frames, whose VOR is larger than the given threshold 0.5. The success plot shows the ratios of successful frames at the thresholds varied from 0 to 1. We use the area under the curve (AUC) of each success plot to rank the comparison trackers.

4.2 Experiment results and analyses

Two groups of experiments are carried out to quantitatively and qualitatively evaluate the proposed approach. The first group is performed on OTB [28] which contains 50 image sequences. We use this group experiments to quantitatively evaluate the overall performance of our tracker and to compare with the other 8 state-of-the-art trackers. Another group experiments are carried out on 12 challenging sequences to qualitatively evaluate our tracker mainly.

Quantitative evaluation The overall performance of our tracker and other 8 compared trackers are shown as in Fig. 3. We use one pass evaluation (OPE) for the overall performance, and precision and success rate as an evaluation criterion, as shown in Fig. 3a, b, respectively. Obviously, the overall performance of our tracker outperforms the other 8 state-of-the-art trackers. What is more, we divide 50 image sequences into different groups according to the different attributes of the image sequences (see datasets of Sect. 4.1). Then, we also use precision and success rate to evaluate the performance of the tracker on different attributes. Due to space limitations, ten groups precision and success plots on 10 different attributes are provided in the supplemental material. The results also demonstrate that our tracker is clearly more accurate and robust.

For better illustrate the proposed method is effective. We give the precision and success rate on another 12 challenging sequences more detailedly, as shown in Table 1. From Table 1 we can see clearly that our approach has a better performance on most challenging sequences. For instances, our tracker achieved the precision score with 0.99 on jogging-2 which has fully occlusion challenge with a short time , while the CCT [31], DSST [3], KCF [10] just obtained 0.19, 0.19, 0.16, respectively. Lemming is a challenging sequence which has occlusion and deformation et al. challenges, and our tracker also achieved the highest score 0.93 while the Struck [6], KCF [10] and DSST [3] obtained 0.50, 0.49, 0.43, respectively. The average precision score of our tracker has improved 30% than that of the second best tracker CCT [31]. It is also obviously that our tracker has achieved the best success rate, and it average success rate of the proposed tracker also has improved 30% than the second best tracker CCT [31].

Qualitative evaluation The second group experiments are carried out on 12 challenging sequences to evaluate the proposed approach more intuitive. Due to space constraints, we just give the center location error (CLE) of the frame by frame on 6 challenging sequences, as shown in Fig. 4a–f. More results are provided in the supplemental material. From Fig. 4, we can see clearly that our tracker has the lowest center location error on the most frames of the most challenging sequences. Specifically, just like jogging-1 (Fig. 4b) and jogging-2 (Fig. 4c), the CLE of our tracker is lower than

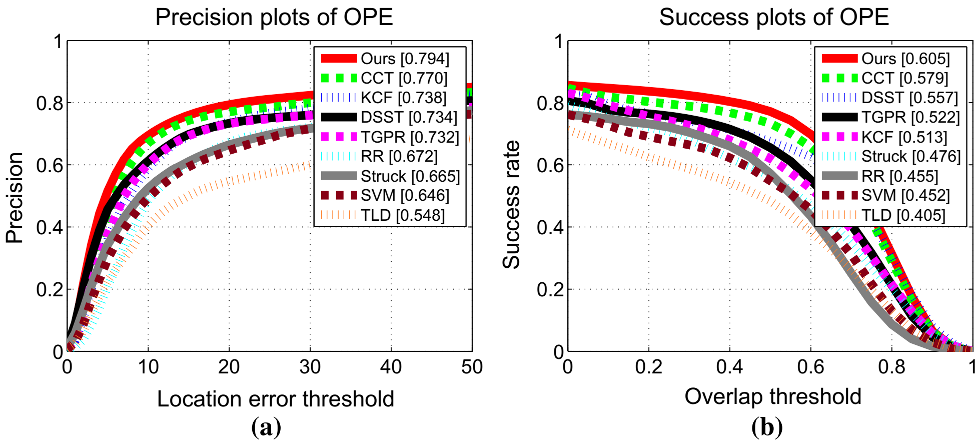
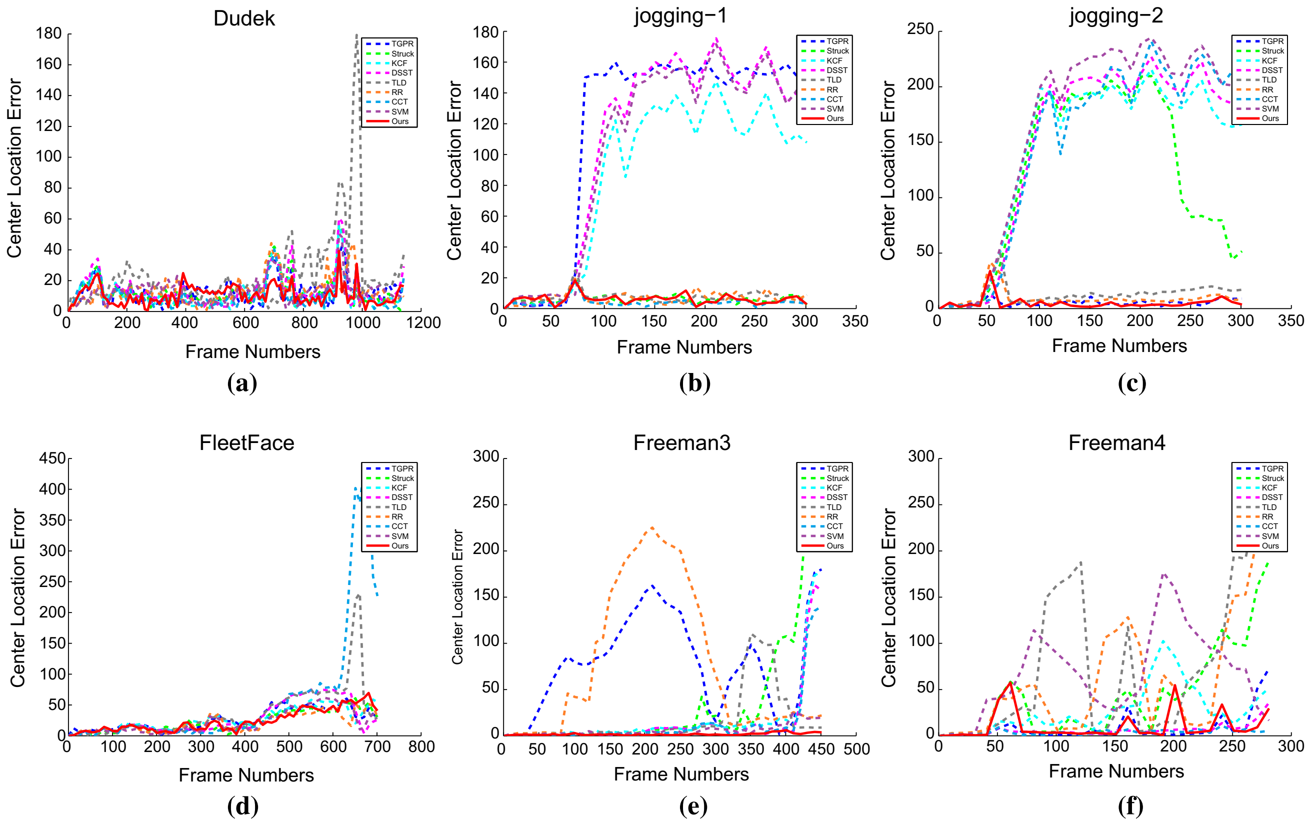


Fig. 3 Comparison with eight state-of-the-art trackers on 50 image sequences of the precision and success plots using one pass evaluation (OPE)

Table 1 Percentage of successful frames whose center location error (CLE) within the threshold 20 pixels and the percentage of successful frames whose overlap ratio (VOR) passes the threshold 0.5

	TGPR	Struck	KCF	DSST	TLD	RR	CCT	SVM	Ours
Dudek	0.87/0.88	0.86/0.97	0.88/0.98	0.82/*0.99*	0.64/0.67	0.85/0.87	*0.90*/**1.00**	0.89/0.96	**0.91**/0.97
jogging-1	0.22/0.22	0.97/0.90	0.23/0.22	0.23/0.22	0.97/0.96	0.98/0.96	*0.98/0.97*	0.24/0.23	**0.98/0.97**
jogging-2	*0.99*/**0.99**	0.18/0.16	0.16/0.16	0.19/0.18	0.95/0.95	0.97/0.96	0.19/0.19	0.16/0.14	0.99/*0.98*
Suv	0.53/0.54	0.18/0.16	0.98/**0.98**	0.98/0.98	0.94/0.94	0.96/0.92	*0.98*/0.98	0.52/0.53	**0.98**/*0.98*
Fleetface	0.50/0.59	0.57/**0.83**	0.46/0.67	0.62/0.70	0.48/0.44	0.57/0.69	0.61/0.67	*0.62*/0.66	**0.64**/*0.77*
Freeman3	0.18/0.08	0.67/0.33	0.91/0.27	0.91/0.33	0.83/*0.65*	0.39/0.15	0.91/0.32	*0.92*/0.43	**0.97/0.95**
Freeman4	0.90/*0.74*	0.41/0.24	0.53/0.18	0.96/0.44	0.37/0.22	0.45/0.10	**1.00**/0.63	0.16/0.12	*0.94*/**0.88**
Lemming	0.45/0.37	0.50/0.48	0.49/0.43	0.43/0.27	0.80/0.63	0.60/0.57	0.70/0.70	*0.82/0.77*	**0.93/0.93**
Sylvester	0.96/**0.95**	**0.99**/*0.93*	0.84/0.82	0.84/0.74	0.91/0.85	0.88/0.32	0.85/0.80	0.93/0.66	*0.97*/0.92
Tiger2	0.86/*0.89*	0.43/0.43	0.36/0.36	0.30/0.30	0.35/0.20	0.47/0.23	*0.86*/0.87	0.42/0.28	**0.93/0.93**
Woman	0.94/0.94	**1.00**/0.94	0.94/*0.94*	0.94/0.93	0.40/0.33	0.34/0.29	0.20/0.20	0.97/0.19	*0.97*/**0.94**
Walking2	1.00/0.74	0.71/0.41	0.43/0.38	**1.00/1.00**	0.56/0.21	0.97/0.97	*1.00/1.00*	0.91/0.41	0.97/0.97
Average	0.70/0.66	0.62/0.57	0.60/0.53	0.68/0.59	0.69/0.59	0.70/0.59	*0.76/ 0.69*	0.62/0.45	**0.93/0.93**

The best result is highlighted in bold and the second best result is highlighted in italics and the average value follows in the end



(a) (b) (c) (d) (e) (f)

Fig. 4 Comparison with eight different trackers in center location error (CLE) of frame by frame on 6 challenging sequences

CCT [31], DSST [3] and TGPR [4] when the full occlusion happened. When the appearance changed slight quickly, the CLE of our tracker is also lower than most other trackers, as shown in Fleetface (Fig. 4d), Freeman3 (Fig. 4e) and Freeman4 (Fig. 4f). The other CLE results can also demonstrate that our tracker outperforms the other eight state-of-the-art trackers. For better understanding the proposed approach achieved promising performance, the tracked results of the trackers in some representative frames are listed in the supplemental material. These tracked results also indicated that our tracker is effective and robust.

5 Conclusion

In this paper, we proposed a score function to predict the optimal candidate directly instead of learning a classifier. Exploiting the score function can avoid the drift problem. Moreover, to solve the problem of a large number of training samples, we impose a sparse constraint on the score function and use ℓ_1-regularized least squares to choose some representative support samples. Then, to evaluate the effectiveness and robustness of the proposed approach, we carry out two groups experiments on the OTB [28] and another 12 challenging sequences. Both quantitative and qualitative evaluations are performed to validate our approach, and experimental results demonstrate that the proposed approach achieved promising performance.

Acknowledgements This work is supported by the National Nature Science Foundation of China (No. 61402122), the 2014 Ph.D. Recruitment Program of Guizhou Normal University, the Outstanding Innovation Talents of Science and Technology Award Scheme of Education Department in Guizhou Province (Qianjiao KY word [2015]487).

References

1. Bao, C., Wu, Y., Ling, H., Ji, H.: Real time robust l1 tracker using accelerated proximal gradient approach. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1830–1837. IEEE (2012)
2. Chen, Z., You, X., Zhong, B., Li, J., Tao, D.: Dynamically Modulated Mask Sparse Tracking. IEEE Trans. Cybern. doi:10.1109/TCYB.2016.2577718
3. Danelljan, M., Häger ,G., Khan, F., Felsberg, M.: Accurate scale estimation for robust visual tracking. In: British Machine Vision Conference, Nottingham, pp. 1–11 (2014)
4. Gao, J., Ling, H., Hu, W., Xing, J.: Transfer learning based visual tracking with gaussian processes regression. In: Computer Vision—ECCV, pp. 188–203 (2014)
5. Han, Z., Jiao, J., Zhang, B., Ye, Q., Liu, J.: Visual object tracking via sample-based adaptive sparse representation (adasr). Pattern Recognit. **44**(9), 2170–2183 (2011)
6. Hare, S., Saffari, A., Torr, P.H.S.: Struck: structured output tracking with kernels. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 263–270 (2011)
7. He, Z., Yi, S., Cheung, Y.-M., You, X., Tang, Y.Y.: Robust object tracking via key patch sparse representation. IEEE Trans. Cybern. 1–11 (2016). doi:10.1109/TCYB.2016.2514714
8. He, Z., Li, X., You, X., Tao, D., Tang, Y.Y.: Connected Component Model for Multi-Object Tracking. IEEE Transactions on Image Processing. **25**(8), 3698–3711 (2016)
9. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In: European Conference on Computer Vision, pp. 702–715 (2012)
10. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. **37**(3), 583–596 (2015)
11. Jia, X., Lu, H., Yang, M.-H.: Visual tracking via adaptive structural local sparse appearance model. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1822–1829. IEEE (2012)
12. Jing, X.-Y., Wu, F., Zhu, X., Dong, X., Ma, F., Li, Z.: Multi-spectral low-rank structured dictionary learning for face recognition. Pattern Recogn. **59**, 14–25 (2016)
13. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. IEEE Trans. Pattern Anal. Mach. Intell. **34**(7), 1409–1422 (2012)
14. Koh, K., Kim, S., Boyd, S.: l1 ls: a matlab solver for large-scale l1-regularized least squares problems. Stanford University, pp. 1–6 (2007)
15. Kwon, J., Lee, K.M.: Visual tracking decomposition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1269–1276 (2010)
16. Li, X., Liu, Q., He, Z., Wang, H., Zhang, C., Chen, W.-S.: A multi-view model for visual tracking via correlation filters. Knowl.-Based Syst. **113**, 88–99 (2016)
17. Liu, B., Huang, J., Yang, L., Kulikowsk, C.: Robust tracking using local sparse appearance model and k-selection. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1313–1320. IEEE (2011)
18. Ma, X., Liu, Q., He, Z., Zhang, X., Chen, W.-S.: Visual tracking via exemplar regression model. Knowl.-Based Syst. **106**, 26–37 (2016)
19. Mei, X., Ling, H.: Robust visual tracking using l1 minimization. In: IEEE 12th International Conference on Computer Vision, pp. 1436–1443 (2009)
20. Ou, W., You, X., Tao, D., Zhang, P., Tang, Y., Zhu, Z.: Robust face recognition via occlusion dictionary learning. Pattern Recogn. **47**(4), 1559–1572 (2014)
21. Ou, W., Yu, S., Li, G., Lu, J., Zhang, K.: Xie, G: Multi-view non-negative matrix factorization by patch alignment framework with view consistency. Neurocomputing **204**, 116–124 (2016)
22. Ristic, B., Arulampalam, S., Gordon, N.: Beyond the Kalman Filter: Particle Filters for Tracking Applications, vol. 685. Artech House, Boston (2004)
23. Ross, D.A., Lim, J., Lin, R.-S., Yang, M.-H.: Incremental learning for robust visual tracking. Int. J. Comput. Vis. **77**(1–3), 125–141 (2008)
24. Santner, J., Leistner, C., Saffari, A., Pock, T., Bischof, H.: Prost: Parallel robust online simple tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 723–730 (2010)
25. Shan, D., Zhang, C.: Visual tracking using ipca and sparse representation. SIViP **9**(4), 913–921 (2015)
26. Wang, X., Wang, Y., Wan, W., Hwang, J.-N.: Object tracking with sparse representation and annealed particle filter. SIViP **8**(6), 1059–1068 (2014)
27. Wang, N., Shi, J., Yeung, D.-Y., Jia, J.: Understanding and diagnosing visual tracking systems. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3101–3109 (2015)
28. Wu, Y., Lim, J., Yang, M.-H.: Online object tracking: a benchmark. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2411–2418 (2013)
29. Wu, F., Jing, X.-Y., You, X., Yue, D., Hu, R., Yang, J.-Y.: Multi-view low-rank dictionary learning for image classification. Pattern Recogn. **50**, 143–154 (2016)
30. Yi, S., Lai, Z., He, Z., Cheung, Y.-M., Liu, Y.: Joint sparse principal component analysis. Pattern Recogn. **61**, 524–536 (2017)
31. Zhu, G., Wang, J., Wu, Y., Lu, H.: Collaborative correlation tracking. In: Proceedings of British Machine Vision Conference, pp. 184.1–184.12 (2015)