

A Method of General Acceleration SRDCF Calculation via Reintroduction of Circulant Structure

Xiaoxiang Hu^(✉) and Yujiu Yang

Shenzhen Key Laboratory of Broad-band Network and Multimedia,
Graduate School at Shenzhen, Tsinghua University, Beijing, China
huxx15@mails.tsinghua.edu.cn, yang.yujiu@sz.tsinghua.edu.cn

Abstract. Discriminatively learned correlation filters (DCF) have been widely used in online visual tracking field due to its simplicity and efficiency. These methods utilize a periodic assumption of the training samples to construct a circulant data matrix, which also introduces unwanted boundary effects. Spatially Regularized Correlation Filters (SRDCF) solved this issue by introducing penalization on correlation filter coefficients. However, which breaks the circulant structure used in DCF. We propose Faster SRDCF (FSRDCF) via reintroduction of circulant structure. The circulant structure of training samples in the spatial domain is fully used, more importantly, we exploit the circulant structure of regularization function in the Fourier domain, which allows the problem to be solved more directly and efficiently. Our approach demonstrates superior performance over other non-spatial-regularization trackers on the OTB2013 and OTB2015.

Keywords: Object tracking · Correlation filter · Circulant structures

1 Introduction

Visual tracking is one of the core problems in the field of computer vision with a variety of applications. Generic visual tracking is to estimate the trajectory of a target in an image sequence, given only its initial state. It is difficult to design a fast and robust tracker from a very limited set of training samples due to various critical issues in visual tracking, such as occlusion, fast motion and deformation.

Recently, Discriminative Correlation Filter (DCF) [1] has been widely used in visual tracking because of its simplicity and efficiency, and there are many improvements [2, 5–7, 12] about DCF to address the above mentioned problems. These methods learn a correlation filter from a set of training samples to encode the targets appearance. Nearly all correlation filter based trackers utilize the circulant structure of training samples proposed in work [10]. The circulant structure allows the correlation filter training and target detection computation efficiently. However, this structure also introduces unwanted boundary effects that

leads to an inaccurate appearance mode. To address the boundary effects problem, Danelljan et al. propose Spatially Regularized Correlation Filters (SRDCF) [6]. The SRDCF introduces penalization to force the correlation filters to concentrate on center of the training patches. This penalization allows the tracker to be trained on a larger area without the effect of background, so the SRDCF can handle some challenging cases such as fast target motion. However, the penalization makes the correlation filters complex to solve, which is unacceptable for an online visual tracking situation. In this work, we revisit the SRDCF, in our formulation, we make full use of the circulant structure to simplify the problem.

In this paper, we propose Faster Spatially Regularized Discriminative Correlation Filters (FSRDCF) for tracking. The circulant structure of training data matrix in the spatial domain is utilized, besides, we exploit the circulant structure of regularization matrix in the Fourier domain. Our approach more computation efficient without any significant degradation in performance and more suitable for online tracking problems. To validate our approach, we preform comprehensive experiments on the most popular benchmark datasets: OTB-2013 [18] with 50 sequences and OTB-2015 [19] with 100 sequences. Our approach obtains a more than twice faster running speed and faster startup time than the baseline tracker SRDCF and achieves state-of-the-art performance over other non-spatial-regularization trackers.

2 Spatially Regularized DCF

After Bolme et al. [1] first introduced the MOSSE filter, lots of notable improvements [5–7, 12, 13] are proposed from different aspects to strengthen the correlation filter based trackers. New features have been widely used, such as HOG [4], Color-Name [17] and deep features [13, 15]; feature integration is also used [11]. To address occlusion, part-based trackers [12] are widely adapted. However, the periodic assumption also produced unwanted boundary effects. Galoogahi et al. [8] investigate the boundary effect issue, their method removes the boundary effects by using a masking matrix to allow the size of training patches larger than correlation filters. They use Alternative Direction Method of Multipliers (ADMM) to solve their problem and have to make transitions between spatial and Fourier domain in every ADMM iteration, which increasing the trackers computational complexity. To get rid of those transitions, Danelljan et al. [6] propose the spatially Regularized Correlation filters (SRDCF), they introduce a spatial weight function to penalize the magnitude of the correlation filter coefficients, and use Gauss-Seidel method to solve the filters, in this way, both the boundary effects and the transitions in [8] are avoided. The work [3] also use a spatial regularization like the SRDCF and derived a simplified inverse method to get a closedform solution. However, these methods still have a relatively high computational complexity. In our proposal, we apply a spatial regularization like [3, 6], by exploiting circulant structure of train samples in the spatial domain and regularization matrix in the Fourier domain, our formulation has no problem transition, correlation filters needed are solved directly.

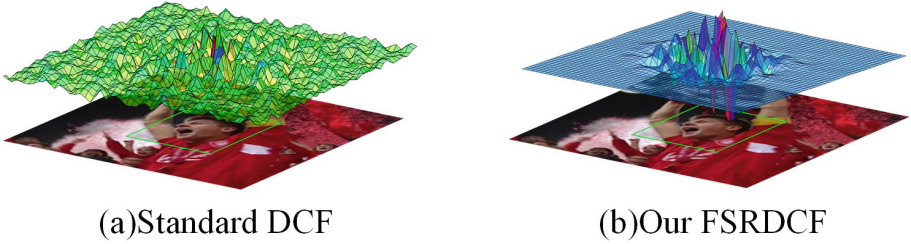


Fig. 1. Visualization of the filter coefficients trained by the standard DCF (a) and our FSRDCF (b). The top layer is the learned filter corresponding to the bottom layer training patch. Target is outlined by the green rectangle. As we can see, our filter puts more attention on the object than the standard DCF.

Standard SRDCF Training and Detection. The way to handle the boundary effects in the SRDCF is most popular in literature. So we give some details of the convolution filters training and Detection after introduced the regularization. In this section, we use the term convolution, because the SRDCF are modeled with convolution instead of correlation, we will give some key differences used in our proposal in Sects. 3 and 4. Convolution filters are learned from a set of training samples $\{(x_k, y_k)\}_{k=1}^t$. Every training sample $x_k \in \mathbb{R}^{d \times M \times N}$ consists of a d -channel feature map with spatial size of $M \times N$ extracted from a training image patch. We use x_k^l to represent the l th feature layer of x_k . y_k is the optimal convolution output corresponding to training sample x_k . The Spatially Regularized Correlation Filters (SRDCF) is obtained from the convex problem,

$$\min_f \sum_{k=1}^t \alpha_k \|S_f(x_k) - y_k\|^2 + \sum_{l=1}^d \|w \odot f^l\|^2. \tag{1}$$

where the $\alpha_k \geq 0$ is the weight of every training sample x_k , spatial regularization is introduced by w , which is a Gaussian shaped function with smaller values in center area and bigger values in marginal area, \odot denotes the element-wise multiplication. $S_f(x_k)$ is the convolution function,

$$S_f(x_k) = \sum_{l=1}^d x_k^l * f^l. \tag{2}$$

where $*$ denotes the circular convolution. With the use of Parseval’s theorem and convolution property, the Eq. 1 is transformed into Fourier domain,

$$\min_{\hat{f}} \sum_{k=1}^t \alpha_k \left\| \sum_{l=1}^d \hat{x}_k^l \odot \hat{f}^l - \hat{y}_k \right\|^2 + \sum_{l=1}^d \left\| \frac{\hat{w}}{MN} * \hat{f}^l \right\|^2. \tag{3}$$

where the hat denotes Discrete Fourier Transformed (DFT) of a variable. For convenience, all variables in Eq. 3 are vectorized, convolution is transformed into matrix multiplication,

$$\min_{\mathbf{f}} \sum_{k=1}^t \alpha_k \left\| \sum_{l=1}^d \mathcal{D}(\hat{\mathbf{x}}_k^l) \hat{\mathbf{f}}^l - \hat{\mathbf{y}}_k \right\|^2 + \sum_{l=1}^d \left\| \frac{\mathcal{C}(\hat{\mathbf{w}})}{MN} \hat{\mathbf{f}}^l \right\|^2. \tag{4}$$

Here, bold letters are the corresponding variables' vectorization form, $\mathcal{D}(\mathbf{v})$ is a diagonal matrix with the elements of the vector in its diagonal. $\mathcal{C}(\hat{\mathbf{w}})$ is a matrix with its rows consist of all of the shift of the vector. Equation 4 is a complex convex problem, because the DFT of a real-valued function is Hermitian symmetric, so the convex problem (4) can transformed into a re-al-valued one by a unitary matrix $B \in \mathbb{R}^{MN \times MN}$,

$$\min_{\tilde{\mathbf{f}}} \sum_{k=1}^t \alpha_k \left\| \sum_{l=1}^d D_k^l \tilde{\mathbf{f}}^l - \tilde{\mathbf{y}}_k \right\|^2 + \sum_{l=1}^d \left\| C \tilde{\mathbf{f}}^l \right\|^2. \tag{5}$$

Here, $D_k^l = BD(\hat{\mathbf{x}}_k^l)B^H$, $\tilde{\mathbf{f}}^l = B\hat{\mathbf{f}}^l$, $\tilde{\mathbf{y}}_k = B\hat{\mathbf{y}}_k$ and $C = \frac{1}{MN}BC(\hat{\mathbf{w}})B^H$, where the H denotes the conjugate transpose of a matrix. Then concatenate all layers of training data and convolution filters, in other words, $\tilde{\mathbf{f}} = ((\tilde{\mathbf{f}}^1)^T, \dots, (\tilde{\mathbf{f}}^d)^T)^T$ and $D_k = (D_k^1, \dots, D_k^d)$, Eq. 5 is simplified as,

$$\min_{\tilde{\mathbf{f}}} \sum_{k=1}^t \alpha_k \left\| D_k \tilde{\mathbf{f}} - \tilde{\mathbf{y}}_k \right\|^2 + \left\| W \tilde{\mathbf{f}} \right\|^2. \tag{6}$$

where $W \in \mathbb{R}^{dMN \times dMN}$ is a block diagonal matrix with its diagonal blocks being equal to C . Letting the derivative of Eq. 6 with respected to $\tilde{\mathbf{f}}$ be zero,

$$\left(\sum_{k=1}^t \alpha_k D_k^H D_k + W^H W \right) \tilde{\mathbf{f}} = \sum_{k=1}^t \alpha_k D_k^H \tilde{\mathbf{y}}_k. \tag{7}$$

Due to the sparsity of D_k and W , problem (7) can be efficiently solved by Gauss-Seidel method with the computational complexity of $\mathcal{O}((d + K^2)dMNN_{GS})$, where the K is the number of non-zero entries in $\hat{\mathbf{w}}$, the N_{GS} is the number of Gauss-Seidel iterations.

Excluding the feature extraction, the total computational complexity of SRDCF tracker is $\mathcal{O}(dSMN \log(MN) + SMN N_{NG} + (d + K^2)dMNN_{GS})$. Here, S denotes the number of scales in the scaling pool, N_{NG} is the number of Newton iterations in sub-grid detection. It's worth noting that the result takes none of the transformations into consideration, especially from Eqs. 4 to 5, which including high dimensional matrix multiplication. In reality, those transformations are time consuming. In our approach, all of them will be by-passed. We'll directly get the correlation filters in Eq. 8.

3 Faster SRDCF

We revisit spatially regularization correlation filters for tracking from Ridge Regression viewpoint. In our proposal, problem is solved more directly by exploiting both circulant structure in training data and regularization function.

Our proposal is to find a function $g(z) = \mathbf{f}^T z$ to minimize the squared error over all training samples x_k (d-channel feature map) and their regression target y_k ,

$$\min_{\mathbf{f}} \sum_{k=1}^t \alpha_k \left\| \sum_{l=1}^d X_k^l \mathbf{f}^l - \mathbf{y}_k^l \right\|^2 + \sum_{l=1}^d \|\mathcal{D}(\mathbf{w}) \mathbf{f}^l\|^2 \tag{8}$$

For simplicity, we let $\mathbf{y}_k^l = \mathbf{y}_k$. In general Ridge Regression problem, each row of X_k^l is a vectorized training sample, here, rows of X_k^l consist of all circular shift of \mathbf{x}_k^l , $X_k^l \mathbf{f}^l$ is the correlation between \mathbf{x}_k^l and \mathbf{f}^l , it's worth noting that $X_k^l \mathbf{f}^l \neq \text{vec}(\mathbf{x}_k^l * \mathbf{f}^l)$, where $\text{vec}(v) = \mathbf{v}$. Now we can directly take derivative of Eq. 1 with respect to \mathbf{f} and let the derivative be zero, then we get,

$$\sum_{l=1}^d \left(X_k^{lH} X_k^l + \mathcal{D}(\mathbf{w})^H \mathcal{D}(\mathbf{w}) \right) \mathbf{f}^l = \sum_{l=1}^d X_k^{lH} \mathbf{y}_k^l \tag{9}$$

Because all variables in Eq. 8 are real-valued, so we use $(*)^T$ and $(*)^H$ equivalently. Due to the circulant structure of X_k^l , we have,

$$X_k^l = \ddot{\mathbf{F}} \mathcal{D}(\hat{\mathbf{x}}_k^l) \ddot{\mathbf{F}}^H \tag{10}$$

where $\ddot{\mathbf{F}} = \mathbf{F} \otimes \mathbf{F}$ is two-dimensional DFT matrix for vectorized two-dimensional signals, \mathbf{F} is known as DFT matrix, \otimes denotes the Kronecker product. Both $\ddot{\mathbf{F}}$ and \mathbf{F} are constant matrix and unitary. We apply Eqs. 10 to 9,

$$\sum_{l=1}^d \mathbf{F} \left(\mathcal{D}(\hat{\mathbf{x}}_k^{l*} \odot \hat{\mathbf{x}}_k^l) + \mathbf{F}^H \mathcal{D}(\mathbf{w})^H \mathcal{D}(\mathbf{w}) \mathbf{F} \right) \mathbf{F}^H = \sum_{k=1}^t \sum_{l=1}^d \mathbf{F} \mathcal{D}(\hat{\mathbf{x}}_k^{l*}) \ddot{\mathbf{F}}^H \mathbf{y}_k^l \tag{11}$$

We can see that because of the introduction of regularization w , a simple closed-form solution can't be obtained from Eq. 11 like the way in work [10]. So far, we only use the circulant structure of training data matrix X_k^l in the spatial domain. From now on, we will further exploit the circulant structure of the regularization matrix $\mathcal{D}(\mathbf{w})$.

From Eq. 10, we can know that a spatially circulant matrix can be diagonalized by the matrix of $\ddot{\mathbf{F}}$ in Fourier domain, however, from another point of view, we can also get,

$$\ddot{\mathbf{F}}^H X_k^l \ddot{\mathbf{F}} = \mathcal{D}(\hat{\mathbf{x}}_k^l) \tag{12}$$

The first row of X_k^l is equal to $\mathcal{F}^{-1}(\hat{\mathbf{x}}_k^l)$, all other rows are the circular shifts of $\mathcal{F}^{-1}(\hat{\mathbf{x}}_k^l)$. So if we have a diagonal matrix, then we can transform it to a circulant matrix by $\ddot{\mathbf{F}}$. In Eq. 11, $\mathcal{D}(\mathbf{w})$ is a real-valued diagonal matrix, so we have

$$\mathbf{F}^H \mathcal{D}(\mathbf{w})^H \mathcal{D}(\mathbf{w}) \mathbf{F} = \mathbf{F} \mathcal{D}(\mathbf{w})^H \mathcal{D}(\mathbf{w}) \mathbf{F}^H = \mathbf{R}^H \mathbf{R} \tag{13}$$

where \mathbf{R} is circulant matrix constructed from $\mathcal{F}(\mathbf{w})$, where \mathcal{F} denotes DFT. For a real-valued function, unitary DFT and IDFT have the same results, here, we treat $\mathcal{D}(\mathbf{w})$ as a spatial domain signal, so we use DFT instead of IDFT. If we

choose a real-valued even regularization function w , therefore, $\mathcal{F}^{-1}(\mathbf{w})$ is a real-valued vector, then we will get a real-valued regularization matrix. Applying Eqs. 13 to 11, we have,

$$\sum_{k=1}^t \sum_{l=1}^d \mathbf{F} \left(\mathcal{D}(\hat{\mathbf{x}}_k^* \odot \hat{\mathbf{x}}_k^l) + \mathbf{R}^H \mathbf{R} \right) \ddot{\mathbf{F}}^H \mathbf{f}^l = \sum_{k=1}^t \sum_{l=1}^d \mathbf{F} \mathcal{D}(\hat{\mathbf{x}}_k^*) \ddot{\mathbf{F}}^H \mathbf{y}_k^l \quad (14)$$

Here, we call \mathbf{R} the potential circulant structure of regularization function w in the Fourier domain. By using the unitary property, we can further get,

$$\sum_{k=1}^t \sum_{l=1}^d \left(\mathcal{D}(\hat{\mathbf{x}}_k^l \odot \hat{\mathbf{x}}_k^l) + \mathbf{R}^H \mathbf{R} \right) (\ddot{\mathbf{F}}\ddot{\mathbf{F}})^H \hat{\mathbf{f}}^l = \sum_{k=1}^t \sum_{l=1}^d \mathcal{D}(\hat{\mathbf{x}}_k^l) (\ddot{\mathbf{F}}\ddot{\mathbf{F}})^H \hat{\mathbf{y}}_k^l \quad (15)$$

where $\ddot{\mathbf{F}}\ddot{\mathbf{F}} = (\mathbf{F} \otimes \mathbf{F})(\mathbf{F} \otimes \mathbf{F}) = (\mathbf{F}\mathbf{F}) \otimes (\mathbf{F}\mathbf{F})$ is a permutation matrix. To simplify Eq. 15, we define $\hat{\mathbf{f}} = ((\hat{\mathbf{f}}^1)^T, \dots, (\hat{\mathbf{f}}^d)^T)^T$, $\hat{\mathbf{x}}_k = ((\hat{\mathbf{x}}_k^1)^T, \dots, (\hat{\mathbf{x}}_k^d)^T)^T$, then the Eq. 15 can be equivalently expressed as,

$$\sum_{k=1}^t \left(\mathcal{D}(\hat{\mathbf{x}}_k^* \odot \hat{\mathbf{x}}_k) + \mathbf{B}^H \mathbf{B} \right) \mathcal{P}(\hat{\mathbf{f}}) = \sum_{k=1}^t \hat{\mathbf{x}}_k^* \odot \mathcal{P}(\hat{\mathbf{y}}_k^l) \quad (16)$$

where \mathbf{B} is $dMN \times dMN$ block diagonal matrix with each diagonal block being equal to \mathbf{R} , $\mathcal{P}(\cdot)$ is a permutation function according to $\ddot{\mathbf{F}}\ddot{\mathbf{F}}$. In Sect. 4, we will find that we just need to find the solution of $\mathcal{P}(\hat{\mathbf{f}}) = \hat{\mathbf{f}}_p$ instead of $\hat{\mathbf{f}}$, so what we really used equation is,

$$\sum_{k=1}^t \left(\mathcal{D}(\hat{\mathbf{x}}_k^* \odot \hat{\mathbf{x}}_k) + \mathbf{B}^H \mathbf{B} \right) \hat{\mathbf{f}}_p = \sum_{k=1}^t \hat{\mathbf{x}}_k^* \odot \mathcal{P}(\hat{\mathbf{y}}) \quad (17)$$

Because we use the same regression targets for all frames, we define $\hat{\mathbf{y}} = ((\hat{\mathbf{y}}^1)^T, \dots, (\hat{\mathbf{y}}^d)^T)^T$. Equation 17 defines a $dMN \times dMN$ linear system of equations, its coefficients matrix is real-valued, so we can solve it directly. We can see that for each frame what we need to do is element-wise multiplication. The regularization part $\mathbf{B}^H \mathbf{B}$ and regression targets part $\mathcal{P}(\hat{\mathbf{y}})$ is constant for all frames.

In Eq. 17, if we choose $\mathbf{B}^H \mathbf{B} = \lambda \mathbf{I}$, which is equivalent to choose a regularization matrix w with all elements being equal to $\sqrt{\lambda}$, we get a standard *DCF*. In this paper, w is a real-valued even Gaussian shaped function, profiting from its smooth property, we get a sparse coefficients matrix for Eq. 17, which makes significant difference for optimization. The Gauss-Seidel method is used to solve Eq. 17. Figure 1 shows the visualization of the filter solved from our proposal.

4 Our Tracking Framework

In this section, we describe our tracking framework according to the Faster Spatially Regularized Discriminative Correlation Filters proposed in Sect. 3.

4.1 Training

At the first frame, to give a better initial point for Gauss-Seidel methods, we obtain a precise solution of Eq. 17 by,

$$\hat{\mathbf{f}}_{p1} = (\mathcal{D}(\hat{\mathbf{x}}_1^* \odot \hat{\mathbf{x}}_1) + \mathbf{B}^H\mathbf{B})^{-1}(\hat{\mathbf{x}}_1^* \odot \mathcal{P}(\hat{\mathbf{y}})) \tag{18}$$

In the subsequent frames, the starting point in current frame (time t) is the optimization results in last frame (time $t - 1$). For simplicity, we define A_t and \mathbf{b}_t as,

$$A_t = \sum_{k=1}^t \left(\mathcal{D}(\hat{\mathbf{x}}_k^* \odot \hat{\mathbf{x}}_k) + \mathbf{B}^H\mathbf{B} \right) \tag{19}$$

$$\mathbf{b}_t = \sum_{k=1}^t \hat{\mathbf{x}}_k^* \odot \mathcal{P}(\hat{\mathbf{y}}) \tag{20}$$

Equation 17 can be rewrite as $A_t \hat{\mathbf{f}}_p = \mathbf{b}_t$, we split A_t into data part D_t and regularization part $\mathbf{B}^H\mathbf{B}$, split \mathbf{b}_t into \mathbf{d}_t and $\mathcal{P}(\hat{\mathbf{y}})$, then we update our model by,

$$A_t = (1 - \gamma)D_{t-1} + \gamma\mathcal{D}(\hat{\mathbf{x}}_t^* \odot \hat{\mathbf{x}}_t) + \mathbf{B}^H\mathbf{B} \tag{21}$$

$$\mathbf{b}_t = ((1 - \gamma)\mathbf{d}_{t-1} + \gamma\hat{\mathbf{x}}_t^*) \odot \mathcal{P}(\hat{\mathbf{y}}) \tag{22}$$

where $D_1 = \mathcal{D}(\hat{\mathbf{x}}_1^* \odot \hat{\mathbf{x}}_1)$, $\mathbf{d}_1 = \hat{\mathbf{x}}_1^*$. In Eqs. 21 and 22, $\mathbf{B}^H\mathbf{B}$, $\mathcal{P}(\hat{\mathbf{y}})$ are constant during model updating. We just need to precompute once for a sequence. To get new correlation filters $\hat{\mathbf{f}}_p$, a fixed N_{GS} numbers of Gauss-Seidel iterations are conducted after model updating Eqs. 21 and 22.

4.2 Detection

At the detection stage, according Eq. 8, the location of the target is estimated by finding the peak correlation between correlation filters $\hat{\mathbf{f}}_p$ and new feature maps,

$$\max \sum_{l=1}^d X_t^l \mathbf{f}^l = \max \sum_{l=1}^d x_t^l \star f^l \tag{23}$$

where \star denotes the correlation operator. For the sake of computational efficiency, we use convolution instead of correlation,

$$\max \sum_{l=1}^d x_t^l \star f^l = \max \sum_{l=1}^d x_t^l * (-f^l) \tag{24}$$

where $-$ denotes 180° rotation operator. Both x_t and f are real-valued, so their DFT is Hermitian symmetric. Computing Eq. 24 in Fourier domain, finally we get,

$$\max \mathcal{F}^{-1} \left(\sum_{l=1}^l \hat{\mathbf{x}}_t^l \odot \hat{\mathbf{f}}_p^l \right) \tag{25}$$

So in Eq. 16, we directly solve $\mathcal{P}(\hat{\mathbf{f}}) = \hat{\mathbf{f}}_p$ instead of $\hat{\mathbf{f}}$. At detection stage, we also use the scaling pool technique in paper [16] and Fast Sub-grid Detection in work [6].

5 Experiments

On benchmark datasets: OTB2013, OTB2015. To make a fair comparison with the baseline tracker SRDCF [6], we use most of the parameters used in SRDCF throughout all our experiments. Because we need to get real-valued DFT coefficients of the regularization function w and use Matlab as our implementation tool, so we reconstructed w from function $w(m, n) = \mu + \eta(m/\sigma_1)^2 + \eta(n/\sigma_2)^2$, where $[\sigma_1, \sigma_2] = \beta[P, Q]$, $P \times Q$ is the size of target. $m = [-(M/2) : (M-2)/2]$, $n = [-(N/2) : (N-2)/2]$. In our experiments, β is set to 0.8. Our source code is fully available at the github website¹.

5.1 Baseline Comparison

We do comprehensive comparison between our approach and the baseline tracker SRDCF [6]. Accuracy, robustness and speed are taken into consideration. In this section, all experiments are performed on a standard desktop computer with Intel Core i5-6400 processor. For the baseline tracker, we use the Matlab implementation provided by authors.

Our comparison follows the protocol proposed in [18]. One-Pass Evaluation (OPE), Temporal Robustness Evaluation (TRE) and Spatial Robustness Evaluation (SRE) are performed. TRE runs trackers on 20 different length subsequences segmented from the original sequences, SRE run trackers with 12 different initializations constructed from shifted or scaled ground truth bounding box. After running the trackers, we report the overall results using the area under the curve (AUC) based on success plot and mean overlap precision (OP). Besides, attribute-based evaluation results are also reported. The OP is calculated as the percentage of frames where the intersection-over-union overlap with the ground truth exceeds a threshold of 0.5. Attributes are including scale variation (SV), occlusion (OCC), deformation (DEF), fast motion (FM), in-plane-rotation (IPR), out-plane-rotation (OPR), background cluster (BC) and low resolution (LR), illumination variation (IV), out of view (OV), motion blur (MB).

In Table 1, our tracker runs at 11.4 fps, 11.1 fps on OTB-2013 and OTB-2015 datasets respectively, which are twice faster than the SRDCF on both datasets. At the same time, our approach start up much faster which is very important when tracker needs to be initialized frequently. Table 2 shows OPE results on OTB-2013 and OTB-2015, for clarity, we reported 9 attribute-based evaluation results. For overall performance, our approach outperforms the baseline tracker by 1.1%, 2.7% in AUC and OP respectively on OTB-2013 dataset and achieves

¹ <https://github.com/KnockKnock13/FSRDCF.git>.

Table 1. The comparison of speed and start-up time on OTB-2013 and OTB-2015. Our approach runs more than twice fast as the baseline tracker SRDCF and a faster startup time.

	OTB-2013		OTB-2015	
	Speed (fps)	Start-up (s)	Speed (fps)	Start-up (s)
SRDCF	5.7	1.27	5.3	1.36
FSRDCF	11.4	0.51	11.1	0.50

Table 2. Robustness evaluation comparison on OTB-2013 and OTB-2015 datasets. Both trackers achieve equivalent results in TRE on OTB-2015 dataset.

		OTB-2013									Overall	
		SV	OCC	DEF	OV	IPR	OPR	BC	LR	IV	AUC	OP
OTB13	SRDCF	59.5	62.7	63.5	55.5	57.3	60.4	58.7	42.6	57.6	63.0	78.9
	FSRDCF	61.8	63.6	65.4	54.9	59.6	63.0	61.9	42.1	58.6	64.4	81.6
OTB15	SRDCF	56.9	55.7	54.7	46.1	54.6	55.1	58.4	48.1	60.9	59.9	73.1
	FSRDCF	56.4	56.1	54.3	45.6	56.4	56.3	59.0	46.8	60.8	59.5	73.4

equivalent performance on OTB-2015 dataset. For attribute-based evaluation, our method wins in most attribute sub-datasets on OTB- 2013 dataset. Both trackers have no significant difference on OTB-2015 dataset. Table 3 shows the robustness evaluation results. Except for our tracker is a little bit more sensitive to the background due to bigger derivation parameters, we think two trackers have the same robustness performance.

5.2 OTB-2013 and OTB-2015 Datasets

Finally, we perform a comprehensive comparison with 9 recent state-of-art trackers: DLSSVM [16], SCT4 [2], MEEM [20], KCF [10], DSST [5], SAMF [11], LCT [14], Struck [9] and the baseline tracker SRDCF [6].

State-of-the-Art Comparison. We show the results of comparison with state-of-the-art trackers on OTB-2013 and OTB-2015 datasets over 100 videos in Table 4, only the results for the top 8 trackers are reported in consideration of space. The results are presented in OP and ranking according to performance on OTB-2015 dataset. In Table 4, we also give the running speed of trackers. The best results on both datasets are obtained by our tracker with mean OP of 81.6%, and 73.4%, outperforming the best non-spatial regularization trackers by 8.4% and 6.3% respectively. From the perspective of running speed, our approach runs at 11.1 frames per second, which is more than twice faster than the tracker ranking the second. Our tracker gets a better balance between accuracy and efficiency.

Table 3. Comparison with baseline tracker on OTB-2013 and OTB-2015. The results in the table are based on success plot and all reported in percent.

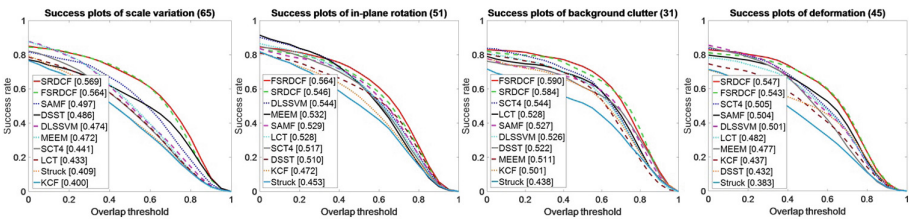
	OTB-2013		OTB-2015	
	SRE	TRE	SRE	TRE
SRDCF	0.569	0.647	0.542	0.613
FSRDCF	0.557	0.650	0.522	0.607

Table 4. State-of-the-art trackers comparison on OTB-2013 and OTB-2015 datasets in OP (in percent). The best two results are shown in red and blue respectively. Our approach achieves the best results on both datasets and have a balanced perform on accuracy and speed.

	DSST	SCT4	DLSSVM	MEEM	LCT	SAMF	SRDCF	FSRDCF
OTB-2013	66.7	73.9	72.5	70.6	73.8	73.2	78.9	81.6
OTB-2015	61.3	62.0	62.4	62.7	62.9	67.1	73.1	73.4
Speed (fps)	29.4	32.2	9.5	8.2	20.3	16.8	5.3	11.1

Figure 3 shows the success plots on OTB-2013 and OTB-2015 datasets. The trackers are ranked according to the area under the curve (AUC) and displayed in the legend. Our tracker ranks the first on OTB-2013 with a AUC of 64.4%, outperforming the best non-spatial regularization tracker by 5.1%, and ranks the second with a AUC of 59.5%, outperforming the best non-spatial-regularization tracker by 4.3% on OTB-2015.

Robustness Comparison. Like in Sect. 5.1, we perform SRE and TRE to compare the robustness of our tracker to the state-of-the-art trackers. Figure 2 shows success plots for SRE and TRE on OTB-2015 dataset with 100 videos. Our approach outperforms the best non-spatial regularization tracker 1% and 2% in SRE and TRE respectively.

**Fig. 2.** Attribute-based evaluations of our approach on OTB-2015 dataset. Number in bracket of each plot title is the videos in corresponding sub-dataset. Our tracker demonstrates superior performance compared to other non-spatial regularization trackers.

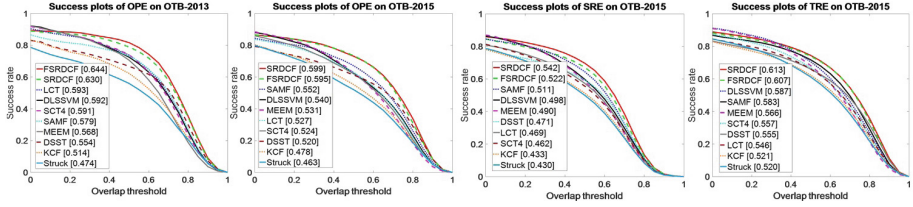


Fig. 3. Success plots showing a comparison with state-of-the-art trackers on OTB-2013 and OTB-2015 datasets. Our FSRDCF ranks the first on OTB-2013 and the second on OTB-2015. Robustness to initialization comparison on the OTB-2015 dataset. Success plots for both SRE and TRE are shown, our tracker achieves state-of-the-art performance.

Attribute Based Comparison. We performed attribute-based evaluations of our approach on OTB-2015 and compare to other state-of-the-art trackers. Our approach wins on 10 attribute sub-datasets compared to other non-spatial-regularization trackers, Fig. 2 shows the success plots of 4 different attributes on OTB-2015 dataset. Due to the using of spatial regularization, the spatially regularized trackers can learn more discriminative filters and detect targets from a larger area than standard DCF, so our tracker have big advantages in situations such as occlusion, background cluster and fast motion over other non-spatial-regularization trackers.

6 Conclusion

We introduce a new formulation of Spatially Regularized Discriminative Correlation Filters (FSRDCF) to efficiently learn a spatially regularized correlation filter. The use of circulant structure of data matrix in the spatial domain and circulant structure of regularization function in the Fourier domain significantly simplify the problem construction and solving. In our approach, both problem construction and solving are in the spatial domain. Our approach validated on the OTB-2013 and OTB-2015 datasets, and obtains a more than twice faster running speed and much faster start-up time than the baseline tracker SRDCF without any performance degradation. At the same time, our approach demonstrates superior performance compared to other non-spatial-regularization trackers.

Acknowledgments. This work was supported in part by the ShenZhen Research Fund for the development of strategic emerging industries (Nos. JCYJ20160301151844537 and CXZZ20150504163316885). In addition, we thank the anonymous reviewers for their careful read and valuable comments.

References

1. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: IEEE CVPR, pp. 2544–2550 (2010)

2. Choi, J., Chang, H.J., Jeong, J., Demiris, Y., Choi, J.Y.: Visual tracking using attention-modulated disintegration and integration. In: IEEE CVPR, pp. 4321–4330 (2016)
3. Cui, Z., Xiao, S., Feng, J., Yan, S.: Recurrently target-attending tracking. In: IEEE CVPR, pp. 1449–1458 (2016)
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE CVPR, pp. 886–893 (2005)
5. Danelljan, M., Häger, G., Khan, F.S., Felsberg, M.: Accurate scale estimation for robust visual tracking. In: IEEE BMVC (2014)
6. Danelljan, M., Häger, G., Khan, F.S., Felsberg, M.: Learning spatially regularized correlation filters for visual tracking. In: IEEE ICCV, pp. 4310–4318 (2015)
7. Danelljan, M., Häger, G., Khan, F.S., Felsberg, M.: Adaptive decontamination of the training set: a unified formulation for discriminative visual tracking. In: IEEE CVPR, pp. 1430–1438 (2016)
8. Galoogahi, H.K., Sim, T., Lucey, S.: Correlation filters with limited boundaries. In: IEEE CVPR, pp. 4630–4638 (2015)
9. Hare, S., Saffari, A., Torr, P.H.S.: Struck: structured output tracking with kernels. In: IEEE ICCV, pp. 263–270 (2011)
10. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. *IEEE Trans. PAMI* **37**(3), 583–596 (2015)
11. Li, Y., Zhu, J.: A scale adaptive kernel correlation filter tracker with feature integration. In: IEEE ECCV, pp. 254–265 (2014)
12. Liu, T., Wang, G., Yang, Q.: Real-time part-based visual tracking via adaptive correlation filters. In: IEEE CVPR, pp. 4902–4912 (2015)
13. Ma, C., Huang, J., Yang, X., Yang, M.: Hierarchical convolutional features for visual tracking. In: IEEE ICCV, pp. 3074–3082 (2015)
14. Ma, C., Yang, X., Zhang, C., Yang, M.: Long-term correlation tracking. In: IEEE CVPR, pp. 5388–5396 (2015)
15. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. In: IEEE CVPR, pp. 4293–4302 (2016)
16. Ning, J., Yang, J., Jiang, S., Zhang, L., Yang, M.: Object tracking via dual linear structured SVM and explicit feature map. In: IEEE CVPR, pp. 4266–4274 (2016)
17. van de Weijer, J., Schmid, C., Verbeek, J.J., Larlus, D.: Learning color names for real-world applications. *IEEE TIP* **18**(7), 1512–1523 (2009). <https://doi.org/10.1109/TIP.2009.2019809>
18. Wu, Y., Lim, J., Yang, M.: Online object tracking: a benchmark. In: IEEE CVPR, pp. 2411–2418 (2013)
19. Wu, Y., Lim, J., Yang, M.: Object tracking benchmark. *IEEE Trans. PAMI* **37**(9), 1834–1848 (2015). <https://doi.org/10.1109/TPAMI.2014.2388226>
20. Zhang, J., Ma, S., Sclaroff, S.: MEEM: robust tracking via multiple experts using entropy minimization. In: IEEE ECCV, pp. 188–203 (2014)