Neurocomputing 214 (2016) 109–120

Contents lists available at ScienceDirect

Neurocomputing

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

Geometric affine transformation estimation via correlation filter for visual tracking

Fanghui Liu, Tao Zhou, Jie Yang*

Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, China

ARTICLE INFO

Article history:
Received 16 December 2015
Received in revised form
12 May 2016
Accepted 19 May 2016
Available online 6 June 2016

Keywords:
Correlation filter
Part-based
Affine transformation estimation
Object tracking

ABSTRACT

Correlation filter achieves promising performance with high speed in visual tracking. However, conventional correlation filter based trackers cannot tackle affine transformation issues such as scale variation, rotation and skew. To address this problem, in this paper, we propose a part-based representation tracker via kernelized correlation filter (KCF) for visual tracking. A Spatial–Temporal Angle Matrix (STAM), severed as confidence metric, is proposed to select reliable patches from parts via multiple correlation filters. These stable patches are used to estimate a 2D affine transformation matrix of the target in a geometric method. Specially, the whole combination scheme for these stable patches is proposed to exploit sampling space in order to obtain numerous affine matrices and their corresponding candidates. The diversiform candidates would help to seek for the optimal candidate to represent the object's accurate affine transformation in a higher probability. Both qualitative and quantitative evaluations on VOT2014 challenge and Object Tracking Benchmark (OTB) show that the proposed tracking method achieves favorable performance compared with other state-of-the-art methods.

1. Introduction

Visual tracking is an attractive research area in computer vision with many real-world applications such as video surveillance, human computer interaction and motion analysis [1,2]. There are many challenging factors (partial occlusions, scale variation, shape deformation, varying illumination and so on), as the main limitation of improvements of trackers. Notwithstanding many achievements [3,4] have been made in visual tracking during the past several years, there are still improving space in trackers for more complicated situations.

Most tracking methods can be categorized two classes: generative or discriminative methods. Generative methods focus on searching the most similar candidate with the minimal reconstruction error; discriminative approaches cast tracking problem as a binary classification problem that separates the target from the background. In generative methods, various appearance models are based on subspace learning [5,6], sparse representation [7,8]. Ma et al. [9] exploit linear subspace learning to local linear subspace learning for visual tracking. In [10], bayes joint decision and estimation is incorporated into structure sparsity representation. In discriminative methods, a classifier distinguishes the target from the background via discriminative features [11,12]. Chen et al. [13] combine Support Vector Data Description and Structured Output SVM to composite the appearance model. Besides, manifold ranking [14,15] and label propagation [16] are representative semi-supervised learning methods, which has been incorporated into visual tracking framework. Convolutional neural network (CNN) is served as a black-box feature extractor to encode more semantic features and discriminative information [17,18] to exploit represent ability of a tracker.

Recently correlation filter has been successfully applied in visual tracking because of its favorable location property [19–21]. Correlation filters are designed to produce correlation peaks on the target while yielding low response to the background. The typical representative correlation filter is minimum output sum of squared error (MOSSE) filter [22], which is the first correlation filter based method applied in visual tracking. MOSSE seeks for a filter (loosely called a "template" in some literature, or called "classifier" in visual tracking area) by minimizing the output sum of squared error between actual correlation outputs and desired "Gaussian-shape" correlation outputs. And then the target is searched in a relatively larger search window in the next frame, whose location is determined by the maximum value in correlation responses. Considering the tracking performance is limited to linear classifier, by taking advantage of kernel trick, the tracking system with kernelized correlation filter (KCF) [23] achieves superior performance with high speed.

* Corresponding author.
E-mail addresses: lfhsgre@outlook.com (F. Liu), zhou.tao@sjtu.edu.cn (T. Zhou), jieyang@sjtu.edu.cn (J. Yang).

http://dx.doi.org/10.1016/j.neucom.2016.05.079
0925-2312/© 2016 Elsevier B.V. All rights reserved.

However, KCF tracker cannot be adaptive to affine transformation (e.g. scale, rotation and skew) of the target. Various improvement works have been proposed to tackle scale variation issue. Two conventional solutions to scale variation problem are multi-scale search scheme [24] and part-based methods [25–27]. Danelljan et al. [24] relieve the scaling issue using feature pyramid and 3-dimensional correlation filter. On the other hand, part-based methods [25,28] become popular because of their robustness to partial occlusions. And they can solve scale variation problem involved with the relationship among each part. In [29], by assigning adaptive weights to confidence maps of each independent patch, a joint map can be constructed to predict the new state via particle filter method. Reliable Patch Trackers (RPT) [27] adopt Voting-like scheme to cluster homo-trajectory patches for the scale factor estimation. Specially, sequential Monte Carlo framework [26] is easily introduced into part-based methods to reflect the relationship among these patches. The shortcoming is that particles with adaptive weighting maybe have a high computational complexity cost. Besides, phase spectrum implemented by phase correlation filter [30] is incorporated into KCF to estimate size change of the target. Despite that these above methods tackle scale variation issue well, they all overlook other affine transformations (rotation and skew).

To account for object rotation and deformation in visual tracking, keypoint-based trackers [31,32] represent the object by local descriptors like SIFT, SURF features. In this case, visual tracking problem can be regarded as a matching problem between two consecutive frames. However, it may be difficult for these trackers to capture global information and spatial information among neighboring points.

Motivated by the above issues, in this paper, we propose a part-based correlation filter tracker with geometric method, which is able to handle scale variation, deformation, rotation, partial occlusions and other challenging factors. The main contributions of this paper are as follows.

(i) Spatial–Temporal Angle Matrix (STAM), served as a confidence metric criterion, is proposed to measure how likely a part is reliable, which takes spatial information among patches and temporal information between two consecutive frames into consideration.

(ii) The whole combination scheme is proposed to construct a further various candidate set whose samples are with different affine transformation matrices with geometric method. It exploits the diversification of sample space to get a final reliable and accurate tracking result.

The remainder of the paper is organized as follows. Section 2 gives the relevant related work of KCF tracker. Section 3 demonstrates details of the proposed method. Experimental results on two benchmarks and performance evaluation are included in Section 4. Finally, conclusion is given in Section 5.

2. Related work about KCF tracker

In this section, we briefly introduce related content about KCF tracker [23]. A classifier is learned to find the corresponding relation between input image patch $\mathbf{x}_i$ and its label y_i in training set, denoted as $f(\mathbf{x}_i) = y_i$. The goal of training problem in KCF tracker is to find a function that minimizes the squared error over samples x_i and their labels y_i,

$$\min_{\mathbf{w}} \sum_i (f(\mathbf{x}_i, \mathbf{w}) - y_i)^2 + \lambda \parallel \mathbf{w} \parallel^2 \tag{1}$$

where the function $f(\mathbf{x}_i, \mathbf{w}) = \mathbf{w}^T\mathbf{x}_i$ and λ is a regularization term to avoid over-fitting. This formula uses regularized quadratic loss function, named as Regularized Least Squares (RLS). The objective function has a closed-form solution, and the complex version is shown in the Fourier domain,

$$\mathbf{w} = (\mathbf{X}^{\dagger}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^{\dagger}\mathbf{y} \tag{2}$$

where $\mathbf{X}$ is a matrix whose rows are vectorized training samples, $\mathbf{X}^{\dagger}$ is the Hermitian transpose, $\mathbf{y}$ is the corresponding label vector, and $\mathbf{I}$ is an identity matrix.

To introduce the kernel functions mapping the inputs of a linear problem to a non-linear feature-space $\varphi(\mathbf{x})$, $\mathbf{w}$ can be represented by a linear combination of input data samples $\mathbf{w} = \sum_i \alpha_i \varphi(\mathbf{x}_i)$. Then $f(\mathbf{x}_i)$ takes the form by Representer Theorem,

$$f(\mathbf{x}_i) = \sum_{i=1}^{n} \alpha_i \kappa(\mathbf{x}_i, \mathbf{x}_j) \tag{3}$$

where $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle \varphi(\mathbf{x}_i), \varphi(\mathbf{x}_j) \rangle$ is the kernel function. Define the kernel matrix $\mathbf{K}$ with elements $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$. Thus the solution $\mathbf{w}$ of Eq. (1) is inverted to obtain α by kernel functions,

$$\boldsymbol{\alpha} = (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{y} \tag{4}$$

Circulant matrix $\mathbf{C}(x)$ is introduced to avoid complex matrix inverse operator of Eq. (4). Since the kernel matrix $\mathbf{K}$ is a circulant matrix, the solution of Eq. (4) is obtained,

$$\boldsymbol{\alpha} = \mathcal{F}^{-1}\left(\frac{\mathcal{F}(\mathbf{y})}{\mathcal{F}(\mathbf{k}^{\mathbf{xx}}) + \lambda} \right) \tag{5}$$

where $\mathcal{F}$ denotes the Fourier transform and $\mathcal{F}^{-1}$ is the inverse Fourier transform. $\mathbf{k}^{\mathbf{xx}}$ is the first row of the kernel matrix $\mathbf{K} = \mathbf{C}(\mathbf{k}^{\mathbf{xx}})$.

In tracking process, the target can be detected by the trained parameter $\boldsymbol{\alpha}$ and training samples $\mathbf{x}$ in a new frame. For a new sample $\mathbf{z}$, the confidence value is calculated as,

$$\mathbf{y} = \mathcal{F}^{-1}(\mathcal{F}(\mathbf{k}^{\mathbf{xz}}) \odot \mathcal{F}(\boldsymbol{\alpha})) \tag{6}$$

where $\odot$ is the element-wise product. The new position of the target is determined by the position with a maximum value in $\mathbf{y}$.

3. The proposed method

Fig. 1shows the main steps of the proposed method. We first sample nine overlapped local image patches inside the target region with a spatial layout. Then the entire target and nine part-based patches are tracked by KCF tracker, to locate them in the next frame. By imposing Peak-to-Sidelobe Ratio (PSR) and Spatial–Temporal Angle Matrix (STAM) as confidence metric, stable patches are selected to estimate affine transformation of the target. Changes of these center locations can derive an affine matrix to reflect the object's scale variation, rotation and skew. Finally these stable patches are employed to update parameters in KCF tracker regardless of the remaining unreliable patches.

3.1. Stable patches obtained for location

In this subsection, we present a reliable part-based selection scheme to seek for stable patches. In our proposed method, each part is regarded as independent and two constraints are imposed to seek for stable patches.

3.1.1. Peak-to-Sidelobe Ratio (PSR)

In signal processing theory, Peak-to-Sidelobe Ratio [33] is widely used to measure the signal peak strength in response map. It quantifies the sharpness of the correlation peak, as an effective confidence metric in visual tracking [29,27]. The higher PSR value

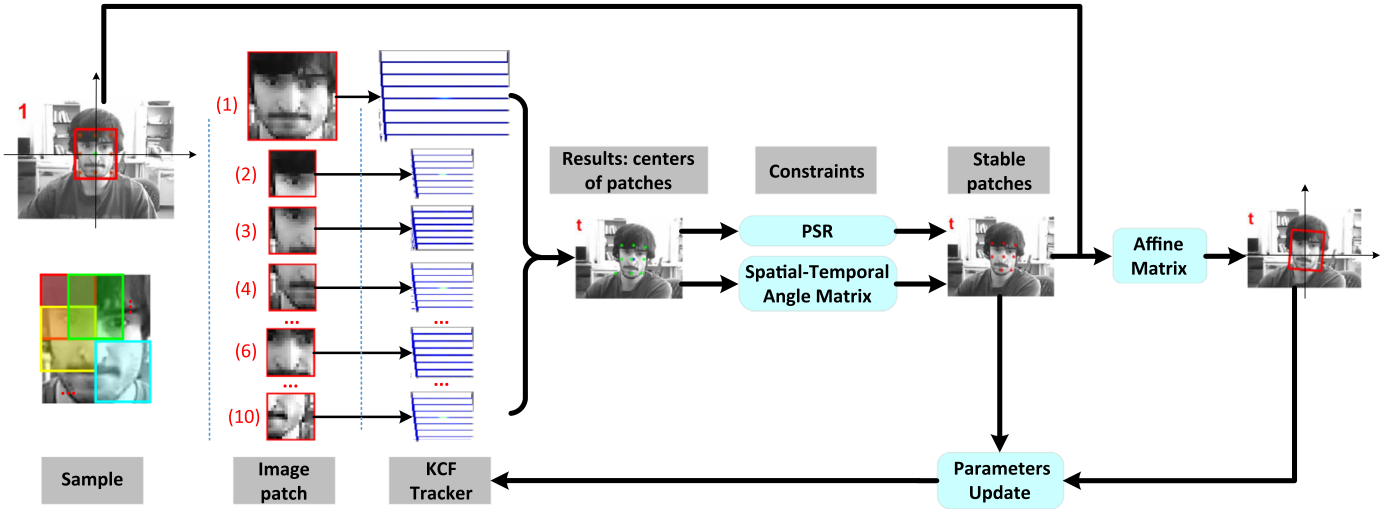


Fig. 1. The flowchart of the proposed tracking framework.

means more confident detection or location in a new frame. PSR is defined as,

$$\mathrm{PSR} = \frac{\max(\mathbf{R}) - \mu_{\mathbf{R}}}{\sigma_{\mathbf{R}}} \tag{7}$$

where $\mathbf{R}$ represents a response map, $\mu_{\mathbf{R}}$ and $\sigma_{\mathbf{R}}$ are the mean value and standard deviation of $\mathbf{R}$ respectively. Therefore PSR can be treated as the confidence for a patch to measure whether it is tracked properly.

For KCF tracker, the PSR typically exceeds 20, which indicates very strong peaks. Statistical results show that when PSR drops to around 10, it is an indication that the detection of corresponding part is not reliable. And the corresponding part may suffer heavy occlusions or drifts. In this study, we set the threshold $psr_{thres} = 15$. Define an indicator vector $\mathbf{w}$ to show which patch is reliable to be reserved ($w_i = 1$) and which one is rejected ($w_i = 0$) by PSR value,

$$\forall\ i = 1, 2, \ldots, 10,\ w_i = \begin{cases} 1, & \text{if } psr_i \le psr_{thres}; \\ 0, & \text{otherwise}. \end{cases} \tag{8}$$

The first problem is to estimate the target's center location. If PSR value of the first patch (the entire target) exceeds psr_{thres}, we use its center location as the estimated center of the target in the next frame. If the whole patch is regarded as unreliable (PSR value is too low), the center location of the target should be estimated by other local reliable parts. A local reliable patch can derive a center location of the target based on geometrical configuration. The output center location of the target is the mean value of these respective center locations estimated by their corresponding reliable local patches.

3.1.2. *Spatial–Temporal Angle Matrix (STAM)*

It is deficient to obtain stable patches only by PSR value because spatial information among these parts are not taken into consideration. An angle matrix is constructed to represent temporal relationship between the two consecutive frames and spatial relationship among these patches, so called Spatial–Temporal Angle Matrix $\mathbf{M} \in \mathbb{R}^{10\times 10}$ shown in Fig. 2.

It is not difficult to find that STAM is a symmetric matrix and elements in its domain diagonal are zero. Specially, θ_6 in $\mathbf{M}$ is fixed to zero just because the sixth local patch's center is the same with the whole target's center in theory. A small change in geometric position can result in a significant angle change. This value could not provide certain significance and thereby be overlooked. Between the two consecutive frames, the spatial relationship between a part-based patch and another part is stable and does not easily change in a short time. Normally, if each patch and the whole target are tracked well, l_{T-1}^{ij} is parallel to l_T^{ij} between the two consecutive frames and θ_{ij} should be close to zero or a small value. The larger the θ_{ij} is, the less reliable the patch tracking result is.

But without ignorance, the target with affine transformation would introduce additional changes in $\mathbf{M}$. We cast $\mathbf{M}$ into two parts: the first row and the other rows. Elements in the first row of $\mathbf{M}$ denote angle changes from the whole target's center to each local patch's center between the two consecutive frames;

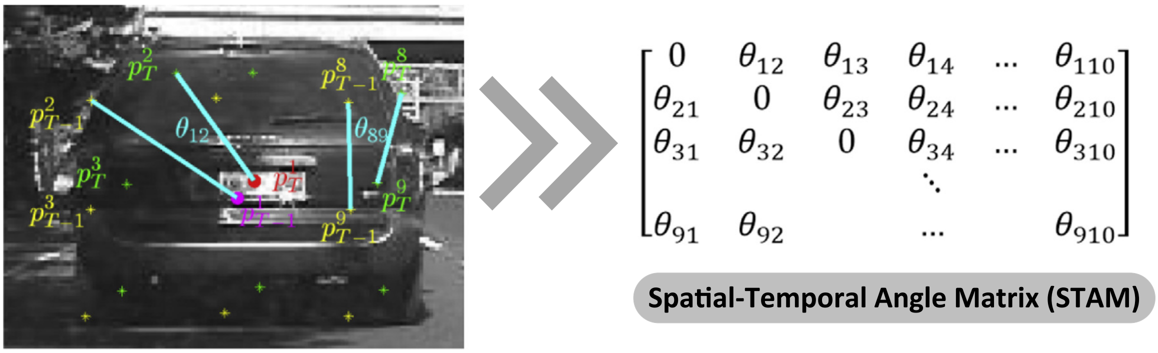

p_T^2
p_{T-1}^2
p_T^3
p_{T-1}^3
θ_{12}
p_T^1
p_{T-1}^1
p_{T-1}^8
p_T^8
θ_{89}
p_T^9
p_{T-1}^9
$\begin{bmatrix} 0 & \theta_{12} & \theta_{13} & \theta_{14} & \cdots & \theta_{110} \\ \theta_{21} & 0 & \theta_{23} & \theta_{24} & \cdots & \theta_{210} \\ \theta_{31} & \theta_{32} & 0 & \theta_{34} & \cdots & \theta_{310} \\ & & & \ddots & & \\ \theta_{91} & \theta_{92} & & \cdots & & \theta_{910} \end{bmatrix}$
Spatial-Temporal Angle Matrix (STAM)

Fig. 2. Spatial–Temporal Angle Matrix. The centroid of the entire target is p_{T-1}^1 at the $(T-1)$-frame and $p_T{}^1$ at the T-frame. The center of each overlapped local patch from top to down, left to right is defined as p_{T-1}^2, p_{T-1}^3, …, p_{T-1}^9 and p_{T-1}^{10} in yellow color at the $(T-1)$-frame. Similarly, at the T-frame, these centers of local patches are $p_T{}^2$,…, $p_T{}^9$ and $p_T{}^{10}$ in green color respectively. A connection line l_{T-1}^{ij} between p_{T-1}^i and p_{T-1}^j, another connection line l_T^{ij} between $p_T{}^i$ and $p_T{}^j$. These two lines form an angle θ_{ij} to construct STAM. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

elements in another row of **M** represent angle changes from a local patch's center to another patch. Different affine transformations lead to different changes in these two parts. For example, when the object suffers scale variation, elements in **M** remain stable. When the object suffers in-plane-of-rotation, some changes might be revealed on θ_{1i}, whereas θ_{ij} $(i, j \neq 1)$ seems to change relatively small. When the object occurs skew, some elements in the first row of **M** would be stable while some changes might be revealed on a few elements of θ_{ij} $(i, j \neq 1)$. It is relatively complex to qualitatively analyze value changes in **M** when the target suffers skew. Overall, if the target is tracked well, considering smoothness between the two consecutive frames, most elements in **M** are still stable in spite of affine transformation on the target. On the contrary, if a patch occurs drifts (regarded as outlier), quite a number of values in **M** might be large. Thus the angle change is an evaluation criterion that measures the reliability level of a patch's tracking result.

Values in **M** that exceed θ_{thres} are recorded. If the value θ_{1i} occurs in the first row of **M**, we drop out the corresponding i-th patch. If this value θ_{ij} occurs another row of **M**, we drop out the corresponding with larger row sum. Assumed that the sum of i-th row is with the larger one, it means that the i-th patch is less stable than the j-th patch. In our experiment, θ_{thres} is set to 15°. For example, in the left figure of Fig. 2, θ_{12} and θ_{89} exceed θ_{thres}. Depending on the above rules, the second and the eighth patches are abandoned.

Through these two criteria, unreliable part-based patches are rejected and stable patches are reserved to estimate the affine matrix $\mathbf{G} \in \mathbb{R}^{2\times 2}$. The affine matrix and center location of the target determine the final tracking result.

3.2. Affine matrix estimation

After stable patches obtained, how to estimate the object's affine transformation matrix is a deliberate problem. In this subsection, center points of stable patches are utilized to calculate the affine matrix. These center points reflect changes in the object's scale, deformation and rotation compared with corresponding centroids in the first frame. Specially, the whole combination scheme is proposed to exploit sampling space for various candidates with diversiform affine transformations shown in Fig. 3.

Supposed that m stable patches have been obtained at t-th frame, their respective centers are represented by 2D points (pixel coordinates in an image), named as $p_t^1, p_t^2, \ldots, p_t^m$. Compared to these corresponding patches' centers location at the first frames (denoted as $p_1^1, p_1^2, \ldots, p_1^m$), an affine matrix $\mathbf{A} \in \mathbb{R}^{3\times 3}$ can be used to describe changes of these center points at t-th frame. The reason why we do not adopt data points at $(t-1)$-th frame as the primitive points is that there are narrow changes between these two consecutive frames. Besides, this scheme more easily results in error accumulation compared to the adoption of those points at the first frame. Take points p_1^1 and p_t^1 as example,

$$\begin{bmatrix} x_t^1 \\ y_t^1 \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} a_{11} & a_{12} & d_x \\ a_{21} & a_{22} & d_y \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} x_1^1 \\ y_1^1 \\ 1 \end{bmatrix} \tag{9}$$

where p_t^1, p_1^1 are represented using homogeneous coordinates $[x_t^1, y_t^1, 1]^T$ and $[x_1^1, y_1^1, 1]^T$. In the affine matrix $\mathbf{A}$, vector $\vec{\mathbf{d}} = [d_x, d_y]^T$ represents translation from the first frame to the t-th frame. Considering center location of the target has been obtained, the translation vector $\vec{\mathbf{d}}$ is known. Thus after centering these points into the same coordinate, only four parameters a_{11}, a_{12}, a_{21} and a_{22} need to be estimated. We denote them as the 2D affine matrix **G**:

$$\underbrace{\begin{bmatrix} x_t^1 & x_t^2 & \cdots & x_t^m \\ y_t^1 & y_t^2 & \cdots & y_t^m \end{bmatrix}}_{\mathbf{Y}} = \underbrace{\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}}_{\mathbf{G}} \underbrace{\begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^m \\ y_1^1 & y_1^2 & \cdots & y_1^m \end{bmatrix}}_{\mathbf{X}} \tag{10}$$

where these m stable patches' center locations at the first and t-th frames, as data points **X** and **Y** respectively, are used to estimate the affine matrix **G**. This affine matrix is a nonsingular matrix and it can represent affine transformation including rigid rotation, similarity and so on. Normally, only using two 2D points can estimate these four parameters in **G**. If m is larger than 2, the affine matrix estimation problem is inverted to solve a linear overdetermined equation system:

$$\min_{\mathbf{G}} \| \mathbf{Y}^T - \mathbf{X}^T \mathbf{G}^T \|_2^2 \tag{11}$$

The least square solution is $\mathbf{G} = \mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}$. If there is only one stable patch left, this problem degenerates to an underdetermined equation system. In this case, we only use the stable patch to calculate the scale factor regardless of other parameters in **G**.

However, when we use $m\ (>2)$ patches to solve a linear overdetermined equation system in order to estimate the affine matrix **G**, we could not guarantee that the estimated affine matrix $\mathbf{G}^*$ is fit for each 2D point if noise point(s) is mixed in. The matrix $\mathbf{G}^*$ is a tradeoff result among numerous data points. In other words, an affine matrix estimated by fewer data points might be more appropriate. It has a similar fashion with “sparse view” vs. “averaged view”. These two views have their respective pros and cons. An affine matrix obtained by a few data points (“sparse”)

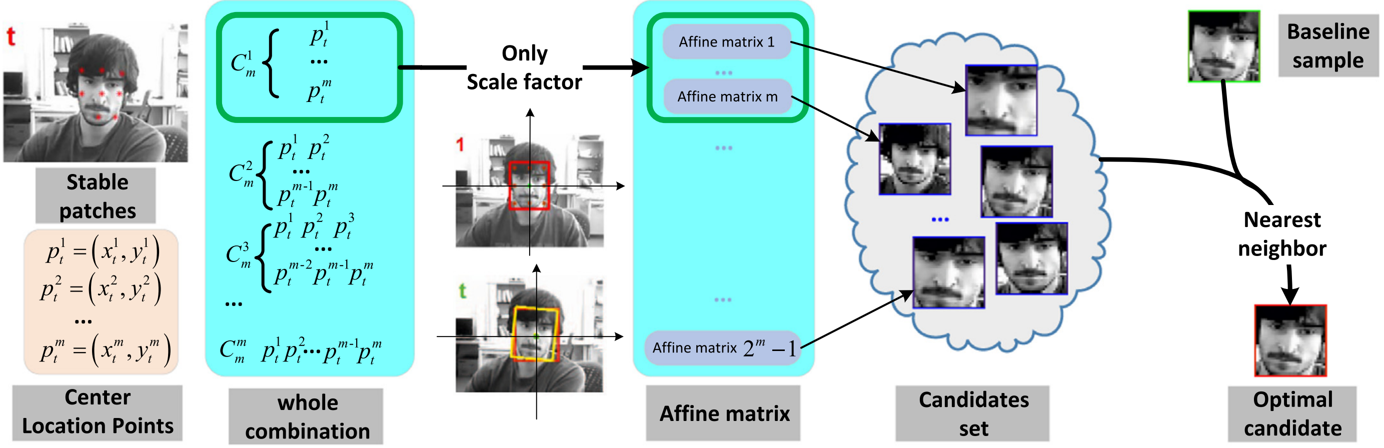


Fig. 3. The flowchart of affine matrix estimation to obtain the optimal candidate.

might be accurate but unfaithful to represent the object's transformation. All stable points are often to provide a relatively stable affine matrix estimation ("averaged view").

To combine their respective merits, the whole combination scheme is proposed to exploit sampling space for various affine transformation matrices. We give the whole combination of all these m data points from ${C_m}^1$ to ${C_m}^m$ shown in Fig. 3. Each combination of ${C_m}^k$ represents that k different points, selected from all m stable points, are used to estimate the affine matrix. Different selections lead to different affine matrices, which can also broaden the diversity of affine transformations. There are $C_m^1 + C_m^2 + \cdots + C_m^m = 2^m - 1$ affine matrices and their respective affine transformations.

Based on different affine matrices, numerous candidates are sampled by these various affine transformations. One affine matrix corresponds to one candidate. In total, $2^m - 1$ candidates are produced and form the candidate set. Compared with only one affine matrix estimated by all m data points, the number of candidates rises from only one to $2^m - 1$. The diversiform candidates would lead to a higher probability to seek for the optimal candidate.

A simple but effective method is proposed to seek for the optimal candidate from the candidate set. Among the candidate set, the nearest neighbor of a baseline sample is chosen as the optimal candidate. The baseline sample is obtained by the mean value of tracking results in the previous 5 frames. The distance metric is with ℓ_2 norm in image grayscale value.

3.3. Parameters updating in KCF tracker

In conventional KCF tracker, the classifier coefficients are updated simply with a fixed learning rate γ,

$$\begin{cases} \mathcal{F}(\boldsymbol{\alpha})^t = (1-\gamma)\mathcal{F}(\boldsymbol{\alpha})^{t-1} + \gamma\mathcal{F}(\boldsymbol{\alpha}); \\ \hat{\mathbf{x}}^t = (1-\gamma)\hat{\mathbf{x}}^{t-1} + \gamma\hat{\mathbf{x}}, \end{cases} \tag{12}$$

where $\mathcal{F}(\boldsymbol{\alpha})$ is the classifier coefficients and $\mathbf{x}$ is input image patch. In our method, the update schemes in KCF tracker are formed as,

$$\begin{cases} \mathcal{F}(\boldsymbol{\alpha})_i^t = (1-\gamma)\mathcal{F}(\boldsymbol{\alpha})_i^{t-1} + \gamma\,[s_i\mathcal{F}(\boldsymbol{\alpha})_i + (1-s_i)C_\alpha^i]; \\ \hat{\mathbf{x}}_i^t = (1-\gamma)\hat{\mathbf{x}}_i^{t-1} + \gamma\,[s_i\hat{\mathbf{x}}_i + (1-s_i)C_{\mathbf{x}}^i], \end{cases} \tag{13}$$

where C_α^i is defined as the mean value of $\mathcal{F}(\boldsymbol{\alpha})_i$ and $C_{\mathbf{x}}^i$ is the mean value of $\hat{\mathbf{x}}_i^t$, as uniform priors respectively. For i-th part, if it is regarded as stable, $s_i{=}1$ means that update schemes are the same as that of traditional KCF tracker. If it is regarded as unreliable, $s_i{=}0$ means that the current parameters $\mathcal{F}(\boldsymbol{\alpha})_i$ and $\hat{\mathbf{x}}_i$ are not accurate. These parameters are substituted for their mean values for update.

4. Experimental results

In this section, we give details of our experimental implementation and discuss the results of tacking performance evaluation on VOT2014 [34] and Object Tracking Benchmark (OTB) [35]. And the effectiveness of individual schemes have been verified.

4.1. Experimental setup

The proposed tracker was implemented in MATLAB without further optimization. All experiments were conducted on a regular PC with Intel Xeon E5506 CPU (2.13 GHz) and 24 GB memory. The corresponding parameters were all the same as those of the KCF tracker: interpolation factor=0.02, Gaussian kernel correlation $\sigma = 0.5$, and regulation term $\lambda = 10^{-4}$, a HOG cell size of 4×4 and 9 orientations bins.

4.2. Overall performance

We give an overall performance on two benchmarks including VOT2014 and OTB. In VOT Challenge, two evaluation criteria (i.e., accuracy and robustness) are used. Accuracy is measured as the Pascal VOC Overlap Ratio (VOR) [36]. It measures overlapping degree between the tracked bounding box and the ground truth box, defined as $e = \frac{area(R_T \cap R_G)}{area(R_T \cup R_G)}$, where R_T and R_G are the area of tracked and ground truth box respectively. The robustness indicates the number of failures to track an object in a sequence. A failure is determined when the VOR score goes to zero. In [34,37], they point out that there is no meaning for a tracker to calculate related tracking evaluation after its VOR score declines to zero. Therefore a restart scheme is incorporated into a tracker in VOT challenge, which is the most difference from other benchmarks.

Two typical evaluation criteria are utilized in OTB. The first one is mean Center Location Error (CLE), which is pixel distance between the centroid of the tracking result and the ground truth. The second one is the VOR as aforementioned. Based on these two evaluation metrics, precision plot and success plot [35] show the percentage of the threshold and successfully tracked frames respectively to rank these trackers. Generally, success plot is relatively more impeccable than precision plot.

4.2.1. Results on VOT2014

There are 38 trackers tested on 25 sequences in the VOT2014 competition. Ranking results of each tracker on VOT2014 are summarized in Table 1 and visualized by the AR rank plots [34] shown in Fig. 4. Specially, four correlation filter-based trackers including KCF [23], DSST [24], SAMF [38] and our tracker (named as GACF tracker) are indicated by *italic* in Table 1. The AR rank plots (tested on baseline and region_noise experiments) are visualized results in accuracy-robustness rank space, in which each tracker is marked a point. A tracker is better if it resides closer to the top-right corner of the plot.

From these ranking results, in terms of accuracy criterion, four correlation filter-based trackers show a superior performance than other methods. Despite that KCF tracker ranks the first averaged on both experiments, the other three trackers are followed narrowly. In terms of robustness criterion, PLT_13 and PLT_14 trackers, extended of Struck [39], provide a robust tracking performance on both two experiments. Our tracker also shows a comparable performance on robustness criterion compared with other correlation filter-based trackers.

In sum, benefited from correlation filter, the proposed tracker achieves appealing performances in both accuracy and robustness. According to the average ranks, our method ranks the first due to its robustness performance than KCF, SAMF and DSST. Since DSST is the winner of the VOT2014 challenge, the comparison with it can validate the performance of our tracker to a large extent. The good performance of our tracker on robust criterion is mainly determined by reliable patches selection scheme. PSR is introduced into the verification of the target's center location in our method. The whole patch is regarded as unreliable, the center location can be estimated by other stable patches. It means that our tracker has a better location ability and is not relatively easier to lose to locate the target compared to DSST.

4.2.2. Results on OTB

OTB includes 29 trackers and 51 sequences. Besides, two additional trackers based on correlation filter, KCF and DSST, are

Table 1
Ranking results. The top, second and third lowest average ranks are shown in **bold**, *italic* and ***bolditalic*** respectively. The **Ranking** column displays a joined ranking for both experiments, which are also used to order the trackers.

	Baseline		**Region_noise**		**Overall**		
Trackers	**Accuracy**	**Robustness**	**Accuracy**	**Robustness**	**Accuracy**	**Robustness**	**Ranking**
SIR_PF	12.80	14.20	10.56	16.36	11.68	15.28	13.48
ABS	11.76	10.08	9.64	8.56	10.70	9.32	10.01
qwsEDFT	8.96	12.08	9.52	14.28	9.24	13.18	11.21
EDFT	10.92	15.40	11.16	17.04	11.04	16.22	13.63
aStruck	12.96	12.44	12.88	14.44	12.92	13.44	13.18
IIVTv2	15.16	18.16	16.96	16.56	16.06	17.36	16.71
VTDMG	10.16	12.80	9.20	10.24	9.68	11.52	10.60
MCT	10.08	8.20	8.20	7.40	9.14	7.80	8.47
SAMF	***4.48***	8.56	**4.12**	8.36	***4.30***	8.46	*6.38*
LT_FLO	10.48	20.56	9.96	19.48	10.22	20.02	15.12
Matrioska	13.48	13.12	11.76	16.68	12.62	14.90	13.76
BDF	12.76	12.52	12.76	11.08	12.76	11.80	12.28
MatFlow	12.64	***5.20***	10.20	9.32	11.42	7.26	9.34
PLT_13	10.00	**4.84**	10.72	***4.88***	10.36	***4.86***	7.61
IMPNCC	15.80	22.84	19.60	20.88	17.70	21.86	19.78
Struck	10.68	14.56	11.00	12.88	10.84	13.72	12.28
ThunderStruck	11.56	15.76	10.96	12.56	11.26	14.16	12.71
IPRT	15.56	14.36	14.64	14.56	15.10	14.46	14.78
PLT_14	9.12	**4.84**	8.76	**4.52**	8.94	**4.68**	6.81
ACAT	8.16	10.96	8.72	9.40	8.44	10.18	9.31
eASMS	9.04	9.32	6.84	9.32	7.94	9.32	8.63
FoT	12.40	19.16	13.32	21.32	12.86	20.24	16.55
HMMTxD	5.76	11.28	*4.48*	10.96	5.12	11.12	8.12
ACT	10.12	10.96	10.08	10.60	10.10	10.78	10.44
DSST	5.12	8.12	***4.28***	7.28	*4.70*	7.70	***6.20***
DynMS	12.28	11.28	11.88	12.52	12.08	11.90	11.99
PTp	21.00	12.04	16.60	11.84	18.80	11.94	15.37
KCF	**3.68**	8.56	4.84	8.76	**4.26**	8.66	6.46
FSDT	14.80	22.84	14.88	18.88	14.84	20.86	17.85
OGT	10.12	18.00	10.12	18.28	10.12	18.14	14.13
DGT	8.92	*5.68*	6.20	6.80	7.56	*6.24*	6.9
CMT	13.84	16.44	17.00	16.04	15.42	16.24	15.83
LGTv1	15.60	9.04	14.08	6.56	14.84	7.80	11.32
IVT	14.56	20.60	18.36	18.84	16.46	19.72	18.09
NCC	12.56	30.56	12.96	31.04	12.76	30.80	21.78
FRT	14.12	26.96	16.96	26.32	15.54	26.64	21.09
MIL	22.84	15.92	28.48	16.56	25.66	16.24	20.95
CT	17.44	19.36	18.28	18.16	17.86	18.76	18.31
GACF	*5.10*	7.52	4.92	*5.86*	5.01	6.69	**5.85**

introduced into comparisons. We show the overall performance of One Pass Evaluation (OPE) for our tracker and compare it with some other state-of-the-arts (ranked within top 10) as shown in Fig. 5. The top 5 trackers about success rate include DSST [24], KCF [23], SCM [8], Struck [39] and our tracker. Our tracker ranks the first on success plot and precision plot. Specially, on success plot, the proposed tracker is with nearly 5% improvements compared to KCF.

4.3. Attribute based performance analysis

4.3.1. Results on VOT2014

Per-visual-attribute ranking plots for the baseline experiment are shown in Fig. 6. In VOT2014 challenge, it offers six attributes including *camera motion*, *illumination change*, *motion change*, *occlusion*, *size change* and *no degradation*. Each frame of every sequence is labelled with different attributes, which provides more clear per-attribute analysis than OTB [34]. The improvement of our tracker over the other three correlation filter-based tracker is most apparent at *size change* and *motion change* attributes. On *occlusion* attribute, the proposed tracker performs as well as DSST while our method does not achieve satisfactory performance on *camera motion* and *illumination change* compared to KCF, SAMF and DSST.

4.3.2. Results on OTB

Each of the 51 benchmark sequences is annotated with attributes that indicate what kinds of challenging factors occur within it. There are eleven attribute factors: *Occlusion*, *Illumination Variation*, *Scale Variation*, *Deformation*, *Motion Blur*, *Fast Motion*, *In-Plane Rotation*, *Out-of-Plane Rotation*, *Out-of-View*, *Background Clutters* and *Low Resolution*. Fig. 7 shows five main attributes on success plot and precision plot.

Four attributes are related to affine transformation including *Out-of-Plane Rotation*, *In-Plane Rotation*, *Deformation* and *Scale Variation*. On success plot, the proposed tracker ranks the first on *Out-of-Plane Rotation* and *Deformation* attribute. On *Scale Variation* attribute, our tracker ranks the third followed by DSST and SCM on success plot and the second on precision plot. Compared to DSST with multi-scale searching scheme, our proposed method based on part-based representation does not perform well as expected. But it is still with improvements about 8.4% than KCF tracker. Overall on these four attributes related to affine transformation, the proposed tracker performs well and remains stable and reliable. Besides, it has many improvements compared to KCF tracker to some extent when affine transformation (e.g. scale, rotation, skew) is considered. On *Occlusion* and *Background Clutter* attributes, the proposed method ranks the first owe to part-based representation and updating scheme. It effectively avoids templates contaminated when the target occurs occlusion or drifts.

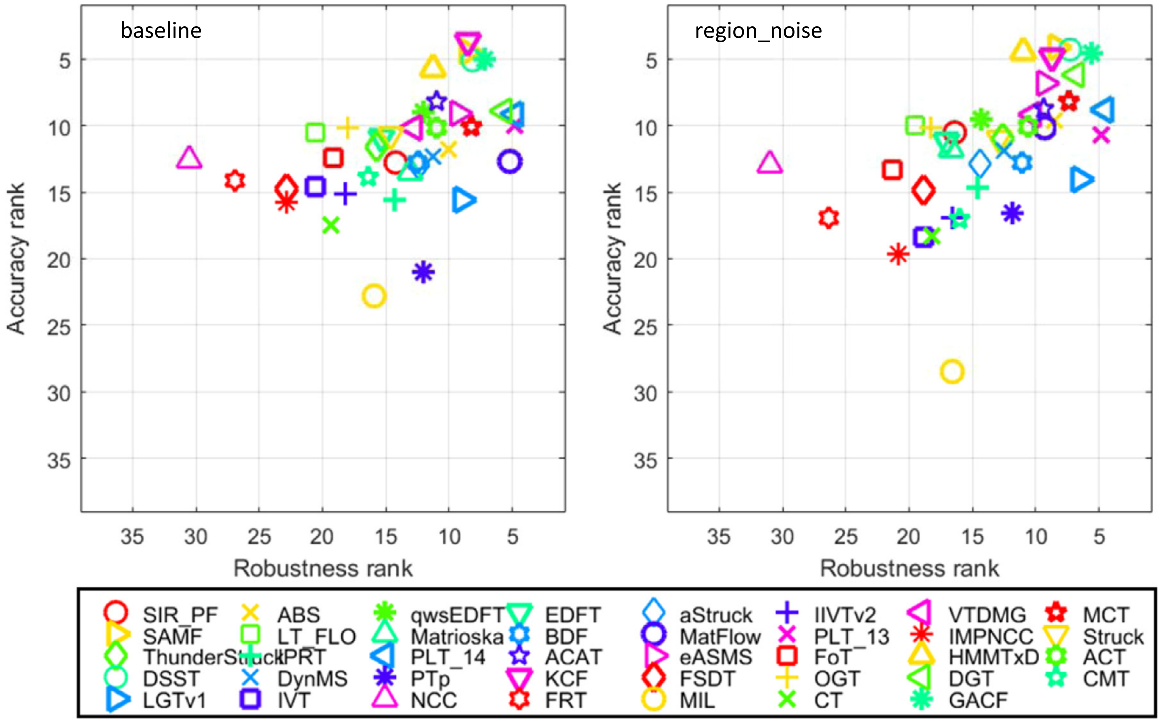


Fig. 4. The accuracy-robustness ranking plots with respect to the two experiments. Tracker is better if it resides closer to the top-right corner of the plot.

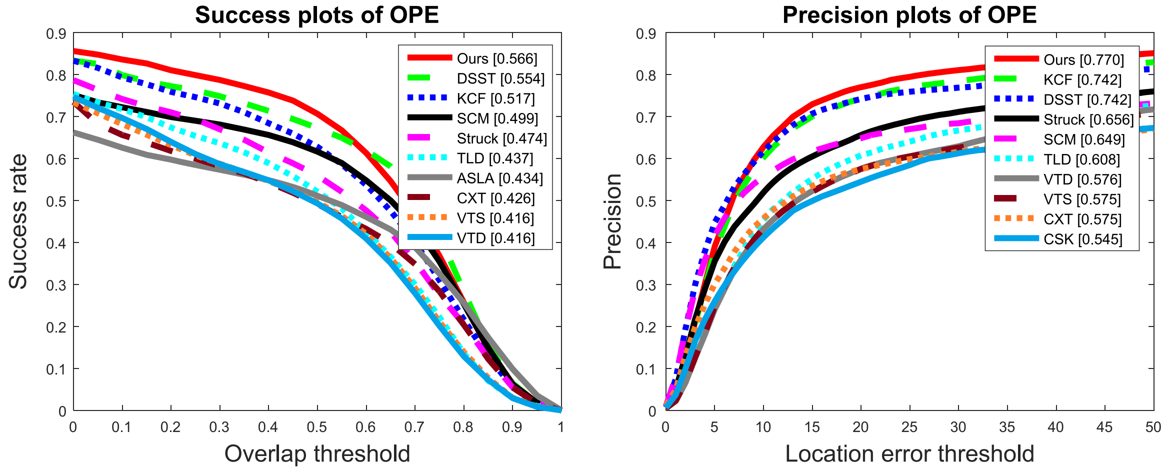


Fig. 5. Plots of OPE. The performance score for each tracker is shown in the legend. For each figure, the top 10 trackers are presented for clarity.

In the above two benchmarks with attribute analysis, results are not mutually exclusive on most attributes except for *size change* or *scale variation*. The main reason is that *size change* is not a vital factor leading to tracking failure compared with other attributes such as *occlusion* and *illumination change*. In most cases, a tracker loses to locate the target due to other attributes despite that the sequence is annotated with *scale variation* in OTB. It still has an unfavorable influence on OPE plots with *scale variation*.

In *occlusion* attribute, our tracker performs better than DSST in terms of accuracy criterion and success plot. Confidence metric including PSR and STAM plays an indispensable role to select more reliable patches. An occluded patch cannot be regarded as a stable patch to estimate the center location and affine transformation of the target. Besides, parameters update scheme prevent an unreliable patch from updating the appearance model, which effectively avoids the appearance model contaminated. These two reasons help the proposed tracker handle occlusions and noises.

4.4. Qualitative analysis

To display tracking performance of our proposed method in an intuitive view, several representative frames from sequences with different attributes are shown in Fig. 8.

Occlusion: Sequences *Tiger1*, *Liquor* contain severe occlusions and in sequences *Carscale* and *David3*, the target suffers relatively slight occlusions. In *Tiger1* sequence, the target is occluded by leaf several times. SCM and Struck definitely lose to locate the target. KCF and our method perform well while DSST fails to achieve promising performance. In *Liquor* sequence, the target suffers severe occlusions, and appearance model of the target is similar to occlusions, leading to difficult to track the object. DSST, Struck and SCM occur drifts to some extent whereas KCF and our method perform superior location ability.

Scale variation: The sequences *Carscale*, *David* and *Dog1* are three representative video sequences with scale variation attribute. Compared to KCF tracker, our proposed method based on

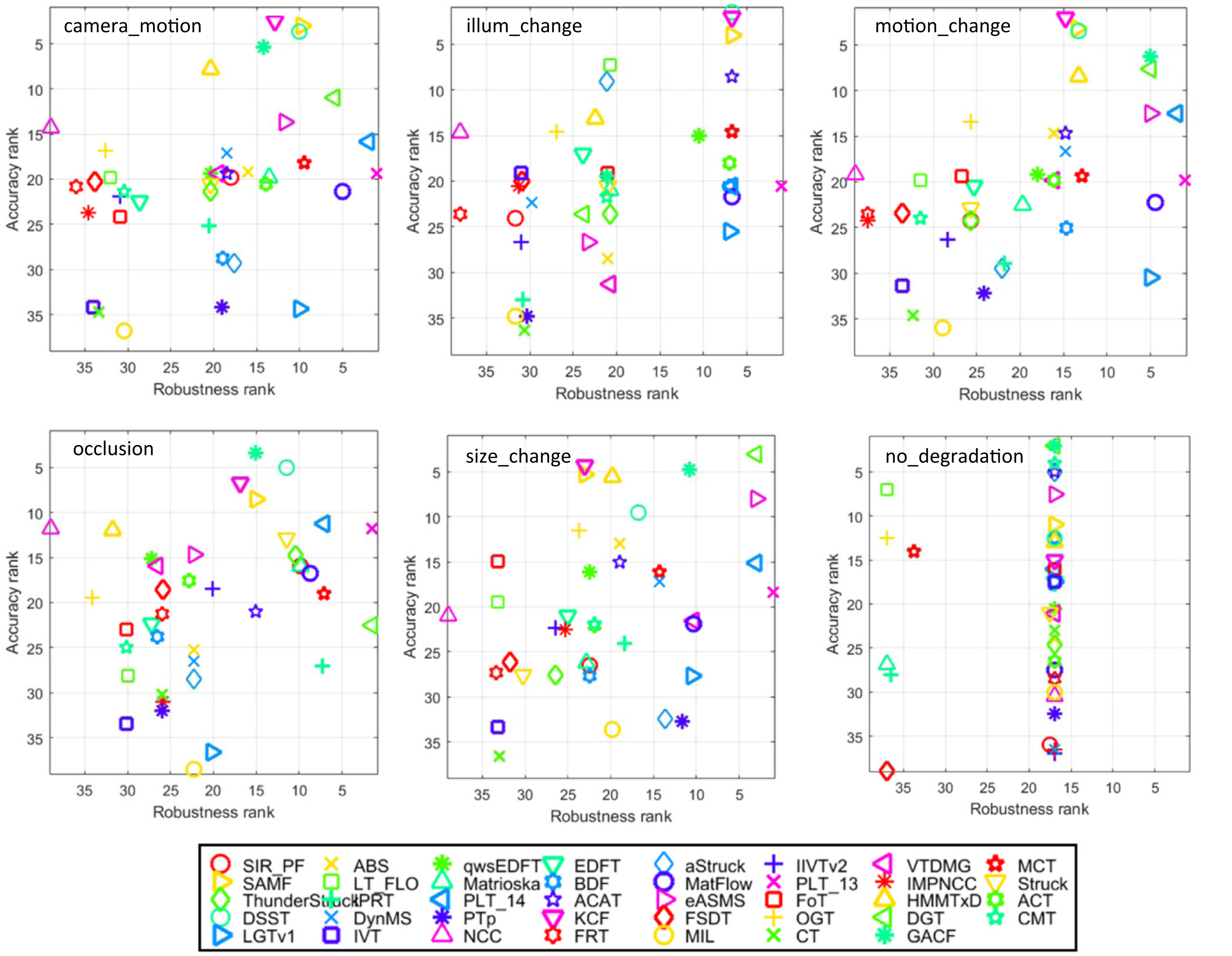


Fig. 6. The accuracy-robustness ranking plots of baseline experiment with respect to the six sequence attributes. The tracker is better if it resides closer to the top-right corner of the plot.

part-based representation can effectively tackle scale variation issues more than favorable center location performance. In these three sequences, especially *Carscale* sequence, performance of our tracker is not inferior than DSST.

Rotation: Rotation of the target is cast into two categories: in-plane rotation and out-of-plane rotation. In-plane rotation is a typical transformation in 2D affine transformation. We give some picture descriptions about the proposed method in *Faceocc2* and *Freeman1* sequences. In these sequences, our method can accurately capture the target's rotation. However, KCF, DSST and Struck only use top-left vertex coordinates, weight and height of tracking bounding box. These methods do not tackle rotation transformation. It is necessary to point out that SCM based on particle filter framework easily solves these scale variations, shape deformation

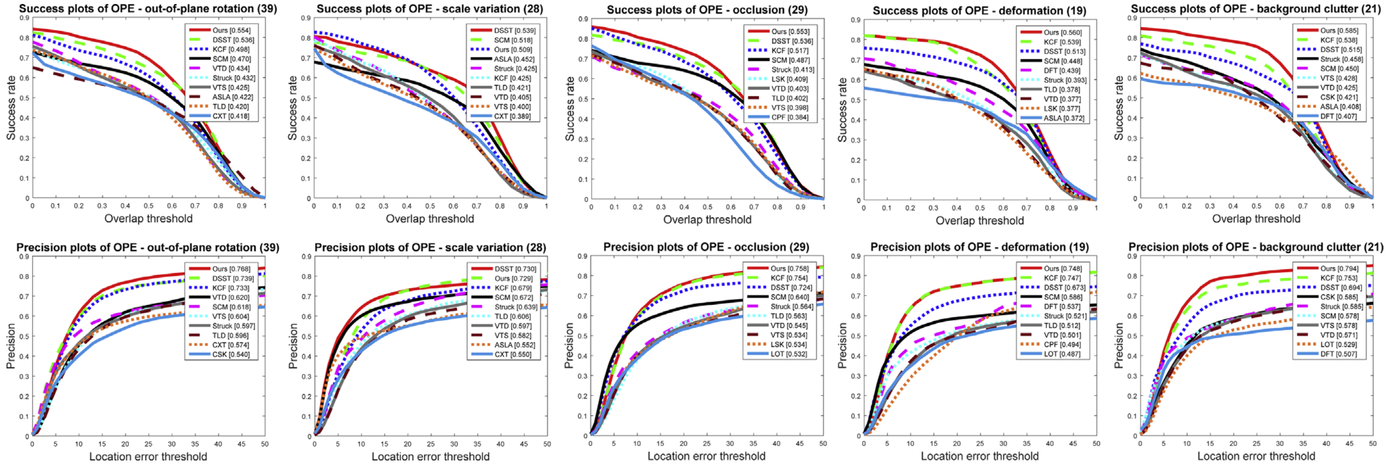


Fig. 7. Success plots and precision plots of OPE on five attributes.

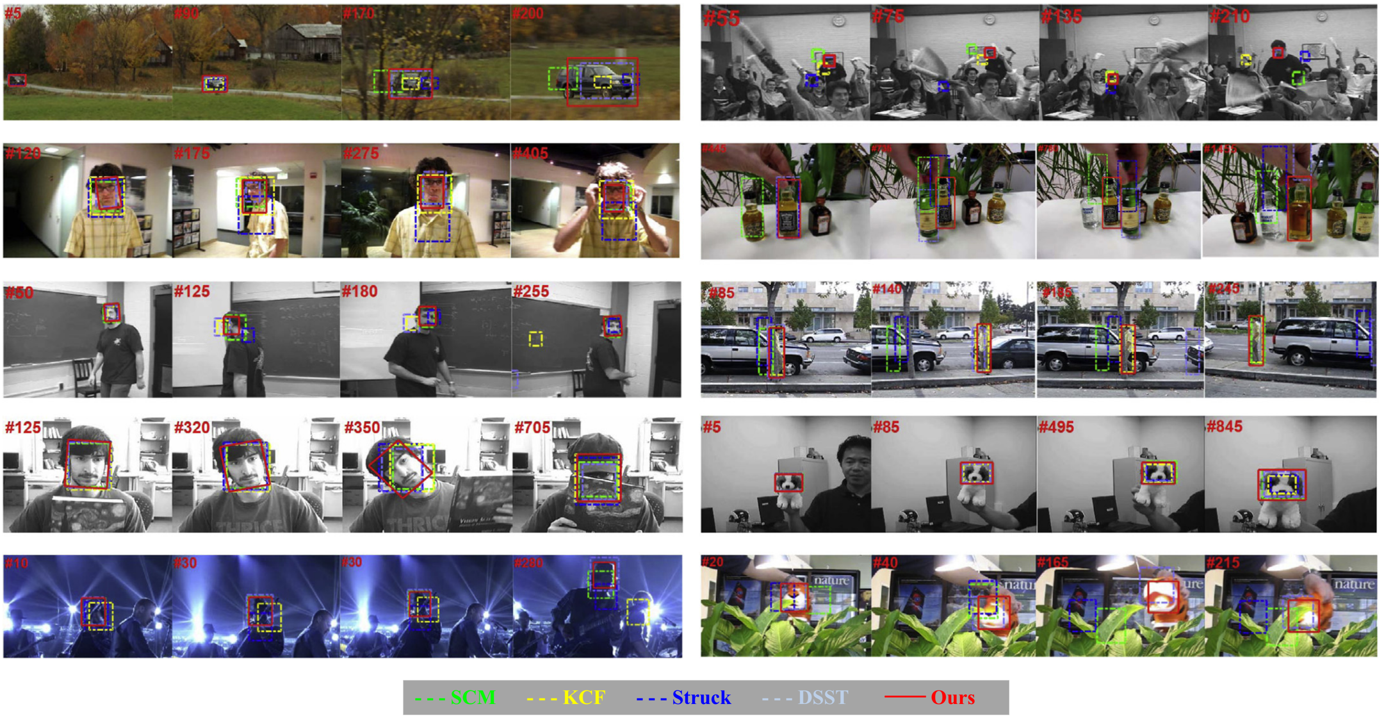


Fig. 8. Tracking results from challenging frames compared with the top 5 trackers. And subfigures in the first row is *Carscle* and *Freeman4*; the second row is *David* and *Liquor*; the third row is *Freeman1* and *David3*; the fourth row is *Faceocc2* and *Dog1*; the last row is *Shaking* and *Tiger1*.

problems. But its tracking results are shown in rectangle tracking box in OTB just for a uniform display framework.

It is relatively more intractable to take out-of-plane rotation attribute into consideration due to huge difference on the target compared with that in previous frame. *Freeman1* and *David3* contain out-of-plane rotation when these two men turn around. At #125, #180 frames in sequence *Freeman1*, and #140, #189 frames in *David3* sequence, most methods are easily confused. They readily lose to locate the target such as SCM, Struck and DSST. Our method still performs stable and accurate.

Deformation: Performances of many trackers are limited to this challenging factor. Appearance model changes dramatically during shape deformation. For example, in *Tiger1* sequence, many trackers do not perform well. On the other hand, our proposed method and DSST still track the target but DSST tracker loses accuracy to locate the tiger.

Background clutter: It is difficult to locate the target precisely in center location under a noisy background. Trackers easily fail to distinguish the target from the background clutter, and lead to drifts. In *Freeman4* sequence, the object not only lies in a noisy background but also occurs heavy occlusions. Only the proposed method tracks the target well and does not reduce tracking precision. The other methods occur drifts to some extent.

Illumination variation: Drastic change of illumination makes the sequence intractable to track the target. Appearance model changes dramatically during illumination variation. In *Shaking* sequence, KCF tracker and Struck method lose to accurately capture the target due to illumination variation. Our method can successfully tail the target throughout entire sequences, which is attributed to the appearance model based on part-based representation with great effect on resisting the light change.

Other attributes: The remaining attributes are all manifest in these sequences to different extents. For example, the *Freeman4* sequence contains *low resolution*. And the *Liquor* sequence contains *fast motion*, accompanied by *motion blur* at the same time. Our proposed method shows a favorable performance on these sequences.

In sum, the proposed tracker shows promising tracking results, and it is able to distinguish the target from their surrounding background with severe occlusions. It has the ability to tackle affine transformations but not limited to only tackle scale variation problem. The test results on OTB with main attributes have shown that the proposed tracker is effective and robust.

4.5. *Key component validation*

In this section, we quantitatively discuss the effects of the Spatial–Temporal Angle Matrix, the whole combination scheme and update scheme. Their respective tracking results are shown in Fig. 9.

4.5.1. *The STAM constraint*

We first introduce a simplest tracker without STAM constraint and the whole combination scheme for various candidates. In this condition, only PSR is imposed for stable patches selection. These stable patches are directly used to estimate an affine matrix, related to only one candidate, as the output tracking result. This tracker named as “No STAM and Combination” achieves the lowest tracking performance, and even worse than KCF tracker on success plot and precision plot. It is easy to find that only PSR constraint could not ensure a patch as reliable.

Based on the above tracker, STAM is introduced to select more stable patches from the perspective of spatial and temporal relationships. This tracker, named as “No combination”, indicates that the only difference between this tracker and the proposed method is that it does not adopt the whole combination scheme to produce numerous candidates. When spatial and temporal relationships are taken into consideration, it achieves a significant increase from 0.493 to 0.543 on success plot. The tracking results mean that with the STAM constraint, the true stable patches are selected to calculate the reliable affine matrix.

4.5.2. *The whole combination*

When the whole combination scheme is incorporated into this tracker, a 2.3% up is to reach 0.566 in the proposed method on

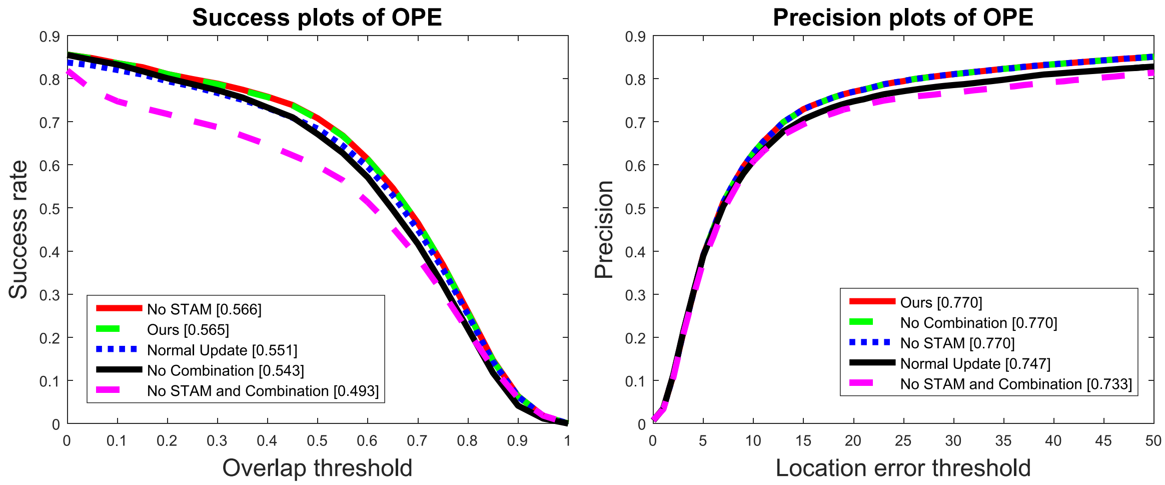


Fig. 9. Success plots and precision plots of key components.

success plot. Success plot on whether STAM or the whole combination scheme exists show effects of these individual components.

Besides, there might be doubtful about how to choose the angle threshold θ_{thres}. And that θ_{thres} is fixed with 15° being rational or not. Therein, we give a test about these patches without STAM, but the whole combination scheme is added. We call this tracker as “No STAM”. It reflects the influence of STAM on the final tracking results under the foundation of the whole scheme. Due to the whole combination scheme and no STAM constraint, the number of candidates is larger than that of the proposed method. We can say the candidate set in the proposed tracker is a subset of that produced in “No STAM” tracker. The optimal candidate in the proposed method must be in the larger candidate set. So the tracking result of “No STAM” tracker must not be weaker than the proposed tracker.

The tracking results show that there is merely no difference between these two trackers based on the whole combination shown in Fig. 9. It indicates that the angle threshold in STAM constraint is proper. If θ_{thres} is too small, few stable patches are reserved and these might not be sufficient to obtain the best candidate to represent the precise affine transformation of the target. If this threshold is set too large, STAM constraint plays a minor role on stable patches selection. In this case, there is no difference on whether STAM constraint is adopted.

The reason why we adopt STAM constraint and set this angle threshold is that the proposed method shows significant speed ups without performance degradation. We will analyze running speed about this in the next section. STAM constraint is necessary to speed-up and tracking performance does not decrease significantly. And this angle threshold is an appropriate value.

4.5.3. Update scheme

Different update schemes can also bring in different tracking performance. Compared to update scheme proposed in Eq. (13), if the conventional update scheme in Eq. (12) is adopted, tracking performance on success plot is decline, down from 0.565 to 0.551. The update scheme has a slightly significance on tracking performance indeed.

4.6. Influence of the number of patches

Different numbers of local patches have various influences on the final tracking result. The smaller local patches would lead to inaccuracy in affine matrix estimation after two confidence metrics. If the number of patches is too large, each patch's size is small and it contains relatively little information. Besides, time consuming rises up due to increase of local patches. We evaluate our proposed tracker with different patches on VOT2014 dataset. Considering the number of patches cannot be arbitrarily assigned, we use conventional partition (i.e., four, nine, sixteen) with no overlapped or overlapped spatial layout shown in Fig. 10. Suppose that the size of the whole patch is $M \times N$, “4(no overlapped)” means that the size of each local patch is $0.5M \times 0.5N$ without no overlapped spatial layout. The local patch is with $0.25M$ step size with horizontal overlapped spatial layout in “6(overlapped)”. Based on “6(overlapped)”, the proposed method utilizes “9(overlapped)” added in vertical direction with $0.25N$ step size. In “16(overlapped)” case, the whole patch is equally divided into 16 local patches without any overlapped. The “12(overlapped)” layout considers overlap in vertical direction with $0.25N$ step size. The overlap ratio and the number of tracking failures for each tracker in the baseline experiment summarized in Table 2.

The results show that if the number of patches is small, only a few reliable patches are preserved to calculate affine transformation. In this case, it cannot achieve a satisfactory estimation. As patches increase, under the foundation of unchange of each part's size, tracking results (including accuracy and robustness) performs

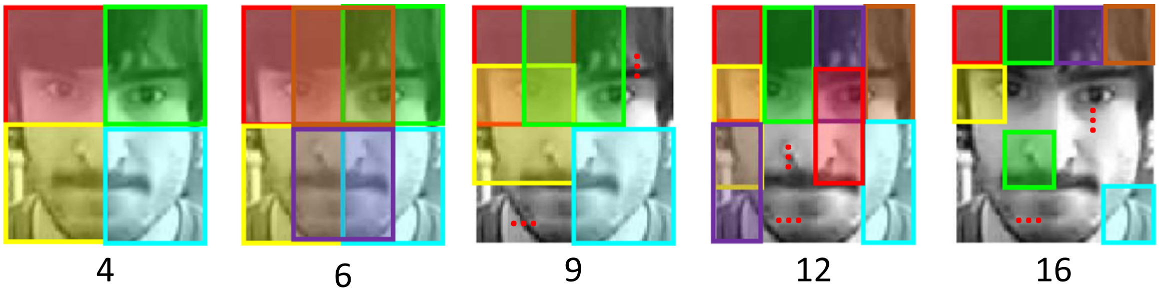


Fig. 10. Geometry configuration for different numbers of parts spatial layout.

Table 2
Performance with different numbers of patches.

Numbers	4(no overlapped)	6(overlapped)	9(overlapped)	12(overlapped)	16(no overlapped)	KCF	DSST
Accuracy	0.543	0.601	0.619	0.613	0.599	0.540	0.622
Robustness	1.71	1.22	1.08	1.04	1.33	1.80	1.16

Table 3
FPS (frames per second) of correlation filter trackers.

Tracker	KCF	DSST	Ours	No STAM
FPS	118.57	21.04	36.28	24.91

better. When the number of patches is from 9 to 12, the size of each local patch is $0.5M \times 0.25N$, which leads to increase in robustness and decline in accuracy.

When the target is divided into 16 patches, each part is too small to reflect more adequate information and it is easy to lose accuracy and robustness to locate the target with a dramatic declines.

4.7. Speed analysis

In this section, we give a description about running time of three trackers based on correlation filter in MATLAB version without further optimization, including KCF, DSST and our proposed tracker in Table 3. Specially, the running time of "No STAM" tracker is also provided.

Compared to KCF tracker, time expenses of our proposed method concentrate on these several steps. First, nine part-based patches are tacked by KCF tracker; second, to search nearest neighbor also spends some time. The size of each local patch is quarter of that of the target. Supposed that the input image is $N \times N$, the time complexity rises from $O(N^2 \log N)$ to $O\left(\frac{25}{16}N^2 \log N\right)$. For DSST tracker, multi-scale searching scheme is indeed time-consuming. It adopts 33 scale level in each frame to search for the best scale factor. Its time complexity is about $O(33N^2 \log N)$. Despite that these time computational complexities are all in the sane order magnitudes, DSST is relatively time-consuming.

Compared to DSST, the extra time consumption mainly costs on searching nearest neighbor. The time complexity of nearest neighbor algorithm is about $O(K^2)$, where K is the number of samples. In our tracker, K often ranges from 2^4 to 2^6 (not too large) in a statistic result. In general, our proposed method is nearly twice times faster than DSST tracker.

The last column of Table 3 shows FPS of "No STAM" tracker. The proposed tracker adopts relatively fewer patches to construct the candidate set. Fewer patches lead to fewer candidates, and the running time of searching nearest neighbor will also rise up. In "No STAM" tracker, seven, eight or nine local patches produce hundreds of candidates. While in the proposed tracker, the number of the remainder stable patches K often ranges from 2^4 to 2^6 aforementioned. The number of candidates falls dramatically from several hundreds to about dozens. This selection would lead to much simplification on searching nearest neighbor among these fewer neighbors. From tracking results on the above success plot, compared to "No STAM", the proposed method shows significant speed ups without performance degradation.

5. Conclusion

This paper proposes a part-based representation tracker based on correlation filter. It can effectively tackle affine transformation and achieve a comparable tracking performance compared to other trackers based on correlation filters. The two selection criteria including PSR and STAM are effective to obtain stable patches. The whole combination scheme boards the diversity of affine matrices and related candidates, which plays an important role on performance improvements. Encouraging empirical performance from our extensive experiments by comparing with several state-of-the-art trackers have demonstrated the effectiveness and robustness of the proposed tracker.

Acknowledgment

This research is partly supported by NSFC, China (No. 61273258) and 863 Plan, China (No. 2015AA042308).

References

[1] Y. Wu, J. Lim, M.-H. Yang, Object tracking benchmark, IEEE Trans. Pattern Anal. Mach. Intell. 38 (4) (2015) 1.

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

[3] J. Gao, H. Ling, W. Hu, J. Xing, Transfer learning based visual tracking with Gaussian process regression, in: Proceedings of European Conference on Computer Vision. Zurich: Springer, 2014:188–203.

[4] Y. Sui, Y. Tang, L. Zhang, Discriminative low-rank tracking, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, 3002–3010.

[5] D.A. Ross, J. Lim, R.-s. Lin, M.-h. Yang, Incremental learning for robust visual tracking, Int. J. Comput. Vision 77 (1–3) (2008) 125–141.

[6] D. Wang, H. Lu, M.H. Yang, Online object tracking with sparse prototypes, IEEE Trans. Image Process. 22 (1) (2013) 314–325.

[7] X. Mei, H. Ling, Robust visual tracking and vehicle classification via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (11) (2011) 2259–2272, http://dx.doi.org/10.1109/TPAMI.2011.66.

[8] W. Zhong, H. Lu, M.H. Yang, Robust object tracking via sparse collaborative appearance model, IEEE Trans. Image Process. 23 (2014) 2356–2368.

[9] L. Ma, X. Zhang, W. Hu, J. Xing, J. Lu, J. Zhou, Local subspace collaborative tracking, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, 4301–4309.

[10] X. Yun, Z.-L. Jing, Kernel joint visual tracking and recognition based on structured sparse representation, Neurocomputing 193 (2016) 181–192.

[11] B. Babenko, M.-H. Yang, S. Belongie, Robust object tracking with online multiple instance learning, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1619–1632.

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

[13] D. Chen, Z. Yuan, G. Hua, Y. Wu, N. Zheng, Description-discrimination collaborative tracking, in: Computer Vision – ECCV 2014, 345–360.

[14] T. Zhou, X. He, K. Xie, K. Fu, J. Zhang, J. Yang, Robust visual tracking via efficient manifold ranking with low-dimensional compressive features, Pattern Recognit. 48 (8) (2015) 2459–2473.

[15] X. Yang, M. Wang, D. Tao, Robust visual tracking via multi-graph ranking, Neurocomputing 159 (2015) 35–43, http://dx.doi.org/10.1016/j.neucom.2015.02.046 URL: ⟨http://linkinghub.elsevier.com/retrieve/pii/S0925231215002064⟩.

[16] Y. Wu, M. Pei, M. Yang, J. Yuan, Y. Jia, Robust discriminative tracking via landmark-based label propagation, IEEE Trans. Image Process. 24 (5) (2015) 1510–1523.

[17] L. Wang, W. Ouyang, X. Wang, H. Lu, Visual tracking with fully convolutional networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 3119–3127.

[18] G. Wu, W. Lu, G. Gao, C. Zhao, J. Liu, Regional deep learning model for visual tracking, Neurocomputing 175 (2016) 310–323.

[19] Z. Chen, Z. Hong, D. Tao, An Experimental Survey on Correlation Filter-based Tracking, arXiv:1509.05520, 2015.

[20] K. Zhang, L. Zhang, Q. Liu, D. Zhang, M.-H. Yang, Fast visual tracking via dense spatio-temporal context learning, in: Computer Vision – ECCV 2014, Zurich, Springer, 2014 127–141.

[21] M. Danelljan, H. Gustav, F.S. Khan, M. Felsberg, Learning spatially regularized correlation filters for visual tracking, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, 4310–4318.

[22] D. Bolme, J. Beveridge, B. Draper et al. Visual object tracking using adaptive correlation filters, in: Proceedings of Computer Vision and Pattern Recognition (CVPR), 2010, 2544-2550.

[23] 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.

[24] M. Danelljan, G. Häger, F. Khan, M. Felsberg, Accurate scale estimation for robust visual tracking, in: British Machine Vision Conference, Nottingham, September, BMVA Press, 2014.

[25] A. Adam, E. Rivlin, I. Shimshoni, Robust fragments-based tracking using the integral histogram, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 2006, 798–805.

[26] R. Yao, Q. Shi, C. Shen, Y. Zhang, A. Hengel, Part-based visual tracking with online latent structural learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, 2363–2370.

[27] Y. Li, J. Zhu, S.C. Hoi, Reliable patch trackers: robust visual tracking by exploiting reliable patches, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, 353–361.

[28] L. Cehovin, M. Kristan, A. Leonardis, Robust visual tracking using an adaptive coupled-layer visual model, IEEE Trans. Pattern Anal. Mach. Intell. 35 (4) (2012) 1, http://dx.doi.org/10.1109/TPAMI.2012.145.

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

[30] L. Zhang, D. Bi, Y. Zha, S. Gao, H. Wang, T. Ku, Robust and fast visual tracking via spatial kernel phase correlation filter, Neurocomputing. http://dx.doi.org/10.1016/j.neucom.2015.10.131.

[31] S. Hare, A. Saffari, P.H. Torr, Efficient online structured output learning for keypoint-based object tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, 2012, 1894–1901.

[32] W. Bouachir, G.-A. Bilodeau, Collaborative part-based tracking using salient local predictors, Comput. Vis. Image Underst. 137 (2015) 88–101.

[33] M. Savvides, B. Kumar, P.K. Khosla, Cancelable biometric filters for face recognition, in: Proceedings of the 17th International Conference on Pattern Recognition, 2004, ICPR 2004, Cambridge, 922–925.

[34] A. Leonardis, Matej Kristan, Roman Pflugfelder, J. Matas, L. Čehovin, Georg Nebehay, et al., The visual object tracking vot2014 challenge results, in: Computer Vision – ECCV 2014 Workshops, 2014.

[35] Y. Wu, J. Lim, M.H. Yang, Online object tracking: a benchmark, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, 2411–2418.

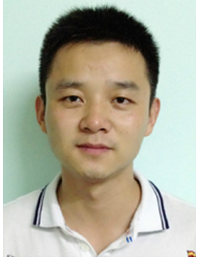
[36] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The Pascal visual object classes (voc) challenge, Int. J. Comput. Vis. 88 (2) (2010) 303–338.

[37] M. Kristan, J. Matas, A. Leonardis, T. Vojir, R. Pflugfelder, G. Fernandez, G. Nebehay, F. Porikli, L. Cehovin, A novel performance evaluation methodology for single-target trackers, IEEE Trans. Pattern Anal. Mach. Intell, http://dx.doi.org/10.1109/TPAMI.2016.2516982.

[38] Y. Li, J. Zhu, A scale adaptive kernel correlation filter tracker with feature integration, in: Computer Vision – ECCV 2014 Workshops, Zurich, Springer, 2014, pp. 254–265.

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

Fanghui Liu received the B.S. degree from Harbin Institute of Technology, China, in 2014. He is currently pursuing the Ph.D. degree at the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, under the supervision of Prof. Jie Yang. His research areas mainly include visual tracking, subspace clustering and Bayesian learning.



Tao Zhou received the M.S. degree in computer application technology from Jiangnan University, Wuxi, China, in 2012. Currently, he is pursuing the Ph.D. degree at the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China. His current research interests include object detection, visual tracking and machine learning.

Jie Yang received his Ph.D. from the Department of Computer Science, Hamburg University, Germany, in 1994. Currently, he is a professor at the Institute of Image Processing and Pattern recognition, Shanghai Jiao Tong University, China. He has led many research projects (e.g., National Science Foundation, 863 National High Tech. Plan), had one book published in Germany, and authored more than 200 journal papers. His major research interests are object detection and recognition, data fusion and data mining, and medical image processing.