Digital Signal Processing 72 (2018) 115–125

Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

Visual tracking using global sparse coding and local convolutional features

Xianyou Zeng [a,b], Long Xu [c,*], Lin Ma [d], Ruizhen Zhao [a,b], Yigang Cen [a,b]

[a] Institute of Information Science, Beijing Jiaotong University, Beijing, 100044, China
[b] Key Laboratory of Advanced Information Science and Network Technology of Beijing, Beijing, China
[c] Key Laboratory of Solar Activity, National Astronomical Observatories, Chinese Academy of Sciences, Beijing, 100012, China
[d] Tencent AI Lab, Shenzhen, 518052, China

ARTICLE INFO

Article history:
Available online 19 October 2017

Keywords:
Visual tracking
Sparse representation
Local convolutional feature
Collaborative model

ABSTRACT

Visual tracking is a challenging task in many computer vision applications due to factors such as occlusion, scale variations, background clutter, and so on. In this paper, we present a robust tracking algorithm by representing the target at two levels: global and local levels. Accordingly, the tracking algorithm is composed of two parts: global and local parts. The global part is a discriminative model which separates the foreground object from the background based on holistic features. In the local part, we explore the target's local representation by a set of filters convolving the target region at each position. Then, the global part and local part are integrated into a collaborative model to construct the final tracker. Experiments on the tracking benchmark dataset with 50 challenging videos demonstrate the robustness and effectiveness of the proposed algorithm, outperforming several state-of-the-art models.

1. Introduction

Visual tracking is one of the most important research topics in multimedia processing and has been widely used in human behavior analysis, video surveillance, security, military, transportation, aerospace, and so on. Although significant progress has been made in the past years [1–7], tracking still remains a challenging task, since only ground-truth in the first frame can be used and the target may undergo many challenges, such as illumination change, partial occlusion, pose variation, and shape deformation, etc. All these challenges may result in failed tracking.

To address the above challenges for robust tracking, various representation schemes are introduced into tracking task, such as pixel-based tracker [38], feature-based trackers (e.g. Haar-like features [2,4,5], HOG descriptors [7,8]), sparse-based trackers [10, 31–37], subspace representation based trackers [29,30] and multi-level quantization tracker [39].

As to feature-based trackers, prior approaches [2–11] focus on exploiting hand-crafted features to describe the target. However, these hand-crafted features are designed for certain scenarios. Thus, they cannot be generalized for all generic objects. Traditionally, hand-crafted features are followed by support vector machine (SVM) [12–14] to accomplish vision tasks like classification and recognition. Recently, deep networks trained on large scale dataset of image classification (e.g. Ref. [15]) can directly learn features from raw data instead of hand-crafted features, and have demonstrated great success in many vision tasks, such as object recognition [16], object detection [17], detection and segmentation [18], and image classification [19]. Existing methods have also explored the usage of deep networks for visual tracking. Li et al. [20] incorporate a three-layer convolutional neural network (CNN) trained on-line for visual tracking. Zhou et al. [21] combine an ensemble of deep networks for visual tracking. However, the two methods have not demonstrated excellent results due to the lack of sufficient training samples. To overcome the difficulty caused by a limited amount of training samples, many researchers try to adopt a transfer learning method by first pre-training a deep network with a large number of auxiliary data and then transferring the pre-trained model to online visual tracking. Wang et al. [22] train a stacked denoising autoencoder on an auxiliary tiny image data set to learn generic feature and then employ it for online tracking. Fan et al. [23] develop a human tracking algorithm that uses fully convolutional network to learn a specific feature extractor. Hong et al. [24] use pre-trained CNN features to construct target-specific saliency maps for online tracking. All these methods treat deep networks as black-box feature extractors. Zhang et al. [25] recently present a simple two-layer convolutional network based

* Corresponding author.
E-mail address: lxu@nao.cas.cn (L. Xu).

https://doi.org/10.1016/j.dsp.2017.10.007
1051-2004/

tracker that does not need to be pre-trained with a large amount of auxiliary data and has fully taken into account the similar local structural and inner geometric information among the targets over consequent frames. However, it only exploits the local information of the target, while ignores the holistic information provided by the target.

In this paper, we aim to integrate the advantages of both holistic and local information from the target. Thus, the proposed algorithm is made up of global part and local part. In the global part, we establish a positive dictionary using samples close to the target, while samples away from the target are used to establish a negative dictionary. Then, global confidences are obtained based on sparse reconstruction errors by using the positive and negative dictionaries, respectively. In the local part, we extract local representation by using a bank of filters convolving each candidate and compute local similarities between the candidates and target. Finally, the optimal location is estimated by maximizing the collaborative probability which combines global confidence and local similarity. The main contributions of this paper are summarized as follows:

(1) We define the global confidences through sparse reconstruction errors with positive and negative dictionaries, which can distinguish the target from the background clutter accurately.
(2) A collaborative tracking model is proposed to utilize both holistic and local information of the target.
(3) Our method is performed on the tracking benchmark dataset with 50 challenging videos [26] and achieves favorable results compared with state-of-the-art methods.

The rest of this paper is organized as follows. Section 2 introduces the related work. Section 3 describes the proposed tracker in details. Section 4 presents the quantitative and qualitative comparisons between the proposed tracker and some state-of-the-art trackers. Section 5 concludes this paper.

2. Related work

There are rich literatures in object tracking and good reviews can be found in Refs. [26–28]. Here, we present some most relevant work which motivates our visual tracking research.

There are plentiful methods that focus on designing effective appearance representations. The holistic templates have been widely used in visual tracking. Ross et al. [29] propose the IVT method and introduce a low dimensional PCA subspace to handle appearance variations, which has been further improved in Ref. [30]. Mei and Ling [31] present a l_1 tracking method and utilize a generative sparse representation of templates to account for occlusion. However, heavy computational overhead in this approach hampers its tracking speed. Very recent efforts have been made to improve the l_1 tracker in terms of speed and accuracy in Refs. [32] and [33].

Meanwhile, the local templates have attracted much attention due to their robustness to partial occlusion and deformation. Adam et al. [1] use a set of local image patch histograms to represent a target object. Liu et al. [34] represent a target object with the histograms of sparse coding of local patches. In Ref. [35], Wang et al. encode the local patches inside the target region and concatenate the sparse codes of these patches to represent the target. Zhang et al. [25] utilize a bank of filters to convolve the target region at each position to extract useful local structural feature for the target representation. Moreover, some algorithms try to represent the target by exploiting the combination of holistic and local templates for tracking [36,37].

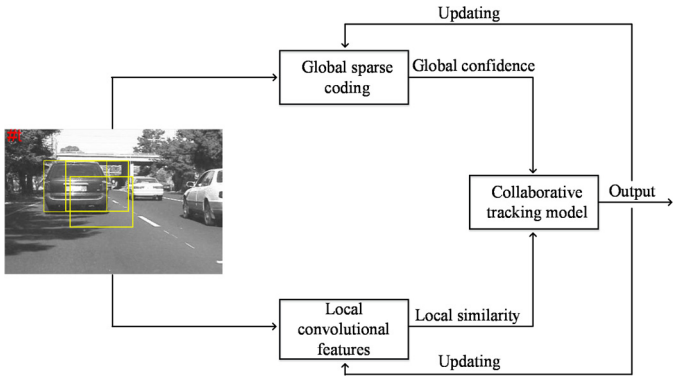

Fig. 1. Flowchart of the proposed tracking algorithm.

3. The proposed tracker

In this paper, we represent the target at both global and local levels. As illustrated in Fig. 1, the proposed tracking algorithm consists of two parts: global and local parts which generate global and local representations of the target respectively. Afterwards, the two parts are integrated into a collaborative model for tracking. More details are stated in the rest of this section.

3.1. Global sparse coding for visual tracking

Referring to the target in the first frame, a bunch of positive and negative samples are extracted to establish positive dictionary and negative dictionary. These two dictionaries construct a global dictionary. To implement tracking task, each candidate target sampled from subsequent frames is firstly sparsely decomposed on the global dictionary. Then, a term named as global confidence is computed based on the reconstruction errors using the two different sub-dictionaries.

As shown in Fig. 2, we crop n_p positive samples from a circular region specified by

$$\Omega^{+} = \left\{ (x_{pos}, y_{pos}) \,\middle|\, \sqrt{(x_{pos} - x_0)^2 + (y_{pos} - y_0)^2} \le \partial \right\}$$

in the first frame, where (x_0, y_0) and (x_{pos}, y_{pos}) denote the center locations of the target object and positive samples, respectively. Positive samples are warped to a canonical size (32×32 in our experiments) and used for the construction of the positive dictionary D_p. Then, n_n negative samples are randomly drawn from the region denoted by

$$\Omega^{-} = \left\{ (x_{neg}, y_{neg}) \,\middle|\, |x_{neg} - x_0| \ge \frac{w_0}{4}, |y_{neg} - y_0| \ge \frac{h_0}{4} \right\},$$

where w_0 and h_0 stand for the width and height of the target object, (x_{neg}, y_{neg}) denotes the centers of negative samples. (w_0, h_0) changes a lot with regard to different sequences and different tracking objects (e.g., 107×87 for car4 sequence of Fig. 8 and 32×73 for doll sequence of Fig. 11). Similarly, negative samples are warped and stacked together to form the negative dictionary D_n. In this way, we can obtain a global dictionary including positive and negative samples.

Since the dimension of each column in global dictionary and candidates is very high, and the corresponding gray-scale features of them are highly redundant, we implement feature selection (see Fig. 3) before sparse decomposition of candidates to extract discriminative features as

$$\min_s \left\| D^T s - p \right\|_2^2 + \lambda \|s\|_1 \tag{1}$$

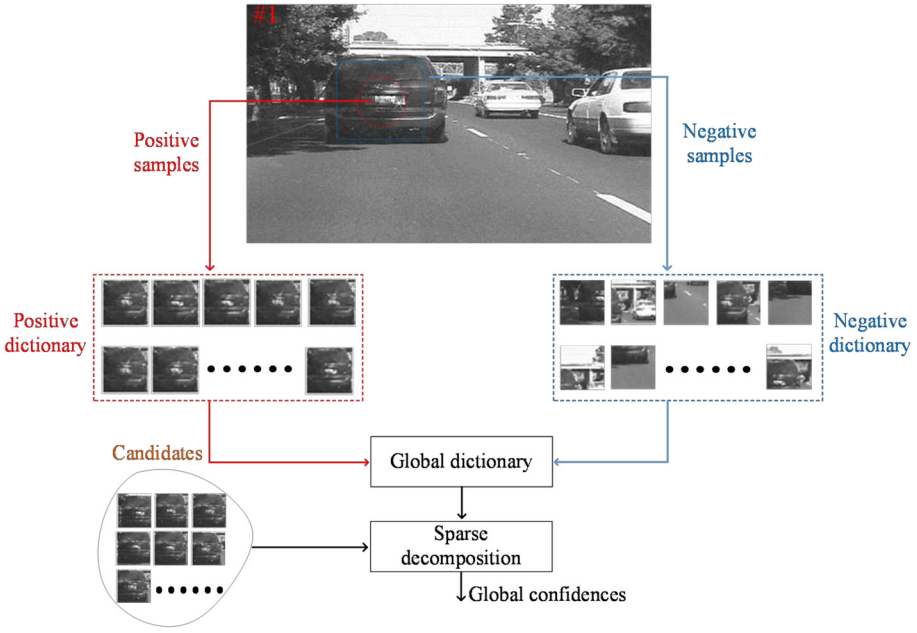

Fig. 2. Workflow of global sparse coding for visual tracking.

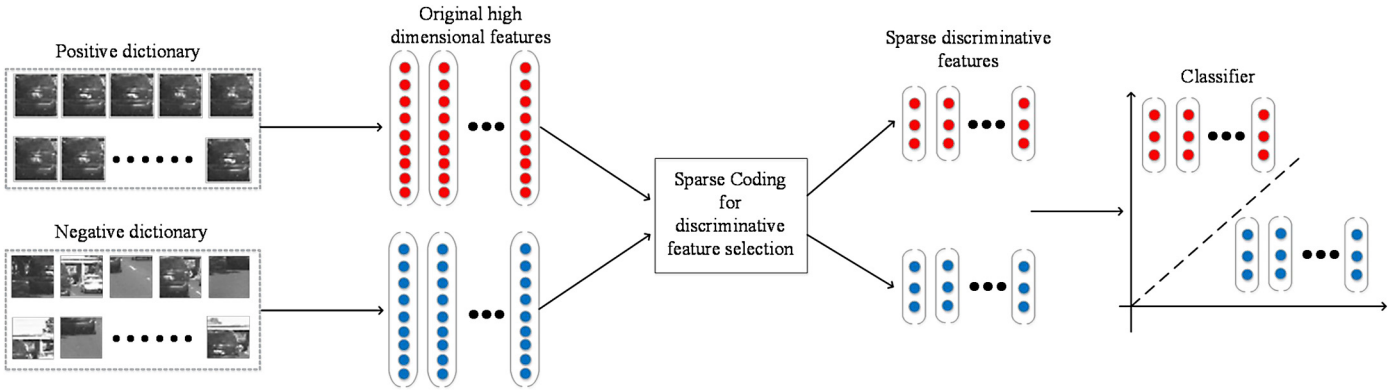

Fig. 3. The process of sparse feature selection.

where $D = [D_p, D_n] \in R^{K\times(n_p+n_n)}$ is the global dictionary composed of n_p positive samples and n_n negative samples, K is the dimension of gray-scale features before feature selection. Each element of the vector $p \in R^{(n_p+n_n)\times 1}$ represents the class label of each sample, i.e. $+1$ for positive samples and -1 for negative samples. The solution s of Eq. (1) is a sparse vector, which enables itself to be used as a classifier. The index of the nonzero elements of s corresponds to the selected feature dimensionalities of global dictionary and candidates.

Given a candidate region, it can be represented by the global dictionary with the coefficients α solved by

$$\min_{\alpha} \left\|x' - D'\alpha\right\|_2^2 + \lambda\|\alpha\|_1 \tag{2}$$

where $D' = [D'_p, D'_n] \in R^{K'\times(n_p+n_n)}$ and $x' \in R^{K'\times 1}$ denote the global dictionary and the candidate after feature selection respectively, λ is a control parameter. Based on the assumption that a target image region should produce a smaller reconstruction error by using the positive dictionary, but vice versa by using the negative dictionary, we define the global confidence as follows.

$$H(x) = \frac{\varepsilon_n}{\varepsilon_p + \mu} \tag{3}$$

where $\varepsilon_p = \left\|x' - D'_p\alpha_p\right\|_2^2$ is the reconstruction error of the candidate x by using the positive dictionary, and α_p is the corresponding sparse coefficient vector. Similarly, $\varepsilon_n = \left\|x' - D'_n\alpha_n\right\|_2^2$ is the reconstruction error of the candidate x by using the negative dictionary, and α_n is the corresponding sparse coefficient vector. The variable μ is a constraint factor to avoid dividing by zero. The global confidence exploits the distinct properties of the foreground and the background in computing the reconstruction errors, which can effectively distinguish the target from a complicated background.

3.2. Local convolutional features for tracking

Like most of trackers in the literatures, the proposed tracker is performed on grayscale images. The color images are firstly transformed into grayscale ones before performing tracking task. The target and each candidate are warped to $n \times n$ ($n = 32$ in our method) pixels. We use a sliding window of size $w \times w$ to densely sample a set of overlapping local image patches $Y = \{Y_1, Y_2, ..., Y_l\}$ inside the warped target region, where $Y_i \in R^{w\times w}$ is the i-th image patch and $l = (n - w + 1) \times (n - w + 1)$. Then, the k-means algorithm is applied to these local patches and d different cluster centers can be obtained. Each cluster center is preprocessed by subtracting the mean and l_2 normalization to construct the ob-

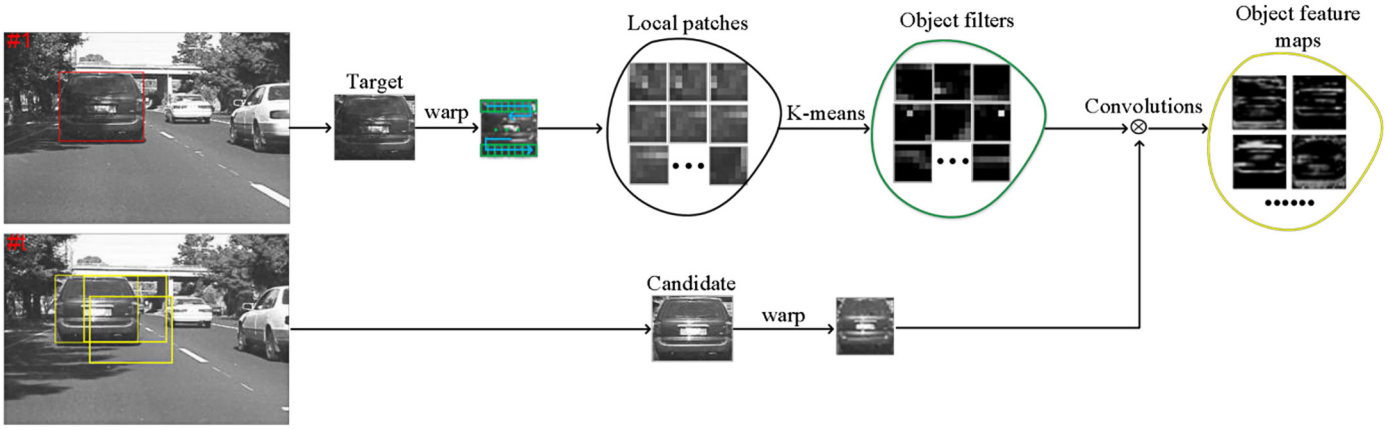

Fig. 4. The processes of construction of object filters and convolution operation for getting object feature maps.

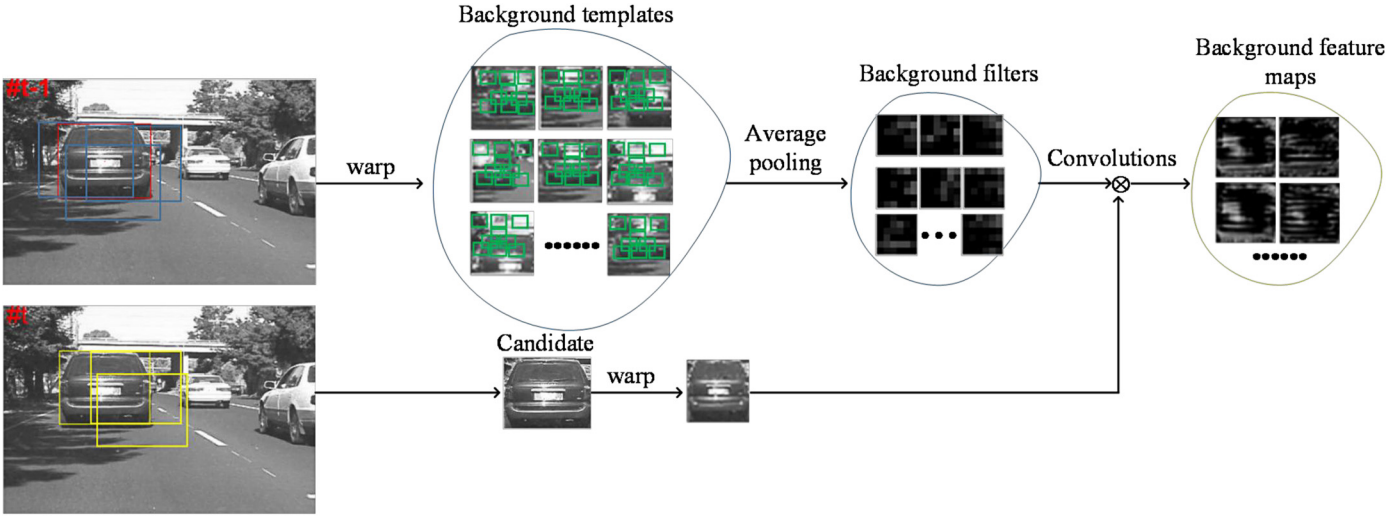

Fig. 5. The processes of construction of background filters and convolution operation for getting background feature maps.

ject filters $F^o = \{F_1^o, F_2^o, ..., F_d^o\}$. Given an input candidate x, each object filter F_i^o convolves it and generates an object feature map $S_i^o \in R^{(n-w+1)\times(n-w+1)}$, where $S_i^o = F_i^o \otimes x$ and $\otimes$ denotes the operation of convolution. The whole process is shown in Fig. 4.

The background context around the target can help to discriminate the target from the background. As shown in Fig. 5, we choose m background templates surrounding the target. Each background template is partitioned into small local image patches. For each background template, d different local image patches are selected. Then, the average pooling method is applied to obtain the background filters as

$$F^b = \left\{ F_1^b = \frac{1}{m}\sum_{i=1}^{m} F_{i,1}^b, ..., F_d^b = \frac{1}{m}\sum_{i=1}^{m} F_{i,d}^b \right\} \tag{4}$$

where $F_{i,1}^b, F_{i,2}^b, ..., F_{i,d}^b$ are the selected d different patches of the i-th background template. Similarly, each background filter is preprocessed in the same way as object filters. After preprocessing, each background filter F_i^b convolves the candidate x and produces a background feature map $S_i^b \in R^{(n-w+1)\times(n-w+1)}$, where $S_i^b = F_i^b \otimes x$. Taking into account both target and background context, the final convolutional feature map is obtained by $S_i = S_i^o - S_i^b$. Each convolutional feature map S_i is converted to a vector $f_i \in R^{(n-w+1)^2\times 1}$ and concatenated to form a representation.

$$\rho = [f_1, f_2, ..., f_d] \tag{5}$$

where $\rho \in R^{(n-w+1)^2\times d}$ is the representation of the candidate target x.

The similarity between the candidate x and the target is computed by

$$L(x) = e^{-\|\psi-\rho\|_2^1} \tag{6}$$

where ψ is target representation. This similarity measure is based on local convolutional features, so we call it local similarity. Target representation ψ is generated by the background filters and the object filters convolving the target region in the first frame. It is updated every frame and the update scheme is presented in section 3.4.

3.3. Collaborative tracking model

Visual tracking has been commonly carried out within the Bayesian filtering framework. Given the observation set $O_t = \{o_1, o_2, ..., o_t\}$ up to the frame t, our goal is to determine a posterior probability $p(s_t|O_t)$ by

$$p(s_t|O_t) \propto p(o_t|s_t) \int p(s_t|s_{t-1})\, p(s_{t-1}|O_{t-1})\, ds_{t-1} \tag{7}$$

where s_t is the target state at frame t, $p(s_t|s_{t-1})$ is the motion model and $p(o_t|s_t)$ is the observation model. The motion model $p(s_t|s_{t-1})$ describes the state transition between consecutive frames and is often set as Gaussian distribution $p(s_t|s_{t-1}) =$

$N(s_t; s_{t-1}, \sum)$, where $\sum = diag(\sigma_x, \sigma_y, \sigma_s)$ is a diagonal covariance matrix whose elements are the variances of the affine parameters. The observation model $p(o_t|s_t)$ estimates the likelihood of observation o_t at the state s_t belonging to the target class. The particle filter is an effective realization of Bayesian filtering, in which the state is predicted regardless of the underlying distribution. The optimal state is obtained by the maximum a posterior estimation (MAP) over N samples,

$$\hat{s_t} = \arg\max_{s_t^i} p\left(o_t \left| s_t^i \right.\right) p\left(s_t^i \left| \hat{s_{t-1}} \right.\right) \tag{8}$$

where s_t^i is the i-th sample at frame t. The observation model $p\left(o_t \left| s_t^i \right.\right)$ in (8) plays a key role in visual tracking. In our algorithm, the observation model is constructed by

$$p\left(o_t \left| s_t^i \right.\right) \propto \nu H\left(s_t^i\right) + (1-\nu) L\left(s_t^i\right) \tag{9}$$

where $H\left(s_t^i\right)$ is the global confidence of sample s_t^i according to (3), $L\left(s_t^i\right)$ is the local similarity of sample s_t^i according to (6), ν is the weight for balancing the contributions of these two parts. From (9), it can be observed that we combine the global and local parts by the way of weighted balance rather than by simply multiplying the confidences of these two parts.

3.4. Model updating

Since the appearance of an object often changes significantly during the tracking process, the update scheme is important and necessary. In this paper, we develop an update scheme, which updates the global part and the local part independently.

For the global part, we update the negative dictionary every several frames (5 in our experiments). The n_n new negative samples $\left\{(x_{neg}, y_{neg}) \left| |x_{neg} - x^*| \geq \frac{w^*}{4}, |y_{neg} - y^*| \geq \frac{h^*}{4} \right.\right\}$ are collected far away from the current tracked target to update the negative dictionary, where (x^*, y^*) denotes the center position of the current tracked target, w^* and h^* denote the width and height of the current tracked target. The positive dictionary remains the same in the tracking process. As the global part is a discriminative model and aims to distinguish the foreground from the background, it is important to ensure that the positive dictionary is correct.

For the local part, the object filters are fixed during the tracking process, alleviating the drift problem effectively. Similar strategy has been adopted in [37,41,42], in which one single scale or multi-scale static dictionaries learned from the first frame are exploited to sparsely represent the tracked target. We extract background templates from the image regions around the tracked target to update the background filters. Motivated by the work [40], we utilize the following object function to de-noise the representation of the tracked target, and make it more robust to appearance variation.

$$\hat{\varphi} = \arg\min_{\varphi} \lambda_1 \|\varphi\|_1^1 + \frac{1}{2}\|\varphi - c\|_2^2 \tag{10}$$

where $c \in R^{(n-w+1)^2 d}$ is a column vector by concatenating all the elements of representation of the tracked target, λ_1 is set to $\lambda_1 = median(abs(c))$. The solution of the above function can be achieved by a soft thresholding function

$$\hat{\varphi} = sign(c) \max(0, abs(c) - \lambda_1) \tag{11}$$

where $sign(\bullet)$ is a sign function. $\hat{\varphi}$ is adjusted to have the same size as target representation ψ. In order to capture the appearance changes, the target representation is incrementally updated by

$$\psi_t = \eta \psi_{t-1} + (1-\eta)\hat{\varphi_t} \tag{12}$$

where η is the update rate, ψ_{t-1} is the target representation at frame $t-1$ and $\hat{\varphi_t}$ is the denoised representation of the tracked target at frame t. This incremental update scheme not only adapts to the target appearance variations, but also alleviates the drift problem. The main steps of the proposed algorithm are summarized in Algorithm 1.

Algorithm 1 The proposed tracking algorithm.

Input: Global dictionary D, object filters F^o, background filters F_{t-1}^b, target state $\hat{s_{t-1}}$, target representation ψ_{t-1}.

1: Sample N candidate particles $\left\{s_t^i\right\}_{i=1}^N$ with the motion model $p\left(s_t^i \left| \hat{s_{t-1}} \right.\right)$.
2: For each particle s_t^i, compute its global confidence and local similarity using Eq. (3) and Eq. (6).
3: Calculate the observation likelihood $p(o_t|s_t^i)$ of each particle s_t^i by Eq. (9).
4: Find the tracked target $\hat{s_t}$ through the maximal observation likelihood.
5: Extract negative samples to update the global dictionary D every five frames.
6: Draw background templates to update background filters F_t^b every frame.
7: Compute the denoised representation $\hat{\varphi_t}$ of the tracked target by Eq. (11) and update target representation ψ_t by Eq. (12).

Output: Tracked target $\hat{s_t}$ and target representation ψ_t.

4. Experimental results

4.1. Experimental setup

The numbers of positive samples n_p and negative samples n_n are 50 and 200, respectively. The sampling radii of positive samples is set as $\partial = 1$. The regularization parameter λ in Eq. (1) is set to be 0.001. The variable λ in Eq. (2) is fixed to be 0.01. The size of the filter is set as 6×6 ($w = 6$) and the number of filters is set as 100 ($d = 100$). The variance matrix of affine parameters is set as $\Sigma = diag(4, 4, 0.01)$. In our approach, $N = 600$ particles are used. The number of background templates m is 20. The update rate η in Eq. (12) and the weight ν in Eq. (9) are set as 0.95 and 0.5, respectively.

4.2. Evaluation metrics

To evaluate the performance of our proposed tracker, we conduct experiments on 50 fully-annotated videos of the tracking benchmark dataset [26]. These videos cover different challenging situations in object tracking: low resolution (LR), in-plane rotation (IPR), out-of-plane rotation (OPR), scale variation (SV), occlusion (OCC), deformation (DEF), background clutters (BC), illumination variation (IV), motion blur (MB), fast motion (FM), and out-of-view (OV). With the same initial positions of the targets in the first frame, we compare with several state-of-the-art tracking algorithms including KCF [7], CNT [25], ASLA [10], SCM [37], Struck [4], TLD [9], VTD [3], TGPR [11], and DLT [22].

For quantitative evaluation, we use the precision plot and the success plot to evaluate all the trackers. The success plot is based on the overlap ratio and illustrates the percentage of frames where the overlap ratio between the tracked bounding box and the ground truth bounding box is higher than a threshold $t_0 \in [0, 1]$. The area under curve (AUC) of each success plot is used to rank the tracking algorithms. Meanwhile, the precision plot demonstrates the percentage of frames where the distance between the tracked target location and the ground truth location is within a given threshold. We report the results of one pass evaluation (OPE) based on the average success and precision rate for all the trackers.

4.3. Quantitative comparisons

Fig. 6 shows the precision plots and success plots which illustrate the performance of the 10 tracking algorithms on 50 videos.

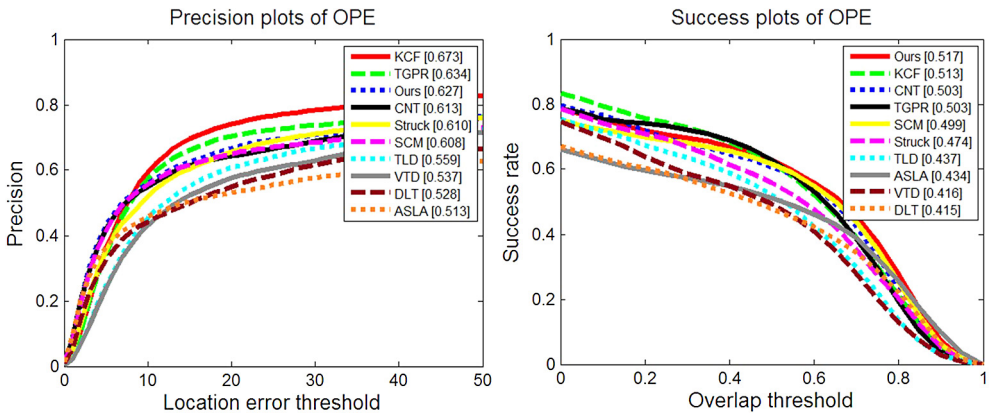

Fig. 6. The success plots and precision plots of OPE for the 10 trackers. The performance score of precision plot is at error threshold of 20 pixels. The legends show the precision scores and AUC values for each tracker. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

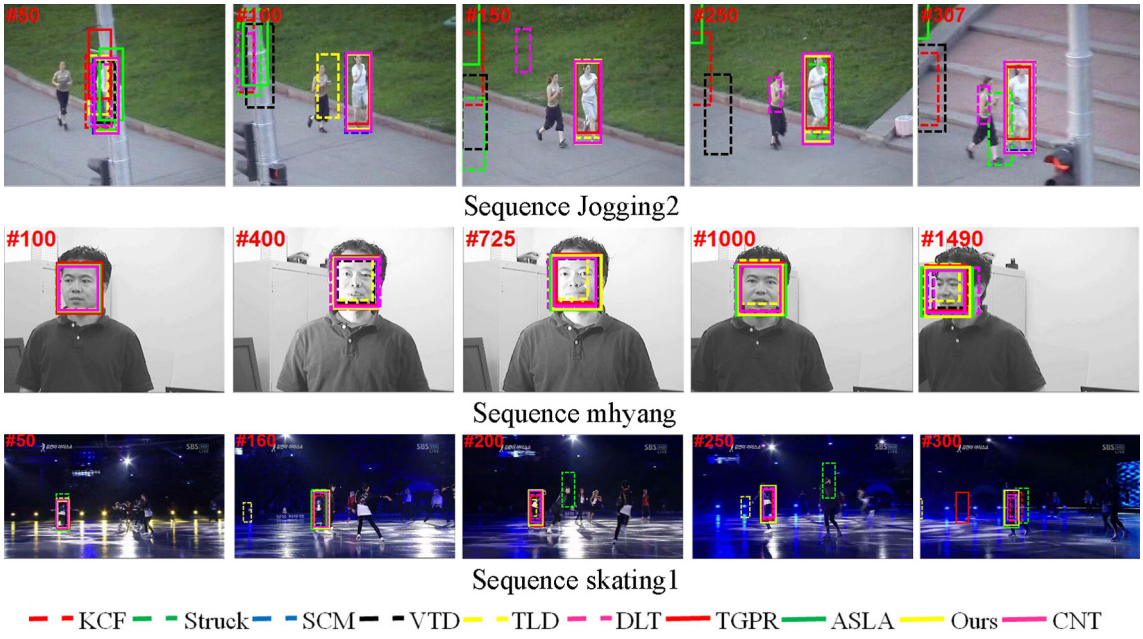

Fig. 7. Qualitative results of the 10 trackers over sequences Jogging2, mhyang and skating1, in which the targets undergo deformation. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Note that all the plots are generated by using the code library from the benchmark evaluation [26]. As for CNT [25], KCF [7], TGPR [11] and DLT [22], we obtain the results by using the source codes provided by the authors. The proposed tracker ranks first based on the success rate while third based on the precision rate. In the precision plot, the precision score of our method is 0.627, which is slightly lower than the TGPR method (0.634) but performs better than the CNT method (0.613), the Struck method (0.610) and the SCM method (0.608). In the success plot, the AUC of the proposed algorithm achieves 0.517, which outperforms the CNT method (0.503), the TGPR method (0.503), the Struck method (0.474) and the SCM method (0.499). The proposed method performs better than the CNT method due to the consideration of the local information and holistic information of the target. Moreover, we note that the KCF and TGPR methods achieve higher precision scores than our method, but lower AUC scores. The main reason for this is that the two methods predict the location of the target precisely but don't handle scale variations of the target well.

4.4. Qualitative comparisons

We select sixteen challenging sequences including deformation, illumination variation, occlusion, background clutter, scale variation and pose change to evaluate our proposed tracker. The experimental results on these videos are illustrated in Figs. 7–12.

Deformation: Fig. 7 shows the tracking results in three sequences where the targets undergo shape deformation. In the Jogging2 sequence, the target undergoes both occlusion and deformation. The ASLA, Struck, KCF, VTD and DLT methods can't recapture the target and undergo large drift when the person goes across the lamp post and reappears in the screen (e.g. #100, #150 and #250). The TLD method locks on to another person at frame #100, but it can obtain the correct target again using a reinitialization mechanism in the subsequent frames. On the contrary, the TGPR, CNT, SCM and our method perform well throughout the sequence. For the mhyang sequence, the ASLA, DLT and our method perform better than other methods and achieve higher overlap rate. The target in the skating1 sequence undergoes significant appearance variation due to non-rigid body deformation and drastic illumina-

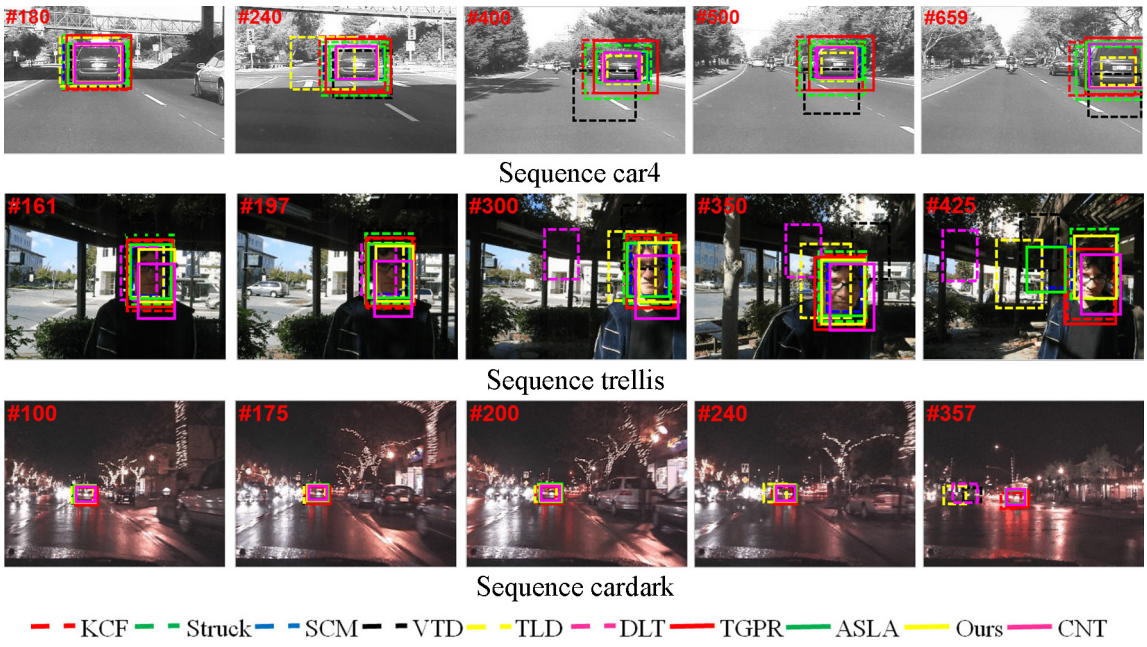

Fig. 8. Qualitative results of the 10 trackers over sequences car4, trellis and cardark, in which the targets undergo illumination changes. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

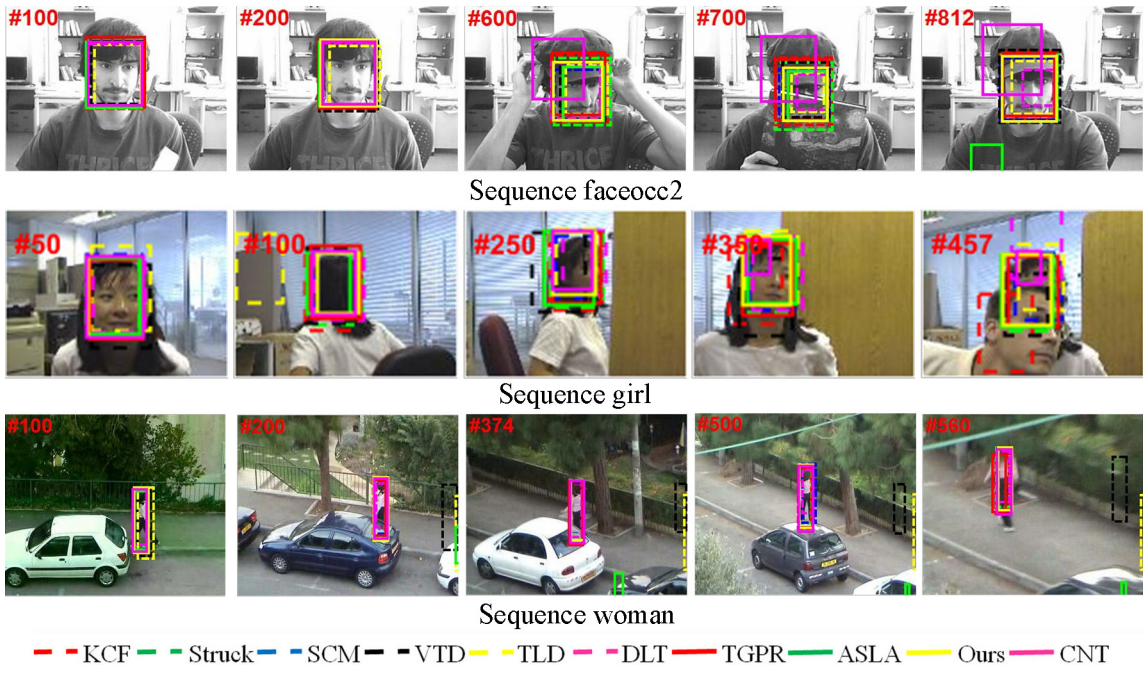

Fig. 9. Qualitative results of the 10 trackers over sequences faceocc2, girl and woman, in which the targets undergo heavy occlusion or partial occlusion. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

tion variation. The ASLA, SCM, VTD and our method perform better than other methods.

Illumination variations: Fig. 8 demonstrates the tracking results in three sequences with illumination variations. In the car4 sequence, the target suffers from illumination and scale variations. The VTD method drifts off the target after the target passes through the bridge (seen from #240, #400 and #500). The TLD method drifts away from the target when the target undergoes drastic illumination change at frame #240. Although the TGPR, Struck and KCF methods can successfully track the target, they don't handle scale variations well (e.g. #400 and #500). In contrast, the CNT, DLT, SCM, ASLA and our method are able to achieve accurate results in terms of both location and scale due to the affine motion model. In the trellis sequence, the person walks from a dark place to a bright environment, where the target meets with significant illumination variations. The TLD and VTD methods drift away to background (e.g. #300, #350, and #425). The SCM and ASLA methods lose the target at frame #425. The DLT method fails to track the target from frame #300. The Struck and our method are able to track the target with better accuracy than the CNT, TGPR and KCF methods. In the cardark sequence, the target moves in a night scene with low contrast and illumination change. The TLD method drifts off the target from frame #175 and loses the target at frame #357. The VTD and DLT methods

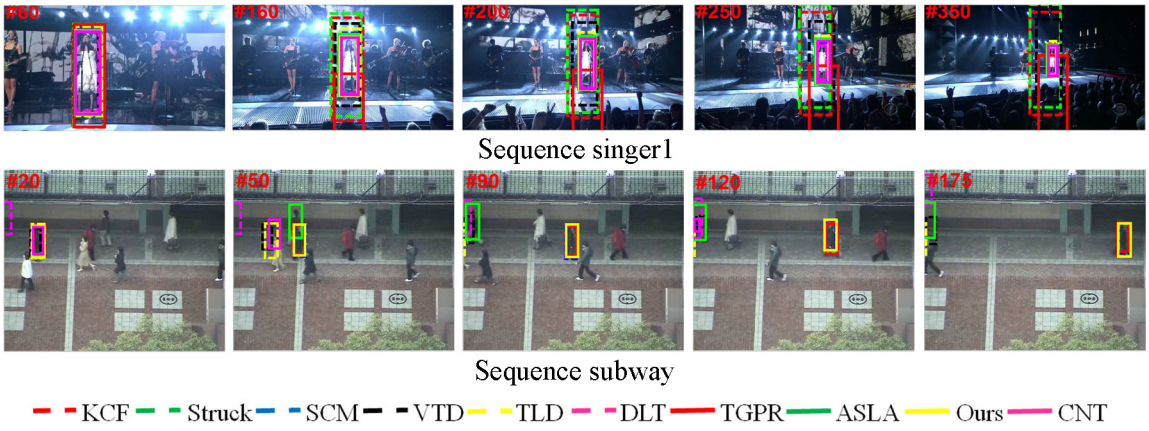

Fig. 10. Qualitative results of the 10 trackers over sequences singer1 and subway, in which the targets undergo background clutters. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

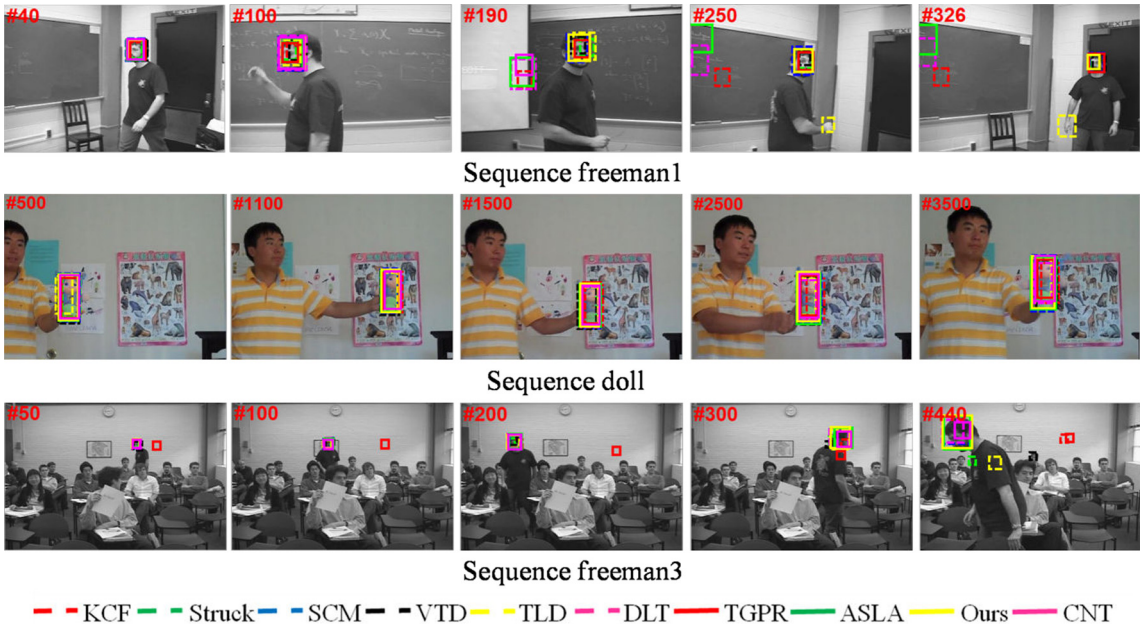

Fig. 11. Qualitative results of the 10 trackers over sequences freeman1, doll and freeman3, in which the targets undergo scale variations. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

lose track of the target at frame #357. The Struck, CNT, ASLA, SCM, TGPR and our method perform well throughout the entire sequence.

Occlusion: As shown in Fig. 9, the tracked objects in faceocc2, girl and woman sequences encounter heavy occlusion or partial occlusion. In the woman sequence, the target experiences pose variation together with frequent long-time partial occlusion. Except for the TLD, VTD and ASLA methods, the other seven methods perform well in this sequence. In the faceocc2 sequence, the target undergoes heavy occlusion. Most trackers are able to track the target from the start to the end. However, the CNT method drifts away from the target from frame #600 to frame #812. The ASLA method doesn't perform well at the end (seen from #812). The target in the girl sequence undergoes rotation and partial occlusion. The ASLA, SCM, Struck and our method perform better than other methods. The robustness of our method against occlusion attributes to local appearance update scheme in which the soft thresholding strategy effectively reduces the influence of occlusion.

Background clutter: Fig. 10 shows some screenshots of the tracking results in the singer1 and subway sequences in which the targets appear in background clutters. The singer1 sequence is challenging as the background is cluttered and the target experiences illumination variations. The TGPR method severely deviates from the target when significant illumination variations occur (e.g. #160, #200, and #250). Furthermore, the target undergoes large scale variations. Although the VTD, KCF and Struck methods can successfully track the target, they fail to track scale variations. The CNT, ASLA, SCM, DLT and our method perform better compared with other methods and achieve favorable results. The subway sequence contains numerous challenges, such as background clutter, occlusion and pose change. The DLT method can't track the target correctly from the start of the sequence (e.g. #20). The ASLA, CNT, VTD and TLD methods fail to track the target and lock onto a wrong target (seen from #50, #90 and #175). The Struck, TGPR, SCM, KCF and our method precisely keep track of the target to the end. Our method performs well in this sequence as discriminative features are selected to separate the target from the cluttered background in the global part.

Scale variations: Fig. 11 illustrates some results over three challenging sequences with scale variations. In the freeman1 sequence, a person undergoes a large scale variation in his face. The ASLA, CNT, DLT and KCF methods cannot track the target from frame

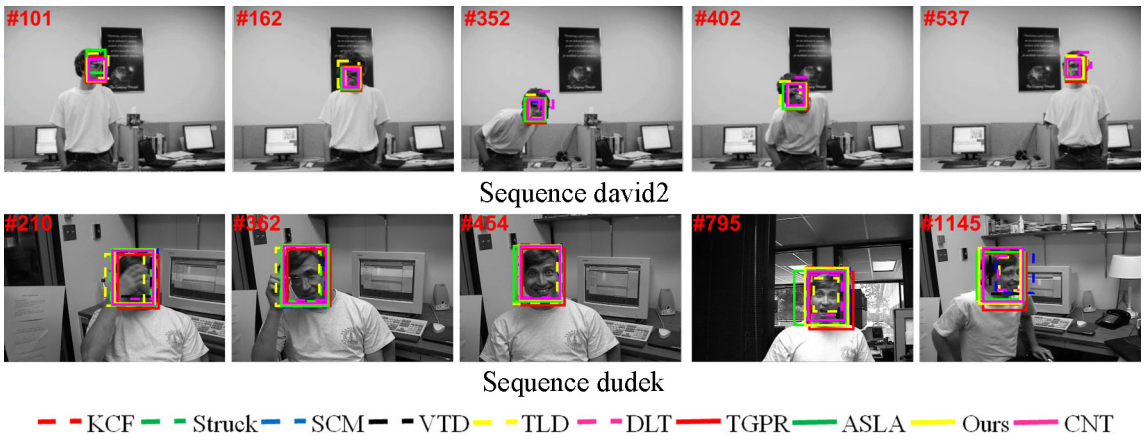

Fig. 12. Qualitative results of the 10 trackers over sequences david2 and dudek, in which the targets undergo pose change. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

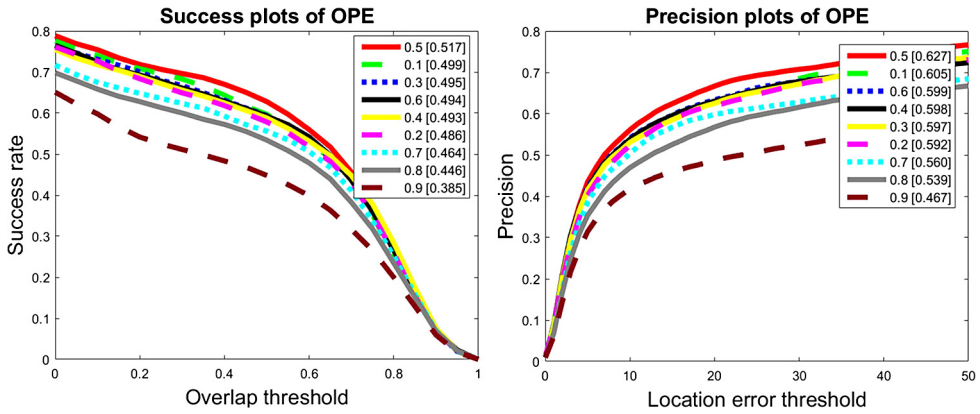

Fig. 13. Comparison of the tracking performance of our tracker with different weight values. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

#190 to the end. The TLD method performs unstably and loses the target in the tracking process (e.g. #250 and #326). The SCM, TGPR, VTD, Struck and our method can track the target well. The target in the doll sequence undergoes a long time scale variation and rotation. The ASLA, SCM and our method perform better than other methods and achieve higher accuracy. In the freeman3 sequence, the TGPR method fails to track the target (e.g. #50, #200, and #440). The VTD, Struck and KCF methods show large deviation away from the target (seen from #300 and #440). The ASLA, SCM, CNT, DLT and our method can successfully track the target till to the end but the ASLA and our method perform best in handling scale changes.

Pose change: Fig. 12 shows the comparison of all the trackers when dealing with the challenge of pose change on two sequences, where the frames in both two sequences are randomly selected from the whole sequences. Obviously, the target object is the human face in both two sequences. In the david2 sequence, the man swings his head randomly and changes pose all the time. After checking all the frames of david2 sequence, we find that all the methods except DLT can cope with this challenge well and successfully track the target in the whole sequence. The drift occurs to DLT in frames #352, #402 and #537. The reason for DLT failing is that the features learned offline from training set may not well adapt to target appearance variations during tracking. In the dudek sequence, the target goes through multiple challenges of pose change, occlusion and background clutter. All the methods can win these challenges to some extent and achieve good performance. In Fig. 12, the bounding boxes in yellow color gives the results of our proposed method. It can be observed that they can precisely enclose the human face in all the frames, indicating our proposed method do well for pose change challenge.

4.5. Discussion and analysis

4.5.1. Proposed algorithm with different combined weights

The weight ν is an important parameter in our method, which controls the trade-off between the contributions from global part and local part. In this subsection, we investigate the tracking performance of the proposed tracker with different weight ν values. Fig. 13 illustrates the quantitative results on the tracking benchmark dataset including the success plots and precision plots. Experimental results show that the proposed tracker performs best when the value of ν is 0.5.

4.5.2. Performance under different associative mechanisms

In this subsection, we compare the tracking performance of our method based on two different combination way of global confidence and local similarity. One is by means of multiplication. The other is by means of weighted balance and the weight (ν) is 0.5. From Fig. 14, it can be observed that the latter gives better results. Thus, we integrate global confidence and local similarity in the observation model by the way of weighted balance.

4.5.3. Effects of key components

To validate the effectiveness of key components of the proposed method, we propose four variants of our method: (1) one utilizes positive dictionary update in which we update positive dictionary every five frames; (2) one does not involve the soft thresholding

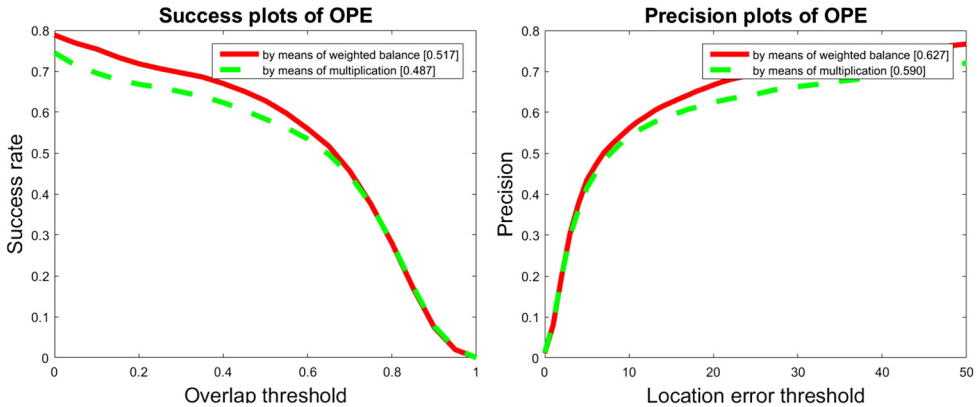

Fig. 14. Comparison of the tracking performance of our tracker with two different integration way of the two confidences in the collaborative model.

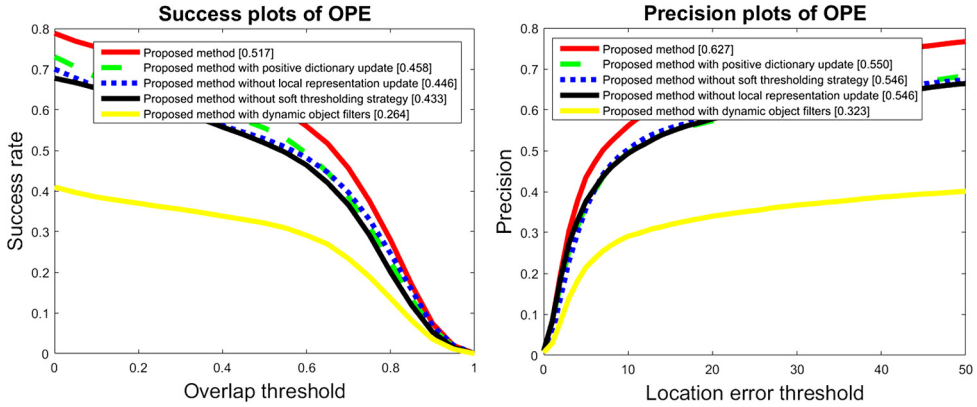

Fig. 15. Success plots and precision plots of OPE for four variants of our method. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

strategy; (3) one does not use local representation update scheme of (12); (4) one employs dynamic object filters in which the object filters are updated in the same way as background filters. Fig. 15 shows the quantitative results on the tracking benchmark dataset. From the results, we can see that with positive dictionary update, the AUC score of success rate reduces by 5.9%. Meanwhile, proposed method without soft thresholding strategy and proposed method without local representation update can only achieve AUC score of 0.433 and 0.446, which are both lower than the proposed algorithm with 0.517. Furthermore, the results for proposed method with dynamic object filters are much worse than the proposed algorithm. These results show that static positive dictionary, soft thresholding strategy, fixed object filters, and local representation update components play crucial roles in the proposed algorithm for robust visual tracking. They effectively alleviate and help overcome some degree of drifting.

5. Conclusion

In this paper, we propose and demonstrate an effective and robust tracking algorithm which aims to mine the information of the target from both global and local levels. In the global level, holistic features are utilized to separate the target object from the background via positive and negative dictionaries to encode sparse coefficients. In the local level, we employ a bank of filters to extract local convolutional features of the target and obtain the local representation of the target. Moreover, a soft thresholding strategy is incorporated to de-noise the target representation. The contributions of global level and local level are integrated into a unified model. Extensive evaluation on the benchmark dataset demonstrates the proposed tracking algorithm is competitive among all compared algorithms.

Acknowledgment

This work was partially supported by the National Natural Science Foundation of China under grants 61572461, 61472257, 61572067 and 11433006, CAS 100-Talents (Dr. Xu Long).

References

[1] A. Adam, E. Rivlin, I. Shimshoni, Robust fragments-based tracking using the integral histogram, IEEE Conf. Comput. Vis. Pattern Recognit. (2006) 798–805.
[2] B. Babenko, M.-H. Yang, S. Belongie, Robust object tracking with online multiple instance learning, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1619–1632.
[3] J. Kwon, K.M. Lee, Visual tracking decomposition, IEEE Conf. Comput. Vis. Pattern Recognit. (2010) 1269–1276.
[4] S. Hare, A. Saffari, P.H.S. Torr, Struck: structured output tracking with kernels, IEEE Int. Conf. Comput. Vis. (2011) 263–270.
[5] K. Zhang, L. Zhang, M.-H. Yang, Real-time compressive tracking, in: Proc. 12th European Conf. on Computer Vision, 2012, pp. 864–877.
[6] M. Danelljan, et al., Adaptive color attributes for real-time visual tracking, IEEE Conf. Comput. Vis. Pattern Recognit. (2014) 1090–1097.
[7] J.F. Henriques, et al., High-speed tracking with kernelized correlation filters, IEEE Trans. Pattern Anal. Mach. Intell. 37 (3) (2015) 583–596.
[8] D. Du, et al., Discriminative hash tracking with group sparsity, IEEE Trans. Cybern. 46 (8) (2016) 1914–1925.
[9] Z. Kalal, K. Mikolajczyk, J. Matas, Tracking-learning-detection, IEEE Trans. Pattern Anal. Mach. Intell. 34 (7) (2012) 1409–1422.
[10] X. Jia, H. Lu, M.-H. Yang, Visual tracking via adaptive structural local sparse appearance model, IEEE Conf. Comput. Vis. Pattern Recognit. (2012) 1822–1829.
[11] J. Gao, et al., Transfer learning based visual tracking with Gaussian processes region, in: Proc. European Conf. on Computer Vision, 2014, pp. 188–203.

[12] Bin Gu, Victor S. Sheng, A robust regularization path algorithm for ν-support vector classification, IEEE Trans. Neural Netw. Learn. Syst. (2016), https://doi.org/10.1109/TNNLS.2016.2527796.
[13] Bin Gu, Xingming Sun, Victor S. Sheng, Structural minimax probability machine, IEEE Trans. Neural Netw. Learn. Syst. 28 (7) (2017) 1646–1656.
[14] Bin Gu, Victor S. Sheng, Keng Yeow Tay, Walter Romano, Shuo Li, Incremental support vector learning for ordinal regression, IEEE Trans. Neural Netw. Learn. Syst. 26 (7) (2015) 1403–1416.
[15] J. Deng, et al., Imagenet: a large-scale hierarchical image database, IEEE Conf. Comput. Vis. Pattern Recognit. (2009) 248–255.
[16] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv:1409.1556, 2014.
[17] W. Ouyang, et al., Deepid-net: deformable deep convolutional neural networks for object detection, IEEE Conf. Comput. Vis. Pattern Recognit. (2015) 2403–2412.
[18] R. Girshick, et al., Rich feature hierarchies for accurate object detection and semantic segmentation, IEEE Conf. Comput. Vis. Pattern Recognit. (2014) 580–587.
[19] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proc. Adv. Neural Information Processing Systems, 2012, pp. 1097–1105.
[20] H. Li, Y. Li, F. Porikli, Robust online visual tracking with a single convolutional neural network, in: Proc. 12th Asian Conf. Computer Vision, 2014, pp. 194–209.
[21] X. Zhou, et al., An ensemble of deep neural networks for object tracking, in: Proc. IEEE Int. Conf. Image Process, 2014, pp. 843–847.
[22] N. Wang, D.-Y. Yeung, Learning a deep compact image representation for visual tracking, Proc. Adv. Neural Inf. Process. Syst. (2013) 809–817.
[23] J. Fan, et al., Human tracking using convolutional neural networks, IEEE Trans. Neural Netw. 21 (10) (2010) 1610–1623.
[24] S. Hong, et al., Online tracking by learning discriminative saliency map with convolutional neural network, in: Proc. Int. Conf. Mach. Learn, 2015, pp. 597–606.
[25] K. Zhang, et al., Robust visual tracking via convolutional networks without training, IEEE Trans. Image Process. 25 (4) (2016) 1779–1792.
[26] Y. Wu, J. Lim, M.-H. Yang, Online object tracking: a benchmark, IEEE Conf. Comput. Vis. Pattern Recognit. (2013) 2411–2418.
[27] A. Yilmaz, O. Javed, M. Shah, Object tracking: a survey, ACM Comput. Surv. 38 (4) (2006) 1–45.
[28] K. Cannons, A Review of Visual Tracking, Tech. Rep. CSE-2008-07, Dept. Comput. Sci., York Univ., Toronto, ON, Canada, 2008.
[29] D.A. Ross, et al., Incremental learning for robust visual tracking, Int. J. Comput. Vis. 77 (1–3) (2008) 125–141.
[30] D. Wang, H. Lu, M.-H. Yang, Least soft-threshold squares tracking, IEEE Conf. Comput. Vis. Pattern Recognit. (2013) 2371–2378.
[31] X. Mei, H. Ling, Robust visual tracking and vehicle classification via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (11) (2011) 2259–2272.
[32] C. Bao, et al., Real time robust L1 tracker using accelerated proximal gradient approach, IEEE Conf. Comput. Vis. Pattern Recognit. (2012) 1830–1837.
[33] T. Zhang, et al., Robust visual tracking via multi-task sparse learning, IEEE Conf. Comput. Vis. Pattern Recognit. (2012) 2042–2049.
[34] B. Liu, et al., Robust tracking using local sparse appearance model and K-selection, IEEE Conf. Comput. Vis. Pattern Recognit. (2011) 1313–1320.
[35] Q. Wang, et al., Online discriminative object tracking with local sparse representation, IEEE WACV (2012) 345–352.
[36] Y. Yang, et al., Global coupled learning and local consistencies ensuring for sparse-based tracking, Neurocomputing 160 (c) (2015) 191–205.
[37] W. Zhong, H. Lu, M.-H. Yang, Robust object tracking via sparsity-based collaborative model, IEEE Conf. Comput. Vis. Pattern Recognit. (2012) 1838–1845.
[38] S. Duffner, C. Garcia, Pixeltrack: a fast adaptive algorithm for tracking non-rigid objects, IEEE Int. Conf. Comput. Vis. (2013) 2480–2487.
[39] Z. Hong, et al., Tracking using multilevel quantizations, in: Proc. 13th European Conf. on Computer Vision, 2014, pp. 155–171.
[40] M. Elad, M.A.T. Figueiredo, Y. Ma, On the role of sparse and redundant representations in image processing, Proc. IEEE 98 (6) (2010) 972–982.
[41] H. Fan, J. Xiang, F. Ni, Multilayer feature combination for visual tracking, Asian Conf. Pattern Recognit. (2015) 589–593.
[42] Z. Wang, et al., Robust object tracking via multi-scale patch based sparse coding histogram, Multimed. Tools Appl. 76 (10) (2017) 12181–12203.

Long Xu (M'12) received his M.S. degree in applied mathematics from Xidian University, Xi'an, China, in 2002, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. He was a Postdoc with the Department of Computer Science, City University of Hong Kong, the Department of Electronic Engineering, Chinese University of Hong Kong, from July Aug. 2009 to Dec. 2012. From Jan. 2013 to March 2014, he was a Postdoc with the School of Computer Engineering, Nanyang Technological University, Singapore. Currently, he is with the Key Laboratory of Solar Activity, National Astronomical Observatories, Chinese Academy of Sciences. His research interests include image/video processing, wavelet, machine learning, and computer vision.

He was selected into the 100-Talents Plan, Chinese Academy of Sciences, 2014.

Xianyou Zeng is currently a Ph.D. Candidate in Institute of Information Science, Beijing Jiaotong University. He received the B.S. degree from South China Normal University, Guangzhou, in 2013. His research interests include object tracking, sparse representation and computer vision.

Ruizhen Zhao received his Ph.D. degree from Xidian University. After that he was with a postdoctoral fellow in Institute of Automation, Chinese Academy of Sciences. He is currently a professor and supervisor of doctor student. His research interests include wavelet transform and its applications, algorithms of image and signal processing, compressive sensing and sparse representation.

Lin Ma (M'13) is now a Researcher at Huawei Noah's Ark Lab, Hong Kong. He received his Ph.D. degree in Department of Electronic Engineering at the Chinese University of Hong Kong (CUHK) in 2013. He received the B.E., and M.E. degrees from Harbin Institute of Technology, Harbin, China, in 2006 and 2008, respectively, both in computer science.

He was a Research Intern in Microsoft Research Asia from Oct. 2007 to Mar. 2008. He was a Research Assistant with the Department of Electronic Engineering, CUHK, from Nov. 2008 to Jul. 2009. He was a Visiting Student with the School of Computer Engineering, Nanyang Technological University (NTU), from Jul. 2011 to Sep. 2011. His research interests lie in the areas of deep learning and multimodal learning, specifically for image and language, image/video processing and quality assessment.

He got the best paper award in Pacific-Rim Conference on Multimedia (PCM) 2008. He was awarded the Microsoft Research Asia fellowship in 2011. He was a finalist to HKIS young scientist award in engineering science in 2012.

Yigang Cen received the Ph.D. degree in control science engineering from Huazhong University of Science and Technology in 2006. He is currently a professor and a supervisor of doctor students with Beijing Jiaotong University, Beijing, China. His research interests include machine vision, compressed sensing, sparse representation, low-rank matrix reconstruction, and wavelet construction theory.