
High Speed Kernelized Correlation Filters without Boundary Effect

Ming TANG, Linyu ZHENG, Bin YU, Jinqiao WANG

University of Chinese Academy of Sciences

National Lab of Pattern Recognition, Institute of Automation, CAS, Beijing 100190, China

tangm@nlpr.ia.ac.cn

Abstract

In recent years, correlation filter based trackers (CF trackers) have attracted much attention in vision community because of their top performance in both location and speed. However, the boundary effect imposed by the periodic assumption for the efficient optimization seriously limits their location accuracy. Although there existed many modern works to relax the boundary effect of CF trackers, they all are not able to eliminate the boundary effect thoroughly as well as exploit the kernel trick to improve their location accuracy, and their speeds are reduced greatly. Either relaxing the boundary effect or being able to exploit kernel trick to improve the accuracy, and either relaxing the boundary effect or keeping CF trackers running at high speed have been two dilemmas in the society of visual tracking. To solve these problems, in this paper, we propose a high speed kernel correlation filter without the boundary effect (nBEKCF). Unlike all current CF trackers which exploited real and virtual image patches to train their regression functions, we construct a set of non-orthogonal bases with a group of circulant basic samples and utilize a function defined in Hilbert space and a set of densely sampled, *totally real* patches to regress a Gaussian goal. This way, the boundary effect is eliminated thoroughly in theory and the kernel trick can be employed naturally. To ensure nBEKCF runs at high speed, we present two efficient algorithms, ACSII and CCIM, to significantly accelerate the evaluation of kernel matrix *without* employing fast Fourier transform (FFT). By exploiting the circulant structure of basic samples, the efficiency of CCIM in evaluating correlation exceeds that of FFT remarkably. Preliminary experimental results on two public datasets, OTB-2013 and OTB-2015, show that, without bells and whistles, our nBEKCF outperforms representative trackers with hand-crafted features, in the meanwhile, runs at 70 fps on average.

1 Introduction

Visual object tracking is one of the fundamental problems in computer vision with many applications. In the model-free tracking task, given the state (e.g., position and size) of a target object in the initial frame, the goal is to estimate the states of the target in the subsequent frames. It is commonly known that despite significant progress in recent decades [35, 20, 23, 30, 24, 28, 25, 27, 21, 45, 43, 16, 37, 5, 33], visual object tracking is still a challenging problem [41] due to some extremely challenging factors (e.g. large appearance changes, occlusions, background clutters and fast motion), very limited training samples, and the requirement for low computational cost. Therefore, it is crucial to construct a robust appearance model from very limited samples to distinguish target from distractive background, while maintain high speeds in tracking.

Since 2010, correlation filter based trackers (CF trackers) have been being proposed and almost dominated the tracking domain in recent years [3, 18, 13, 8, 19, 10, 12, 4, 6, 31, 11, 29, 26].

CF trackers learn their filters from circular samples to regress Gaussian response maps, and their learning and detection are accelerated with the convolution theorem and fast Fourier transform (FFT). Bolme *et al.* [3] proposed the Minimum Output Sum of Squared Error (MOSSE) for very high speed tracking on gray-scale images. They used a base image patch and the circulant virtual ones to train the appearance model directly in Fourier domain. Henriques *et al.* [19] proposed the kernelized correlation filter (KCF) which extended MOSSE to not only learn a multi-channel filter, but also take the advantage of kernel trick to improve its location accuracy. Compared to MOSSE, KCF achieved much higher accuracy on OTB-2013 [42] when exploiting HOG feature [7] and Gaussian kernel, in the meanwhile, it is still able to run at high speed. However, the basic periodic assumption leads to the approximation of real samples, producing the unwanted boundary effect [22] in MOSSES and KCF. This problem negatively affects the discriminativity of appearance models and greatly degrades the location accuracy of MOSSE and KCF. In order to relax the boundary effect, Galoogahi *et al.* [15] and Danelljan *et al.* [9] proposed background-aware correlation filters (BACF) and spatially regularized discriminatively correlation filters (SRDCF), respectively. In BACF, a rectangular mask covered the circular samples, and then the alternating direction method of multipliers (ADMM) was employed to solve the optimization problem with equality constraints. In SRDCF, a smooth spatial regularization factor was introduced into the regularizer to penalize the filter coefficients depending on their spatial location. Even though BACF, and SRDCF relaxed the boundary effect, they still can not eliminate the boundary effect thoroughly in theory. And as the costs of relaxing the boundary effect in their ways, SRDCF and BACF can only run at low speed if powerful features of high dimensionality are exploited, in the meanwhile, they are not able to exploit the kernel trick to improve their accuracy, because it is not clear what the window shape is in the non-linear kernel space, despite the shape is known in an image or filter.

To improve the discriminativity of KCF, Tang and Feng [34] and Tang, *et al.* [36] derived two multi-kernel learning (MKL) [32] based correlation filters (MKCF and MKCFup) which are able to take advantage of the invariance-discriminative power spectrums of various features [39] to improve the location accuracy. Nevertheless, MKCFup, the latest MKL version of KCF, still suffers from the boundary effect, although its location performance outperforms that of KCF.

In summary, up till now, there are at least two principal lines to improve MOSSE and KCF. The first one is to weight the filter or samples with a mask in MOSSE or the KCF of linear kernel, alleviating the boundary effect greatly and improving the location accuracy remarkably. However, the trackers on this line, such as SRDCF and BACF, are unable to employ powerful non-linear kernels to improve their performance. And all trackers on this line are not able to run at high speed if powerful features of high dimensional are exploited. The second line to improve MOSSE and KCF is to modify the objective function of KCF, such as designing more complicated objective functions [2], or introducing the MKL into KCF to adaptively exploit multiple (non-linear) kernels. Although the trackers on this line often maintained high speed in tracking, they suffer from the boundary effect. Consequently, either relaxing the boundary effect or being able to exploit kernel trick to improve the accuracy, and either relaxing the boundary effect or keeping CF trackers running at high speed have been two dilemmas in the visual tracking community.

To solve these two dilemmas, in this paper, we propose a high speed kernel correlation filter without the boundary effect (nBEKCF). Unlike all current CF trackers which exploited several real and virtual patches generated from a single image patch to train their regression functions, we construct a set of non-orthogonal bases with a group of circulant basic samples and utilize a function defined in Hilbert space and a set of densely sampled, *totally real* patches to regress a Gaussian response map. This way, the boundary effect is eliminated thoroughly in theory and the kernel trick can be exploited naturally. In order to treat multiple images and model update problems, we introduce the modeling scheme over multiple images [13] into our nBEKCF. To ensure nBEKCF runs at high speed, inspired by the integral image [40], we present two efficient algorithms, ACSII and CCIM, to significantly accelerate the evaluation of kernel correlation matrices *without* employing FFT. And by exploiting the circulant structure of basic samples, the efficiency of CCIM in evaluating correlation exceed that of FFT remarkably in our task.

In summary, our main contributions are as follows.

- We propose a conceptually simple, flexible and novelty correlation filter entitled nBEKCF which eliminates the boundary effect thoroughly in theory as well as is able to exploit kernel trick naturally.

- We present two efficient algorithms, ACSII and CCIM, which fully exploit the circulant structure of basic samples to construct kernel correlation matrices *without* employing FFT, exceeding the efficiency of FFT in our task.
- We test our novel CF tracker, nBEKCF, on two public datasets, OTB-2013 and OTB-2015. Preliminary experimental results show that, without bells and whistles, our nBEKCF outperforms representative trackers with hand-crafted features, in the meanwhile, runs at 70 fps on average.

We will update our latest experimental results on arxiv.org as soon as they are ready, and release the code to facilitate the future research in visual community.

2 Kernelized Correlation Filters without Boundary Effect (nBEKCF)

Let $\mathcal{H}$ be a Hilbert space of functions $f : \mathbb{R}^{m \times n \times D} \rightarrow \mathbb{R}$. Suppose $\kappa : \mathbb{R}^{m \times n \times D} \times \mathbb{R}^{m \times n \times D} \rightarrow \mathbb{R}$ can be represented as dot product in $\mathcal{H}$, $\mathbf{X}^{0,0} \in \mathbb{R}^{m \times n \times D}$ is an $m \times n \times D$ sample with $\mathbf{x}_{i,j}^{0,0} \in \mathbb{R}^D$, $i = 0, \dots, m-1$ and $j = 0, \dots, n-1$, as its element, $\mathbf{X}_d^{0,0}$ is the d^{th} channel of $\mathbf{X}^{0,0}$. And the circulant samples are generated via

$$\mathbf{X}_d^{i',j'} = \mathbf{P}_m' \mathbf{X}_d^{0,0} \mathbf{Q}_n^{j'}, \quad i' = 0, \dots, m-1, \quad j' = 0, \dots, n-1, \quad d = 0, \dots, D-1, \quad (1)$$

where $\mathbf{P}_m$ and $\mathbf{Q}_n$ are the permutation matrices of $m \times m$ and $n \times n$, respectively [14],

$$\mathbf{P}_m = \begin{bmatrix} \mathbf{0}_{m-1}^\top & 1 \\ \mathbf{I}_{m-1} & \mathbf{0}_{m-1} \end{bmatrix}, \quad \mathbf{Q}_n = \begin{bmatrix} \mathbf{0}_{n-1} & \mathbf{I}_{n-1} \\ 1 & \mathbf{0}_{n-1}^\top \end{bmatrix},$$

$\mathbf{P}_m^\rho$ and $\mathbf{Q}_n^\rho$ are the ρ^{th} power of $\mathbf{P}_m$ and $\mathbf{Q}_n$, respectively, $\mathbf{0}_{l-1}$ is $(l-1) \times 1$ vector of all zeros, $\mathbf{I}_{l-1}$ is $(l-1) \times (l-1)$ identity matrix, $l \in \{m, n\}$. Suppose the elements of $\mathbf{X}_d^{i',j'}$ are $x_{d,i,j}^{i',j'} \in \mathbb{R}$, $\mathbf{X}^{i',j'} \in \mathbb{R}^{m \times n \times D}$ with $\mathbf{x}_{i,j}^{i',j'}$ as its elements, where $\mathbf{x}_{i,j}^{i',j'} = \{x_{0,i,j}^{i',j'}, \dots, x_{D-1,i,j}^{i',j'}\}$, and $\mathbf{Z}^{i'',j''} \in \mathbb{R}^{m \times n \times D}$ are also the $m \times n \times D$ samples with $\mathbf{z}_{i,j}^{i'',j''} \in \mathbb{R}^D$ as its elements, where $i = 0, \dots, m-1$, $j = 0, \dots, n-1$, $i'' = 0, \dots, M-1$, and $j'' = 0, \dots, N-1$. $\mathcal{X} \equiv \{\mathbf{X}^{i',j'}\}$, $\mathcal{Z} \equiv \{\mathbf{Z}^{i'',j''}\}$, $|\mathcal{X}| = mn$, $|\mathcal{Z}| = MN$, $M \geq m$, $N \geq n$. $|\bullet|$ is the number of elements in set $\bullet$. The elements of $\mathcal{Z}$, which is not circulant in general, come from the dense sampling of an image patch. Consider $\{\kappa(\mathbf{X}^{i',j'}, \cdot)\}$ as a set of basis functions, and let

$$f(\mathbf{Z}) = \sum_{i'=0}^{m-1} \sum_{j'=0}^{n-1} \alpha_{i',j'} \kappa(\mathbf{X}^{i',j'}, \mathbf{Z}). \quad (2)$$

$f(\mathbf{Z})$ is called the kernelized correlation filter without boundary effect (nBEKCF). We then construct the following ridge regression problem.

$$\min_{\boldsymbol{\alpha}} F(\boldsymbol{\alpha}) \equiv \frac{1}{2} \sum_{i''=0}^{M-1} \sum_{j''=0}^{N-1} \left(f(\mathbf{Z}^{i'',j''}) - y_{i'',j''} \right)^2 + \frac{\lambda}{2} \boldsymbol{\alpha}^\top \boldsymbol{\alpha} = \frac{1}{2} \|\mathbf{K}\boldsymbol{\alpha} - \mathbf{y}\|_2^2 + \frac{\lambda}{2} \boldsymbol{\alpha}^\top \boldsymbol{\alpha}, \quad (3)$$

where $\mathbf{K}$ is called the $MN \times mn$ kernel correlation matrix of $\mathcal{Z}$ and $\mathcal{X}$ with $\kappa(\mathbf{X}^{i',j'}, \mathbf{Z}^{i'',j''})$'s as its elements, $\boldsymbol{\alpha} = (\alpha_{i',j'} | i' = 0, \dots, m-1, j' = 0, \dots, n-1)$, $\mathbf{y} = (y_{i'',j''} | i'' = 0, \dots, M-1, j'' = 0, \dots, N-1)$, and $\lambda \geq 0$ is the regularization parameter.

To solve for $\boldsymbol{\alpha}$, let $\nabla_{\boldsymbol{\alpha}} F(\boldsymbol{\alpha}) = 0$; it is achieved that $(\mathbf{K}^\top \mathbf{K} + \lambda \mathbf{I}) \boldsymbol{\alpha} = \mathbf{K}^\top \mathbf{y}$. Because $\mathbf{K}^\top \mathbf{K}$ is semi-positive definite, $\mathbf{K}^\top \mathbf{K} + \lambda \mathbf{I}$ is invertible if $\lambda > 0$. Consequently, if $\lambda > 0$, the optimal solution of Problem (3) is

$$\boldsymbol{\alpha}^* = (\mathbf{K}^\top \mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{K}^\top \mathbf{y}. \quad (4)$$

It is noted that if $\mathcal{Z}$ is circulant and $\mathcal{X} = \mathcal{Z}$, $\mathbf{K}$ will be a Gram matrix, and the loss of Problem (3) is exactly the same the loss in normal kernelized correlation filters (KCF) [18].

2.1 Extend to Multiple Images

In visual tracking, the appearance model is often trained with multiple images from different times to improve its robustness. For example, Hare, *et al.* [17] and Danelljan, *et al.* [13] trained appearance models with historical samples. In this section, we will introduce the scheme proposed in [13] into nBEKCF for modelling with multiple images.

Specifically, in nBEKCF, the ridge regression problem of multiple images can be modelled as

$$\min_{\alpha_Q} F_Q(\alpha_Q) \equiv \frac{1}{2} \sum_{q=1}^Q \beta^q \|\mathbf{K}_q \alpha_Q - \mathbf{y}\|_2^2 + \frac{\lambda}{2} \alpha_Q^\top \alpha_Q, \quad (5)$$

where $\mathbf{K}_q$ is the kernel correlation matrix of $\mathcal{Z}_q$ and $\mathcal{X}$, $q = 1, \dots, Q$, $\mathcal{Z}_q$ comes from frame q . $\beta^1 = (1 - \gamma)^{Q-1}$, $\beta^q = \gamma(1 - \gamma)^{Q-q}$, $q = 2, \dots, Q$, Q is the number of historical frames, $\gamma \in (0, 1)$ is the learning rate. $\sum_{q=1}^Q \beta^q = 1$.

Let $\nabla_{\alpha_Q} F_Q(\alpha_Q) = 0$; it is achieved that

$$\sum_{q=1}^Q \beta^q (\mathbf{K}_q^\top \mathbf{K}_q + \lambda \mathbf{I}) \alpha_Q = \sum_{q=1}^Q \beta^q \mathbf{K}_q^\top \mathbf{y}.$$

Because

$$\sum_{q=1}^Q \beta^q (\mathbf{K}_q^\top \mathbf{K}_q + \lambda \mathbf{I}) = \sum_{q=1}^Q \beta^q \mathbf{K}_q^\top \mathbf{K}_q + \lambda \mathbf{I}$$

and $\sum_{q=1}^Q \beta^q \mathbf{K}_q^\top \mathbf{K}_q$ is semi-positive definite, $\sum_{q=1}^Q \beta^q (\mathbf{K}_q^\top \mathbf{K}_q + \lambda \mathbf{I})$ is positive definite if $\lambda > 0$. Consequently, if $\lambda > 0$, the optimal solution of Problem (5) is

$$\alpha_Q^* = \left(\sum_{q=1}^Q \beta^q (\mathbf{K}_q^\top \mathbf{K}_q + \lambda \mathbf{I}) \right)^{-1} \sum_{q=1}^Q \beta^q \mathbf{K}_q^\top \mathbf{y}. \quad (6)$$

While the images come sequentially, an efficient update scheme similar to what was done in [13] can be designed as follows. Let

$$\alpha_Q^* = \left(\mathbf{K}_Q^{(S)} \right)^{-1} \mathbf{v}_Q,$$

where

$$\mathbf{K}_Q^{(S)} = \sum_{q=1}^Q \beta^q (\mathbf{K}_q^\top \mathbf{K}_q + \lambda \mathbf{I}), \quad \mathbf{v}_Q = \sum_{q=1}^Q \beta^q \mathbf{K}_q^\top \mathbf{y}.$$

Then

$$\mathbf{K}_Q^{(S)} = (1 - \gamma) \mathbf{K}_{Q-1}^{(S)} + \gamma (\mathbf{K}_Q^\top \mathbf{K}_Q + \lambda \mathbf{I}), \quad \mathbf{v}_Q = (1 - \gamma) \mathbf{v}_{Q-1} + \gamma \mathbf{K}_Q^\top \mathbf{y}.$$

3 Detection of Target Object

Suppose $\mathcal{Z}'$ is the test set. $|\mathcal{Z}'| = M'N'$. According to Eq. (2), nBEKCF evaluates the response of $\mathbf{Z}^{i'',j''} \in \mathcal{Z}'$ as

$$f(\mathbf{Z}^{i'',j''}) = \sum_{i'=0}^{m-1} \sum_{j'=0}^{n-1} \alpha_{i',j'} \kappa(\mathbf{X}^{i',j'}, \mathbf{Z}^{i'',j''}).$$

Then, the response map, $f(\mathcal{Z}')$, of $\mathcal{Z}'$ is $\{f(\mathbf{Z}^{i'',j''}) | \mathbf{Z}^{i'',j''} \in \mathcal{Z}'\} \triangleq \mathbf{K}' \alpha$, where $\mathbf{K}'$ is the $M'N' \times mn$ correlation matrix of $\mathcal{Z}'$ with $\kappa(\mathbf{X}^{i',j'}, \mathbf{Z}^{i'',j''})$'s as its elements. Note that it is not necessary that $M' = M$ or $N' = N$, although $M' = M$ and $N' = N$ in our current experiments.

4 Fast Evaluation of Correlation Matrix

While solving for the optimal α^* and detecting the target object, correlation matrices $\mathbf{K}_Q$ and $\mathbf{K}'$, each includes $MN \times mn$ different elements, have to be evaluated first. And the evaluation of each element of $\mathbf{K}_Q$ and $\mathbf{K}'$ is involved in samples of D channels. In some time-sensitive tasks, such as visual object tracking, MN , mn , and D are huge, *e.g.*, $MN = 60 \times 60$, $mn = 15 \times 20$, and $D = 41$. If there is no efficient algorithm to evaluate $\mathbf{K}_Q$ and $\mathbf{K}'$, it is almost impossible to apply nBEKCF in such tasks. On the other hand, it ought to be noticed that, while solving for α^* , the computational cost of matrix inversion is usually not a main bottleneck for efficient solution if the inversion is achieved through solving a system of linear equations, because the $mn \times mn$ coefficient matrices are usually not large. For example, the coefficient matrices are only 300×300 in the above example.

Some kernels, such as dot-product kernels, polynomial kernels, and radial basis function kernels [19] are often employed to evaluate the elements of $\mathbf{K}_Q$ and $\mathbf{K}'$. And the evaluation of these usually used kernels are involved in the correlation and/or autocorrelation of samples. In general, the computational complexity of $\mathcal{X} \diamond \mathcal{Z}$, where $\diamond$ is the correlation operator, is $O(m^2 n^2 M N D)$, and that of the autocorrelation $\mathcal{Z} \diamond \mathcal{Z}$ is $O(mn M^2 N^2 D)$. Despite fast Fourier transform (FFT) can accelerate the correlation with the computational complexity $O(mn M N (\log M + \log N) D)$, it is still not satisfactory for some time-sensitive tasks because it is not able to exploit the cyclicity of $\mathcal{X}$. And FFT is not able to accelerate the autocorrelation in our visual object tracking task. Consequently, the computational burdens of correlation and autocorrelation are still main bottlenecks for some time-sensitive tasks, such as visual object tracking.

The key to acceleration of correlation and autocorrelation is to fully exploit the structure of samples, *i.e.*, $\mathcal{X}$ is circulant and $\mathcal{Z}$ is densely sampled from an image patch. In this section, we will develop two algorithms, autocorrelation with squared integral image (ACSII) and circulant correlation with integral matrix (CCIM), to efficiently evaluate $\mathcal{Z} \diamond \mathcal{Z}$ and $\mathcal{X} \diamond \mathcal{Z}$. The computational complexities of ACSII and CCIM are $O(M N D)$ and $O(mn M N D)$, respectively, lower than those with FFT.

In order to experimentally verify the superior efficiency of CCIM over that of FFT, we evaluate $\mathcal{X} \diamond \mathcal{Z}$ under a typical situation in visual object tracking, *i.e.*, $m = 15$, $n = 20$, $M = N = 60$, and $D = 31 + 10 = 41$, which are the case if CN [38] and HOG with cell sizes being 4×4 are employed. It is not surprised to see that CCIM takes only 30ms, while FFT 530ms, to achieve $\mathcal{X} \diamond \mathcal{Z}$. We also evaluate $\mathcal{Z} \diamond \mathcal{Z}$ in the above example with ACSII. $\mathcal{Z} \diamond \mathcal{Z}$ is then achieved in 3ms.

4.1 Autocorrelation with Squared Integral Image (ACSII)

While $\mathcal{Z}$ is sampled from a single image patch $\mathbf{Z} \in \mathbb{R}^{M \times N \times D}$, the necessary computation for $\mathcal{Z} \diamond \mathcal{Z}$ is the squares, $\|\mathbf{z}_{i'',j''}\|^2$, where $\mathbf{z}_{i'',j''} \in \mathbb{R}^D$ is the element of $\mathbf{Z}$, $i'' = 0, \dots, M-1$, and $j'' = 0, \dots, N-1$. We then design an integral image $\mathbf{I}$ to evaluate

$$(\mathbf{z}^{i'',j''})^\top \mathbf{z}^{i'',j''} \triangleq \sum_{i=i''}^{m+i''-1} \sum_{j=j''}^{n+j''-1} \sum_{d=0}^{D-1} Z_{i,j,d}^2$$

after obtaining all the squares, where $\mathbf{z}^{i'',j''} \in \mathcal{Z}$ with top left corner coordinates (i'',j'') . The detailed steps are shown in Algorithm 1.

4.2 Circulant Correlation with Integral Matrix (CCIM)

It is clear that $\mathbf{X}^{i',j'} \diamond \mathbf{Z} = \left\{ \langle \mathbf{X}^{i',j'}, \mathbf{Z}^{i'',j''} \rangle \mid i'' = 0, \dots, M-m+1, j'' = 0, \dots, N-n+1 \right\}$, where $\langle, \rangle$ is the dot product. Because $\mathcal{X}$ is circulant, the cost of whole computation, *i.e.*, $\mathbf{X}^{i',j'} \diamond \mathbf{Z}$'s for all $\mathbf{X}^{i',j'} \in \mathcal{X}$, can be reduced significantly. Before formally presenting our efficient algorithm, Circulant Correlation with Integral Matrix (CCIM), we first explain its basic logic steps as follows.

(L1) Only the necessary computation is conducted. Because $\mathcal{X}$ is circulant, the necessary computations for the whole $\mathbf{X}^{i',j'} \diamond \mathbf{Z}$'s are $\langle \mathbf{x}_{i,j}^{0,0}, \mathbf{z}_{i'',j''} \rangle$, where $i = 0, \dots, m-1$, $j = 0, \dots, n-1$, $i'' = 0, \dots, M-1$, $j'' = 0, \dots, N-1$.

Algorithm 1 Autocorrelation with Square Integral Image (ACSII)

- **Input:** $\mathbf{Z} \in \mathbb{R}^{M \times N \times D}$ with $Z_{i,j,d}$'s as its elements, m , and n , where $m \leq M$ and $n \leq N$. D is the number of channels.
- **Output:** autocorrelation $\mathbf{B} \in \mathbb{R}^{(M-m+1) \times (N-n+1)}$ with $B_{i,j}$'s as its elements.
- **Construct squared image** Set $M \times N$ matrix $\mathbf{A}$ with $A_{i,j}$'s as its elements.

$$A_{i,j} = \sum_{d=0}^{D-1} Z_{i,j,d}^2.$$

- **Construct squared integral image $\mathbf{I}$**

1. $I_{0,0} = A_{0,0}$,
2. for $i = 1$ to $M - 1$: $I_{i,0} = I_{i-1,0} + A_{i,0}$.
3. for $j = 1$ to $N - 1$: $I_{0,j} = I_{0,j-1} + A_{0,j}$.
4. for $i = 1$ to $M - 1$, $j = 1$ to $N - 1$: $I_{i,j} = I_{i-1,j} + I_{i,j-1} - I_{i-1,j-1} + A_{i,j}$.

- **Evaluate autocorrelation of $\mathbf{Z}$**

1. $B_{0,0} = I_{m-1,n-1}$,
2. for $i = 1$ to $M - m$: $B_{i,0} = I_{i+m-1,n-1} - I_{i-1,n-1}$.
3. for $j = 1$ to $N - n$: $B_{0,j} = I_{m-1,j+n-1} - I_{m-1,j-1}$.
4. for $i = 1$ to $M - m$, $j = 1$ to $N - n$:

$$B_{i,j} = I_{i+m-1,j+n-1} - I_{i-1,j+n-1} - I_{i+m-1,j-1} + I_{i-1,j-1}.$$

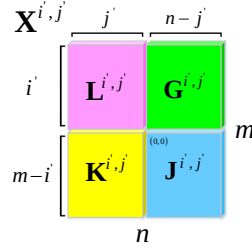


Figure 1: The division of circulant correlation filter $\mathbf{X}^{i',j'}$ into four sub-filters, $\mathbf{L}^{i',j'}$, $\mathbf{G}^{i',j'}$, $\mathbf{K}^{i',j'}$, and $\mathbf{J}^{i',j'}$, by the position of $\mathbf{x}_{0,0}^{0,0}$. See text for details.

(L2) Correlation filter $\mathbf{X}^{i',j'}$ is divided into four sub-filters (sub-matrices), *i.e.*, $\mathbf{L}^{i',j'}$, $\mathbf{G}^{i',j'}$, $\mathbf{K}^{i',j'}$, and $\mathbf{J}^{i',j'}$, by the position of $\mathbf{x}_{0,0}^{0,0}$. This is shown in Fig.1. In each sub-matrices, the relative spatial positions of different elements are the same as their relative positions in $\mathbf{X}^{0,0}$. And the relative spatial positions of two elements belonging to different sub-matrices are different from their original relative positions in $\mathbf{X}^{0,0}$. This way, $\mathbf{X}^{i',j'} \diamond \mathbf{Z}$ is decomposed into four parts, *i.e.*, $\mathbf{X}^{i',j'} \diamond \mathbf{Z} = d_L^{i',j'} + d_G^{i',j'} + d_K^{i',j'} + d_J^{i',j'}$.

(L3) The novel integral matrix is constructed.

(L4) $d_L^{i',j'}$, $d_G^{i',j'}$, $d_K^{i',j'}$, $d_J^{i',j'}$ are achieved with the integral matrix *within constant duration*.

(L5) $d_L^{i',j'}$, $d_G^{i',j'}$, $d_K^{i',j'}$, $d_J^{i',j'}$ are aligned and then summed. Then the correlation $\mathbf{X}^{i',j'} \diamond \mathbf{Z}$ is achieved.

An example to illustrate how CCIM works is provided in supplementary material. The formal description of CCIM is presented in Algorithm 2. The mathematical proof of its correctness is also included in supplementary material. To improve the efficiency of CCIM further, any $\mathbf{P}_M^i \mathbf{S} \mathbf{Q}_N^j$ is conducted with periodic extension and retrieval, rather than with matrix multiplication.

5 Implementation Details

In our experiments, the color names (CN) and HOG are employed as features in nBEKCF. Both cell sizes of CN and HOG are 4×4 . The numbers of CN and HOG channels are 10 and 31, respectively.

Algorithm 2 Circulant Correlation with Integral Matrix (CCIM)

- **Input:** $\mathbf{X}^{0,0} \in \mathbb{R}^{m \times n \times D}$ with $X_{i,j,d}^{0,0}$'s as its elements, $\mathbf{Z} \in \mathbb{R}^{M \times N \times D}$ with $\mathbf{Z}_d \in \mathbb{R}^{M \times N}$ as its d^{th} channel matrix. $m \leq M$ and $n \leq N$, D is the number of channels.
 - **Output:** $\mathbf{C}^{i',j'} \in \mathbb{R}^{(M-m+1) \times (N-n+1)}$, the correlation of $\mathbf{X}^{i',j'}$ and $\mathbf{Z}$, $i' = 0, \dots, m-1$, and $j' = 0, \dots, n-1$.
 - **Initialize**
 - for $i = 0$ to $m-1$, $j = 0$ to $n-1$:
Construct $M \times N$ fundamental matrix $\mathbf{B}^{i,j} = \mathbf{P}_M^{-i} \left(\sum_{d=0}^{D-1} X_{i,j,d}^{0,0} \cdot \mathbf{Z}_d \right) \mathbf{Q}_N^{-j}$.
 - **Construct integral matrix** $\{\mathbf{M}_{i,j} \in \mathbb{R}^{M \times N} | i = 0, \dots, m-1; j = 0, \dots, n-1\}$
 1. $\mathbf{M}_{0,0} = \mathbf{B}^{0,0}$,
 2. for $i = 1$ to $m-1$: $\mathbf{M}_{i,0} = \mathbf{M}_{i-1,0} + \mathbf{B}^{i,0}$.
 3. for $j = 1$ to $n-1$: $\mathbf{M}_{0,j} = \mathbf{M}_{0,j-1} + \mathbf{B}^{0,j}$.
 4. for $i = 1$ to $m-1$, $j = 1$ to $n-1$: $\mathbf{M}_{i,j} = \mathbf{M}_{i-1,j} + \mathbf{M}_{i,j-1} - \mathbf{M}_{i-1,j-1} + \mathbf{B}^{i,j}$.
 - **Evaluate correlation of X and Z**
 - for $i' = 0$ to $m-1$, $j' = 0$ to $n-1$:
 1. $\mathbf{S}^{L,i',j'} = \mathbf{M}_{m-1,n-1} - \mathbf{M}_{m-1,n-j'-1} - \mathbf{M}_{m-i'-1,n-1} + \mathbf{M}_{m-i'-1,n-j'-1}$. If $i' = 0$ or $j' = 0$, $\mathbf{S}^{L,i',j'} = \mathbf{0}_{M \times N}$.
 2. $\mathbf{S}^{G,i',j'} = \mathbf{M}_{m-1,n-j'-1} - \mathbf{M}_{m-i'-1,n-j'-1}$. If $i' = 0$, $\mathbf{S}^{G,i',j'} = \mathbf{0}_{M \times N}$.
 3. $\mathbf{S}^{K,i',j'} = \mathbf{M}_{m-i'-1,n-1} - \mathbf{M}_{m-i'-1,n-j'-1}$. If $j' = 0$, $\mathbf{S}^{K,i',j'} = \mathbf{0}_{M \times N}$.
 4. $\mathbf{S}^{J,i',j'} = \mathbf{M}_{m-i'-1,n-j'-1}$.
 5. $\mathbf{S}_L^{i',j'} = \mathbf{P}_M^{m-i'} \mathbf{S}^{L,i',j'} \mathbf{Q}_N^{n-j'}$,
 6. $\mathbf{S}_G^{i',j'} = \mathbf{P}_M^{m-i'} \mathbf{S}^{G,i',j'} \mathbf{Q}_N^{n-j'}$,
 7. $\mathbf{S}_K^{i',j'} = \mathbf{P}_M^{m-i'} \mathbf{S}^{K,i',j'} \mathbf{Q}_N^{n-j'}$,
 8. $\mathbf{S}_J^{i',j'} = \mathbf{P}_M^{m-i'} \mathbf{S}^{J,i',j'} \mathbf{Q}_N^{n-j'}$.
 9. $\mathbf{C}^{i',j'} = (\mathbf{S}_L^{i',j'} + \mathbf{S}_G^{i',j'} + \mathbf{S}_K^{i',j'} + \mathbf{S}_J^{i',j'})((0,0), (M-m, N-n))$, where $\mathbf{H}((a_1, b_1), (a_2, b_2))$ is the sub-matrix of matrix $\mathbf{H}$ with (a_1, b_1) and (a_2, b_2) as its top left and down right corner subscripts.
-

The response map $\mathbf{y}$ is identical to that in KCF [18]. The learning rate $\gamma = 0.01$. While we construct the kernel correlation matrices, Gaussian kernel is employed with $\sigma = 4$. m and n are the height and width of the target object. The size of search region is set $M = N = 3\sqrt{mn}$. The initial target object of $mn < 1000$ is considered small object, and will be resized to 1000, keeping the aspect ratio unchanged in whole sequence. The scale of target object is estimated with fDSST [11].

6 Experimental Results

The nBEKCF was implemented in MATLAB and C++. The experiments were performed on a PC with Intel Core i7 3.60GHz CPU and 8GB RAM. The speed of nBEKCF is 70 fps on average. The experimental results reported in this section is preliminary.

All trackers are quantitatively evaluated by six criteria, (i) Center Error, which is calculated as the average Euclidean distance between the centers of located objects and their ground truths in a sequence; (ii) Distance Precision, which is the percentage of frames where the objects are located within the center errors of 0 to t_c pixels, with $t_c = 20$; (iii) Precision Plot, which is simply a curve of the distance precisions with t_c changing from 0 to 50 pixels; (iv) Overlap Ratio, which is defined as the average ratio of intersection and union of the estimated bounding box and ground truth in a sequence; (v) Overlap Precision, which is the percentage of frames with the overlap ratio exceeding t_o in a sequence, with $t_o = 0.5$; (vi) Success Plot, which is simply a curve of overlap precisions with the overlap ratio changing from 0 to 1, and AUC is the area under the success plot.

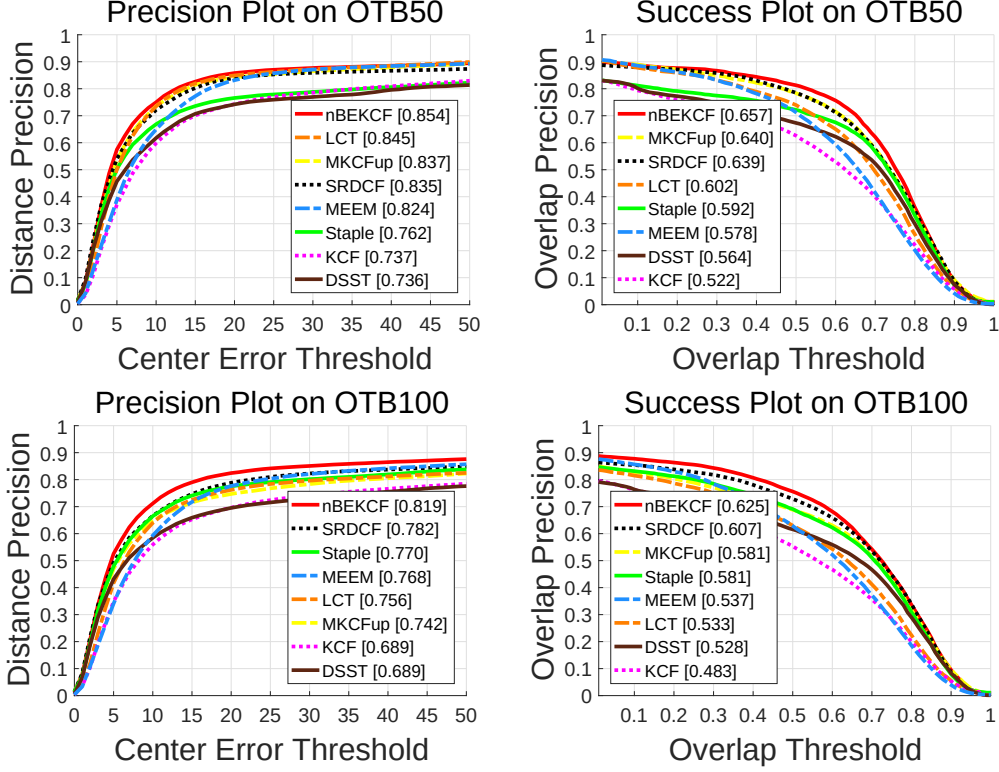


Figure 2: The precision and success plots of nBEKCF, MKCFup, SRDCF, Staple [1], LCT [28], KCF, DSST [8], MEEM [44] on OTB-2013 [42] and OTB-2015 [41]. The average precision scores and AUCs of the trackers on the sequences are reported in the legends.

Table 1: Comparison with state-of-the-art trackers on the OTB-2013 and OTB-2015 datasets using mean overlap precision. The best and second best results for each dataset are shown in red and blue colors, respectively. Our nBEKCF achieves a gain of 2.76% and 2.74% on OTB-2013 and OTB-2015, respectively, compared to the second best SRDCF.

	nBEKCF	MKCFup	SRDCF	Staple	LCT	KCF	DSST	MEEM
Mean OP (OTB-2013)	0.8123	0.7836	0.7847	0.7214	0.7393	0.6256	0.6726	0.7124
Mean OP (OTB-2015)	0.7556	0.6885	0.7282	0.6909	0.6298	0.5528	0.6150	0.6284

We compare nBEKCF with SRDCF and several other state-of-the-art trackers on OTB-2013 [42] and OTB-2015 [41]. SRDCF is selected because it is one of the representative trackers which only try to alleviate the boundary effect to improve its location accuracy. Fig. 2 shows the results. It is seen from the figure that nBEKCF outperforms SRDCF on the average precision score and AUC by 1.8% and 1.9% on OTB-2013, respectively, and 3.7% and 1.8% on OTB-2015, respectively.

Table. 1 shows the mean Overlap Precision (OP) of our nBEKCF and MKCFup, SRDCF, Staple [1], LCT [28], KCF, DSST [8], MEEM [44] on OTB-2013 and OTB-2015 datasets. The best results on these two datasets are obtained by nBEKCF with a mean OP of 81.23% and 75.56% respectively, leading to a significant gain of 2.76% and 2.74% compared to the second best SRDCF.

7 Conclusions

A novel high speed kernelized correlation filter without boundary effect (nBEKCF) is presented in this paper. Without boundary effect, being able to exploit kernel trick, and running at high speed, for the first time in the history of correlation filters, nBEKCF possesses all these characteristics that were hard to manage simultaneously.

References

- [1] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H.S. Torr. Staple: Complementary learners for real-time tracking. In *Proc. Computer Vision and Pattern Recognition*, 2016.
- [2] A. Bibi, M. Mueller, and B. Ghanem. Target response adaptation for correlation filter tracking. In *Proc. European Conference on Computer Vision*, 2016.
- [3] D. Bolme, R. Beveridge, B. Draper, and Y. Lui. Visual object tracking using adaptive correlation filters. In *Proc. Computer Vision and Pattern Recognition*, 2010.
- [4] J.-W. Choi, H. Chang, J. Jeong, Y. Demiris, and J.-Y. Choi. Visual tracking using attention-modulated disintegration and integration. In *Proc. Computer Vision and Pattern Recognition*, 2016.
- [5] J.-W. Choi, H. Chang, S. Yun, T. Fischer, Y. Demiris, and J.-Y. Choi. Attentional correlation filter network for adaptive visual tracking. In *Proc. Computer Vision and Pattern Recognition*, 2017.
- [6] Z. Cui, S. Xiao, J. Feng, and S. Yan. Recurrently target-attending tracking. In *Proc. Computer Vision and Pattern Recognition*, 2016.
- [7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. Computer Vision and Pattern Recognition*, 2005.
- [8] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *Proc. British Machine Vision Conference (BMVC)*, 2014.
- [9] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Learning spatially regularized correlation filters for visual tracking. In *Proc. International Conference on Computer Vision*, 2015.
- [10] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking. In *Proc. Computer Vision and Pattern Recognition*, 2016.
- [11] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Discriminative scale space tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.
- [12] M. Danelljan, A. Robinson, F. Shahbaz Khan, and M. Felsberg. Learning continuous convolution operators for visual tracking. In *Proc. European Conference on Computer Vision*, 2016.
- [13] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. van de Weijer. Adaptive color attributes for real-time visual tracking. In *Proc. Computer Vision and Pattern Recognition*, 2014.
- [14] P. David. *Circulant Matrices*. Chelsea Publishing Company, 2nd edition, 1994.
- [15] H. Galoogahi, A. Fagg, and S. Lucey. Learning background-aware correlation filters for visual tracking. In *Proc. International Conference on Computer Vision*, 2017.
- [16] B. Han, J. Sim, and H. Adam. Branchout: Regularization for online ensemble tracking with convolutional neural networks. In *Proc. Computer Vision and Pattern Recognition*, 2017.
- [17] S. Hare, A. Saffari, and P. Torr. Struck: Structured output tracking with kernels. In *Proc. International Conference on Computer Vision*, 2011.
- [18] J. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *Proc. European Conference on Computer Vision*, 2012.
- [19] J. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.37:pp.583–596, 2015.
- [20] Z. Hong, C. Wang, X. Mei, D. Prokhorov, and D. Tao. Tracking using multilevel quantizations. In *Proc. European Conference on Computer Vision*, 2014.
- [21] Y. Hua, K. Alahari, and C. Schmid. Online object tracking with proposal selection. In *Proc. International Conference on Computer Vision*, 2015.
- [22] B. Kumar, A. Mahalanobis, and R. Juday. *Correlation Pattern Recognition*. Cambridge University Press, 2005.
- [23] J. Kwon, J. Roh, K.-M. Lee, and L. Van Gool. Robust visual tracking with double bounding box model. In *Proc. European Conference on Computer Vision*, 2014.
- [24] D. Lee, J.-Y. Sim, and C.-S. Kim. Visual tracking using pertinent patch selection and masking. In *Proc. Computer Vision and Pattern Recognition*, 2014.
- [25] T. Liu, G. Wang, and Q. Yang. Real-time part-based visual tracking via adaptive correlation filters. In *Proc. Computer Vision and Pattern Recognition*, 2015.
- [26] A. Lukezic, T. Tomas Vojir, L.-C. Zajc, J. Matas, and M. Kristan. Discriminative correlation filter with channel and spatial reliability. In *Proc. Computer Vision and Pattern Recognition*, 2017.
- [27] C. Ma, J. Huang, X. Yang, and M. Yang. Hierarchical convolutional features for visual tracking. In *Proc. International Conference on Computer Vision*, 2015.
- [28] C. Ma, X. Yang, C. Zhang, and M. Yang. Long-term correlation tracking. In *Proc. Computer Vision and Pattern Recognition*, 2015.
- [29] M. Mueller, N. Smith, and B. Ghanem. Context-aware correlation filter tracking. In *Proc. Computer Vision and Pattern Recognition*, 2017.
- [30] H. Nam, S. Hong, and B. Han. Online graph-based tracking. In *Proc. European Conference on Computer Vision*, 2014.

- [31] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and Y. M-H. Hedged deep tracking. In *Proc. Computer Vision and Pattern Recognition*, 2016.
- [32] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521, 2008.
- [33] D. Rozumnyi, J. Kotera, S. Sroubek, L. Novotny, and J. Matas. The world of fast moving objects. In *Proc. Computer Vision and Pattern Recognition*, 2017.
- [34] M. Tang and J. Feng. Multi-kernel correlation filter for visual tracking. In *Proc. International Conference on Computer Vision*, 2015.
- [35] M. Tang and X. Peng. Robust tracking with discriminative ranking lists. *IEEE Transactions on Image Processing*, Vol.21(No.7):3273–3281, 2012.
- [36] M. Tang, B. Yu, F. Zhang, and J. Wang. High-speed tracking with multi-kernel correlation filters. In *Proc. Computer Vision and Pattern Recognition*, 2018.
- [37] J. Valmadre, L. Luca Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr. End-to-end representation learning for correlation filter based tracking. In *Proc. Computer Vision and Pattern Recognition*, 2017.
- [38] J. van de Weijer, C. Schmid, J. Verbeek, and D. Larlus. Learning color names for real-world applications. *IEEE Transactions on Image Processing*, 18(7):1512–1524, 2009.
- [39] M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *Proc. International Conference on Computer Vision*, 2007.
- [40] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.
- [41] Y. Wu, J. Lim, and M. Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37:pp.1834–1848, 2015.
- [42] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking - a benchmark. In *Proc. Computer Vision and Pattern Recognition*, 2013.
- [43] S. Yun, J.-W. Choi, Y. Yoo, K. Yun, and J.-Y. Choi. Action-decision networks for visual tracking with deep reinforcement learning. In *Proc. Computer Vision and Pattern Recognition*, 2017.
- [44] J. Zhang, S. Ma, and S. Sclaroff. MEEM: robust tracking via multiple experts using entropy minimization. In *Proc. European Conference on Computer Vision*, 2014.
- [45] T. Zhang, A. Bibi, and B. Ghanem. In defense of sparse tracking: Circulant sparse tracking. In *Proc. Computer Vision and Pattern Recognition*, 2016.

Supplementary Material

High Speed Kernelized Correlation Filters without Boundary Effect

Ming TANG, Linyu ZHENG, Bin YU, Jinqiao WANG

University of Chinese Academy of Sciences

National Lab of Pattern Recognition, Institute of Automation, CAS, Beijing 100190, China

Abstract

This supplementary material contains three parts. 1) Illustrations of Algorithm 2, CCIM. 2) Mathematical proof of the correctness of CCIM. 3) Other experimental results on popular data sets.

1 Notation

Let $\diamond$ and $\cdot$ be the correlation operator and dot product, respectively, $\mathbf{H}((a_1, b_1), (a_2, b_2))$ be the sub-matrix of matrix $\mathbf{H}$ with (a_1, b_1) and (a_2, b_2) as its top left and down right corner subscripts, respectively, and $H_{a,b}$ be an element of matrix $\mathbf{H}$.

2 Illustration of Algorithm 2 (CCIM)

Let two images be

$$\mathbf{X}^{0,0} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad \text{and} \quad \mathbf{Z} = \begin{bmatrix} a & b & c & d & e \\ f & g & h & i & j \\ k & l & m & n & o \\ p & q & r & s & t \\ u & w & x & y & z \end{bmatrix},$$

respectively, where the numbers in $\mathbf{X}^{0,0}$ indicate the elements of $\mathbf{X}^{0,0}$, rather than the elements' real feature values. $\mathbf{X}^{0,0}$ and $\mathbf{Z}$ are called base correlation filters and signal, respectively. Let 5×5 minimal evaluation matrices $\mathbf{A}^{i,j} = \mathbf{X}_{i,j}^{0,0} \cdot \mathbf{Z}$, and 5×5 fundamental matrices $\mathbf{B}^{i,j} = \mathbf{P}_5^{-i} \mathbf{A}^{i,j} \mathbf{Q}_5^{-j}$, where $\mathbf{P}_5$ and $\mathbf{Q}_5$ are defined by Eq.(2) with $M = N = 5$, $i = 0, 1, 2$, and $j = 0, 1, 2$. $\mathbf{A}^{i,j}$'s and $\mathbf{B}^{i,j}$'s are shown in Fig.1¹.

2.1 Correlation of $\mathbf{X}^{0,0}$ and $\mathbf{Z}$

It is clear that $\mathbf{X}^{0,0} \diamond \mathbf{Z}(0,0) = \langle (1, 2, 3, 4, 5, 6, 7, 8, 9), (a, b, c, f, g, h, k, l, m) \rangle$. According to the construction of $\mathbf{A}^{i,j}$'s and $\mathbf{B}^{i,j}$'s, $\mathbf{X}_{i,j}^{0,0} \cdot \mathbf{Z}_{i,j} = \mathbf{B}_{0,0}^{i,j}$, $i = 0, 1, 2$ and $j = 0, 1, 2$. $\mathbf{X}_{i,j}^{0,0} \cdot \mathbf{Z}_{i,j}$'s are marked by the blue blocks in Fig.1(b). Therefore,

$$\mathbf{X}^{0,0} \diamond \mathbf{Z}(0,0) = \sum_{i=0}^2 \sum_{j=0}^2 \mathbf{B}_{0,0}^{i,j}.$$

¹All the numbers of equations and figures refer to the equations and figures of this supplementary material.

Figure 1: (a) Minimal evaluation matrices $\mathbf{A}^{i,j}$'s. (b) Fundamental matrices, $\mathbf{B}^{i,j}$'s, with sub-matrices marked by red bounding boxes. These sub-matrices are used while evaluating $\mathbf{X}^{0,0} \diamond \mathbf{Z}$. See Sec.2.1 for details. (c) Fundamental matrices, $\mathbf{B}^{i,j}$'s, with sub-matrices marked by red bounding boxes. These sub-matrices are used while evaluating $\mathbf{X}^{1,1} \diamond \mathbf{Z}$. See Sec.2.2 for details.

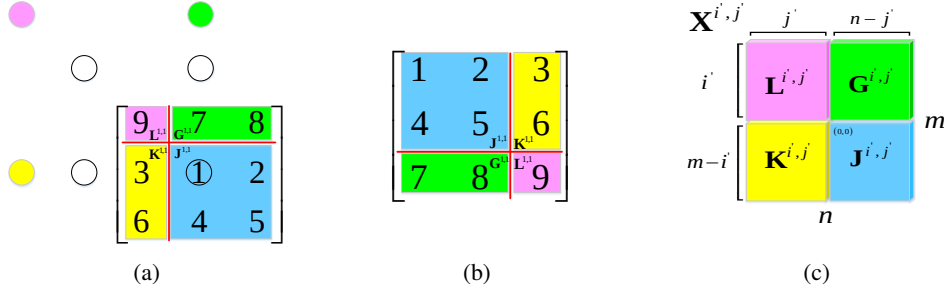


Figure 2: (a) Based on the position of $X_{0,0}^{0,0}$, $\mathbf{X}^{1,1}$ is divided into four parts, $\mathbf{L}^{1,1} = (9)$, $\mathbf{G}^{1,1} = (7, 8)$, $\mathbf{K}^{1,1} = (3, 6)$, and $\mathbf{J}^{1,1} = (1, 2, 4, 5)$. There are pink, green, yellow, and blue blocks. Each color block indicates that the items of correlation $\mathbf{X}^{1,1} \diamond \mathbf{Z}$, which contain the elements covered by that color block as their factors, locate the identical position of different fundamental matrices $\mathbf{B}^{i,j}$'s. See Sec.2.2 for details. (b) The positions of $\mathbf{L}^{1,1}$, $\mathbf{G}^{1,1}$, $\mathbf{K}^{1,1}$, and $\mathbf{J}^{1,1}$ in $\mathbf{X}^{0,0}$. (c) The division of a general circulant correlation filter $\mathbf{X}^{i',j'}$ into four parts, $\mathbf{L}^{i',j'}$, $\mathbf{G}^{i',j'}$, $\mathbf{K}^{i',j'}$, and $\mathbf{J}^{i',j'}$ by the position of $X_{0,0}^{0,0}$.

Similarly, $\mathbf{X}^{0,0} \diamond \mathbf{Z}(i'', j'') = \langle (1, 2, 3, 4, 5, 6, 7, 8, 9), \mathbf{z}_{J;i'',j''}^{0,0} \rangle$, where

$$\mathbf{z}_{J;i'',j''}^{0,0} \triangleq (Z_{i'',j''}, Z_{i'',j''+1}, Z_{i'',j''+2}, \\ Z_{i''+1,j''}, Z_{i''+1,j''+1}, Z_{i''+1,j''+2}, \\ Z_{i''+2,j''}, Z_{i''+2,j''+1}, Z_{i''+2,j''+2}),$$

$i'' = 0, 1, 2$ and $j'' = 0, 1, 2$, and $X_{i,j}^{0,0} \cdot Z_{i''+i,j''+j} = B_{i'',j''}^{i,j}$. Therefore,

$$\mathbf{X}^{0,0} \diamond \mathbf{Z}(i'', j'') = \sum_{i=0}^2 \sum_{j=0}^2 B_{i'',j''}^{i,j},$$

or set $\mathbf{C}^{0,0} = \sum_{i=0}^2 \sum_{j=0}^2 \mathbf{B}^{i,j}$, and then

$$\mathbf{X}^{0,0} \diamond \mathbf{Z} = \mathbf{C}^{0,0}((0, 0), (2, 2)).$$

2.2 Correlation of $\mathbf{X}^{1,1}$ and $\mathbf{Z}$

In general, as shown in Fig.2(c), any circulant correlation filter $\mathbf{X}^{i',j'}$ generated with Eq.(1) can always be divided into four parts, $\mathbf{L}^{i',j'}$, $\mathbf{G}^{i',j'}$, $\mathbf{K}^{i',j'}$, and $\mathbf{J}^{i',j'}$ by the position of $X_{0,0}^{0,0}$. Note that $\mathbf{L}^{i',j'}$ and $\mathbf{G}^{i',j'}$ or $\mathbf{L}^{i',j'}$ and $\mathbf{K}^{i',j'}$ will not exist if $i' = 0$ or $j' = 0$. Further, $\mathbf{X}^{i',j'} \diamond \mathbf{Z}$ can be achieved through evaluating four sub-correlations, $d_L^{i',j'}$, $d_G^{i',j'}$, $d_K^{i',j'}$, and $d_J^{i',j'}$, as expressed in Eq.(3). In the following, we will explain how CCIM works with $\mathbf{X}^{1,1} \diamond \mathbf{Z}$.

$\mathbf{X}^{1,1}$, which is shown in Fig.2(a), is generated with $i' = j' = 1$, and $m = n = 3$. $\mathbf{X}^{1,1}$ can be divided into four parts, $\mathbf{L}^{1,1}$, $\mathbf{G}^{1,1}$, $\mathbf{K}^{1,1}$, and $\mathbf{J}^{1,1}$. Fig.2(b) presents their original positions in $\mathbf{X}^{0,0}$. According to Eq.(3), we have

$$\mathbf{X}^{1,1} \diamond \mathbf{Z}(i'', j'') = \langle \mathbf{L}^{1,1}, \mathbf{z}_{L;i'',j''}^{1,1} \rangle + \langle \mathbf{G}^{1,1}, \mathbf{z}_{G;i'',j''}^{1,1} \rangle + \\ \langle \mathbf{K}^{1,1}, \mathbf{z}_{K;i'',j''}^{1,1} \rangle + \langle \mathbf{J}^{1,1}, \mathbf{z}_{J;i'',j''}^{1,1} \rangle,$$

where $\mathbf{L}^{1,1} = (9)$, $\mathbf{z}_{L;i'',j''}^{1,1} = (Z_{i'',j''})$, $\mathbf{G}^{1,1} = (7, 8)$, $\mathbf{z}_{G;i'',j''}^{1,1} = (Z_{i'',j''+1}, Z_{i'',j''+2})$, $\mathbf{K}^{1,1} = (3, 6)$, $\mathbf{z}_{K;i'',j''}^{1,1} = (Z_{i''+1,j''}, Z_{i''+2,j''})$, $\mathbf{J}^{1,1} = (1, 2, 4, 5)$, and $\mathbf{z}_{J;i'',j''}^{1,1} = (Z_{i''+1,j''+1}, Z_{i''+1,j''+2}, Z_{i''+2,j''+1}, Z_{i''+2,j''+2})$, and

$$\mathbf{X}^{1,1} \diamond \mathbf{Z} = \mathbf{L}^{1,1} \diamond \mathbf{Z}_L^{1,1} + \mathbf{G}^{1,1} \diamond \mathbf{Z}_G^{1,1} + \mathbf{K}^{1,1} \diamond \mathbf{Z}_K^{1,1} + \mathbf{J}^{1,1} \diamond \mathbf{Z}_J^{1,1},$$

where $\mathbf{Z}_L^{1,1} = \mathbf{Z}((0,0), (2,2))$, $\mathbf{Z}_G^{1,1} = \mathbf{Z}((0,1), (2,4))$, $\mathbf{Z}_K^{1,1} = \mathbf{Z}((1,0), (4,2))$, and $\mathbf{Z}_J^{1,1} = \mathbf{Z}((1,1), (4,4))$.

While $(i'', j'') = (0, 0)$, we have $\mathbf{z}_{L;0,0}^{1,1} = (a)$, $\mathbf{z}_{G;0,0}^{1,1} = (b, c)$, $\mathbf{z}_{K;0,0}^{1,1} = (f, k)$, $\mathbf{z}_{J;0,0}^{1,1} = (g, h, l, m)$, and $\langle \mathbf{L}^{1,1}, \mathbf{z}_{L;0,0}^{1,1} \rangle = 9a$. According to Fig.1(c), $\langle \mathbf{L}^{1,1}, \mathbf{z}_{L;0,0}^{1,1} \rangle = B_{3,3}^{2,2}$ which is marked by the pink block in $\mathbf{B}^{2,2}$ of Fig.1(c)².

It can be seen from Fig.1(c) that while (i'', j'') traverses all elements of $S = \{(i'', j'') | i'' = 0, 1, 2; j'' = 0, 1, 2\}$, $\langle \mathbf{L}^{1,1}, \mathbf{z}_{L;i'',j''}^{1,1} \rangle$ produces nine values, $9a, 9b, 9c, 9f, 9g, 9h, 9k, 9l$, and $9m$. Suppose the 0th and 4th rows are adjacent, and the 0th and 4th columns are adjacent too in $\mathbf{B}^{i,j}$'s. Then, the 9 values consist of a sub-matrix of $\mathbf{B}^{2,2}$, which is marked by a red bounding box in $\mathbf{B}^{2,2}$ of Fig.1(c). Set $\mathbf{S}_L^{1,1} = \mathbf{B}^{2,2}$. Cyclically shifting the marked sub-matrix to the top left corner of $\mathbf{S}_L^{1,1}$, we achieve

$$\mathbf{L}^{1,1} \diamond \mathbf{Z}_L^{1,1} = \mathbf{S}_L^{1,1}((0,0), (2,2)).$$

Similarly, while $(i'', j'') = (0, 0)$, $\langle \mathbf{G}^{1,1}, \mathbf{z}_{G;0,0}^{1,1} \rangle = 7b + 8c = B_{3,1}^{2,0} + B_{3,1}^{2,1}$, $B_{3,1}^{2,0}$ and $B_{3,1}^{2,1}$ are marked with two green blocks in $\mathbf{B}^{2,0}$ and $\mathbf{B}^{2,1}$ of Fig.1(c). While (i'', j'') traverses all elements of $S = \{(i'', j'') | i'' = 0, 1, 2; j'' = 0, 1, 2\}$, $\langle \mathbf{G}^{1,1}, \mathbf{z}_{G;i'',j''}^{1,1} \rangle$ produces nine values, $7b + 8c, 7c + 8d, 7d + 8e, 7g + 8h, 7h + 8i, 7i + 8j, 7l + 8m, 7m + 8n$, and $7n + 8o$. Observe the sub-matrices of $\mathbf{B}^{2,0}$ and $\mathbf{B}^{2,1}$ which are marked by two red bounding boxes in $\mathbf{B}^{2,0}$ and $\mathbf{B}^{2,1}$ of Fig.1(c). These nine values are just the elements of the summation of the two sub-matrices. Cyclically shifting the marked sub-matrices to the top left corners of $\mathbf{B}^{2,0}$ and $\mathbf{B}^{2,1}$, and setting $\mathbf{S}_G^{1,1} = \mathbf{B}^{2,0} + \mathbf{B}^{2,1}$, we achieve

$$\mathbf{G}^{1,1} \diamond \mathbf{Z}_G^{1,1} = \mathbf{S}_G^{1,1}((0,0), (2,2)).$$

Similarly, while $(i'', j'') = (0, 0)$, $\langle \mathbf{K}^{1,1}, \mathbf{z}_{K;0,0}^{1,1} \rangle = 3f + 6k = B_{1,3}^{0,2} + B_{1,3}^{1,2}$, $B_{1,3}^{0,2}$ and $B_{1,3}^{1,2}$ are marked with two yellow blocks in $\mathbf{B}^{0,2}$ and $\mathbf{B}^{1,2}$ of Fig.1(c). While (i'', j'') traverses all elements of $S = \{(i'', j'') | i'' = 0, 1, 2; j'' = 0, 1, 2\}$, $\langle \mathbf{K}^{1,1}, \mathbf{z}_{K;i'',j''}^{1,1} \rangle$ produces nine values, $3f + 6k, 3g + 6l, 3h + 6m, 3k + 6p, 3l + 6q, 3m + 6r, 3p + 6u, 3q + 6w$, and $3r + 6x$. Observe the sub-matrices of $\mathbf{B}^{0,2}$ and $\mathbf{B}^{1,2}$ which are marked by two red bounding boxes in $\mathbf{B}^{0,2}$ and $\mathbf{B}^{1,2}$ of Fig.1(c). These nine values are just the elements of the summation of the two sub-matrices. Cyclically shifting the marked sub-matrices to the top left corners of $\mathbf{B}^{0,2}$ and $\mathbf{B}^{1,2}$, and setting $\mathbf{S}_K^{1,1} = \mathbf{B}^{0,2} + \mathbf{B}^{1,2}$, we achieve

$$\mathbf{K}^{1,1} \diamond \mathbf{Z}_K^{1,1} = \mathbf{S}_K^{1,1}((0,0), (2,2)).$$

Similarly, while $(i'', j'') = (0, 0)$, $\langle \mathbf{J}^{1,1}, \mathbf{z}_{J;0,0}^{1,1} \rangle = 1g + 2h + 4l + 5m = B_{1,1}^{0,0} + B_{1,1}^{0,1} + B_{1,1}^{1,0} + B_{1,1}^{1,1}$, $B_{1,1}^{0,0}$, $B_{1,1}^{0,1}$, $B_{1,1}^{1,0}$, and $B_{1,1}^{1,1}$ are marked with four blue blocks in $\mathbf{B}^{0,0}$, $\mathbf{B}^{0,1}$, $\mathbf{B}^{1,0}$ and $\mathbf{B}^{1,1}$ of Fig.1(c). While (i'', j'') traverses all elements of $S = \{(i'', j'') | i'' = 0, 1, 2; j'' = 0, 1, 2\}$, $\langle \mathbf{J}^{1,1}, \mathbf{z}_{J;i'',j''}^{1,1} \rangle$ produces nine values, $1g + 2h + 4l + 5m, 1h + 2i + 4m + 5n, 1i + 2j + 4n + 5o, 1l + 2m + 4q + 5r, 1m + 2n + 4r + 5s, 1n + 2o + 4s + 5t, 1q + 2r + 4w + 5x, 1r + 2s + 4x + 5y$, and $1s + 2t + 4y + 5z$. Observe the sub-matrices of $\mathbf{B}^{0,0}$, $\mathbf{B}^{0,1}$, $\mathbf{B}^{1,0}$ and $\mathbf{B}^{1,1}$ which are marked by four red bounding boxes in $\mathbf{B}^{0,0}$, $\mathbf{B}^{0,1}$, $\mathbf{B}^{1,0}$ and $\mathbf{B}^{1,1}$ of Fig.1(c). These nine values are just the elements of the summation of the four sub-matrices. Cyclically shifting the marked sub-matrices to the top left corners of $\mathbf{B}^{0,0}$, $\mathbf{B}^{0,1}$, $\mathbf{B}^{1,0}$ and $\mathbf{B}^{1,1}$, and setting $\mathbf{S}_J^{1,1} = \mathbf{B}^{0,0} + \mathbf{B}^{0,1} + \mathbf{B}^{1,0} + \mathbf{B}^{1,1}$, we achieve

$$\mathbf{J}^{1,1} \diamond \mathbf{Z}_J^{1,1} = \mathbf{S}_J^{1,1}((0,0), (2,2)).$$

²In fact, according to the example in Sec.2.1 and Fig.2(b), the item with 9 as its factor should be shifted to the position marked with the pink circle in Fig.2(a). Therefore, $\langle \mathbf{L}^{1,1}, \mathbf{z}_{L;0,0}^{1,1} \rangle = 9a$ should be shifted to the position marked with the pink circle. Then, $9a$ is shifted to $(3,3)$ of $\mathbf{B}^{2,2}$ according to the construction of $\mathbf{B}^{2,2}$ through the cyclic-shift of $\mathbf{A}^{2,2}$. Similarly, the items produced by $\langle \mathbf{G}^{1,1}, \mathbf{z}_{G;0,0}^{1,1} \rangle$ and $\langle \mathbf{K}^{1,1}, \mathbf{z}_{K;0,0}^{1,1} \rangle$ should be shifted to the positions marked with green and yellow circles, respectively. According to the same reason, these items are shifted to the $(3,1)$'s of $\mathbf{B}^{2,0}$ and $\mathbf{B}^{2,1}$, and the $(1,3)$'s of $\mathbf{B}^{0,2}$ and $\mathbf{B}^{1,2}$, respectively. Finally, the four items generated by $\langle \mathbf{J}^{1,1}, \mathbf{z}_{J;0,0}^{1,1} \rangle$ are shifted to the position marked by the blue circle, i.e., $(1,1)$'s of $\mathbf{B}^{0,0}$, $\mathbf{B}^{0,1}$, $\mathbf{B}^{1,0}$ and $\mathbf{B}^{1,1}$.

Finally, set $\mathbf{C}^{1,1} = \sum_{W \in \{L, G, K, J\}} \mathbf{S}_W^{1,1}$, then,

$$\mathbf{X}^{1,1} \diamond \mathbf{Z} = \mathbf{C}^{1,1}((0, 0), (2, 2)).$$

2.3 Design Integral Matrix to Accelerate Correlation

According to the beginning of this section, $X_{i,j}^{0,0}$ is the common factor of all elements of $\mathbf{B}^{i,j}$. Replacing $X_{i,j}^{0,0}$ by $\mathbf{B}^{i,j}$ in $\mathbf{X}^{0,0}$, we achieve

$$\mathbf{T} = \begin{bmatrix} \mathbf{B}^{0,0} & \mathbf{B}^{0,1} & \mathbf{B}^{0,2} \\ \mathbf{B}^{1,0} & \mathbf{B}^{1,1} & \mathbf{B}^{1,2} \\ \mathbf{B}^{2,0} & \mathbf{B}^{2,1} & \mathbf{B}^{2,2} \end{bmatrix}.$$

Based on the examples given in Secs.2.1 and 2.2, it is necessary to evaluate $\sum_{i=0}^2 \sum_{j=0}^2 \mathbf{B}^{i,j}$ to achieve $\mathbf{X}^{0,0} \diamond \mathbf{Z}$, and to evaluate $\mathbf{B}^{2,2}$, $\mathbf{B}^{2,0} + \mathbf{B}^{2,1}$, $\mathbf{B}^{0,2} + \mathbf{B}^{1,2}$, and $\mathbf{B}^{0,0} + \mathbf{B}^{0,1} + \mathbf{B}^{1,0} + \mathbf{B}^{1,1}$ to obtain $\mathbf{X}^{1,1} \diamond \mathbf{Z}$. In order to eliminate unnecessary additions, the three summations of matrices, which are necessary for evaluating $\mathbf{X}^{1,1} \diamond \mathbf{Z}$, should be reused to evaluate $\mathbf{X}^{0,0} \diamond \mathbf{Z}$. Notice that each above summation of $\mathbf{B}^{i,j}$'s is involved with the summation of the elements of $\mathbf{T}$'s sub-matrices. Constructing the integral matrix $\mathbf{M}_\mathbf{I}$ with CCIM, we have

$$\mathbf{M}_\mathbf{I} = \begin{bmatrix} \mathbf{M}_{0,0} & \mathbf{M}_{0,1} & \mathbf{M}_{0,2} \\ \mathbf{M}_{1,0} & \mathbf{M}_{1,1} & \mathbf{M}_{1,2} \\ \mathbf{M}_{2,0} & \mathbf{M}_{2,1} & \mathbf{M}_{2,2} \end{bmatrix},$$

where $\mathbf{M}_{s,t} = \sum_{0 \leq i \leq s, 0 \leq j \leq t} \mathbf{B}^{i,j}$. Then, $\mathbf{M}_{2,2} = \sum_{i=0}^2 \sum_{j=0}^2 \mathbf{B}^{i,j}$, $\mathbf{M}_{2,2} - \mathbf{M}_{1,2} - \mathbf{M}_{2,1} + \mathbf{M}_{1,1} = \mathbf{B}^{2,2}$, $\mathbf{M}_{2,1} - \mathbf{M}_{1,1} = \mathbf{B}^{2,0} + \mathbf{B}^{2,1}$, $\mathbf{M}_{1,2} - \mathbf{M}_{1,1} = \mathbf{B}^{0,2} + \mathbf{B}^{1,2}$, and $\mathbf{M}_{1,1} = \mathbf{B}^{0,0} + \mathbf{B}^{0,1} + \mathbf{B}^{1,0} + \mathbf{B}^{1,1}$. It is seen that the five summations of $\mathbf{B}^{i,j}$'s are achieved within constant duration by means of $\mathbf{M}_\mathbf{I}$. This way, the unnecessary additions of $\mathbf{B}^{i,j}$'s are eliminated.

In general, because $\mathbf{W}^{i',j'}$ is a sub-matrix of $\mathbf{X}^{0,0}$, where $\mathbf{W} = \mathbf{L}, \mathbf{G}, \mathbf{K}, \mathbf{J}$, the $\mathbf{B}^{i,j}$'s with the elements of $\mathbf{W}$ as their common factors also constitute a $\mathbf{T}$'s sub-matrix. While evaluating $d_W^{i',j'}$, where $W \in \{L, G, K, J\}$, similar to the examples in Secs.2.1 and 2.2, it is necessary to evaluate the summation of all elements of the $\mathbf{T}$'s sub-matrix. For a given $\mathbf{X}^{0,0}$, while (i', j') traverses all possible pairs, *i.e.*, $\mathbf{X}^{i',j'}$ traverses all possible circulant correlation filters, the brute force evaluation of the summations will arise a great deal of unnecessary computation, decreasing the efficiency of correlations greatly especially when the size of base correlation filter is large. Therefore, it is a natural choice to construct the integral matrix, as described in CCIM, to calculate such summations without unnecessary additions.

3 Correctness of Algorithm 2 (CCIM)

Suppose $\mathbf{X}^{0,0} \in \mathbb{R}^{m \times n \times D}$ is the $m \times n \times D$ base correlation filter with $X_{i,j,d}^{0,0}$ as its elements, (i, j) is the spatial coordinate, d is the channel, circulant correlation filters

$$\mathbf{X}^{i',j'} = \mathbf{P}_m^{i'} \mathbf{X}^{0,0} \mathbf{Q}_n^{j'}, \quad i' = 0, \dots, m-1, \quad j' = 0, \dots, n-1, \quad (1)$$

where $\mathbf{P}_{l_1}$, $l_1 \in \{m, M\}$, and $\mathbf{Q}_{l_2}$, $l_2 \in \{n, N\}$, are the permutation matrices of $l_1 \times l_1$ and $l_2 \times l_2$, respectively [3],

$$\mathbf{P}_{l_1} = \begin{bmatrix} \mathbf{0}_{l_1-1}^\top & 1 \\ \mathbf{I}_{l_1-1} & \mathbf{0}_{l_1-1} \end{bmatrix}, \quad \mathbf{Q}_{l_2} = \begin{bmatrix} \mathbf{0}_{l_2-1} & \mathbf{I}_{l_2-1} \\ 1 & \mathbf{0}_{l_2-1}^\top \end{bmatrix}, \quad (2)$$

$\mathbf{P}_{l_1}^\rho$ and $\mathbf{Q}_{l_2}^\rho$ are the ρ^{th} power of $\mathbf{P}_{l_1}$ and $\mathbf{Q}_{l_2}$, respectively, $\mathbf{0}_{l-1}$ is $(l-1) \times 1$ vector of all zeros, $\mathbf{I}_{l-1}$ is $(l-1) \times (l-1)$ identity matrix, $l \in \{l_1, l_2\}$. Note that the matrix operation works only on the spatial coordinate in Eq.(1). $\mathbf{Z} \in \mathbb{R}^{M \times N \times D}$ is the $M \times N \times D$ signal matrix with $Z_{i'',j'',d}$ as its elements. $\mathbf{Z}_d \in \mathbb{R}^{M \times N}$ is its d^{th} channel matrix.

Let $M \times N$ minimal evaluation matrix $\mathbf{A}^{i,j} = \sum_{d=0}^{D-1} X_{i,j,d}^{0,0} \cdot \mathbf{Z}_d$, where $i = 0, \dots, m-1, j = 0, \dots, n-1$, and $M \times N$ fundamental matrix $\mathbf{B}^{i,j} = \mathbf{P}_M^{-i} \mathbf{A}^{i,j} \mathbf{Q}_N^{-j}$. Note that $\mathbf{B}^{i,j}$'s are generated through cyclic-shifts of $\mathbf{A}^{i,j}$. Then, $X_{i,j}^{0,0}$ only appears in $\mathbf{A}^{i,j}$ and $\mathbf{B}^{i,j}$ as a common factor of their elements. The sets of whole $\mathbf{A}^{i,j}$'s and $\mathbf{B}^{i,j}$'s are denoted as $\mathfrak{A}$ and $\mathfrak{B}$, respectively.

In this appendix, we will prove that CCIM will produce $\mathbf{X}^{i',j'} \diamond \mathbf{Z}$ correctly. To simplify the mathematical expression, we only present our proof in the case of $D = 1$. It is straightforward to extend the proof to the general case of $D > 1$.

In general, as shown in Fig.2(c), given i' and j' , $\mathbf{X}^{i',j'}$ can always be divided into four parts, $\mathbf{L}^{i',j'}$, $\mathbf{G}^{i',j'}$, $\mathbf{K}^{i',j'}$, and $\mathbf{J}^{i',j'}$ by the position of $X_{0,0}^{0,0}$. $\mathbf{L}^{i',j'}$, $\mathbf{G}^{i',j'}$, $\mathbf{K}^{i',j'}$, and $\mathbf{J}^{i',j'}$ are $i' \times j'$, $i' \times (n-j')$, $(m-i') \times j'$, and $(m-i') \times (n-j')$, respectively, and their top left elements are $L_{0,0}^{i',j'} = X_{m-i',n-j'}^{0,0}$, $G_{0,0}^{i',j'} = X_{m-i',0}^{0,0}$, $K_{0,0}^{i',j'} = X_{0,n-j'}^{0,0}$, and $J_{0,0}^{i',j'} = X_{0,0}^{0,0}$, respectively. Note that $\mathbf{L}^{i',j'}$ and $\mathbf{G}^{i',j'}$ or $\mathbf{L}^{i',j'}$ and $\mathbf{K}^{i',j'}$ will not exist if $i' = 0$ or $j' = 0$.

Let

$$\begin{aligned} \mathbf{X}^{i',j'} \diamond \mathbf{Z}(i'', j'') &= \sum_{s=0}^{m-1} \sum_{t=0}^{n-1} X_{s,t}^{i',j'} \cdot Z_{i''+s, j''+t} \\ &= d_L^{i',j'}(i'', j'') + d_G^{i',j'}(i'', j'') + d_K^{i',j'}(i'', j'') + d_J^{i',j'}(i'', j''), \end{aligned} \quad (3)$$

where

$$\begin{aligned} d_L^{i',j'}(i'', j'') &= \sum_{s=0}^{i'-1} \sum_{t=0}^{j'-1} X_{s,t}^{i',j'} \cdot Z_{i''+s, j''+t} \\ &= \langle \mathbf{L}^{i',j'}, \mathbf{Z}((i'', j''), (i'' + i' - 1, j'' + j' - 1)) \rangle, \\ d_G^{i',j'}(i'', j'') &= \sum_{s=0}^{i'-1} \sum_{t=j'}^{n-1} X_{s,t}^{i',j'} \cdot Z_{i''+s, j''+t} \\ &= \langle \mathbf{G}^{i',j'}, \mathbf{Z}((i'', j'' + j'), (i'' + i' - 1, j'' + n - 1)) \rangle, \\ d_K^{i',j'}(i'', j'') &= \sum_{s=i'}^{m-1} \sum_{t=0}^{j'-1} X_{s,t}^{i',j'} \cdot Z_{i''+s, j''+t} \\ &= \langle \mathbf{K}^{i',j'}, \mathbf{Z}((i'' + i', j''), (i'' + m - 1, j'' + j' - 1)) \rangle, \\ d_J^{i',j'}(i'', j'') &= \sum_{s=i'}^{m-1} \sum_{t=j'}^{n-1} X_{s,t}^{i',j'} \cdot Z_{i''+s, j''+t} \\ &= \langle \mathbf{J}^{i',j'}, \mathbf{Z}((i'' + i', j'' + j'), (i'' + m - 1, j'' + n - 1)) \rangle, \end{aligned}$$

$i'' = 0, \dots, M - m$, and $j'' = 0, \dots, N - n$.

3.1 Evaluation of $d_L^{i',j'}(i'', j'')$

While evaluating $d_L^{i',j'}(i'', j'')$, $L_{0,0}^{i',j'}$ and $Z_{i'',j''}$ multiply. According to the construction of $\mathfrak{A}$ and $\mathfrak{B}$, we have

$$L_{0,0}^{i',j'} \cdot Z_{i'',j''} = A_{i'',j''}^{m-i',n-j'}.$$

Because

$$\mathbf{B}^{m-i',n-j'} = \mathbf{P}_M^{-(m-i')} \mathbf{A}^{m-i',n-j'} \mathbf{Q}_N^{-(n-j')},$$

therefore

$$L_{0,0}^{i',j'} \cdot Z_{i'',j''} = B_{(i''-(m-i')) \bmod M, (j''-(n-j')) \bmod N}^{m-i',n-j'}.$$

On the other hand, $L_{p,q}^{i',j'} = X_{m-i'+p, n-j'+q}^{0,0}$, where $p = 0, \dots, i' - 1$ and $q = 0, \dots, j' - 1$. It only appears in $\mathbf{B}^{m-i'+p, n-j'+q}$, and

$$L_{p,q}^{i',j'} \cdot Z_{i''+p, j''+q} = A_{i''+p, j''+q}^{m-i'+p, n-j'+q}.$$

We also have

$$\begin{aligned}\mathbf{B}^{m-i'+p, n-j'+q} &= \mathbf{P}_M^{-(m-i'+p)} \mathbf{A}^{m-i'+p, n-j'+q} \mathbf{Q}_N^{-(n-j'+q)} \\ &= \mathbf{P}_M^{-(m-i')} \mathbf{P}_M^{-p} \mathbf{A}^{m-i'+p, n-j'+q} \mathbf{Q}_N^{-q} \mathbf{Q}_N^{-(n-j')}.\end{aligned}$$

Let

$$\mathbf{A}^{(L)} = \mathbf{P}_M^{-p} \mathbf{A}^{m-i'+p, n-j'+q} \mathbf{Q}_N^{-q}.$$

Then, $A_{i'', j''}^{(L)} = A_{i''+p, j''+q}^{m-i'+p, n-j'+q}$. That is, $L_{p, q}^{i', j'} \cdot Z_{i''+p, j''+q} = A_{i'', j''}^{(L)}$. Because

$$\mathbf{B}^{m-i'+p, n-j'+q} = \mathbf{P}_M^{-(m-i')} \mathbf{A}^{(L)} \mathbf{Q}_N^{-(n-j')},$$

therefore

$$L_{p, q}^{i', j'} \cdot Z_{i''+p, j''+q} = B_{(i''-(m-i')) \bmod M, (j''-(n-j')) \bmod N}^{m-i'+p, n-j'+q}.$$

That is, $L_{0, 0}^{i', j'} \cdot Z_{i'', j''}$ and $L_{p, q}^{i', j'} \cdot Z_{i''+p, j''+q}$ possess the same subscripts, although they belong to different fundamental matrices.

It is seen from Fig.2(c) that the top left and bottom right elements of $\mathbf{L}^{i', j'}$ are $(X_{m-i', n-j'}^{0, 0}, X_{m-1, n-1}^{0, 0})$. Let $\mathbf{S}^{L, i', j'} = \sum_{i=m-i'}^{m-1} \sum_{j=n-j'}^{n-1} \mathbf{B}^{i, j}$. According to the construction of integral matrix, $\mathbf{S}^{L, i', j'}$ can be evaluated within constant time as

$$\mathbf{S}^{L, i', j'} = \mathbf{M}_{m-1, n-1} - \mathbf{M}_{m-1, n-j'-1} - \mathbf{M}_{m-i'-1, n-1} + \mathbf{M}_{m-i'-1, n-j'-1}.$$

Then, we have

$$d_L^{i', j'}(i'', j'') = S_{(i''-(m-i')) \bmod M, (j''-(n-j')) \bmod N}^{L, i', j'}.$$

3.2 Evaluation of $d_G^{i', j'}(i'', j'')$

While evaluating $d_G^{i', j'}(i'', j'')$, $G_{0, 0}^{i', j'}$ and $Z_{i'', j''+j'}$ multiply. According to the construction of $\mathfrak{A}$ and $\mathfrak{B}$, we have

$$G_{0, 0}^{i', j'} \cdot Z_{i'', j''+j'} = A_{i'', j''+j'}^{m-i', 0}.$$

Because

$$\mathbf{B}^{m-i', 0} = \mathbf{P}_M^{-(m-i')} \mathbf{A}^{m-i', 0},$$

therefore

$$G_{0, 0}^{i', j'} \cdot Z_{i'', j''+j'} = B_{(i''-(m-i')) \bmod M, j''+j'}^{m-i', 0}.$$

On the other hand, $G_{p, q}^{i', j'} = X_{m-i'+p, q}^{0, 0}$, where $p = 0, \dots, i'-1$ and $q = 0, \dots, n-j'-1$. It only appears in $\mathbf{B}^{m-i'+p, q}$, and

$$G_{p, q}^{i', j'} \cdot Z_{i''+p, j''+j'+q} = A_{i''+p, j''+j'+q}^{m-i'+p, q}.$$

We also have

$$\begin{aligned}\mathbf{B}^{m-i'+p, q} &= \mathbf{P}_M^{-(m-i'+p)} \mathbf{A}^{m-i'+p, q} \mathbf{Q}_N^{-q} \\ &= \mathbf{P}_M^{-(m-i')} \mathbf{P}_M^{-p} \mathbf{A}^{m-i'+p, q} \mathbf{Q}_N^{-q}.\end{aligned}$$

Let

$$\mathbf{A}^{(G)} = \mathbf{P}_M^{-p} \mathbf{A}^{m-i'+p, q} \mathbf{Q}_N^{-q}.$$

Then, $A_{i'', j''+j'}^{(G)} = A_{i''+p, j''+j'+q}^{m-i'+p, q}$. That is, $G_{p, q}^{i', j'} \cdot Z_{i''+p, j''+j'+q} = A_{i'', j''+j'}^{(G)}$. Because

$$\mathbf{B}^{m-i'+p, q} = \mathbf{P}_M^{-(m-i')} \mathbf{A}^{(G)},$$

therefore

$$G_{p, q}^{i', j'} \cdot Z_{i''+p, j''+j'+q} = B_{(i''-(m-i')) \bmod M, j''+j'}^{m-i'+p, q}.$$

That is, $G_{0, 0}^{i', j'} \cdot Z_{i'', j''+j'}$ and $L_{p, q}^{i', j'} \cdot Z_{i''+p, j''+j'+q}$ possess the same subscripts, although they belong to different fundamental matrices.

It is seen from Fig.2(c) that the top left and bottom right elements of $\mathbf{G}^{i',j'}$ are $(X_{m-i',0}^{0,0}, X_{m-1,n-j'-1}^{0,0})$. Let $\mathbf{S}^{G,i',j'} = \sum_{i=m-i'}^{m-1} \sum_{j=0}^{n-j'-1} \mathbf{B}^{i,j}$. According to the construction of integral matrix, $\mathbf{S}^{G,i',j'}$ can be evaluated within constant time as

$$\mathbf{S}^{G,i',j'} = \mathbf{M}_{m-1,n-j'-1} - \mathbf{M}_{m-i'-1,n-j'-1}.$$

Then, we have

$$d_{\mathbf{G}}^{i',j'}(i'',j'') = S_{(i''-(m-i')) \bmod M, j''+j'}^{G,i',j'}.$$

3.3 Evaluation of $d_K^{i',j'}(i'',j'')$

While evaluating $d_K^{i',j'}(i'',j'')$, $K_{0,0}^{i',j'}$ and $Z_{i''+i',j''}$ multiply. According to the construction of $\mathfrak{A}$ and $\mathfrak{B}$, we have

$$K_{0,0}^{i',j'} \cdot Z_{i''+i',j''} = A_{i''+i',j''}^{0,n-j'}.$$

Because

$$\mathbf{B}^{0,n-j'} = \mathbf{A}^{0,n-j'} \mathbf{Q}_N^{-(n-j')},$$

therefore

$$K_{0,0}^{i',j'} \cdot Z_{i''+i',j''} = B_{i''+i',(j''-(n-j')) \bmod N}^{0,n-j'}.$$

On the other hand, $K_{p,q}^{i',j'} = X_{p,n-j'+q}^{0,0}$, where $p = 0, \dots, m-i'-1$ and $q = 0, \dots, j'-1$. It only appears in $\mathbf{B}^{p,n-j'+q}$, and

$$K_{p,q}^{i',j'} \cdot Z_{i''+i'+p,j''+q} = A_{i''+i'+p,j''+q}^{p,n-j'+q}.$$

We also have

$$\begin{aligned} \mathbf{B}^{p,n-j'+q} &= \mathbf{P}_M^{-p} \mathbf{A}^{p,n-j'+q} \mathbf{Q}_N^{-(n-j'+q)} \\ &= \mathbf{P}_M^{-p} \mathbf{A}^{p,n-j'+q} \mathbf{Q}_N^{-q} \mathbf{Q}_N^{-(n-j')}. \end{aligned}$$

Let

$$\mathbf{A}^{(K)} = \mathbf{P}_M^{-p} \mathbf{A}^{p,n-j'+q} \mathbf{Q}_N^{-q}.$$

Then, $A_{i''+i',j''}^{(K)} = A_{i''+i'+p,j''+q}^{p,n-j'+q}$. That is, $K_{p,q}^{i',j'} \cdot Z_{i''+i'+p,j''+q} = A_{i''+i',j''}^{(K)}$. Because

$$\mathbf{B}^{p,n-j'+q} = \mathbf{A}^{(K)} \mathbf{Q}_N^{-(n-j')},$$

therefore

$$K_{p,q}^{i',j'} \cdot Z_{i''+i'+p,j''+q} = B_{i''+i',(j''-(n-j')) \bmod N}^{p,n-j'+q}.$$

That is, $K_{0,0}^{i',j'} \cdot Z_{i''+i',j''}$ and $K_{p,q}^{i',j'} \cdot Z_{i''+i'+p,j''+q}$ possess the same subscripts, although they belong to different fundamental matrices.

It is seen from Fig.2(c) that the top left and bottom right elements of $\mathbf{K}^{i',j'}$ are $(X_{0,n-j'}^{0,0}, X_{m-i'-1,n-1}^{0,0})$. Let $\mathbf{S}^{K,i',j'} = \sum_{i=0}^{m-i'-1} \sum_{j=n-j'}^{n-1} \mathbf{B}^{i,j}$. According to the construction of integral matrix, $\mathbf{S}^{K,i',j'}$ can be evaluated within constant time as

$$\mathbf{S}^{K,i',j'} = \mathbf{M}_{m-i'-1,n-1} - \mathbf{M}_{m-i'-1,n-j'-1}.$$

Then, we have

$$d_K^{i',j'}(i'',j'') = S_{i''+i',(j''-(n-j')) \bmod N}^{K,i',j'}.$$

3.4 Evaluation of $d_J^{i',j'}(i'',j'')$

While evaluating $d_J^{i',j'}(i'',j'')$, $J_{0,0}^{i',j'}$ and $Z_{i''+i',j''+j'}$ multiply. According to the construction of $\mathfrak{A}$ and $\mathfrak{B}$, we have $J_{0,0}^{i',j'} \cdot Z_{i''+i',j''+j'} = A_{i''+i',j''+j'}^{0,0}$. And $\mathbf{B}^{0,0} = \mathbf{A}^{0,0}$. Therefore, $J_{0,0}^{i',j'} \cdot Z_{i''+i',j''+j'} = B_{i''+i',j''+j'}^{0,0}$.

On the other hand, $J_{p,q}^{i',j'} = X_{p,q}^{0,0}$, where $p = 0, \dots, m - i' - 1$ and $q = 0, \dots, n - j' - 1$. It only appears in $\mathbf{B}^{p,q}$, and

$$J_{p,q}^{i',j'} \cdot Z_{i''+i'+p,j''+j'+q} = A_{i''+i'+p,j''+j'+q}^{p,q}.$$

We also have $\mathbf{B}^{p,q} = \mathbf{P}_M^{-p} \mathbf{A}^{p,q} \mathbf{Q}_N^{-q}$. Then, $B_{i''+i',j''+j'}^{p,q} = A_{i''+i'+p,j''+j'+q}^{p,q}$. Therefore

$$J_{p,q}^{i',j'} \cdot Z_{i''+i'+p,j''+j'+q} = B_{i''+i',j''+j'}^{p,q}.$$

That is, $J_{0,0}^{i',j'} \cdot Z_{i''+i',j''+j'}$ and $J_{p,q}^{i',j'} \cdot Z_{i''+i'+p,j''+j'+q}$ possess the same subscripts, although they belong to different fundamental matrices.

It is seen from Fig.2(c) that the top left and bottom right elements of $\mathbf{J}^{i',j'}$ are $(X_{0,0}^{0,0}, X_{m-i'-1,n-j'-1}^{0,0})$. Let $\mathbf{S}^{J,i',j'} = \sum_{i=0}^{m-i'-1} \sum_{j=0}^{n-j'-1} \mathbf{B}^{i,j}$. According to the construction of integral matrix, $\mathbf{S}^{J,i',j'}$ can be evaluated within constant time as

$$\mathbf{S}^{J,i',j'} = \mathbf{M}_{m-i'-1,n-j'-1}.$$

Then, we have

$$d_{J,i''}^{i',j'}(i'',j'') = S_{i''+i',j''+j'}^{J,i',j'}.$$

3.5 Correlation of $\mathbf{X}^{i',j'}$ and $\mathbf{Z}$

According to the proofs in Secs.3.1, 3.2, 3.3, and 3.4, we can get the $d_L^{i',j'}(i'',j'')$'s, $d_G^{i',j'}(i'',j'')$'s, $d_K^{i',j'}(i'',j'')$'s, and $d_J^{i',j'}(i'',j'')$'s for all (i'',j'') 's. Therefore, $\mathbf{S}^{L,i',j'}$, $\mathbf{S}^{G,i',j'}$, $\mathbf{S}^{K,i',j'}$, and $\mathbf{S}^{J,i',j'}$ include the correlation results between $\mathbf{L}^{i',j'}$, $\mathbf{G}^{i',j'}$, $\mathbf{K}^{i',j'}$, $\mathbf{J}^{i',j'}$ and their corresponding parts of $\mathbf{Z}$.

Let

$$\begin{aligned} \Delta_r^{(L)}(i'',j'') &= (i'' - (m - i')) \bmod M - i'', \\ \Delta_c^{(L)}(i'',j'') &= (j'' - (n - j')) \bmod N - j'', \\ \Delta_r^{(G)}(i'',j'') &= (i'' - (m - i')) \bmod M - i'', \\ \Delta_c^{(G)}(i'',j'') &= j', \\ \Delta_r^{(K)}(i'',j'') &= i', \\ \Delta_c^{(K)}(i'',j'') &= (j'' - (n - j')) \bmod N - j'', \\ \Delta_r^{(J)}(i'',j'') &= i', \\ \Delta_c^{(J)}(i'',j'') &= j', \end{aligned} \tag{4}$$

and

$$\begin{aligned} \mathbf{S}_L^{i',j'} &= \mathbf{P}_M^{-\Delta_r^{(L)}(i'',j'')} \mathbf{S}_{L,i',j'} \mathbf{Q}_N^{-\Delta_c^{(L)}(i'',j'')}, \\ \mathbf{S}_G^{i',j'} &= \mathbf{P}_M^{-\Delta_r^{(G)}(i'',j'')} \mathbf{S}_{G,i',j'} \mathbf{Q}_N^{-\Delta_c^{(G)}(i'',j'')}, \\ \mathbf{S}_K^{i',j'} &= \mathbf{P}_M^{-\Delta_r^{(K)}(i'',j'')} \mathbf{S}_{K,i',j'} \mathbf{Q}_N^{-\Delta_c^{(K)}(i'',j'')}, \\ \mathbf{S}_J^{i',j'} &= \mathbf{P}_M^{-\Delta_r^{(J)}(i'',j'')} \mathbf{S}_{J,i',j'} \mathbf{Q}_N^{-\Delta_c^{(J)}(i'',j'')}. \end{aligned}$$

Then, $d_L^{i',j'}(i'',j'')$, $d_G^{i',j'}(i'',j'')$, $d_K^{i',j'}(i'',j'')$, and $d_J^{i',j'}(i'',j'')$ are shifted to $S_{L;i'',j''}^{i',j'}$, $S_{G;i'',j''}^{i',j'}$, $S_{K;i'',j''}^{i',j'}$, and $S_{J;i'',j''}^{i',j'}$, respectively, where $S_{L;i'',j''}^{i',j'}$, $S_{G;i'',j''}^{i',j'}$, $S_{K;i'',j''}^{i',j'}$, and $S_{J;i'',j''}^{i',j'}$ are the elements of $\mathbf{S}_L^{i',j'}$, $\mathbf{S}_G^{i',j'}$, $\mathbf{S}_K^{i',j'}$, and $\mathbf{S}_J^{i',j'}$, respectively, with (i'',j'') as their common subscripts.

Now we prove $\mathbf{P}_M^{-\Delta_r^{(L)}(i'',j'')} = \mathbf{P}_M^{-\Delta_r^{(L)}(0,0)}$.

$$\because 0 \leq i' \leq m - 1, \therefore -m \leq -(m - i') \leq -1 \leq 0. \because M \geq m, \therefore 0 \leq M - m \leq M - (m - i') \leq M - 1. \therefore \mathbf{P}_M^{-\Delta_r^{(L)}(0,0)} = \mathbf{P}_M^{-(M-(m-i'))}.$$

Now we split the set $\{0, \dots, M - m\}$ into two parts, $S_l = \{0, \dots, m - i' - 1\}$ and $S_t = \{m - i', \dots, M - m\}$. If $i'' \in S_l$, then $0 \leq i'' \leq m - i' - 1$. $\therefore -M \leq -(m - i') \leq i'' - (m - i') \leq -1$.

In this case, $\Delta_r^{(L)}(i'', j'') = (i'' - (m - i')) \bmod M - i'' = M + i'' - (m - i') - i'' = M - (m - i')$. If $i'' \in S_t$, then $m - i' \leq i'' \leq M - m$. $\therefore 0 \leq i'' - (m - i') \leq M - m - (m - i') < M$. In this case, $\Delta_r^{(L)}(i'', j'') = (i'' - (m - i')) \bmod M - i'' = i'' - (m - i') - i'' = -(m - i')$. $\therefore \mathbf{P}_M^{-(M-(m-i'))} = \mathbf{P}_M^{m-i'}$, $\therefore \mathbf{P}_M^{-\Delta_r^{(L)}(i'', j'')} = \mathbf{P}_M^{-(M-(m-i'))}$, $\therefore \mathbf{P}_M^{-\Delta_r^{(L)}(i'', j'')} = \mathbf{P}_M^{-\Delta_r^{(L)}(0,0)}$. According to this proof, we also have $\mathbf{P}_M^{-\Delta_r^{(G)}(i'', j'')} = \mathbf{P}_M^{-\Delta_r^{(G)}(0,0)}$.

Similarly, $\mathbf{Q}_N^{-\Delta_c^{(L)}(i'', j'')} = \mathbf{Q}_N^{-\Delta_c^{(L)}(0,0)}$ and $\mathbf{Q}_N^{-\Delta_c^{(K)}(i'', j'')} = \mathbf{Q}_N^{-\Delta_c^{(K)}(0,0)}$.

Consequently, all increments in Eq.(4) are not related to (i'', j'') , and are constant, given (i', j') . All elements of $\mathbf{S}^{L, i', j'}((M - m + i', N - n + j'), ((M - 2m + i') \bmod M, (N - 2n + j') \bmod N))$, those of $\mathbf{S}^{G, i', j'}((M - m + i', j'), ((M - 2m + i') \bmod M, N - n + j'))$, those of $\mathbf{S}^{K, i', j'}((i', N - n + j'), (M - m + i', (N - 2n + j') \bmod N))$, and those $\mathbf{S}^{J, i', j'}((i', j'), (M - m + i', N - n + j'))$ are shifted with the same numbers of rows and columns, respectively. Noticed that $d_L^{i', j'}(0, 0)$, $d_G^{i', j'}(0, 0)$, $d_K^{i', j'}(0, 0)$, and $d_J^{i', j'}(0, 0)$ are shifted to $S_{L;0,0}^{i', j'}$, $S_{G;0,0}^{i', j'}$, $S_{K;0,0}^{i', j'}$, and $S_{J;0,0}^{i', j'}$, respectively, therefore,

$$\begin{aligned} & \mathbf{S}_L^{i', j'}((0, 0), (M - m - 1, N - n - 1)) \\ &= \mathbf{S}^{L, i', j'}((M - m + i', N - n + j'), ((M - 2m + i') \bmod M, (N - 2n + j') \bmod N)), \\ & \mathbf{S}_G^{i', j'}((0, 0), (M - m - 1, N - n - 1)) \\ &= \mathbf{S}^{G, i', j'}((M - m + i', j'), ((M - 2m + i') \bmod M, N - n + j')), \\ & \mathbf{S}_K^{i', j'}((0, 0), (M - m - 1, N - n - 1)) \\ &= \mathbf{S}^{K, i', j'}((i', N - n + j'), (M - m + i', (N - 2n + j') \bmod N)), \\ & \mathbf{S}_J^{i', j'}((0, 0), (M - m - 1, N - n - 1)) \\ &= \mathbf{S}^{J, i', j'}((i', j'), (M - m + i', N - n + j')). \end{aligned}$$

Let $\mathbf{C}^{i', j'} = \mathbf{S}_L^{i', j'} + \mathbf{S}_G^{i', j'} + \mathbf{S}_K^{i', j'} + \mathbf{S}_J^{i', j'}$. Then, according to Eq.(3), $\mathbf{C}_{i'', j''}^{i', j'} = \mathbf{X}^{i', j'} \diamond \mathbf{Z}(i'', j'')$.

Consequently, the correlation of $\mathbf{X}^{i', j'}$ and $\mathbf{Z}$,

$$\mathbf{X}^{i', j'} \diamond \mathbf{Z}(i'', j'') = \{C_{i'', j''}^{i', j'} | i'' = 0, \dots, M - m - 1, j'' = 0, \dots, N - n - 1\},$$

or briefly,

$$\mathbf{X}^{i', j'} \diamond \mathbf{Z} = \mathbf{C}^{i', j'}((0, 0), (M - m - 1, N - n - 1)).$$

Q.E.D.

4 Other Experimental Results

We evaluate nBEKCF on VOT2016 dataset [8] which contains 60 sequences with substantial variations and challenges. We follow the VOT challenge protocol to compare trackers, where a tracker is re-initialized whenever its failure is observed. The evaluation criteria on VOT2016 are different from those on OTB-2013 and OTB-2015. On VOT2016, the accuracy and robustness of each tracker are evaluated in terms of the overlap ratios between bounding boxes and ground-truths and the number of re-initializations. In addition, VOT2016 mainly reports the expected average overlap (EAO) which integrates the accuracy and robustness of a tracker, and ranks trackers according to it, since the EAO score may provide an insight to the comprehensive performance of a tracker.

We compare our method against other top 9 trackers, SRDCF [2], DSST [1], NSAMF [8], deepMKCF [8], SAMF [7], KCF [6], ColorKCF [8], TGPR [4], and Struck [5], of VOT2016 challenge. Figs.3 shows the EAO ranking and accuracy-robustness plots, where the closer to the top-right corner of the plots a tracker is, the better the performance of the tracker is considered. It is seen that nBEKCF is ranked the second best in EAO. Although the EAO of nBEKCF is slightly less than that of SRDCF, it runs much faster than SRDCF (70fps vs. 7fps), and its memory requirement is much smaller than SRDCF's (130MB vs. 600MB), as shown in Table.1.

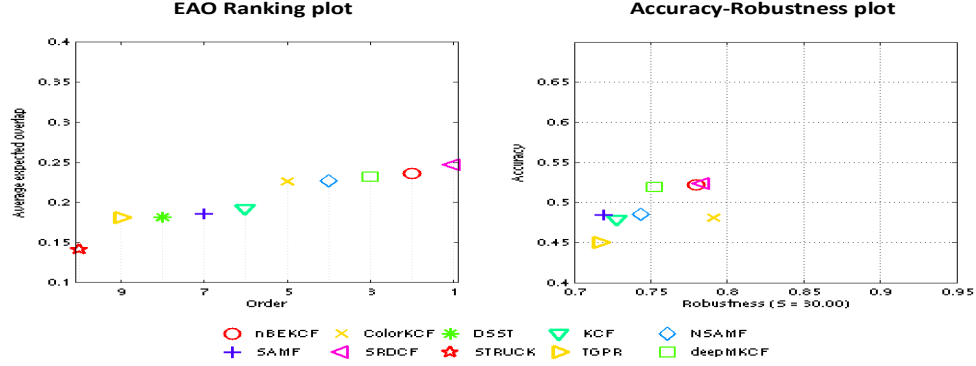


Figure 3: The EAO ranking and accuracy-robustness plots of nBEKCF, SRDCF [2], DSST [1], NSAMF [8], deepMKCF [8], SAMF [7], KCF [6], ColorKCF [8], TGPR [4], and Struck [5].

We also compare our nBEKCF and the above 9 trackers with the criteria used on OTB50 and OTB100. The results are shown in Fig.4. It is seen from the figure that nBEKCF outperforms the second best SRDCF on the average precision score and AUC by 2.0% and 2.1%, respectively. Table 2 shows the means of overlap precisions of all compared trackers, where our nBEKCF is also the best.

Table 1: Memory Requirement (MB) of Trackers.

Tracker	nBEKCF	SRDCF	DSST	KCF	NSAMF	SAMF	LCT	MEEM
MemoryReq	130	600	50	45	45	45	90	60

References

- [1] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *Proc. British Machine Vision Conference (BMVC)*, 2014.
- [2] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Learning spatially regularized correlation filters for visual tracking. In *Proc. International Conference on Computer Vision*, 2015.
- [3] P. David. *Circulant Matrices*. Chelsea Publishing Company, 2nd edition, 1994.
- [4] J. Gao, H. Ling, and J. Hu, W. and Xing. Transfer learning based visual tracking with gaussian process regression. In *Proc. European Conference on Computer Vision*, 2014.
- [5] S. Hare, A. Saffari, S. Golodetz, V. Vineet, M. Cheng, and P. Torr. Struck: Structured output tracking with kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.
- [6] J. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.37:pp.583–596, 2015.
- [7] Y. Li and J. Zhu. A scale adaptive kernel correlation filter tracker with feature integration. In *Proc. European Conference on Computer Vision Workshop*, 2014.
- [8] M. Matej Kristan, A. Leonardis, J. Matas, M. Felsberg, and et. al. The visual object tracking vot2016 challenge results. In *Proc. Computer Vision - ECCV 2016 Workshops*, 2016.

Table 2: Comparison of the 9 top trackers on the VOT2016 dataset in mean overlap precision. The best and second best results are shown in red and blue colors, respectively. Our nBEKCF achieves a gain of 4.71%, compared to the second best SRDCF.

VOT2016	nBEKCF	ColorKCF	deepMKCF	DSST	KCF	NSAMF	SAMF	SRDCF	Struck	TGPR
MeanOP	0.4812	0.3341	0.3840	0.3840	0.3315	0.3998	0.3642	0.4341	0.2337	0.2745

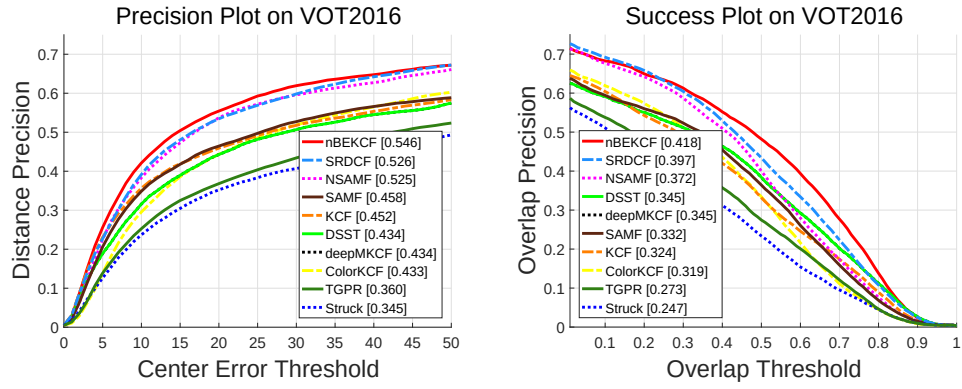


Figure 4: The precision and success plots of nBEKCF, SRDCF [2], DSST [1], NSAMF [8], deepMKCF [8], SAMF [7], KCF [6], ColorKCF [8], TGPR [4], and Struck [5].