



Learning Noise-Aware Correlation Filter for Visual Tracking

Xinyan Liang¹, Xiao Wang¹, Jin Tang¹, and Chenglong Li^{1,2}(✉)

¹ School of Computer Science and Technology, Anhui University, Hefei, China
lxy2626@foxmail.com, wangxiaocvpr@foxmail.com, jtang99029@foxmail.com,
lc113140@foxmail.com

² Center for Research on Intelligent Perception and Computing, NLP, CASIA,
Beijing, China

Abstract. Correlation filter has recently attracted much attention in visual tracking due to their excellent performance on both accuracy and efficiency. However, the adopted features, such as Colors, HOG and deep features, usually include noises and/or corruptions which might disturb the tracking performance. To handle this problem, we propose a novel noise-aware correlation filter method for robust visual tracking. In particular, we decompose the input feature matrix into a “clean” feature matrix and a sparse noise matrix, and then use the “clean” feature to train the correlation filter. To optimize the proposed correlation filter, we design an efficient ADMM (alternation direction of multipliers) solver. Extensive experimental results on the OTB-2013 dataset show that the proposed approach performs favorably against state-of-the-art trackers.

Keywords: Correlation filter · Feature decomposition
“Clean” feature

1 Introduction

Visual tracking is one of the most challenging and active tasks in computer vision and has drawn much attention due to its wide applications, such as video surveillance, human-computer interactions, and self-driving cars. Given the ground truth in the initial frame, the goal of visual tracking is to find all bounding boxes of the target object in subsequent frames. Despite many recent breakthroughs in visual tracking, it still remains challenge due to diverse factors, such as occlusion, object deformation, scale variation and background clutter.

Correlation Filter (CF) has recently attracted much attention in visual tracking [1, 6, 7, 11, 14, 15, 24, 29–31] due to their excellent performance in both accuracy and efficiency. CF trackers employ the cyclic shifts to generate dense samples, and diagonalize them in the Fourier domain by using the Fast Fourier Transform (FFT). It enables trackers robust and high speed. The seminal work of CF tracking is proposed by Bolme et al. [1], which achieves hundreds of frames



Fig. 1. Some examples of noises in bounding boxes.

per second and high tracking accuracy. However, MOSSE only employs the simple feature to represent objects, *i.e.*, brightness feature, without enough to be adopted in some complicated situations. To improve the tracking performance, most successful CF trackers use a discriminative object representation with either strong hand-crafted features such as HOG [7, 15, 24], color names [9, 24], or deep features [6, 8, 25]. Recent work has integrated deep features [25] trained on large dataset, such as ImageNet, to represent objects. In addition, multiple type of features are also employed together to robustly represent the tracked object, such as HOG features and color name [7, 24], and HOG features, color name and deep features [6, 8]. Although these trackers have achieved appealing results in both accuracy and computational efficiency, they ignore that these features might be polluted by noises or corruptions. As shown in Fig. 1, the bounding box of objects usually has several background information, which caused by occlusion or irregular shape of objects. Noises in features result in model drifting by influencing the learned appearance model and filter. Figure 2 shows the tracking results of Dual Correlation Filter (DCF) [15] on sequence **Soccer**. It illustrates that DCF will lose the object with noises in bounding box, which suggests that noises influence the tracking performance.

Motivated by the robust principal component analysis (RPCA) [3], Sui [28] decompose the feature matrix into a low-rank matrix and noise matrix. But the optimization of low-rank constraints refers to singular value decomposition (SVD). And SVD has high computation complexity and extremely time consuming. It influences the efficiency of trackers while real-time is a crucial factors in visual tracking. Therefore, we propose a simple and efficient feature decompose algorithms in this paper. According to [19–22, 28], we decompose the feature into the “clean” feature and noises. We do not impose the low-rank constraints on the “clean” feature and only impose the sparse constraint on the noises due to observations from Li *et al.* [17–19, 21, 22]. They think that the noise matrix is the sparse sample-specific corruptions, *i.e.*, a few patches are corrupted and others are clean. Motivated by these works, we suppose that the noise in features is also sparse. We aim to learn the “clean” feature through imposing the sparse constraint on noises. The noise-aware filter is optimized by the learned “clean” feature for mitigate noises effects on filter. The simple feature decomposition is incorporated into CF tracking framework. It improves the tracking accuracy by suppressing noises effects on features and filters.

This paper makes the following contributions to CF tracking and related applications as follows. First, we propose a noise-aware correlation filter tracking



Fig. 2. The example represents the effectiveness of our methods. Green, red, and blue box represents the ground truth, the tracking results of DCF, DCF_{our} , respectively. (Color figure online)

algorithms based on feature decomposition. The noise-aware filter and “clean” feature can be jointly optimized in an unified framework. Second, we also design an efficient ADMM (Alternation Direction Method of Multipliers) algorithm [2] that can optimize the filter and the “clean” features in a framework. Third, extensive experiments are carried out on public benchmark datasets. The evaluation results demonstrate the effectiveness of the proposed approach against the baseline methods.

2 Related Work

Recently, CF has obtained great achievement in visual tracking due to its accuracy and computational efficiency. Bolme *et al.* [1] first introduce the CF into visual tracking, which achieves hundreds of frames per second, and high tracking accuracy. However, there is a problem about MOSSE that it only employs the simple brightness feature of image that isn’t enough to adapt to some complicated situations.

More and more discriminative features are utilized in tracking to represent objects for improving the performance, such as HOG [7, 15, 24], Color Names [9, 24] and deep features [6, 8, 25]. In addition, several trackers [6–8, 24] employ multiple type features to represent the object for more robust tracking. To further enhance the ability to classify objects from the background, kernel tricks, which make the inseparable samples in low-dimension are mapped to the high-dimension space to achieve the purpose of classification, are used in CF tracking. Henriques *et al.* [14, 15] employ the kernel trick to improve performance. To adaptively employ complementary features, Tang *et al.* [30] propose a multi-kernel learning algorithm to improve performance. However, different kernels of MKCF may restrict each other in training and updating, which limits its improvement over KCF [15]. In addition, the increased computational cost of MKCF in comparison to KCF limits the tracking speed. Therefore, Tang *et al.* [31] employ a different way [30] to introduce the MKL into KCF. The way not only adaptively exploits multiple complementary features and non-linear

kernels more effectively than MKCF, but also keeps relative high speed. CF framework usually faces a problem that boundary effect which is caused by utilizing a periodic assumption of the training samples to efficiently learn a classifier. To address the boundary effect, SRDCF [7] is proposed by introducing the spatially regularized into the learning of correlation filter to penalize the filter coefficients near the boundary. In CSR-DCF [24], spatial reliability map is constructed to adjust the filter support to the part of object suitable for tracking. To adapt the size variation, several adaptive scale processing tracker [5, 23] are investigated. Danelljan *et al.* [5] utilize the two correlation filters to capture the location translation and scale estimation, respectively. Li *et al.* [23] employ an effective scale adaptive scheme and integrate the HOG features and Color Name features to boost the tracking performance.

In addition, Danelljan *et al.* [8] utilize continuous convolution to integrate multi-resolution feature maps. The factorized convolution operator and the generative sample space model are introduced into tracking [6] for addressing the over-fitting and computational complexity. CFnet [32] is the first to introduce the correlation filter into a deep neural network as a differentiable layer. Sun *et al.* [29] treat the filter as the element-wise product of a base filter and a reliability term to learn the discriminative and reliable information for improving the tracker's accuracy and robustness.

3 Review of Correlation Filters Tracking

In this section, we simply introduce the classical correlation filter tracking framework. Given the training set $T = [(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)]$, we find a linear regression function $h(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$ that minimizes the squared error over samples \mathbf{x}_i and their regression targets y_i . The model can be written as follows:

$$\min_{\mathbf{w}} \sum_i (h(\mathbf{x}_i) - y_i)^2 + \lambda \|\mathbf{w}\|^2 \quad (1)$$

where λ is a regularization parameter that controls overfitting. Equation (1) has a closed-form, which is given by [26]

$$\mathbf{w} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}. \quad (2)$$

where \mathbf{X} is circulant matrix generated by the base sample \mathbf{x} , \mathbf{I} is identity matrix. The per row of circulant matrix \mathbf{X} is one virtual sample \mathbf{x}_i obtained through the cyclic shift of the base sample \mathbf{x} . Let $\mathbf{y} = [y_1, y_2, \dots, y_n]^T$ is a regression target of samples \mathbf{X} . Each element y_i is a regression target of \mathbf{x}_i .

To calculate in the Fourier domain, the solution (2) is transformed into the complex version as follows:

$$\mathbf{w} = (\mathbf{X}^H \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^H \mathbf{y}. \quad (3)$$

where \mathbf{X}^H is the Hermitian transpose, that is the transpose of the complex-conjugate of \mathbf{X} , $\mathbf{X}^H = (\mathbf{X}^*)^T$. The circulant matrix \mathbf{X} can be expressed as

diagonal of \mathbf{x} by the Discrete Fourier Transform (DFT) [12]:

$$\mathbf{X} = F \text{diag}(\hat{\mathbf{x}}) F^H \quad (4)$$

where $\hat{\mathbf{x}}$ denotes the DFT of \mathbf{x} , that is $\hat{\mathbf{x}} = \mathcal{F}(\mathbf{x})$, $\text{diag}(\mathbf{x})$ denotes the diagonal matrix of a vector \mathbf{x} . \mathbf{F} is a constant matrix that does not depend on the generating vector \mathbf{x} , as $\mathcal{F}(\mathbf{z}) = \sqrt{n}F\mathbf{z}$. The notation n is the size of the generating vector \mathbf{x} . From now on, we will always use a hat $\hat{\mathbf{a}}$ as shorthand for the DFT of vector \mathbf{a} .

The property (4) of circulant matrix can be applied to the solution (3), which is expressed as follows:

$$\hat{\mathbf{w}} = \frac{\hat{\mathbf{x}}^* \odot \hat{\mathbf{y}}}{\hat{\mathbf{x}}^* \odot \hat{\mathbf{x}} + \lambda} \quad (5)$$

where \odot and the fraction denote element-wise product and division, respectively. And \mathbf{x}^* represents the complex-conjugate of \mathbf{x} .

The Eq. (1) can be transformed into dual domain. The dual objective function can be written as follows:

$$\min_{\alpha} \frac{1}{4\lambda} \alpha^T \mathbf{X} \mathbf{X}^T \alpha + \frac{1}{4} \alpha^T \alpha - \alpha^T \mathbf{y} \quad (6)$$

where α is the dual variable. The two solutions from the objective function (1) and (6) are related by $\mathbf{w} = \frac{\mathbf{X}^T \alpha}{2\lambda}$. Here, for clarity and avoiding the calculation of cyclic matrix, the dual form is rewritten as:

$$\min_{\alpha} \frac{1}{4\lambda} \alpha^T \mathbf{C}(\mathbf{x}) \mathbf{C}(\mathbf{x})^T \alpha + \frac{1}{4} \alpha^T \alpha - \alpha^T \mathbf{y} \quad (7)$$

where $\mathbf{C}(\mathbf{x})$ denotes the cyclic matrix generated by the base sample \mathbf{x} . The variable α can be optimized efficiently in the Fourier domain:

$$\hat{\alpha} = \frac{\hat{\mathbf{y}}}{\frac{1}{2\lambda} \hat{\mathbf{x}}^* \odot \hat{\mathbf{x}} + \frac{1}{2}} \quad (8)$$

where the fraction denote element-wise division.

The final response map ϕ is calculated through the following equation:

$$\phi = \mathcal{F}^{-1}(\hat{\mathbf{w}} \odot \hat{\mathbf{x}}) = \mathcal{F}^{-1}(\hat{\mathbf{k}}^{\hat{\mathbf{x}}\hat{\mathbf{z}}} \odot \hat{\alpha}) \quad (9)$$

where \mathcal{F}^{-1} denotes the inverse operation of DFT, $\hat{\mathbf{k}}^{\hat{\mathbf{x}}\hat{\mathbf{z}}}$ is the kernel of training samples \mathbf{x} and candidate patches \mathbf{z} . Then we can localize the target in current frame through the response map.

CF trackers usually employ several hand-craft features (*e.g.* HOG features, Color Name features) or deep learning features to represent the object for robust tracking. However, these features might be polluted by noises or corruptions. According to Eqs. (8) and (9), noises in features will influence the learned appearance model and filter and thus limit the tracking performance. As shown in Fig. 2, DCF and DCF_{our} can both work well when the tracked object has not noises.

However, DCF loses the tracked object but the DCF_{our} can successfully track the object when the object includes the noises. It illustrates that noises influence the learned filters and then limit the trackers performance. Therefore, we propose a noise-aware correlation filter tracking algorithm to mitigate the noises effect on filters. In next section, we will mainly introduce how the algorithms works in correlation filter tracking framework.

4 Methods

In this section, we give a detailed description about how to learn jointly noise-aware filter and “clean” feature through feature decomposition for suppressing the influence of noises.

4.1 Noise-Aware Correlation Filter

Objects can be represented using hand-craft features (*e.g.* HOG features [10], Color Name features [33]). Besides the hand-craft features, the deep features extracted from the VGG model [27] also are applied to represent the object. However, whether hand-craft features or deep features, they may be polluted by noises or corruptions. RPCA [3] has a powerful capability to suppress noises or corruptions. But the optimization of RPCA refers to singular value decomposition (SVD). And SVD has high computation complexity and extremely time consuming. It influences the efficiency of trackers. Therefore, we propose a simple and efficient feature decompose algorithms, which decomposes the feature \mathbf{x} into the “clean” feature \mathbf{z} and noises \mathbf{e} . And we impose the sparse constraint on noises and do not impose the low-rank constraint on the “clean” feature. The learned “clean” feature is used to optimize the noise-aware filter for mitigate noises effects.

$$\mathbf{x} = \mathbf{z} + \mathbf{e} \quad (10)$$

The Eq. (10) is incorporated into CF tracking framework to jointly optimize variables. The “clean” feature \mathbf{z} is used to optimize the noise-aware filter and make the filter more robust. The dual form model (7) can be reformulated as Eq. (11), which can learn the “clean” features and the noise-aware filter to improve the accuracy and robustness.

$$\begin{aligned} \min_{\alpha, \mathbf{z}, \mathbf{e}} \quad & \frac{1}{4\lambda} \alpha^T C(\mathbf{z}) C(\mathbf{z})^T \alpha + \frac{1}{4} \alpha^T \alpha - \alpha^T \mathbf{y} + \beta \|\mathbf{e}\|_0 \\ \text{s.t.} \quad & \mathbf{x} = \mathbf{z} + \mathbf{e} \end{aligned} \quad (11)$$

where λ is a regularization parameters that controls overfitting, and β is balanced parameter.

Because the l_0 norm is non-convexity, it is difficult to directly optimize the Eq. (11). To overcome the obstacles, we will use convex relaxation to relax the

non-convex sparsity terms into the convex sparsity terms. We replace the l_0 norm with l_1 norm using the convex relaxation. Thus, the Eq. (11) can be relaxed as:

$$\begin{aligned} \min_{\alpha, \mathbf{z}, \mathbf{e}} \quad & \frac{1}{4\lambda} \alpha^T \mathbf{C}(\mathbf{z}) \mathbf{C}(\mathbf{z})^T \alpha + \frac{1}{4} \alpha^T \alpha - \alpha^T \mathbf{y} + \beta \|\mathbf{e}\|_1 \\ \text{s.t.} \quad & \mathbf{x} = \mathbf{z} + \mathbf{e} \end{aligned} \quad (12)$$

Although the Eq. (12) seems complex and is not joint convex, subproblem of each variable is convex by fixing other variables and has a closed-form solution. Therefore, the model can be optimized by the ADMM (alternating direction method of multipliers) algorithm [2]. As demonstrated in the experiments, the parameters of Eq. (12) are easy to adjust, and the tracking performance is insensitive to parameter variations.

4.2 Optimization

In this section, we mainly introduce how to solve the objective function (12). Although the variables of the (12) are not joint convex, the subproblem of each variable with others fixed is convex and has a closed-form solution. The ADMM is a effective solver of the problems like (12). By introducing augmented Lagrange multipliers, the optimization function (12) can be written as the following augmented Lagrange function:

$$\begin{aligned} \mathcal{L}_{(\alpha, \mathbf{z}, \mathbf{e})} = \quad & \frac{1}{4\lambda} \alpha^T \mathbf{C}(\mathbf{z}) \mathbf{C}(\mathbf{z})^T \alpha + \frac{1}{4} \alpha^T \alpha - \alpha^T \mathbf{y} + \beta \|\mathbf{e}\|_1 \\ & + \frac{\mu}{2} \|\mathbf{x} - \mathbf{z} - \mathbf{e} + \frac{\mathbf{p}}{\mu}\|_2^2 - \frac{1}{2\mu} \|\mathbf{p}\|_2^2 \end{aligned} \quad (13)$$

where $\mu > 0$ is the penalty parameter and \mathbf{p} is the Lagrangian multipliers. The ADMM method updates one of the variables by minimizing \mathcal{L} with other variables fixed. By updating these variables iteratively, the convergence can be guaranteed [2]. Besides the Lagrangian multipliers \mathbf{p} , there are three variables that need to be updated, including $\alpha, \mathbf{z}, \mathbf{e}$. The closed form solution of each subproblems are as follows.

Update α (with others fixed): The optimization (13) with respect to the variable α can be formulated as follows:

$$\alpha = \underset{\alpha}{\operatorname{argmin}} \quad \frac{1}{4\lambda} \alpha^T \mathbf{C}(\mathbf{z}) \mathbf{C}(\mathbf{z})^T \alpha + \frac{1}{4} \alpha^T \alpha - \alpha^T \mathbf{y} \quad (14)$$

For calculating the variable α , we take the derivative of the α -subproblem (14) and set it to 0. The variable α has the closed-form solution. With some algebra, the closed-form solution of the variable α can be formulated as follows:

$$\alpha = \left(\frac{1}{2\lambda} \mathbf{C}(\mathbf{z}) \mathbf{C}(\mathbf{z})^T + \frac{1}{2} \mathbf{I} \right)^{-1} \mathbf{y} \quad (15)$$

where \mathbf{I} is the identity matrix. The amount of computation cost of Eq. (15) is large, mainly from matrix inverse and multiplication in spatial domain. For the

fast operation in Fourier domain, the property (4) of cyclic matrix is introduced into the solution (15). The variable α is updated with only the base sample as follows:

$$\hat{\alpha}^{k+1} = \frac{\hat{\mathbf{y}}}{\frac{1}{2\lambda}(\hat{\mathbf{z}}^*)^k \odot \hat{\mathbf{z}}^k + \frac{1}{2}} \quad (16)$$

where the fraction denotes the element-wise division. Finally, the α can be obtained via $\alpha = \mathcal{F}^{-1}(\hat{\alpha})$.

Update \mathbf{z} (with others fixed): The \mathbf{z} is updated through solving the subproblem (17) corresponding to \mathbf{z} with the closed-form solution.

$$\mathbf{z} = \underset{\mathbf{z}}{\operatorname{argmin}} \frac{1}{4\lambda} \alpha^T \mathbf{C}(\mathbf{z}) \mathbf{C}(\mathbf{z})^T \alpha + \frac{\mu}{2} \|\mathbf{x} - \mathbf{z} - \mathbf{e} + \frac{\mathbf{p}}{\mu}\|_2^2 \quad (17)$$

To solve and calculate efficiently in Fourier domain, the Eq. (17) is formulated as follows by Parsevaal's theorem:

$$\hat{\mathbf{z}} = \underset{\hat{\mathbf{z}}}{\operatorname{argmin}} \frac{1}{4\lambda} \hat{\alpha}^H \operatorname{diag}(\hat{\mathbf{z}}) \operatorname{diag}(\hat{\mathbf{z}}^*) \hat{\alpha} + \frac{\mu}{2} \|\hat{\mathbf{x}} - \hat{\mathbf{z}} - \hat{\mathbf{e}} + \frac{\hat{\mathbf{p}}}{\mu}\|_2^2 \quad (18)$$

where $\operatorname{diag}(\hat{\mathbf{z}})$ denotes the diagonal matrix of a vector $\hat{\mathbf{z}}$. With some algebra, the first term of Eq. (18) can be transformed as the following forms:

$$\hat{\mathbf{z}} = \underset{\hat{\mathbf{z}}}{\operatorname{argmin}} \frac{1}{4\lambda} (\hat{\alpha} \odot \hat{\mathbf{z}})^H (\hat{\alpha} \odot \hat{\mathbf{z}}) + \frac{\mu}{2} \|\hat{\mathbf{x}} - \hat{\mathbf{z}} - \hat{\mathbf{e}} + \frac{\hat{\mathbf{p}}}{\mu}\|_F^2 \quad (19)$$

where the derivative of $(\hat{\alpha} \odot \hat{\mathbf{z}})^H (\hat{\alpha} \odot \hat{\mathbf{z}})$ is $2(\hat{\alpha}^* \odot \hat{\alpha} \odot \hat{\mathbf{z}})$ with some algebra. The solution of $\hat{\mathbf{z}}$ is obtained by setting the derivative of (19) to 0.

$$\hat{\mathbf{z}}^{k+1} = \frac{\mu(\hat{\mathbf{x}} - \hat{\mathbf{e}}^k) + \hat{\mathbf{p}}^k}{\frac{1}{2\lambda}(\hat{\alpha}^*)^{k+1} \odot \hat{\alpha}^{k+1} + \mu^k} \quad (20)$$

where fraction denotes the element-wise division. Finally, the ‘‘clean’’ feature \mathbf{z} can be obtained using the formula: $\mathbf{z} = \mathcal{F}^{-1}(\hat{\mathbf{z}})$.

Update \mathbf{e} (with others fixed): The optimization (13) with respect to the variable \mathbf{e} is formulated as follows:

$$\mathbf{e} = \underset{\mathbf{e}}{\operatorname{argmin}} \beta \|\mathbf{e}\|_1 + \frac{\mu}{2} \|\mathbf{x} - \mathbf{z} - \mathbf{e} + \frac{\mathbf{p}}{\mu}\|_2^2 \quad (21)$$

The noise \mathbf{e} is obtained by the soft-thresholding (or shrinkage) method [4] with closed-form solution:

$$\mathbf{e}^{k+1} = \mathcal{S}_{\frac{\beta}{\mu^k}}(\mathbf{x} - \mathbf{z}^{k+1} + \frac{\mathbf{p}^k}{\mu^k}) \quad (22)$$

where $\mathcal{S}_{\frac{\beta}{\mu}}(\mathbf{x})$ is the soft-thresholding operator for a vector \mathbf{x} with parameter $\frac{\beta}{\mu}$. Here, the $\mathcal{S}_{\frac{\beta}{\mu}}(\mathbf{x})$ can be calculated through the Eq. (23).

$$\mathcal{S}_{\frac{\beta}{\mu}}(\mathbf{x}) = (\operatorname{sign}(\mathbf{x})) \odot (\max(0, |\mathbf{x}| - \frac{\beta}{\mu})) \quad (23)$$

Algorithm 1. Optimization Procedure to Equation(13)

Require:

The object feature matrix \mathbf{x} , and the parameter β, λ , and μ ;
 Set $\mathbf{z}^0 = \mathbf{x}$, $\mathbf{p}^0 = \mathbf{e}^0 = \mathbf{0}$, $\alpha^0 = \mathbf{1}$, $\mu_0 = 5$, $\mu_{max} = 20$, $\rho = 3$, $\tau = 10^{-10}$,
 $maxIter = 3$, and $k = 0$.

Ensure: α, \mathbf{z} , and \mathbf{e} .

- 1: **while** not converged **do**
 - 2: Update \mathbf{z}^{k+1} by Equation(20);
 - 3: Update \mathbf{e}^{k+1} by Equation(22);
 - 4: Update α^{k+1} by Equation(16);
 - 5: Update Lagrange multipliers as followings:
 - 6: $\mathbf{p}^{k+1} = \mathbf{p}^k + \mu(\mathbf{x}^{k+1} - \mathbf{z}^{k+1} - \mathbf{e}^{k+1})$;
 - 7: Update μ_{k+1} by $\mu_{k+1} = \min(\mu_{max}, \rho\mu_k)$;
 - 8: Update k by $k = k + 1$;
 - 9: Check the convergence condition: the maximum element changes of \mathbf{z}, \mathbf{e} , and α between two consecutive iterations are less than τ or the maximum number of iterations reaches $maxIter$.
 - 10: **end while**
-

Besides the above variables, the Lagrange multiplier \mathbf{y} is updated by following the Eq. (24):

$$\mathbf{p}^{k+1} = \mathbf{p}^k + \mu^k(\mathbf{x} - \mathbf{z}^{k+1} - \mathbf{e}^{k+1}) \quad (24)$$

Since each subproblem of (13) is convex, we can guarantee that the limit point by our algorithm satisfies the Nash equilibrium conditions [35]. The details of optimization procedure are shown in Algorithm 1.

4.3 Tracking

In this section, we briefly introduce the process of the localization and update steps of the proposed algorithm.

Localization. Features are extracted from the searching area in current frame. We employ the learned variables \mathbf{z}, α in previous frame to locate the object in current frame. The response map can be obtained as follows:

$$\phi = \mathcal{F}^{-1}(\hat{\mathbf{k}}^{\hat{\mathbf{x}}\hat{\mathbf{z}}} \odot \hat{\alpha}) \quad (25)$$

where $\hat{\mathbf{k}}^{\hat{\mathbf{x}}\hat{\mathbf{z}}}$ denotes the kernel between $\hat{\mathbf{z}}$ and $\hat{\mathbf{z}}$, and the $\hat{\mathbf{z}}$ represents the learned target appearance model using the “clean” feature, the $\hat{\mathbf{x}}$ denotes the feature in current frame. We locate the tracked object using the response map.

Learning-Update. The searching area is extracted in current frame. The extracted feature from searching area are feed into the Eq. (13) to optimize the dual variable and the “clean” feature. For capturing the variation of tracked target’s appearance, the dual variable and the target appearance model are updated

by an autoregressive model with learning rate. The updated strategy of dual variable α and appearance model $\hat{\mathbf{z}}$ follows the following formulation.

$$\begin{aligned}\hat{\alpha}^t &= (1 - \eta)\hat{\alpha}^{t-1} + \eta\hat{\alpha} \\ \hat{\mathbf{z}}^t &= (1 - \eta)\hat{\mathbf{z}}^{t-1} + \eta\hat{\mathbf{z}}\end{aligned}\tag{26}$$

where η is learning rate. According to the above solution procedure, it does not refer to the large amount of computation operations like matrix inverse.

5 Experiments

In this section, we mainly introduce the implementation details of our model, evaluation datasets and evaluation metrics, and the analysis of experimental results compared with baseline and other state-of-the-art trackers in public benchmark.

5.1 Experimental Setup

Implementation Details and Parameters: To demonstrate the effectiveness of our model, we select the two different baseline trackers to implement the model. They are dual correlation filter methods (DCF) [15] based on hand-craft features and HCF [25] based on the deep features. The two baseline models both employ the dual model to optimize the problem. Firstly, the implementation details of embedding our model in DCF (DCF_{our}) are introduced. Standard HOG [10] descriptors are used to represent the object in DCF. We only embedded the feature decomposition model in DCF to jointly learn noise-aware filters and the “clean” feature for improving tracking performance. The β and λ in Eq. (13) is set to 0.005 and 0.1, respectively.

Next, we mainly introduce the implementation details of embedding our model in HCF. Ma *et al.* [25] extract three layers deep features from VGGNet [27], that are conv4, conv3-4 and conv5-4. The three convolution features independently learn the filter and the appearance model. The final response map is fused by three different response maps obtained by the three convolution features. We only implement noise-aware correlation filter model based on different convolution layers. The β and λ in Eq. (13) is set to 0.01 and 1e-4, respectively. The experiments are carried out on a PC with an Intel i7 4.2 GHz CPU and 32G RAM.

As seen from the above section, the parameter settings for DCF_{our} and HCF_{our} have great discrepancy, but this discrepancy is considered reasonable because DCF_{our} and HCF_{our} are different models. All parameters are optimal by varying them on a certain scope. Moreover, when we slightly adjust the parameters, tracking performance only change a little and Table 1 shows the results of the proposed method DCF_{our} and HCF_{our} with different parameters.

Datasets and Evaluation Metrics: Our method is evaluated on benchmark dataset: OTB-2013 [34] with 50 sequences. The images are annotated with

Table 1. The precision rate (PR) of the proposed method DCF_{our} and HCF_{our} with different parameters.

	Param	Setting	PR	Param	Setting	PR		Param	Setting	PR	Param	Setting	PR
HCF_{our}	λ	1e-2	0.888	β	0.005	0.893	DCF_{our}	λ	0.05	0.724	β	0.001	0.728
		1e-3	0.890		0.01	0.895			0.1	0.739		0.005	0.739
		1e-4	0.895		0.02	0.892			0.2	0.735		0.01	0.728

ground truth bounding boxes and various visual attributes. For the OTB-2013 dataset, we employ the one-pass evaluation (OPE) and use two metrics: precision rate (PR) and success rate (SR). PR is the percentage of frames whose output location is within the given threshold distance of ground truth. That is to say, it computes the average Euclidean distance between the center locations of the tracked target and the manually labeled ground-truth positions of all the frames. SR is the ratio of the number of successful frames whose overlap between predicted and ground truth bounding box is larger than a threshold. In the legend, we report the area under curve (AUC) of success plot and precision score at 20 pixels threshold corresponding to the one-pass evaluation for each tracking method.

Compared Trackers: To identify the effectiveness of our model, we evaluate the proposed model with comparisons to several state-of-the-art methods for evaluations. Several trackers based on correlation filter are selected to evaluate the performance, including DSST [5], KCF [15], CSR-DCF [24], SRDCF [7] and SAMF [23]. In addition, we also select several representative trackers to compare with our methods, Struck [13], TLD [16], SCM [36].

Table 2. The Mean FPS compared with DCF, KCF, DSST, SAMF, CSR-DCF trackers.

	DCF_{our}	DCF	KCF	DSST	SAMF	SRDCF	CSR-DCF
Mean FPS	77.52	564.64	374.16	60.30	11.49	7.81	18.18

5.2 Tracking Speed

The tracking speed is crucial in many realistic tracking applications. We therefore generalize the tracking speed about DCF and DCF_{our} in Table 2. In addition, Table 2 enumerates other trackers based on correlation filter, including DCF, KCF, DSST, SAMF, SRDCF, CSR-DCF. It is verified that DCF_{our} performs at about 77.52 FPS (frames per second) to achieve real-time tracking (equivalent to approximately 20 FPS) although it achieves the lower tracking speed than DCF.

RPCA [3] has a powerful capability to suppress noises or corruptions. RPCA is a NP-hard problem because it simultaneously involves $rank$ and ℓ_0 . To overcome these obstacles, a convex relaxation of the problem is proposed, we use $\|\cdot\|_*$ and ℓ_1 to replace $rank$ and ℓ_0 , respectively. It also should be note that although the convex relaxations are leveraged, the obtained problem is still non-convex. Fortunately, this problem is convex with respect to each of them when others are fixed. ADMM algorithm [2] has proven to be an efficient and effective solver of RPCA. However, it involves SVD in the process of iterative solution using ADMM. SVD has high computational complexity and extremely time consuming. Although the proposed method is also optimized by ADMM, SVD is not involved in the optimization.

Table 3. The precision rate (PR%) and success rate (SR%) on OTB-2013 over DCF, DCF_{our}, HCF, and HCF_{our}, DSST, SAMF, SRDCF, CSR-DCF, Struct, TLD, and SCM

	DCF	DCF _{our}	HCF	HCF _{our}	DSST	KCF	SAMF	SRDCF	CSR-DCF	Struct	TLD	SCM
PR	72.8	73.9	89.1	89.5	73.7	74.0	82.3	82.8	82.3	65.6	60.8	64.9
SR	50.8	51.4	63.5	65.4	55.4	51.4	60.5	60.8	59.6	55.9	52.1	61.6

5.3 Comparison with Baseline Methods

We compare our methods with the baseline trackers to demonstrate the strength of our proposed methods in this part.

Overall Evaluation: To illustrate the effectiveness of our proposed model, we implement the model in DCF and HCF, named DCF_{our} and HCF_{our}, respectively. Table 3 shows the overall evaluation results between DCF, HCF and DCF_{our}, HCF_{our} in OTB-2013. Benefiting from our proposed model, the DCF_{our} outperforms DCF in PR/SR by 1.1%/0.6% and 0.4%/1.9%, respectively. HCF_{our} achieves slight improvement over HCF in PR. But HCF_{our} exceeds 1.9% in SR over HCF. This evaluative performance illustrates that the feature decomposition scheme can improve the CF tracking performance. Table 4 represents tracking performance on several challenging factors between our trackers and the baseline trackers.

Attribute-Based Evaluation: We further analyze the tracking performance under different challenging attributes (*e.g.*, background clutter (BC), occlusion (OCC), fast motion (FM)) annotated in the benchmark OTB-2013. Table 4 shows PR and SR of one-pass evaluation (OPE) for ten main challenging factors. We mainly take DCF_{our} and DCF as an example to analysis the evaluative results. According to Table 4, we have the following observations.

First, our model is effective in handling motion blur (MB). In general, MB results in appearance degradation. And how to learn a good feature is critical to

Table 4. Attributed-based PR/SR on OTB-2013 compared with DCF, HCF, SRDCF, CSR-DCF, SAMF, KCF, DSST trackers.

Trackers	FM	BC	SV	MB	IV	OCC	LR	OPR	DEF	IPR
DCF	55.9/44.1	71.9/52.2	65.4/41.6	58.8/47.0	69.9/48.1	72.6/50.2	71.2/48.8	63.2/54.4	74.0/53.1	70.4/48.7
DCF _{our}	59.9/45.4	75.8/53.8	67.5/42.4	65.0/49.2	72.5/49.1	74.4 /51.2	37.8/31.0	72.7/49.5	74.1/53.5	72.8/50.1
HCF	78.4/58.3	88.3/63.4	88.3/60.0	84.6/62.4	84.3/59.9	87.8/64.1	89.7/60.2	86.8/61.5	88.0/65.9	86.7/60.1
HCF _{our}	79.0/61.4	85.6/62.8	88.3/62.9	80.5/61.9	85.7/ 61.8	88.1/66.4	87.5/62.3	87.5/63.4	87.3/65.1	85.6/61.2
SRDCF	77.2/58.4	81.4/58.8	77.7/57.5	78.2/59.2	79.0/58.6	83.7/60.9	40.4/33.8	83.3/60.0	84.9/62.8	74.8/54.0
CSR-DCF	68.8/53.0	77.4/55.4	71.2/51.7	74.6/57.4	75.5/56.1	78.7/58.4	40.2/32.3	81.3/57.7	87.5/64.1	78.2/55.7
SAMF	68.2/53.5	72.7/55.4	78.7/55.7	67.2/53.9	69.6/52.0	85.3/62.1	55.4/43.3	79.7/57.5	75.2/58.4	75.4/54.8
KCF	60.2/46.0	75.3/53.5	67.9/42.7	65.0/49.7	72.8/49.4	74.9/51.4	38.1/31.2	72.9/49.6	74.0/53.4	72.5/49.7
DSST	51.7/43.5	69.4/51.7	73.0/54.1	54.7/46.4	73.5/56.3	71.6/53.4	49.7/40.9	73.3/53.5	66.0/51.0	76.5/56.0

address the MB challenge. The DCF_{our} achieves superior performance in PR/SR by 6.2%/2.2% over DCF. In comparison with the baseline, the excellent performance in MB illustrates DCF_{our} can learn the “clean” feature to better represent objects. Second, our method also achieve excellent performance in handling BC and OCC. BC and OCC lead to features to be disturbed by the background. As shown in Table 4, DCF_{our} outperforms DCF in PR/SR in 3.9%/1.6% and 1.8%/1.0% over BC and OCC, respectively. It demonstrates that the learned “clean” feature and noise-aware filter help to suppress the interference of noises. Finally, our method also represents the powerful strength on scale variation (SV) and illumination variation (IV). The critical important point is how to capture the variation of objects caused by the two challenging factors. For SV and IV, DCF_{our} trackers both achieve the superior performance in PR/SR over baseline. Compared the baseline, the DCF_{our} promotes 2.1%/0.8% and 2.6%/1.0% in SV and IV. It demonstrates our model can capture the variation of objects to learn the appearance model and the robust filter.

In addition, our method also achieves excellent performance in other challenging factors (*fast motion (FM)*, *low Resolution (LR)*) and several factors (*in-plane rotation (LPR)*, *out-of-plane rotation (OPR)*). In summary, the learned feature can better represent the object and optimize the filter.

5.4 Comparison with State-of-the-Art Trackers

Overall Evaluation: For comprehensive evaluation, we select several state-of-the-start trackers, including DSST [5], KCF [15], CSR-DCF [24], SRDCF [7], SAMF [23], Struck [13], TLD [16] and SCM [36] to compare with our proposed methods. The Table 3 represents the evaluation results between our proposed model and comparison trackers. As shown in Table 3, DCF_{our} outperforms DSST in PR/SR while DCF achieves lower performance than the DSST in PR/SR. DCF_{our} achieves comparable performance in PR/SR against KCF. This observation strongly illustrates that the proposed feature decomposition can help to improve the performance of CF trackers.

Attribute-Based Evaluation: We also select several trackers, including DSST [5], KCF [15], CSR-DCF [24], SRDCF [7] and SAMF [23] to evaluate the performance on different challenging attributes. Table 4 shows PR and SR of one-pass evaluation (OPE) for ten main challenging factors in OTB-2013 benchmark, including fast motion (FM), background clutter (BC), motion blur (MB), deformation (DEF), illumination variation (IV), low resolution (LR), occlusion (OCC) and scale variation (SV), respectively.

In this experimental analysis, we mainly focus on DCF and DCF_{our} to analyze the performance of this proposed algorithm. According to Table 4, we have the following observations. The first conclusion is that the proposed feature decomposition can benefit the CF trackers to learn a “clean” feature to deal with the MB. In OTB-2013, DCF is lower than DSST in PR but the DCF_{our} is greater than DSST and even outperforms KCF in PR. Although the performance of DCF_{our} is still lower than SAMF, CSR-DCF, SRDCF, HCF, it is explained that DCF_{our} is only based on the DCF to incorporate feature decomposition without any improvement while these extension trackers employ several tricks such as rich features and updating strategy besides their own improvement. The second conclusion is that our method also help to improvement the tracker’s performance in BC and OCC. According to Table 4, DCF_{our} even outperforms the SAMF, KCF and DSST in PR for BC. Finally, it is observed that our method also improves the strength in handling SV and IV. In OTB-2013 benchmark, the performance of DCF_{our} outperforms the DSST in PR/SR while the performance of DCF_{our} is still lower than these trackers that have scale processing, including DSST, SAMF, SRDCF, CSR-DCF, HCF. Because the DCF_{our} have no the scale process model and only mixes the feature decomposition scheme to learn “clean” feature and noise-aware filter. In summary, the proposed feature decomposition learns more robust filter and “clean” feature to improve the performance.

6 Conclusion

In this paper, feature decomposition is introduced into correlation filter tracking to learn “clean” feature and noise-aware filter for improve the tracking accuracy and robustness. The “clean” feature and noise-aware filter are jointly optimized in an unified framework to mitigate noises effect in filters and features. The proposed tracking framework utilizes the learned “clean” feature to represent objects and the noise-aware filter to classify the object from background. As a result, it has the advantages of several existing correlation filter trackers such as suppressing the influence of noises. Both qualitative and quantitative evaluations on challenging datasets demonstrate that the effectiveness of proposed tracking algorithm against baseline methods.

Acknowledgment. This work is jointly supported by National Natural Science Foundation of China (61702002, 61472002), China Postdoctoral Science Foundation, Natural Science Foundation of Anhui Province (1808085QF187), Natural Science Foundation of Anhui Higher Education Institution of China (KJ2017A017), and Co-Innovation Center for Information Supply & Assurance Technology, Anhui University.

References

1. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 2544–2550. IEEE (2010)
2. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends® Mach. Learn.* **3**(1), 1–122 (2011)
3. Candès, E.J., Li, X., Ma, Y., Wright, J.: Robust principal component analysis? *J. ACM (JACM)* **58**(3), 11 (2011)
4. Chen, M., Ganesh, A., Lin, Z., Ma, Y., Wright, J., Wu, L.: Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix. *Coordinated Science Laboratory Report no. UILU-ENG-09-2214* (2009)
5. Danelljan, M., Hager, G., Khan, F., Felsberg, M.: Accurate scale estimation for robust visual tracking. In: Proceedings of British Machine Vision Conference. BMVA Press (2014)
6. Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: ECO: efficient convolution operators for tracking. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. IEEE (2017)
7. Danelljan, M., Hager, G., Shahbaz Khan, F., Felsberg, M.: Learning spatially regularized correlation filters for visual tracking. In: Proceedings of IEEE International Conference on Computer Vision, pp. 4310–4318. IEEE (2015)
8. Danelljan, M., Robinson, A., Shahbaz Khan, F., Felsberg, M.: Beyond correlation filters: learning continuous convolution operators for visual tracking. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9909, pp. 472–488. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46454-1_29
9. Danelljan, M., Shahbaz Khan, F., Felsberg, M., Van de Weijer, J.: Adaptive color attributes for real-time visual tracking. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1090–1097. IEEE (2014)
10. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(9), 1627–1645 (2010)
11. Galoogahi, H.K., Fagg, A., Lucey, S.: Learning background-aware correlation filters for visual tracking. In: Proceedings of IEEE Conference on Computer Vision, pp. 1144–1152. IEEE (2017)
12. Gray, R.M.: Toeplitz and circulant matrices: a review. *Found. Trends® Commun. Inf. Theory* **2**(3), 155–239 (2006)
13. Hare, S., Saffari, A., Torr, P.H.: Struck: structured output tracking with kernels. In: Proceedings of IEEE International Conference on Computer Vision, pp. 263–270. IEEE (2011)
14. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7575, pp. 702–715. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33765-9_50
15. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(3), 583–596 (2015)
16. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(7), 1409–1422 (2012)

17. Li, C., Cheng, H., Hu, S., Liu, X., Tang, J., Lin, L.: Learning collaborative sparse representation for grayscale-thermal tracking. *IEEE Trans. Image Process.* **25**(12), 5743–5756 (2016)
18. Li, C., Liang, X., Lu, Y., Zhao, N., Tang, J.: RGB-T object tracking: benchmark and baseline. arXiv preprint [arXiv:1805.08982](https://arxiv.org/abs/1805.08982) (2018)
19. Li, C., Lin, L., Zuo, W., Tang, J.: Learning patch-based dynamic graph for visual tracking. In: *Proceedings of The AAAI Conference on Artificial Intelligence*, pp. 4126–4132. AAAI (2017)
20. Li, C., Lin, L., Zuo, W., Tang, J., Yang, M.H.: Visual tracking via dynamic graph learning. *IEEE TPAMI* (2018). <https://doi.org/10.1109/TPAMI.2018.2864965>
21. Li, C., Wu, X., Bao, Z., Tang, J.: ReGLE: spatially regularized graph learning for visual tracking. In: *Proceedings of the ACM on Multimedia Conference*, pp. 252–260. ACM (2017)
22. Li, C., Zhao, N., Lu, Y., Zhu, C., Tang, J.: Weighted sparse representation regularized graph learning for RGB-T object tracking. In: *Proceedings of the ACM on Multimedia Conference*, pp. 1856–1864. ACM (2017)
23. Li, Y., Zhu, J.: A scale adaptive kernel correlation filter tracker with feature integration. In: Agapito, L., Bronstein, M.M., Rother, C. (eds.) *ECCV 2014*. LNCS, vol. 8926, pp. 254–265. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16181-5_18
24. Lukezic, A., Vojir, T., Cehovin, L., Matas, J., Kristan, M.: Discriminative correlation filter with channel and spatial reliability. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE (2017)
25. Ma, C., Huang, J.B., Yang, X., Yang, M.H.: Hierarchical convolutional features for visual tracking. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3074–3082. IEEE (2015)
26. Rifkin, R., Yeo, G., Poggio, T.: Regularized least-squares classification. *Nato Sci. Ser. Sub Ser. III Comput. Syst. Sci.* **190**, 131–154 (2003)
27. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: *CoRR* (2014)
28. Sui, Y., Tang, Y., Zhang, L.: Discriminative low-rank tracking. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3002–3010. IEEE (2015)
29. Sun, C., Wang, D., Lu, H., Yang, M.H.: Correlation tracking via joint discrimination and reliability learning. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 489–497. IEEE (2018)
30. Tang, M., Feng, J.: Multi-kernel correlation filter for visual tracking. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3038–3046. IEEE (2015)
31. Tang, M., Yu, B., Zhang, F., Wang, J.: High-speed tracking with multi-kernel correlation filters. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4874–4883. IEEE (2018)
32. Valmadre, J., Bertinetto, L., Henriques, J., Vedaldi, A., Torr, P.H.: End-to-end representation learning for correlation filter based tracking. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5000–5008. IEEE (2017)
33. Van De Weijer, J., Schmid, C., Verbeek, J., Larlus, D.: Learning color names for real-world applications. *IEEE Trans. Image Process.* **18**(7), 1512–1523 (2009)
34. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: a benchmark. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2411–2418. IEEE (2013)

35. Xu, Y., Yin, W.: A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM J. Imaging Sci.* **6**(3), 1758–1789 (2013)
36. Zhong, W., Lu, H., Yang, M.H.: Robust object tracking via sparsity-based collaborative model. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1838–1845. IEEE (2012)