Dual-scale weighted structural local sparse appearance model for object tracking

Xianyou Zeng¹,², Long Xu³, Yigang Cen¹,², Ruizhen Zhao¹,² ∗, Wanli Feng¹,²

¹Institute of Information Science, Beijing Jiaotong University, Beijing 100044, People’s Republic of China
²Key Laboratory of Advanced Information Science and Network Technology of Beijing, Beijing 100044, People’s Republic of China
³National Astronomical Observatories, Chinese Academy of Sciences, Beijing 100012, People’s Republic of China
∗E-mail: rzhzhao@bjtu.edu.cn

Abstract: It is a great challenge to develop an effective appearance model for robust visual tracking due to various interfering factors, such as pose change, occlusion, background clutter etc. More and more visual tracking methods tend to exploit the local appearance model to deal with the above challenges. In this study, the authors present a simple yet effective weighted structural local sparse appearance model, which can better describe the target appearance information through patch-based generative weight. To further improve the robustness of tracking, they implement this appearance model on two-scale patches. The two derived appearance models are then combined to form a collaborative model to play their advantages. Extensive experiments on the tracking benchmark dataset show that the proposed method performs favourably against several state-of-the-art methods.

1 Introduction

Computer vision, which uses computer to simulate human vision system to perceive, understand and analyse the surrounding environment, has a wide range of applications, such as scene labelling [1], road detection [2], automatic recognition [3, 4], image labelling [5], person re-identification [6], human pose recovery [7, 8] and so on. Visual tracking, being one of the fundamental topics of computer vision, plays a critical role in numerous lines of research including video surveillance, vehicle navigation, motion analysis, aeronautics, and astronautics. Although significant progress has been made in the past few decades, visual tracking technology is still a challenging problem due to the numerous uncertain factors in the tracking process, such as partial occlusion (OCC), illumination change, background clutter, viewpoint variations etc.

Generally, an online tracking algorithm includes three main components: (i) a motion model, which models the temporal consistency of the states of the target and produces a number of candidate targets based on the target of the current frame for forecasting the target in the next frame; (ii) an appearance model, which is used to represent the tracked target in the tracking process; (iii) an observation model, which evaluates the likelihood of each candidate target being the true target and selects the best candidate as the tracking result. This study focuses on the appearance model, which is the core of a tracking algorithm.

Existing appearance models can be classified as either generative models (e.g. [9–15]) or discriminative models (e.g. [16–22]). Discriminative models cast tracking as a binary classification problem that aims to distinguish the tracked target from its surrounding background by employing the information from both positive and negative samples. Many classic and recent machine learning techniques could promote the progress of tracking algorithms, including boosting [17], support vector machine [23], naive bayes [20], random forest [24], multiple instance learning [18], metric learning [25], structured learning [19], latent variable learning [26], correlation filter [27] and so on.

Generative models formulate the tracking as a problem searching for the image observation with minimal reconstruction error or the most similar to the target model. In generative tracking methods, the object appearance representation is very important. Many representation schemes have been proposed, including template-based (see [11, 15, 28]), subspace-based (see [12, 29]), and sparse representation-based (see [13, 30–32]) models. In the template-based algorithms, colour histogram [33] and pixel intensity [34] are used to model object appearance for visual tracking. In recent years, subspace-based algorithms have been widely studied. Ross et al. [12] propose an incremental visual tracking (IVT) method, which represents the tracked target by using a low-dimensional principal component analysis (PCA) subspace and online updates the PCA subspace to capture the appearance change of the target. Motivated by the work of Wright et al. [35], some researchers introduce sparse representation into visual tracking and achieve great success. Mei and Ling [13] adopt a holistic appearance model of the target and track the target by solving a \( l_1 \) minimisation problem. Liu et al. [30] propose a method, which exploits a local appearance model based on histograms of sparse coefficients and the mean-shift algorithm for tracking.

In this study, we present an efficient tracking algorithm based on a dual-scale weighted structural local sparse appearance (WLSA) model. The proposed method fully takes into account the similar local structural and inner geometric layout information of the target in an image sequence. Local patches in a fixed spatial layout within a target region are extracted and encoded by a dictionary established beforehand. In the local appearance model, all individual patches are stacked together to form a complete representation of the target, which means each local patch can describe some extent of the target appearance information. Thus, patch-based weights representing the importance of local patches in describing the target appearance information are computed through the structural reconstruction error. Based on the patch-based weight and alignment-pooling method, the coding coefficients of the patches are integrated to obtain a robust representation of a target object. This helps locate the target accurately and handle partial OCC effectively. Combining spatial and structural information of different scales can make representation more discriminative and robust, so the final likelihood of one candidate region is estimated based on the association of patches extracted with two different scales. The contributions of this work are summarised as follows:
i. We propose a simple yet effective weight calculation method to help local patches better describe the target appearance information.

ii. A novel WLSA model is presented for robust representation.

iii. To further improve the tracking performance, the proposed WLSA model is implemented in two different patch scales.

iv. The proposed method is evaluated on a large benchmark dataset and achieves competitive results compared with some state-of-the-art methods.

The rest of this work is arranged as follows. In Section 2, we briefly introduce the related work. In Section 3, we describe the proposed method in detail. Section 4 presents the experimental results of the proposed method and some state-of-the-art methods. Finally, a conclusion about this work is given in Section 5.

2 Related work

Sparse representation has been widely studied and successfully applied in the field of visual tracking [13, 30–32, 36–40]. With sparsity constraints, a signal can be represented as a linear combination of only a few basis vectors. In [13, 31], a candidate target is sparsely represented by a linear combination of the atoms of a dictionary composed of target templates and trivial templates. The sparse representation problem is then solved through $l_1$ minimisation with non-negativity constraints. In [36], dynamic group sparsity constraints of spatial and temporal adjacency are incorporated to improve the efficiency and robustness of tracking algorithm. Liu et al. [30] employ a local sparse representation scheme to model the target appearance and use the sparse coding histogram to represent the basis parts of the target. Zhang et al. [40] propose a multi-task tracking algorithm, which exploits the interdependency between candidate regions based on joint group sparsity constraints. Jia et al. [41] extract local patches in a fixed spatial layout and adopt an alignment pooling across the local patches to obtain an adaptive structural local sparse appearance (ASLA) model for robust tracking. Zhao et al. [42] and Xie et al. [43] construct structural local sparse appearance models on patches of two scales and seven scales, respectively, and then associate these models for effective tracking. Our method is inspired by the work in [41–43]. First, we propose a novel object representation method, namely our WLSA model. For the purpose of increasing the tracking performance, the proposed WLSA model is then carried out on local image patches of two scales and constructs two local appearance models. Finally, we combine these two local models as a dual-scale appearance model by a collaborative mechanism.

3 Proposed tracking method

3.1 WLSA model

Most tracking methods use either a holistic template or local appearance model to represent the target. In this work, a local sparse representation is developed to model the appearance of target patches and represent them with the corresponding sparse coefficients. Given an object image, we can use a sliding window with a fixed size to extract a set of local image patches with a spatial layout. If we have a set of target templates $T = \{T_1, T_2, \ldots, T_n\}$, we extract local patches from each target template in the same way. Then a dictionary $D = \{d_1, d_2, \ldots, d_n\} \in \mathbb{R}^{q \times n \times N}$ to encode local patches inside the possible candidate regions can be obtained, where $q$ is the dimension of the image patch vector, $n$ is the number of target templates, $N$ is the number of local patches sampled within each target template. Each column of the dictionary $D$ is obtained by $l_2$ normalisation on a vectorised image patch extracted from $T$. The first $n$ frames are tracked using the nearest neighbour searching algorithm [44] and tracking results (warped to a canonical size of $32 \times 32$) are used to form the target template set $T$.

Similarly, we extract local patches within a candidate target and turn them into vectors in the same way, which are denoted as $Y = \{y_1, y_2, \ldots, y_N\} \in \mathbb{R}^{q \times N}$. Then, the following three stages are performed as shown in Fig. 1.

3.1.1 Sparse coding: With the dictionary $D$, each patch $y_i$ can be sparsely represented as a linear combination of the atoms of the dictionary by solving

$$\min_{b_i} \|y_i - Db_i\|^2 + \lambda \|b_i\|_1, \quad \text{s.t.} \quad b_i \geq 0,$$

where $\|\cdot\|_2$ and $\|\cdot\|_1$ denote the $l_2$ and $l_1$ normalisation, respectively, $\lambda$ is the regularisation parameter, $b_i \in \mathbb{R}^{N \times 1}$ is the sparse code of the patch $y_i$, and $b_i \geq 0$ means all the elements of $b_i$ are non-negative. $B = [b_1, b_2, \ldots, b_N]$ represents the sparse codes of $N$ local patches within the candidate region.

3.1.2 Accumulation and weighting: According to the target templates, the sparse coefficient vector of the patch $y_i$ is split into $n$ (the number of the target templates) segments, i.e. $b_i^T = [b_i^{(1T)}, b_i^{(2T)}, \ldots, b_i^{(NT)}]$, where $b_i^{(k)} \in \mathbb{R}^{q \times 1}$ denotes the $k$th segment of the coefficient vector $b_i$. These segmented coefficients are accumulated and normalised to obtain $v_i$ for the patch $y_i$

$$v_i = \frac{1}{C} \sum_{k=1}^{n} b_i^{(k)} = \frac{1}{C} \begin{bmatrix} v_{i1} \\ v_{i2} \\ \vdots \\ v_{iN} \end{bmatrix}, \quad i = 1, 2, \ldots, N,$$

where vector $v_i$ corresponds to the patch $y_i$ and $C$ is a normalisation term, $C = v_{i1} + v_{i2} + \cdots + v_{iN}$.

In local appearance model, each local patch represents a fixed part of the target object. Hence, each patch can describe the different extent of appearance information of the target, i.e. each patch has a different importance. Here, we introduce a set of weights to embody the importance of patches. After obtaining the sparse code $b_i$ of the patch $y_i$, we reformulate the sparse code $b_i$ as

![Flowchart of the proposed WLSA model](Image)
In order to make full use of the sparse code \( b_0 \), a penalty term \( \|D((1-o_0) \odot b_0)\|_2 \) is added to (4), so the reconstruction error \( e_i \) can be rewritten as follows:

\[
e_i = \|y_i - D(o_i \odot b_0)\|_2^2 + \gamma \|D((1-o_0) \odot b_0)\|_2,
\]

where \( \gamma \) is a control parameter. If the candidate is perfect, the local patch \( y_i \) can be represented well by the sub-dictionary \( D_0 \). At the same time, the penalty term will be very small, otherwise very large.

The weight of the patch \( y_i \) is calculated by

\[
w_i = \exp(-e_i).
\]

3.1.3 Alignment pooling: The candidate target contains \( N \) local image patches and all the vectors \( z_i \) of it can form a square matrix \( Z = [z_1, z_2, \ldots, z_N] \). According to the spatial layout of the target, the appearance variation of a local patch can be best described by the patches at the same position of the target templates (i.e., using the sparse coefficients with the aligned position). As in [41], we take the diagonal elements of the square matrix \( Z \) as the pooled features of the local patches of the candidate region

\[
f = \text{diag}(Z) = \begin{bmatrix} f_1 \vline w_1v_1 \vline \ldots \vline w_1v_N \\ f_2 \vline w_2v_2 \vline \ldots \vline w_2v_N \\ \vdots & \vdots & \ddots & \vdots \\ f_N \vline w_Nv_1 \vline \ldots \vline w_Nv_N \end{bmatrix}
\]

where \( f \) is the vector of pooled features, \( v_i \) reflects the structural similarity between the \( i \)th patch of the candidate target and the \( i \)th patch of the target object.
After the above three stages, the likelihood of the candidate target being the true target is measured by

$$ p = \sum_{k=1}^{N} f_k $$

$$ = f_1 + f_2 + \ldots + f_N $$

$$ = w_1v_1 + w_2v_2 + \ldots + w_Nv_N. $$

Different from the model in [41], which utilises the same weight computed by (7) and (8). In the third stage, the vector of pooled patch. With the sparsity assumption, each local patch thereby mitigating the influence of noise introduced by target appearance variations.

3.2 Dual-scale WLSA (DSWLSA) model

To further improve tracking performance, we adopt representations of two different patch sizes to model target appearance. We extract local patches of different sizes within target templates and construct dictionaries at different sizes, i.e. $D^s = \{d^s_i\}_{i=1:n \times N^s, s=1, 2}$, where $d^s_i \in R^{N^s \times 1}$ is the $i$th column representing a vectorised and $l_i$ normalised image patch on the $s$th scale, $N^s$ is the dimensionality of image patches on the $s$th scale, $N^s$ is the number of local image patches on the $s$th scale. Let $Y^s = \{y^s_i\}_{i=1:N^s, s=1, 2}$ be the vectorised image patches of different scales extracted from a candidate target, where $y^s_i \in R^{N^s \times 1}$ is the $i$th local patch under the $s$th scale. According to the first stage of Section 3.1, we generate sparse coefficients for each local patch. With the sparsity assumption, each local patch $y^s_i$ can be encoded with the dictionary $D^s$ by solving

$$ \min_{b^s_i} \| y^s_i - D^s b^s_i \|^2 + \lambda \| b^s_i \|_1, \quad \text{s.t.} \quad b^s_i \geq 0, $$

where $b^s_i \in R^{N^s \times N^s \times 1}$ is the corresponding sparse coefficient of patch $y^s_i$. After the accumulation and weighting operations of the second stage, a new square matrix $Z^s = [z^s_1, z^s_2, \ldots, z^s_{N^s}]$ can be obtained, where $z^s_i = w^s_i \times v^s_i$, $w^s_i$ is the weight of the patch $y^s_i$ computed by (7) and (8). In the third stage, the vector of pooled features under the $s$th scale can be obtained by

$$ f^s = \text{diag}(Z^s), s = 1, 2. $$

The detailed process of the DSWLSA model can be seen in Fig. 3.

3.3 Collaborative tracking model

In this study, object tracking is carried out within the particle filter framework. Let affine parameters $x_i = (l_x, l_y, \mu_x, \mu_y, \mu_z)$ represent the target state where $l_i$ and $l_i$ denote the horizontal and vertical translation, $\mu_i$ denotes the rotation angle, $\mu_x$ and $\mu_y$ are scale and aspect ratio, and $\mu_z$ is the skew parameter. Given the observation set of target $o_i = \{o_1, o_2, \ldots, o_l\}$ up to frame $t$, the current target state $x_i$ can be obtained by the maximum a posteriori estimation

$$ \hat{x}_t = \arg \max_{x} p(x_t|o_{1:t}), $$

where $x_t$ indicates the $i$th sample at frame $t$. Based on the Bayes theorem, the posterior probability $p(x_t|o_{1:t})$ can be estimated recursively by

$$ p(x_t|o_{1:t}) \propto p(o_t|x_t) \int p(x_t|x_{t-1})p(x_{t-1}|o_{1:t-1})dx_{t-1}, $$

where $p(x_t|x_{t-1})$ denotes the motion model and $p(o_t|x_t)$ denotes the observation model. The motion model $p(x_t|x_{t-1})$ describes the state transition of the target between consecutive frames and is often formulated as

$$ p(x_t|x_{t-1}) = N(x_t; x_{t-1}, \Sigma), $$

where $\Sigma$ is a diagonal covariance matrix whose elements are the variances of affine parameters.

The observation model $p(o_t|x_t)$ describes the likelihood of the observation $o_t$ at state $x_t$ belonging to the target class. It plays a key role in robust tracking. In our method, the observation model is constructed as

$$ p(o_t|x_t) \propto \eta \sum_{k=1}^{N} f^s_k + (1 - \eta) \sum_{k=1}^{N} f^s_k, $$

where the first term of the right side of the equation denotes the similarity between the candidate and the target on the first scale, the second term of the right side of the equation denotes the
Then, we reconstruct a new template by

\[ x = Ua + e = [U]e + \nu \]

(16)

where \( U \) is the matrix of PCA basis vectors, \( a \) is the coefficients of PCA basis vectors and \( \nu \) indicates the corrupted or occluded pixels in \( x \). As the error caused by OCC and noise is arbitrary and sparse, the problem in (16) can be solved as \( l_i \) regularised least square problem

\[
\min \| x - Hc \|_2^2 + \lambda \| c \|_1.
\]

(17)

where \( H = [U] \), \( c = [ae] \) and \( \lambda \) is the regularisation parameter. Then, we reconstruct a new template by

\[ T_{\text{new}} = Ua. \]

(18)

The new template \( T_{\text{new}} \) is used for updating the target template set \( T \). To do this, we generate a cumulative probability sequence \( L_p = \{0, 1/(2^{n-1} - 1), 3/(2^{n-1} - 1), \ldots, 1\} \) to assign different updating probability to the templates. A random number \( r \) is generated according to the uniform distribution on the unit interval \([0, 1]\). Then, we select the template to be discarded by determining which section the random number \( r \) lies in and add the new template \( T_{\text{new}} \) to the end of the target template set \( T \). The proposed tracking algorithm is summarised in Algorithm 1.

**Algorithm 1:** The proposed tracking algorithm

**Input:** Target state \( x_i \) in the first frame; the number of templates \( n \); update frequency \( \nu \).

**Output:** Tracked targets \( s_t, t = 2, \ldots, S \).

1: **Initialisation:**
   Template set \( T \) and estimated target set \( \Phi = \phi \).

2: **Construct dictionaries** \( D^1 \) and \( D^2 \) using local patches of two different scales of the template set.

3: **while** \( t = n + 1, \ldots, S \) **do**

4: Generate candidate targets and extract local patches with two different scales from each candidate.

5: For each candidate, compute the vectors of the pooled features \( f^1 \) and \( f^2 \) using (10), respectively.

6: Compute the observation likelihood of each candidate using (15).

7: Find the tracked target \( x_i \) with maximal observation likelihood.

8: Put the tracked target \( x_i \) in the estimated target set \( \Phi = \Phi_x \).

9: **if** size \( (\Phi) = \nu \) **then**

10: Update PCA basis and the mean vector using \( \Phi \), then \( \Phi = \phi \).

11: Reconstruct a new template \( T_{\text{new}} \) using (18) and update \( T \).

12: Reconstruct the dictionaries \( D^1 \) and \( D^2 \) with local patches of two different scales of the new template set \( T \).

13: **end if**

14: **end while**

**4 Experiments**

**4.1 Experimental setup**

The proposed algorithm is implemented in MATLAB and runs at 0.5 frames per second on an Intel Core i7-4770 CPU (3.4 GHz) machine. The number of templates \( n \) is 10. The \( l_i \) minimisation problem is solved with the SPAMS package [46]. The regularisation parameter \( \lambda \) and variable \( y \) are, respectively, set to 0.01 and 0.001. For the motion model, the variances of the affine parameters are set to \((l_1, l_2, \mu_1, \mu_2, \mu_3) = (6, 6, 0.01, 0, 0.005, 0)\). The number of samples for particle filter is chosen to be 600. In our experiments, the target and each candidate are warped to \( 32 \times 32 \) pixels. Local image patches of two different scales are extracted within the target and each candidate region with 8 pixels as step length. The first size is \( 16 \times 16 \) and the second size is \( 8 \times 8 \). For the template update, ten eigenvectors are used to implement incremental subspace learning and the update frequency \( \nu \) is set as 5. In our approach, ratio factor \( \eta \) in (15) is set to 0.2. All the parameters are fixed for all the experiments.

**4.2 Traditional evaluation**

In this work, we use 29 publicly available video sequences from the recent CVPR13 benchmark dataset [47]. These sequences contain different challenges, such as illumination variation (IV), partial OCC, scale variation (SV) etc. We compare the proposed method (DSWLSA) with nine state-of-the-art methods: ASLA [41], sparsity-based collaborative model (SCM) [48], IVT [12], struck tracking (Struck) [19], tracking learning detection (TLD) [21], transfer Gaussian processes regression (TGPR) [26], kernelised correlation filters (KCF) [27], visual tracking decomposition (VTD) [15], deep learning tracking (DLT) [49].

**4.2.1 Quantitative evaluation:** Two traditional metrics are used to evaluate the above-mentioned methods. The first metric is the centre error, which is defined as the Euclidean distance between the centre locations of tracked objects and their corresponding labelled ground truth. The second one is the overlap rate, usually defined as score = \((\text{area}(R_T \cap R_G))/\text{area}(R_T \cup R_G))\), where \( R_T \) is the tracking bounding box and \( R_G \) is the ground truth bounding box. Table 1 reports the average centre errors in pixels, where a smaller average error means a more accurate result. From Table 1, we can see that the proposed tracker ranks within the top three on 20 out of 29 test sequences. Table 2 presents the average overlap rates, where a larger value means a more accurate result. Table 2 shows the proposed tracker ranks within the top three on 23 out of 29 test sequences. From Tables 1 and 2, the proposed tracking algorithm achieves very excellent performance on the most of the test sequences. However, there are outliers (nine sequences in Table 1 and six sequences in Table 2). The reason may be twofold. First, we employ a fixed instead of adaptive ratio factor for connecting dual scales in the proposed method, which compromises the performance of the proposed method on some sequences. Second, the proposed method is not robust for low resolution (LR), motion blur (MB), fast motion (FM) and out of view, which can be seen from the quantitative results of ‘dudek’ and ‘boy’ sequences.
Table 1: Average centre error (in pixels)

| Sequence | ASLA | SCM | IVT | Struck | TLD | TGPR | KCF | VTD | DLT | Ours |
|----------|------|-----|-----|--------|-----|------|-----|-----|-----|------|
| doll     | 11.84| 3.45| 32.65| 8.92   | 6.01| 5.97 | 8.27| 7.29| 5.70| 3.02 |
| mhyang   | 1.70 | 2.41| 1.87 | 2.59   | 9.51| 4.43 | 3.92| 4.35| 2.43| 2.31 |
| sylvester| 15.23| 7.97| 34.17| 6.30   | 7.31| 5.71 | 13.30| 19.60| 19.70| 6.64 |
| svu      | 74.81| 4.56| 57.31| 49.81  | 13.03| 89.47| 5.31| 57.24| 4.40| 4.36 |
| dog1     | 4.87 | 7.00| 3.46 | 5.66   | 4.19| 5.86 | 4.15| 4.11| 5.00| 4.39 |
| dudek    | 15.26| 10.77| 9.62 | 11.45  | 18.05| 17.27| 11.38| 10.30| 8.81| 13.97|
| faceocc1 | 78.06| 13.04| 18.42| 18.78  | 27.37| 13.73| 15.98| 20.20| 22.72| 14.22|
| faceocc2 | 19.34| 9.02| 7.42 | 5.96   | 12.28| 7.56 | 7.67 | 8.29 | 10.87| 8.34 |
| car4     | 1.59 | 4.27| 2.15 | 8.69   | 12.84| 6.11 | 9.47 | 36.99| 2.95 | 2.61 |
| boy      | 106.07| 51.02| 91.25| 3.84   | 4.49| 3.38 | 2.67 | 5.77 | 2.53 | 8.37 |
| woman    | 139.75| 7.88| 176.51| 4.17   | 139.94| 10.38| 10.09| 118.86| 9.11 | 7.61 |
| trellis  | 7.58 | 7.01| 119.78| 6.92   | 31.08| 7.29 | 8.21 | 32.25| 86.23| 3.07 |
| walking2 | 37.42| 1.65| 2.46 | 11.16  | 44.56| 5.90 | 59.57| 46.24| 2.18 | 2.51 |
| girl     | 3.28 | 2.60| 22.46| 2.57   | 9.77 | 7.70 | 11.92| 8.60 | 10.09| 3.49 |
| david2   | 1.45 | 3.41| 1.17 | 1.50   | 4.98 | 2.05 | 2.29 | 2.85 | 10.43| 1.40 |
| david    | 5.07 | 4.34| 4.82 | 42.80  | 5.12 | 5.31 | 8.06 | 11.59| 66.19| 3.54 |
| fish     | 3.85 | 8.54| 5.67 | 3.40   | 6.54 | 5.62 | 4.08 | 16.79| 26.46| 3.72 |
| freeman3 | 3.17 | 3.19| 35.76| 16.83  | 29.33| 88.28| 19.57| 23.95| 4.00 | 2.01 |
| skating1 | 59.86| 16.38| 247.02| 82.94  | 145.45| 81.57| 7.67 | 2.13 | 95.32| 38.38 |
| walking1 | 1.89 | 2.49| 1.61 | 4.62   | 10.23| 5.01 | 4.26 | 5.78 | 15.37| 1.81 |
| singer1  | 3.29 | 2.72| 11.31| 14.51  | 7.99 | 120.29| 12.59| 4.19 | 3.04 | 3.95 |
| jogging2 | 169.86| 4.15| 138.22| 107.69 | 13.56| 5.39 | 144.03| 122.19| 79.70| 2.97 |
| freeman1 | 105.66| 6.87| 11.64| 14.27  | 39.70| 9.34 | 94.62| 10.32| 104.78| 8.04 |
| football | 15.00| 16.30| 14.34| 17.31  | 14.26| 5.94 | 14.80| 13.64| 191.46| 5.71 |
| cardark  | 1.54 | 1.30| 8.43 | 0.95   | 27.47| 2.13 | 5.76 | 16.47| 18.82| 1.16 |
| carscale | 24.64| 33.38| 11.90| 36.43  | 22.60| 21.41| 16.14| 38.45| 22.65| 10.86 |
| mountainBike | 8.98 | 10.59| 7.66 | 8.62   | 216.13| 6.20 | 7.74 | 9.78 | 15.05| 8.76 |
| subway   | 138.37| 3.45| 130.78| 4.46   | 149.76| 3.04 | 3.17 | 141.31| 170.81| 2.93 |
| crossing | 1.85 | 1.57| 2.36 | 2.81   | 24.34| 7.07 | 2.42 | 26.13| 1.65 | 1.66 |
| average  | 36.60| 8.67| 41.80| 17.45  | 36.48| 19.29| 16.80| 29.02| 33.63| 5.75 |

Top three results are shown in bold font.

4.2.2 Qualitative evaluation: IVs: Fig. 4 presents the tracking results in several sequences with IVs. In the fish sequence, the target undergoes obvious light changes. At the same time, the camera moves fast. The DLT method cannot perform well from the start of the sequence (e.g. #240). SCM and VTD methods perform unstably and show deviation away from the target position during the tracking (see #360 and #420). In contrast, the other seven methods successfully track the target from the start to the end. For the Sylvester sequence, the Struck, TGPR and our DSWLSA method perform better than other methods and achieve more accurate tracking results. The target in the doll sequence undergoes a long time scale and illumination changes. The VTD, KCF, TGPR, and IVT methods fail to track the target at the end (seen from #240, #350 and #500). The TLD method deviates away from the target when the girl rotates her head (e.g. #240 and #350). The ASLA, Struck, SCM and our DSWLSA tracker persistently track the target throughout the whole sequence. In the carscale sequence, the target meets with rotation and OCC. The ASLA, IVT, DLT and our DSWLSA method perform better than other methods. The boy sequence is challenging due to many factors: FM, MB, as well as rotation. The ASLA, SCM and IVT trackers perform poorly (seen from #340, #500 and #602). Unlike the mentioned three methods, the other seven methods robustly win multiple challenges and steadily track the target to the end.

OCC: For visual tracking, OCC is one of the most common and critical challenges. Fig. 6 illustrates tracking results from six challenging sequences where the targets are severely or long-term occluded. In the faceocc1 sequence, the woman occludes her face with a book frequently. Except for the ASLA method, the other methods are able to track the target to some extent but the SCM, TGPR and our DSWLSA method perform favourably against other methods. The target in the faceocc2 sequence undergoes heavy OCC. Most trackers can successfully track the target from the start to the end. However, the ASLA method drifts to the background at the end (see #812). In the jogging2 sequence, a person is almost completely occluded by the lamp post for a short term. The ASLA, DLT, Struck, VTD, KCF and IVT methods cannot obtain the target again and undergo large drift after the person goes across the obstacle (see #100 and #150). The TLD method locks to another person at frame #100, but it can re-detect the target in the...
Table 2 Average overlap rate

| Sequence  | ASLA | SCM | IVT | Struck | TLD | TGPR | KCF | VTD | DLT | Ours |
|-----------|------|-----|-----|--------|-----|------|-----|-----|-----|------|
| doll      | 0.82 | 0.83| 0.43| 0.55  | 0.57| 0.57 | 0.57| 0.53| 0.65| 0.84| 0.85 |
| mhyang    | 0.92 | 0.81| 0.80| 0.82  | 0.63| 0.77 | 0.77| 0.80| 0.73| 0.89| 0.84 |
| sylvester | 0.59 | 0.69| 0.52| 0.72  | 0.67| 0.73 | 0.73| 0.64| 0.62| 0.46| 0.70 |
| sv        | 0.49 | 0.75| 0.41| 0.52  | 0.69| 0.48 | 0.48| 0.88| 0.45| 0.78| 0.79 |
| dog1      | 0.72 | 0.70| 0.74| 0.55  | 0.59| 0.60 | 0.70| 0.73| 0.80| 0.90| 0.79 |
| dudek     | 0.74 | 0.77| 0.75| 0.73  | 0.65| 0.70 | 0.77| 0.75| 0.68| 0.59| 0.78 |
| faceocc1  | 0.32 | 0.79| 0.73| 0.73  | 0.58| 0.77 | 0.77| 0.75| 0.68| 0.59| 0.78 |
| faceocc2  | 0.65 | 0.73| 0.73| 0.79  | 0.62| 0.77 | 0.77| 0.75| 0.74| 0.65| 0.76 |
| car4      | 0.75 | 0.76| 0.87| 0.49  | 0.63| 0.50 | 0.48| 0.36| 0.88| 0.72 |
| boy       | 0.37 | 0.38| 0.26| 0.76  | 0.66| 0.75 | 0.79| 0.63| 0.83| 0.74 |
| woman     | 0.15 | 0.66| 0.15| 0.73  | 0.13| 0.76 | 0.72| 0.14| 0.59| 0.70 |
| trellis   | 0.80 | 0.67| 0.25| 0.61  | 0.48| 0.61 | 0.63| 0.45| 0.26| 0.87 |
| walking2  | 0.37 | 0.81| 0.79| 0.51  | 0.31| 0.60 | 0.39| 0.33| 0.82| 0.83 |
| girl      | 0.71 | 0.68| 0.17| 0.75  | 0.57| 0.67 | 0.55| 0.55| 0.54| 0.72 |
| david2    | 0.90 | 0.74| 0.70| 0.87  | 0.69| 0.82 | 0.81| 0.69| 0.48| 0.88 |
| david     | 0.75 | 0.72| 0.64| 0.24  | 0.71| 0.59 | 0.54| 0.56| 0.25| 0.80 |
| fish      | 0.85 | 0.75| 0.77| 0.86  | 0.81| 0.82 | 0.84| 0.56| 0.41| 0.87 |
| freeman3  | 0.75 | 0.72| 0.39| 0.26  | 0.44| 0.05 | 0.32| 0.30| 0.69| 0.81 |
| skating1  | 0.50 | 0.47| 0.07| 0.31  | 0.19| 0.41 | 0.49| 0.53| 0.43| 0.60 |
| walking   | 0.77 | 0.71| 0.76| 0.57  | 0.45| 0.59 | 0.53| 0.61| 0.41| 0.78 |
| singer1   | 0.79 | 0.87| 0.57| 0.36  | 0.73| 0.23 | 0.35| 0.49| 0.85| 0.78 |
| jogging2  | 0.14 | 0.73| 0.14| 0.20  | 0.66| 0.77 | 0.12| 0.13| 0.13| 0.76 |
| freeman1  | 0.27 | 0.61| 0.43| 0.34  | 0.28| 0.41 | 0.22| 0.35| 0.28| 0.60 |
| football  | 0.53 | 0.48| 0.55| 0.53  | 0.49| 0.70 | 0.54| 0.56| 0.23| 0.67 |
| cardark   | 0.85 | 0.64| 0.66| 0.89  | 0.45| 0.81 | 0.62| 0.54| 0.59| 0.87 |
| carscale  | 0.61 | 0.59| 0.63| 0.41  | 0.45| 0.41 | 0.42| 0.43| 0.62| 0.59 |
| mountainBike | 0.72 | 0.67| 0.73| 0.71  | 0.20| 0.75 | 0.71| 0.70| 0.47| 0.74 |
| subway    | 0.19 | 0.72| 0.17| 0.65  | 0.18| 0.69 | 0.74| 0.16| 0.02| 0.73 |
| crossing  | 0.78 | 0.77| 0.30| 0.68  | 0.40| 0.62 | 0.70| 0.32| 0.72| 0.79 |
| average   | 0.81 | 0.70| 0.52| 0.59  | 0.51| 0.62 | 0.59| 0.51| 0.56| 0.79 |

Top three results are shown in bold font.

Fig. 4 Tracking results on several sequences with IVs

subsequent frames. The TGPR, SCM and our DSWLSA method perform well and precisely track the person to the end of the sequence. In the suv sequence, the target is heavily occluded by dense tree branches and another bus. The IVT and Struck methods are sensitive to OCC and cannot track the target properly (see #95 and #550). The ASLA method fails to track the target from frame #700 to the end. The TGPR, VTD and TLD methods lose the target at frame #550, but the TLD method is able to reacquire the target and track the target to the end. The SCM, KCF, DLT and our DSWLSA method successfully track the target throughout the entire sequence. In the woman sequence, the person is partially occluded by cars for a long time. The KCF, TGPR, SCM, Struck, DLT and our DSWLSA method perform well and obtain satisfying results. In the walking2 sequence, the walking woman undergoes a long-term OCC by a man together with SV. The TLD, VTD, ASLA and KCF methods severely drift away when the woman reappears again (e.g. #240). The TGPR and Struck methods cannot handle SV well (e.g. #400 and #500). The SCM, IVT, DLT and our DSWLSA method perform better compared with other methods and achieves more accurate results.

SVs: Fig. 7 shows some representative results on five sequences with SVs. The singer1 sequence is very challenging as the target moves far away from the camera with a large-scale change. Furthermore, the stage light changes drastically and the background is cluttered. The TGPR method performs poorly when significant IVs occur (e.g. #140, #200 and #350). The IVT method slightly deviates from the target position (e.g. #140, #200 and #250). The Struck, KCF and VTD methods fail to track the SV of the target. Unlike the above methods, the ASLA, TLD, SCM, DLT and our DSWLSA method perform very well in this sequence. For the walking sequence, the ASLA, IVT and our DSWLSA method achieve better performance. In the freeman1 sequence, a person undergoes a large SV and rotation in his face appearance. The DLT, KCF, ASLA and TLD methods cannot track the target from frame #150 to the end. The IVT method loses the target finally (e.g. #326). The VTD, TGPR and Struck methods perform unstably and drift away from the target in the tracking process (e.g. #90 and #150). The SCM and our DSWLSA method can track the target well. In the freeman3 sequence, a person moves towards the camera with a large SV in his face. Along with the scale, changes are pose variation and LR, which make the tracking task more difficult. The VTD, IVT and Struck methods drift to the
background regions (e.g. #400 and #460). The TGPR method fails to track the target from the beginning of the sequence (e.g. #50). The TLD method cannot stably track the target and loses track of the target, which can be seen from frames #140, #400 and #460. The KCF method unsuccessfully keeps track of the target at the end (e.g. #460). The ASLA, SCM, DLT and our DSWLSA method successfully track the target throughout the entire sequence but our DSWLSA method performs best.

Background clutters (BCs): Fig. 8 shows the performance comparison of all the trackers in handling the challenge of background clutter on four sequences. In the subway sequence, the DLT method cannot track the target correctly from the start of this sequence (see #20). The VTD, ASLA, IVT and TLD methods fail to track the target and drift away to the background when the tracking person is occluded by another walking person (e.g. #42). The SCM, Struck, TGPR, KCF and our DSWLSA method precisely track the target to the end. In the football sequence, the scene is cluttered. The similarity between the target and the background makes the tracking more challenging. Only the TGPR and our DSWLSA method robustly track the target at all frames. The trellis sequence contains many challenging factors, such as background clutter, OPR, and IV. The TLD and VTD methods perform unstably and some of their tracking results shake away frequently (see #400 and #500). The DLT and IVT methods fail to track the target after 200 frames (e.g. #270, #400 and #500). The ASLA method does not perform well at frame #400. The SCM and our DSWLSA method lock the target more stably and obtain better accuracy than the Struck, TGPR and KCF methods. For the dudek sequence, all the methods carry out well in this sequence. In the MountainBike sequence, the target meets with background clutter as well as rotation. Except for the TLD method, the other nine methods are able to reliably track the target to the end.

Deformation (DEF): In Fig. 9, sequences skating1, mhyang, david and crossing are selected to evaluate the robustness of trackers against non-rigid DEF. In the skating1 sequence, the dancer continuously changes pose. At the same time, the target suffers from drastic illumination change and background clutter. The IVT method performs poorly (see #50, #150 and #200). The TLD is unable to persistently locate the target (e.g. #150, #230 and #310). The TGPR, ASLA, Struck, SCM and DLT methods severely deviate from the target in the end (see #310). The VTD and our DSWLSA method successfully track the target and achieve the most stable performance. In the mhyang sequence, the ASLA, DLT and our DSWLSA method perform better than other methods and achieve more excellent tracking results. In the david sequence, a person moves from a dark room to a bright area with a large IV and he changes the orientation of his face from time to time. The DLT and Struck methods fail to lock the target (see #459, #599, #649 and #770). The ASLA, SCM and our DSWLSA method achieve...
the best performance. For the crossing sequence, non-rigid DEF, scale, and IVs are the main challenges. The TLD and VTD methods lose the target (see #80 and #120). The IVT method is less effective for this sequence (e.g. #60, #80 and #120). The rest seven methods successfully track the target until the end and our DSWLSA method achieves the highest accuracy.

4.3 Benchmark evaluation

In this subsection, we evaluate our algorithm on the recent CVPR13 benchmark [47]. The precision plot and the success plot are applied to evaluate all the compared trackers. The success plot is based on the overlap rate and shows the percentage of frames whose overlap rate is higher than a given threshold \( t_0 \in [0, 1] \). The area under the curve (AUC) of each success plot is often used to rank the tracking algorithms. The precision plot demonstrates the percentage of frames where the tracked location is within the given threshold distance to the ground truth. In Fig. 10, we report the results of one pass evaluation based on the average success and precision rate for the ten trackers. It can be seen from this figure that our DSWLSA method achieves very promising tracking performance (ranks 1 on the success rate and ranks 2 on the precision rate) in the benchmark evaluation.

In the recent CVPR13 benchmark, all the 50 video sequences are annotated with 11 different attributes: LR, in-plane rotation (IPR), OPR, SV, OCC, DEF, BCs, IV, MB, FM, and out-of-view (OV). Results on these annotated attributes can be used to analyse the strength and weakness of each tracker. Fig. 11 shows the success plot of each attribute. We note that the proposed tracker achieves top three results on 7 out of 11 attributes, and outperforms the baseline ASLA method on all 11 attributes. On the sequences with attributes of OPR and SV, the proposed tracker ranks first among all the evaluated trackers. For the sequences with attributes of OCC and IPR, the proposed tracker ranks second among all the evaluated trackers with a narrow margin (<1%) to the best performing method KCF. On the sequences with IV, DEF and background clutter attributes, the proposed tracker ranks third which follows the KCF and TGPR methods. The proposed tracker ranks outside of the top three for the sequences with MB and FM attributes since it employs a simple motion model based on...
Table 3 Speed comparison of our tracker with nine state-of-the-art methods on the recent CVPR13 benchmark

| Trackers | ASLA | SCM | IVT | Struck | TLD | TGPR | KCF | VTD | DLT | Ours |
|----------|------|-----|-----|--------|-----|------|-----|-----|-----|------|
| speed, fps | 1.4  | 0.4 | 27.7 | 10.0   | 21.7| 0.7  | 233.7| 3.8 | 8.6 | 0.5  |

Fig. 12 Comparison of the tracking performance of the proposed algorithm with different ratio factors on the recent CVPR13 benchmark

Fig. 13 Comparison of the tracking performance of the ASLA and WLSA models on the recent CVPR13 benchmark

Fig. 14 Success plots and precision plots of dual-scale model and two single-scale models on the recent CVPR13 benchmark

stochastic search. We expect to use the optical flow to obtain the prior knowledge of the target location, as done in [9] and insert it for more robust tracking. For the sequences with an OV attribute, the proposed tracker ranks in the middle because we do not restore the target once it moves out of the view. Since we build target template and candidate target with simple grey-scale features, the proposed tracker does not achieve ideal performance on the sequences of LR attribute and ranks sixth.

Table 3 reports the comparison of tracking speed in terms of frame per second (fps). All testing trackers are implemented under the same configuration for fair comparison. From Table 3, KCF, IVT and TLD rank in the top three. The proposed tracker performs not well since it is not optimised regarding speed in this work.

4.4 Proposed algorithm with different ratio factors

The ratio factor \( \eta \) is a crucial parameter in our method, which controls the trade-off between the contributions of two different scale models. In this subsection, we investigate the tracking performance of the proposed algorithm with different ratio factors in the collaborative model. Fig. 12 shows the quantitative results on the recent CVPR13 benchmark. Experimental results show that the proposed algorithm achieves the best performance when the value of the ratio factor \( \eta \) is 0.2.

4.5 Effectiveness of patch-based weight

In this subsection, we present experimental results to validate the effectiveness of the proposed patch-based weight. We evaluate ASLA model and our WLSA model on the recent CVPR13 benchmark. Here, our WLSA tracker and the baseline ASLA tracker use the default patch size of 16 × 16. Fig. 13 shows the precision plots and success plots of these two trackers. From the results, we can see that our WLSA tracker improves the tracking performance of the original ASLA tracker by 7.6% and 9.9% in success score and precision score, respectively. That can be attributed to the added patch-based weight. The WLSA model weights the contaminated local patch less while the reliable patch more, which reduces the influence of noise caused by target appearance variations.

4.6 Validation of dual-scale design

In this subsection, we present experimental results to validate the effectiveness of our dual-scale design. Here, we use three trackers, including the dual-scale model DSLSA and two single-scale models. WLSA (8 × 8) and WLSA (16 × 16) are utilised to denote the two single-scale models. Fig. 14 shows that the dual-scale tracker performs more robustly than the two single-scale trackers, which demonstrates the effectiveness of our dual-scale design.

5 Conclusion

In this study, we propose an effective WLSA model, which fully exploits structural and local information of the target through the patch-based weight and alignment pooling method. Then, this appearance model is carried out on two scale patches to model target appearance, respectively. The contributions of the two derived appearance models are integrated in a unified manner to construct the final dual-scale tracker, which increases the accuracy and robustness of tracking, on account of absorbing their advantages. Quantitative and qualitative comparisons with numerous state-of-the-art methods on a large benchmark dataset demonstrate the effectiveness and superiority of the proposed algorithm. Further work includes investigating a particle selection method to speed up the proposed tracker. In addition, we are planning to utilise efficient re-detection mechanism and location prior of online learning for more effective tracking.

6 Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (NSFC) under grant nos. 61572461, 11790358, 11435006 and 61872034, and the CAS ‘100-Talents’ (Dr Xu Long).

7 References

[1] Wang, Q., Guo, J., Yuan, Y.: ‘A joint convolutional neural networks and context transfer for street scenes labeling’, IEEE Trans. Intell. Trans. Syst., 2018, 19, (5), pp. 1457–1470
[2] Wang, Q., Gao, J., Yuan, Y.: ‘Embedding structured contour and location prior in strengthened fully convolutional networks for road detection’, IEEE Trans. Intell. Trans. Syst., 2018, 19, (1), pp. 230–241
[3] Tao, D., Guo, Y., Li, Y., et al.: ‘Tensor rank preserving discriminant analysis for facial recognition’, IEEE Trans. Image Process., 2018, 27, (1), pp. 325–334
[4] Hong, C., Yu, J., You, J., et al.: ‘Multi-view ensemble manifold regularization for 3D object recognition’, Inf. Sci., 2015, 320, (1), pp. 395–405
[5] Tao, D., Cheng, J., Gao, X., et al.: ‘Robust sparse coding for mobile image labeling on the cloud’, IEEE Trans. Circuits Syst. Video Technol., 2017, 27, (1), pp. 62–72
[6] Tao, D., Guo, Y., Yu, B., et al.: ‘Deep multi-view feature learning for person re-identification’, IEEE Trans. Circuits Syst. Video Technol., 2017, 99, (PP), pp. 1–1
[7] Hong, C., Yu, J., Wan, J., et al.: ‘Multimodal deep autoencoder for human pose recovery’, IEEE Trans. Image Process., 2015, 24, (12), pp. 5659–5670
