Learning Spatially Regularized Correlation Filters for Visual Tracking

Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, Michael Felsberg
Computer Vision Laboratory, Linköping University, Sweden
{martin.danelljan, gustav.hager, fahad.khan, michael.felsberg}@liu.se

Abstract

Robust and accurate visual tracking is one of the most challenging computer vision problems. Due to the inherent lack of training data, a robust approach for constructing a target appearance model is crucial. Recently, discriminatively learned correlation filters (DCF) have been successfully applied to address this problem for tracking. These methods utilize a periodic assumption of the training samples to efficiently learn a classifier on all patches in the target neighborhood. However, the periodic assumption also introduces unwanted boundary effects, which severely degrade the quality of the tracking model.

We propose Spatially Regularized Discriminative Correlation Filters (SRDCF) for tracking. A spatial regularization component is introduced in the learning to penalize correlation filter coefficients depending on their spatial location. Our SRDCF formulation allows the correlation filters to be learned on a significantly larger set of negative training samples, without corrupting the positive samples. We further propose an optimization strategy, based on the iterative Gauss-Seidel method, for efficient online learning of our SRDCF. Experiments are performed on four benchmark datasets: OTB-2013, ALOV++, OTB-2015, and VOT2014. Our approach achieves state-of-the-art results on all four datasets. On OTB-2013 and OTB-2015, we obtain an absolute gain of 8.0% *and* 8.2% *respectively, in mean overlap precision, compared to the best existing trackers.*

1. Introduction

Visual tracking is a classical computer vision problem with many applications. In generic tracking the task is to estimate the trajectory of a target in an image sequence, given only its initial location. This problem is especially challenging. The tracker must generalize the target appearance from a very limited set of training samples to achieve robustness against, *e.g.* occlusions, fast motion and deformations. Here, we investigate the key problem of learning a robust appearance model under these conditions.

Recently, Discriminative Correlation Filter (DCF) based approaches [5, 8, 10, 19, 20, 24] have successfully been applied to the tracking problem [23]. These methods learn a correlation filter from a set of training samples. The correlation filter is trained to perform a circular sliding window operation on the training samples. This corresponds to assuming a periodic extension of these samples (see figure 1). The periodic assumption enables efficient training and detection by utilizing the Fast Fourier Transform (FFT).

(a) Original image. (b) Periodicity in correlation filters.

Figure 1. Example image (a) and the underlying periodic assumption (b) employed in the standard DCF methods. The periodic assumption (b) leads to a limited set of negative training samples, that fails to capture the true image content (a). As a consequence, an inaccurate tracking model is learned.

As discussed above, the computational efficiency of the standard DCF originates from the periodic assumption at both training and detection. However, this underlying assumption produces unwanted boundary effects. This leads to an inaccurate representation of the image content, since the training patches contain periodic repetitions. The induced boundary effects mainly limit the standard DCF formulation in two important aspects. Firstly, inaccurate negative training patches reduce the discriminative power of the learned model. Secondly, the detection scores are only accurate near the center of the region, while the remaining scores are heavily influenced by the periodic repetitions of the detection sample. This leads to a very restricted target search region at the detection step.

The aforementioned limitations of the standard DCF for-

mulation hamper the tracking performance in several ways. (a) The DCF based trackers struggle in cases with fast target motion due to the restricted search region. (b) The lack of negative training patches leads to over-fitting of the learned model, significantly affecting the performance in cases with *e.g*. target deformations. (c) The mentioned limitations in training and detection also reduce the potential of the tracker to re-detect the target after an occlusion. (d) A naive expansion of the image area used for training the correlation filter corresponds to using a larger periodicity (see figure 1). Such an expansion results in an inclusion of substantial amount of background information within the positive training samples. These corrupted training samples severely degrade the discriminative power of the model, leading to inferior tracking results. In this work, we tackle these inherent problems by re-visiting the standard DCF formulation.

1.1. Contributions

In this paper, we propose Spatially Regularized Discriminative Correlation Filters (SRDCF) for tracking. We introduce a spatial regularization component within the DCF formulation, to address the problems induced by the periodic assumption. The proposed regularization weights penalize the correlation filter coefficients during learning. The spatial weights are based on the a priori information about the spatial extent of the filter. Due to the spatial regularization, the correlation filter can be learned on larger image regions. This enables a larger set of negative patches to be included in the training, leading to a more discriminative model.

Due to the online nature of the tracking problem, a computationally efficient learning scheme is crucial. Therefore, we introduce a suitable optimization strategy for the proposed SRDCF. The online capability is achieved by exploiting the sparsity of the spatial regularization function in the Fourier domain. We propose to apply the iterative Gauss-Seidel method to solve the resulting normal equations. Additionally, we introduce a strategy to maximize the detection scores with sub-grid precision.

We perform comprehensive experiments on four benchmark datasets: OTB-2013 [33] with 50 videos, ALOV++ [30] with 314 videos, VOT2014 [23] with 25 videos and OTB-2015 [34] with 100 videos. Compared to the best existing trackers, our approach obtains an absolute gain of 8.0% and 8.2% on OTB-2013 and OTB-2015 respectively, in mean overlap precision. Our method also achieves the best overall results on ALOV++ and VOT2014. Additionally, our tracker won the OpenCV State of the Art Vision Challenge in tracking [25] (there termed DCFSIR).

2. Discriminative Correlation Filters

Discriminative correlation filters (DCF) is a supervised technique for learning a linear classifier or a linear regressor. The main difference from other techniques, such as support vector machines [6], is that the DCF formulation exploits the properties of circular correlation for efficient training and detection. In recent years, the DCF based approaches have been successfully applied for tracking. Bolme *et al*. [5] first introduced the MOSSE tracker, using only grayscale samples to train the filter. Recent work [9, 8, 10, 20, 24] have shown a notable improvement by learning multi-channel filters on multi-dimensional features, such as HOG [7] or Color-Names [31]. However, to become computationally viable, these approaches rely on harsh approximations of the standard DCF formulation, leading to sub-optimal learning. Other work have investigated offline learning of multi-channel DCF:s for object detection [13, 18] and recognition [4]. But these methods are too computationally costly for online tracking applications.

The circular correlation within the DCF formulation has two major advantages. Firstly, the DCF is able to make extensive use of limited training data by implicitly including all shifted versions of the given samples. Secondly, the computational effort for training and detection is significantly reduced by performing the necessary computations in the Fourier domain and using the Fast Fourier Transform (FFT). These two advantages make DCF:s especially suitable for tracking, where training data is scarce and computational efficiency is crucial for real-time applications.

By employing a circular correlation, the standard DCF formulation relies on a periodic assumption of the training and detection samples. However, this assumption produces unwanted boundary effects, leading to an inaccurate description of the image. These inaccurate training patches severely hamper the learning of a discriminative tracking model. Surprisingly, this problem has been largely ignored by the tracking community. Galoogahi *et al*. [14] investigate the boundary effect problem for single-channel DCF:s. Their approach solve a constrained optimization problem, using the Alternating Direction Method of Multipliers (ADMM), to ensure a correct filter size. This however requires a transition between the spatial and Fourier domain in each ADMM iteration, leading to an increased computational complexity. Different to [14], we propose a spatial regularization component in the objective. By exploiting the sparsity of our regularizer, we efficiently optimize the filter directly in the Fourier domain. Contrary to [14], we target the problem of multi-dimensional features, such as HOG, crucial for the overall tracking performance [10, 20].

2.1. Standard DCF Training and Detection

In the DCF formulation, the aim is to learn a multi-channel convolution[1] filter f from a set of training examples $\{(x_k, y_k)\}_{k=1}^{t}$. Each training sample x_k consists of a d-dimensional feature map extracted from an image re-

[1] We use convolution for mathematical convenience, though correlation can equivalently be used.

gion. All samples are assumed to have the same spatial size $M \times N$. At each spatial location $(m, n) \in \Omega := \{0, \dots, M-1\} \times \{0, \dots, N-1\}$ we thus have a d-dimensional feature vector $x_k(m, n) \in \mathbb{R}^d$. We denote feature layer $l \in \{1, \dots, d\}$ of x_k by x_k^l. The desired output y_k is a scalar valued function over the domain Ω, which includes a label for each location in the sample x_k.

The desired filter f consists of one $M \times N$ convolution filter f^l per feature layer. The convolution response of the filter f on a $M \times N$ sample x is given by

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

Here, $*$ denotes circular convolution. The filter is obtained by minimizing the L^2-error between the responses $S_f(x_k)$ on the training samples x_k, and the labels y_k,

$$\varepsilon_t(f) = \sum_{k=1}^{t} \alpha_k \left\| S_f(x_k) - y_k \right\|^2 + \lambda \sum_{l=1}^{d} \left\| f^l \right\|^2. \tag{2}$$

Here, the weights $\alpha_k \geq 0$ determine the impact of each training sample and $\lambda \geq 0$ is the weight of the regularization term. Eq. 2 is a linear least squares problem. Using Parseval's formula, it can be transformed to the Fourier domain, where the resulting normal equations have a block diagonal structure. The Discrete Fourier Transformed (DFT) filters $\hat{f}^l = \mathscr{F}\{f^l\}$ can then be obtained by solving MN number of $d \times d$ linear equation systems [13].

For efficiency reasons, the learned DCF is typically applied in a sliding-window-like manner by evaluating the classification scores on all cyclic shifts of a test sample. Let z denote the $M \times N$ feature map extracted from an image region. The classification scores $S_f(z)$ at all locations in this image region can be computed using the convolution property of the DFT,

$$S_f(z) = \mathscr{F}^{-1}\left\{ \sum_{l=1}^{d} \hat{z}^l \cdot \hat{f}^l \right\}. \tag{3}$$

Here, $\cdot$ denotes point-wise multiplication, the hat denotes the DFT of a function and $\mathscr{F}^{-1}$ denotes the inverse DFT. The FFT hence allows the detection scores to be computed in $\mathcal{O}(dMN \log MN)$ complexity instead of $\mathcal{O}(dM^2N^2)$.

Note that the operation $S_f(x)$ in (1) corresponds to applying the linear classifier f, in a sliding window fashion, to the periodic extension of the sample x (see figure 1). This introduces unwanted periodic boundary effects in the training (2) and detection (3) steps.

3. Spatially Regularized Correlation Filters

We propose to use a spatial regularization component in the standard DCF formulation. The resulting optimization problem is solved in the Fourier domain, by exploiting the sparse nature of the proposed regularization.

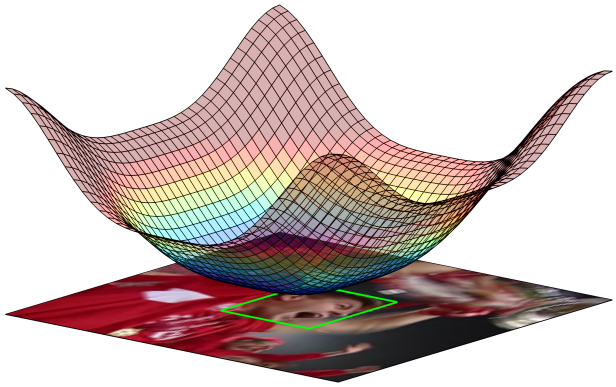


Figure 2. Visualization of the spatial regularization weights w employed in the learning of our SRDCF, and the corresponding image region used for training. Filter coefficients residing in the background region are penalized by assigning higher weights in w. This significantly mitigates the emphasis on background information in the learned classifier.

3.1. Spatial Regularization

To alleviate the problems induced by the circular convolution in (1), we replace the regularization term in (2) with a more general Tikhonov regularization. We introduce a spatial weight function $w : \Omega \to \mathbb{R}$ used to penalize the magnitude of the filter coefficients in the learning. The regularization weights w determine the importance of the filter coefficients f^l, depending on their spatial locations. Coefficients in f^l residing outside the target region are suppressed by assigning higher weights in w and vice versa. The resulting optimization problem is expressed as,

$$\varepsilon(f) = \sum_{k=1}^{t} \alpha_k \left\| S_f(x_k) - y_k \right\|^2 + \sum_{l=1}^{d} \left\| w \cdot f^l \right\|^2. \tag{4}$$

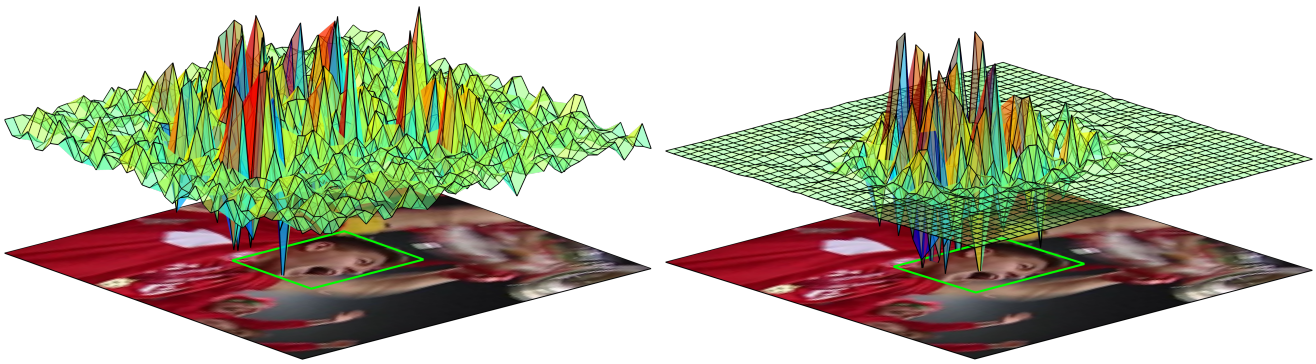
The regularization weights w in (4) are visualized in figure 2. Visual features close to the target edge are often less reliable than those close to the target center, due to *e.g.* target rotations and occlusions. We therefore let the regularization weights change smoothly from the target region to the background. This also increases the sparsity of w in the Fourier domain. Note that (4) simplifies to the standard DCF (2) for uniform weights $w(m, n) = \sqrt{\lambda}$.

By applying Parseval's theorem to (4), the filter f can equivalently be obtained by minimizing the resulting loss function (5) over the DFT coefficients $\hat{f}$,

$$\check{\varepsilon}(\hat{f}) = \sum_{k=1}^{t} \alpha_k \left\| \sum_{l=1}^{d} \hat{x}_k^l \cdot \hat{f}^l - \hat{y}_k \right\|^2 + \sum_{l=1}^{d} \left\| \frac{\hat{w}}{MN} * \hat{f}^l \right\|^2. \tag{5}$$

The second term in (5) follows from the convolution property of the inverse DFT. A vectorization of (5) gives,

$$\check{\varepsilon}(\hat{f}) = \sum_{k=1}^{t} \alpha_k \left\| \sum_{l=1}^{d} \mathcal{D}(\hat{\mathbf{x}}_k^l) \hat{\mathbf{f}}^l - \hat{\mathbf{y}}_k \right\|^2 + \sum_{l=1}^{d} \left\| \frac{\mathcal{C}(\hat{\mathbf{w}})}{MN} \hat{\mathbf{f}}^l \right\|^2. \tag{6}$$



(a) Standard DCF. (b) Our SRDCF.

Figure 3. Visualization of the filter coefficients learned using the standard DCF (a) and our approach (b). The surface plots show the filter values f^l and the corresponding image region used for training. In the standard DCF, high values are assigned to the background region. The larger influence of background information at the detection stage deteriorates tracking performance. In our approach, the regularization weights penalizes filter values corresponding to features in the background. This increases the discriminative power of the learned model, by emphasizing the appearance information within the target region (green box).

Here, bold letters denote a vectorization of the corresponding scalar valued functions and $\mathcal{D}(\mathbf{v})$ denotes the diagonal matrix with the elements of the vector $\mathbf{v}$ in its diagonal. The $MN \times MN$ matrix $\mathcal{C}(\hat{\mathbf{w}})$ represents circular 2D-convolution with the function $\hat{w}$, *i.e.* $\mathcal{C}(\hat{\mathbf{w}})\hat{\mathbf{f}}^l = \mathrm{vec}(\hat{w} * \hat{f}^l)$. Each row in $\mathcal{C}(\hat{\mathbf{w}})$ thus contains a cyclic permutation of $\hat{\mathbf{w}}$.

The DFT of a real-valued function is known to be Hermitian symmetric. Therefore, minimizing (4) over the set of real-valued filters f^l, corresponds to minimizing (5) over the set of Hermitian symmetric DFT coefficients $\hat{f}^l$. We reformulate (6) to an equivalent real-valued optimization problem, to ensure faster convergence by preserving the Hermitian symmetry. Let $\rho : \Omega \to \Omega$ be the point-reflection $\rho(m,n) = (-m \mod M, -n \mod N)$. The domain Ω can be partitioned into Ω_0, Ω_+ and Ω_-, where $\Omega_0 = \rho(\Omega_0)$ and $\Omega_- = \rho(\Omega_+)$. Thus, Ω_0 denote the part of the spectrum with no corresponding reflected frequency, and Ω_- contains the reflected frequencies in Ω_+. We define,

$$\tilde{f}^l(m,n) = \begin{cases} \hat{f}^l(m,n), & (m,n) \in \Omega_0 \\ \frac{\hat{f}^l(m,n) + \hat{f}^l(\rho(m,n))}{\sqrt{2}}, & (m,n) \in \Omega_+ \\ \frac{\hat{f}^l(m,n) - \hat{f}^l(\rho(m,n))}{i\sqrt{2}}, & (m,n) \in \Omega_- \end{cases} \quad (7)$$

such that $\tilde{f}^l$ is real-valued by the Hermitian symmetry of $\hat{f}^l$. Here, i denotes the imaginary unit. Eq. 7 can be expressed by a unitary $MN \times MN$ matrix B such that $\tilde{\mathbf{f}}^l = B\hat{\mathbf{f}}^l$. By (7), B contains at most two non-zero entries in each row.

The reformulated variables from (6) are defined as $\tilde{\mathbf{y}}_k = B\hat{\mathbf{y}}_k$, $D_k^l = B\mathcal{D}(\hat{\mathbf{x}}_k^l)B^{\mathrm{H}}$ and $C = \frac{1}{MN}B\mathcal{C}(\hat{\mathbf{w}})B^{\mathrm{H}}$, where $^{\mathrm{H}}$ denotes the conjugate transpose of a matrix. Since B is unitary, (6) can equivalently be expressed as,

$$\tilde{\varepsilon}(\tilde{\mathbf{f}}^1 \dots \tilde{\mathbf{f}}^d) = \sum_{k=1}^{t} \alpha_k \left\| \sum_{l=1}^{d} D_k^l \tilde{\mathbf{f}}^l - \tilde{\mathbf{y}}_k \right\|^2 + \sum_{l=1}^{d} \left\| C\tilde{\mathbf{f}}^l \right\|^2. \quad (8)$$

All variables in (8) are real-valued. The loss function (8) is then simplified by defining the fully vectorized real-valued filter as the concatenation $\tilde{\mathbf{f}} = \left((\tilde{\mathbf{f}}^1)^{\mathrm{T}} \cdots (\tilde{\mathbf{f}}^d)^{\mathrm{T}}\right)^{\mathrm{T}}$,

$$\tilde{\varepsilon}(\tilde{\mathbf{f}}) = \sum_{k=1}^{t} \alpha_k \left\| D_k\tilde{\mathbf{f}} - \tilde{\mathbf{y}}_k \right\|^2 + \left\| W\tilde{\mathbf{f}} \right\|^2. \quad (9)$$

Here we have defined the concatenation $D_k = (D_k^1 \cdots D_k^d)$ and W to be the $dMN \times dMN$ block diagonal matrix with each diagonal block being equal to C. Finally, (9) is minimized by solving the normal equations $A_t\tilde{\mathbf{f}} = \tilde{\mathbf{b}}_t$, where

$$A_t = \sum_{k=1}^{t} \alpha_k D_k^{\mathrm{T}} D_k + W^{\mathrm{T}}W \quad (10a)$$

$$\tilde{\mathbf{b}}_t = \sum_{k=1}^{t} \alpha_k D_k^{\mathrm{T}} \tilde{\mathbf{y}}_k. \quad (10b)$$

Here, (10) defines a real $dMN \times dMN$ linear system of equations. The fraction of non-zero elements in A_t is smaller than $\frac{2d+K^2}{dMN}$, where K is the number of non-zero Fourier coefficients in $\hat{w}$. Thus, A_t is sparse if w has a sparse spectrum. The DFT coefficients for the filters are obtained by solving the system (10) and applying $\hat{\mathbf{f}}^l = B^{\mathrm{H}}\tilde{\mathbf{f}}^l$.

Figure 3 visualizes the filter learned by optimizing the standard DCF loss (2) and the proposed formulation (4), using the spatial regularization weights w in figure 2. In the standard DCF, large values are spatially distributed over the whole filter. By penalizing filter coefficients corresponding to background, our approach learns a classifier that emphasizes visual information within the target region.

A direct application of a sparse solver to the normal equations $A_t\tilde{\mathbf{f}} = \tilde{\mathbf{b}}_t$ is computationally very demanding (even when the standard regularization $W^{\mathrm{T}}W = \lambda I$ is used and the number of features is small ($d > 2$)). Next, we propose an efficient optimization scheme to solve the normal equations for online learning scenarios, such as tracking.

3.2. Optimization

For the standard DCF formulation (2) the normal equations have a block diagonal structure [13]. However, this block structure is not attainable in our case due to the structure of the regularization matrix $W^{\mathrm{T}}W$ in (10a). We propose an iterative approach, based on the Gauss-Seidel, for efficient online computation of the filter coefficients.

The Gauss-Seidel method decomposes the matrix A_t into a lower triangular part L_t and a strictly upper triangular part U_t such that $A_t = L_t + U_t$. The algorithm then proceeds by solving the following triangular system for $\tilde{\mathbf{f}}^{(j)}$ in each iteration $j = 1, 2, \ldots$,

$$L_t \tilde{\mathbf{f}}^{(j)} = \tilde{\mathbf{b}}_t - U_t \tilde{\mathbf{f}}^{(j-1)}. \tag{11}$$

This lower triangular equation system is solved efficiently using forward substitution and by exploiting the sparsity of L_t and U_t. The Gauss-Seidel recursion (11) converges to the solution of $A_t\tilde{\mathbf{f}} = \tilde{\mathbf{b}}$ whenever the matrix A_t is symmetric and positive definite. The construction of the weights w (see section 5.1) ensures that both conditions are satisfied.

4. Our Tracking Framework

Here, we describe our tracking framework, based on the Spatially Regularized Discriminative Correlation Filters (SRDCF) proposed in section 3.

4.1. Training

At the training stage, the model is updated by first extracting a new training sample x_t centered at the target location. Here, t denotes the current frame number. We then update A_t and $\tilde{\mathbf{b}}_t$ in (10) with a learning rate $\gamma \geq 0$,

$$A_t = (1-\gamma)A_{t-1} + \gamma\left(D_t^{\mathrm{T}} D_t + W^{\mathrm{T}}W\right) \tag{12a}$$

$$\tilde{\mathbf{b}}_t = (1-\gamma)\tilde{\mathbf{b}}_{t-1} + \gamma D_t^{\mathrm{T}} \tilde{\mathbf{y}}_t. \tag{12b}$$

This corresponds to using exponentially decaying weights α_k in the loss function (4). In the first frame, we set $A_1 = D_1^{\mathrm{T}} D_1 + W^{\mathrm{T}}W$ and $\tilde{\mathbf{b}}_1 = D_1^{\mathrm{T}}\tilde{\mathbf{y}}_1$. Note that the regularization matrix $W^{\mathrm{T}}W$ can be precomputed once for the entire sequence. The update strategy (12) ensures memory efficiency, since it does not require storage of all samples x_k. After the model update (12), we perform a fixed number N_{GS} of Gauss-Seidel iterations (11) per frame to compute the new filter coefficients.

For the initial iteration $\tilde{\mathbf{f}}_t^{(0)}$ in frame t, we use the filter computed in the previous frame, *i.e.* $\tilde{\mathbf{f}}_t^{(0)} = \tilde{\mathbf{f}}_{t-1}^{(N_{\mathrm{GS}})}$. In the first frame, the initial estimate $\tilde{\mathbf{f}}_1^{(0)}$ is obtained by solving the $MN \times MN$ linear system,

$$\left(\sum_{p=1}^{d} (D_1^p)^{\mathrm{T}} D_1^p + dC^{\mathrm{T}}C\right)\tilde{\mathbf{f}}_1^{l,(0)} = (D_1^l)^{\mathrm{T}}\tilde{\mathbf{y}}_1 \tag{13}$$

for $l = 1, \ldots, d$. This provides a starting point for the Gauss-Seidel optimization in the first frame. The systems in (13) share the same sparse coefficients and can be solved efficiently with a direct sparse solver.

4.2. Detection

At the detection stage, the location of the target in a new frame t is estimated by applying the filter $\hat{f}_{t-1}$ that has been updated in the previous frame. Similar to [24], we apply the filter at multiple resolutions to estimate changes in the target size. The samples $\{z_r\}_{r\in\{\lfloor\frac{1-S}{2}\rfloor,\ldots,\lfloor\frac{S-1}{2}\rfloor\}}$ are extracted centered at the previous target location and at the scales a^r relative to the current target scale. Here, S denotes the number of scales and a is the scale increment factor. The sample z_r is constructed by resizing the image according to a^r before the feature computation.

Fast Sub-grid Detection: Generally, the training and detection samples x_k and z_k are constructed using a grid strategy with a stride greater than one pixel. This leads to only computing the detection scores (3) on a coarser grid. We employ an interpolation approach that allows computation of pixel-dense detection scores. The detection scores (3) are efficiently interpolated with trigonometric polynomials by utilizing the computed DFT coefficients. Let $\hat{s} := \mathscr{F}\{S_f(z)\} = \sum_{l=1}^{d} \hat{z}^l \cdot \hat{f}^l$ be the DFT of the detection scores $S_f(z)$ evaluated at the sample z. The detection scores $s(u,v)$ at the continuous locations $(u,v) \in [0,M) \times [0,N)$ in z are interpolated as,

$$s(u,v) = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} \hat{s}(m,n) e^{i2\pi\left(\frac{m}{M}u + \frac{n}{N}v\right)}. \tag{14}$$

Here, i denotes the imaginary unit. We aim to find the sub-grid location that corresponds to the maximum score: $(u^*, v^*) = \arg\max_{(u,v)\in[0,M)\times[0,N)} s(u,v)$. The scores s are first evaluated at all grid locations $s(m,n)$ using (3). The location of the maximal score $(u^{(0)}, v^{(0)}) \in \Omega$ is used as the initial estimate. We then iteratively maximize (14) using Newton's method, by starting at the location $(u^{(0)}, v^{(0)})$. The gradient and Hessian in each iteration are computed by analytically differentiating (14). We found that only a few iterations is sufficient for convergence.

We apply the sub-grid interpolation strategy to maximize the classification scores s_r computed at the sample z_r. The procedure is applied for each scale level independently. The scale level with the highest maximal detection score is then used to update target location and scale.

Excluding the feature extraction, the total computational complexity of our tracker sums up to $\mathcal{O}(dSMN \log MN + SMNN_{\mathrm{Ne}} + (d+K^2)dMNN_{\mathrm{GS}})$. Here, N_{Ne} denotes the number of iterations in the sub-grid detection. In our case, the expression is dominated by the last term, which originates from the filter optimization.

5. Experiments

Here, we present a comprehensive evaluation of the proposed method. Result are reported on four benchmark datasets: OTB-2013, OTB-2015, ALOV++ and VOT2014.

5.1. Details and Parameters

The weight function w is constructed by starting from a quadratic function $w(m,n) = \mu + \eta(m/P)^2 + \eta(n/Q)^2$ with the minimum located at the sample center. Here $P \times Q$ denotes the target size, while μ and η are parameters. The minimum value of w is set to $\mu = 0.1$ and the impact of the regularizer is set to $\eta = 3$. In practice, only a few DFT coefficients in the resulting function have a significant magnitude. We simply remove all DFT coefficients smaller than a threshold to ensure a sparse spectrum $\hat{w}$, containing about 10 non-zero coefficients. Figure 1 visualizes the resulting weight function w used in the optimization.

Similar to recent DCF based trackers [8, 20, 24], we also employ HOG features, using a cell size of 4×4 pixels. Samples are represented by a square $M \times N$ grid of cells (*i.e.* $M = N$), such that the corresponding image area is proportional to the area of the target bounding box. We set the image region area of the samples to 4^2 times the target area and set the initial scale to ensure a maximum sample size of $M = 50$ cells. Samples are multiplied by a Hann window [5]. We set the label function y_t to a sampled Gaussian with a standard deviation proportional to the target size [8, 19]. The learning rate is set to $\gamma = 0.025$ and we use $N_{\text{GS}} = 4$ Gauss-Seidel iterations. All parameters remain fixed for all videos and datasets. Our Matlab implementation[2] runs at 5 frames per second on a standard desktop computer.

5.2. Baseline Comparison

Here, we evaluate the impact of the proposed spatial regularization component and compare it with the standard DCF formulation. First, we investigate the consequence of simply replacing the proposed regularizer with the standard DCF regularization in (2), without altering any parameters. This corresponds to using uniform regularization weights $w(m,n) = \sqrt{\lambda}$, in our framework. We set $\lambda = 0.01$ following [8, 10, 19]. For a fair comparison, we also evaluate both our and the standard regularization using a smaller sample size relative to the target, by setting the size as in [8, 10, 19].

Table 1 shows the mean overlap precision (OP) for the four methods on the OTB-2013 dataset. The OP is computed as the fraction of frames in the sequence where the intersection-over-union overlap with the ground truth exceeds a threshold of 0.5 (PASCAL criterion). The standard DCF benefits from using smaller samples to avoid corrupting the positive training samples with background information. On the other hand, the proposed spatial regularization enables an expansion of the image region used for training the filter, without corrupting the target model. This leads to a more discriminative model, resulting in a gain of 7.0% in mean OP compared to the standard DCF formulation.

[2]Available at `http://www.cvl.isy.liu.se/research/objrec/visualtracking/regvistrack/index.html`.

	Conventional sample size		**Expanded sample size**	
Regularization	Standard	Ours	Standard	Ours
Mean OP (%)	71.1	72.2	50.1	**78.1**

Table 1. A comparison of tracking performance on OTB-2013 when using the standard regularization (2) and the proposed spatial regularization (4), in our tracking framework. The comparison is performed both with a conventional sample size (used in existing DCF based trackers) and our expanded sample size.

	LSHT	ASLA	Struck	ACT	TGPR	KCF	DSST	SAMF	MEEM	**SRDCF**
OTB-2013	47.0	56.4	58.8	52.6	62.6	62.3	67	69.7	*70.1*	**78.1**
OTB-2015	40.0	49.0	52.9	49.6	54	54.9	60.6	*64.7*	63.4	**72.9**

Table 2. A comparison with state-of-the-art trackers on the OTB-2013 and OTB-2015 datasets using mean overlap precision (in percent). The best two results for each dataset are shown in red and blue fonts respectively. Our SRDCF achieves a gain of 8.0% and 8.2% on OTB-2013 and OTB-2015 respectively compared to the second best tracker on each dataset.

Additionally, we compare our method with Correlation Filters with Limited Boundaries (CFLB) [14]. For a fair comparison, we use the same settings as in [14] for our approach: single grayscale channel, no scale estimation, no sub-grid detection and the same sample size. On the OTB-2013, the CFLB achieves a mean OP of 48.6%. Whereas the mentioned baseline version of our tracker obtains a mean OP of 54.3%, outperforming [14] by 5.7%.

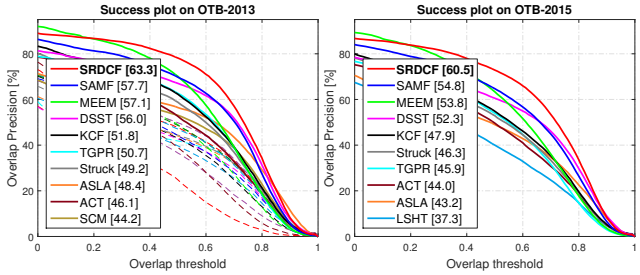
5.3. OTB-2013 Dataset

We provide a comparison of our tracker with 24 state-of-the-art methods from the literature: MIL [2], IVT [28], CT [36], TLD [22], DFT [29], EDFT [12], ASLA [21], L1APG [3], CSK [19], SCM [37], LOT [26], CPF [27], CXT [11], Frag [1], Struck [16], LSHT [17], LSST [32], ACT [10], KCF [20], CFLB [14], DSST [8], SAMF [24], TGPR [15] and MEEM [35].

5.3.1 State-of-the-art Comparison

Table 2 shows a comparison with state-of-the-art methods on the OTB-2013 dataset, using mean overlap precision (OP) over all 50 videos. Only the results for the top 10 trackers are reported. The MEEM tracker, based on an online SVM, provides the second best results with a mean OP of 70.1%. The best result on this dataset is obtained by our tracker with a mean OP of 78.1%, leading to a significant gain of 8.0% compared to MEEM.

Figure 4a shows the success plot over all the 50 videos in OTB-2013. The success plot shows the mean overlap preci-



(a) OTB-2013 (b) OTB-2015

Figure 4. Success plots showing a comparison with state-of-the-art methods on OTB-2013 (a) and OTB-2015 (b). For clarity, only the top 10 trackers are displayed. Our SRDCF achieves a gain of 5.6% and 5.7% on OTB-2013 and OTB-2015 respectively, compared to the second best methods.

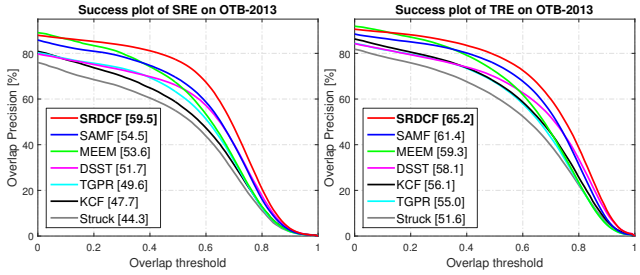


Figure 5. Comparison with respect to robustness to initialization on OTB-2013. We show success plots for both the spatial (SRE) and temporal (TRE) robustness. Our approach clearly demonstrates robustness in both scenarios.

sion (OP), plotted over the range of intersection-over-union thresholds. The trackers are ranked using the *area under the curve* (AUC), displayed in the legend. Among previous DCF based trackers, DSST and SAMF provides the best performance, with an AUC score of 56.0% and 57.7%. Our approach obtains an AUC score of 63.3% and significantly outperforms the best existing tracker (SAMF) by 5.6%.

5.3.2 Robustness to Initialization

Visual tracking methods are known to be sensitive to initialization. We evaluate the robustness of our tracker by following the protocol proposed in [33]. Two different types of initialization criteria, namely: temporal robustness (TRE) and spatial robustness (SRE), are evaluated. The SRE corresponds to tracker initialization at different positions close to the ground-truth in the first frame. The procedure is repeated with 12 different initializations for each video in the dataset. The TRE criteria evaluates the tracker by initializations at 20 different frames, with the ground-truth.

Figure 5 shows the success plots for TRE and SRE on the OTB-2013 dataset with 50 videos. We include all the top 7 trackers in figure 4a for this experiment. Among the existing methods, SAMF and MEEM provide the best results. Our SRDCF achieves a consistent gain in performance over these trackers on both robustness evaluations.

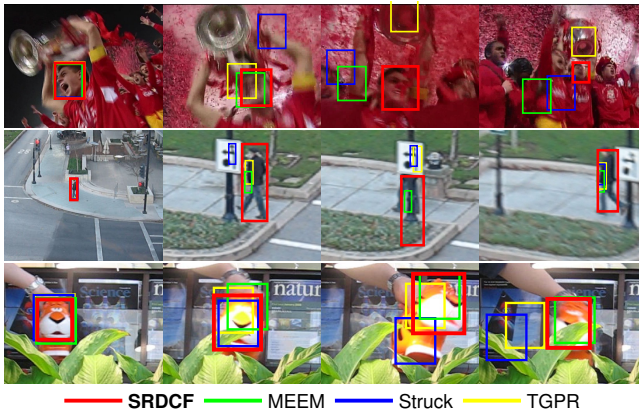


Figure 6. Qualitative comparison of our approach with state-of-the-art trackers on the *Soccer*, *Human6* and *Tiger2* videos. Our approach provides consistent results in challenging scenarios, such as occlusions, fast motion, background clutter and target rotations.

5.3.3 Attribute Based Comparison

We perform an attribute based analysis of our approach on the OTB-2013 dataset. All the 50 videos in OTB-2013 are annotated with 11 different attributes, namely: illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out-of-view, background clutter and low resolution. Our approach outperforms existing trackers on 10 attributes.

Figure 7 shows example success plots of four different attributes. Only the top 10 trackers in each plot are displayed for clarity. In case of out-of-plane rotations, (MEEM) achieves an AUC score of 57.2%. Our tracker provides a gain of 3.3% compared to MEEM. Among the existing methods, the two DCF based trackers DSST and SAMF provide the best results in case of scale variation. Both these trackers are designed to handle scale variations. Our approach achieves a significant gain of 4.1% over DSST. Note that the standard DCF trackers struggle in the cases of motion blur and fast motion due to the restricted search area. This is caused by the induced boundary effects in the detection samples of the standard DCF trackers. Our approach significantly improves the performance compared to the standard DCF based trackers in these cases. Figure 6 shows a qualitative comparison of our approach with existing methods on challenging example videos. Despite no explicit occlusion handling component, our tracker performs favorably in cases with occlusion.

5.4. OTB-2015 Dataset

We provide a comparison of our approach on the recently introduced OTB-2015. The dataset extends OTB-2013 and contains 100 videos. Table 2 shows the comparison with the top 10 methods, using mean overlap precision (OP) over all 100 videos. Among the existing methods, SAMF and MEEM provide the best results with mean OP of 64.7%

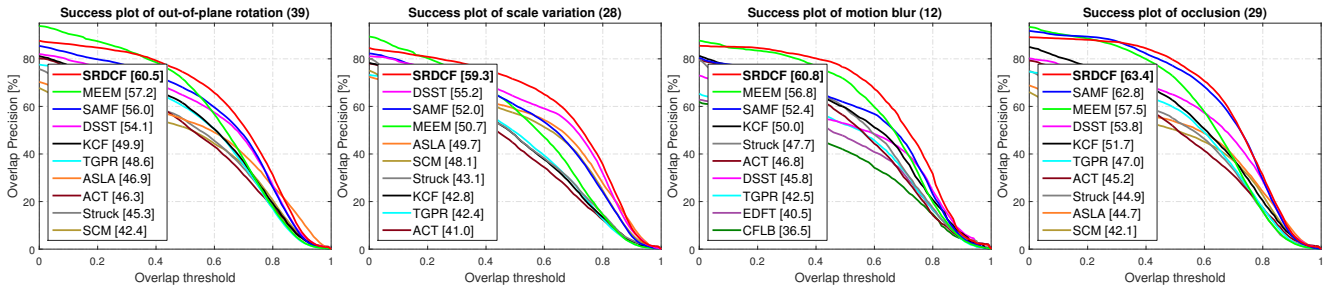


Figure 7. Attribute-based analysis of our approach on the OTB-2013 dataset with 50 videos. Success plots are shown for four attributes. Each plot title includes the number of videos associated with the respective attribute. Only the top 10 trackers for each attribute are displayed for clarity. Our approach demonstrates superior performance compared to existing trackers in these scenarios.

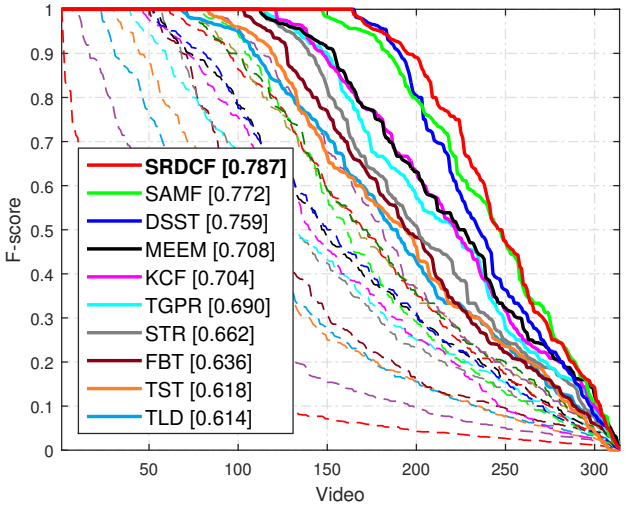


Figure 8. Survival curves comparing our approach with 24 trackers on ALOV++. The mean F-scores for the top 10 trackers are shown in the legend. Our approach achieves the best overall results.

and 63.4% respectively. Our tracker outperforms the best existing tracker by 8.2% in mean OP.

Figure 4b shows the success plot over all the 100 videos. Among the standard DCF trackers, SAMF provides the best results with an AUC score of 54.8%. The MEEM tracker achieves an AUC score of 53.8%. Our tracker obtains an AUC score of 60.5%, outperforming SAMF by 5.7%.

5.5. ALOV++ Dataset

We also perform experiments on the ALOV++ dataset [30], containing 314 videos with 89364 frames in total. The evaluation protocol employs survival curves based on F-score, where a higher F-score indicates better performance. The survival curve is constructed by plotting the sorted F-scores of all 314 videos. We refer to [30] for details.

Our approach is compared with the 19 trackers evaluated in [30]. We also add the top 5 methods from our OTB comparison. Figure 8 shows the survival curves and the average F-scores of the trackers. MEEM obtains a mean F-score of 0.708. Our approach obtains the best overall performance compared to 24 trackers with a mean F-score of 0.787.

	Overlap	Failures	Acc. Rank	Rob. Rank	Final Rank
SRDCF	0.63	15.90	6.43	10.08	**8.26**
DSST	0.64	16.90	5.99	11.17	*8.58*
SAMF	0.64	19.23	5.87	14.14	10.00

Table 3. Results for the top 3 trackers on VOT2014. The mean overlap and failure rate is reported in the first two columns. The accuracy rank, robustness rank and the combined final rank are shown in the remaining columns. Our tracker obtains the best performance on this dataset.

5.6. VOT2014 Dataset

Finally, we present results on VOT2014 [23]. Our approach is compared with the 38 participating trackers in the challenge. We also add MEEM in the comparison. In VOT2014, the trackers are evaluated both in terms of accuracy and robustness. The accuracy score is based on the overlap with ground truth, while the robustness is determined by the failure rate. The trackers are restarted at each failure. The final rank is based on the accuracy and robustness in each video. We refer to [23] for details.

Table 3 shows the final ranking scores over all the videos in VOT2014. Among the existing methods, the DSST approach provides the best results. Our tracker achieves the top final rank of 8.26, outperforming DSST and SAMF.

6. Conclusions

We propose Spatially Regularized Discriminative Correlation Filters (SRDCF) to address the limitations of the standard DCF. The introduced spatial regularization component enables the correlation filter to be learned on larger image regions, leading to a more discriminative appearance model. By exploiting the sparsity of the regularization operation in the Fourier domain, we derive an efficient optimization strategy for learning the filter. The proposed learning procedure employs the Gauss-Seidel method to solve for the filter in the Fourier domain. We perform comprehensive experiments on four benchmark datasets. Our SRDCF outperforms existing trackers on all four datasets.

Acknowledgments: This work has been supported by SSF (CUAS) and VR (VIDI, EMC^2, ELLIIT, and CADICS).

References

[1] A. Adam, E. Rivlin, and Shimshoni. Robust fragments-based tracking using the integral histogram. In *CVPR*, 2006.

[2] B. Babenko, M.-H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *CVPR*, 2009.

[3] C. Bao, Y. Wu, H. Ling, and H. Ji. Real time robust l1 tracker using accelerated proximal gradient approach. In *CVPR*, 2012.

[4] V. N. Boddeti, T. Kanade, and B. V. K. V. Kumar. Correlation filters for object alignment. In *CVPR*, 2013.

[5] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In *CVPR*, 2010.

[6] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[8] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *BMVC*, 2014.

[9] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg. Coloring channel representations for visual tracking. In *SCIA*, 2015.

[10] M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer. Adaptive color attributes for real-time visual tracking. In *CVPR*, 2014.

[11] T. B. Dinh, N. Vo, and G. Medioni. Context tracker: Exploring supporters and distracters in unconstrained environments. In *CVPR*, 2011.

[12] M. Felsberg. Enhanced distribution field tracking using channel representations. In *ICCV Workshop*, 2013.

[13] H. K. Galoogahi, T. Sim, and S. Lucey. Multi-channel correlation filters. In *ICCV*, 2013.

[14] H. K. Galoogahi, T. Sim, and S. Lucey. Correlation filters with limited boundaries. In *CVPR*, 2015.

[15] J. Gao, H. Ling, W. Hu, and J. Xing. Transfer learning based visual tracking with gaussian process regression. In *ECCV*, 2014.

[16] S. Hare, A. Saffari, and P. Torr. Struck: Structured output tracking with kernels. In *ICCV*, 2011.

[17] S. He, Q. Yang, R. Lau, J. Wang, and M.-H. Yang. Visual tracking via locality sensitive histograms. In *CVPR*, 2013.

[18] J. F. Henriques, J. Carreira, R. Caseiro, and J. Batista. Beyond hard negative mining: Efficient detector learning via block-circulant decomposition. In *ICCV*, 2013.

[19] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *ECCV*, 2012.

[20] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *PAMI*, 2015.

[21] X. Jia, H. Lu, and M.-H. Yang. Visual tracking via adaptive structural local sparse appearance model. In *CVPR*, 2012.

[22] Z. Kalal, J. Matas, and K. Mikolajczyk. P-n learning: Bootstrapping binary classifiers by structural constraints. In *CVPR*, 2010.

[23] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, and et al. The visual object tracking vot2014 challenge results. In *ECCV Workshop*, 2014.

[24] Y. Li and J. Zhu. A scale adaptive kernel correlation filter tracker with feature integration. In *ECCV Workshop*, 2014.

[25] OpenCV. The opencv state of the art vision challenge. `http://code.opencv.org/projects/opencv/wiki/VisionChallenge`. Accessed: 2015-09-17.

[26] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan. Locally orderless tracking. In *CVPR*, 2012.

[27] P. Perez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *ECCV*, 2002.

[28] D. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *IJCV*, 77(1):125–141, 2008.

[29] L. Sevilla-Lara and E. G. Learned-Miller. Distribution fields for tracking. In *CVPR*, 2012.

[30] A. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual tracking: An experimental survey. *PAMI*, 36(7):1442–1468, 2014.

[31] J. van de Weijer, C. Schmid, J. J. Verbeek, and D. Larlus. Learning color names for real-world applications. *TIP*, 18(7):1512–1524, 2009.

[32] D. Wang, H. Lu, and M.-H. Yang. Least soft-threshold squares tracking. In *CVPR*, 2013.

[33] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *CVPR*, 2013.

[34] Y. Wu, J. Lim, and M.-H. Yang. Object tracking benchmark. *PAMI*, 2015.

[35] J. Zhang, S. Ma, and S. Sclaroff. MEEM: robust tracking via multiple experts using entropy minimization. In *ECCV*, 2014.

[36] K. Zhang, L. Zhang, and M. Yang. Real-time compressive tracking. In *ECCV*, 2012.

[37] W. Zhong, H. Lu, and M.-H. Yang. Robust object tracking via sparsity-based collaborative model. In *CVPR*, 2012.