Learning Spatial-Temporal Regularized Correlation Filters for Visual Tracking

Feng Li[1], Cheng Tian[1], Wangmeng Zuo *[1], Lei Zhang[2], and Ming-Hsuan Yang[3]

[1]School of Computer Science and Technology, Harbin Institute of Technology, China
[2]Department of Computing, The Hong Kong Polytechnic University, China
[3]School of Engineering, University of California, Merced, USA

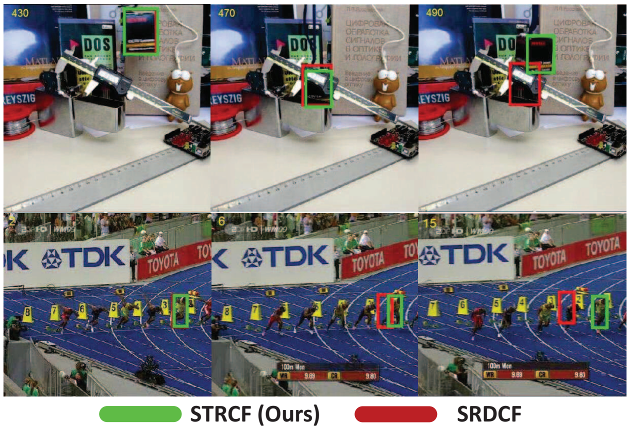
fengli_hit@hotmail.com, tcoperator@163.com, wmzuo@hit.edu.cn, csdzhang@comp.polyu.edu.hk,
mhyang@ucmerced.edu

Abstract

Discriminative Correlation Filters (DCF) are efficient in visual tracking but suffer from unwanted boundary effects. Spatially Regularized DCF (SRDCF) has been suggested to resolve this issue by enforcing spatial penalty on DCF coefficients, which, inevitably, improves the tracking performance at the price of increasing complexity. To tackle online updating, SRDCF formulates its model on multiple training images, further adding difficulties in improving efficiency. In this work, by introducing temporal regularization to SRDCF with single sample, we present our spatial-temporal regularized correlation filters (STRCF). The STRCF formulation can not only serve as a reasonable approximation to SRDCF with multiple training samples, but also provide a more robust appearance model than SRDCF in the case of large appearance variations. Besides, it can be efficiently solved via the alternating direction method of multipliers (ADMM). By incorporating both temporal and spatial regularization, our STRCF can handle boundary effects without much loss in efficiency and achieve superior performance over SRDCF in terms of accuracy and speed. Compared with SRDCF, STRCF with hand-crafted features provides a $5\times$ *speedup and achieves a gain of 5.4% and 3.6% AUC score on OTB-2015 and Temple-Color, respectively. Moreover, STRCF with deep features also performs favorably against state-of-the-art trackers and achieves an AUC score of 68.3% on OTB-2015.*

1. Introduction

Recent years have witnessed the rapid advances of discriminative correlation filters (DCFs) in visual tracking. Benefited from the periodic assumption of training samples, the DCF can be learned very efficiently in the frequency domain via fast Fourier transform (FFT). For example, the tracking speed of the earliest DCF-based tracker, *i.e.*, MOSSE [4], can reach 700 frames per second (FPS). Along with the introduction of feature representation [14, 28], nonlinear kernel [19], scale estimation [11, 23, 24], max-margin classifiers [43], spatial regularization [13, 18], and continuous convolution [15], DCF-based trackers have been greatly improved and significantly advanced the state-of-the-art tracking accuracy. However, such performance improve-



(a)

	SRDCF	SRDCF(-M)	SRDCF(-MS)	KCF	STRCF (HOG)
OTB-2015	72.7	69.3	61.5	55.5	79.2
Temple-Color	62.8	59.1	51.6	47.1	67.8
Avg.OP	67.8	64.2	56.6	51.3	73.5
Avg.FPS	5.3	7.6	32.3	167.4	28.9

(b)

Figure 1: (a) The results of STRCF and SRDCF [13] on two videos with occlusion and deformation. (b) A comparison of SRDCF variants and STRCF using HOG in terms of mean OP (%) and speed (FPS) on OTB-2015 and Temple-Color. The best three results are shown in red, blue and green fonts, respectively.

*Corresponding author.

This work is supported by NSFC grants (No.s 61671182 and 61471146), NSF CAREER (No. 1149783) and HK RGC GRF grant (PolyU 152240/15E).

ment is not obtained without any extra cost. Most top-ranked trackers, *e.g.*, SRDCF [13] and C-COT [15], have gradually lost the characteristic speed and realtime capability of early DCF-based trackers. For example, the speed of SRDCF [13] using the hand-crafted HOG feature is ∼6 FPS, while that of the baseline KCF [19] is ∼170 FPS.

For better understanding on this issue, we dissect the tradeoff between accuracy and speed in SRDCF. In general, the inefficiency of SRDCF can be attributed to three factors: (i) scale estimation, (ii) spatial regularization, and (iii) formulation on large training set. Fig. 1b lists the tracking speed and accuracy of SRDCF and its variants on two popular benchmarks, including SRDCF($-$M) (*i.e.*, removing (iii)), SRDCF($-$MS) (*i.e.*, removing (ii)&(iii)), and KCF (*i.e.*, removing (i)&(ii)&(iii)). We note that when removing (iii), linear interpolation [4, 11] is adopted as an alternative strategy for online model updating. From Fig. 1b, it can be seen that the tracker still maintains its real-time ability ($\sim 33FPS$) when adding scale estimation. But the tracking speed decreases significantly with the further introduction of spatial regularization and formulation on large training set. Therefore, it is valuable to develop a solution for taking use of (ii) and (iii) without much loss in efficiency.

In this paper, we study the solution for taking the benefit of spatial regularization and formulation on large training set without much loss in efficiency. On the one hand, the high complexity of SRDCF mainly comes at the formulation on multiple training images. By removing the constraint, SRDCF with single image can be efficiently solved via ADMM. Due to the convexity of SRDCF, the ADMM can also guarantee to converge to global optimum. On the other hand, in SRDCF spatial regularization is integrated into the formulation on multiple training images for the coupling of DCF learning and model updating, which does benefit the tracking accuracy. Motivated by online Passive-Aggressive (PA) learning [6], we introduce a temporal regularization to SRDCF with single image, resulting in our spatial-temporal regularized correlation filters (STRCF). STRCF is a rational approximation of the full SRDCF formulation on multiple training images, and can be exploited for simultaneous DCF learning and model updating. Besides, the ADMM algorithm can also be directly used to solve STRCF. Thus, STRCF incorporates both spatial and temporal regularization into DCF, and can be adopted to speed up SRDCF.

Furthermore, as an extension of online PA algorithm [6], STRCF can also provide a more robust appearance model than SRDCF in the case of significant appearance variations. Fig. 1a illustrates the tracking results on two sequences with occlusion and deformation. Compared with SRDCF, we can see that, with the introduction of the temporal regularization, STRCF performs more robustly to occlusion while adapting well to large appearance variation. From Fig. 1b, STRCF not only runs at real-time tracking speed ($\sim 30FPS$), but also leads to $+5.7\%$ performance gain over SRDCF by average mean OP on two datasets. To sum up, STRCF can achieve remarkable improvements over the baseline SRDCF on all the datasets, and runs at more than $5\times$ faster tracking speed.

We perform comparative experiments on several benchmarks, including OTB-2015 [40], Temple-Color [25], and VOT-2016 [22]. STRCF performs favorably in terms of accuracy, robustness and speed in comparison with the state-of-the-art CF-based and CNN trackers.

The contributions of this paper are as follows:

- A STRCF model is presented by incorporating both spatial and temporal regularization into the DCF framework. Based on online PA, STRCF can not only serve as a rational approximation of the SRDCF formulation on multiple training images, but also provide a more robust appearance model than SRDCF in the case of large appearance variations.
- An ADMM algorithm is developed for solving STRCF efficiently, where each sub-problem has the closed-form solution. And our algorithm can empirically converge within very few iterations.
- Our STRCF with hand-crafted feature can run in real-time, achieves notable improvements over SRDCF by tracking accuracy. Furthermore, our STRCF with deep features performs favorably in comparison with the state-of-the-art trackers [9, 15].

2. Related Work

This section first provides a brief survey on DCF trackers and then focuses on spatial regularization and formulation on large training set that are most relevant to our STRCF.

2.1. Discriminative Correlation Filters

Using DCFs for adaptive tracking starts with MOSSE [4], which learns the CFs with few samples in the frequency domain. Notable improvements have been made to this popular tracker to address several limiting issues. For example, Henriques *et al*. [19] learn the kernelized CFs (KCF) via kernel trick. The multi-channel version of MOSSE is also studied in [21]. And more discriminative features are widely used, such as HOG [8], color names (CN) [14] and deep CNN features [28, 34]. To cope with the size change and occlusion, several scale-adaptive [11, 23, 24] and part-based trackers [26, 27] are further investigated. Besides, long-term tracking [29], continuous convolution [15] and particle filter based methods [42] are also developed to improve the tracking accuracy and robustness. Due to the space limitation, here we only review the methods from spatial regularization and formulation on large training set that are close to our algorithm.

2.2. Spatial Regularization

The circulant shifted samples in DCF-based trackers always suffer from periodic repetitions on boundary positions, thereby significantly degrading the tracking performance. Several spatial regularization methods have been suggested to alleviate the unwanted boundary effects. Galoogahi *et al*. [18] pre-multiply the image patches with a fixed masking matrix containing the target regions, and then solve the constrained optimization problem via ADMM. However, their method can only be applied to single channel DCFs. Danelljan *et al*. [13] propose a spatial regularization term to penalize the DCF coefficients depending on their spatial locations and suggest the Gauss-Seidel algorithm to solve the resulting normal equations. The work [7] also employs a similar spatial regularization term, but the spatial regularization matrix is predicted with a multi-directional RNN for identifying the reliable components. These two methods, however, are unable to exploit the circulant structure in learning, resulting in higher computational cost. More recently, Galoogahi *et al*. [17] extend [18] to multiple channels and further speed up the tracker towards real-time. Compared with these methods, our STRCF has several merits: (1) while STRCF serves as an approximation to [13] on multiple training samples, it can be solved more efficiently with the proposed ADMM algorithm. (2) with the introduction of the temporal regularization, STRCF can learn a more robust appearance model than [13, 17], thereby leading to superior tracking performance.

2.3. Formulation on large training set

One of the most critical challenges in visual tracking is to learn and maintain a robust and fast appearance model in the case of large appearance variations. To this end, MOSSE [4] implements simultaneous DCF learning and model updating by learning the CFs with multiple training samples from historical tracking results. Similar strategy of incorporating large training set into the formulation can also be found in [10, 12, 15, 21]. In practice, robust CFs can be learned by taking the samples at different time instances into consideration. However, this leads to superior performance at the price of higher computational burden. In comparison with these methods, KCF [19] and its variants [3, 11] decouple the DCF learning and model updating, and further exploit the circulant structure for high efficiency. As a result, KCF with HOG feature can run at more than 150 FPS on a single CPU. Following this work, there also exist several heuristic methods [27, 38] to address the naive model updating issues. These methods, however, obtain inferior performance than DCF-based trackers with large training set. Compared with these trackers, STRCF can not only be solved efficiently by avoiding the deployment of large training set, but also benefit from simultaneous DCF learning and model updating by introducing the temporal regularization.

3. Spatially Regularized DCF

In this section, we first revisit the SRDCF tracker, and then present our STRCF model motivated by online PA. Finally, an ADMM is developed to solve the STRCF model.

3.1. Revisit SRDCF

Denote by $\mathcal{D} = \{(\mathbf{x}_k \, \mathbf{y}_k)\}_{k=1}^{T}$ a training set of multiple images. Each sample $\mathbf{x}_k = [\mathbf{x}_k^1, ..., \mathbf{x}_k^D]$ consists of D feature maps with size of $M \times N$. And $\mathbf{y}_k$ is the predefined Gaussian shaped labels. The SRDCF [13] is formulated by minimizing the following objective,

$$\arg\min_{\mathbf{f}} \sum_{k=1}^{T} \alpha_k \left\| \sum_{d=1}^{D} \mathbf{x}_k^d * \mathbf{f}^d - \mathbf{y}_k \right\|^2 + \sum_{d=1}^{D} \left\| \mathbf{w} \cdot \mathbf{f}^d \right\|^2, \qquad (1)$$

where $\cdot$ denotes the Hadamard product, $*$ stands for the convolution operator, $\mathbf{w}$ and $\mathbf{f}$ are the spatial regularization matrix and correlation filter, respectively. α_k indicates the weight to each sample $\mathbf{x}_k$ and is set to emphasize more to the recent samples. In [13], Danelljan *et al*. employ the Gauss-Seidel method to iteratively update the filters $\mathbf{f}$. Please refer to [13] for more implementation details.

However, although SRDCF is effective in suppressing the adverse boundary effects, it also increases the computational burden due to the following two reasons:

(***i***) *The failure of exploiting circulant matrix structure*. For the sake of learning a robust correlation filter $\mathbf{f}$, DCF trackers incorporate several historical samples $\{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=1}^{T}$ for training [11]. However, unlike other CF-based trackers learned with only the sample from the current frame, the formulation on multiple images breaks the circulant matrix structure, resulting in high computation burden. As for SRDCF, the optimization becomes even more difficult due to the spatial regularization term.

(***ii***) *The large linear equations and Gauss-Seidel solver*. Eqn. (1) results in a $DMN \times DMN$ large sparse linear equation system. While the Gauss-Seidel method is suggested to solve Eqn. (1) using the property of sparse matrix, it still remains high computational complexity. In addition, the SRDCF tracker also needs a long start-up time to learn the discriminative correlation filters in the first frame due to the low convergence speed of Gauss-Seidel method.

Both the spatial regularization and formulation on multiple images will break the circulant matrix structure. Fortunately, these two issues can be circumvented to improve the tracking speed. The formulation on multiple images can be relaxed to a STRCF model on single image by introducing the temporal regularization. Furthermore, the introduction of spatial regularization can be addressed by exploiting an equivalent reformulation solved by ADMM efficiently.

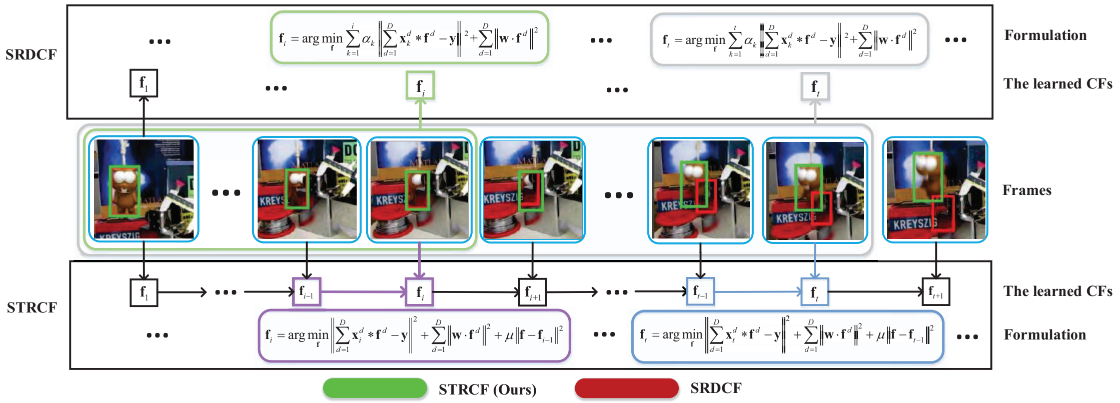

SRDCF
$\mathbf{f}_i = \arg\min_{\mathbf{f}} \sum_{k=1}^{i} \alpha_k \left\| \sum_{d=1}^{D} \mathbf{x}_k^d * \mathbf{f}^d - \mathbf{y} \right\|^2 + \sum_{d=1}^{D} \left\| \mathbf{w} \cdot \mathbf{f}^d \right\|^2$
$\mathbf{f}_t = \arg\min_{\mathbf{f}} \sum_{k=1}^{t} \alpha_k \left\| \sum_{d=1}^{D} \mathbf{x}_k^d * \mathbf{f}^d - \mathbf{y} \right\|^2 + \sum_{d=1}^{D} \left\| \mathbf{w} \cdot \mathbf{f}^d \right\|^2$
Formulation
$\mathbf{f}_1$
$\mathbf{f}_i$
$\mathbf{f}_t$
The learned CFs
Frames
STRCF
$\mathbf{f}_1$
$\mathbf{f}_{i-1}$
$\mathbf{f}_i$
$\mathbf{f}_{i+1}$
$\mathbf{f}_{t-1}$
$\mathbf{f}_t$
$\mathbf{f}_{t+1}$
The learned CFs
$\mathbf{f}_i = \arg\min_{\mathbf{f}} \left\| \sum_{d=1}^{D} \mathbf{x}_i^d * \mathbf{f}^d - \mathbf{y} \right\|^2 + \sum_{d=1}^{D} \left\| \mathbf{w} \cdot \mathbf{f}^d \right\|^2 + \mu \left\| \mathbf{f} - \mathbf{f}_{i-1} \right\|^2$
$\mathbf{f}_t = \arg\min_{\mathbf{f}} \left\| \sum_{d=1}^{D} \mathbf{x}_t^d * \mathbf{f}^d - \mathbf{y} \right\|^2 + \sum_{d=1}^{D} \left\| \mathbf{w} \cdot \mathbf{f}^d \right\|^2 + \mu \left\| \mathbf{f} - \mathbf{f}_{t-1} \right\|^2$
Formulation
🟩 STRCF (Ours)
🟥 SRDCF

Figure 2: A comparison of SRDCF and STRCF on model learning. SRDCF learns the CFs with multiple samples from historical tracking results and emphasizes more to the recent samples. Thus it may suffer from over-fitting to the recent inaccurate samples and results in tracking failure in the case of occlusion. In contrast, our STRCF trains the CF $\mathbf{f}_t$ with the sample from current frame and the learned CF $\mathbf{f}_{t-1}$. Benefited from online PA, STRCF can successfully follow the targets by *passively* updating the CFs in the case of occlusion.

3.2. STRCF

In online classification, when a new instance comes on each round, the algorithm first predicts its label, and then updates the classifier based on the newly instance-label pair. On the one hand, the learning algorithm should be *passive* to make the updated classifier similar to the previous one. On the other hand, the learning algorithm should be *aggressive* to guarantee the new instance be corrected classified. Thus, Crammer *et al.* [6] suggest an online passive-aggressive (PA) algorithm by introducing a temporal regularization, and derive the bound on the cumulative loss of PA w.r.t. the best fixed predictor.

Motivated by PA, we introduce a temporal regularization term $\|\mathbf{f} - \mathbf{f}_{t-1}\|^2$, resulting in our spatial-temporal regularized CF (STRCF) model,

$$\arg\min_{\mathbf{f}} \frac{1}{2} \left\| \sum_{d=1}^{D} \mathbf{x}_t^d * \mathbf{f}^d - \mathbf{y} \right\|^2 + \frac{1}{2} \sum_{d=1}^{D} \left\| \mathbf{w} \cdot \mathbf{f}^d \right\|^2 + \frac{\mu}{2} \left\| \mathbf{f} - \mathbf{f}_{t-1} \right\|^2, \tag{2}$$

where $\mathbf{f}_{t-1}$ denotes the CFs utilized in the $(t-1)$-th frame, and μ denotes the regularization parameter. Here, $\sum_{d=1}^{D} \|\mathbf{w} \cdot \mathbf{f}^d\|^2$ denotes the spatial regularizer, and $\|\mathbf{f} - \mathbf{f}_{t-1}\|^2$ denotes the temporal regularizer.

STRCF can also be treated as an extension of online PA from two aspects: (i) Instead of classification, STRCF is an online learning of linear regression; (ii) Instead of instance-wise updating, the samples in STRCF come at the batch level (*i.e.* all the shift versions of an image) on each round. Therefore, STRCF naturally inherits the merits of online PA on adaptively balancing the tradeoff between aggressive and passive model learning, thus leading to more robust models in the case of large appearance variations. In Fig. 2, we compare STRCF with SRDCF on sequence *Lemming* to highlight their relationships on CF model learning. From it we can make the following observations: (i) Similar to SRDCF, STRCF also implements simultaneous DCF learning and model updating with the introduction of temporal regularizer, thus can serve as a rational approximation of SRDCF with multiple training samples; (ii) In the case of occlusion, while SRDCF suffers from over-fitting to recent corrupted samples, STRCF can alleviate this by *passively* updating the CFs to keep it close to the previous ones.

3.3. Optimization algorithm

The model in Eqn. (2) is convex, and can be minimized to obtain the globally optimal solution via ADMM. To this end, we first introduce an auxiliary variable $\mathbf{g}$ by requiring $\mathbf{f} = \mathbf{g}$ and the stepsize parameter γ, then the Augmented Lagrangian form of Eqn. (2) can be formulated as

$$\mathcal{L}(\mathbf{w}, \mathbf{g}, \mathbf{s}) = \frac{1}{2} \left\| \sum_{d=1}^{D} \mathbf{x}_t^d * \mathbf{f}^d - \mathbf{y} \right\|^2 + \frac{1}{2} \sum_{d=1}^{D} \left\| \mathbf{w} \cdot \mathbf{g}^d \right\|^2 \tag{3}$$

$$+ \sum_{d=1}^{D} (\mathbf{f}^d - \mathbf{g}^d)^T \mathbf{s}^d + \frac{\gamma}{2} \sum_{d=1}^{D} \left\| \mathbf{f}^d - \mathbf{g}^d \right\|^2 + \frac{\mu}{2} \|\mathbf{f} - \mathbf{f}_{t-1}\|^2,$$

where $\mathbf{s}$, μ are the Lagrange multiplier and penalty factor, respectively. By introducing $\mathbf{h} = \frac{1}{\gamma}\mathbf{s}$, Eqn. (3) can be reformulated as

$$\mathcal{L}(\mathbf{w}, \mathbf{g}, \mathbf{h}) = \frac{1}{2} \left\| \sum_{d=1}^{D} \mathbf{x}_t^d * \mathbf{f}^d - \mathbf{y} \right\|^2 + \frac{1}{2} \sum_{d=1}^{D} \left\| \mathbf{w} \cdot \mathbf{g}^d \right\|^2 \tag{4}$$

$$+ \frac{\gamma}{2} \sum_{d=1}^{D} \left\| \mathbf{f}^d - \mathbf{g}^d + \mathbf{h}^d \right\|^2 + \frac{\mu}{2} \|\mathbf{f} - \mathbf{f}_{t-1}\|^2.$$

The ADMM algorithm is then adopted by alternatingly solving the following subproblems,

$$\begin{cases} \mathbf{f}^{(i+1)} = \arg\min_{\mathbf{f}} \left\| \sum_{d=1}^{D} \mathbf{x}_t^d * \mathbf{f}^d - \mathbf{y} \right\|^2 + \gamma \|\mathbf{f} - \mathbf{g} + \mathbf{h}\|^2 + \mu \|\mathbf{f} - \mathbf{f}_{t-1}\|^2 \\ \mathbf{g}^{(i+1)} = \arg\min_{\mathbf{g}} \sum_{d=1}^{D} \left\| \mathbf{w} \cdot \mathbf{g}^d \right\|^2 + \gamma \|\mathbf{f} - \mathbf{g} + \mathbf{h}\|^2 \\ \mathbf{h}^{(i+1)} = \mathbf{h}^{(i)} + \mathbf{f}^{(i+1)} - \mathbf{g}^{(i+1)}. \end{cases} \tag{5}$$

We detail the solution to each subproblem as follows: **Subproblem f**: Using the Parseval's theorem, the first row of Eqn. (5) can be rewritten in the Fourier domain as

$$\operatorname*{argmin}_{\hat{\mathbf{f}}} \left\| \sum_{d=1}^{D} \hat{\mathbf{x}}_t^d \cdot \hat{\mathbf{f}}^d - \hat{\mathbf{y}} \right\|^2 + \gamma \left\| \hat{\mathbf{f}} - \hat{\mathbf{g}} + \hat{\mathbf{h}} \right\|^2 + \mu \left\| \hat{\mathbf{f}} - \hat{\mathbf{f}}_{t-1} \right\|^2, \tag{6}$$

where $\hat{\mathbf{f}}$ denotes the discrete Fourier transform (DFT) of the filter $\mathbf{f}$. From Eqn. (6), we can see that the j-th element of the label $\hat{\mathbf{y}}$ only depends on the j-th element of the filter $\hat{\mathbf{f}}$ and sample $\hat{\mathbf{x}}_t$ across all D channels. Denote by $\mathcal{V}_j(\mathbf{f}) \in \mathbb{R}^D$ the vector consisting of the j-th elements of $\mathbf{f}$ along all D channels. Eqn. (6) can be further decomposed into MN subproblems, where each of them is defined as

$$\begin{aligned} \operatorname*{argmin}_{\mathcal{V}_j(\hat{\mathbf{f}})} & \left\| \mathcal{V}_j(\hat{\mathbf{x}}_t)^\top \mathcal{V}_j(\hat{\mathbf{f}}) - \hat{y}_j \right\|^2 + \mu \left\| \mathcal{V}_j(\hat{\mathbf{f}}) - \mathcal{V}_j(\hat{\mathbf{f}}_{t-1}) \right\|^2 \\ & + \gamma \left\| \mathcal{V}_j(\hat{\mathbf{f}}) - \mathcal{V}_j(\hat{\mathbf{g}}) + \mathcal{V}_j(\hat{\mathbf{h}}) \right\|^2. \end{aligned} \tag{7}$$

Taking the derivative of Eqn. (7) be zero, we can get the closed-form solution for $\mathcal{V}_j(\hat{\mathbf{f}})$,

$$\mathcal{V}_j(\hat{\mathbf{f}}) = (\mathcal{V}_j(\hat{\mathbf{x}}_t)\mathcal{V}_j(\hat{\mathbf{x}}_t)^\top + (\mu + \gamma)I)^{-1}\mathbf{q}, \tag{8}$$

where the vector $\mathbf{q}$ takes the form as $\mathbf{q} = \mathcal{V}_j(\hat{\mathbf{x}}_t)\hat{y}_j + \gamma\mathcal{V}_j(\hat{\mathbf{g}}) - \gamma\mathcal{V}_j(\hat{\mathbf{h}}) + \mu\mathcal{V}_j(\hat{\mathbf{f}}_{t-1})$. Since $\mathcal{V}_j(\hat{\mathbf{x}}_t)\mathcal{V}_j(\hat{\mathbf{x}}_t)^\top$ is rank-1 matrix, Eqn. (8) can be solved with the Sherman-Morrsion formula [33], and we have

$$\mathcal{V}_j(\hat{\mathbf{f}}) = \frac{1}{\mu + \gamma}\left(I - \frac{\mathcal{V}_j(\hat{\mathbf{x}}_t)\mathcal{V}_j(\hat{\mathbf{x}}_t)^\top}{\mu + \gamma + \mathcal{V}_j(\hat{\mathbf{x}}_t)^\top\mathcal{V}_j(\hat{\mathbf{x}}_t)}\right)\mathbf{q}. \tag{9}$$

Note that Eqn. (9) only contains vector multiply-add operation and thus can be computed efficiently. $\mathbf{f}$ can be further obtained by the inverse DFT of $\hat{\mathbf{f}}$.
Subproblem g: From the second sub-equation of Eqn. (5), each element of $\mathbf{g}$ can be computed independently, and thus the closed-form solution of $\mathbf{g}$ can be computed by,

$$\mathbf{g} = (\mathbf{W}^\top\mathbf{W} + \gamma I)^{-1}(\gamma\mathbf{f} + \gamma\mathbf{h}). \tag{10}$$

where $\mathbf{W}$ represents the $DMN \times DMN$ diagonal matrix concatenated with D diagonal matrices $\mathrm{Diag}(\mathbf{w})$.
Updating stepsize parameter γ: The stepsize parameter γ is updated as in Eqn. (11),

$$\gamma^{(i+1)} = min(\gamma^{\max}, \rho\gamma^{(i)}), \tag{11}$$

where $\gamma^{\max}$ denotes the maximum value of γ and the scale factor ρ.
Complexity Analysis. Since Eqn. (6) is separable in each pixel location, we should solve MN subproblems and each is a system of linear equations with D variables. With Sherman-Morrison formula, each system can be solved in $\mathcal{O}(D)$. [1]Thus, the complexity of solving $\hat{\mathbf{f}}$ is $\mathcal{O}(DMN)$. Taking the DFT and inverse DFT into account, the complexity of solving $\mathbf{f}$ is $\mathcal{O}(DMN\log(MN))$. And the computational cost for $\mathbf{g}$ is $\mathcal{O}(DMN)$. Hence, the overall cost of our algorithm is $\mathcal{O}(DMN\log(MN)N_I)$, where N_I represents the maximum number of iterations. In addition, compared with SRDCF, our ADMM algorithm does not need a start-up time to initialize the CFs in the first frame.
Convergence. Note that the STRCF model is convex, and each sub-problem in ADMM algorithm has closed-form solution. Therefore, it satisfies the Eckstein-Bertsekas condition [16], and is guaranteed to converge to global optimum. In addition, We empirically find that the proposed ADMM can converge within 2 iterations on most of the sequences, and thus N_I is set to 2 for efficiency.

4. Experimental Results

In this section, we first compare our STRCF with the state-of-the-art trackers in terms of both hand-crafted and CNN features on the OTB-2015 dataset. Then, we analyze the impacts of the temporal regularization and hyper-parameter μ on tracking performance using OTB-2015. Finally, we conduct comparative experiments on Temple-Color and VOT-2016 benchmarks.

Following the settings in SRDCF [13], we crop the square region centered at the target, in which the side length of the region is $\sqrt{5WH}$ (W and H represent the width and height of the target, respectively). Then we extract HOG, CN [14] and CNN features for the image region. The features are further weighted by a cosine window to reduce the boundary discontinuities. As for the ADMM algorithm, we set the hyper-parameter in Eqn. (2) to $\mu = 16$ throughout all the experiments. The initial stepsize parameter $\gamma^{(0)}$, the maximum value $\gamma^{\max}$ and scale factor ρ are set to 10, 100 and 1.2, respectively. Our STRCF is implemented with Matlab 2017a and all the experiments are run on a PC equipped with Intel i7 7700 CPU, 32GB RAM and a single NVIDIA GTX 1070 GPU. The source code of our tracker is publicly available at https://github.com/lifeng9472/STRCF.

4.1. The OTB-2015 benchmark

The OTB-2015 benchmark [40] is a popular tracking dataset which consists of 100 fully annotated video sequences with 11 different attributes, such as abrupt motion, illumination variation, scale variation and motion blurring. We evaluate the trackers based on the One Pass Evaluation (OPE) protocol provided in [40], where overlap precision (OP) metric is employed by calculating the bounding box overlaps exceeding 0.5 in a sequence. Besides, we also provide the overlap success plots containing the OP metric over a range of thresholds.

We compare STRCF with 20 state-of-the-art trackers, including trackers using hand-crafted features (*i.e.* SRDCF

[1] please refer to [43] for more details.

	SRDCF [13]	BACF [20]	ECO-HC [9]	SRDCFDecon [10]	Staple [1]	Staple+CA[30]	SAMF+AT [3]	SAMF [24]	MEEM [41]	DSST [11]	KCF [19]	STRCF (HOG)	STRCF (HOGCN)
Mean OP	72.7	77.5	79.6	77	71	73.8	68	64.4	62.3	62.2	55.5	79.2	79.6
FPS	5.8	26.7	42	2.0	76.6	35.3	2.2	23.2	22.4	20.4	171.8	31.5	24.3

Table 1: The mean OP (in %) and FPS results of trackers with hand-crafted features on OTB-2015. The best three results are shown in red, blue and green fonts, respectively.

	ECO [9]	DeepSRDCF [13]	SiameseFC [2]	FCNT [37]	HDT [34]	MSDAT [39]	HCF [28]	C-COT [15]	CF-Net [36]	DeepSTRCF
Mean OP	85.5	76.8	71	67.1	65.8	65.6	65.6	82.7	73	84.2
FPS	9.8	0.2	83.7	1.2	2.7	25	10.2	0.8	78.4	5.3

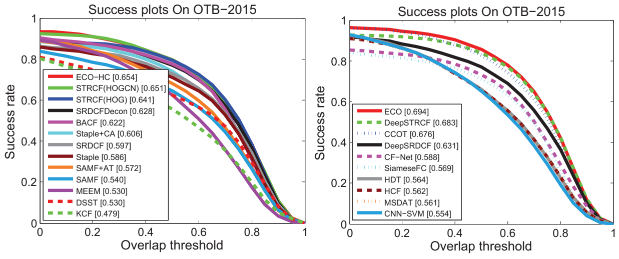
Table 2: The OP metric (in %) and FPS results of trackers with deep features on OTB-2015. The best three results are shown in red, blue and green fonts, respectively.

[13], BACF [20], ECO-HC [9], SRDCFDecon [10], Staple [1], Staple+CA[30], SAMF+AT [3], DSST [11], SAMF [24], MEEM [41] and KCF [19]) and using CNN features (*i.e.* ECO [9], DeepSRDCF [12], HCF [28], HDT [34], C-COT [15], FCNT [37], SiameseFC [2], CF-Net [2] and MSDAT [39]). Note that we employ the publicly available codes or results provided by the authors for fair comparison.

4.1.1 Comparison with hand-crafted based trackers

We compare the proposed STRCF with other state-of-the-art trackers using hand-crafted features. Table 1 gives the results of the mean OP and FPS on OTB-2015. As shown in Table 1, STRCF performs significantly better than most of the competing trackers except ECO-HC and surpasses its counterpart SRDCF by 6.9%. We owe these significant improvements to the introduction of the temporal regularization. STRCF is also superior to the SRDCFDecon tracker which follows the SRDCF work and addresses the corrupted sample problem by re-weighting the samples in the training set. It indicates that the introduction of temporal regularization is more helpful than multiple samples training with explicit sample re-weighting. Besides, our method also outperforms the recent CF-based trackers: BACF [17], SAMF+AT [3] and Staple+CA [30]. Overall, the only tracker performing comparably with STRCF on OTB-2015 is the ECO-HC [9]. It is worth noting that ECO-HC adopts the Gaussian Mixture Model (GMM)-based generative sample space method to reduce the number of samples for training, and employs continuous convolution and factorized convolution for boosting the performance. In contrast, even our STRCF does not consider continuous convolution and factorized convolution techniques, it still yields favorable performance against the competing trackers.

In addition, we also report the tracking speed (FPS) comparison on OTB-2015 dataset in Table 1. One can see that STRCF (HOGCN) runs at 24.3 FPS and is nearly 4.2× than its counterpart SRDCF (5.8 FPS), validating the high efficiency of the proposed ADMM over the SRDCF solver (*i.e.* the Gauss-Seidel algorithm). STRCF (HOG) using HOG feature performs even faster and obtains a real-time speed of 31.5 FPS, which is 1.2× faster than recent BACF tracker.



(a) (b)

Figure 3: A comparison of the overlap success plots with the state-of-the-art trackers on OTB-2015 dataset. (a) Trackers with hand-crafted features. (b) Trackers with deep features.

Finally, we provide the overlap success curve of the competing trackers with the hand-crafted features on OTB-2015 dataset, which is ranked using the Area-Under-the-Curve (AUC) score. As shown in Fig. 3a, our STRCF achieves an AUC score of 65.1% and ranks the second best performance among all the trackers. Similar to the mean OP results, STRCF also outperforms its counterparts SRDCF and SRDCFDecon by a gain of 5.4% and 2.3%, respectively.

4.1.2 Video Attribute Based Comparison

In this section, we perform quantitative analysis of the total 11 video attributes on the OTB-2015 dataset. Our STRCF outperforms most of the competing trackers except ECO-HC on all the attributes. Due to the page limits, here we only provide the overlap success plots of 4 attributes in Fig. 4 and the remaining results can be found in the supplementary material.

In the case of out of view (OV) and occlusion (OCC), the target always encounters with partial or fully disappearance from the camera, which leads to an adverse impact on model updating. Trackers using multiple samples training with naive sample weighting strategy (*i.e.* SRDCF) or linear interpolation updating (*i.e.* Staple and SAMF+AT) suffer from significant degradation because of over-fitting to the recent samples. Benefited from the temporal regularization, our STRCF can adaptively make the balance between

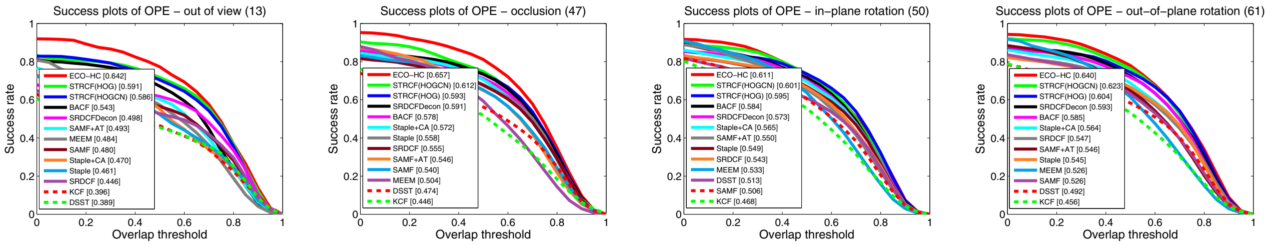


Figure 4: The overlap success plots of the competing trackers with 4 video attributes on the OTB-2015 dataset.

updating the CFs with the latest samples and keeping close to the previously learned CFs, and thus is robust to such kinds of variations. In particular, STRCF achieves remarkable improvements over its baseline SRDCF, i.e., 14.5% and 5.7% gains on these two attributes, respectively. And it also outperforms the SRDCFDecon tracker by an AUC score of 9.3% and 2.1% on these two attributes. As for the In-plane/Out-of-plane rotation attributes, STRCF also performs better than most of the trackers and is superior to the baseline SRDCF by 5.8% and 7.6%, respectively.

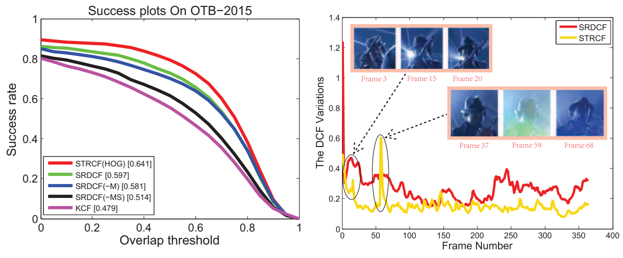
4.1.3 Comparison with deep feature-based trackers

To further assess STRCF, we follow the settings in C-COT [15], and combine the outputs of *conv*3 layer from VGG-M network [35] with HOGCN features for STRCF training (we name it as DeepSTRCF for simplicity). Using mean OP and speed as performance metrics, Table 2 compares DeepSTRCF with the state-of-the-art trackers based on deep features on OTB-2015. One can see that DeepSTRCF achieves a mean OP of 84.2% and performs much better than the SRDCF with CNN features (*i.e*. DeepSRDCF) by a gain of 7.4%, demonstrating the effectiveness of the temporal regularization. It even outperforms than C-COT with both spatial regularization and continuous convolution by a gain of 1.2% on OTB-2015. In terms of the tracking speed, the best performance belongs to SiameseFC (83.7 FPS), followed by CF-Net (78.4 FPS) and MSDAT (25 FPS), while DeepSTRCF runs at 5.3 FPS. The higher speed of these trackers, however, comes at the cost of much lower accuracy in comparison to STRCF. Furthermore, We also provide the overlap success curves of the competing trackers in Fig. 3b. One can see that DeepSTRCF ranks the second and outperforms DeepSRDCF with a margin of 5.2% on OTB-2015.

4.2. Internal Analysis of the proposed approach

4.2.1 Impacts of the Temporal regularization

In this section, we investigate the impacts of the temporal regularization on the proposed STRCF approach using the OTB-2015 dataset. Fig. 5a gives the overlap success plot of different SRDCF variants (discussed in Section 1) and our STRCF. Compared with the KCF method, we can see that the introduction of scale estimation (*i.e*. SRDCF(-MS)) and spatial regularization (*i.e*. SRDCF(-M)) can boost the performance by 3.5% and 6.7%, respectively. Besides, SRDCF also outperforms SRDCF(-M) by 1.6% with the coupling of DCF learning and model updating. However, when incorporating the temporal regularization into SRDCF(-M) formulation, our STRCF can bring notable improvements over both SRDCF(-M) and SRDCF with a gain of 6% and 4.4%, respectively. This can be explained by the merits of online PA on adaptively balancing the tradeoff between aggressive and passive model updating.



(a) (b)

Figure 5: Ablative study on the STRCF method. (a) The overlap success plot of the SRDCF variants and our STRCF on OTB-2015. (b) The visualization results of temporal CF variation against frames on sequence *Shaking*.

To further illustrate the differences of STRCF and SRDCF on model learning, we visualize the temporal CF variation (*i.e*. $\frac{\|\mathbf{f}_t - \mathbf{f}_{t-1}\|^2}{z}$, where z is the normalization factor) against frames on sequence *Shaking* in Fig. 5b. From it we can draw the following conclusions: (1) Compared with the SRDCF tracker, our STRCF *passively* updates the CFs in most frames with small appearance variations, thus leading to more robust DCF variations. (2) While SRDCF suffers from slow appearance variations (*i.e*. occlusion in the 3∼20-th frames), our STRCF is dominated by the *passive* model learning and thus insensitive to these variations. (3) In the case of sudden appearance variations (*i.e*. the illumination changes in the 58∼68-th frames), STRCF can benefit from the *aggressive* model learning and better adapt to these situations than SRDCF. It should be noted that these phenomena are ubiquitous in various video attributes, and the visualizations and analysis of the temporal CF variations on more videos can be found on the supplementary material.

	DSST [11]	ECO [9]	Staple [1]	MDNet_N [32]	TCNN [31]	SRDCF [13]	BACF [20]	SRDCFDecon [10]	DeepSRDCF [12]	ECO-HC [9]	STRCF	DeepSTRCF
EAO	0.181	0.375	0.295	0.257	0.325	0.247	0.223	0.262	0.276	0.322	0.279	0.313
Accuracy	0.5	0.53	0.54	0.53	0.54	0.52	0.56	0.53	0.51	0.54	0.53	0.55
Robustness	2.72	0.73	1.35	1.2	0.96	1.5	1.88	1.42	1.17	1.08	1.32	0.92

Table 3: A comparison with the state-of-the-art trackers on VOT-2016 dataset.

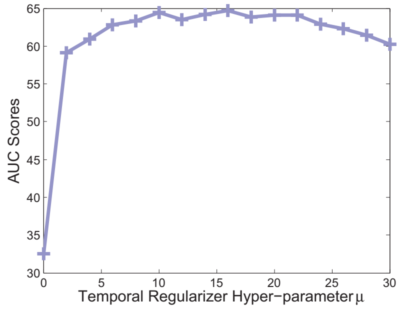


Figure 6: Impacts of the temporal regularization parameter μ on OTB-2015 dataset.

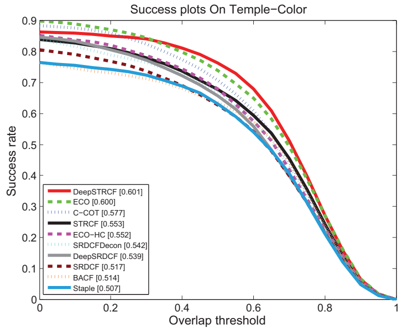


Figure 7: The overlap success plot of different trackers on Temple-Color. Only the top 10 trackers are displayed for clarity.

4.2.2 Effect of regularization parameter μ

We further analyze the effect of regularization parameter μ on the tracking performance of STRCF with hand-crafted features. The regularization parameter μ determines the rate at which to replace the learned CF $\mathbf{f}_{t-1}$ from previous frames with the new sample $\mathbf{x}$ in the current frame. The lower the parameter μ, the higher relevance of filter $\mathbf{f}$ given to the sample $\mathbf{x}$. In Fig. 6, it is shown that the accuracy of STRCF tracker is significantly affected by the choice of μ. From Fig. 6, the best performance is achieved around $\mu = 16$. Note that when $\mu = 0$, STRCF is trained only with the current frame and ignores all historical information, thus it even performs worse than KCF.

4.3. The Temple-Color Benchmark

We perform comparative experiments on Temple-Color dataset [25] which consists of 128 color sequences. We compare STRCF and DeepSTRCF with the state-of-the-art trackers mentioned above except CF-Net [36] which trained the network on Temple-Color . Fig. 7 shows the comparison of overlap success plots for different trackers. We note that STRCF is on par with ECO-HC and surpasses its counterparts SRDCF, DeepSRDCF and SRDCFDecon by 3.6%, 1.4% and 1.1%, respectively. Meanwhile, DeepSTRCF performs the best among the competing trackers and achieves an AUC score of 60.1%, further demonstrating the effectiveness of STRCF on deep features.

4.4. The VOT-2016 Benchmark

We also report the results on Visual Object Tracking 2016 benchmark (VOT-2016) [22], which consists of 60 challenging videos. We evaluate the trackers in terms of accuracy, robustness and expected average overlap (EAO) [5]. The accuracy measures the average overlap ratio between the predicted bounding box and the ground-truth. The robustness computes the average number of tracking failures over the sequence. And EAO averages the no-reset overlap of a tracker on several short-term sequences.

We compare STRCF and DeepSTRCF with state-of-the-art trackers, including MDNet [32] (VOT-2015 winner) and TCNN [31] (VOT-2016 winner). Table 3 lists the results of different trackers on VOT-2016 dataset. We can see from Table 3 that STRCF performs significantly better than the BACF and SRDCF methods in terms of the EAO metric. In addition, DeepSTRCF also performs favorably against its counterpart DeepSRDCF by a gain of 3.7% in EAO metric.

5. Conclusion

In this paper, we propose the spatial-temporal regularized correlation filters (STRCF) to address the inefficiency problem of SRDCF. By introducing the temporal regularizer to SRDCF formulation with single sample, STRCF serves as an approximation of SRDCF with multiple training samples. Moreover, as an extension of online PA, STRCF can adaptively balance the tradeoff between aggressive and passive model learning, thus leading to more robust models in the case of large appearance variations. An ADMM algorithm is developed to solve the STRCF model. We perform experiments on three benchmarks, and the results show that STRCF with hand-crafted features is superior than the baseline SRDCF by accuracy and speed. Moreover, STRCF with deep features also performs favorably against state-of-the-art trackers in terms of accuracy and robustness. In future, we will further improve our STRCF by investigating whether the temporal regularizer can be compatible to SAMF+AT [3], Staple+CA [30], and the GMM and continuous convolution in ECO [9].

References

[1] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. Torr. Staple: Complementary learners for real-time tracking. In *CVPR*, 2016. 6, 8

[2] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr. Fully-convolutional siamese networks for object tracking. In *ECCV*, 2016. 6

[3] A. Bibi, M. Mueller, and B. Ghanem. Target response adaptation for correlation filter tracking. In *ECCV*, 2016. 3, 6, 8

[4] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In *CVPR*, 2010. 1, 2, 3

[5] L. Čehovin, A. Leonardis, and M. Kristan. Visual object tracking performance measures revisited. *TIP*, 25(3):1261–1274, 2016. 8

[6] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *JMLR*, 7(3):551–585, 2006. 2, 4

[7] Z. Cui, S. Xiao, J. Feng, and S. Yan. Recurrently target-attending tracking. In *CVPR*, 2016. 3

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 2

[9] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg. Eco: Efficient convolution operators for tracking. In *CVPR*, 2017. 2, 6, 8

[10] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg. Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking. In *CVPR*, 2016. 3, 6, 8

[11] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg. Discriminative scale space tracking. *TPAMI*, PP(99):1–1, 2016. 1, 2, 3, 6, 8

[12] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Convolutional features for correlation filter based visual tracking. In *ICCVW*, 2015. 3, 6, 8

[13] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Learning spatially regularized correlation filters for visual tracking. In *ICCV*, 2015. 1, 2, 3, 5, 6, 8

[14] M. Danelljan, F. S. Khan, M. Felsberg, and J. V. D. Weijer. Adaptive color attributes for real-time visual tracking. In *CVPR*, 2014. 1, 2, 5

[15] M. Danelljan, A. Robinson, F. Shahbaz Khan, and M. Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *ECCV*, 2016. 1, 2, 3, 6, 7

[16] J. Eckstein and D. P. Bertsekas. On the douglas-rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1-3):293–318, 1992. 5

[17] H. K. Galoogahi, A. Fagg, and S. Lucey. Learning background-aware correlation filters for visual tracking. In *ICCV*, 2017. 3, 6

[18] H. K. Galoogahi, T. Sim, and S. Lucey. Correlation filters with limited boundaries. In *CVPR*, 2015. 1, 3

[19] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *TPAMI*, 37(3):583–596, 2015. 1, 2, 3, 6

[20] H. Kiani Galoogahi, A. Fagg, and S. Lucey. Learning background-aware correlation filters for visual tracking. In *ICCV*, 2017. 6, 8

[21] H. Kiani Galoogahi, T. Sim, and S. Lucey. Multi-channel correlation filters. In *ICCV*, 2013. 2, 3

[22] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernández, T. Vojir, G. Hager, G. Nebehay, and R. Pflugfelder. The visual object tracking vot2016 challenge results. In *ECCVW*, 2016. 2, 8

[23] F. Li, Y. Yao, P. Li, D. Zhang, W. Zuo, and M. H. Yang. Integrating boundary and center correlation filters for visual tracking with aspect ratio variation. In *ICCVW*, 2017. 1, 2

[24] Y. Li and J. Zhu. A scale adaptive kernel correlation filter tracker with feature integration. In *ECCVW*, 2014. 1, 2, 6

[25] P. Liang, E. Blasch, and H. Ling. Encoding color information for visual tracking: Algorithms and benchmark. *TIP*, 2015. 2, 8

[26] S. Liu, T. Zhang, X. Cao, and C. Xu. Structural correlation filter for robust visual tracking. In *CVPR*, 2016. 2

[27] T. Liu, G. Wang, and Q. Yang. Real-time part-based visual tracking via adaptive correlation filters. In *CVPR*, 2015. 2, 3

[28] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang. Hierarchical convolutional features for visual tracking. In *ICCV*, 2015. 1, 2, 6

[29] C. Ma, X. Yang, C. Zhang, and M. H. Yang. Long-term correlation tracking. In *CVPR*, 2015. 2

[30] M. Mueller, N. Smith, and B. Ghanem. Context-aware correlation filter tracking. In *CVPR*, 2017. 6, 8

[31] H. Nam, M. Baek, and B. Han. Modeling and propagating cnns in a tree structure for visual tracking. *arXiv preprint arXiv:1608.07242*, 2016. 8

[32] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, 2016. 8

[33] M. Pedersen, K. Baxter, B. Templeton, and D. Theobald. The matrix cookbook. *Technical University of Denmark*, 7:15, 2008. 5

[34] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M. H. Yang. Hedged deep tracking. In *CVPR*, 2016. 2, 6

[35] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 7

[36] J. Valmadre, L. Bertinetto, J. F. Henriques, A. Vedaldi, and P. H. Torr. End-to-end representation learning for correlation filter based tracking. In *CVPR*, 2017. 6, 8

[37] L. Wang, W. Ouyang, X. Wang, and H. Lu. Visual tracking with fully convolutional networks. In *ICCV*, 2015. 6

[38] M. Wang, Y. Liu, and Z. Huang. Large margin object tracking with circulant feature maps. In *CVPR*, 2017. 3

[39] X. Wang, H. Li, Y. Li, F. Shen, and F. Porikli. Robust and real-time deep tracking via multi-scale domain adaptation. In *arXiv:1701.00561*, 2017. 6

[40] Y. Wu, J. Lim, and M.-H. Yang. Object tracking benchmark. *TPAMI*, 37(9):1834–1848, 2015. 2, 5

[41] J. Zhang, S. Ma, and S. Sclaroff. Meem: robust tracking via multiple experts using entropy minimization. In *ECCV*, 2014. 6

[42] T. Zhang, C. Xu, and Y. Ming-Hsuan. Multi-task correlation particle filter for robust object tracking. In *CVPR*, 2017. 2

[43] W. Zuo, X. Wu, L. Lin, L. Zhang, and M.-H. Yang. Learning support correlation filters for visual tracking. *arXiv:1601.06032*, 2016. 1, 5