High-speed Tracking with Multi-kernel Correlation Filters

Ming Tang[1]*, Bin Yu[1], Fan Zhang[2], and Jinqiao Wang[1]
[1]National Lab of Pattern Recognition, Institute of Automation, CAS, Beijing 100190, China
[2]School of Info. & Comm. Eng., Beijing University of Posts and Telecommunications

Abstract

Correlation filter (CF) based trackers are currently ranked top in terms of their performances. Nevertheless, only some of them, such as KCF [26] and MKCF [48], are able to exploit the powerful discriminability of non-linear kernels. Although MKCF achieves more powerful discriminability than KCF through introducing multi-kernel learning (MKL) into KCF, its improvement over KCF is quite limited and its computational burden increases significantly in comparison with KCF. In this paper, we will introduce the MKL into KCF in a different way than MKCF. We reformulate the MKL version of CF objective function with its upper bound, alleviating the negative mutual interference of different kernels significantly. Our novel MKCF tracker, MKCFup, outperforms KCF and MKCF with large margins and can still work at very high fps. Extensive experiments on public data sets show that our method is superior to state-of-the-art algorithms for target objects of small move at very high speed.

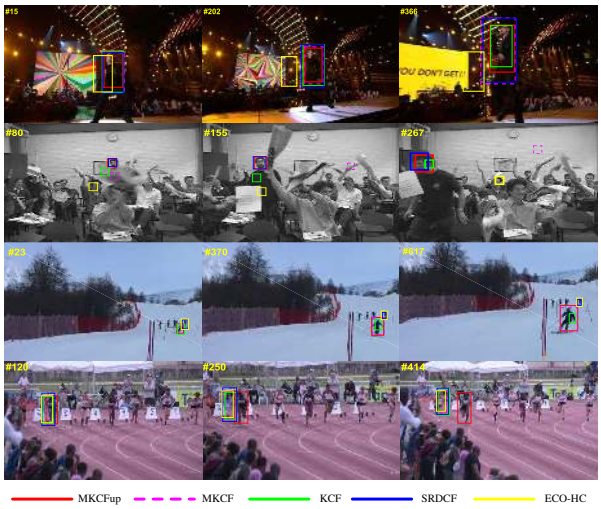

Figure 1. Qualitative comparison of our novel multi-kernel correlation filters tracker, MKCFup, with state-of-the-art trackers, KCF [26], MKCF [48], SRDCF [12], and ECO_HC [9] on challenging sequences, singer2 and freeman4 of OTB2013 [55] and ski_long and running_100_m_2 of NfS [17].

1. Introduction

Visual object tracking is one of the most challenging problems in computer vision [49, 28, 32, 42, 35, 39, 36, 38, 29, 59, 57, 23, 50, 6, 46]. To adapt to unpredictable variations of object appearance and background during tracking, the tracker could select a single strong feature that is robust to any variation. However, this strategy has been known to be difficult [51, 20], especially for a model-free tracking task in which no prior knowledge about the target object is known except for the initial frame. Therefore, designing an effective and efficient scheme to combine several complementary features for tracking is a reasonable alternative [54, 56, 33, 16, 1, 53, 60, 58].

Since 2010, correlation filter based trackers (CF trackers) have been being proposed and almost dominated the tracking domain in recent years [4, 25, 16, 10, 26, 13, 15, 5, 8, 43, 14, 41, 37]. Bolme *et al.* [4] reignited the interests in correlation filters in the vision community by proposing a CF tracker, called minimum output sum of squared error (MOSSE), with classical signal processing techniques. MOSSE used a base image patch and several virtual ones to train the correlation filter directly in the Fourier domain, achieving top accuracy and fps then. Later, the expression of MOSSE in the spatial domain turned out to be the ridge regression [45] with a linear kernel [25]. Therefore, in order to exploit the powerful discriminability of non-linear kernels, Henriques *et al.* [25, 26] utilized the circulant structure produced by a base sample to propose an efficient kernelized correlation filter based tracker (KCF). Danelljan *et al.* [16] extended the KCF with the historically weighted objective function and low-dimensional adaptive color channels. To adaptively employ complementary features in KCF, Tang and Feng [48] derived a multi-kernel

*The corresponding author (tangm@nlpr.ia.ac.cn). This work was supported by Natural Science Foundation of China under Grants 61375035 and 61772527. The code is available at http://www.nlpr.ia.ac.cn/mtang/Publications.htm.

learning (MKL) [44] based correlation filter (MKCF) which is able to take advantage of the invariance-discriminative power spectrums of various features [51] to improve the location performance. By introducing a mask on the samples into the loss item of correlation filter formulation, Galoogani *et al*. [19] proposed the correlation filter with limited boundaries (CFLB) to address the boundary effect [31]. And Danelljan *et al*. [12] introduced a smooth spatial regularization factor within the regularizer to restrain the boundary effect. In [9], Danelljan *et al*. employed the dimensionality reduction, linear weighting of features, and sample clustering to further improve the SRDCF proposed in [12] in both location accuracy and fps.

Up till now, there are at least two principal lines to improve MOSSE and KCF. The first one is to weight the filter or samples with a mask in MOSSE or the KCF of linear kernel, alleviating the negative boundary effect greatly and improving the location performance remarkably. However, the trackers on this line, such as CFLB, SRDCF, C-COT [15], and ECO_HC [9], are unable to employ powerful non-linear kernels. And the other line is to improve the objective function of KCF, such as designing more complicated objective functions [2], or introducing the MKL into KCF to adaptively exploit multiple (non-linear) kernels. Although MKCF, the MKL version of KCF, is more discriminative than KCF, its improvement over KCF is quite limited because different kernels of MKCF may restrict each other in training and updating. And unfortunately, the computational cost of MKCF increases significantly in comparison to KCF. Specifically, the MKCF's improvement over KCF on AUC is only about $2\% \sim 3\%$, while its fps drops dramatically from averagely about 300 of KCF to 30. It is noticed that such an improvement of introducing MKL into KCF is similar to that of introducing MKL into single kernel binary classifier [51], where the improvement of MKL version is about 2%.

In this paper, we will introduce the MKL into KCF in a different way than [48] to adaptively exploit multiple complementary features and non-linear kernels more effectively than in MKCF. We reformulate the MKL version of CF objective function with its upper bound, alleviating the negative mutual interference of complementary features significantly while keeping very large fps. In fact, our novel MKCF tracker, *i.e*., MKCFup, outperforms KCF and the KCF with scaling on AUC about 16% and 7%, respectively, at about 150 fps. A qualitative comparison shown in Fig. 1 indicates that our novel tracker, MKCFup, outperforms other state-of-the-art trackers in challenging sequences singer2 and freeman4 of OTB2013 [55] and ski_long and running_100_m_2 of NfS [17].

The remainder of this paper is organized as follows. In Sec.2, we briefly overview the related work. Sec.3 first simplifies the solution of MKCF, then analyzes its shortcoming, and finally derives a novel multi-kernel correlation filter with the upper bound of objective function. Sec.4 provides some necessary implementation details. Experimental results and comparison with state-of-the-art approaches are presented in Sec.5. Sec.6 summarizes our work.

2. Related Work

Multi-kernel learning (MKL) aims at simultaneously learning a kernel and the associated predictor in supervised learning settings. Rakotomamonjy *et al.* [44] proposed an efficient algorithm, named SimpleMKL, for solving the MKL problem through reduced gradient descent in a primal formulation. Varma and Ray [51] extended the MKL formulation in [44] by introducing an additional constraint on combinational coefficients and applied it to object classification. Vedaldi *et al*. [52] and Gehler and Nowozin [20] applied MKL based approaches to object detection and classification. Cortes *et al*. [7] studied the problem of learning kernels of the same family with an L_2 regularization for ridge regression (RR) [45]. Tang and Feng [48] extended the MKL formulation of [44] to RR, and presented a different multi-kernel RR approach. In this paper, differently from all above approaches, we derive a novel multi-kernel correlation filter through optimizing the upper bound of multi-kernel version of KCF's objective function.

In addition to the correlation filter based trackers aforementioned, generalizations of KCF to other applications have also been proposed [3, 18, 24] in recent years. And Henriques *et al*. [27] utilized the circulant structure of Gram matrix to speed up the training of pose detectors in the Fourier domain. It is noted that all these approaches are unable to employ multiple kernels or non-linear kernels simultaneously. In this paper, we propose a novel multi-kernel correlation filter which is able to fully take advantage of invariance-discriminative power spectrums of various features at really high speed.

3. Multi-kernel Correlation Filters with Upper Bound

In this section, we will first review the multi-kernel correlation filter (MKCF) [48], simplify its optimization, then analyze its drawback, and finally derive a novel multi-kernel correlation filter with upper bound. Readers may refer to [44, 21] for more details on multi-kernel learning.

3.1. Simplified Multi-kernel Correlation Filter

The goal of a ridge regression [45] is to solve the Tikhonov regularization problem,

$$\min_{f} \frac{1}{2} \sum_{i=0}^{l-1} (f(\mathbf{x}_i) - y_i)^2 + \lambda ||f||_k^2, \tag{1}$$

where l is the number of samples, f lies in a bounded convex subset of an RKHS defined by a positive definite kernel function $k(,)$, $\mathbf{x}_i$s and y_is are the samples and their regression targets, respectively, and $\lambda \geq 0$ is the regularization parameter.

As a special case of ridge regression, correlation filters generate their training set $\{\mathbf{x}_i | i = 0, \ldots, l-1\}$ by cyclically shifting a base sample, $\mathbf{x} \in \mathbb{R}^l$, such that $\mathbf{x}_i = \mathbf{P}_l^i \mathbf{x}$, where $\mathbf{P}_l$ is the permutation matrix of $l \times l$ [26], and the y_is are often Gaussian labels.

By means of the Representer Theorem [47], the optimal solution f^* to Problem (1) can be expressed as $f^*(\mathbf{x}) = \sum_{i=0}^{l-1} \alpha_i k(\mathbf{x}_i, \mathbf{x})$. Then, $||f||_k^2 = \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}$, where $\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \ldots, \alpha_{l-1})^\top$, and $\mathbf{K}$ is the positive semi-definite kernel matrix with $\kappa_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ as its elements, and Problem (1) becomes

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^l} \frac{1}{2} ||\mathbf{y} - \mathbf{K}\boldsymbol{\alpha}||_2^2 + \frac{\lambda}{2} \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} \tag{2}$$

for $\boldsymbol{\alpha}$, where $\mathbf{y} = (y_0, y_1, \ldots, y_{l-1})^\top$.

It has been shown that using multiple kernels instead of a single one can improve the discriminability [34, 51]. Given the base kernels, k_m, where $m = 1, 2, \ldots, M$, a usual approach is to consider $k(\mathbf{x}_i, \mathbf{x}_j)$ to be a convex combination of base kernels, *i.e.*, $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{d}^\top \mathbf{k}(\mathbf{x}_i, \mathbf{x}_j)$, where $\mathbf{k}(\mathbf{x}_i, \mathbf{x}_j) = (k_1(\mathbf{x}_i, \mathbf{x}_j), k_2(\mathbf{x}_i, \mathbf{x}_j), \ldots, k_M(\mathbf{x}_i, \mathbf{x}_j))^\top$, $\mathbf{d} = (d_1, d_2, \ldots, d_M)^\top$, $\sum_{m=1}^M d_m = 1$, and $d_m \geq 0$. Hence we have $\mathbf{K} = \sum_{m=1}^M d_m \mathbf{K}_m$, where $\mathbf{K}_m$ is the m^{th} base kernel matrix with $\kappa_{ij}^m = k_m(\mathbf{x}_i, \mathbf{x}_j)$ as its elements. Substituting $\mathbf{K}$ for that in (2), we obtain the constrained optimization problem as follows.

$$\begin{aligned} \min_{\boldsymbol{\alpha}, \mathbf{d}} \quad & F(\boldsymbol{\alpha}, \mathbf{d}), \\ \text{s.t.} \quad & \textstyle\sum_{m=1}^M d_m = 1, \\ & d_m \geq 0, \quad m = 1, \ldots, M, \end{aligned} \tag{3}$$

where

$$F(\boldsymbol{\alpha}, \mathbf{d}) = \frac{1}{2} \left\| \mathbf{y} - \sum_{m=1}^M d_m \mathbf{K}_m \boldsymbol{\alpha} \right\|_2^2 + \frac{\lambda}{2} \boldsymbol{\alpha}^\top \sum_{m=1}^M d_m \mathbf{K}_m \boldsymbol{\alpha}. \tag{4}$$

The optimal solution to Problem (3) can be expressed as

$$f^*(\mathbf{x}) = \sum_{i=0}^{l-1} \alpha_i \mathbf{d}^\top \mathbf{k}(\mathbf{x}_i, \mathbf{x}). \tag{5}$$

Given $\mathbf{d}$ in Problem (3), we get an unconstrained quadratic programming problem w.r.t. $\boldsymbol{\alpha}$. And given $\boldsymbol{\alpha}$, Problem (3) is the constrained quadratic programming w.r.t. $\mathbf{d}$. Let $\{\mathbf{K}_m\}$ be positive semi-definite. Then, it is clear that given $\mathbf{d}$, $F(\boldsymbol{\alpha}, \mathbf{d})$ is convex w.r.t. $\boldsymbol{\alpha}$, and given $\boldsymbol{\alpha}$, $F(\boldsymbol{\alpha}, \mathbf{d})$ is convex w.r.t. $\mathbf{d}$.

To solve for $\boldsymbol{\alpha}$, let $\nabla_{\boldsymbol{\alpha}} F(\boldsymbol{\alpha}, \mathbf{d}) = 0$; it is achieved that

$$\boldsymbol{\alpha} = \left(\sum_{m=1}^M d_m \mathbf{K}_m + \lambda \mathbf{I} \right)^{-1} \mathbf{y}, \tag{6}$$

where $\mathbf{I}$ is an $l \times l$ identity matrix. And $\mathbf{d}$ can be determined with the quadprog function in Matlab's optimization toolbox. Initially, $\forall m$, $d_m = 1/M$. Then, because $F(\boldsymbol{\alpha}, \mathbf{d}) \geq 0$, alternately evaluating Eq. (6) with fixed $\mathbf{d}$ and invoking the quadprog function with fixed $\boldsymbol{\alpha}$ for $\mathbf{d}$ will achieve a local optimal solution $(\boldsymbol{\alpha}^*, \mathbf{d}^*)$.

3.1.1 Fast Evaluation in Training

As stated in Sec. 3.1, the training samples are cyclically shifting in correlation filters. Therefore, the optimization processes of $\boldsymbol{\alpha}$ and $\mathbf{d}$ can be speeded up by means of the fast Fourier transform (FFT) pair, $\mathcal{F}$ and $\mathcal{F}^{-1}$.

At first, the evaluation of first rows $\mathbf{k}_m$s of kernel matrices $\mathbf{K}_m$s can be accelerated with FFT because the samples are circulant [25, 26]. Because $\mathbf{K}_m$s are circulant [25], the inverses and the sum of circulant matrices are circulant [22]. Then the evaluation of Eq. (6) can be accelerated as

$$\boldsymbol{\alpha} = \mathcal{F}^{-1} \left(\frac{\mathcal{F}(\mathbf{y})}{\mathcal{F}\left(\sum_{m=1}^M d_m \mathbf{k}_m \right) + \lambda} \right). \tag{7}$$

According to Eq. (4), given $\boldsymbol{\alpha}$, the optimization function $F(\mathbf{d}; \boldsymbol{\alpha})$ w.r.t. $\mathbf{d}$ can be expressed as

$$F(\mathbf{d}; \boldsymbol{\alpha}) = \frac{1}{2} \mathbf{d}^\top \mathbf{A}_{\mathbf{d}} \mathbf{d} + \frac{1}{2} \mathbf{d}^\top \mathbf{B}_{\mathbf{d}} + \frac{1}{2} \mathbf{y}^\top \mathbf{y}, \tag{8}$$

where

$$\mathbf{A}_{\mathbf{d}} = \begin{pmatrix} \boldsymbol{\alpha}^\top \mathbf{K}_1^\top \mathbf{K}_1 \boldsymbol{\alpha} & \cdots & \boldsymbol{\alpha}^\top \mathbf{K}_1^\top \mathbf{K}_M \boldsymbol{\alpha} \\ \vdots & \ddots & \vdots \\ \boldsymbol{\alpha}^\top \mathbf{K}_M^\top \mathbf{K}_1 \boldsymbol{\alpha} & \cdots & \boldsymbol{\alpha}^\top \mathbf{K}_M^\top \mathbf{K}_M \boldsymbol{\alpha} \end{pmatrix}, \tag{9}$$

and

$$\mathbf{B}_{\mathbf{d}} = \left(\mathbf{b}_{\mathbf{d}}^\top \mathbf{K}_1 \boldsymbol{\alpha}, \ldots, \mathbf{b}_{\mathbf{d}}^\top \mathbf{K}_M \boldsymbol{\alpha} \right)^\top, \tag{10}$$

$\mathbf{b}_{\mathbf{d}} = \lambda \boldsymbol{\alpha} - 2\mathbf{y}$. The evaluation of $\mathbf{A}_{\mathbf{d}}$ and $\mathbf{B}_{\mathbf{d}}$ can be accelerated by evaluating $\mathbf{K}_m \boldsymbol{\alpha}$ with $\mathcal{F}^{-1}(\mathcal{F}^*(\mathbf{k}_m) \odot \mathcal{F}(\boldsymbol{\alpha}))$, where $m = 1, \ldots, M$.

3.1.2 Fast Detection

According to Eq. (5), the MKCF evaluates the responses of all test samples $\mathbf{z}_n = \mathbf{P}_l^n \mathbf{z}$, $n = 0, 1, \ldots, l-1$, in the current frame $p+1$ as

$$y^n(\mathbf{z}) = \sum_{m=1}^M d_m \sum_{i=0}^{l-1} \alpha_i k_m(\mathbf{z}_n, \mathbf{x}_{m,i}^p), \tag{11}$$

where $\mathbf{z}$ is the base test sample, $\mathbf{x}^p_{m,i} = \mathbf{P}^i_l \mathbf{x}^p_m$, $\mathbf{x}^p_m$ is the weighted average of the m^{th} feature of historical locations till frame p. Formally,

$$\mathbf{x}^p_m = (1-\eta_m)\mathbf{x}^{p-1}_m + \eta_m R(D(\iota(p), s^*_p), \zeta, m), \quad (12)$$

where $\eta_m \in [0,1]$ is the learning rate of kernel m for the appearance of training samples, $\iota(p)$ and s^*_p are the optimal location and scale of target object in frame p, respectively, ζ is the pre-defined scale for the image sequence, $D(\iota(p), s^*_p)$ is the image patch determined by $\iota(p)$ and s^*_p in frame p, $R(D, \zeta, m)$ denotes D re-sampled by ζ for kernel m, and $\mathbf{x}^0_m$ is the feature in the initial frame.

Because $k_m(,)$'s are permutation-matrix-invariant, the response map, $\mathbf{y}(\mathbf{z})$, of all virtual samples generated by $\mathbf{z}$ can be evaluated as

$$\mathbf{y}(\mathbf{z}) \equiv (y^0(\mathbf{z}), \ldots, y^{l-1}(\mathbf{z}))^\top = \sum_{m=1}^{M} d_m \mathbf{C}(\mathbf{k}^p_m)\alpha, \quad (13)$$

where $\mathbf{k}^p_m = (k^p_{m,0}, \ldots, k^p_{m,l-1})$, $k^p_{m,i} = k_m(\mathbf{z}, \mathbf{P}^i_l \mathbf{x}^p_m)$, and $\mathbf{C}(\mathbf{k}^p_m)$ is the circulant matrix with $\mathbf{k}^p_m$ as its first row. Therefore, the response map can be accelerated as follows.

$$\mathbf{y}(\mathbf{z}) = \sum_{m=1}^{M} d_m \mathcal{F}^{-1}\left(\mathcal{F}^*(\mathbf{k}^p_m) \odot \mathcal{F}(\boldsymbol{\alpha})\right). \quad (14)$$

The element of $\mathbf{y}(\mathbf{z})$ which takes the maximal value is accepted as the optimal location of object in frame $p+1$. And the target's optimal scale is determined with fDSST [14].

3.2. Shortcoming of Multi-kernel Correlation Filter

In order to achieve the robust performance of location, MKCF is updated with the weighted average of historical samples. To improve the location performance further, we would like to train a common MKCF (*i.e.*, common $\boldsymbol{\alpha}$ and $\mathbf{d}$) for the historical samples, just like what was done in [16]. Then, the optimization function should be as follows.

$$\begin{aligned} &F_e(\boldsymbol{\alpha}, \mathbf{d}) = \\ &\sum_{j=1}^{p} \beta^j \left(\frac{1}{2} \left\| \mathbf{y} - \sum_{m=1}^{M} d_m \mathbf{K}^j_m \boldsymbol{\alpha} \right\|_2^2 + \frac{\lambda}{2} \boldsymbol{\alpha}^\top \sum_{m=1}^{M} d_m \mathbf{K}^j_m \boldsymbol{\alpha} \right) \\ &= \frac{1}{2} \sum_{m=1}^{M} \sum_{j=1}^{p} \beta^j \left(\mathbf{y}^\top \mathbf{y} - 2 d_m \mathbf{y}^\top \mathbf{K}^j_m \boldsymbol{\alpha} + \lambda d_m \boldsymbol{\alpha}^\top \mathbf{K}^j_m \boldsymbol{\alpha} \right) \\ &+ \frac{1}{2} \sum_{j=1}^{p} \beta^j \boldsymbol{\alpha}^\top \sum_{m=1}^{M} d_m \mathbf{K}^j_m \sum_{m=1}^{M} d_m \mathbf{K}^j_m \boldsymbol{\alpha}, \end{aligned}$$

where β^j is the weight of optimization function of the sample in frame j, $\mathbf{K}^j_m$ is the circulant kernel matrix with $\mathbf{k}^j_m$ as its first row, $\mathbf{k}^j_m = (k^j_{m,0}, \ldots, k^j_{m,l-1})$, $k^j_{m,i} = k_m(\mathbf{z}, \mathbf{P}^i_l \mathbf{x}^j_m)$, $j = 1, \ldots, p$. $\mathbf{x}^j_m$ is evaluated by using Eq. (12) where j is used instead of p.

Commonly, different kernels (i.e., features) should be equipped with different weights β^j, as their robustness is different throughout an image sequence. For example, the colors of the target object may vary more frequently than its HOG in an image sequence. Nevertheless, it is impossible for different kernels to set different β^j in $F_e(\boldsymbol{\alpha}, \mathbf{d})$, because different kernels are multiplied by each other and can not be separated into different items. Therefore, it is expectable that the location performance will be affected negatively if $F_e(\boldsymbol{\alpha}, \mathbf{d})$, instead of $F(\boldsymbol{\alpha}, \mathbf{d})$, is used in Problem (3), because different kernels have to share the same weight β^j.

3.3. Extension of Multi-kernel Correlation Filter with Upper Bound

Let $\mathbf{y}_c = \mathbf{y}/\mathbf{M}$. We have

$$\begin{aligned} F(\boldsymbol{\alpha}, \mathbf{d}) &= \frac{1}{2} \left\| \mathbf{y} - \sum_{m=1}^{M} d_m \mathbf{K}_m \boldsymbol{\alpha} \right\|_2^2 + \frac{\lambda}{2} \boldsymbol{\alpha}^\top \sum_{m=1}^{M} d_m \mathbf{K}_m \boldsymbol{\alpha} \\ &\le \frac{1}{2} \sum_{m=1}^{M} \left(\|\mathbf{y}_c - d_m \mathbf{K}_m \boldsymbol{\alpha}\|_2^2 + \lambda d_m \boldsymbol{\alpha}^\top \mathbf{K}_m \boldsymbol{\alpha} \right) \\ &\equiv U_{F(\boldsymbol{\alpha}, \mathbf{d})}. \end{aligned}$$

We then treat $U_{F(\boldsymbol{\alpha}, \mathbf{d})}$, the upper bound of $F(\boldsymbol{\alpha}, \mathbf{d})$, as the optimization function of MKCF and introduce the historical samples into it. Consequently, the final optimization objective for training a common multi-kernel correlation filter for the whole historical samples can be expressed as follows.

$$F_p(\boldsymbol{\alpha}_p, \mathbf{d}_p) \equiv \frac{1}{2} \sum_{j=1}^{p} \sum_{m=1}^{M} \beta^j_m u^{j,m}_{F(\boldsymbol{\alpha}, \mathbf{d})},$$

where

$$u^{j,m}_{F(\boldsymbol{\alpha}, \mathbf{d})} = \|\mathbf{y}_c - d_{m,p} \mathbf{K}^j_m \boldsymbol{\alpha}_p\|_2^2 + \lambda d_{m,p} \boldsymbol{\alpha}_p^\top \mathbf{K}^j_m \boldsymbol{\alpha}_p,$$

$\beta^1_m = (1-\gamma_m)^{p-1}$, $\beta^j_m = \gamma_m (1-\gamma_m)^{p-j}$, $j = 2, \ldots, p$, p is the number of historical frames, $\gamma_m \in (0,1)$ is the learning rate of kernel m for the common MKCF, $\mathbf{K}^j_m$ is the Gram matrix of the m^{th} kernel for the samples in frame j, $\boldsymbol{\alpha}_p = (\alpha_{0,p}, \alpha_{1,p}, \ldots, \alpha_{l-1,p})^\top$ and $\mathbf{d}_p = (d_{1,p}, d_{2,p}, \ldots, d_{M,p})^\top$ are dual vector and weight vector of all kernels when frame p is processed, respectively, and $\sum_{m=1}^{M} d_{m,p} = 1$. And the new optimization problem for the MKCF with whole samples is

$$\begin{aligned} \min_{\boldsymbol{\alpha}_p, \mathbf{d}_p} \quad & F_p(\boldsymbol{\alpha}_p, \mathbf{d}_p), \\ \text{s.t.} \quad & \textstyle\sum_{m=1}^{M} d_{m,p} = 1, \\ & d_{m,p} \ge 0, \quad m = 1, \ldots, M. \end{aligned} \quad (15)$$

This is a constrained optimization problem. And similar to Problem (3), given $\mathbf{d}_p$, $F_p(\boldsymbol{\alpha}_p, \mathbf{d}_p)$ is convex and unconstrained w.r.t. $\boldsymbol{\alpha}_p$, and given $\boldsymbol{\alpha}_p$, $F_p(\boldsymbol{\alpha}_p, \mathbf{d}_p)$ is convex and constrained w.r.t. $\mathbf{d}_p$.

Because $F_p(\boldsymbol{\alpha}_p, \mathbf{d}_p)$ is unconstrained w.r.t. $\boldsymbol{\alpha}_p$, to solve for $\boldsymbol{\alpha}_p$, let $\nabla_{\boldsymbol{\alpha}_p} F_p(\boldsymbol{\alpha}_p, \mathbf{d}_p) = 0$; we achieve that

$$\boldsymbol{\alpha}_p = \left(\sum_{j=1}^{p} \sum_{m=1}^{M} \beta_m^j \left((d_{m,p} \mathbf{K}_m^j)^2 + \lambda d_{m,p} \mathbf{K}_m^j \right) \right)^{-1} \cdot \sum_{j=1}^{p} \sum_{m=1}^{M} \beta_m^j d_{m,p} \mathbf{K}_m^j \mathbf{y}_c, \tag{16}$$

which can be evaluated efficiently with FFT as follows.

$$\begin{aligned} \mathcal{A}_p &\equiv \mathcal{F}(\boldsymbol{\alpha}_p) \\ &= \frac{\sum_{j=1}^{p} \sum_{m=1}^{M} \beta_m^j \mathcal{F}(d_{m,p} \mathbf{k}_m^j) \odot \mathcal{F}(\mathbf{y}_c)}{\sum_{j=1}^{p} \sum_{m=1}^{M} \beta_m^j \mathcal{F}(d_{m,p} \mathbf{k}_m^j) \odot (\mathcal{F}(d_{m,p} \mathbf{k}_m^j) + \lambda)}. \end{aligned}$$

Set

$$\mathcal{A}_p = \frac{\mathcal{A}_p^N}{\mathcal{A}_p^D} = \frac{\sum_{m=1}^{M} \mathcal{A}_{m,p}^N}{\sum_{m=1}^{M} \mathcal{A}_{m,p}^D}, \tag{17}$$

where

$$\begin{aligned} \mathcal{A}_{m,p}^N &= (1 - \gamma_m) \mathcal{A}_{m,p-1}^N + \gamma_m \mathcal{F}(d_{m,p} \mathbf{k}_m^p) \odot \mathcal{F}(\mathbf{y}_c), \\ \mathcal{A}_{m,p}^D &= (1 - \gamma_m) \mathcal{A}_{m,p-1}^D + \\ &\quad \gamma_m \mathcal{F}(d_{m,p} \mathbf{k}_m^p) \odot (\mathcal{F}(d_{m,p} \mathbf{k}_m^p) + \lambda), \end{aligned}$$

if $p > 1$. In the initial frame, $p = 1$. Then

$$\begin{aligned} \mathcal{A}_{m,1}^N &= \mathcal{F}(d_{m,1} \mathbf{k}_m^1) \odot \mathcal{F}(\mathbf{y}_c), \\ \mathcal{A}_{m,1}^D &= \mathcal{F}(d_{m,1} \mathbf{k}_m^1) \odot (\mathcal{F}(d_{m,1} \mathbf{k}_m^1) + \lambda). \end{aligned}$$

Therefore, $\mathcal{A}_p$ can be evaluated efficiently frame by frame.

Solving for $\mathbf{d}_p$ in Problem (15) will have to deal with a constrained optimization problem. This means that it is difficult to obtain an iteration scheme for the optimal $\mathbf{d}_p^*$ which is as efficient as the one for $\boldsymbol{\alpha}_p^*$. Now let us investigate the constraints in Problem (15). It is clear that there are three purposes for adding these constraints in Problem (15). (1) $d_{m,p} \geq 0$, $m = 1, \ldots, M$, are necessary to ensure $\sum_{m=1}^{M} d_{m,p}$ is convex combination. (2) $\sum_{m=1}^{M} d_{m,p} = 1$ is necessary to ensure the optimal $\mathbf{d}_p^*$ is unique and its value is finite. (3) Both $d_{m,p} \geq 0$ and $\sum_{m=1}^{M} d_{m,p} = 1$ are necessary to ensure there exists at least an m such that $d_{m,p} > 0$. Therefore, if we are able to design an algorithm to optimize the unconstrained problem

$$\min_{\boldsymbol{\alpha}_p, \mathbf{d}_p} F_p(\boldsymbol{\alpha}_p, \mathbf{d}_p) \tag{18}$$

w.r.t. $\mathbf{d}_p$, such that the above three requirements are satisfied implicitly, then the explicit constraints in Problem (15) can be canceled. In the rest of this section, we will first derive an efficient algorithm to optimize Problem (18) w.r.t. $\mathbf{d}_p$, and then prove that the optimal $\mathbf{d}_p^*$ indeed implicitly satisfies the above requirements for the optimal solution if $d_{m,1} > 0$, $m = 1, \ldots, M$.

To solve for $\mathbf{d}_p$ in Problem (18), let $\nabla_{\mathbf{d}_p} F_p(\boldsymbol{\alpha}_p, \mathbf{d}_p) = 0$. Then, it is achieved that

$$d_{m,p} = \frac{\sum_{j=1}^{p} \beta_m^j (\mathbf{K}_m^j \boldsymbol{\alpha}_p)^\top (2\mathbf{y}_c - \lambda \boldsymbol{\alpha}_p)}{2 \sum_{j=1}^{p} \beta_m^j (\mathbf{K}_m^j \boldsymbol{\alpha}_p)^\top (\mathbf{K}_m^j \boldsymbol{\alpha}_p)},$$

where $m = 1, \ldots, M$. Set

$$d_{m,p} = \frac{d_{m,p}^N}{d_{m,p}^D}, \tag{19}$$

where

$$\begin{aligned} d_{m,p}^N &= (1 - \gamma_m) d_{m,p-1}^N + \gamma_m (\mathbf{K}_m^p \boldsymbol{\alpha}_p)^\top (2\mathbf{y}_c - \lambda \boldsymbol{\alpha}_p), \\ d_{m,p}^D &= (1 - \gamma_m) d_{m,p-1}^D + 2\gamma_m (\mathbf{K}_m^p \boldsymbol{\alpha}_p)^\top (\mathbf{K}_m^p \boldsymbol{\alpha}_p), \end{aligned}$$

if $p > 1$. And if $p = 1$, then

$$\begin{aligned} d_{m,1}^N &= (\mathbf{K}_m^1 \boldsymbol{\alpha}_1)^\top (2\mathbf{y}_c - \lambda \boldsymbol{\alpha}_1), \\ d_{m,1}^D &= 2(\mathbf{K}_m^1 \boldsymbol{\alpha}_1)^\top (\mathbf{K}_m^1 \boldsymbol{\alpha}_1). \end{aligned}$$

It is clear that $\mathbf{K}_m^p \boldsymbol{\alpha}_p$ can be accelerated with

$$\mathcal{F}^{-1}(\mathcal{F}^*(\mathbf{k}_m^p) \odot \mathcal{F}(\boldsymbol{\alpha}_p)) = \mathcal{F}^{-1}(\mathcal{F}^*(\mathbf{k}_m^p) \odot \mathcal{A}_p).$$

Therefore, $d_{m,p}$ can be evaluated efficiently, and optimal solution $\mathbf{d}_p^*$ can be obtained efficiently frame by frame.

Theorem 1 *Suppose that $\mathbf{K}_m^j$ is circulant Gram matrix, $\lambda > 0$, all components of $\mathbf{y}_c$ is positive, and also suppose $d_{m,p}^t > 0$, $m = 1, \ldots, M$, $j = 1, \ldots, p$, $t = 1, 2, \ldots$, where $d_{m,p}^t$ is the t^{th} iteration on frame p when solving Problem (18) with alternative evaluation of $\boldsymbol{\alpha}_p$ and $\mathbf{d}_p$. Then,*

(1) $d_{m,p}^{t+1} > 0$,

(2) $c_l \cdot \lambda/2 + c_l \cdot b^{\min} < d_{m,p}^{t+1} < c_u \cdot \lambda/2 + c_u \cdot b^{\max}$, where c_l and c_u are two constants determined by $\mathbf{y}_c$, discrete Fourier transform matrix, β_m^j, and the eigenvalues of $\mathbf{K}_m^j$, $b^{\min}$ and $b^{\max}$ are two constants related to $d_{m,p}^t$, β_m^j, and the eigenvalues of $\mathbf{K}_m^j$.

The proof can be found in the supplementary material.

It can be seen from Theorem 1 that the range of $d_{m,p}^{t+1}$ is totally determined by two lines w.r.t. λ when $d_{m,p}^1$ is fixed. The smaller λ, the smaller $d_{m,p}^{t+1}$, therefore, the smaller the components of final optimal solution $\mathbf{d}_p^*$. That is, the components of $\mathbf{d}_p^*$ are always finite and controlled by λ. It is

obvious that $\mathbf{d}_p^*$ satisfies the three requirements for the optimal solution of Problem (18) w.r.t. $\mathbf{d}_p$, given the initial $d_{m,p}^1 > 0, m = 1, \ldots, M$.

More refined analysis on the relationship of λ and optimal $\mathbf{d}_p^*$ is complex, because the bounds of $\mathbf{d}_p^*$ heavily depend on the eigenvalues of all kernel matrices which are constructed with practical samples and an additional scale parameter in the kernel. Therefore, we will experimentally show the further numerical relation between λ and $\mathbf{d}_p^*$ in Sec. 5.1.

Based on the above analysis, it is concluded that the optimization objective of the extension of MKCF is Problem (18), and its optimization process is as follows. Initially, $d_{m,1} = 1/M$, $m = 1, \ldots, M$. Then alternately evaluate Eq. (17) with fixed $\mathbf{d}_p$ and Eq. (19) with fixed $\boldsymbol{\alpha}_p$. Because $F_p(\boldsymbol{\alpha}_p, \mathbf{d}_p) \geq 0$ is convex w.r.t. $\boldsymbol{\alpha}_p$ and $\mathbf{d}_p$, respectively, such iterations will converge to a local optimal solution $(\boldsymbol{\alpha}_p^*, \mathbf{d}_p^*)$. In our experiments, a satisfactory convergency $(\boldsymbol{\alpha}_p^*, \mathbf{d}_p^*)$ on frame p can be achieved in three iterations of Eq. (17) and Eq. (19).

The fast determination of the optimal location and scale of target object in frame $p+1$ is the same as that of MKCF described in Sec. 3.1.2, where $\boldsymbol{\alpha} = \boldsymbol{\alpha}_p^*$ and $\mathbf{d} = \mathbf{d}_p^*$.

4. Implementation Details

In our experiments, the color and HOG are used as features in MKCFup. Considering the tradeoff between the discriminability and computational cost, we employ a kernel for each of color and HOG, *i.e.*, $M = 2$. As in [16, 26, 11, 48], the multiple channels of the color and HOG are concatenated into a single vector, respectively.

The color scheme proposed by [16] is adopted as our color feature, except that we reduce the dimensionality of color to four with principal component analysis (PCA). Normal nine gradient orientations and 4×4 cell size are utilized in HOGs. The dimensionality of our HOGs is also reduced to four with PCA to speed up MKCFup. Gaussian kernel is used for both features with $\sigma_{\text{color}} = 0.515$ and $\sigma_{\text{HOG}} = 0.6$ for color sequences and $\sigma_{\text{color}} = 0.3$ and $\sigma_{\text{HOG}} = 0.4$ for gray sequences. Employing Gaussian kernel to construct kernel matrices ensures that all $\mathbf{K}_m$s are positive definite [40]. The learning rates $\gamma_{\text{color}} = 0.0174$ and $\gamma_{\text{HOG}} = 0.0173$ for color sequences, and $\gamma_{\text{color}} = 0.0175$ and $\gamma_{\text{HOG}} = 0.018$ for gray sequences. The learning rates of sample appearance $\eta_{\text{color}} = \gamma_{\text{color}}$ and $\eta_{\text{HOG}} = \gamma_{\text{HOG}}$ for both color and gray sequences.

In order to reduce high-frequency noise in the frequency domain stemming from the large discontinuity between opposite edges of a cyclic-extended image patch, the feature patches are banded with Hann window. Because there is only one true sample in each frame, it is well known that too large a search region in KCF will reduce the location performance [25, 16]. Therefore, the search region is set 2.5 times larger than the bounding box of target object, which is the same as that in KCF and CN_2 [16].

5. Experimental Results

The MKCFup was implemented in MATLAB. The experiments were performed on a PC with Intel Core i7 3.40GHz CPU and 8GB RAM.

It is well-known that all samples of MOSSE, KCF, MKCF, and MKCFup are circulant. Therefore, their search region can not be set too large [12]. Too large a search region will include too much background, significantly reducing the discriminability of filters for target object against background. Consequently, the search regions of above CF trackers have to be set experientially around 2.5 times larger than the object bounding boxes [26, 48], much smaller than those of CFLB, SRDCF, and ECO_HC [19, 12, 9]. It is obvious that it will be impossible for any tracker to catch the target object once the target moves out of its search region in the next frame. Therefore, CFLB, SRDCF, and ECO_HC are better for locating the target object of large move than KCF, MKCF, and MKCFup.

An even worse situation for KCF, MKCF, and MKCFup is that, according to the experimental experiences on correlation filter based trackers [4, 25, 10, 48], even if the target is in the search region in next frame, its location may still be unreliable when the target moving near to the boundaries of the search region. Specifically, it is often difficult for the CF trackers, such as MOSSE, CN_2 [16], KCF, MKCF, and MKCFup which use only one base sample, to obtain a reliable location by using response maps if the ratio of the center distance of target object over the bounding box in two frames is larger than 0.6 when the background clutter is present. Consequently, it is suitable for the above CF trackers to track the target object with quite small move between two frames. In this paper, the move of target object is defined as *small*, if the offset ratio

$$\tau \equiv \frac{\|\mathbf{c}(\mathbf{x}_t) - \mathbf{c}(\mathbf{x}_{t+\delta})\|_2}{\sqrt{w(\mathbf{x}_t) \cdot h(\mathbf{x}_t)}} < 0.6, \tag{20}$$

where $\mathbf{c}()$, $w()$, and $h()$ are the center, width, and height of sample, respectively. $\delta = 1$ if there is no occlusion for the target object, otherwise δ is the amount of frames from starting to ending occlusion. A sequence is accepted to contain the target object of large move if there exists two adjacent frames or the occlusion of target object such that $\tau > 0.6$. It is noted that the above definition of offset ratio for small move is quite rough, because it neglects the possible big difference between width and height.

According to the above discussion, two visual tracking benchmarks, OTB2013 [55] and NfS [17] were utilized to compare different trackers in this paper, because most of sequences of OTB2013 and most of high frequency part of NfS only contain small move of the target object.

In our experiments, the trackers are evaluated in one-pass evaluation (OPE) using both precision and success plots [55], calculated as percentages of frames with center errors lower than a threshold and the intersection-over-union (IoU) overlaps exceeding a threshold, respectively. Trackers are ranked using the precision score with center error lower than 20 pixels and area-under-the-curve (AUC), respectively, in precision and success plots.

In this paper, to simplify the experiments, we only compare those state-of-the-art trackers which merely employ the hand-crafted features color or HOG.

5.1. Relationship of optimal weight $\mathbf{d}_p^*$ and regularization parameter λ

Fig. 2 shows the numerical relation of λ and $\mathbf{d}_p^*$ obtained on OTB2013 when initially $\mathbf{d}_p^1 = (0.5, 0.5)$. In the experiment, $\lambda \in \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3, 10^4\}$. According to Theorem 1, we set $d_{m,p}^* = d_{m,p}^*(\lambda)$, because $d_{m,p}^*$ is a function of λ, and

$$\overline{d}(\lambda) = \frac{1}{MPS} \sum_m \sum_p \sum_i d_{m,p,i}^*(\lambda),$$

$$(\delta_{\max}(\lambda), \delta_{\min}(\lambda)) = (\max_{m,p,i} d_{m,p,i}^*(\lambda), \min_{m,p,i} d_{m,p,i}^*(\lambda)),$$

where P and S are the number of selected frames in each image sequence and the total number of selected sequences, respectively, p and i represent the number of selected frame and the number of selected sequence, respectively, and $d_{m,p,i}^*$ is the optimal weight of the m^{th} kernel at frame p of sequence i. In our experiment, specifically, $P = 10$ and $S = 20$. That is, for each λ, ten frames are randomly sampled from each of 20 randomly selected sequences out of OTB2013, and $d_{m,p}^*(\lambda)$s on these frames are used to calculate $\overline{d}(\lambda)$ and two deviations, $\delta_{\max}(\lambda)$ and $\delta_{\min}(\lambda)$. To demonstrate the relationship more clearly, λ and its three functions are shown with logarithmic function.

It is interesting to notice that the relation of the averages of λ and optimal $\mathbf{d}_p^*$ is almost linear when $\lambda < 10^{-1}$ or $\lambda \geq 1$. And $\delta_{\max}(\lambda)$ and $\delta_{\min}(\lambda)$ drop significantly when $\lambda < 10^{-1}$. When $\lambda \leq 0.05$, the deviations are really close to the average, and the relation of λ and $\mathbf{d}_p^*$ itself is approximately linear. Surprisingly, $\frac{1}{M}\sum_{m=1}^{M} d_{m,p}^* \approx 0.5$ for the frames of all sequences when $\lambda < 10^{-1}$ in our experiment. That is, $\sum_{m=1}^{M} d_{m,p}^* \approx 1$, because $M = 2$. This means that the constraint of Problem (15) on the sum of all components of the optimal $\mathbf{d}_p$ is satisfied implicitly and approximately, while optimizing Problem (18) w.r.t. $\mathbf{d}_p$ with the iterations of Eq. (17) and Eq. (19).

5.2. Comparison among MKCFs

In this section, we consider KCF as a special case of the original MKCF [48] with $M = 1$. To verify our improvement on KCF and MKCF is effective, we compare KCF, KCFscale, MKCF, fMKCF, and MKCFup on OTB2013, where KCFscale is the KCF with the scaling scheme of patch pyramid, and fMKCF is a variant of MKCF whose features and scaling scheme are the same as those adopted by MKCFup, and the optimization of $\mathbf{d}$ that is more efficient than the one in [48], as described in Sec. 3.1.1, is adopted. Fig. 3 reports the results. It is concluded from the figure that MKCFup outperforms KCF and KCFscale with large margins in both center precision and IoU, and that the novel objective function and training scheme of MKCFup improve the location performance with the average precision score of 83.5% and the average AUC score of 64.1%, significantly outperforming MKCF and fMKCF by 6.8% and 4.9% and 7.7% and 6.1%, respectively. It is noticed that the location performances of fMKCF are inferior to those of MKCF, although its fps is higher than MKCF's (50 vs. 30).

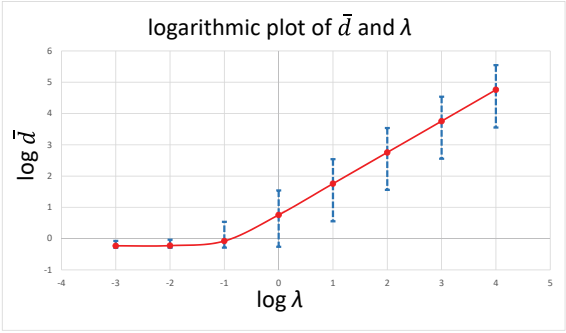

Figure 2. The numerical relationship between regularization parameter λ and $\overline{d}$, the average of $d_{m,p}^*$ over m on OTB2013 [55]. Besides $\overline{d}$, two deviations away from $\overline{d}$ are also presented. The logarithmic function is employed to make the relation more clear. See Sec. 5.1 for details.

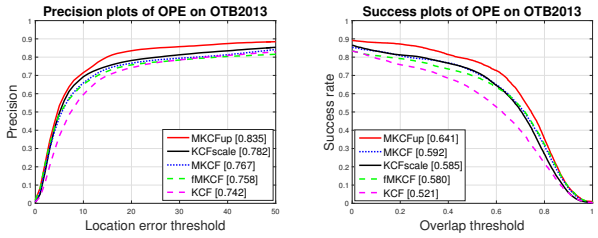

Figure 3. The precision and success plots of KCF [26], KCFscale, MKCF [48], fMKCF, and MKCFup on OTB2013 [55]. See Sec. 5.2 for details. The average precision scores and AUCs of the trackers on the sequences are reported in the legends.

5.3. Comparison to State-of-the-art Trackers with Handcrafted Features

We compare our MKCFup to other 6 trackers, KCF, KCFscale, MKCF, SRDCF, fDSST, and ECO_HC on OTB2013 and NfS. Fig. 4 shows the results. It can be seen

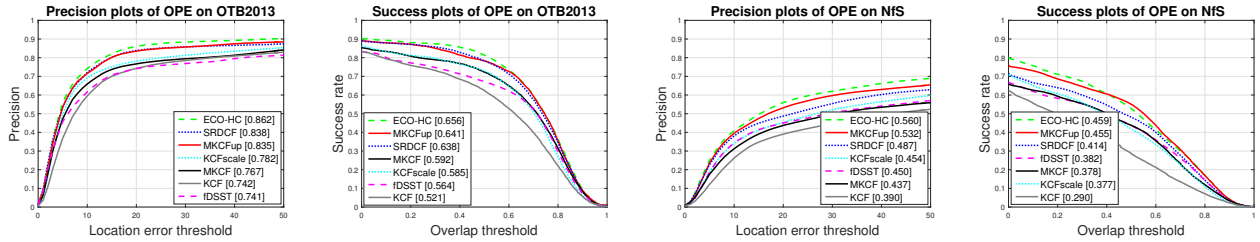

Figure 4. The precision and success plots of MKCFup, KCF, KCFscale, MKCF, SRDCF, fDSST, and ECO_HC on OTB2013 [55] and NfS [17]. The average precision scores and AUCs of the trackers on the sequences are reported in the legends.

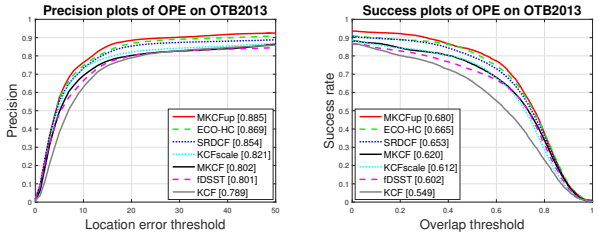

Figure 5. The precision and success plots of MKCFup, KCF, KCFscale, MKCF, SRDCF, fDSST, and ECO_HC on small move sequences of OTB2013 [55]. The average precision scores and AUCs of the trackers on the sequences are reported in the legends.

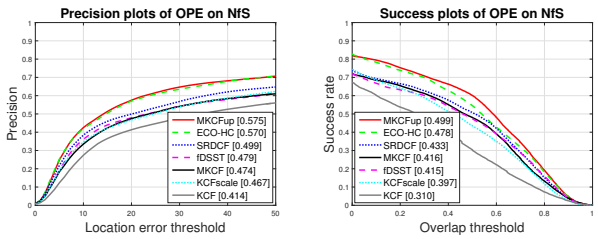

Figure 6. The precision and success plots of MKCFup, KCF, KCFscale, MKCF, SRDCF, fDSST, and ECO_HC on small move sequences of NfS [17]. The average precision scores and AUCs of the trackers on the sequences are reported in the legends.

that MKCFup outperforms all other trackers in both precision scores and AUCs, except for ECO_HC, on two benchmarks. ECO_HC is able to exploit larger search regions than MKCFup does to catch the target object of large move, whereas MKCFup is not. Therefore, ECO_HC outperforms MKCFup on the whole benchmarks.

5.4. Comparison on Sequences of Small Move

By means of Eq. (20), it is found that there exist six sequences which contain large move in OTB2013.[1] We then removed them from the benchmark and compared our MKCFup, KCF, KCFscale, MKCF, SRDCF, fDSST, and ECO_HC on the rest sequences. Fig. 5 reports the results. It is seen that MKCFup outperforms SRDCF and ECO_HC on the average precision score and AUC by 3.1% and 2.7% and 1.6% and 1.5%, respectively, on the small move sequences of OTB2013.

To verify the advantage of MKCFup further, we removed the large move sequences[2] from NfS by means of Eq. (20), and compared the above trackers on the rest 84 sequences. Note that an occluded target object is considered undergoing large move if its $\tau > 0.6$ between the two frames of starting and ending occlusion. Fig. 6 shows the results. It is seen that MKCFup outperforms SRDCF and ECO_HC on the average precision score and AUC by 7.6% and 6.6% and 0.5% and 2.1%, respectively, on small move sequences of NfS.

Table 1 lists the amount of frames the trackers can process per second.

Table 1. The amount of frames processed per second (fps) with different trackers.

Tracker	KCF	MKCF	fMKCF	fDSST	SRDCF	ECO-HC	MKCFup
fps	297	30	50	80	6	39	150

According to the above experiments, it can be concluded that MKCFup outperforms state-of-the-art trackers, such as SRDCF and ECO_HC, with much higher fps as long as the move of target object is small.

6. Conclusions and Future Work

A novel tracker, MKCFup, has been presented in this paper. By optimizing the upper bound of the objective function of original MKCF and introducing the historical samples into the upper bound, we derived the novel MKCFup. It has been demonstrated that the discriminability of MKCFup is more powerful than those of state-of-the-art trackers, such as SRDCF and ECO-HC, although its search region is much smaller than theirs. And the MKCFup'fps is much larger than state-of-the-art trackers'. In conclusion, MKCFup outperforms state-of-the-arts trackers with handcrafted features at high speed if the target object moves small.

[1]The 6 sequences which contain the target object with large move in OTB2013 are boy, matrix, tiger2, ironman, couple, jumping.

[2]The 16 sequences which contain the target object with large move in NfS are airboard_1, airtable_3, bee, bowling3, football_skill, parkour, pingpong8, basketball_1, basketball_3, basketball_6, bowling2, dog_2, pingpong2, motorcross, person_scooter, soccer_player_3.

References

[1] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H.S. Torr. Staple: Complementary learners for real-time tracking. In *Proc. Computer Vision and Pattern Recognition*, 2016. 1

[2] A. Bibi, M. Mueller, and B. Ghanem. Target response adaptation for correlation filter tracking. In *Proc. European Conference on Computer Vision*, 2016. 2

[3] V. Boddeti, T. Kanade, and B. Kumar. Correlation filters for object alignment. In *Proc. Computer Vision and Pattern Recognition*, 2013. 2

[4] D. Bolme, R. Beveridge, B. Draper, and Y. Lui. Visual object tracking using adaptive correlation filters. In *Proc. Computer Vision and Pattern Recognition*, 2010. 1, 6

[5] J.-W. Choi, H. Chang, J. Jeong, Y. Demiris, and J.-Y. Choi. Visual tracking using attention-modulated disintegration and integration. In *Proc. Computer Vision and Pattern Recognition*, 2016. 1

[6] J.-W. Choi, H. Chang, S. Yun, T. Fischer, Y. Demiris, and J.-Y. Choi. Attentional correlation filter network for adaptive visual tracking. In *Proc. Computer Vision and Pattern Recognition*, 2017. 1

[7] C. Cortes, M. Mohri, and A. Rostamizadeh. l_2 regularization for learning kernels. In *Proc. Uncertainty in Artificial Intelligence*, 2009. 2

[8] Z. Cui, S. Xiao, J. Feng, and S. Yan. Recurrently target-attending tracking. In *Proc. Computer Vision and Pattern Recognition*, 2016. 1

[9] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg. Eco: Efficient convolution operators for tracking. In *Proc. Computer Vision and Pattern Recognition*, 2017. 1, 2, 6

[10] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *Proc. British Machine Vision Conference (BMVC)*, 2014. 1, 6

[11] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Convolutional features for correlation filter based visual tracking. In *Proc. International Conference on Computer Vision Workshop: VOT*, 2015. 6

[12] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Learning spatially regularized correlation filters for visual tracking. In *Proc. International Conference on Computer Vision*, 2015. 1, 2, 6

[13] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking. In *Proc. Computer Vision and Pattern Recognition*, 2016. 1

[14] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Discriminative scale space tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 1, 4

[15] M. Danelljan, A. Robinson, F. Shahbaz Khan, and M. Felsberg. Learning continuous convolution operators for visual tracking. In *Proc. European Conference on Computer Vision*, 2016. 1, 2

[16] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. van de Weijer. Adaptive color attributes for real-time visual tracking. In *Proc. Computer Vision and Pattern Recognition*, 2014. 1, 4, 6

[17] H. Galoogahi, A. Fagg, C. Huang, D. Ramanan, and S. Lucey. Need for speed: A benchmark for higher frame rate object tracking. In *Proc. International Conference on Computer Vision*, 2017. 1, 2, 6, 8

[18] H. Galoogahi, T. Sim, and S. Lucey. Multi-channel correlation filters. In *Proc. International Conference on Computer Vision*, 2013. 2

[19] H. Galoogahi, T. Sim, and S. Lucey. Correlation filters with limited boundaries. In *Proc. Computer Vision and Pattern Recognition*, 2015. 2, 6

[20] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *Proc. International Conference on Computer Vision*, 2009. 1, 2

[21] M. Gönen and E. Alpaydın. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12:pp.2211–2268, 2011. 2

[22] R. Gray. *Toeplitz and Circulant Matrices: A review*. Now Publishers Inc., 2006. 3

[23] B. Han, J. Sim, and H. Adam. Branchout: Regularization for online ensemble tracking with convolutional neural networks. In *Proc. Computer Vision and Pattern Recognition*, 2017. 1

[24] J. Henriques, J. Carreira, R. Caseiro, and J. Batista. Beyond hard negative mining: Efficient detector learning via block-circulant decomposition. In *Proc. International Conference on Computer Vision*, 2013. 2

[25] J. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *Proc. European Conference on Computer Vision*, 2012. 1, 3, 6

[26] J. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.37:pp.583–596, 2015. 1, 3, 6, 7

[27] J. Henriques, P. Martins, R. Caseiro, and J. Batista. Fast training of pose detectors in the fourier domain. In *Proc. Neural Information Processing Systems (NIPS)*, 2014. 2

[28] Z. Hong, C. Wang, X. Mei, D. Prokhorov, and D. Tao. Tracking using multilevel quantizations. In *Proc. European Conference on Computer Vision*, 2014. 1

[29] Y. Hua, K. Alahari, and C. Schmid. Online object tracking with proposal selection. In *Proc. International Conference on Computer Vision*, 2015. 1

[30] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.34:pp.1409–1422, 2012.

[31] B. Kumar, A. Mahalanobis, and R. Juday. *Correlation Pattern Recognition*. Cambridge University Press, 2005. 2

[32] J. Kwon, J. Roh, K.-M. Lee, and L. Van Gool. Robust visual tracking with double bounding box model. In *Proc. European Conference on Computer Vision*, 2014. 1

[33] X. Lan, A. Ma, and P. Yuen. Multi-cue visual tracking using robust feature-level fusion based on joint sparse representation. In *Proc. Computer Vision and Pattern Recognition*, 2014. 1

[34] G. Lanckriet, T. De Bie, N. Cristianini, M. Jordan, and W. Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20:2626–2635, 2004. 3

[35] D. Lee, J.-Y. Sim, and C.-S. Kim. Visual tracking using pertinent patch selection and masking. In *Proc. Computer Vision and Pattern Recognition*, 2014. 1

[36] T. Liu, G. Wang, and Q. Yang. Real-time part-based visual tracking via adaptive correlation filters. In *Proc. Computer Vision and Pattern Recognition*, 2015. 1

[37] A. Lukezic, T. Tomas Vojir, L.-C. Zajc, J. Matas, and M. Kristan. Discriminative correlation filter with channel and spatial reliability. In *Proc. Computer Vision and Pattern Recognition*, 2017. 1

[38] C. Ma, J. Huang, X. Yang, and M. Yang. Hierarchical convolutional features for visual tracking. In *Proc. International Conference on Computer Vision*, 2015. 1

[39] C. Ma, X. Yang, C. Zhang, and M. Yang. Long-term correlation tracking. In *Proc. Computer Vision and Pattern Recognition*, 2015. 1

[40] C. Micchelle. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constructive Approximation*, Vol.2:pp.11–22, 1986. 6

[41] M. Mueller, N. Smith, and B. Ghanem. Context-aware correlation filter tracking. In *Proc. Computer Vision and Pattern Recognition*, 2017. 1

[42] H. Nam, S. Hong, and B. Han. Online graph-based tracking. In *Proc. European Conference on Computer Vision*, 2014. 1

[43] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and Y. M-H. Hedged deep tracking. In *Proc. Computer Vision and Pattern Recognition*, 2016. 1

[44] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521, 2008. 2

[45] R. Rifkin, G. Yeo, and T. Poggio. Regularized least-squares classification. *Nato Science Series Sub Series III: Computer and Systems Sciences*, pp131-154.:2003, 190. 1, 2

[46] D. Rozumnyi, J. Kotera, S. Sroubek, L. Novotny, and J. Matas. The world of fast moving objects. In *Proc. Computer Vision and Pattern Recognition*, 2017. 1

[47] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT press Cambridge, MA, 2002. 3

[48] M. Tang and J. Feng. Multi-kernel correlation filter for visual tracking. In *Proc. International Conference on Computer Vision*, 2015. 1, 2, 6, 7

[49] M. Tang and X. Peng. Robust tracking with discriminative ranking lists. *IEEE Transactions on Image Processing*, Vol.21(No.7):3273–3281, 2012. 1

[50] J. Valmadre, L. Luca Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr. End-to-end representation learning for correlation filter based tracking. In *Proc. Computer Vision and Pattern Recognition*, 2017. 1

[51] M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *Proc. International Conference on Computer Vision*, 2007. 1, 2, 3

[52] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *Proc. International Conference on Computer Vision*, 2009. 2

[53] L. Wang, W. Ouyang, X. Wang, and H. Lu. Stct: Sequentially training convolutional networks for visual tracking. In *Proc. Computer Vision and Pattern Recognition*, 2016. 1

[54] Y. Wu, G. Blasch, G. Chen, L. Bai, and H. Ling. Multiple source data fusion via sparse representation for robust visual tracking. In *FUSION*, 2011. 1

[55] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking - a benchmark. In *Proc. Computer Vision and Pattern Recognition*, 2013. 1, 2, 6, 7, 8

[56] F. Yang, H. Lu, and M. Yang. Robust visual tracking via multiple kernel boosting with affinity constraints. *IEEE Transactions on Circuits and Systems for Video Technology*, 24:pp.242–254, 2014. 1

[57] S. Yun, J.-W. Choi, Y. Yoo, K. Yun, and J.-Y. Choi. Action-decision networks for visual tracking with deep reinforcement learning. In *Proc. Computer Vision and Pattern Recognition*, 2017. 1

[58] L. Zhang, J. Varadarajan, and P.-N. Suganthan. Robust visual tracking using oblique random forests. In *Proc. Computer Vision and Pattern Recognition*, 2017. 1

[59] T. Zhang, A. Bibi, and B. Ghanem. In defense of sparse tracking: Circulant sparse tracking. In *Proc. Computer Vision and Pattern Recognition*, 2016. 1

[60] T. Zhang, C. Xu, and M.-H. Yang. Multi-task correlation particle filter for robust object tracking. In *Proc. Computer Vision and Pattern Recognition*, 2017. 1