J. Vis. Commun. Image R. 64 (2019) 102602

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Part-based visual tracking via structural support correlation filter ☆

Zhangjian Ji [a,b,c,*], Kai Feng [a,b], Yuhua Qian [a,b,c]

[a] School of Computer & Information Technology, Shanxi University, Taiyuan, China
[b] Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Shanxi University, Taiyuan, China
[c] Institute of Big Data Science and Industry, Shanxi University, Taiyuan, China

ARTICLE INFO

Article history:
Received 24 June 2018
Revised 19 July 2019
Accepted 10 August 2019
Available online 14 August 2019

Keywords:
Object tracking
Support vector machines
Correlation filter
Structural learning
Temporal consistency
Scale estimation

ABSTRACT

To better deal with the partial occlusion issue and improve their efficiency of part-based and support vector machines (SVM) based trackers, we propose a novel part-based structural support correlation filter tracking method, which absorbs the strong discriminative ability from SVM and the excellent property of part-based tracking methods which is less sensitive to partial occlusion. Then, our proposed model can learn the support correlation filter of each part jointly by a star structure model, which preserves the spatial layout structure among parts and tolerates outliers of parts. In addition, our model introduces inter-frame consistencies of local parts to mitigate the drift problem. Finally, our model can accurately estimate the scale changes of object by the relative distance change among reliable parts. The extensive empirical evaluations on three benchmark datasets: OTB2015, TempleColor128 and VOT2015 demonstrate that the proposed method achieves comparable performance against several state-of-the-art trackers and runs in real time.

1. Introduction

Visual object tracking has been an important research topic in the computer vision field and has a wide range of practical applications, *e.g.*, intelligent surveillance, autonomous navigation of vehicles, human computer interaction, action recognition. Although great progress has been made in the past decades, it is still a challenging problem to design a robust visual tracking algorithm for real scenes, due to some complex situations, *e.g.*, partial occlusion, illumination variation, pose changes, background clutter, complex motion and object blur. Here, we mainly investigate the key problem of learning a robust tracking model under these conditions mentioned above.

As is known, the discriminative models [1–5] have better performance than the generative models [6–10] in visual tracking. They seek to design a robust classifier to detect the target, and establish an optimal mechanism to update the model at each frame. For example, in order to realize the visual tracking, Avidan [3] adopted the SVM as an off-line binary classifier to detect target at each frame. Hare et al. [2] applied the SVM with structured output to tracking the target because of its success in object detection. Although these two methods obtain the good results in visual tracking, the complex optimization still brings them the high computational complexity, which would make them not meet real-time applications, especially when considering the scale change of target and increasing feature dimensions of target representation. Recently, correlation filter (CF) utilizing the circulant property of dense sampling of base sample has attracted extensive attention in visual tracking due to its significant computational efficiency and robustness. Nevertheless, how to exploit the circulant property to accelerate SVM-based trackers remains unaddressed. Later, in view of the success of the max-margin CF (MMCF) [11] in the localization and classification of image, Zuo et al. [12] developed the novel discriminative tracking algorithms based on support correlation filters that perform efficiently and accurately. Although obtained competitive results both in accuracy and robustness, all these methods are sensitive to the occlusion or partial occlusion.

To deal with the above issues, deformable part-based tracking methods [13–17] become more popular partially because of their favorable property of robustness against partial occlusion. Yao et al. [17] employed an online structured output learning with latent variables to learn the weight parameters for an object and its parts, and distinguish the target object from the background using the weight parameters consequently. But their method fails to resolve the high computational complexity of the SVM. The researchers in [14,15] brought the correlation filter into the part-based tracking methods which improves the tracking efficiency and robustness. However, their approaches ignore the spatial relations among object parts. More recently, Liu et al. [16] improved the performance of their tracker by introducing struc-

☆ This paper has been recommended for acceptance by Zicheng Liu.

* Corresponding author.
E-mail address: jizhangjian@sxu.edu.cn (Z. Ji).

https://doi.org/10.1016/j.jvcir.2019.102602
1047-3203/

tural constraints among parts into correlation filter. But they also don't consider the temporal consistency of motion model which would help to alleviate the problem of drift away from object.

Considering the existing problems of the methods mentioned above, in this paper, we build an efficient part-based support vector correlation filter tracking algorithm which is able to deal with partial occlusion and deformation effectively. Our method adopts the support vector correlation filer as the classifier of each part which absorbs strong discriminative ablility from SVM and speeds-ups the SVM by the FFT in the Fourier domain. Then, our proposed model can learn the support correlation filter of each part jointly by a star structure model, which preserves the spatial layout structure among parts and tolerates outliers of parts. To further enhance the robustness of our model, we take into consideration the temporal consistency of each part, and incorporate it into our model to mitigate the issue of drift away from object. In addition, in order to adapt our tracker to scale changes of tracked target, we estimate the scale changes of object by the relative distance changes of the reliable part pairs. Finally, different from other multi-part trackers, we only estimate the position of the whole object by the tracking results of all visible parts, where each part is distinguished whether to be occluded by the PSR and appearance similarity.

2. Related work

In this section, we only introduce the methods closely related to this work: SVM-based trackers, correlation filter trackers and part-based trackers in detail. For a survey of more tracking methods, we refer the reader to [18–20].

SVM-based tracker: Babenko et al. [1] employed an online Multiple Instance Learning based appearance model to resolve the sample ambiguity problem. Hare et al. [2] used the structure SVM with kernels to track the whole target. Li et al. [21] utilized the structure SVM to predict the object location in RGB-T tracking. In [22], an explicit feature mapping function is used to approximate nonlinear kernels. However, the complex optimization of SVM still brings them the high computational complexity, which prevents them from using the higher dimensional features. In 2013, Henriques et al. [23] first applied the circulant property for training of support vector regression efficiently to detect pedestrians. Inspired by this work, Zuo et al. [12] adopted the circulant property to design the support correlation filter tracker that perform efficiently and accurately, which lower the computational complexity $\mathcal{O}(n^4)$ of SVM based trackers to $\mathcal{O}(n^2\log(n))$ for an $n \times n$ image patch. Wang et al. [24] proposed a novel structured SVM based tracking method which takes dense circular samples into account in both training and detection processes.

Correlation filter trackers: Bolme et al. [4] first introduced the correlation filter into the visual tracking field because it can achieve the appealing results in terms of accuracy, robustness and speed. Afterwards, Henriques et al. [25] incorporated multi-channel features into their kernelized correlation filters (KCF) framework to improve the accuracy and robustness of the tracker. However, they are only limited to estimate the target translation and signify poor performance when the targets of sequences involve significant scale variations. Thus, in order to adapt to the scale changes of the tracked target, Montero et al. [26] use a similar approach (scale ratios between matched relevant keypoints) as in TLD [27,28] to estimate the size of tracked target. Danelljan et al. [29] proposed a separable scale filter based on a scale pyramid representation to estimate the scale variation of target. And Li et al. [30] adopted a multiple scales searching strategy to surmount the limitation that the conventional correlation filter (CF) trackers cannot handle the scale variation of tracked target. Li et al. [31] proposed a multi-view correlation tracker which fused several features and selected the more discriminative features to do tracking in order to avoid drifts. Although the traditional correlation filter has obtained great success, unwanted boundary effects produced by the Fast Fourier Transform (FFT) result in an inaccurate description of the image, which will severely degrade the discriminative power of the learned model. To resolve this issue, Galoogahi et al. [32] chose a larger searching size and then cropped the central patch of the signal that is same as the size of the filter by the binary matrix $\mathbf{P}$ in each Alternating Direction Method of Multipliers (ADMM) iteration. Danelljan et al. [33] utilized a spatially regularized component to deal with the boundary effect caused by the FFT, which achieves better tracking accuracy.

Part-based tracker: To deal with the occlusion, many part-based trackers divided the entire target into separate parts [34,14,15,13,16,35,21,36,37]. Liu et al. [15] adapted the correlation filter as part classifiers. Li et al. [21] learned a dynamic graph model according to the intrinsic relationship among image patches. Akin et al. [35] proposed a deformable part-based correlation filter tracking approach which depends on coupled interactions between a global filter and several part filters. Lukežič et al. [38] presented a new class of layered part-based trackers that apply a geometrically constrained constellation of local correlation filters for object localization. He et al. [36] proposed a robust tracker based on key patch sparse representation (KPSR) to reduce the disturbance of partial occlusion or unavoidable background information. Sun et al. [37] proposed a shape preserved kernelized correlation filter (SP-KCF) which can accommodate target shape information for robust tracking.

3. Structural support correlation filter tracker

In this section, we present an efficient part-based support vector correlation filter tracking algorithm. Since the proposed approach works in the framework of support correlation filter, we first briefly review the theory of support correlation filter in Section 3.1. Then, in Section 3.2, we deduce the support correlation filter model in nonlinear space. Subsequently, in Section 3.3, we give a detailed description of our proposed part-based structural support correlation filter tracker. Next the detailed solving procedures of our tracking approach are deduced in Section 3.4. Finally, in Section 3.5, we introduce a valid method that estimates the scale changes of object. Meanwhile, we also present a model update strategy by using the feedback from tracking results to avoid the model corruption.

In order to make our paper more readable, we first define some generic notations that will be useful before deriving our model, which is shown in Table 1.

3.1. Review of support correlation filter

Given a vectorized image patch $\mathbf{x} \in \mathbb{R}^{MN}$, Zuo et al. [12] learn a support correlation filter $\mathbf{w}$ and a bias b to classify any circular shift image $\mathbf{x}_{m,n}$ of $\mathbf{x}$ by

$$y_{m,n} = sgn\left(\mathbf{w}^T\mathbf{x}_{m,n} + b\right), \tag{1}$$

Note that $m \in \{0,1,\ldots,M-1\}$ and $n \in \{0,1,\ldots,N-1\}$. $y_{m,n}$ denotes corresponding class label of one possible observation $\mathbf{x}_{m,n}$ of a target object and all circular shift image $\mathbf{x}_{m,n}$ forms a circulant matrix $\mathbf{X}$. In general, $\mathbf{X}$ can be expressed as

$$\mathbf{X} = F^H diag(\hat{\mathbf{x}})F, \tag{2}$$

Then, classify all the samples of $\mathbf{X}$ by

$$\mathbf{y} = sgn\left(\mathcal{F}^{-1}(\hat{\mathbf{x}}^* \circ \hat{\mathbf{w}}) + b\mathbf{e}\right), \tag{3}$$

Table 1
Define some generic notations which will be used in our work.

Notation	Explanation
M, N	defined two given positive integers
$\mathbb{R}$	The set of real numbers
$\hat{\mathbf{u}}$	The Fourier coefficients of $\mathbf{u}, \forall\ \mathbf{u} \in \mathbb{R}^{MN}$
$\hat{\mathbf{u}}^*$	The complex conjugate of the Fourier coefficients of $\mathbf{u}$
$\mathcal{F}(\bullet)$	The Fourier transform
$\mathcal{F}^{-1}(\bullet)$	The inverse of $\mathcal{F}$
F	The base vectors of the discrete Fourier transform
F^H	The Hermitian transpose of F
$\circ$	Indicated the element-wise multiplication of any two vectors
$\widehat{\max}\{\bullet, 0\}$	Calculated the maximum value between each elementof any vector and the zero
$\mathbf{e}$	defined an $MN \times 1$ vector, each element of which is 1
$\mathbf{E}$	defined an $MN \times MN$ matrix, each element of which is 1

Given the training sample set $\mathbf{X}$ that consists of all circular shift image $\mathbf{x}_{m,n}$ and its corresponding class label $\mathbf{y} = \left[y_{0,0}, \ldots, y_{m,n}, \ldots, y_{M-1,N-1}\right]^T$, they use the squared hinge loss to define the SVM model [39] as follows:

$$\begin{aligned} &\min_{\mathbf{w},b} \|\mathbf{w}\|^2 + C\|\xi\|^2 \\ &s.t.\ \ \mathbf{y} \circ \left(\mathbf{X}\mathbf{w}^T + b\mathbf{e}\right) \geqslant \mathbf{e} - \xi, \end{aligned} \tag{4}$$

where $\xi = \left[\xi_{0,0}, \ldots, \xi_{m,n}, \ldots, \xi_{M-1,N-1}\right]^T$ is the vector of slack variables, C is a trade-off parameter.

Based on the circulant property of $\mathbf{X}$, the SVM model can be equivalently formulated as:

$$\begin{aligned} &\min_{\mathbf{w},b} \|\mathbf{w}\|^2 + C\|\xi\|^2 \\ &s.t.\ \ \mathbf{y} \circ \left(\mathcal{F}^{-1}(\hat{\mathbf{x}}^* \circ \hat{\mathbf{w}}) + b\mathbf{e}\right) \geqslant \mathbf{e} - \xi, \end{aligned} \tag{5}$$

In the SVM discriminative model, Zuo et al. [12] assign binary class label by the confidence map of object position [40], where the confidence map is defined as:

$$s(\mathbf{p}_{m,n}, \mathbf{p}^\star) = \Gamma exp\left(-\eta \|\mathbf{p}_{m,n} - \mathbf{p}^\star\|^\lambda\right), \tag{6}$$

where $\mathbf{p}^\star$ denotes the centre position of the interested object $\mathbf{x}^\star, \mathbf{p}_{m,n}$ represents the centre position of the circular shift image $\mathbf{x}_{m,n}, \Gamma$ is a normalization constant, η and λ are the scale and shape parameters respectively. With the confidence map, the class label $\mathbf{y}$ can be obtained by

$$y_{m,n} = \begin{cases} 1 & if\ s(\mathbf{p}_{m,n}, \mathbf{p}^\star) \geqslant \theta_u \\ -1 & if\ s(\mathbf{p}_{m,n}, \mathbf{p}^\star) \leqslant \theta_l \\ 0 & otherwise \end{cases}, \tag{7}$$

where θ_l and θ_u are lower and upper thresholds respectively.

In order to exploit the property of the circulant matrix to learn the model (5), let $\xi = \mathbf{v} + \mathbf{e} - \mathbf{y} \circ \left(\mathcal{F}^{-1}(\hat{\mathbf{w}} \circ \hat{\mathbf{x}}^*) + b\mathbf{e}\right)$, and then it can be rewritten as:

$$\begin{aligned} &\min_{\mathbf{w},b,\mathbf{v}} \|\mathbf{w}\|^2 + C\|\mathbf{y} \circ \left(\mathcal{F}^{-1}(\hat{\mathbf{x}}^* \circ \hat{\mathbf{w}}) + b\mathbf{e}\right) - \mathbf{e} - \mathbf{v}\|^2 \\ &\quad s.t.\ \ \mathbf{v} \succeq 0, \end{aligned} \tag{8}$$

where $\mathbf{v}$ is an auxiliary variable and $\succeq$ denotes that each element of $\mathbf{v}$ is greater than or equal to zero.

3.2. Support correlation filter in nonlinear space

To make the support correlation filter (SCF) model to be extended to learn the nonlinear decision function, we now derive a "dual version" for the SCF model. In this derivation we partially follow Vapnik [41]. We start with re-expressing the SVM model in (4) as:

$$\begin{aligned} &\min_{\mathbf{w},b,\mathbf{v},\boldsymbol{\alpha}} \|\mathbf{w}\|^2 + \boldsymbol{\alpha}^T\left(\mathbf{e} + \mathbf{v} - \mathbf{y} \circ \left(\mathbf{X}\mathbf{w}^T + b\mathbf{e}\right) - \xi\right) + C\|\xi\|^2 \\ &s.t.\ \ \mathbf{v} \succeq 0, \end{aligned} \tag{9}$$

Here $\boldsymbol{\alpha}$ is the Lagrange multiplier (it also represents the solution of SCF in the dual space). We let $\mathbf{q} = \mathbf{y} + \mathbf{y} \circ \mathbf{v}$, where $\mathbf{v} \succeq 0$, and then the model (9) can be rewritten as:

$$\min_{\mathbf{w},b,\mathbf{q},\boldsymbol{\alpha},\xi} \|\mathbf{w}\|^2 + \boldsymbol{\alpha}^T\left(\mathbf{q} - \left(\mathbf{X}\mathbf{w}^T + b\mathbf{e}\right) - \mathbf{y} \circ \xi\right) + C\|\xi\|^2. \tag{10}$$

Solving the model (10) with respect to $\mathbf{w}$, we can obtain $\mathbf{w} = \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{X}$. Then Substituting this into (10), we obtain

$$\min_{\xi,b,\mathbf{q},\boldsymbol{\alpha}} -\frac{1}{4}\boldsymbol{\alpha}^T\mathbf{X}\mathbf{X}^T\boldsymbol{\alpha} + \boldsymbol{\alpha}^T(\mathbf{q} - b\mathbf{e}) - \boldsymbol{\alpha}^T(\mathbf{y} \circ \xi) + C\|\xi\|^2. \tag{11}$$

Calculating (11) with respect to ξ, we obtain $\xi = \frac{1}{2C}\mathbf{y}^T\boldsymbol{\alpha}$. Then Substituting this into (11), we get

$$\min_{b,\mathbf{q},\boldsymbol{\alpha}} -\frac{1}{4}\boldsymbol{\alpha}^T\mathbf{X}\mathbf{X}^T\boldsymbol{\alpha} + \boldsymbol{\alpha}^T(\mathbf{q} - b\mathbf{e}) - \frac{1}{4C}\boldsymbol{\alpha}^T\boldsymbol{\alpha}, \tag{12}$$

Thus, the closed form solution to our sub-problem on $\boldsymbol{\alpha}$ can be formulated as

$$\boldsymbol{\alpha} = \frac{1}{4}\left(\mathbf{X}\mathbf{X}^T + \frac{1}{C}\mathbf{E}\right)^{-1}(\mathbf{q} - b\mathbf{e}). \tag{13}$$

Given a non-linear mapping function $\varphi(\mathbf{x})$, we define $K(\mathbf{x}, \mathbf{x}') = \langle\varphi(\mathbf{x}), \varphi(\mathbf{x}')\rangle$, which can be used by some kernel function (e.g., Gaussian RBF and polynomial) with permutation invariant. Based on the circulant property of $\mathbf{X}, \mathbf{X}\mathbf{X}^T$ can be represented as

$$\mathbf{X}\mathbf{X}^T = F^H diag(\hat{\mathbf{x}} \circ \hat{\mathbf{x}}^*)F, \tag{14}$$

Then, introducing non-linear feature mapping $\varphi(\mathbf{x})$ into the formula (14), it can be revised as

$$F^H diag(\varphi(\hat{\mathbf{x}}) \circ \varphi(\hat{\mathbf{x}}^*))F = F^H\hat{\mathbf{k}}^{\mathbf{xx}}F = \mathbf{K}, \tag{15}$$

where $\hat{\mathbf{k}}^{\mathbf{xx}}$ is the Fourier transform of $K(\mathbf{x}, \mathbf{x})$ and $\mathbf{K}$ is a circulant kernel matrix.

Thus, the solution to the sub-problem of the kernelized support correlation filter on $\boldsymbol{\alpha}$ can be formulated as

$$\boldsymbol{\alpha} = \frac{1}{4}\left(\mathbf{K} + \frac{1}{C}\mathbf{E}\right)^{-1}(\mathbf{q} - b\mathbf{e}). \tag{16}$$

3.3. Formulation of structural support correlation filter

The support correlation filter model mentioned above is only to learn a holistic appearance model, which is not robust for partial occlusion. In order to tackle this issue, we introduce part-based tracking strategy to the support correlation filter model. Given a target object, it is divided into L non-overlapping parts with $M \times N$ pixels. Then, we can learn the dual optimization variable $\boldsymbol{\alpha}_l$ of support correlation filter $\mathbf{w}_l$ of each part via (17)

$$\min_{b,\mathbf{q}_l,\boldsymbol{\alpha}_l} \sum_{l=1}^{L} -\frac{1}{4}\boldsymbol{\alpha}_l^T \mathbf{X}_l \mathbf{X}_l^T \boldsymbol{\alpha}_l + \boldsymbol{\alpha}_l^T(\mathbf{q}_l - b_l\mathbf{e}) - \frac{1}{4C}\boldsymbol{\alpha}_l^T\boldsymbol{\alpha}_l, \tag{17}$$

Here $\mathbf{q}_l = \mathbf{y} + \mathbf{y} \circ \mathbf{v}_l$, where $\mathbf{v}_l$ is an auxiliary variable corresponding to the lth part. The b_l corresponds to the bias of the lth part in the model and the $\mathbf{X}_l$ consists of all circular shift image $\mathbf{x}_{m,n}$ of the lth part, where $l = 1, \ldots, L$.

Intuitively, the motion model of each local part should be close to each other to cover the entire target. Therefore, they should select the similar circle shifts to make them have similar motion [16]. In order to characterize the similar motion among local parts and tolerate slight discrepancy among them, we introduce a customized Laplacian regularization term in the model (17), that is

$$\min_{b,\mathbf{q}_l,\boldsymbol{\alpha}_l} \sum_{l=1}^{L} -\tfrac{1}{4}\boldsymbol{\alpha}_l^T \mathbf{X}_l \mathbf{X}_l^T \boldsymbol{\alpha}_l + \boldsymbol{\alpha}_l^T(\mathbf{q}_l - b_l\mathbf{e}) - \tfrac{1}{4C}\boldsymbol{\alpha}_l^T\boldsymbol{\alpha}_l + \tfrac{\delta}{2}\sum_{i,j} \|\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_j\|^2 \omega_{i,j} \quad \forall\ i,j \in L \ \ and \ \ i \neq j, \tag{18}$$

where δ is the weight parameter of the Laplacian regularization term. $\omega_{i,j}$ denotes a penalty parameter whether two parts i and j have similar motion. If $\omega_{i,j}$ is larger, the motion of two parts is more consistent, vice versa.

However, this fully connected structure makes it intractable to solve the model in (18) because we only alternately solve one of $\alpha_i, i = 1, \ldots, L$ at a time when solving the model (18) and don't simultaneously solve all the $\alpha_i, i = 1, \ldots, L$ each time. Thus, under the situation of hardly lowering the performance of the model in (18), we simplify the connected structure of the model in (18) by a star model. In the star model, each local part is connected by an edge with a dummy part $\mathbf{x}_r$ which can be represented by the mean image of all the local parts, *i.e.* there are no direct relation between any two parts. Thus, this requires a minor adaptation of the model in (18), that is

$$\min_{b,\mathbf{q}_l,\boldsymbol{\alpha}_l} \sum_{l=1}^{L} -\frac{1}{4}\boldsymbol{\alpha}_l^T \mathbf{X}_l \mathbf{X}_l^T \boldsymbol{\alpha}_l + \boldsymbol{\alpha}_l^T(\mathbf{q}_l - b_l\mathbf{e}) - \frac{1}{4C}\boldsymbol{\alpha}_l^T\boldsymbol{\alpha}_l + \frac{\delta}{2}\sum_{l=1}^{L} \|\boldsymbol{\alpha}_l - \boldsymbol{\alpha}_r\|^2 \omega_{l,r}, \tag{19}$$

Here $\boldsymbol{\alpha}_r$ denotes dual optimization variable of support correlation filter $\mathbf{w}_r$ of the dummy part $\mathbf{x}_r$. Because the target moves smoothly between consecutive two frames, we can use the motion consistency among parts in $(t-1)$th frame to describe the motion relation among parts at the current frame. So, we define $\omega_{l,r}$ to decrease exponentially with the hyper-distance of support correlation filters $\mathbf{w}_l$ and $\mathbf{w}_r$ of the lth part $\mathbf{x}_l$ and the dummy part $\mathbf{x}_r$ in $(t-1)$th frame, i.e.,

$$\omega_{l,r} = \exp\left(-\frac{1}{2}\frac{\|\mathbf{w}_l^{t-1} - \mathbf{w}_r^{t-1}\|^2}{\kappa^2}\right), \tag{20}$$

where κ is a smooth factor.

In practice, according to the observation, we found that the appearance of tracked object changes smoothly over time. Thus the selected training samples should be similar in consecutive frames. That is to say, the corresponding $\boldsymbol{\alpha}_l^{t-1}$ of each local part in $(t-1)$th frame should be close to that in tth frame, which is called temporal consistency. Therefore, we may introduce temporal constrain term into the model (19) and revise it as follows

$$\min_{b,\mathbf{q}_l,\boldsymbol{\alpha}_l} \sum_{l=1}^{L}\left(-\frac{1}{4}\boldsymbol{\alpha}_l^T \mathbf{X}_l \mathbf{X}_l^T \boldsymbol{\alpha}_l + \boldsymbol{\alpha}_l^T(\mathbf{q}_l - b_l\mathbf{e}) - \frac{1}{4C}\boldsymbol{\alpha}_l^T\boldsymbol{\alpha}_l\right) + \frac{\delta}{2}\sum_{l=1}^{L}\|\boldsymbol{\alpha}_l - \boldsymbol{\alpha}_r\|^2\omega_{l,r} + \frac{\beta}{2}\sum_{l=1}^{L}\|\boldsymbol{\alpha}_l^t - \boldsymbol{\alpha}_l^{t-1}\|^2, \tag{21}$$

where β is the controlling factor of the temporal constrain term.

Given non-linear mapping function $\varphi(\mathbf{x})$ and the derivation of formulas (14) and (15), our model in (21) can be extended to learn a kernelized structured support correlation filter model, i.e.

$$\min_{b_l,\mathbf{q}_l,\boldsymbol{\alpha}_l} \sum_{l=1}^{L}\left(-\frac{1}{4}\boldsymbol{\alpha}_l^T \mathbf{K}_l \boldsymbol{\alpha}_l + \boldsymbol{\alpha}_l^T(\mathbf{q}_l - b_l\mathbf{e}) - \frac{1}{4C}\boldsymbol{\alpha}_l^T\boldsymbol{\alpha}_l\right) + \frac{\delta}{2}\sum_{l=1}^{L}\|\boldsymbol{\alpha}_l - \boldsymbol{\alpha}_r\|^2\omega_{l,r} + \frac{\beta}{2}\sum_{l=1}^{L}\|\boldsymbol{\alpha}_l^t - \boldsymbol{\alpha}_l^{t-1}\|^2. \tag{22}$$

where $\mathbf{K}_l$ is a circulant kernel matrix corresponding to the lth part.

According to the above points, our models in (21) and (22) can learn the support correlation filter parameters of all local parts jointly and distinguish the parts from the background. Our model is also resistant to partial occlusion. Besides, it has high efficiency and robustness.

3.4. Optimization

In this subsection, we utilize the Alternating Direction Method of Multipliers (ADMM) method [42] to solve the optimization problem in (21). When keeping other variables fixed, the ADMM method can iteratively update one of the variables $b_l, \mathbf{q}_l, \boldsymbol{\alpha}_l, \boldsymbol{\alpha}_r$ by minimizing (21), which can guarantee the convergence of our proposed model. Consequently, updating steps corresponding to all the variables are as follows:

Step 1: update$\boldsymbol{\alpha}_r$ (with others fixed): $\boldsymbol{\alpha}_r$ can be updated by solving the following optimization problem

$$\boldsymbol{\alpha}_r = \arg\min_{\boldsymbol{\alpha}_r} \frac{\delta}{2}\sum_{l=1}^{L}\|\boldsymbol{\alpha}_l - \boldsymbol{\alpha}_r\|^2\omega_{l,r}, \tag{23}$$

and its solution is

$$\boldsymbol{\alpha}_r = \frac{1}{\sum_{l=1}^{L}\omega_{l,r}}\sum_{l=1}^{L}\omega_{l,r}\boldsymbol{\alpha}_l. \tag{24}$$

Step 2: update$\mathbf{q}_l$ (with others fixed): before computing $\mathbf{q}_l$, we firstly need to calculate the variable $\mathbf{v}_l$. Combining the models (8) and (9), the subproblem on $\mathbf{v}_l$ becomes

$$\mathbf{v}_l = \arg\min_{\mathbf{v}_l} \|\mathbf{v}_l - (\mathbf{y} \circ (\mathbf{X}_l\mathbf{w}_l^T + b_l\mathbf{e}) - 1)\|^2. \quad s.t. \ \ \mathbf{v}_l \succeq 0 \tag{25}$$

Then, $\mathbf{v}_l$ has the following closed form solution:

$$\mathbf{v}_l = \widetilde{\max}\{\mathbf{y} \circ (\mathbf{X}_l\mathbf{w}_l^T + b_l\mathbf{e}) - \mathbf{e}, 0\}. \tag{26}$$

In view of $\mathbf{w}_l = \frac{1}{2}\boldsymbol{\alpha}_l^T\mathbf{X}$, the formula (26) can be modified as

$$\mathbf{v}_l = \widetilde{\max}\left\{\mathbf{y} \circ \left(\frac{1}{2}\mathbf{X}_l\mathbf{X}_l^T\boldsymbol{\alpha}_l + b_l\mathbf{e}\right) - \mathbf{e}, 0\right\}, \tag{27}$$

When $\mathbf{x}_l$ is mapped to the kernel feature space, the amended version of the formula (27) is as follows

$$\mathbf{v}_l = \widetilde{\max}\left\{\mathbf{y} \circ \left(\frac{1}{2}\mathbf{K}_l\boldsymbol{\alpha}_l + b_l\mathbf{e}\right) - \mathbf{e}, 0\right\}, \tag{28}$$

Known $\mathbf{v}_l$ from the aforementioned formulas (27) or (28), we can calculate $\mathbf{q}_l$ by

$$\mathbf{q}_l = \mathbf{y} + \mathbf{y} \circ \mathbf{v}_l. \tag{29}$$

Step 3: updateb_l (with others fixed): we exploit the method of solving the parameter b in [12] to calculate the b_l, *i.e.*

$$b_l = \bar{q}_l. \tag{30}$$

where $\bar{q}_l$ is the mean of $\mathbf{q}_l$.

Step 4: update$\boldsymbol{\alpha}_l$ (with others fixed): The minimization problem (21) with respect to $\{\boldsymbol{\alpha}_l\}_{l=1}^{L}$ can be decomposed into L mutually independent subproblems. The lth subproblem to update $\boldsymbol{\alpha}_l$ can be equivalently re-expressed as

$$\begin{aligned}\boldsymbol{\alpha}_l = \arg\min_{\boldsymbol{\alpha}_l} & -\tfrac{1}{4}\boldsymbol{\alpha}_l^T\mathbf{X}_l\mathbf{X}_l^T\boldsymbol{\alpha}_l + \boldsymbol{\alpha}_l^T(\mathbf{q}_l - b_l\mathbf{e}) \\ & -\tfrac{1}{4C}\boldsymbol{\alpha}_l^T\boldsymbol{\alpha}_l + \tfrac{\delta}{2}\|\boldsymbol{\alpha}_l - \boldsymbol{\alpha}_r\|^2\omega_{l,r} \\ & +\tfrac{\beta}{2}\|\boldsymbol{\alpha}_l^t - \boldsymbol{\alpha}_l^{t-1}\|^2,\end{aligned} \tag{31}$$

Then, for each $\boldsymbol{\alpha}_l$, the closed form solution of the formula (31) is shown as follows

$$\boldsymbol{\alpha}_l = \left(\tfrac{1}{2}\mathbf{X}_l\mathbf{X}_l^T + \tfrac{1}{2C}\mathbf{E} - \delta\omega_{l,r}\mathbf{E} - \beta\mathbf{E}\right)^{-1}\left((\mathbf{q}_l - b_l\mathbf{e}) - \delta\omega_{l,r}\boldsymbol{\alpha}_r - \beta\boldsymbol{\alpha}_l^{t-1}\right). \tag{32}$$

The detailed ADMM algorithm that solves our model (21) is given in Algorithm 1, where the convergence is reached when the change of solution $\boldsymbol{\alpha}_l$ is below a pre-defined threshold (e.g. $\tau = 10^{-3}$ in our work) or the number of iteration is greater than the maximum iterations *Iter*.

Algorithm 1. Solving the optimization problem defined by the model (21)

Input: Training data: $\mathbf{X}_l$ and $\mathbf{y}$. Initialization of parameters δ, β, C

Output: $\{\boldsymbol{\alpha}_l, b_l\}_{l=1}^{L}$

1: Initialize $num \leftarrow 1, \boldsymbol{\alpha}_l^{t(1)} = \frac{\mathbf{X}_l^T\mathbf{y}}{\mathbf{X}_l\mathbf{X}_l^T + \frac{1}{C}\mathbf{E}}, \boldsymbol{\alpha}_l^{t(0)} = \mathbf{0}, b_l^t = \bar{y}$, where $\bar{y}$ is the mean of $\mathbf{y}$.
2: **while** $num \leqslant Iter$ or $|\boldsymbol{\alpha}_l^{t(i)} - \boldsymbol{\alpha}_l^{t(i-1)}| > \tau$ **do**
3: Update $\boldsymbol{\alpha}_r$ via (24)
4: **for** $l = 1$ to L **do**
5: Update $\mathbf{q}_l$ via (27) and (29)
6: Update b_l via (30)
7: calculate $\boldsymbol{\alpha}_l^t$ via (32)
8: **end for**
9: $num \leftarrow num + 1$
10: **end while**

As shown in Algorithm 1, its major computing cost is that we need to calculate the matrix inverse and multiplication in spatial domain when updating $\boldsymbol{\alpha}_l^t$ via (32). However, in view of the circulant structure property of $\mathbf{X}_l, \boldsymbol{\alpha}_l^t$ can be calculated very efficiently in the Fourier domain. Thus, the formula (32) can be rewritten as the version (33) in the Fourier domain.

$$\hat{\boldsymbol{\alpha}}_l^t = \frac{\hat{\mathbf{q}}_l - b_l\hat{\mathbf{e}} - \delta\omega_{l,r}\hat{\boldsymbol{\alpha}}_r - \beta\hat{\boldsymbol{\alpha}}_l^{t-1}}{\frac{1}{2}\hat{\mathbf{x}}_l \circ \hat{\mathbf{x}}_l^* + \frac{1}{2C}\hat{\mathbf{e}} - \delta\omega_{l,r}\hat{\mathbf{e}} - \beta\hat{\mathbf{e}}}, \tag{33}$$

where $\frac{\bullet}{\bullet}$ denotes the element-wise division.

When the sample $\mathbf{x}_l$ is mapped to the kernel feature space, the updating step with respect to $\boldsymbol{\alpha}_l$ needs a minor modification, that is

$$\boldsymbol{\alpha}_l = \left(\tfrac{1}{2}\mathbf{K}_l + \tfrac{1}{2C}\mathbf{E} - \delta\omega_{l,r}\mathbf{E} - \beta\mathbf{E}\right)^{-1}\left((\mathbf{q}_l - b_l\mathbf{e}) - \delta\omega_{l,r}\boldsymbol{\alpha}_r - \beta\boldsymbol{\alpha}_l^{t-1}\right), \tag{34}$$

Meanwhile, the corresponding version of the formula (34) in the Fourier domain is as follows

$$\hat{\boldsymbol{\alpha}}_l^t = \frac{\hat{\mathbf{q}}_l - b_l\hat{\mathbf{e}} - \delta\omega_{l,r}\hat{\boldsymbol{\alpha}}_r - \beta\hat{\boldsymbol{\alpha}}_l^{t-1}}{\frac{1}{2}\hat{\mathbf{k}}^{\mathbf{xx}} + \frac{1}{2C}\hat{\mathbf{e}} - \delta\omega_{l,r}\hat{\mathbf{e}} - \beta\hat{\mathbf{e}}}. \tag{35}$$

where $\frac{\bullet}{\bullet}$ denotes the element-wise division.

To solve the optimization problem defined by (22), we only need to use the formulas (28) and (34) to replace the formulas (27) and (32) in the updating steps.

Finally, known $\hat{\boldsymbol{\alpha}}_l$, the $\boldsymbol{\alpha}_l$ can be obtained by $\boldsymbol{\alpha}_l = \mathcal{F}^{-1}(\hat{\boldsymbol{\alpha}}_l)$. Moreover, to speed up the Algorithm 1, it can be implemented in matrix form without the "for" loop.

3.5. Tracking

At the tracking stage of nonlinear feature space, when obtaining the coefficient vector $\hat{\boldsymbol{\alpha}}_l^{t-1}$ and bias b_l of each local part in the previous frame, we can estimate the response map of each local patch $\mathbf{z}_l$ at the current frame by the following formula

$$\mathbf{f}_l^t = \mathcal{F}^{-1}\left(\hat{\mathbf{k}}_l^{\mathbf{xz}} \circ \hat{\boldsymbol{\alpha}}_l^{t-1}\right) + b_l\mathbf{e}, \tag{36}$$

where $\hat{\mathbf{k}}_l^{\mathbf{xz}}$ denotes the Fourier transform of $K(\mathbf{x}, \mathbf{z})$ for the lth part. The position $\mathbf{p}_l^t$ of the lth part can be determined by the maximum value of $\mathbf{f}_l^t$, e.g. the position $\mathbf{p}_l^t$ of each part can be expressed as

$$\mathbf{p}_l^t = \mathbf{p}_l^{t-1} + \boldsymbol{\Delta}_l^t, \tag{37}$$

Here $\boldsymbol{\Delta}_l^t$ denotes the translation of the lth part at time t.

Then we can estimate the final position $\mathbf{p}_g^t$ of object by the translation $\boldsymbol{\Delta}_l^t$ of all the parts, that is

$$\mathbf{p}_g^t = \mathbf{p}_g^{t-1} + \sum_{l=1}^{L}\pi_l\boldsymbol{\Delta}_l^t, \tag{38}$$

where π_l is the weight parameter of corresponding part.

Because different parts of the target may suffer different appearance changes, illumination variation or occlusion in different frames, intuitively, if we assign the same weight to each part, the falsely tracker part may be overemphasized which will lead to drift problem. In order to handle this issue, we should adaptively give each part a different weight according to its reliability. In our work, we exploit the peak-to-sidelobe ratio (PSR) to define the weight of each part because the higher PSR usually means more reliable part, where the PSR is defined as

$$\phi_l = \frac{\max(\mathbf{f}_l) - \mu_l}{\sigma_l}, \tag{39}$$

where μ_l and σ_l are the mean and the standard deviation of $\mathbf{f}_l$ respectively.

In addition, for tracking problem, the appearance similarity between two consecutive frames is helpful for distinguishing whether the part is reliable or not. Taking this observation into consideration, we define the appearance similarity d to determine whether the part is reliable, i.e.

$$d_l = exp\left(-\frac{\|\mathbf{x}_l^t - \mathbf{x}_l^{t-1}\|^2}{\gamma^2}\right), \tag{40}$$

where γ is a hyperparameter, and $\mathbf{x}$ is the vector representation of object appearance features, where the appearance features use the color histogram features. Combining two indicators above, we

(a) (b)

Fig. 1. Visualization of the PSR and appearance similarity of the part denoted by the yellow bounding box in the frame #83 and #141. (a) The target is occluded in the frame #83. The $PSR = 5.01191$ and the appearance similarity $d = 0.11002$ of the part denoted by the yellow bounding box. (b) The target occurred the deformation in the frame #141. The $PSR = 5.1559$ and the appearance similarity $d = 0.27286$ of the part denoted by the yellow bounding box. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

distinguish whether the lth part is occluded or has a large pose change. If ϕ_l and d_l are less than the pre-defined threshold, this part is unreliable. As shown in Fig. 1(a), when the part that denoted by the yellow bounding box is occluded, its PSR and appearance similarity both become smaller. According to our two criteria, this part is unreliable. But in the Fig. 1(b), the target occurs the deformation. If using our two criteria, this part is reliable and if only using the PSR, then this part is unreliable. In fact, for the deformed part, we need to update its model to avoid drift.

To avoid erroneous estimations further, we use the PSR value and the appearance similarity d_l from the reliable parts to calculate the corresponding weight π_j, i.e.

$$\pi_j = (1-\varpi)\frac{\phi_j}{\sum_{j=1}^{J}\phi_j} + \varpi\frac{d_j}{\Sigma_{j=1}^{J}d_j}, \tag{41}$$

where J denotes the number of all reliable parts, ϖ is a fusion parameter. In our work, ϖ is set as 0.4. By now, the formula (38) can be modified as

$$\mathbf{p}_g^t = \mathbf{p}_g^{t-1} + \sum_{j=1}^{J}\pi_j\Delta_j^t, \tag{42}$$

Here if $J = 0$, this means that all parts are unreliable. At this time, we use the translation of target in the previous frame to approximate that of target in the current frame because the motion of target hardly keep steady between two consecutive frames in most cases.

Update scheme: During tracking, the object appearance may change because of a number of challenging factors such as illumination change and pose change. Hence it is necessary to update part classifiers over time. Our tracking model is made up of the learned target appearance $\mathbf{x}_l$ and the transformed classifier coefficients $\boldsymbol{\alpha}_l$. For each patch, our model parameters are updated by

$$\begin{aligned}\mathbf{x}_l^t &= (1-\rho_l)\mathbf{x}_l^{t-1} + \rho_l\mathbf{x}_l \\ \boldsymbol{\alpha}_l^t &= (1-\rho_l)\boldsymbol{\alpha}_l^{t-1} + \rho_l\boldsymbol{\alpha}_l,\end{aligned} \tag{43}$$

where ρ_l is a learning rate parameter. The $\boldsymbol{\alpha}_l$ is calculated by simple linear interpolation. The $\mathbf{x}_l$ is updated by taking the current appearance into account.

However, if using a fixed learning rate ρ in the updating process, the whole model will be contaminated in the remaining frames once the tracker loses the object. Thus, to avoid producing errors, It is apparent that the model of the occluded part should not be updated and other parts should adaptively adjust their learning rate based on the corresponding reliable weight. Therefore the learning rate of each part is updated by the following scheme

$$\rho_l = \begin{cases}\pi_l\varrho & if\ \ \phi_l > \epsilon\ \ or\ \ d_l > \varepsilon \\ 0 & otherwise\end{cases}, \tag{44}$$

where ϱ is a fixed learning rate, ϵ and ε are two predefined thresholds.

Thus, contrary to traditional correlation filter based trackers, due to exploiting the adaptive updating strategy, our method can still maintain the tracking accuracy by using the results of the previous frame even when all part are occluded at one frame.

Scaling: To adapt the scale change of object, most of correlation filter based trackers [43,29,30] utilize a discriminative filter or a search pool that is based on pyramidal structure to estimate the object size. Despite obtaining outstanding results, these methods do not accurately estimate the current object size. So, to tackle this issue, we adopt the ration of the relative distance among local parts as in [35] to estimate the object size accurately because it's positively correlated with the scale of the target. In addition, to improve the accuracy of estimating object size further, in our work, we only use the change rate of relative distance among reliable local parts to estimate the object size. Therefore, the object scale S^t is calculated by

$$S^t = \frac{S^{t-1}}{J(J-1)}\sum_{i=1}^{J}\sum_{j=1}^{J}\frac{\|\mathbf{p}_i^t - \mathbf{p}_j^t\|^2}{\|\mathbf{p}_i^{t-1} - \mathbf{p}_j^{t-1}\|^2}\quad (i \neq j). \tag{45}$$

where $\mathbf{p}_i^t$ represents the position of part i in the tth frame. Because at least two reliable parts can make the formula (45) feasible, we keep the scale size of the preview frame unchange when only one part is available. In addition, the scale of the target does not change dramatically between two consecutive frames. To estimate the scale of target more robustly, we utilize the moving average to calculate the scale of target at the current frame.

So far, the theoretical part of the algorithm has been completely introduced above. For better comprehending our proposed method, it is summarized in Algorithm 2.

Algorithm 2. Scale structural support kernel correlation filter tracking algorithm (ScaleSSKCF)

Input: Image frames $\{I_t\}_1^T$, initial object position $\mathbf{p}_g^1$

Output: Target position of each frame $\left\{\mathbf{p}_g^t\right\}_2^T$

1: **repeat**
2: Calculate the position $\mathbf{p}_l^{t-1}$ of each part based on the last target position $\mathbf{p}_g^{t-1}$.
3: Crop an image patch $\mathbf{x}_l$ from I_t at the patch position $\mathbf{p}_l^{t-1}$ of the $t-1$ time and extract the corresponding feature representation.
4: Calculate the filter coefficient $\boldsymbol{\alpha}_l$ and bias b_l of each patch by the Algorithm 1.
5: Detection the position $\mathbf{p}_l^t$ of each patch via (36).
6: Distinguish whether the lth part is reliable by ϕ_l and d_l.
7: Compute the target position $\mathbf{p}_g^t$ at the current time via (42).
8: Estimate the scale of the target via (45).
9: Update learned target appearance $\mathbf{x}_l$ and the transformed classifier coefficients $\boldsymbol{\alpha}_l$ with the formulas (43) and (44).
10: **until** end of video sequence.

4. Experiments

In the experimental part, we use the several benchmark datasets: TempleColor128,[1] OTB2015[2] and VOT2015[3] and their related evaluation protocols [44–46] to evaluate the proposed ScaleSSKCF algorithms. First, we introduce the experimental setup. Next, we evaluate two variants of our proposed method, i.e., OWSC (our algorithm without structural constraint) and OWTC (our algorithm without temporal consistent), to analyze the effect of structural constrain term and temporal constrain term in our proposed method. Finally, our proposed algorithm is compared with some the most related state-of-the-art methods.

4.1. Experimental setup

Our proposed approach is implemented in native MATLAB 2014a on a 3.6GHZ Intel i7 Core4 PC with 4G RAM. The average running speed is around 40 frames per second. The optimization takes 5 iterations in the first frame and 2 or 3 iterations for each online update. In our method, the feature extraction takes up 48% of the total consuming time. But the optimization is only 3%.

Parameters: Our tracker involves a few model parameters, *i.e.*, trade-off parameter C, scale parameter η and shape parameter λ of confidence maps, the weight parameter δ of the Laplacian regulation term, the controlling factor β of the temporal constrain term, and lower and upper thresholds (θ_l, θ_u) in (7). In addition, other parameters include the smooth factor κ in (20) and hyperparameter in (40). For online tracking, the model is updated by linear interpolation with the adaption rate ϱ in (44). In our experiments, the detailed parameters setting is shown in Table 2, where padding means the magnification of the image region samples relative to the target bounding box. The number of local parts L is adaptively determined by the aspect ratio of object $\frac{O_N}{O_M}$, where O_N and O_M separately denote the width and height of object. If $0.6 < \frac{O_N}{O_M} < 1.6$, the target is divided into 2×2 local parts, *i.e.*, $L = 4$; if $\frac{O_N}{O_M} \leqslant 0.6$, the target is partitioned into 3×1 local parts, *i.e.*, $L = 3$; if $\frac{O_N}{O_M} \geqslant 1.6$, we sample 1×3 local parts on the target. The detailed partitioning method is shown in Fig. 2. Note that, any other part sampling methods can also be adopted.

Datasets and evaluation metrics: To assess the performance of the proposed tracker, extensive experiments are carried on several public benchmark datasets such as TempleColor128 [46], OTB-2015 [47] and VOT2015 [45]. In the TempleColor128 and OTB-2015 datasets, we adopt two metrics used in [46] including distance precision (DP) and overlap precision (OP). The DP is the relative number of frames in the sequence where the center location error is smaller than a certain threshold. As in [44], the DP values at a threshold of 20 pixels are reported in our work. The OP is defined as the percentage of frames where the bounding box overlap surpasses a certain threshold. We report the results at a threshold of 0.5, which correspond to the PASCAL evaluation criterion [48]. Except for the DP and OP metrics, the precision and success plots [44] have also been adopted to measure the overall tracking performance. For the precision and success plots, we respectively use the DP value of each tracker and the area under curve (AUC) score of each success plot to rank the tracking algorithms. In VOT2015 sequences, we utilize evaluation criterion proposed in [45].

4.2. Key component validation

Here, on the TempleColor128 dataset [46], we discuss the impact of structural constraint term and temporal consistent term in our algorithm. Based on the algorithm analysis in Section 3, the performance of our algorithm should decrease to some extent without structural constraint term and temporal consistent term, which is shown in Table 3. The OWSC and OWTC respectively denote the absence of structural constraint term and temporal consistent term in our model. Overall, the performance of the proposed algorithm is best among these three methods (*e.g.* OWSC, OWTC and ScaleSSKCF (ours)). Seen from the comparison, the performance of OWSC is worst, which means that the structural constraint term of our tracking model plays the most important role in the performance of our algorithm.

4.3. Evaluation on OTB2015 dataset

Here, we provide a comparison of our method with 7 state-of-the-art and the most related methods from the literature: SRDCF [33], RPT [14], SKSCF [12], samf [30], Staple [49], lct2 [50] and DPCF [35] on the OTB2015 dataset. But a few most related methods (*e.g.*, SCF [16] and RPAC [15]) are not included in our comparative experiments because their source codes are not open to the public and they didn't do the corresponding experiments on this dataset in their paper.

4.3.1. State-of-the-art comparison

The quantitative comparison among these selected methods is reported in Table 4, using mean overlap precision (OP) and mean distance precision (DP) over all 100 video sequences of OTB2015. Seen from the Table 4, our method achieves the best result by 72% on the mean OP metric. However, the SRDCF obtains the best result on the mean DP. The main reason is that the SRDCF introduces the spatial regulation term to deal with the boundary effect caused by the FFT, which makes it learn a more discriminative model. The performance of our method is almost the same as its but our speed is about 20 times faster (For a more fair comparison of speed, please refer to the results in VOT2015). Although the performance of DPCF is slightly superior to our method on the mean

[1] The sequences together with the ground-truth and matlab evaluation toolkit is available at: http://www.dabi.temple.edu/hbling/data/TColor-128/TColor-128.html.
[2] The sequences together with the ground-truth and matlab code is available at: http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html.
[3] http://www.votchallenge.net/vot2015/dataset.html.

Table 2
Parameters setting of our proposed method (ScaleSSKCF).

Parameters	Padding	η	(θ_l, θ_u)	C	λ	δ	β	κ	ϵ	ε	ϱ	Bins of HOG	Cell size	Orientations
Value	1.8	$0.1 * \sqrt{MN}$	(0.4, 0.9)	10^4	2	0.05	5	3	5.5	0.2	0.015	31	4×4	9

Fig. 2. Visualization of the target's partition based on the target's aspect ratio.

DP, the score of its OP is obviously lower than the one of our method, that's because the DPCF limits the scale changing range of the target between 0.75 and 1.25 and can't estimate it accurately when the target occurs the large-scale change.

Fig. 3 gives precision and success plots over all 100 sequences in OTB2015. The success plot shows the ratios of successful frames at the thresholds varied from 0 to 1. While the precision plot describes the ratios of frames in which the center location error (CLE) is smaller than a arbitrary threshold ranging from 0 to 50 pixels. The trackers of each sub-figure in Fig. 3 are ranked by their area under the curve (AUC) scores, displayed in the legend. In the success plots of OPE, our method shows comparable results as the SRDCF and significantly outperforms other several correlation filter trackers. For the precision plots, our method is slightly inferior to the DPCF by 0.2%, that may be because the DPCF combines the tracking results of global correlation filter model, and our method also inferior to the Staple by 0.8%, the main reason of which is that the color histogram model of Staple is more robust to the deformation of target.

4.3.2. Attribute based comparison

The sequences in OTB2015 are annotated with 11 different attributes to describe the different challenges in the tracking problem, including illumination variation (IV), scale variation (SV), occlusion (OCC), deformation (DEF), motion blur (MB), fast motion (FM), in plane rotation (IPR), out-of-plane rotation (OPR), out-of-view (OV), background clutters (BC), and low resolution (LR). These attributes are useful for analyzing the performance of trackers in different aspects. The Tables 5 and 6 respectively shows the performance of ours and 7 state-of-the-art methods in terms of AUC (success metrics) and DP (precision metrics) with respect to each attribute. In Table 5, our method has gained 7 the best and 2 the second best out of 11 subcategories for AUC score. In case of deformation, compared with other methods, our method achieves the second best results (The DP score on the center location error is 73.4% and the AUC score on the overlap rate is 54.6%), which is

Table 3
Comparing the results of OWSC, OWTC and ScaleSSKCF based on mean distance precision (DP) and mean overlap precision (OP). The entries in red denote the best results.

Metrics	OWSC	OWTC	ScaleSSKCF (Ours)
mean OP (%)	45.3	47.1	**47.7**
mean DP (%)	60.9	62.9	**64.1**

Table 4
Comparison with state-of-the-art trackers on the 100 sequences of OTB2015. The top two results are highlighted by bold and different colors: red and blue color.

Metrics	RPT	SKSCF	DPCF	SRDCF	samf	lct2	Staple	ScaleSSKCF (Ours)
mean OP (%)	63.2	67.0	68.9	**71.6**	67.9	63.3	70.5	**72.0**
mean DP (%)	76.0	78.1	77.8	**78.8**	77.0	77.0	**78.5**	77.6
mean FPS (s)	1.8	**23**	20	2	9	6	12	**41**

inferior to the ones of Staple because the color histogram model used in the Staple is more robust to the deformation of target. As are shown in Tables 5 and 6, for the sequences involving the fast motion, the performance of our method and other part-based trackers become bad because the searching area of part-based tracking method shrinks, leading to drift problem. However, the SRDCF can obtain the best results because it can learn a strong model that adapts the fast motion of target on the larger samples. For scale variation, our method achieves the better results than other methods except the SRDCF. Note that the DPCF adopts the scale estimation technique similar to ours but its performance is significantly inferior to ours (*e.g.,* our AUC score on the overlap rate exceeds it by 5%). That is because the DPCF limits the scale changes in a small range (from 0.75 to 1.25), which let it not adapt to the large scale change of the target. For the occlusion factor, our tracker obtains the best AUC score of 58.5% on the overlap rate. The main reason is that our method eliminates the effect of the occlusion when updating the discriminative model. For the low resolution sequences, our method obtains the best results which may be attribute to the temporal consistent term in our model. Fig. 4 shows a qualitative comparison of our approach with 7 existing methods on 11 challenging example videos. Both the SRDCF and ScaleSSKCF perform well in the presence of heavy occlusion (*e.g.*, Human6), which can be attributed to the fact that SRDCF learns a discriminative model on larger image region and our method removes the effects of heavy occlusion when updating our tracking model. The lct2 can effectively re-detect target in the case of tracking failure, e.g., the sequence with the heavy occlusion (Shaking), but it cannot perform well in scale variation and illumination changes (*e.g.*, Car1, Car4 and Car24). When the tracked target of the sequence is occluded by similar color barrier (*e.g.*, Box), the Staple performs very bad because the color histogram model used by it does not distinguish them. Compared with these method, our method can estimate the object size more accurately when occurring the large scale variation (*e.g.*, Car1, Car4, Car24, CarScale and Human5). For the sequences (e.g., Skating1) including the deformation, our approach significantly outperforms other several methods because it adopts some tricks (*e.g.*, structural constraint term and temporal consistency term) to make our discriminative model more robust for target deformation.

4.3.3. Robustness evaluation to initialization

We adopt two robustness metrics: spatial robustness (SRE) and temporal robustness (TRE) provided by [44] to evaluate the robustness to initializations. The SRE criteria initializes the tracker with perturbed boxes, which the TRE criteria starts the tracker at the frame corresponding to each segmentation point (each sequence is divided into the 20 segmentation points). The Fig. 5 shows the TRE and SRE success plots of ours method compared with other related trackers. In the success plots of TRE, the performance of

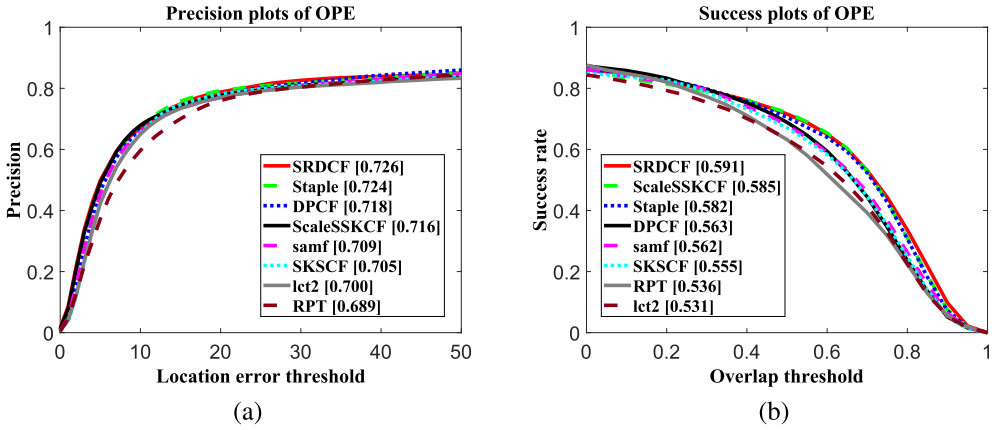

Fig. 3. Precision and success plots over all 100 sequences in OTB2015. The area under the curve (AUC) scores of each tracker are reported in the legends.

Table 5
Success metrics (%) of the trackers for 11 attributes. The top two results are highlighted by red and blue.

Attributes	FM	BC	MB	DEF	IV	IPR	LR	OCC	OPR	OV	SV
RPT	52.0	58.1	51.3	50.2	53.9	52.5	36.2	48.2	50.9	42.2	48.1
SKSCF	52.6	58.0	52.0	51.0	55.5	54.4	35.1	52.1	52.8	40.1	48.8
DPCF	50.3	58.3	53.2	52.9	57.5	52.4	40.0	54.4	55.2	43.5	50.5
SRDCF	58.4	56.6	58.3	53.1	58.7	52.3	49.1	56.2	53.8	43.9	55.6
samf	53.0	55.3	53.3	50.9	54.8	53.1	42.8	55.3	54.2	48.9	50.7
lct2	50.5	54.1	51.2	49.5	51.9	53.0	29.9	48.9	51.2	42.9	43.3
Staple	53.9	57.3	53.9	56.3	59.1	54.3	39.9	55.4	54.1	46.5	52.7
ScaleSSKCF(our)	52.8	63.3	52.0	54.6	60.6	55.0	50.7	58.5	57.0	51.9	55.5

Table 6
Precision metrics (%) of the trackers for 11 attributes. The top two results are highlighted by red and blue.

Attributes	FM	BC	MB	DEF	IV	IPR	LR	OCC	OPR	OV	SV
RPT	68.5	81.4	68.1	72.3	80.1	74.4	59.5	68.0	72.7	54.5	71.2
SKSCF	70.7	82.1	67.9	69.7	76.8	77.7	63.0	72.3	76.0	55.0	72.3
DPCF	67.6	79.8	68.7	73.4	79.2	74.4	66.7	73.2	76.4	54.5	71.7
SRDCF	74.3	74.5	73.5	72.8	76.1	72.1	65.9	74.2	74.4	57.3	75.4
samf	69.9	74.3	67.6	69.0	74.2	74.1	68.4	75.3	75.8	66.3	73.5
lct2	68.0	75.9	66.8	70.6	74.6	78.2	53.7	69.7	75.9	58.2	69.1
Staple	70.9	77.4	69.8	76.8	78.2	76.8	61.0	74.3	74.9	65.8	73.8
ScaleSSKCF(our)	68.5	82.1	68.2	73.4	77.3	75.2	70.7	74.9	76.3	66.0	74.5

our method is second only to that of the SRDCF but is significantly superior to the rest of trackers, especially DPCF and RPT. For the SRE criteria, our method also is slightly inferior to the Staple by only 0.4%. This evaluation demonstrates our method is relatively robust to different spatial and temporal initializations.

4.4. Evaluation on TempleColor 128 dataset

Here, we evaluate our method on the TempleColor128 dataset. The Fig. 6 shows a comparison with 7 state-of-the-art and the most related methods from the literature: Staple [49], SRDCF [33], DPCF [35], lct2 [50], SKSCF [12], RPT [14] and samf [30]. The performance of our method is only ranked the fourth in these methods. The main reason is that TempleColor128 dataset contains about a half sequences with the fast motion [46] and our method is not suitable for dealing with these sequences because the valid searching region becomes smaller when the target is divided into the patches. For the SRDCF, it can learn the filter from the larger searching region because of the spatial regularization term, which makes it against the fast motion. However, its speed is only about 2 frame per second and it cannot meet the realtime applications. The DPCF integrates the results of local and global correlation filter by the minimum spanning tree model. Although its performance is slightly superior to our method, the complex model brings down its speed.

4.5. Evaluation on VOT2015

Finally, we compare our method with other 9 related trackers (CCOT [51], deepSRDCF, Staple [49], SRDCF [33], DPT [38], samf [30], DPCF [35], SKSCF [12] and lct2 [50]) on VOT2015 consisting

Fig. 4. Qualitative comparison of our approach with 7 state-of the-art trackers (denoted in different colors) on the several typically challenging sequences (from left to right and top to down are Box, Car1, Car4, Car24, CarScale, Couple, Skating1, Shaking, Human5, Human6 and Freeman1 respectively). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

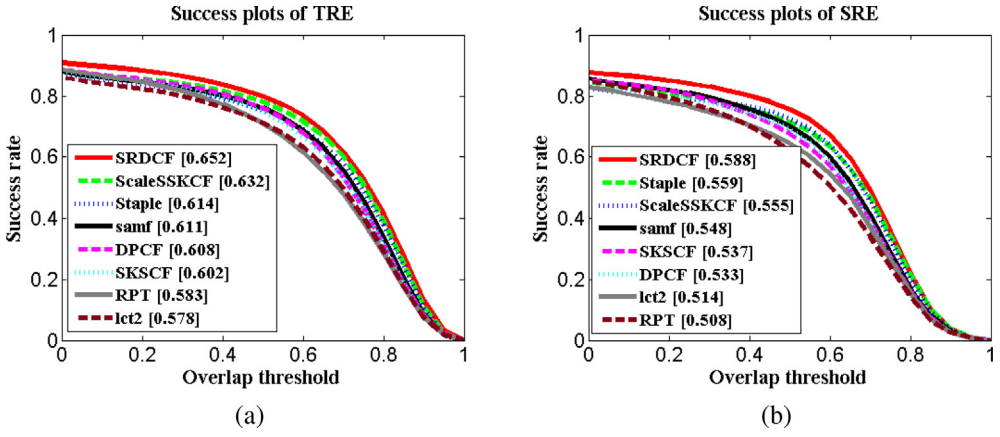

Fig. 5. An evaluation of the temporal and spatial robustness to initializations on the OTB2015 dataset. The area under the curve (AUC) scores of each tracker are reported in the legends.

of 60 challenging videos. Here, we evaluate the performance of the trackers by three metrics (accuracy (overlap with ground truth), robustness (failure rate) and excepted average overlap (EAOP)) provided in [45]. In VOT2015, a tracker is restarted in the case of a failure. In more detail, we refer the readers to [45]. The Table 7 gives their comparison results on VOT2015 according to three metrics mentioned above. Among the compared methods, our method is only ranked the fourth. Note that the results of CCOT and deepSRDCF directly come from the VOT2016 competition. The CCOT and deepSRDCF both use the more discriminative deep convolution features. According to the conclusions in [52], the better features can dramatically improve the tracking performance than the tracker its. Thus, it is not fair to directly compare our method and them.

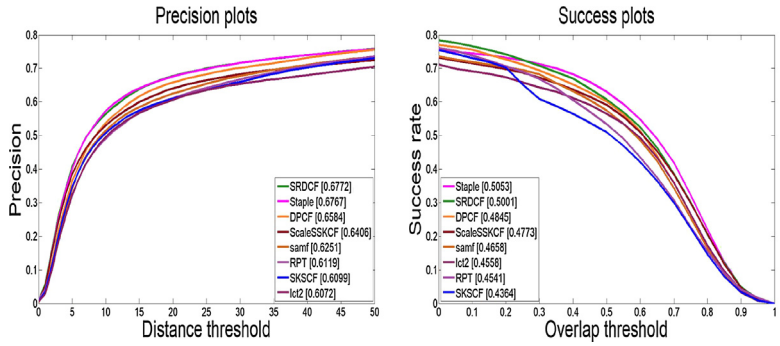

Fig. 6. Precision and success plots over all 128 sequences in TempleColor128 dataset. For the success plots, the area under the curve (AUC) scores of each tracker are reported in the legends. And the precision obtained at threshold 20 is shown in the legends of the precision plots.

Table 7
Comparison results on the VOT2015 dataset. The top two results are highlighted by red and blue.

	CCOT	deepSRDCF	Staple	SRDCF	DPT	samf	DPCF	SKSCF	lct2	ScaleSSKCF(ours)
Accuracy	0.52	0.56	0.53	0.53	0.48	0.51	0.51	0.50	0.52	0.55
Robustness	0.85	1.00	1.35	1.53	1.75	2.08	2.15	2.40	2.52	1.75
EAOP	0.325	0.318	0.291	0.245	0.234	0.202	0.191	0.185	0.175	0.252

As is known, except the accuracy and robustness, the tracking speed is also very crucial in many real tracking application. Therefore, we visualize the expected overlap score with respect to the tracking speed measured in EFO units in Fig. 7, where we exclude the CCOT and deepSRDCF for fairness. Seen from the Fig. 7, our method achieves the better balance between the performance and speed.

5. Conclusions

In this paper, we proposed a scale-adaptive structural support kernel correlation filter tracking model, which is called ScaleSSKCF. Our method combines part-based tracking strategy into support correlation filter tracker by the structural constraint term of the proposed model, which remains the strong discriminability of the support correlation filter (SCF) and also preserves the spatial structure of the target. To reduce the issues of drifting away from the object, we consider the temporal consistency of each part in our model. In addition, we also introduce the occlusion detection and scale estimation into the proposed tracking method, which makes our tracker less sensitive to some complex factors (*e.g.*, partial occlusion and scale variation). Results on three benchmark datasets show that our tracker performs favorably against several state-of-the art tracking methods in terms of accuracy, robustness and speed.

Declaration of Competing Interest

The authors declared that there is no conflict of interest.

Acknowledgment

This work is supported by the National Natural Science Foundation of China Under Grant No. 61602288, 61703252 and Shanxi Provincial Natural Science Foundation of China Under Grant No. 201701D221102. The authors also would like to thank the anonymous reviewers for their valuable suggestions.

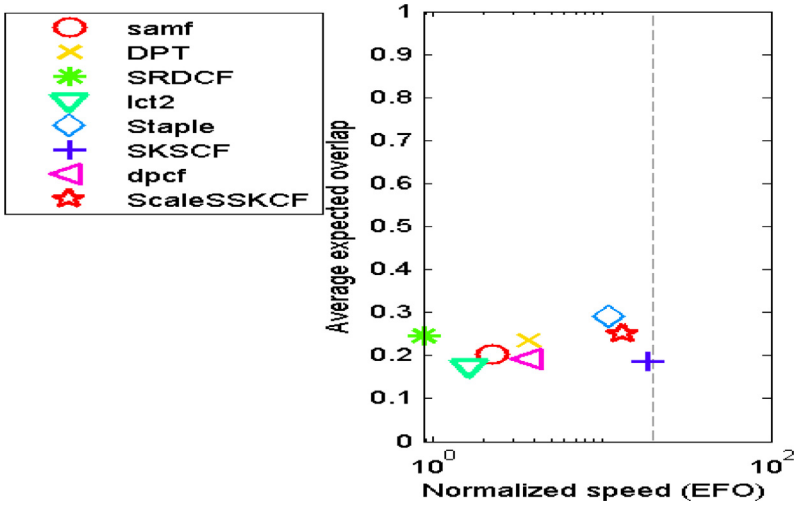

Fig. 7. Expected average overlap scores with respect to the tracking speed in EFO units. The dashed vertical line denotes the estimated real-time performance threshold of 20 EFO units.

References

[1] B. Babenko, M.H. Yang, S. Belongie, Visual tracking with online multiple instance learning, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 983–990.

[2] S. Hare, A. Saffari, P.H.S. Torr, Struck: structured output tracking with kernels, in: IEEE International Conference on Computer Vision, 2011, pp. 263–270.

[3] S. Avidan, Support vector tracking, IEEE Trans. Pattern Anal. Mach. Intell. 26 (8) (2004) 1064–1072.

[4] D.S. Bolme, J.R. Beveridge, B.A. Draper, Y. M. Lui, Visual object tracking using adaptive correlation filters, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

[5] K.H. Zhang, L. Zhang, M.H. Yang, Real-time compressive tracking, in: ECCV, 2012, pp. 864–877.

[6] Z. Ji, W. Wang, Robust object tracking via multi-task dynamic sparse model, in: IEEE International Conference on Image Processing, 2014.

[7] B. Liu, L. Yang, J. Huang, C.A. Kulikowski, Robust and fast collaborative tracking with two stage sparse optimization, in: ECCV, 2010, pp. 624–637.

[8] T. Zhang, B. Ghanem, S. Liu, N. Ahuja, Robust visual tracking via multi-task sparse learning, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 2042–2049.

[9] Z. Ji, W. Wang, N. Xu, Robust object tracking via incremental subspace dynamic sparse model, in: IEEE International Conference on Multimedia and Expo, 2014, pp. 1–6.

[10] X. Mei, H. Ling, Y. Wu, E. Blasch, L. Bai, Minimum error bounded efficient l_1 tracker with occlusion detection, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 1257–1264.

[11] A. Rodriguez, V.N. Boddeti, B.V. Kumar, A. Mahalanobis, Maximum margin correlation filter: a new approach for localization and classification, IEEE Trans. Image Process. 22 (2) (2013) 631–643.

[12] W. Zuo, X. Wu, L. Lin, L. Zhang, M.H. Yang, Learning support correlation filters for visual tracking, 2016. arXiv preprint arXiv: 160106032.

[13] Z. Ji, W. Wang, Object tracking based on local dynamic sparse model, J. Vis. Commun. Image R 28 (2015) 44–52.

[14] Y. Li, J. Zhu, S.C.H. Hoi, Reliable patch trackers: robust visual tracking by exploiting reliable patches, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 353–361.

[15] T. Liu, G. Wang, Q. Yang, Real-time part-based visual tracking via adaptive correlation filters, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 4902–4912.

[16] S. Liu, T. Zhang, X. Cao, C. Xu, Structural correlation filter for robust visual tracking, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 4312–4320.

[17] R. Yao, Q. Shi, C. Shen, Y. Zhang, A. Van Den Hengel, Part-based visual tracking with online latent structural learning, in: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE, 2013, pp. 2363–2370.

[18] X. Li, W. Hu, C. Shen, Z. Zhang, A survey of appearance models in visual object tracking, ACM Trans. Intell. Syst. Technol. 4 (4) (2013) 58–105.

[19] A. Yilmaz, O. Javed, M. Shah, Object tracking: a survey, ACM Comput. Surv. 38 (4) (2006).

[20] C. Li, X. Liang, Y. Lu, N. Zhao, J. Tang, Rgb-t object tracking: Benchmark and baseline, 2018a. arXiv preprint arXiv: 180508982.

[21] C. Li, C. Zhu, Y. Huang, J. Tang, L. Wang, Cross-modal ranking with soft consistency and noisy labels for robust rgb-t tracking, in: European Conference on Computer Vision, Springer, 2018, pp. 808–823.

[22] J. Ning, J. Yang, S. Jiang, L. Zhang, M.H. Yang, Object tracking via dual linear structured svm and explicit feature map, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 4266–4274.

[23] J.F. Henriques, J. Carreira, R. Caseiro, J. Batista, Beyond hard negative mining: efficient detector learning via block-circulant decomposition, in: IEEE International Conference on Computer Vision, 2013, pp. 2760–2767.

[24] M. Wang, Y. Liu, Z. Huang, Large margin object tracking with circulant feature maps, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 2017, pp. 21–26.

[25] J.F. Henriques, R. Caseiro, P. Martins, J. Batista, High speed tracking with kernelized correlation filters, IEEE Trans. Pattern Anal. Mach. Intell. 37 (3) (2015) 583–596.

[26] A.S. Montero, J. Lang, R. Laganière, Scalable kernel correlation filter with sparse feature integration, in: IEEE International Conference on Computer Vision Workshops, 2015, pp. 587–594.

[27] Z. Kalal, K. Mikolajczyk, J. Matas, Tracking-learning-detection, IEEE Trans. Pattern Anal. Mach. Intell. 34 (7) (2012) 1409–1422.

[28] G. Nebehay, R. Pflugfelder, Clustering of static-adaptive correspondences for deformable object tracking, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 2784–2791.

[29] M. Danelljan, G. Häger, M. Felsberg, Accurate scale estimation for robust visual tracking, in: BMVC, 2014.

[30] Y. Li, J. Zhu, A scale adaptive kernel correlation filter tracker with feature integration, in: ECCV, 2014.

[31] X. Li, Q. Liu, Z. He, H. Wang, C. Zhang, W.S. Chen, A multi-view model for visual tracking via correlation filters, Knowl.-Based Syst. 113 (2016) 88–99.

[32] H.K. Galoogahi, T. Sim, S. Lucey, Correlation filters with limited boundaries, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[33] M. Danelljan, G. Häger, F.S. Khan, M. Felsberg, Learning spatially regularized correlation filters for visual tracking, in: IEEE International Conference on Computer Vision, 2015.

[34] P. Chockalingam, N. Pradeep, S. Birchfield, Adaptive fragments based tracking of non-rigid objects using level sets, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 1530–1537.

[35] O. Akin, E. Erdem, A. Erdem, K. Mikolajczyk, Deformable part-based tracking by coupled global and local correlation filters, J. Vis. Commun. Image Represent. 38 (2016) 763–774.

[36] Z. He, S. Yi, Y.M. Cheung, X. You, Y.Y. Tang, Robust object tracking via key patch sparse representation, IEEE Trans. Cybernet. 47 (2) (2016) 354–364.

[37] X. Sun, N.M. Cheung, H. Yao, Y. Guo, Non-rigid object tracking via deformable patches using shape-preserved kcf and level sets, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5495–5503.

[38] A. Lukežič, L. Čehovin, M. Kristan, Deformable parts correlation filters for robust visual tracking, IEEE Trans. Cybernet. (2017) 1–13.

[39] C.P. Lee, C. b. Lin, A study on l_2-loss (squared hinge-loss) multiclass svm, Neural Comput. 25 (5) (2013) 1302–1323.

[40] K. Zhang, L. Zhang, Q. Liu, D. Zhang, M.H. Yang, Fast visual tracking via dense spatio-temporal context learning, in: ECCV, 2014.

[41] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.

[42] S. Boyd, Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends Mach. Learn. 3 (2010) 1–122.

[43] M. Danelljan, F. Shahbaz Khan, M. Felsberg, J. Van de Weijer, Adaptive color attributes for real-time visual tracking, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

[44] Y. Wu, J. Lim, M.H. Yang, Online object tracking: a benchmark, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2013.

[45] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, F. Porikli, L. Cehovin, et al., The visual object tracking vot2015 challenge results, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2015.

[46] P. Liang, E. Blasch, H. Ling, Encoding color information for visual tracking: algorithms and benchmark, IEEE Trans. Image Process. (T-IP) 24 (12) (2015) 5630–5644.

[47] Y. Wu, J. Lim, M.H. Yang, Object tracking benchmark, IEEE Trans. Pattern Anal. Mach. Intell. 37 (9) (2015) 1834–1848.

[48] M. Everingham, L.V. Gool, C. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, Int. J. Comput. Vision 88 (2) (2010) 303–338.

[49] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, P.H. Staple Torr, Complementary learners for real-time tracking, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[50] C. Ma, X. Yang, C. Zhang, M.H. Yang, Long-term correlation tracking, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[51] M. Danelljan, A. Robinson, F.S. Khan, M. Felsberg, Beyond correlation filters: learning continuous convolution operators for visual tracking, in: European Conference on Computer Vision, Springer, 2016, pp. 472–488.

[52] N. Wang, J. Shi, D.Y. Yeung, J. Jia, Understanding and diagnosing visual tracking systems, in: IEEE International Conference on Computer Vision (ICCV), IEEE, 2015, pp. 3101–3109.