Pattern Recognition Letters 84 (2016) 163–169

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Robust visual tracking via constrained correlation filter coding☆

Fanghui Liu[a], Tao Zhou[a], Keren Fu[a,b], Jie Yang[a,*]

[a] Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai 200240, China
[b] Department of Signals and Systems, Chalmers University of Technology, Gothenburg 41296, Sweden

ARTICLE INFO

Article history:
Received 7 April 2016
Available online 28 September 2016

Keywords:
Visual tracking
Constrained correlation filter
Supervised feature coding
Discriminative model

ABSTRACT

Unconstrained correlation filters based trackers achieve superior performance with high speed in visual tracking. However, such unconstrained correlation filters do not impose any hard constraint to their responses to have a certain value, which brings about classification ambiguity on intractable samples (i.e., two similar samples from different classes). To tackle this issue, in this paper, constrained correlation filter is introduced into visual tracking framework to alleviate classification ambiguity for more accurate target location. By imposing distinguishable hard constraints on the response map to different classes, a supervised coding method is proposed to encode various candidate samples by a discriminative filter bank. The learned high-level feature vectors are sent to a Naive Bayes classifier to separate the target from the background. Besides, parameters updating schemes in the constrained filter and classifier are introduced to adapt to appearance changes of the target with less possibility of drifting. Both qualitative and quantitative evaluations on 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 one of the most challenging tasks in computer vision with many real-world applications such as video surveillance, human computer interaction and motion analysis [3,26,27]. The goal of visual tracking task is to track the interested object given its location at the first frame in a video sequence. During tracking process, the target often suffers challenging factors (e.g., partial occlusions, scale variation, shape deformation, variation of illumination), which makes a tracker difficult to accurately locate the target. While much achievements [10,19] have been made in visual tracking during in the past several years, there are still improving space in trackers for more complicated situations.

Generally, tracking methods can be divided into two categories, one is generative method [15,32] and the other is discriminative method [25,29]. Generative methods are often designed to search the most similar candidate to the target with minimizing the reconstruction error, while discriminative methods regard the tracking process as binary classification problem, to separate the target from the background. Such approaches [14,25] rely on rich feature representations to discriminant between the target and background appearance. This paper investigates a discriminative tracker via a supervised feature coding scheme based on correlation filter for visual tracking.

Among these discriminative methods, correlation filters based methods [4,30] have attracted attention because of its favorable target location property (means smaller center location error), and they have achieved competitive results both on accuracy and robustness in visual tracking. Correlation filters (loosely called "template" in some literature, or called "classifier" in visual tracking area) are designed to produce correlation peaks on the target while yielding low response to the background. Moreover, the exhaustive spatial correlation operation can be converted into a dot-production operation in Fourier domain with high speed.

However, there is a main drawback of these unconstrained filters based trackers: no hard constraints are imposed on their responses to have a certain value. In contrast, by the hard constraint in the constrained correlation filter, the correlation peak can be constrained to be $+1$ for the target and -1 for the background (which is far away from the target). Therefore, in testing process, the filter produces a high value (about $+1$) for positive candidates samples (from the target and its neighboring circular area) and a low value (about -1) for the background, which helps separate the target from the background. We give an example shown in Fig. 1 to illustrate the response map difference between constrained and unconstrained correlation filter.

The woman's right eye ("the target") is tracked/located by correlation filter, while her left eye is regarded as an outlier point

☆ This paper has been recommended for acceptance by Prof. S. Sarkar.

* Corresponding author.
E-mail address: jieyang@sjtu.edu.cn (J. Yang).

http://dx.doi.org/10.1016/j.patrec.2016.09.009
0167-8655/

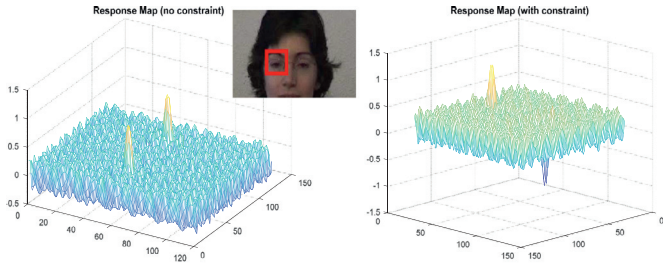

Fig. 1. An illustration of response maps of unconstrained/constrained correlation filter.

which is similar to the target. It seems difficult to distinguish the true and the false from unconstrained correlation filter's response map shown in Fig. 1. Such similar response levels to both target and background easily mislead the subsequent classification, leading to tracking drifts or failure. For constrained correlation filter, in training process, the filter produces high but opposite peaks to these two similar objects with the additional constraint, which can effectively guarantee the response difference to enhance the discriminative ability of target location. Compared to unconstrained correlation filter, constrained filter is easily introduced into pattern classification issue [8,22] because it can produce different discriminative kinds of responses to different patterns.

Motivated by these above issues, we propose a supervised coding method based on constrained correlation filter for visual tracking, in which constrained correlation filter is employed as a feature coder instead of a classifier as usual in unconstrained correlation filter based trackers. To best of our knowledge, constrained correlation filter is the first introduced into the area of visual tracking. The main contributions of this paper are as follows.

(i) The constrained correlation filter based coding method is proposed to produce a discriminative filter bank. It encodes various candidates into corresponding feature vectors by an inner product operation. The resulting high-level feature vectors for candidates are used to separate the target from the background.
(ii) Updating schemes for parameters in the constrained correlation filter and classifier are introduced to adapt to appearance changes of the target with less possibility of drifting.

The remainder of the paper is organized as follows. Section 2 gives the relevant related work of the proposed method. Section 3 demonstrates details of the proposed method. Experimental results on OTB100 and performance evaluation are included in Section 4. Finally, conclusion is drawn in Section 5.

2. Related work

2.1. Correlation filter

Correlation filter has been widely in computer vision, such as image classification [22,23] and face recognition [8,28]. There are two kinds of correlation filter design, one is correlation filter with constraints to the correlation peak; and the other is to remove the correlation filter constraints [24]. The family of constrained correlation filter includes minimum average correlation energy (MACE) filter [21], optimal tradeoff synthetic discriminant function (OTSDF) filter [17] and maximum margin correlation filter (MMCF) [23]. Based on MACE, OTSDF introduces the output noise variance into the objective function to enhance robustness to noise and outliers. MMCF incorporates correlation filter and Support Vector Machine (SVM) [11] into an uniform framework, which can simultaneously localize and classify objects of interest. The typical representative

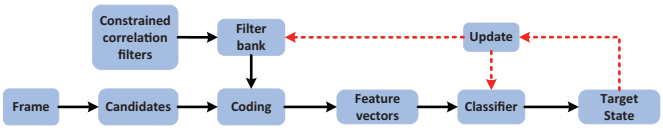

Fig. 2. The flowchart of the proposed method.

unconstrained correlation filters are ASEF [2] and minimum output sum of squared error (MOSSE) filter which is the first correlation filter applied to visual tracking [1].

2.2. Unconstrained correlation filter based tracking method

Recently, unconstrained correlation filters based method show a promising performance on benchmark datasets [22,27]. As pioneered work, MOSSE tracker [1] seeks for a filter 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 determine by the maximum value in correlation responses.

Considering that MOSSE only relies on image intensity information for feature representation, HOG (KCF method) [12], Color Names (ColorTracking method) [7] and channel representations [6] have successfully been employed in unconstrained correlation filter based trackers.

To tackle scale variations issue, DSST [5] method incorporates multi-scale search scheme into KCF method. MUSTER [13] employs Integrated Correlation Filter in short-time tracking. STC [30] adopts an adaptive window size for scale update scheme by adjusting Gaussian weighted function.

Besides, Ma et al. [20] introduce online random fern classifier for long-term tracking. In Li et al. [18], part-based strategy is introduced into correlation filter for scale estimation issue. Different from these existing unconstrained correlation filter based trackers, we introduce the constrained correlation filter into a supervised coding scheme for visual tracking.

3. Proposed method

The flowchart of the proposed method is shown in Fig. 2. The operations of the proposed model are summarized as follows. Positive samples (from the target and its neighboring circular area) and negative samples (far away from the target) at the beginning frames are used to generate a discriminative filter bank by a constrained correlation filter (OTSDF). When a new frame comes, the filter bank encodes various candidates into various feature vectors. These resulting feature vectors are sent to a Naive Bayes classifier to discriminant the target from the background. Finally, the tracking result is used to update parameters in the filter bank and Naive Bayes classifier.

Specifically, we give a detailed description of candidates sampling procedure. The random sampling method mainly depends on the motion model $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, which denotes state transition between two consecutive frames.

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) \sim \mathcal{N}(\mathbf{x}_{t-1}, \sigma^2) \tag{1}$$

where the state $\mathbf{x}_t = [p_x, p_y, \theta, s, \alpha, \phi]$, the corresponding components represent translation on X, Y direction, rotation angle, scale, aspect ratio, and skew respectively. Each state is associated with a candidate (we denote it by a vectorized image patch). In our experiment, each image patch (or called observation) is normalized to 32×32 pixels.

We also present the positive and negative samples generation procedure. Generally, the tracking result in the first frame is

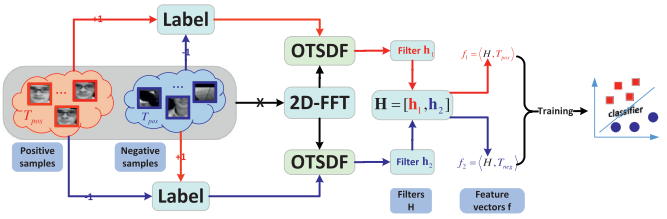

Fig. 3. The framework of constrained correlation filter coding method.

manually chosen as a rectangle box. Define that $\mathcal{I}(x, y)$ is the center of the rectangle box, and the initial positive templates $\mathbf{T}_{pos}$ are sampled by Eq. (1) from an inner circular area that satisfies $\|\mathcal{I}_i - \mathcal{I}(x, y)\| < r$, where $\mathcal{I}_i$ is the center of the ith sampled patch. Similarity, negative templates $\mathbf{T}_{neg}$ are sampled by Eq. (1) from the annular region $r < \|\mathcal{I}_j - \mathcal{I}(x, y)\| < s$, where $\mathcal{I}_j$ is the center of the jth sampled image, r and s are the inner and outer radius of the annular region, respectively.

3.1. Review: OTSDF filter

As a representative constrained correlation filter, OTSDF filter seeks for minimizing the average correlation energy (ACE) and the output noise variance (ONV) with the hard constraints $\mathbf{h}^\dagger\mathbf{x}_i = u_i$, where $\mathbf{x}_i$ represents the vectorized i-th image, $\mathbf{h}\dagger$ represents the conjugate transpose of a filter $\mathbf{h}$, and u_i is the pre-specified peak filter response. It is usually set to $+1$ for a positive sample and -1 for a negative sample. Supposed that the number of training images is N. The average correlation energy is defined as:

$$E_{avg} = \frac{1}{N}E_i = \frac{1}{Nd}\sum_{i=1}^{N}\mathbf{h}^\dagger\mathbf{X}_i\mathbf{X}_i^*\mathbf{h} = \mathbf{h}^\dagger\mathbf{D}\mathbf{h} \tag{2}$$

where $\mathbf{X}_i^*$ is the conjugate of a diagonal matrix $\mathbf{X}_i$, which contains $\mathbf{x}_i$ along its diagonal. The diagonal matrix $\mathbf{D}_i = \mathbf{X}_i\mathbf{X}_i^*$ contains the power spectrum of $\mathbf{x}_i$. And the diagonal matrix $\mathbf{D} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{D}_i$ contains the average power spectral density of all training images.

The average correlation energy is denoted by $E_1 = \mathbf{h}^\dagger\mathbf{D}\mathbf{h}$, and the output noise variance is defined as $E_2 = \mathbf{h}^\dagger\mathbf{P}\mathbf{h}$ where the diagonal matrix $\mathbf{P}$ contains the power spectral density of the input noise (usually approximated by the identity matrix). The objective function of OTSDF filter is formulated as:

$$\min_{\mathbf{h}} E_1 + \beta E_2 \quad s.t. \quad \mathbf{X}^\dagger\mathbf{h} = \mathbf{u}. \tag{3}$$

where β is a regularization parameter. The solution of Eq. (3) is:

$$\mathbf{h} = \mathbf{T}^{-1}\mathbf{X}(\mathbf{X}^\dagger\mathbf{T}^{-1}\mathbf{X})^{-1}\mathbf{u} \tag{4}$$

where $\mathbf{T} = \mathbf{D} + \beta\mathbf{P}$. For $\beta = 0$, OTDSF filter degenerates the MACE filter. The filter $\mathbf{h}$ indicates different responses to the target or the background. In other computer vision applications, the filter $\mathbf{h}$ in OTDSF is employed as a classifier for classification. However, in our method, $\mathbf{h}$ is used for feature coding, which is the most difference between what we use and the conventional OTDSF.

3.2. Constrained correlation filter coding

In our tracking framework, the constrained correlation filter is employed as a feature encoder. The framework of the proposed coding method is shown in Fig. 3.

Two discriminative filters $\mathbf{h}_1$ and $\mathbf{h}_2$ are produced by positive samples and negative samples with different hard constraints via OTSDF.

For filter $\mathbf{h}_1$, these positive training samples $\mathbf{T}_{pos}$ are labeled with $+1$ and negative samples $\mathbf{T}_{neg}$ are with -1. The corresponding hard constraint is defined as $\mathbf{u}_1 = [1, 1, \ldots, 1, -1, -1, \ldots, -1]^\top \in \mathbb{R}^{D_1+D_2}$, where D_1, D_2 are the number of positive training samples and negative training samples, respectively. $\mathbf{h}_1$ is obtained by Eq. (5) with these training samples $\mathbf{X} = [\mathbf{T}_{pos}, \mathbf{T}_{neg}]$ and their corresponding hard constraints $\mathbf{u}_1$.

Different from $\mathbf{h}_1$, the filter $\mathbf{h}_2$ is obtained with the opposite constraint labels. Positive training samples $\mathbf{T}_{pos}$ are labeled with -1 and negative training samples $\mathbf{T}_{neg}$ are with $+1$. The hard constraint $\mathbf{u}_2 = [-1, -1, \ldots, -1, 1, 1, \ldots, 1]^\top \in \mathbb{R}^{D_1+D_2}$ is opposite to what is set in $\mathbf{h}_1$. Filter $\mathbf{h}_2$ is calculated in a similar fashion by Eq. (5):

$$\begin{cases} \min\limits_{\mathbf{h}_1} E_1 + \beta E_2 & \text{s.t. } \mathbf{X}^\dagger\mathbf{h}_1 = \mathbf{u}_1 \\ \min\limits_{\mathbf{h}_2} E_1 + \beta E_2 & \text{s.t. } \mathbf{X}^\dagger\mathbf{h}_2 = \mathbf{u}_2 \end{cases} \tag{5}$$

By imposing different labels to these two filters, filters $\mathbf{h}_1$ and $\mathbf{h}_2$ produce different responses for the target or the background to enhance discriminative ability in visual tracking. Filter $\mathbf{h}_1$ represents that it produces high positive peaks to positive samples and high negative peaks to negative samples, and vice versa for filter $\mathbf{h}_2$. These two filters $\mathbf{h}_1$ and $\mathbf{h}_2$ form a discriminative filter bank $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2]$. The filter bank $\mathbf{H}$ contains two filters whose responses to different classes are opposite.

Constrained correlation filter is designed for each class in training images, and then for each vectorized test sample $\mathbf{y}$, the output feature $\mathbf{f}$ is extracted by the inner products between each class filter $\mathbf{h}_i$ and the test sample $\mathbf{y}$. It is formulated as:

$$\mathbf{f} = \langle \mathbf{H}, \mathbf{y} \rangle = \mathbf{H}^\top\mathbf{y} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_C]^\top\mathbf{y} \tag{6}$$

where C is the number of the classes in the training set. In tracking issue, C is set to 2 just because there are only two classes: the target or the background. For positive samples, their corresponding feature vectors $\mathbf{f}_1 = \mathbf{H}^\top\mathbf{T}_{pos}$. The corresponding feature vectors for negative samples are $\mathbf{f}_2 = \mathbf{H}^\top\mathbf{T}_{neg}$. Feature vectors $\mathbf{f}_1$ and $\mathbf{f}_2$ represent totally opposite responses to the foreground or the background. For each test sample, the corresponding feature vector $\mathbf{f}$ is sent to a Naive Bayes classifier [16] to distinguish the test sample as the target or the background.

With respect to $\mathbf{h}$, it is involved with the matrix inversion operation of $\mathbf{X}^\dagger\mathbf{T}^{-1}\mathbf{X}$. If the number of positive and negative samples ($\mathbf{X}$) is too large, the computational cost is time-consuming. Besides, positive samples are similar to each other, which leads to the invertible matrix problem of $\mathbf{X}$. On the other hand, it is less stable to obtain a filter by fewer samples. To tackle this issue, we choose a circulation operation to produce filters. Ten positive samples and twenty negative samples are equally divided into five groups. Each group contains two positive samples and four negative samples. Each group can generate a filter $\mathbf{h}_1$ for the target and a filter $\mathbf{h}_2$ for the background. In this case, five filters for the positive and five filters for the negative are produced to construct a filter bank $\mathbf{H}$ within ten filters. For each candidate $\mathbf{y}$, its corresponding feature vector is defined as $\mathbf{f} = \mathbf{H}^\top\mathbf{y} \in \mathbb{R}^{10}$.

3.3. Naive bayes classifier construction and updating

The rationale assumption of Naive Bayes classifier is that features are independent given class. In our method, a Naive Bayes classifier is constructed similar with CT tracker [31]. For each feature vector $\mathbf{f}$:

$$F(\mathbf{f}) = \log\left(\frac{\prod_{i=1}^{n} p(f_i|c=1)p(c=1)}{\prod_{i=1}^{n} p(f_i|c=-1)p(c=-1)}\right) \tag{7}$$

where $F(\cdot)$ is a classifier discriminant function, f_i is i-th element in $\mathbf{f}$, where n is the dimension of feature vector $\mathbf{f}$. The binary variable $c \in \{-1, 1\}$ indicates the candidate label, and the prior is assumed to uniform prior $p(c = 1) = p(c = -1) = 1/2$.

By assuming that the conditional distributions $p(f_i|c=1)$ and $p(f_i|c=-1)$ are subjected to Gaussian distribution with four parameters $(\mu_i^+, \sigma_i^+, \mu_i^-, \sigma_i^-)$, where μ_i^+ (μ_i^-) and σ_i^+ (σ_i^-) are mean and standard derivation of the positive (negative) class, respectively. The corresponding optimal candidate $\mathbf{y}^*$ is obtained by maximum a posteriori probability (MAP) in Eq. (7), which has the highest classification confidence value $F(\mathbf{f}*)$.

It is necessary to update parameters in Naive Bayes classifier because the target and the background at each frame all have influence on the distribution of features. The incremental update scheme for these four parameters is the same with CT method [31],

$$\begin{aligned}\mu_i^+ &\leftarrow \lambda\mu_i^+ + (1-\lambda)\mu^+ \\ \sigma_i^+ &\leftarrow \sqrt{\lambda(\sigma_i^+)^2 + (1-\lambda)(\sigma^+)^2 + \lambda(1-\lambda)(\mu_i^+ - \mu^+)^2}\end{aligned} \tag{8}$$

The incremental update scheme for μ_i^- and σ_i^- is with similar fashion in Eq. (8).

3.4. Filter bank update schemes

Model update is essential in tracking process because appearance model often changes due to illumination variation and pose changes. In our tracking framework, filters $\mathbf{h}_1$ and $\mathbf{h}_2$ are used to represent the target in different cues.

To update these two filters, there are a little differences in filter update scheme between $\mathbf{h}_1$ and $\mathbf{h}_2$. When the tracking result $\mathbf{y}^*$ obtained, it can be also used to produce a new filter $\mathbf{h}_1^*$ by Eq. (5). The update scheme for $\mathbf{h}_1$ is followed as:

$$\mathbf{h}_1^t = (1-\gamma)\mathbf{h}_1^{t-1} + \gamma\mathbf{h}_1^* \tag{9}$$

For $\mathbf{h}_2$, it is related to the background. In the proposed method, the negative samples are updated every 5 frames to obtain a new filter $\mathbf{h}_2$ by Eq. (5).

3.5. Target location revision

Statistical results illustrate that value of $F(\mathbf{f})$ roughly ranges from -100 to 100 by adjusting a scale parameter. We find that if the classification confidence $F(\mathbf{y}^*)$ of the optimal candidate typically exceeds 20, the tracking result is highly reliable. When the target occurs occlusions, the highest classification confidence would decline slightly to some extent. If $F(\mathbf{y}^*)$ is smaller than zero, it indicates that the tracking result is unreliable in high probability. In this case, to avoid tracking drifts, we use KCF tracker to revise the location of the target in the current frame. The remainder affine parameters (such as scale factor, rotation angle and skew factor) are all the same with those in the previous frame. The algorithm of the proposed method is summarized as follows.

4. Experimental results

In this section, we give details of our experimental implementation and discuss the results of tracking performance evaluated on Object Tracking Benchmark (OTB-100) [27] with 29 trackers and 100 sequences.

Correlation filter based methods MUSTER [13], DSST [5], KCF [12], MOSSE [1], STC [30], and ColorTracking [7] are introduced into comparisons. Besides, a recent state-of-the-art tracker MEEM [29] is also added for comparisons.

4.1. Experimental setup

All experiments were conducted on a regular PC in MATLAB with Intel i5-6500 CPU (3.20 GHz) and 8 GB memory without further optimization. The following parameters were used for our tests: The regularization parameter in OTSDF filter was set to $\beta = 0.1$. The update parameters λ and γ were fixed with 0.85 and 0.05, respectively.

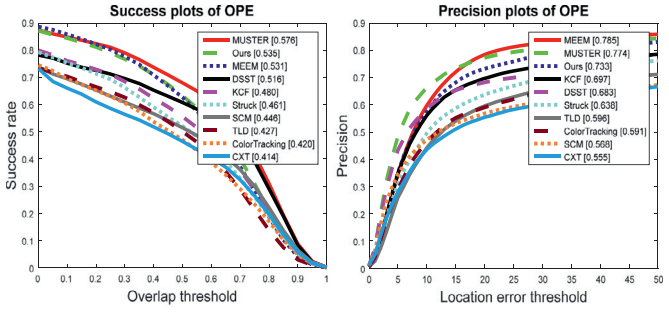

Fig. 4. Plots of OPE on 100 sequences. The performance score for each tracker is shown in the legend. For each figure, the top 10 trackers are presented for clarity.

4.2. Results on OTB

4.2.1. Evaluation metrics

Two typical evaluation criteria are used in our experiments. The first criterion is mean Center Location Error (CLE), which is the pixel distance between the centroid of the tracking result and the ground truth. The other is the Pascal VOC Overlap Ratio (VOR) [9]. 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.

Based on these two evaluation metrics, a frame whose VOR is larger than a threshold (ranges from 0 to 1) is termed as a successful frame, and the ratio of successful frames at this threshold is plotted in success plot. Precision plot is with similar fashion, a frame whose CLE is smaller than a threshold (pixel distance ranges from 0 to 50) is termed as a successful frame, the percentage of successfully tracked frames at a this pixel distance threshold is plotted in precision plot.

In success plot, Area Under Curve (AUC) is used for ranking these trackers. In precision plot, the precisions at 20 pixel distance threshold are used for ranking.[1] These two plots are quantitative analysis results which reflect target location and accuracy ability under a series of thresholds. Generally, success plot is relatively more impeccable than precision plot.

4.2.2. Overall performance

We show the overall performance of One pass evalution (OPE) for our tracker and compare it with some other state-of-the-arts (ranked within top 10) as shown in Fig. 4.

The top 5 trackers about success rate include MUSTER [13], MEEM [29], DSST [5], KCF [12] and our method. Our tracker ranks the second on success plot and the third on precision plot. It shows a favorable location ability and a superior overlap rate performance compared to other methods. With target location revision by KCF tracker, the proposed method is with 5.5% improvements than KCF tracker on success plot.

Considering that only top 10 methods are presented for clarity, we also compare these representative methods in Table 1.

In terms of average CLE evaluation and average VOR criterion, MUSTER, MEEM and our method are the top 3 methods in OTB50 and OTB100. The results show a consistent conclusion as the same as shown in success plot and precision plot.

[1] In success plots and precision plots, plot styles are with fixed ranking order instead of trackers.

Table 1
Average Center Location Errors (CLE), Average VOC Overlap Ratio (VOR) and Average Speeds (FPS) are presented. In each entry, score in the top comes from OTB with 51 tasks, while the score in the bottom comes from OTB100. The first, second and third best scores are highlighted by **bold**, underline and *italic*.

Method	Ours	MUSTER	MEEM	DSST	KCF	ASLA	Struck	TLD	CSK	MOSSE	STC	CN
CLE	*31.2*	**17.3**	22	41.3	35.5	73.1	50.6	48.1	88.8	82.8	68.3	64.8
	35.9	31	**28.8**	50.8	45.3	75.5	49.6	54.9	306	99.6	80.4	81.8
VOR(%)	55.2	**65.0**	57	*56.1*	51.9	43.9	47.8	44.1	40.1	31.8	35.1	44.8
	53	**58.3**	*53.6*	52.2	48.0	40.6	45.9	42.4	38.4	29.1	31.2	42.4
FPS	24.2	2.28	9.75	26.5	94.3	3.82	–[1]	–	*168*	312	**344**	60.5
	24.8	2.16	10.1	25.9	90.7	3.54	–	–	*164*	304	**341**	59.4

[1] In average fps, all methods are implemented with MATLAB except for Struck and TLD in C/C++. We omit these two methods just for fair comparisons.

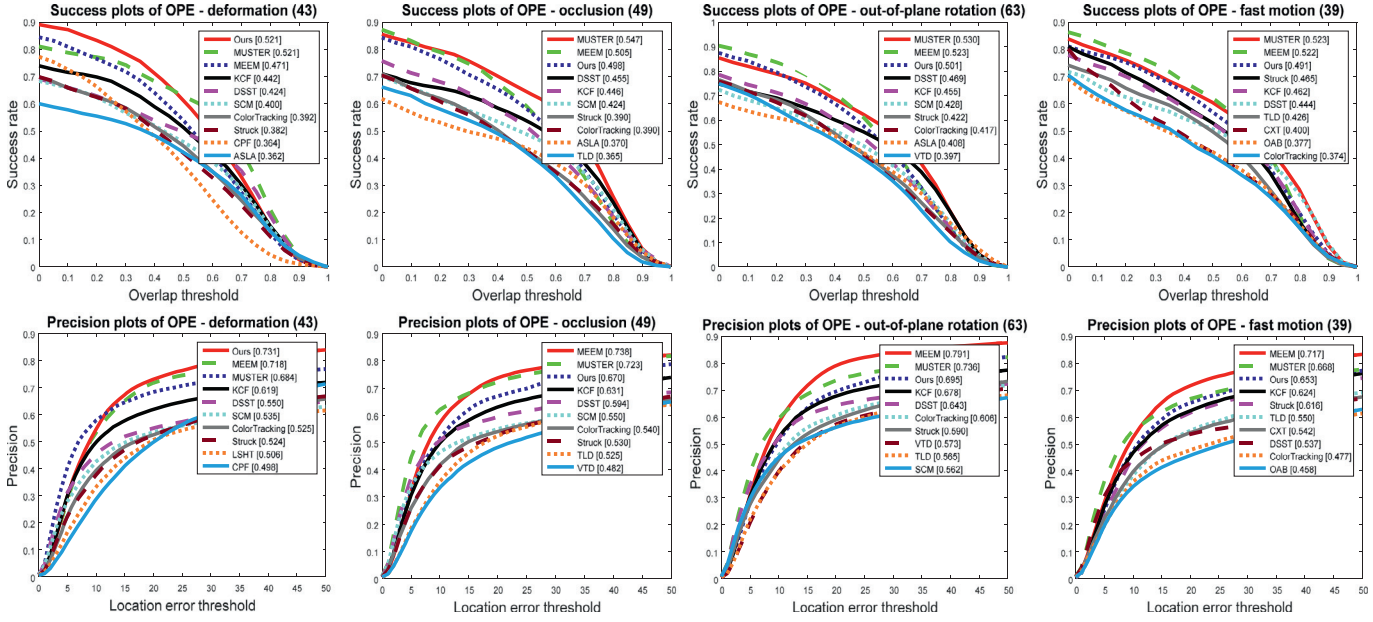

Fig. 5. Success plots and precision plots of OPE on four main attributes.

4.2.3. Running time

The frame per second (fps) about some representative methods are shown in Table 1. The top three fast tracker are STC, MOSSE and CSK respectively. The running speed of the proposed method is about 25 fps while the baseline tracker (KCF) is with almost 90 fps. The main computation cost spends on the filter bank obtained and updated, with respect to solve a quadratic programming problem with hard constraints.

4.2.4. Attribute based performance analysis

Each sequence in OTB is annotated with eleven attributes that indicate what kinds of challenging factors occur within it. In Fig. 5, we give the top 10 trackers on success plots and precision plots of four main attributes including *out-of-plane rotation*, *occlusion*,[2] *deformation* and *fast motion*. In terms of success plots and precision plots, the proposed method ranks the first in *deformation* attribute and the third in the remaining attributes.

4.3. Qualitative analysis

To display tracking performance of our proposed method in an intuitive view, several representative frames from fifty sequences with different attributes in OTB100 are shown in Fig. 6 . The top five trackers including MUSTER, MEEM, DSST and KCF are shown in comparisons.

Occlusion: When the target suffers partial/total occlusions, appearance changes easily lead to tracking failure or track with degraded accuracy. *Tiger2* is a typical video sequence with partial occlusions on the target. When this toy tiger suffers partial occlusions by the leaves, DSST and KCF completely fail to locate the target at #110. MUSTER tracker and MEEM method show a slight fluctuation on tracking results in term of overlap accuracy. With respect to target location performance, the proposed method shows a promising performance on this sequence. In *Jogging.1* sequence, at #80 frame, the target is entirely occluded by a telegraph pole, only our method can still capture the target without any performance degradation.

Scale variation and rotation: These two attributes are respect to affine transformation issues. Compared to KCF tracker based on correlation filters, the proposed method is relatively easier to tackle this problem due to sampling schemes. In *Carscale* sequence, this sequence is a representative video with *scale variation* attribute. Besides, this car is also partially occluded by the branches, which also leads difficult in target location. At frame #170, only our method can cover the car in terms of overlap rate to some relatively more accurate extent. In *Toy* sequence, **scale variation**, **in-plane rotation** and **out-of-plane rotation** have been reflected on this toy. Due to appearance changes, MEEM cannot capture the target accurately. DSST and KCF show tracking degradation on this target to some extent.

[2] In OTB, occlusion attribute includes partial occlusion and total occlusion without being distinguished in attribute analysis.

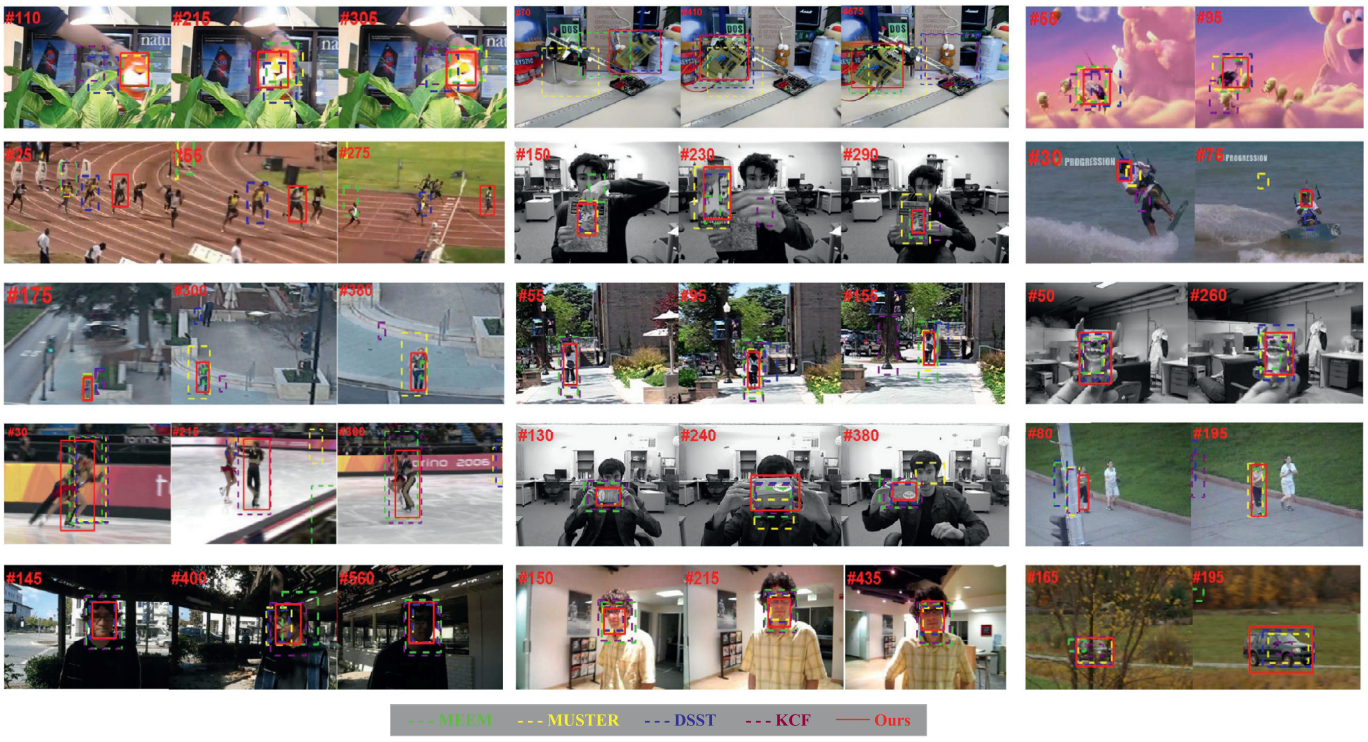

Fig. 6. Tracking results from challenging frames compared with the top 5 trackers. In the first row, subfigures from left to right are from sequences *Tiger2*, *Board* and *Bird2* respectively; The second row represents *Bolt2*, *ClifBar* and *KiteSurf*; The third row denotes *Human6*, *Human7* and *Toy*; The fourth row is with respect to *Skating2.2*, *Twinnings* and *Jogging.1*; The last row represents *Trellis*, *David* and *Carscale*.

Algorithm 1: Algorithm for the proposed constrained filter coding tracker.

for $t = 1$ *to* m **do**
  Use a simple tracker;
  Extract samples $\mathbf{T}_{pos}$, $\mathbf{T}_{neg}$ in tth frame by Eq. (1);
end
Obtain the filter bank $\mathbf{H}$ by Eq. (5);
Encode these samples by Eq. (6) and then train a Naive Bayes classifier.
for $t = m + 1$ *to the end of the sequence* **do**
  S candidates $\mathbf{Y}_{1:S}$ are sampled by Eq. (1) in the motion model;
  Obtain encoding vectors by Eq. (6): $\mathbf{f} = \mathbf{H}^{\top}\mathbf{y}$;
  Classify $\mathbf{f}$ to choose the optimal candidate $\mathbf{y}^*$ by Eq. (7);
  if *the highest classification confidence value* $F(\mathbf{y}^*)$ *is smaller than zero* **then**
    Target Location Revision by KCF tracker;
  end
  Update: Recalculate the filter bank $\mathbf{H}$ by Eqs. (9) and (5);Parameters updating in Naive Bayes classifier by Eq. (8);
end

Deformation: Appearance model often changes a lot due to this attribute. In *Skating2.2* sequence, MUSTER and DSST either fails to track or tracks with degraded accuracy after the targets' pose changes. Despite that KCF tracker achieves a good location performance, it cannot deal with scale variation in this sequence. The proposed method performs well without drifting whereas other methods do not. In *Bolt2* sequence, all methods lose tracking accuracy to some extent when the player suffers pose variation except the proposed method, and they hardly obtain accurate appearance model.

Fast motion: This attribute is always accompanied by **motion blur**. It is difficult to accurately predict the location of the target when abrupt motion occurs, and the appearance changes due to motion blur posed some challenges for accurately locating the target. In *Human7* sequence, KCF and DSST trackers lose to locate the target gradually. And then MEEM method cannot capture scale variations of the target in the following frames. During the whole video sequence, MUSTER and the proposed method provides stable tracking results. The proposed method is more roust than other methods in *Bird2*, *Tiger2* sequences.

There are also other attributes we do not refer to just because of the limited space. For example, **background clutter** manifests on *Board* sequence. On *Trellis* and *David* sequences, tracking performance is mainly effected by **illumination variation** attribute. Drastic change of illumination makes it intractable, accompanied with partial occlusions.

In sum, the proposed method benefits from the fact that the final tracking results are largely dependent on the discriminative high-level feature vectors via a supervised coding scheme. Besides, sampling schemes help our method tractable to solve affine transformation issues compared to KCF tracker.

4.4. Self verification

In this section, we compare with our tracker without target location revision. Besides, different parameters in our method are also taken into consideration shown in Table 2.

Considering target location revision, our method is with 9.0% improvements compared to the one without KCF tracker (for target location revision) on average VOC overlap ratio. The performance improvements also demonstrate that the classification confidence value can be regarded as an indicator to show whether the tar-

Table 2
Average Center Location Errors (CLE) and Average Overlap Scores (OS) are presented from OTB with 51 tasks.

Method	CLE	VOR(%)
Ours ($\beta = 0.1$, $\gamma = 0.05$ and $\lambda = 0.85$)	31.2	55.2
Without revision	47.6	46.2
$\beta = 0.05$	34.8	52.4
Ours ($\beta = 0.1$)	**31.2**	**55.2**
$\beta = 0.2$	39.7	51.2
$\beta = 0.5$	45.9	48.8
$\gamma = 0.01$	33.4	53.9
$\gamma = 0.02$	32.7	54.6
Ours ($\gamma = 0.05$)	**31.2**	**55.2**
$\gamma = 0.1$	35.1	52.0
$\lambda = 0.5$	44.3	49.1
$\lambda = 0.75$	34.5	53.1
Ours($\lambda = 0.85$)	**31.2**	**55.2**
$\lambda = 0.95$	38.7	51.7

get occurs drifts. The results also demonstrate that feature vectors encoded by our method performs better than other correlation filter based trackers with grayscale feature and color feature due to these more discriminative encoding vectors.

There are three main parameters in our method, related to β (the output noise variance), λ (classifier update) and γ (filter bank update). The selection of λ is the same with that in CF tracker. These two parameters are tradeoff between the former information preservation and the current state update. For β, it is a regularization term to consider the output noise influence. If β becomes larger, average correlation energy would be depressed to some extent, which leads to target location ability decline.

Overall the proposed method with constrained correlation filter coding shows a favorable performance compared to other correlation filter methods.

5. Conclusion

This paper proposes a constrained correlation filter coding tracker based on correlation theory. The proposed method with random sampling methods not only effectively tackles affine transformation issues and but also provides more robust supervised feature coding vectors compared to unconstrained correlation filter trackers. The high level feature vectors are good for separating the target from the background. Experimental results on the benchmark demonstrate that our proposed tracker achieves a favorable performance compared to other state-of-the-art methods.

Acknowledgement

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

References

[1] D.S. Bolme, J.R. Beveridge, B.A. Draper, Y.M. Lui, Visual object tracking using adaptive correlation filters, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 2544–2550.
[2] D.S. Bolme, B.A. Draper, J.R. Beveridge, Average of synthetic exact filters, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 2105–2112.
[3] D. Chen, Z. Yuan, G. Hua, Y. Wu, N. Zheng, Description-discrimination collaborative tracking, in: Proceedings of European Conference on Computer Vision, 2014, pp. 345–360.
[4] Z. Chen, Z. Hong, D. Tao, An experimental survey on correlation filter based tracking, 2015. arXiv preprint arXiv:1509.05520.
[5] M. Danelljan, G. Häger, F. Khan, M. Felsberg, Accurate scale estimation for robust visual tracking, in: Proceedings of British Machine Vision Conference, 2014.
[6] M. Danelljan, G. Häger, F.S. Khan, M. Felsberg, Coloring channel representations for visual tracking, in: Proceedings of Scandinavian Conference on Image Analysis, Springer, 2015, pp. 117–129.
[7] M. Danelljan, F. Shahbaz Khan, M. Felsberg, J. Van de Weijer, Adaptive color attributes for real-time visual tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1090–1097.
[8] V.H. Diaz-Ramirez, O.G. Campos-Trujillo, V. Kober, P.M. Aguilar-Gonzalez, Multiclass pattern recognition using adaptive correlation filters with complex constraints, Opt. Eng. 51 (3) (2012) 037203–037203–12.
[9] 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.
[10] 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, 2014, pp. 188–203.
[11] S. Hare, A. Saffari, P.H.S. Torr, Struck: structured output tracking with kernels, in: Proceedings of the IEEE International Conference on Computer Vision, 2011, pp. 263–270.
[12] 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.
[13] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, D. Tao, Multi-store tracker (muster): a cognitive psychology inspired approach to object tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 749–758.
[14] M.D. Jenkins, P. Barrie, T. Buggy, G. Morison, Extended fast compressive tracking with weighted multi-frame template matching for fast motion tracking, Pattern Recognit. Lett. 69 (2016) 82–87.
[15] X. Jia, H. Lu, M.H. Yang, Visual tracking via adaptive structural local sparse appearance model, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1822–1829.
[16] A. Jordan, On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes, in: Proceedings of Advances in Neural Information Processing Systems, vol. 14, 2002, p. 841.
[17] B. Kumar, A. Mahalanobis, D.W. Carlson, Optimal trade-off synthetic discriminant function filters for arbitrary devices, Opt. Lett. 19 (19) (1994) 1556–1558.
[18] 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, pp. 353–361.
[19] C. Ma, C. Liu, F. Peng, J. Liu, Multi-feature hashing tracking, Pattern Recognit. Lett. 69 (2016) 62–71.
[20] C. Ma, X. Yang, C. Zhang, M.-H. Yang, Long-term correlation tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5388–5396.
[21] A. Mahalanobis, B. Vijaya Kumar, D. Casasent, Minimum average correlation energy filters, Appl. Opt. 26 (17) (1987) 3633–3640.
[22] A.L. Matej Kristan, R. Pflugfelder, J. Matas, L. Čehovin, E.A. Georg Nebehay, The visual object tracking VOT 2014 challenge results, in: Proceedings of European Conference on Computer Vision Workshops, 2014.
[23] A. Rodriguez, V.N. Boddeti, B. Kumar, A. Mahalanobis, Maximum margin correlation filter: a new approach for localization and classification, IEEE Trans. Image Process. 22 (2) (2013) 631–643.
[24] B.V.K. Vijaya Kumar, J.a. Fernandez, A. Rodriguez, V.N. Boddeti, Recent advances in correlation filter theory and application, Opt. Pattern Recognit. XXV 9094 (2014) 909404.
[25] 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.
[26] 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, 2015, pp. 3101–3109.
[27] Y. Wu, J. Lim, M.-H. Yang, Object tracking benchmark, IEEE Trans. Pattern Anal. Mach. Intell. 37 (9) (2015) 1834–1848.
[28] Y. Yan, H. Wang, D. Suter, Multi-subregion based correlation filter bank for robust face recognition, Pattern Recognit. 47 (11) (2014) 3487–3501.
[29] J. Zhang, S. Ma, S. Sclaroff, MEEM: robust tracking via multiple experts using entropy minimization, in: Proceedings of European Conference on Computer Vision, 2014, pp. 188–203.
[30] K. Zhang, L. Zhang, Q. Liu, D. Zhang, M.-H. Yang, Fast visual tracking via dense spatio-temporal context learning, in: Proceedings of European Conference on Computer Vision, 2014, pp. 127–141.
[31] K. Zhang, L. Zhang, M.-H. Yang, Fast compressive tracking, IEEE Trans. Pattern Anal. Mach. Intell 36 (10) (2014) 2002–2015.
[32] T. Zhang, S. Liu, N. Ahuja, M.-H. Yang, B. Ghanem, Robust visual tracking via consistent low-rank sparse learning, Int. J. Comput. Vis. 111 (2) (2015) 171–190.