Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Asymmetric discriminative correlation filters for visual tracking*

Shui-wang LI, Qian-bo JIANG, Qi-jun ZHAO, Li LU‡, Zi-liang FENG

National Key Laboratory of Fundamental Science on Synthetic Vision, College of Computer Science, Sichuan University, Chengdu 610065, China

E-mail: lishuiwang0721@163.com; jqianbo@163.com; qjzhao@scu.edu.cn; luli@scu.edu.cn; fengziliang@scu.edu.cn

Received Sept. 20, 2019; Revision accepted Apr. 12, 2020; Crosschecked Sept. 2, 2020

Abstract: Discriminative correlation filters (DCF) are efficient in visual tracking and have advanced the field significantly. However, the symmetry of correlation (or convolution) operator results in computational problems and does harm to the generalized translation equivariance. The former problem has been approached in many ways, whereas the latter one has not been well recognized. In this paper, we analyze the problems with the symmetry of circular convolution and propose an asymmetric one, which as a generalization of the former has a weak generalized translation equivariance property. With this operator, we propose a tracker called the asymmetric discriminative correlation filter (ADCF), which is more sensitive to translations of targets. Its asymmetry allows the filter and the samples to have different sizes. This flexibility makes the computational complexity of ADCF more controllable in the sense that the number of filter parameters will not grow with the sample size. Moreover, the normal matrix of ADCF is a block matrix with each block being a two-level block Toeplitz matrix. With this well-structured normal matrix, we design an algorithm for multiplying an $N \times N$ two-level block Toeplitz matrix by a vector with time complexity $O(N\log N)$ and space complexity $O(N)$, instead of $O(N^2)$. Unlike DCF-based trackers, introducing spatial or temporal regularization does not increase the essential computational complexity of ADCF. Comparative experiments are performed on a synthetic dataset and four benchmarks, including OTB-2013, OTB-2015, VOT-2016, and Temple-Color, and the results show that our method achieves state-of-the-art visual tracking performance.

Key words: Visual tracking; Discriminative correlation filter (DCF); Asymmetric DCF (ADCF)
https://doi.org/10.1631/FITEE.1900507 **CLC number:** TP391

1 Introduction

Visual tracking is an important and challenging task in the field of computer vision. Discriminative correlation filters (DCFs) have successfully been applied to this problem and rapid advances have been witnessed in recent years (Henriques et al., 2015; Danelljan et al., 2017b; Galoogahi et al., 2017; Sun C et al., 2018; Tang et al., 2018; Kart et al., 2019; Sun YX et al., 2019). DCF can learn very efficiently in the frequency domain via the fast Fourier transform (FFT) by virtue of the circular convolution theorem and the Parseval theorem. By introducing scale estimation (Li Y and Zhu, 2014; Danelljan et al., 2017a), better feature representations (Danelljan et al., 2014; Ma et al., 2015), nonlinear kernel (Henriques et al., 2015), spatial and temporal regularizations (Danelljan et al., 2015; Li F et al., 2018), continuous convolution (Danelljan et al., 2016b), region-of-interest (ROI) based pooling (Sun YX et al., 2019), and so on, DCF-based trackers have been improved greatly and advanced the field significantly. The standard DCF

‡ Corresponding author
* Project supported by the National Natural Science Foundation of China (No. 61773270) and the Key Research and Development Project of Sichuan Province, China (No. 2019YFG0491)
ORCID: Shui-wang LI, https://orcid.org/0000-0002-4587-513X; Li LU, https://orcid.org/0000-0001-7904-8821
© Zhejiang University and Springer-Verlag GmbH Germany, part of Springer Nature 2020

is formulated by circular convolution (We use convolution for mathematical convenience, though correlation can equivalently be used) and target positions are represented by filter responses. The translational equivariance of the convolution operator guarantees that the filter response faithfully reflects the translations of the target if the sample is just representing the target. Another remarkable property of DCF is that its normal matrix is a block matrix with each block being a circulant matrix and becomes blockwise diagonal in the Fourier domain (In the special case of one block, it is just a circulant matrix and becomes diagonal under discrete Fourier transform (DFT)), which makes it very successful in real-time visual tracking since FFT is efficient and the original optimization problem is simplified. However, circular convolution is a symmetry operator, in the sense that two finite discrete signals being convoluted are extended with the same period and they can exchange their positions without impact on the response. This symmetry of the circular convolution operator incurs inconvenience and problems for tracking applications.

For one thing, the symmetry requires the filter size equal that of the sample. This degrades the scalability of the model, since the number of filter parameters and thus the model complexity grows nonlinearly with the size of the sample. It is not wise to just control sample sizes, because small samples may not contain the target if the translation of the target is large. As the default settings of the ratio of the sample size to the target size (Table 1) show, there is a growing trend towards larger sizes, disregarding scaling; even so, they seem insufficiently large in some cases. Fig. 1 shows two examples in which the competing trackers fail when the translations of the targets between two neighboring frames are relatively large, but the proposed tracker succeeds since we use a larger ratio. Therefore, we need a technique that allows the sample size (ratio) to increase without adding much model complexity.

Another issue is that the translation equivariance of symmetric convolution is defined with respect to the whole sample. If the ratio is greater than one, then the background is contained in the sample and the response will reflect not the translation of the target but the mixed effect of the target and the background. To clarify this problem, we need a new definition that can describe the variation of the response with respect to only the target rather than to the whole sample, whereby we can study which operator has this property and whether the tracker constructed with it is faithful to the translations of targets.

Table 1 The ratio of sample size A_f to target size A_t of six state-of-the-art trackers

Tracker	Reference	$A_\mathrm{f}/A_\mathrm{t}$
DSST	Danelljan et al. (2017a)	2.0
KCF	Henriques et al. (2015)	2.5
SRDCF	Danelljan et al. (2015)	4.0
ECO	Danelljan et al. (2017b)	4.0
CCOT	Danelljan et al. (2016b)	5.0
STRCF	Li F et al. (2018)	5.0

In this paper, we give the definition of the (weak) generalized translation equivariance property (see Fig. 2a for an intuitive illustration) of a convolution operator and propose an asymmetric convolution operator by which the proposed tracker asymmetric discriminative correlation filter (ADCF) is constructed. The asymmetry reflects that the two operands (sample and filter) are not in an equal position any more. The filter size is definitely required to be smaller than or equal to that of the sample. When they are equal, the convolution operator reduces to a symmetric one; thus, our proposed convolution operator generalizes the symmetric case. Fig. 3 shows the comparison of tracking frameworks between ADCF and DCF. A concrete definition will be given in Section 3.

As mentioned above, the generalized translation equivariance of a generic convolution operator is a desired property for visual tracking, since it describes the faithfulness of the operator to translations of only the target rather than the whole sample. Unfortunately, neither the circulant convolution nor the proposed one has this property, but we can prove that the proposed operator, under a reasonable assumption, has a weak generalized translation equivariance property and also provides information about the translations of the target. Because of its asymmetry property, the proposed tracker ADCF allows different sizes of filters and samples. Hence, one can train the tracker with small samples and apply it to large samples when necessary; moreover, the number of filter parameters does not grow with the sample size, which makes the model complexity of ADCF controllable in terms of parameter optimization.

Fig. 1 Qualitative comparison of our approach with state-of-the-art trackers on two sequences with large translations of objects. Note that all the competing trackers fail on the sequences

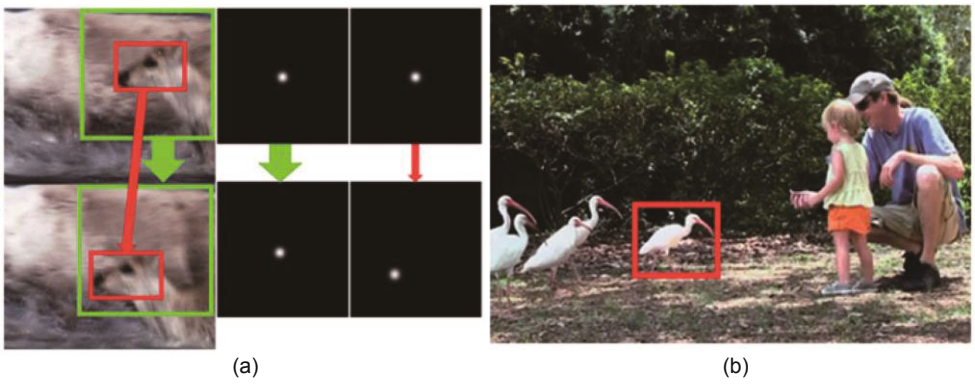

Fig. 2 Illustration of generalized translation equivariance (a) and inaccuracy of rectangular representation of a target (b). The green and red rectangles indicate the sample and target, respectively. In the generalized definition the response reflects the translation of the target instead of the whole sample, which proves its superiority for visual tracking. References to color refer to the online version of this figure

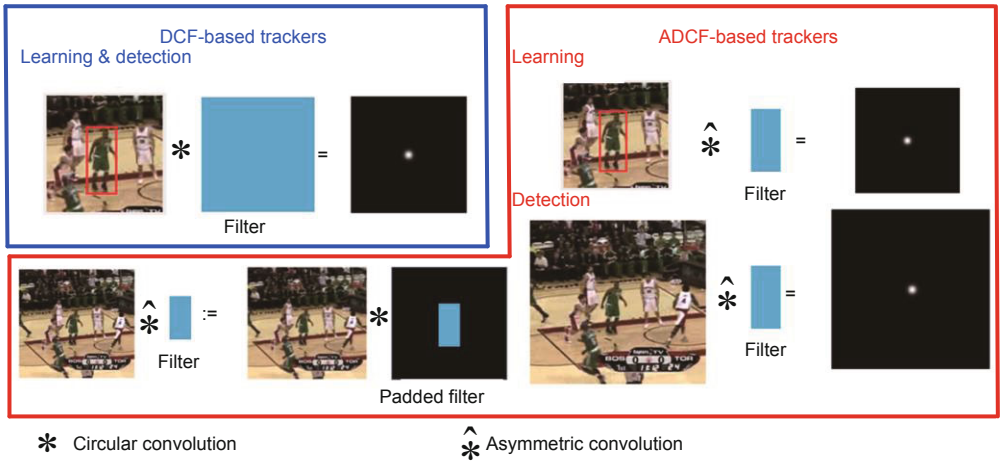

Fig. 3 Comparison of tracking frameworks between DCF and the proposed ADCF. Note that ADCF allows different learning and detection sample sizes

We prove that the derived normal matrix of ADCF is a block matrix with each block being a two-level block Toeplitz matrix, which generalizes the case in DCF where each block is a circulant matrix. However, a Toeplitz matrix dose not become diagonal in the Fourier domain as a circulant matrix does. It is helpless to solve ADCF in the Fourier domain. Therefore, in contrast to DCF, the proposed ADCF is solved in the spatial domain. However, the fact that a Toeplitz matrix can be embedded into a circulant matrix can be used to design fast algorithms for Toeplitz matrix-vector product (Lee, 1986), by which we can therefore use FFT to evaluate the matrix-vector product in iteratively solving, with the conjugate gradient (CG) method, the normal equation of ADCF. Opposite to the case of DCF where the diagonal structure of a normal matrix in the Fourier domain would change when spatial regularization is added, the structure of the Toeplitz matrix does not change in ADCF with spatial and temporal regularization being added, as explained in Section 3. Consequently, whether with spatial and temporal regularizations or not, the computational complexity of ADCF would be essentially the same, which is not true for DCF.

It is worth noting the similarities and differences between the proposed ADCF and the weighted DCF (WDCF), as formulated in Lukezic et al. (2017) and Sun C et al. (2018), with a spatial reliability map taken into account. Both DCF and ADCF can be seen as a special WDCF with a specific reliability map, i.e., a constant function for DCF and a rectangle function for ADCF. However, parameter optimization is much harder in WDCF since the convolution and the Hardamard product (point product) are associated so that convolution is unavoidable in either the spatial or Fourier domain. Only in DCF and ADCF, do good structures of the normal matrix and therefore better computational properties appear.

To evaluate our approach, experiments for comparison are performed on four benchmarks, i.e., OTB-2013 (Wu et al., 2013), OTB-2015 (Wu et al., 2015), VOT-2016 (Kristan et al., 2016), and Temple-Color (Liang et al., 2015). Results show that our algorithm is comparable to the baseline algorithms, while its complexity is more controllable. The contributions of this paper can be summarized as follows:

1. We analyze the problems with the symmetric circular convolution operator and propose an asymmetric one. It satisfies a weak generalized translation equivariance under certain assumptions and allows the filter and the samples to have different sizes. With the proposed operator, we construct the tracker ADCF. The weak generalized translation equivariance guarantees the faithfulness of the tracker to translations of the target rather than of the whole sample, and the flexibility in size makes the model complexity of ADCF more controllable.

2. The proposed ADCF generalizes DCF. We show that the normal matrix of ADCF is a block matrix with each block being a two-level block Toeplitz matrix that generalizes the circulant matrix case of DCF. This fact is very useful for training, and enables us to design an algorithm for multiplying an $N \times N$ two-level block Toeplitz matrix by a vector with time complexity $O(N\log N)$ and space complexity $O(N)$.

3. We explore spatial and temporal regularizations to boost the performance of ADCF and demonstrate the superiority of our proposed method on four benchmarks, i.e., OTB-2013 (Wu et al., 2013), OTB-2015 (Wu et al., 2015), VOT-2016 (Kristan et al., 2016), and Temple-Color (Liang et al., 2015).

2 Related work

This section provides a brief survey on DCF-based trackers. DCF exploits the properties of circular correlation for efficient computation. This has had great success in visual tracking. Using DCF for adaptive tracking was started with MOSSE by Bolme et al. (2010). Henriques et al. (2015) improved the tracking performance by extending the correlation filter to multi-channel inputs and using kernel tricks. Danelljan et al. (2017a) developed new correlation filters to detect the scale changes of targets. To reduce the boundary effects caused by the periodic assumption of DCF, they also proposed spatially regularized DCF (SRDCF) for tracking (Danelljan et al., 2015). Lukezic et al. (2017) introduced the concept of channels and spatial reliabilities to DCF tracking in order to overcome the limitations related to the rectangular shape assumption. Mueller et al. (2017) proposed context-aware DCF to incorporate global context to deal with fast motion, occlusion, or background clutter on the ground that conventional DCF trackers are learned locally.

By introducing temporal regularization to SRDCF, Li F et al. (2018) proposed spatial-temporal regularized correlation filters (STRCF) to achieve more robust appearance models. As for feature representation, the DCF approaches were initially restricted to a single-channel feature map and later extended to multi-channel feature maps (Galoogahi et al., 2013; Danelljan et al., 2014, 2016b), such as histogram of oriented gradient (HOG) (Dalal and Triggs, 2005), color names (Danelljan et al., 2014), and deep convolutional neural network (CNN) features (Ma et al., 2015; Danelljan et al., 2016b; Qi et al., 2016; Sun C et al., 2018; Tang et al., 2018; Kart et al., 2019; Sun YX et al., 2019).

2.1 Discriminative correlation filters

Before the detailed discussion of our approach, we first revisit the details of the conventional DCF. DCF aims to learn a multi-channel convolution filter f from a set of training samples $\{(x_k, y_k)\}_{k=1}^{t}$. All samples are assumed to have the same spatial size $M \times N$. Each training sample x_k, extracted from an image region, consists of d feature maps with x_k^l denoting the l^{th} one. y_k, obtained from a predefined 2D Gaussian shaped label, is the target response map corresponding to x_k, which has the same size as x_k^l. f consists of d 2D convolution filters, each of which will be applied to the corresponding feature channel. The convolution response of f on a sample x is given by

$$R(x; f) = \sum_{l=1}^{d} x^l * f^l, \tag{1}$$

where $*$ denotes the circular convolution operator. DCF is formulated by the following objective:

$$\arg\min_{f} \quad \sum_{k=1}^{t} \alpha_k \|R(x; f) - y_k\|^2 + \lambda \sum_{l=1}^{d} \|f^l\|^2, \tag{2}$$

where $\alpha_k \geq 0$ determines the impact of each training smaple and $\lambda \geq 0$ is the weight of the regularization term (Danelljan et al., 2015). Expression (2) defines 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, and in the Fourier domain the filter $\hat{f} = \{\mathcal{F}\{f^l\}\}_{l=1}^{d}$ can be obtained by solving MN $d \times d$ complex linear equation systems (Danelljan et al., 2015; Galoogahi et al., 2017).

2.2 Spatial regularization

Although the circular convolution in Eq. (1) can be efficiently computed by DFT, it comes at a cost. The operation of circular convolution corresponds to applying the filter f^l, in a sliding window fashion, to the periodic extension of the sample x^l. This periodic extension creates synthetic examples not truly representative of the shifted ones and thus causes unwanted boundary effects which have a dramatic impact on the detection and tracking performance (Danelljan et al., 2015; Galoogahi et al., 2017). To deal with this problem, Danelljan et al. (2015) replaced the regularization term in expression (2) by a more general Tikhonov regularization and proposed SRDCF, which is formulated as follows:

$$\arg\min_{f} \quad \sum_{k=1}^{t} \alpha_k \|R(x; f) - y_k\|^2 + \sum_{l=1}^{d} \|w \circ f^l\|^2, \tag{3}$$

where w is the spatial regularization weight and $\circ$ denotes the Hadamard product. The Gauss-Seidel method is exploited to iteratively update the filter f (Danelljan et al., 2015).

2.3 Temporal regularization

Although SRDCF is effective in suppressing the boundary effects, it also increases the computational complexity since Eq. (3) results in a $dMN \times dMN$ large linear equation system without a well-structured normal matrix, which is much more computationally expensive than Eq. (2). On the other hand, SRDCF learns its filter with multiple samples from historical tracking and assigns larger weights to the recent ones. Consequently, it may suffer from overfitting to the recent inaccurate or noisy samples, especially in the case of occlusion. To overcome these problems, motivated by the online passive-aggressive (PA) algorithm, Li F et al. (2018) introduced a temporal regularization term and relaxed the multi-sample formulation in SRDCF to the one with a single sample, leading to the following STRCF model:

$$\arg\min_{f} \quad \|R(x_t; f) - y\|^2 + \sum_{l=1}^{d} \|w \circ f^l\|^2 + \mu \|f - f_{t-1}\|^2, \tag{4}$$

where f_{t-1} denotes the filter learned in the $(t-1)^{\text{th}}$ frame, and μ is the regularization parameter. The alternating direction method of multipliers (ADMM)

algorithm was used to solve problem (4) in Li F et al. (2018).

3 Asymmetric discriminative correlation filters

In this section, the formulation and optimization algorithm of the proposed ADCF will be given. Spatial and temporal regularizations are introduced to improve ADCF. Because of space limitations, most proofs and strict definitions are given in the appendix.

3.1 Formulation of asymmetric discriminative correlation filters

ADCF also aims to learn a multi-channel convolution filter f from a set of training samples $\{(x_k, y_k)\}_{k=1}^{t}$ with the same spatial size $M \times N$, but the filter may have a smaller size $m \times n$ because of the asymmetry we impose on the proposed operator of which we expect the generalized translation equivariance. For such an operator, we therefore need to define how to "convolute" a filter with a possibly larger sample. Our definition is similar to the convolution used by the CNN but with special boundary processing, so that it is equivalent to the circular convolution of the sample with a new filter. In this way, the convolution theorem can still apply when evaluating this "convolution." This is not generally correct in CNN but crucial for fast detection. This new filter can be constructed by just padding the original one with zeros as we will explain. The way to pad with zeros is not unique. Hence, without loss of generality, we choose a simple implementation.

Assuming for simplicity that the size differences $M - m$ and $N - n$ are even numbers, the proposed asymmetric circular convolution is defined as follows (for a more strict definition see Appendix A):

$$x^l \hat{*} f^l = x * P(f^l; x^l), \tag{5}$$

where $\hat{*}$ and $*$ denote the proposed convolution and the circular convolution respectively, P denotes the symmetric padding operation such that $P(f^l; x^l)$, symmetrically padded from f^l with zeros, has the same size as x^l. The convolution response of filter f to a sample x is now given by

$$\hat{R}(x; f) = \sum_{l=1}^{d} x^l * P(f^l; x^l). \tag{6}$$

The translation equivariance of circular convolution is defined by the property that convolution commutes with translations, i.e., $\tau_{\Delta}(x^l * f^l) = \tau_{\Delta}(x^l) * f^l$, where $\tau_{\Delta}(x^l)$ represents the translation of x^l by Δ. By generalized translation equivariance, we mean

$$\tau_{\Delta}(x^l \circledast f^l) = T_{\Delta}(t^l; x^l) \circledast f^l, \tag{7}$$

where $T_{\Delta}(t^l; x^l)$ represents the translation of the target $t^l \subset x^l$ by Δ, and $\circledast$ can be any well-defined binary operator. When samples are larger than the target, we would not have $\tau_{\Delta}(x^l) = T_{\Delta}(t^l; x^l)$, so the circular convolution operator $*$ does not satisfy Eq. (7). Although the proposed $\hat{*}$ does not satisfy Eq. (7) either, we can prove that under a reasonable assumption it satisfies the weak generalized translation equivariance defined as follows:

$$\tau_{\Delta} B(x^l \circledast f^l) = B(T_{\Delta}(t^l; x^l) \circledast f^l), \tag{8}$$

where $B(x)$ is a binary map masking the maximum value of x as 1 and others 0. If filter f^l and target t^l are of the same size, the assumption we impose just states that $t^l \hat{\cdot} f^l > b^l \hat{\cdot} f^l$, where $\hat{\cdot}$ is the flipped inner product and b^l is any subsample of x^l that contains background and is of the same size as f^l. This assumption is reasonable if filter f is sufficiently discriminative. This weak property says that there is a peak in the filter response that faithfully reflects the translations of the target. We will not give the proof here and leave the details to the appendix. ADCF is formulated by the following objective:

$$\arg\min_{f} \quad \sum_{k=1}^{t} \alpha_k \|\hat{R}(x_k; f) - y_k\|^2 + \lambda \sum_{l=1}^{d} \|f^l\|^2. \tag{9}$$

In contrast to DCF, problem (9) cannot be simply solved by transformation into the Fourier domain. In the next subsection we will derive the normal equation and give a fast algorithm for its solution.

3.2 Optimization algorithm

Let $g_{i,j} = u_i^{\mathrm{T}} v_j$, where u_i and v_j are the i^{th} and j^{th} canonical basis vectors for the M- and N-dimensional spaces respectively, that is, $u_i = [0, ..., 0, 1, 0, ..., 0]_M, v_j = [0, ..., 0, 1, 0, ..., 0]_N$. Denoting the value of $P(f^l; x^l)$ at coordinates (i, j) (The sample and filter are coordinated in a unified way, such as the one used in Matlab) by $P_{i,j}^l$,

$P(f^l;x^l)$ can be represented by

$$P(f^l;x^l)=\sum_{i=1}^{M}\sum_{j=1}^{N}P_{i,j}^l g_{i,j}. \tag{10}$$

Since $P(f^l;x^l)$ is a padded f^l, denoting the set of coordinates of f^l in the coordinate system of $P(f^l;x^l)$ by D, we have

$$P(f^l;x^l)=\sum_{(i,j)\in D}P_{i,j}^l g_{i,j}=\sum_{q(i',j')=(i,j)\in D}f_{i',j'}^l g_{i,j}, \tag{11}$$

where q is a coordinate mapping function that maps the coordinates of f^l in the coordinate system of itself into that of $P(f^l;x^l)$. Then, we have

$$\begin{aligned}\hat{R}(x_k;f)&=\sum_{l=1}^{d}\sum_{q(i',j')=(i,j)\in D}f_{i',j'}^l x_k^l * g_{i,j}\\&=\sum_{l=1}^{d}\sum_{q(i',j')=(i,j)\in D}f_{i',j'}^l T_{i,j}(x_k^l),\end{aligned} \tag{12}$$

where $T_{i,j}(x^l)$ denotes the circular translation (shift) of x^l by i and j in vertical and horizontal directions, respectively. Therefore, problem (9) is equivalent to

$$\arg\min_f \sum_{k=1}^{t}\alpha_k\|\sum_{l=1}^{d}A_k^l\vec{f}^l-\vec{y}_k\|^2+\lambda\sum_{l=1}^{d}\|\vec{f}^l\|^2, \tag{13}$$

where $\vec{f}^l$ denotes the vectorized f^l and $A_k^l=[\vec{T}_{i_1,j_1}(x_k^l)\cdots\vec{T}_{i_m,j_n}(x_k^l)]$, $(i_x,j_y)=q(x,y)\in D$. To further simplify the notations, we define concatenations $A_k=[A_k^1\cdots A_k^d]$ and $\vec{f}=[(\vec{f}^1)^{\mathrm{T}}\cdots(\vec{f}^d)^{\mathrm{T}}]^{\mathrm{T}}$. Problem (13) then becomes

$$\arg\min_f \sum_{k=1}^{t}\alpha_k\|A_k\vec{f}-\vec{y}_k\|^2+\lambda\|\vec{f}\|^2. \tag{14}$$

Finally, expression (14) is minimized by solving the normal equation $B_t\vec{f}=\vec{b}_t$, where

$$B_t=\sum_{k=1}^{t}\alpha_k A_k^{\mathrm{T}}A_k+\lambda^2 I,\quad \vec{b}_t=\sum_{k=1}^{t}\alpha_k A_k^{\mathrm{T}}\vec{y}_k. \tag{15}$$

Here, Eq. (15) defines a real $dmn\times dmn$ linear system of equations, which is $(m/M)^2(n/N)^2$ times that of SRDCF. In spite of this, A_k is an $MN\times dmn$ matrix and is not sparse in general cases. It is computationally costly to evaluate $A_k^{\mathrm{T}}A_k$ and even $A_k^{\mathrm{T}}\vec{y}_k$. Fortunately, as we will prove in Appendix B, $A_k^{\mathrm{T}}A_k$ is a block matrix with each block $(A_k^{\mathrm{T}}A_k)_{i,j}=(A_k^i)^{\mathrm{T}}A_k^j$ being a two-level block Toeplitz matrix. We find that $(A_k^i)^{\mathrm{T}}A_k^j$ is a submatrix of the cross-correlation function (or autocorrelation function if $i=j$) of the periodically extended x_k^i and x_k^j, while $(A_k^l)^{\mathrm{T}}\vec{y}_k$ is a subvector of the circular convolution of x_k^l and $\vec{y}_k$, both of which can be evaluated by FFT. Therefore, we can compute $(A_k^i)^{\mathrm{T}}A_k^j$ or $(A_k^l)^{\mathrm{T}}\vec{y}_k$ much faster if for the former $m^2n^2>>2\log(MN)$ or for the latter $mn>>2\log(MN)$. The fact that $(A_k^l)^{\mathrm{T}}A_k^l$ is an $mn\times mn$ two-level block Toeplitz matrix allows us to store it with spatial complexity $O(mn)$ and multiply it by a vector with time complexity $O(mn\log(mn))$ (Lee, 1986), which is very useful when we have to evaluate the matrix-vector product in iteratively solving the normal equation of ADCF with the CG method.

Despite the fact that the properties of the normal matrix are favorable for parameter optimization, the number of submatrices $(A_k^i)^{\mathrm{T}}A_k^j$ $(1\le i,j\le d)$ increases quadratically with the increase of the number of feature channels. To further reduce the time cost, we impose the condition that the inner product $<T_{i,j}(x_k^l),T_{i',j'}(x_k^{l'})>$ equals zero if $l\neq l'$, so that $A_k^{\mathrm{T}}A_k$ is block-diagonal and therefore the number of submatrices $(A_k^i)^{\mathrm{T}}A_k^j$ equals that of feature channels. This simplification is in fact equivalent to learning, separately, one filter for each feature channel, not as problemmatic as it seems. Under this simplification, problem (13) reduces to

$$\arg\min_f \sum_{l=1}^{d}\Big(\sum_{k=1}^{t}\alpha_k\|A_k^l\vec{f}^l-\vec{y}_k\|^2+\lambda\|\vec{f}^l\|^2\Big), \tag{16}$$

which is equivalent to minimizing d separate subobjectives, defined by the terms inside the parentheses, and finally equivalent to solving the following normal equations:

$$B_t^l=\sum_{k=1}^{t}\alpha_k(A_k^l)^{\mathrm{T}}A_k^l+\lambda^2 I,\quad \vec{b}_t^l=\sum_{k=1}^{t}\alpha_k(A_k^l)^{\mathrm{T}}\vec{y}_k. \tag{17}$$

3.3 Spatial and temporal regularizations

We introduce spatial regularization, in contrast to SRDCF, by the motivation that the rectangular representation of the target is not always accurate (see Fig. 2b for an example), which has been studied, for instance, in Lukezic et al. (2017) and Sun C et al. (2018). These studies used a spatial reliability map to suppress the possibly wrong values of a learned filter. However, in their formulations the Hadamard product, between the filter and the spatial reliability map, and the convolution are associated, which

complicates parameter optimization so much that they had to use very complex optimization algorithms for learning.

Since the inaccuracy of the rectangular representation is caused mainly by its inaccurate boundary, we use, for simplicity, a bowl shaped function to regulate the filter in the same way as SRDCF. Although the inaccurate bowl shape may punish the correct filter values on the boundary, it gives special importance to the center area where the target overlaps most often. Despite the fact that the same trick is used for two different motivations, in SRDCF and here, we still call it spatial regularization.

The introduction of temporal regularization prohibits STRCF from overfitting too fast to recent inaccurate samples, which, however, also stops it from updating fast enough if it is necessary, especially when the target is fast changing. Moreover, to exploit the circulant matrix structure for relieving the computational burden, STRCF is formulated with one sample (Li F et al., 2018), which makes the model focus on local time and it tends to forget historical information. As a result, STRCF would fail when the target is occluded for a long time or the input is polluted by much noise. Therefore, we prefer the formulation with multiple samples and use an adaptive weight for temporal regularization so that we can increase or decrease temporal regularization when needed. With spatial and temporal regularizations, problem (9) turns into

$$\begin{aligned}\arg\min_{f} \quad & \sum_{k=1}^{t}\alpha_k\|\hat{R}(x_k;f)-y_k\|^2+\lambda\sum_{l=1}^{d}\|w\circ f^l\|^2\\ & +\lambda_t\|f-f_{t-\Delta t}\|,\end{aligned} \tag{18}$$

where $1\le\Delta t\le t-1$ is a predefined frame difference, λ is a predefined weight, and λ_t is adapted by the following formula:

$$\begin{cases}\lambda_t=c_0+\mu_1\mu_2,\\ \mu_1=\mathrm{e}^{a_1<x_t,x_{t-\Delta t_1}>+b_1},\\ \mu_2=1/(1+\mathrm{e}^{a_2<x_t,x_{t-\Delta t_2}>+b_2}),\end{cases} \tag{19}$$

where c_0 is a constant, Δt_1 and Δt_2 ($\Delta t_2<\Delta t_1$) are two predefined frame gaps, and a_1 and b_1 are predefined so that μ_1 is a monotonically increasing function which will increase temporal regularization when the target is experiencing fast change or being occluded. Once the target is occluded, the detection samples tend to change very slowly or do not change at all, when the temporal regularization should be strong to slow down filter updating. μ_2 is a soft step function. a_2 and b_2 are predefined to distinguish when the target is being occluded from the one experiencing fast change, so that the temporal regularization is strong in the former case and weak in the latter. The incorporation of spatial and temporal regularizations does not result in a more complex optimization problem. We just need to adjust the normal Eq. (15) to

$$\begin{cases}B_t=\sum_{k=1}^{t}\alpha_k A_k^{\mathrm{T}}A_k+\lambda I_d\otimes D\,(w\vec{\circ}w)+\lambda_t I,\\ \vec{b}_t=\sum_{k=1}^{t}\alpha_k A_k^{\mathrm{T}}\vec{y}_k+\lambda_t\vec{f}_{t-\Delta t},\end{cases} \tag{20}$$

where I_d and $\otimes$ denote the $d\times d$ identity matrix and Kronecker product respectively, and $D(w\,\vec{\circ}\,w)$ denotes the diagonal matrix created by the vectorized Hadamard product $w\circ w$. Eq. (17) can be adjusted analogously.

4 Our tracking framework

Compared with existing DCF-based trackers, we have more flexibility in the sizes of training and detection samples. In our implementation, training samples have smaller size than detection samples, to reduce the training time. The sizes of these samples are determined in the first frame according to the target size and image size, and will not change in the tracking process. On one hand, this simplifies our implementation; on the other hand, frequent changes of the sample size may introduce many untrue feature variations. The following discussion of training and detection is for the simplified model only, but is also applicable to the full model.

4.1 Training

Like most DCF-based trackers, our model is updated iteratively in the training stage after each detection. We use $B_t\vec{f}=\vec{b}_t$ to denote the normal equation to be solved at frame t. In the first frame, the model is set to $B_1^l=(A_1^l)^{\mathrm{T}}A_1^l+\lambda D(w\,\vec{\circ}\,w)$ and $\vec{b}_1^l=(A_1^l)^{\mathrm{T}}\vec{y}_1$, where A_t and y_t are the concatenation and vectorized response respectively as defined in Section 3.2, and the spatial regularization term $D'=\lambda D(w\,\vec{\circ}\,w)$ is computed once for all. Using $A_t'^l$

and $\vec{b'}_t^l$ to record the weighted sum of $(A_t^l)^{\mathrm{T}}A_t^l$ and $(A_t^l)^{\mathrm{T}}\vec{y}_t$, then the model is updated with a learning rate $\gamma \geq 0$ as follows:

$$\begin{cases} A'^l_t = (1-\gamma)A'^l_{t-1} + \gamma(A_t^l)^{\mathrm{T}}A_t^l,\ \vec{b'}_t^l = \vec{b'}_{t-1}^l + (A_t^l)^{\mathrm{T}}\vec{y}_t, \\ B_t^l = A'^l_t + D' + \lambda_t I,\ \vec{b}_t^l = \vec{b'}_t^l + \lambda_t \vec{f}^l_{t-\Delta t}, \end{cases} \tag{21}$$

which corresponds to using exponentially decaying weights α_k in Eq. (20). After the model is updated, we use the CG method with a fixed number of iterations (N_{CG}) to obtain the updated filter coefficients. This is because the CG method produces the exact solution, if existing, after a finite number of iterations, disregarding the round-off error; more importantly, the CG method is good for exploiting the properties of the normal matrix we have shown. In the first frame, the filter coefficients are set to 1 for the initial iteration, while in frame t they are set to the updated filter coefficients in the previous frame.

4.2 Detection

The location of the target in frame t is estimated by applying the filter that has been updated in the $(t-1)^{\mathrm{th}}$ frame to the detection sample extracted from the current frame at the last position. As in Danelljan et al. (2016a), we learn a scale filter online and apply it at multiple resolutions to estimate the scale changes of the target size. The total computational complexity of our tracker is $O((d+2)MN\log(MN)+dN_{\mathrm{CG}}mn\log(mn)+MN)$, excluding feature extraction and scale estimation. Since, theoretically, $N_{\mathrm{CG}} \leq mn$ by the convergence properties of the CG method, if the area of the target mn is taken as fixed, as in our realization, and the samples for training or detection are large enough, then the main computational complexity of ADCF is $O((d+2)MN\log(MN)+MN)$. That is to say, the time cost for parameter optimization is negligible in this situation. Despite this, the Toeplitz matrix-vector product shows significant computational advantage over direct matrix-vector product only when $\log(mn) \ll mn/4$, but in our implementation mn is bounded by 1600 to reduce the number of filter parameters. Therefore, in our implementation, the computational advantage of ADCF is not significant. However, we will show experimentally the computational advantage of Toeplitz matrix-vector product over direct matrix-vector product with respect to mn.

5 Experimental results

5.1 Implementation details

The samples for training and detection are all squares and set according to a predefined two-valued step function, defined by

$$\begin{cases} (y_{\mathrm{t}}, y_{\mathrm{d}}) = (h_{\mathrm{i}} \wedge w_{\mathrm{i}})(\beta_0, \gamma_0) I(0 < p(h_{\mathrm{t}}, w_{\mathrm{t}}) < p_0) \\ \quad + \sum_{i=1}^{n} \sqrt{h_{\mathrm{t}} w_{\mathrm{t}}}(\beta_i, \gamma_i) I(p_{i-1} \leq p(h_{\mathrm{t}}, w_{\mathrm{t}}) < p_i), \\ p(h,w) = \dfrac{h}{w} \vee \dfrac{w}{h}, \end{cases} \tag{22}$$

where $p_n = \infty$, y_{t} and y_{d} denote the edge lengths of the training and detection samples respectively, $(h_{\mathrm{t}}, w_{\mathrm{t}})$ and $(h_{\mathrm{i}}, w_{\mathrm{i}})$ represent the sizes of the target and the image respectively, $\wedge$ and $\vee$ return the smaller and greater numbers of two elements respectively, and I denotes the indicator function. The constants n and $\{(p_i, \beta_i, \gamma_i)\}_1^n$ are successively set to 4, $(1.5, 0.5, 1), (2, 2, 3), (3, 3, 4), (\infty, 3, 3)$. If the target size is small, the relative rounding error of the target position will be significant. Moreover, the size scales down after feature extraction, especially when deep features are used. Since we do not interpolate the response maps to deal with this problem as SRDCF does, we use only hand-crafted features HOG (Dalal and Triggs, 2005) and color names (CN) (Danelljan et al., 2014), with a cell size of 4×4, to evaluate our approach, and resize the image by the following strategy if the target is too small. If $h_{\mathrm{i}} \wedge w_{\mathrm{i}} < 10, 20, 30$, the smaller edge length of the target will be resized to $16, 24, 32$ correspondingly, and the other edge scales proportionally. The spatial regularization weight is set to $\lambda = 120$, and the weight function is constructed by

$$w(u,v) = (1 - \vec{g}(u;\rho,h))(1 - \vec{g}(v;\rho,w))^{\mathrm{T}}, \tag{23}$$

where $\rho = 1.5$ and g is a Gaussian window defined by $g(u;\rho,n) = \exp\left\{-\frac{1}{2}[\rho\frac{u}{(n-1)/2}]^2\right\}$. For temporal regularization, $a_1 = -20\ln 10, b_1 = 3 \times 10^{14}, a_2 = -400, b_2 = 270, c = 500$ and the frame differences are set to $\Delta t = \Delta t_1 = 50, \Delta t_2 = 5$. The learning rate and the number of iterations of CG are $\gamma = 0.02$ and $N_{\mathrm{CG}} = 50$, respectively. Note that there may be small changes in different settings to adapt to different datasets for better performance.

5.2 Baseline trackers, datasets, and evaluation measures

We use three DCF-based trackers KCF (Henriques et al., 2015), SRDCF (Danelljan et al., 2015), and STRCF (Li F et al., 2018) as baselines. Table 2 summarizes these trackers. We also provide a comparison with 13 state-of-the-art methods: MKCF (Tang et al., 2018), SiamTri (Dong and Shen, 2018), ACT (Chen et al., 2018), DAT (Pu et al., 2018), DSST (Danelljan et al., 2017a), CF2 (Ma et al., 2015), SAMF (Li Y and Zhu, 2014), Staple (Bertinetto et al., 2016), MEEM (Zhang et al., 2014), SRDCF (Danelljan et al., 2015), SRDCFdecon (Danelljan et al., 2016a), BACF (Galoogahi et al., 2017), and ECO-HC (Danelljan et al., 2017b).

We perform comparative experiments on four benchmarks. The OTB-2013 dataset (Wu et al., 2013) contains 50 image sequences with 11 different attributes. The OTB-2015 dataset (Wu et al., 2015) is an extension of the OTB-2013 dataset, and contains 50 more video sequences. The VOT-2016 benchmark (Kristan et al., 2016) contains 60 image sequences with 5 challenges. These datasets have been frequently used (Danelljan et al., 2015; Henriques et al., 2015; Nam and Han, 2016; Choi et al., 2017; Li B et al., 2018; Wang et al., 2019) as they include a large variety of environments to evaluate the general tracking performance. The Temple-Color dataset (Liang et al., 2015) consists of 128 color sequences. As performance measures for OTB-2013 and OTB-2015 benchmarks, the One Pass Evaluation (OPE) protocol (Wu et al., 2013, 2015) with two measures, precision and success, is widely used. For ranking trackers, the common threshold of 20 pixels is used for the former and the area under the curve (AUC) is used for the latter (Wu et al., 2013, 2015). The success measure is also used in the Temple-Color dataset (Liang et al., 2015). As for the VOT-2016 benchmark, accuracy, robustness, and the expected average overlap (EAO) are used instead. Our trackers are implemented with Matlab 2017a, and all experiments are run on a PC equipped with Intel i7 7700 CPU and 16 GB memory.

5.3 Quantitative results

5.3.1 OTB-2013 and OTB-2015 benchmarks

Table 3 shows the results of the mean overlap precision (OP) and the number of frames per second (FPS) on OTB-2013 and OTB-2015. As is shown, the proposed ADCF and SRADCF significantly surpass their counterparts KCF and SRDCF, respectively, by 8.9% and 3.4% in terms of mean OP on OTB-2013. As for FPS, our ADCF and STRACF are inferior to their counterparts, but the proposed SRADCF is 3.87 times faster than its counterpart SRDCF. Note that on average we use larger detection sample sizes.

Table 3 The mean OP and FPS results of state-of-the-art trackers on OTB-2013 and OTB-2015

Tracker	Mean OP (%) (OTB-2013)	Mean OP (%) (OTB-2015)	FPS (OTB-2015)
KCF	45.5	55.2	**211.4**
ADCF	54.4	58.0	37.4
MEEM	54.2	62.2	27.6
SAMF	57.1	67.4	28.5
Staple	61.2	70.9	**94.3**
MKCF	65.6	72.2	**64.7**
SiamTri	67.2	74.7	30.6
SRDCF	66.5	72.7	7.1
SRADCF	69.9	74.0	27.5
SRDCFdecon	68.8	75.3	2.5
ECO-HC	**72.3**	**78.4**	51.7
STRCF	**74.2**	**79.6**	29.9
STRACF	**73.4**	**79.7**	25.4

OP: overlap precision; FPS: number of frames per second.
The best three results are in bold

Fig. 4 shows the precision and success plots on OTB-2013 and OTB-2015. As shown in Figs. 4a and 4b, ADCF outperforms KCF in terms of precision with significant gains of 10.4% and 5.0%, respectively, on the two benchmarks; SRADCF is slightly superior to SRDCF, but STRACF is slightly inferior to STRCF. As shown in Figs. 4c and 4d,

Table 2 Baseline and the proposed trackers

Proposed	Baseline	Feature(s)	Scale	Reg.
ADCF	KCF (Henriques et al., 2015) (TPAMI)	HOG	No	No
SRADCF	SRDCF (Danelljan et al., 2015) (ICCV)	HOG	Yes	S
STRACF	STRCF (Li F et al., 2018) (CVPR)	HOG, CN	Yes	S & T

Reg.: regularization. S: spatial; T: temporal

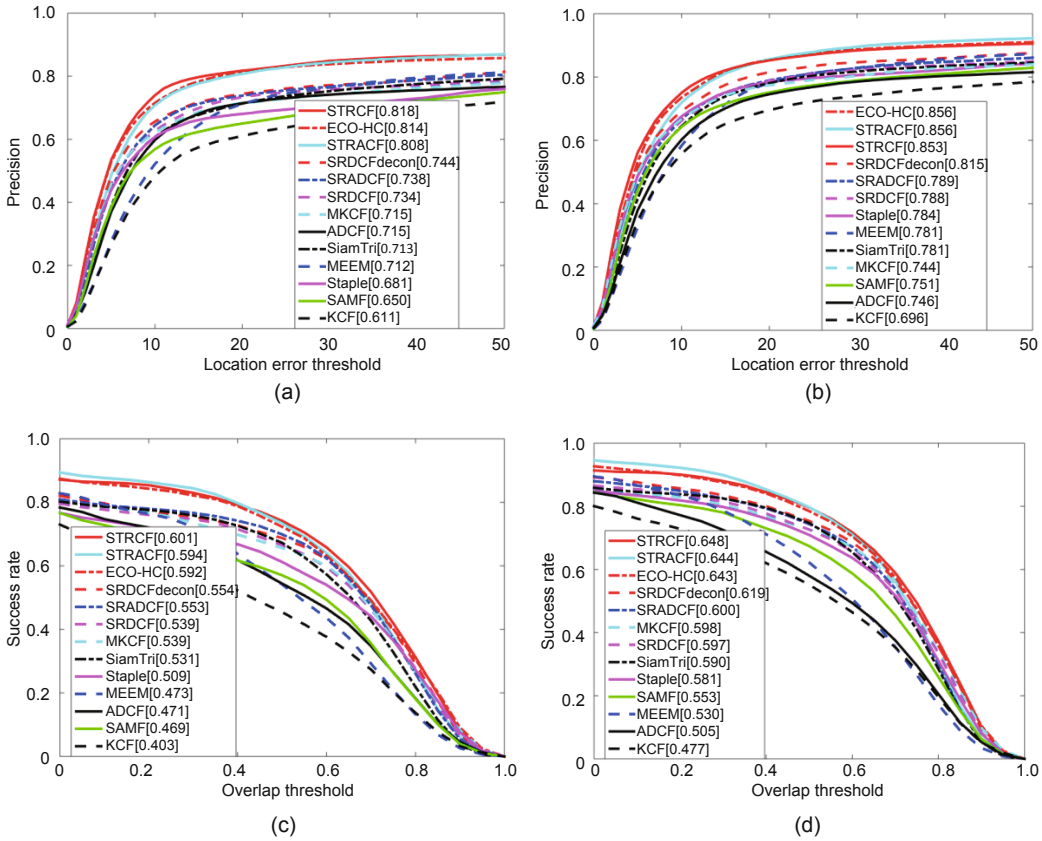

Fig. 4 Precision and success plots: (a) and (b) show the precision plots on OTB-2013 and OTB-2015, respectively; (c) and (d) show the success plots on OTB-2013 and OTB-2015, respectively

ADCF performs better than the counterpart KCF on both benchmarks with gains of 6.8% and 2.8%, respectively, in terms of the success rate. SRADCF slightly surpasses SRDCF on the two benchmarks, but STRACF is slightly behind the counterpart STRCF by gaps of 0.7% and 0.4%, respectively, in terms of the success rate. In fact, the proposed ADCF-based trackers tend to accumulate scale errors in the long run. For one thing, we just adopt a discrete scale estimation strategy and do not use sub-grid detection that interpolates the response map as used by SRDCF, SRDCFdecon, ECO-HC, and STRCF. For another, the filter sizes of the proposed ADCF-based trackers are small relative to their counterparts, so the filters are more sensitive to the noise caused by image scaling. In the future, we will explore how to deal with these problems.

5.3.2 VOT-2016 benchmark

We try to integrate KCF into the VOT-2016 challenge but fail because KCF cannot run through many sequences. The fact that there is no scale estimation may be the cause. Therefore, KCF is excluded in this comparison. The results are reported in Table 4. As shown, the proposed SRADCF is inferior to its counterpart SRDCF but only with a small gap of 0.01 in terms of accuracy. STRCF also outperforms STRACF with small gains, 0.008 and 0.01, respectively, in terms of EAO and accuracy, but STRACF surpasses STRCF in terms of robustness by a gain of 0.04.

5.3.3 Temple-Color benchmark

We perform comparative experiments on the Temple-Color dataset (Liang et al., 2015), which consists of 128 color sequences. We compare the proposed trackers, their counterparts, and some state-of-the-art trackers. Fig. 5 shows the comparison of overlap success plots for different trackers. We note that STRACF is comparable to its counterpart STRCF, and SRADCF and ADCF surpass SRDCF and KCF by 0.7% and 4.4%, respectively.

Table 4 A comparison with state-of-the-art trackers on the VOT-2016 dataset

Tracker	EAO	Accuracy	Robustness
DSST	0.173	0.50	2.58
SAMF	0.166	0.48	2.48
ACT	0.173	0.43	2.37
SRDCF	0.181	**0.52**	2.30
SRADCF	0.189	0.51	2.13
SRDCFdecon	0.193	0.51	2.23
DAT	0.217	0.46	1.72
BACF	0.239	**0.54**	1.67
SiamTri	0.240	0.48	1.85
ECO-HC	**0.279**	**0.52**	**1.28**
STRCF	**0.259**	**0.53**	**1.42**
STRACF	**0.251**	**0.52**	**1.38**

EAO: expected average overlap. The best three results are in bold

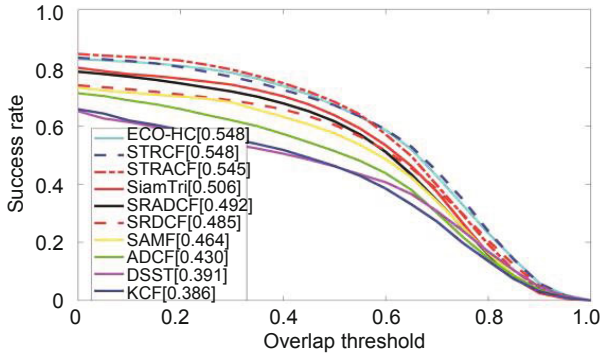

Fig. 5 The overlap success plot of different trackers on Temple-Color. Only 10 trackers are displayed for clarity

5.3.4 Evaluation on a synthetic dataset

As the previous evaluation experiments show, little superiority is observed on the proposed STRACF over its counterpart STRCF. However, the traveling speed of targets in these benchmarks is generally low. The size of the detection sample is basically large enough for these video sequences in all the competing trackers. Moreover, the importance of the weak generalized translation equivariance is not significant in these benchmarks. To show the advantage of the proposed tracker, we need videos in which targets are moving very fast when the size of the detection sample does have large impact on the tracking performance and videos in which the weak generalized translation equivariance does manifest itself. For one thing, to avoid cost and difficulties involved in video collections, and for another to make the experimental conditions controllable, we synthesize a dataset to carry out the evaluation. To get rid of other factors that may make difference on the results, such as appearance and scale of any target, only translations are allowed in each video.

We use a single target to synthesize 100 videos, in which there are 50 videos with a single color as background, denoted by Syn-a, and the other 50 videos with a random noise as background, denoted by Syn-b. See Fig. 6 for examples. In each video, the target is moving with a constant speed but in a random direction, under the condition that the target cannot cross the border of the image and the background is fixed in each video. Both Syn-a and Syn-b are synthesized with 50 speeds ranging from 5 to 250 pixels per frame. Each video has 100 frames with target size 50×50 and image size 500×500. We evaluate each competing tracker with two settings of the ratio $A_\mathrm{f}/A_\mathrm{t}$, the sample size to the target size, specifically $A_\mathrm{f}/A_\mathrm{t} = 5, 10$. Since there is no occlusion or other appearance changes of the target in these videos, we just use the proposed ADCF to evaluate our tracker. The precision plots of all competing trackers on Syn-a with ratios 5 and 10 are shown in Figs. 7a and 7b, respectively, and the precision plot on Syn-b with ratio 10 is shown in Fig. 7c. As can be seen, although Syn-a is a simple tracking task, the precisions of all trackers are below 0.6 when $A_\mathrm{f}/A_\mathrm{t} = 5$. When $A_\mathrm{f}/A_\mathrm{t} = 10$, all precisions are above 0.8. Therefore, 5 is not large enough as a ratio. However, it is the largest value of the ratio in existing DCF-based trackers. More importantly, simply increasing the ratio does not solve the problem. As shown in Fig. 7c, when the background is complex, the precisions of all trackers drop dramatically except for the proposed one. This demonstrates the importance of the weak generalized translation equivariance in visual tracking, especially when the target is moving fast and the background is complex. This, however, has not been well explored before. Moreover, the FPS of each tracker drops when the ratio becomes larger. As shown in Table 5, the FPS of the proposed ADCF drops slower than those of all other trackers except for CF2 and ECO-HC. However, CF2 uses deep features and all detection samples will be resized to a fixed scale, while ECO-HC updates its model only every five frames.

5.3.5 Evaluation of the algorithm for block Toeplitz matrix-vector product

Although theoretically the algorithm proposed in Lee (1986) is faster than the direct matrix-vector

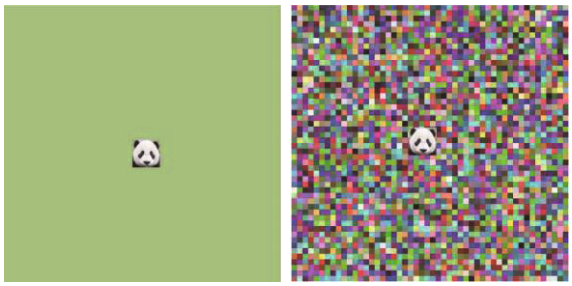

Fig. 6 Example images from the synthetic dataset: (a) image from the first 50 videos, i.e., Syn-a; (b) image from the second 50 videos, i.e., Syn-b

Table 5 FPS of trackers with ratio $A_f/A_t = 5, 10$

Tracker	FPS@5	FPS@10	FPS@5/FPS@10
KCF	69.71	8.89	7.84
DSST	8.81	1.95	4.52
CF2	1.20	0.65	**1.85**
SRDCF	0.62	0.13	4.77
ECO-HC	30.65	7.73	**3.97**
STRCF	9.24	1.79	5.16
ADCF	49.20	11.30	**4.35**

FPS@5/FPS@10 represents the ratio of FPS@5 to FPS@10. The best three results are in bold

product, our implementation of the algorithm in Matlab shows that direct multiplication is faster when the size of the two-level block Toeplitz matrix is small. There are many reasons for this. Direct matrix-vector multiplication is more easily done in parallel, whereas when the matrix is small, the time cost of constructing the polynomials or other necessary storage related operations is more than that of the multiplication itself. To compare the two matrix-vector multiplication approaches, we scale each video to obtain the preset size of the normal matrix, and study how the FPS of the proposed tracker varies with the size of the normal matrix using the two kinds of multiplication. To reduce the scaling error that may be caused by linear interpolation, we choose a subset from OTB-2015 on the condition that the minimum side length of the initial target is greater than 30 pixels, resulting in 70 videos. We use the proposed ADCF to carry out the evaluation. The results are shown in Fig. 8. As shown, when the filter size mn is approximately greater than 4000, the block Toeplitz matrix product is faster than the direct matrix product; when mn is greater than 14 000, the difference is less significant, which confirms that when the sample size MN becomes large enough, the main computational complexity of ADCF is $O((d+2)MN\log(MN)+MN)$, and the time cost for parameter optimization is secondary.

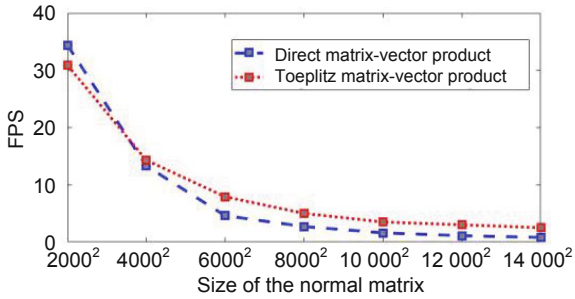

Fig. 8 FPS plot which shows how FPS varies with the size of the normal matrix of ADCF

5.4 Qualitative results

We qualitatively evaluate different trackers on six video sequences (i.e., Human9, Freeman2, Girl2, Rubic, Biker, and Panda from the OTB-2015 dataset). The tracking results are shown in Fig. 9. As shown, our STRACF performs visually more favorable. The sequences Human9, Freeman2, Rubic,

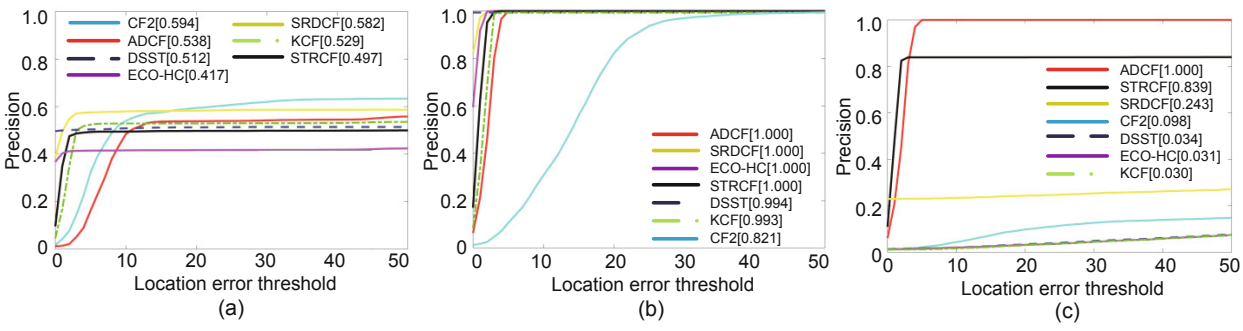

Fig. 7 The precision plots of different trackers on the synthetic datasets. (a) and (b) show the precision plots of all competing trackers on Syn-a with A_f/A_t=5 and 10, respectively; (c) shows the precision plot on Syn-b with A_f/A_t=10

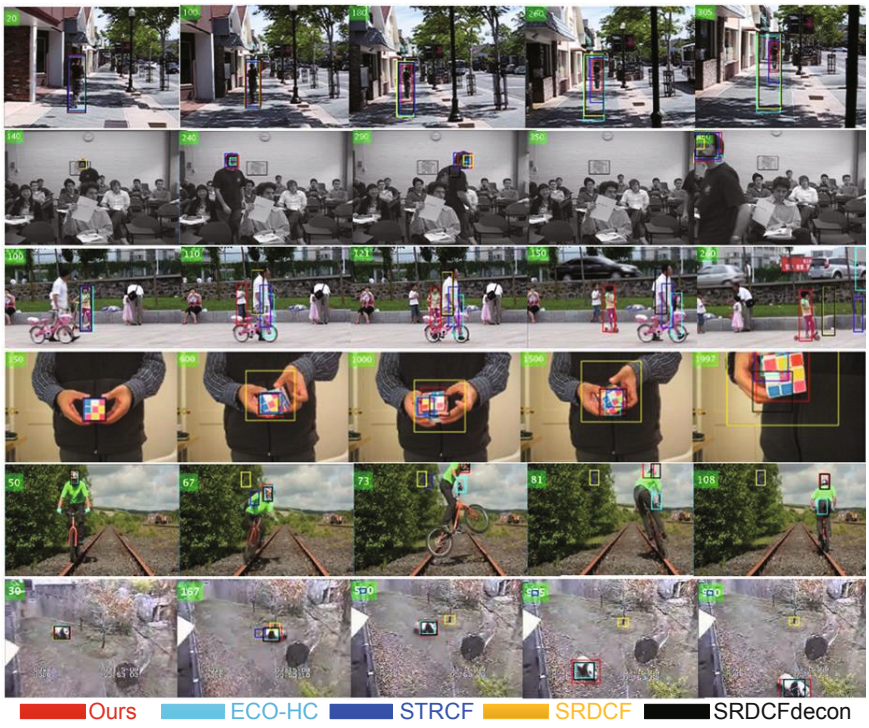

Fig. 9 Qualitative evaluation on six video sequences from OTB-2015 (i.e., Human9, Girl2, Rubic, Freeman2, Biker, and Panda from top to bottom). The results of ECO-HC, STRCF, SRDCF, SRDCFdecon, and our STRACF with different colors are given from left to right

and Panda show that the proposed STRACF is more sensitive to the variations of the target, and therefore the tracking result by the proposed tracker seems to scale more correctly; Girl2 and Rubic show that STRACF succeeds in slowing down and speeding up the filter learning, respectively, when occlusion and fast change occur. When occlusion is removed, the target may come into sight at any location, and therefore the detection sample is supposed to be large so that the target is tracked again. We successfully track the girl because of the larger detection sample we use. The sequence Biker shows that when the target moves fast, the detection sample should be large too; otherwise, the tracking may fail.

6 Conclusions

In this paper, we propose an asymmetric discriminative correlation filter (ADCF) to deal with the problems from which DCF may suffer, that is, inflexibility due to the restriction that training and detection samples must be of the same size as the filter and the lack of generalized translation equivariance. Compared to DCF, ADCF has a weak generalized translation equivariance and its filter size can be smaller than the size of the training or detection sample, which makes ADCF more controllable in terms of the number of filter parameters. The normal matrix of ADCF is well structured, which is good for fast parameter optimization. By simplifying the normal matrix, we derive a fast algorithm for ADCF learning. The proposed STRACF is slightly inferior to the counterpart STRCF on some performance measures. However, in view of its good properties, its improvements over the other two baselines, especially in the synthetic dataset, and the factors that impact the performance, i.e., the simple scale estimation method exploited and the simplification made on the normal matrix, ADCF is still an interesting and promising approach deserving further study for visual tracking.

Contributors

Shui-wang LI and Qian-bo JIANG designed the research. Qi-jun ZHAO, Li LU, and Zi-liang FENG guided the research. Shui-wang LI drafted the manuscript. Qi-jun ZHAO and Li LU revised and finalized the paper.

Compliance with ethics guidelines

Shui-wang LI, Qian-bo JIANG, Qi-jun ZHAO, Li LU, and Zi-liang FENG declare that they have no conflict of interest.

References

Bertinetto L, Valmadre J, Golodetz S, et al., 2016. Staple: complementary learners for real-time tracking. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.1401-1409. https://doi.org/10.1109/CVPR.2016.156

Bolme DS, Beveridge JR, Draper BA, et al., 2010. Visual object tracking using adaptive correlation filters. Proc IEEE Computer Society Conf on Computer Vision and Pattern Recognition, p.2544-2550. https://doi.org/10.1109/CVPR.2010.5539960

Chen BY, Wang D, Li PX, et al., 2018. Real-time 'actor-critic' tracking. Proc 15th European Conf on Computer Vision, p.318-334. https://doi.org/10.1007/978-3-030-01234-2_20

Choi J, Chang HJ, Yun S, et al., 2017. Attentional correlation filter network for adaptive visual tracking. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.4828-4837. https://doi.org/10.1109/CVPR.2017.513

Dalal N, Triggs B, 2005. Histograms of oriented gradients for human detection. Proc IEEE Computer Society Conf on Computer Vision and Pattern Recognition, p.886-893. https://doi.org/10.1109/CVPR.2005.177

Danelljan M, Khan FS, Felsberg M, et al., 2014. Adaptive color attributes for real-time visual tracking. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.1090-1097. https://doi.org/10.1109/CVPR.2014.143

Danelljan M, Häger G, Khan FS, et al., 2015. Learning spatially regularized correlation filters for visual tracking. Proc IEEE Int Conf on Computer Vision, p.4310-4318. https://doi.org/10.1109/ICCV.2015.490

Danelljan M, Häger G, Khan FS, et al., 2016a. Adaptive decontamination of the training set: a unified formulation for discriminative visual tracking. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.1430-1438. https://doi.org/10.1109/CVPR.2016.159

Danelljan M, Robinson A, Khan FS, et al., 2016b. Beyond correlation filters: learning continuous convolution operators for visual tracking. Proc 14th European Conf on Computer Vision, p.472-488. https://doi.org/10.1007/978-3-319-46454-1_29

Danelljan M, Häger G, Khan FS, et al., 2017a. Discriminative scale space tracking. *IEEE Trans Patt Anal Mach Intell*, 39(8):1561-1575. https://doi.org/10.1109/TPAMI.2016.2609928

Danelljan M, Bhat G, Khan FS, et al., 2017b. ECO: efficient convolution operators for tracking. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.6638-6646. https://doi.org/10.1109/CVPR.2017.733

Dong XP, Shen JB, 2018. Triplet loss in Siamese network for object tracking. Proc 15th European Conf on Computer Vision, p.459-474. https://doi.org/10.1007/978-3-030-01261-8_28

Galoogahi HK, Sim T, Lucey S, 2013. Multi-channel correlation filters. Proc IEEE Int Conf on Computer Vision, p.3072-3079. https://doi.org/10.1109/ICCV.2013.381

Galoogahi HK, Fagg A, Lucey S, 2017. Learning background-aware correlation filters for visual tracking. Proc IEEE Int Conf on Computer Vision, p.1135-1143. https://doi.org/10.1109/ICCV.2017.129

Henriques JF, Caseiro R, Martins P, et al., 2015. High-speed tracking with kernelized correlation filters. *IEEE Trans Patt Anal Mach Intell*, 37(3):583-596. https://doi.org/10.1109/TPAMI.2014.2345390

Kart U, Lukezic A, Kristan M, et al., 2019. Object tracking by reconstruction with view-specific discriminative correlation filters. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.1339-1348. https://doi.org/10.1109/CVPR.2019.00143

Kristan M, Leonardis A, Matas J, et al., 2016. The visual object tracking VOT2016 challenge results. Proc Amsterdam on Computer Vision, p.191-217. https://doi.org/10.1007/978-3-319-48881-3_54

Lee D, 1986. Fast multiplication of a recursive block Toeplitz matrix by a vector and its application. *J Complex*, 2(4):295-305. https://doi.org/10.1016/0885-064x(86)90007-5

Li B, Yan JJ, Wu W, et al., 2018. High performance visual tracking with Siamese region proposal network. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.8971-8980. https://doi.org/10.1109/CVPR.2018.00935

Li F, Tian C, Zuo WM, et al., 2018. Learning spatial-temporal regularized correlation filters for visual tracking. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.4904-4913. https://doi.org/10.1109/CVPR.2018.00515

Li Y, Zhu JK, 2014. A scale adaptive kernel correlation filter tracker with feature integration. Proc European Conf on Computer Vision, p.254-265. https://doi.org/10.1007/978-3-319-16181-5_18

Liang PP, Blasch E, Ling HB, 2015. Encoding color information for visual tracking: algorithms and benchmark. *IEEE Trans Image Process*, 24(12):5630-5644. https://doi.org/10.1109/TIP.2015.2482905

Lukezic A, Vojí T, Zajc LC, et al., 2017. Discriminative correlation filter with channel and spatial reliability. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.4847-4856. https://doi.org/10.1109/CVPR.2017.515

Ma C, Huang JB, Yang XK, et al., 2015. Hierarchical convolutional features for visual tracking. Proc IEEE Int Conf on Computer Vision, p.3074-3082. https://doi.org/10.1109/ICCV.2015.352

Mueller M, Smith N, Ghanem B, 2017. Context-aware correlation filter tracking. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.1387-1395. https://doi.org/10.1109/CVPR.2017.152

Nam H, Han B, 2016. Learning multi-domain convolutional neural networks for visual tracking. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.4293-4302. https://doi.org/10.1109/CVPR.2016.465

Pu S, Song Y, Ma C, et al., 2018. Deep attentive tracking via reciprocative learning. Proc 32nd Conf on Neural Information Processing Systems, p.1931-1941.

Qi YK, Zhang SP, Qin L, et al., 2016. Hedged deep tracking. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.4303-4311. https://doi.org/10.1109/CVPR.2016.466

Sun C, Wang D, Lu HC, et al., 2018. Correlation tracking via joint discrimination and reliability learning. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.489-497. https://doi.org/10.1109/CVPR.2018.00058

Sun YX, Sun C, Wang D, et al., 2019. ROI pooled correlation filters for visual tracking. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.5783-5791. https://doi.org/10.1109/CVPR.2019.00593

Tang M, Yu B, Zhang F, et al., 2018. High-speed tracking with multi-kernel correlation filters. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.4874-4883. https://doi.org/10.1109/CVPR.2018.00512

Wang Q, Zhang L, Bertinetto L, et al., 2019. Fast online object tracking and segmentation: a unifying approach. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.1328-1338. https://doi.org/10.1109/CVPR.2019.00142

Wu Y, Lim J, Yang MH, 2013. Online object tracking: a benchmark. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.2411-2418. https://doi.org/10.1109/CVPR.2013.312

Wu Y, Lim J, Yang MH, 2015. Object tracking benchmark. *IEEE Trans Patt Anal Mach Intell*, 37(9):1834-1848. https://doi.org/10.3389/FNINS.2016.00405

Zhang JM, Ma SG, Sclaroff S, 2014. MEEM: robust tracking via multiple experts using entropy minimization. Proc 13th European Conf on Computer Vision, p.188-203. https://doi.org/10.1007/978-3-319-10599-4_13

Appendix A: Proof of the weak translation equivariance of the asymmetric convolution operator

Definition A1 For 2D real-valued arrays $(x_{x_i,y_i}), 0 \le x_i < M, 0 \le y_i < N$, and $(f_{x'_i,y'_i}), 0 \le x'_i < m \le M, 0 \le y'_i < n \le N$, where $M-m$ and $N-n$ are even numbers, the asymmetric convolution of x and f is defined as

$$x \hat{*} f = x * P(f;x), \tag{A1}$$

where $*$ is the circular operator and P is a padding operation defined by

$$\begin{aligned} &P(f;x)|_{x_i,y_i} \\ &= f_{x_i-\delta x,y_i-\delta y} I(\delta x \le x_i \le m+\delta x, \delta y \le y_i \le n+\delta y), \end{aligned} \tag{A2}$$

where $\delta x = (M-m)/2$, $\delta y = (N-n)/2$, and $I(\cdot)$ is the indicator function.

Definition A2 Let x and f be 2D arrays as defined in Definition A1. A binary operator $\circledast$ is said to have the property of generalized translation equivariance if it satisfies

$$\tau_\Delta(x \circledast f) = T_\Delta(o;x) \circledast f, \tag{A3}$$

where $\tau_\Delta(x)$ represents the translation of x by Δ, and $T_\Delta(o;x)$ represents the translation of the subarray $o \subset x$ by Δ in x. If $o = x$, Eq. (A3) becomes $\tau_\Delta(x \circledast f) = \tau_\Delta(x) \circledast f$, and then $\circledast$ is said to have the property of translation equivariance. If $\circledast$ satisfies

$$\tau_\Delta B(x \circledast f) = B(T_\Delta(o;x) \circledast f), \tag{A4}$$

where $B(x)$ is a binary map masking the maximum value of x as 1 and others 0, then $\circledast$ is said to have the property of weak translation equivariance.

Lemma A1 Let x, f, and o be as defined in Definition A2. Suppose o and f are of the same size and $o \hat{\cdot} f > b \hat{\cdot} f$ for any real-valued array b with the same size as o, where $\hat{\cdot}$ represents a flipped inner product defined by

$$\begin{aligned} o \hat{\cdot} f = \sum_{x_i=0}^{M-1} \sum_{y_i=0}^{N-1} &o[(M-x_i) \bmod M, (N-y_i) \bmod N] \\ &\cdot f[x_i, y_i]. \end{aligned} \tag{A5}$$

Denoting the set of coordinates of o in x and f in $P(f;x)$ as R_0 and R_f, respectively, we have

$$B(x * P(f;x)) = B(xI(z \in R_0) * P(f;x)). \tag{A6}$$

Proof The lemma obviously holds for the case $m = M$ and $n = N$. Next, we prove that it also holds for $m < M$ or $n < N$. Let $(\Delta o_1, \Delta o_2)$ and $(\Delta f_1, \Delta f_2)$ denote the coordinates in x and $P(f;x)$, respectively, corresponding to the coordinates $(0,0)$ of o and f. Since

$$\begin{aligned} x[u,v] &= o[u-\Delta o_1, v-\Delta o_2] I((u,v) \in R_\mathrm{o}) \\ &\quad + x[u,v] I((u,v) \in (R \backslash R_\mathrm{o})), \\ P(f;x)[u,v] &= f[u-\Delta f_1, v-\Delta f_2] I((u,v) \in R_f), \end{aligned} \tag{A7}$$

letting u_i and v_i denote $(u-x_i) \bmod M$ and $(v-$

y_i) mod N respectively, we have

$$\begin{aligned}
&x * P(f;x)[u,v]\\
&=\sum_{x_i=0}^{M-1}\sum_{y_i=0}^{N-1} x[(u-x_i) \bmod M,(v-y_i) \bmod N]\\
&\cdot P(f;x)[x_i,y_i]\\
&=\sum_{x_i=0}^{M-1}\sum_{y_i=0}^{N-1} o[u_i-\Delta o_1,v_i-\Delta o_2]I((u_i,v_i)\in R_o)\\
&\cdot f[x_i-\Delta f_1,y_i-\Delta f_2]I((x_i,y_i)\in R_f)\\
&+x[u_i,v_i]I((u_i,v_i)\in (R\backslash R_o),(x_i,y_i)\in R_f)\\
&\cdot f[x_i-\Delta f_1,y_i-\Delta f_2]\\
&=\sum_{x_i=0}^{M-1}\sum_{y_i=0}^{N-1} o[u_i,v_i]f[x_i,y_i]\\
&+x[u_i,v_i]I((u_i,v_i)\in (R\backslash R_o))f[x_i,y_i]\\
&=\sum_{x_i=0}^{M-1}\sum_{y_i=0}^{N-1}(o[u_i,v_i]+x[u_i,v_i])I((u_i,v_i)\in (R\backslash R_o))\\
&\cdot f[x_i,y_i].
\end{aligned}\tag{A8}$$

Similarly, we have

$$xI(z\in R_0)*P(f;x)[u,v]=\sum_{x_i=0}^{M-1}\sum_{y_i=0}^{N-1} o[u_i,v_i]f[x_i,y_i].\tag{A9}$$

According to Eq. (A5) and $o\hat{\cdot}f > b\hat{\cdot}f$ for any real-valued array b, and because $I((M,N)\in R_f)=0$, we know that $x*P(f;x)$ and $xI(z\in R_0)*P(f;x)$ achieve the same maximum value at (M,N), i.e., $o\hat{\cdot}f$. Therefore, it follows that $B(x*P(f;x))=B(xI(z\in R_0)*P(f;x))$.

Theorem A1 Let x, f, and o be as defined in Definition A2. Suppose o and f satisfy the conditions in Lemma A1. Then the asymmetric convolution operator $\hat{*}$ has the property of weak translation equivariance.

Proof Denote the set of coordinates of x by R and that of o in x by $R_0\subset R$. Then the set of coordinates of translated o is $R_1=R_0+\Delta$. Let z denote the coordinate of a point in x and $x[z]$ the value of x at z. If $z\in R_0$ and $z\notin R_1$, using $y[z]$ to denote $T_\Delta(o;x)[z]$, we have

$$\begin{aligned}
T_\Delta(o;x)[z]=&x[z-\Delta]I(z\in R_1)+y[z]I(z\in (R_0\backslash R_1))\\
&+x[z]I(z\in (R\backslash (R_0\cup R_1)))\\
=&\tau_\Delta x[z]I(z\in R_1)+y[z]I(z\in (R_0\backslash R_1))\\
&+x[z]I(z\in (R\backslash (R_0\cup R_1))).
\end{aligned}\tag{A10}$$

Therefore, we have

$$\begin{aligned}
&B(T_\Delta(o;x)\hat{*}f)=B(T_\Delta(o;x)*P(f;x))\\
=&B((\tau_\Delta xI(z\in R_1)+yI(z\in (R_0\backslash R_1))\\
&+xI(z\in (R\backslash (R_0\cup R_1))))*P(f;x))\\
=&B((\tau_\Delta xI(z\in R_1))*P(f;x)) \text{ by Lemma A1.}
\end{aligned}\tag{A11}$$

Because

$$\begin{aligned}
&\tau_\Delta B(x\hat{*}f)=\tau_\Delta B(x*P(f;x))\\
&=\tau_\Delta B(xI(z\in R_0)*P(f;x)) \text{ by Lemma A1}\\
&=B(\tau_\Delta(xI(z\in R_0)*P(f;x)))\\
&=B(\tau_\Delta(xI(z\in R_0))*P(f;x))\\
&=B(((\tau_\Delta x)I(z\in R_1))*P(f;x)),
\end{aligned}\tag{A12}$$

we finally have

$$\tau_\Delta B(x\hat{*}f)=B(T_\Delta(o;x)\hat{*}f).\tag{A13}$$

Appendix B: Proof of the block Toeplitz property of the coefficient matrix

Let $A=(A_{I,J})_{1\le I,J\le N}$ denote an $nN\times nN$ matrix, where $A_{I,J}\in\mathbb{C}^{n\times n}$ is a submatrix of A, denoted by $A_{I,J}=(a_{i,j})_{1\le i,j\le n}$.

A is a block Toeplitz matrix if $A_{I_1,J_1}=A_{I_2,J_2}$ whenever $I_1-J_1=I_2-J_2, 1\le I_1,I_2,J_1,J_2\le N$. If meanwhile any submatrix $A_{I,J}$ is a Toeplitz matrix, i.e., $a_{i_1,j_1}=a_{i_2,j_2}$ whenever $i_1-j_1=i_2-j_2, 1\le i_1,i_2,j_1,j_2\le n$, then A is called a two-level block Toeplitz matrix, in which the sizes of the first and second levels are $N\times N$ and $n\times n$, respectively.

Proposition B1 Let $X=(x_{i,j})$ and $Y=(y_{i,j})$ $(1\le i\le H,\ 1\le j\le W)$ denote two real-valued matrices and $T_{i,j}(X)$ the circular translation of X by i and j in the vertical and horizontal directions, respectively, i.e., $(T_{i,j}(X))_{i',j'}=x_{(i'-i) \bmod H,(j'-j) \bmod W}$. A set of circular translations $\{T_{i,j}\}_{i_1\le i\le i_h,j_1\le j\le j_w}$ $(1\le i_1<i_h\le H, 1\le j_1<j_w\le W)$ can be used to construct an $HW\times hw$ matrix $A(X)$, defined by

$$\begin{aligned}
&A(X)=\\
&[\vec{T}_{i_1,j_1}(X)\ \ \vec{T}_{i_2,j_1}(X)\ \cdots\ \vec{T}_{i_h,j_1}(X)\ \ \cdots\ \vec{T}_{i_h,j_w}(X)],
\end{aligned}\tag{B1}$$

where $\vec{T}_{i,j}(X)$ denotes the vectorized $T_{i,j}(X)$ by stacking columns. Then matrix $B=(A(X))^{\mathrm{T}}A(Y)$ is an $hw\times hw$ two-level block Toeplitz matrix, in which the sizes of the first and second levels are $w\times w$ and $h\times h$, respectively.

Proof Since circular translation and vectorization can be seen as operators acting on any matrix, i.e., $\vec{T}_{i,j}(X)=V(T_{i,j}(X))$, where V denotes the vectorization operation, for simplicity, denote Eq. (B1) as

$$A(X)=[\vec{T}_{i_1,j_1}\quad \vec{T}_{i_2,j_1}\ \cdots\ \vec{T}_{i_h,j_1}\quad \vec{T}_{i_1,j_2}\ \cdots\ \vec{T}_{i_h,j_w}](X). \tag{B2}$$

Let $\mathrm{Sum}(A)$ denote the sum of all entries of matrix A. Then we have

$$\begin{aligned}&(\vec{T}_{i,j}(X))^{\mathrm{T}}\vec{T}_{i',j'}(Y)=\mathrm{Sum}(T_{i,j}(X)\circ T_{i',j'}(Y))\\&=\mathrm{Sum}(X\circ T_{i'-i,j'-j}(Y))=\vec{X}^{\mathrm{T}}\vec{T}_{i'-i,j'-j}(Y),\end{aligned} \tag{B3}$$

where $\circ$ denotes the Hadamard product. Define by $A_j(X)=[\vec{T}_{i_1,j}(X)\quad \vec{T}_{i_2,j}(X)\ \cdots\ \vec{T}_{i_h,j}(X)]$ $(j=j_1,j_2,\cdots,j_w)$ a family of submatrices of $A(X)$. Then $A(X)$ can be written in block matrix form as $A(X)=[A_{j_1}\ A_{j_2}\ \cdots\ A_{j_w}](X)$. Therefore,

$$A(X)^{\mathrm{T}}A(Y)=\begin{bmatrix}A_{j_1}^{\mathrm{T}}A_{j_1} & A_{j_1}^{\mathrm{T}}A_{j_2} & \cdots & A_{j_1}^{\mathrm{T}}A_{j_w}\\ A_{j_2}^{\mathrm{T}}A_{j_1} & A_{j_2}^{\mathrm{T}}A_{j_2} & \cdots & A_{j_2}^{\mathrm{T}}A_{j_w}\\ \vdots & \vdots & & \vdots\\ A_{j_w}^{\mathrm{T}}A_{j_1} & A_{j_w}^{\mathrm{T}}A_{j_2} & \cdots & A_{j_w}^{\mathrm{T}}A_{j_w}\end{bmatrix}(X,Y), \tag{B4}$$

which is a block matrix with $w\times w$ partitions. Now, we have

$$\begin{aligned}&(A_{j'}(X))^{\mathrm{T}}A_j(Y)\\&=\begin{bmatrix}(\vec{T}_{i_1,j'}(X))^{\mathrm{T}}\\(\vec{T}_{i_2,j'}(X))^{\mathrm{T}}\\ \vdots\\(\vec{T}_{i_h,j'}(X))^{\mathrm{T}}\end{bmatrix}[\vec{T}_{i_1,j}(Y)\quad \vec{T}_{i_2,j}(Y)\quad \cdots\quad \vec{T}_{i_h,j}(Y)]\\&=\begin{bmatrix}(\vec{T}_{i_1,j'})^{\mathrm{T}}\vec{T}_{i_1,j} & (\vec{T}_{i_1,j'})^{\mathrm{T}}\vec{T}_{i_2,j} & \cdots & (\vec{T}_{i_1,j'})^{\mathrm{T}}\vec{T}_{i_h,j}\\(\vec{T}_{i_2,j'})^{\mathrm{T}}\vec{T}_{i_1,j} & (\vec{T}_{i_2,j'})^{\mathrm{T}}\vec{T}_{i_2,j} & \cdots & (\vec{T}_{i_2,j'})^{\mathrm{T}}\vec{T}_{i_h,j}\\ \vdots & \vdots & & \vdots\\(\vec{T}_{i_h,j'})^{\mathrm{T}}\vec{T}_{i_1,j} & (\vec{T}_{i_h,j'})^{\mathrm{T}}\vec{T}_{i_2,j} & \cdots & (\vec{T}_{i_h,j'})^{\mathrm{T}}\vec{T}_{i_h,j}\end{bmatrix}(X,Y).\end{aligned} \tag{B5}$$

For simplicity, let $t_{i'-i,j'-j}(X,Y)=\vec{X}^{\mathrm{T}}\vec{T}_{i'-i,j'-j}(Y)$. Because $i_k-i_{k'}=k-k',1\le k,k'\le h$, it follows that

$$\begin{aligned}&(A_{j'}(X))^{\mathrm{T}}A_j(Y)\\&=\begin{bmatrix}t_{0,j-j'} & t_{i_2-i_1,j-j'} & \cdots & t_{i_h-i_1,j-j'}\\ t_{i_1-i_2,j-j'} & t_{0,j-j'} & \cdots & t_{i_{h-1}-i_2,j-j'}\\ \vdots & \vdots & & \vdots\\ t_{i_1-i_h,j-j'} & t_{i_2-i_h,j-j'} & \cdots & t_{0,j-j'}\end{bmatrix}(X,Y)\\&=\begin{bmatrix}t_{0,j-j'} & t_{1,j-j'} & \cdots & t_{h-1,j-j'}\\ t_{-1,j-j'} & t_{0,j-j'} & \cdots & t_{h-2,j-j'}\\ \vdots & \vdots & & \vdots\\ t_{1-h,j-j'} & t_{2-h,j-j'} & \cdots & t_{0,j-j'}\end{bmatrix}(X,Y).\end{aligned} \tag{B6}$$

Therefore, $(A_{j'}(X))^{\mathrm{T}}A_j(Y)$ is an $h\times h$ Toeplitz matrix and $(A_{i'}(X))^{\mathrm{T}}A(Y)_i=(A_{j'}(X))^{\mathrm{T}}A_j(Y)$ if $i'-i=j'-j$. This completes the proof of the proposition. In the special case where $i_1=1,i_h=H$ and $j_1=1,j_w=W$, h just equals H, and then $t_{-1,j-j'}=t_{H-1,j-j'}$, since $(-1)\bmod H=H-1$. By the same token, $t_{-i,j-j'}=t_{H-i,j-j'}$; therefore, $(A_{j'}(X))^{\mathrm{T}}A_j(Y)$ is a circulant matrix, so is $(A(X))^{\mathrm{T}}A(Y)$.