



Research article

Four mathematical modeling forms for correlation filter object tracking algorithms and the fast calculation for the filter

Yingpin Chen^{1,2,*} and Kaiwei Chen^{1,*}

¹ School of Physics and Information Engineering, Minnan Normal University, Zhangzhou 363000, China

² Key Laboratory of Light Field Manipulation and System Integration Applications in Fujian Province, Minnan Normal University, Zhangzhou 363000, China

* **Correspondence: Email:** 110500617@163.com, 1914931478@qq.com.

Abstract: The correlation filter object tracking algorithm has gained extensive attention from scholars in the field of tracking because of its excellent tracking performance and efficiency. However, the mathematical modeling relationships of correlation filter tracking frameworks are unclear. Therefore, many forms of correlation filters are susceptible to confusion and misuse. To solve these problems, we attempted to review various forms of the correlation filter and discussed their intrinsic connections. First, we reviewed the basic definitions of the circulant matrix, convolution, and correlation operations. Then, the relationship among the three operations was discussed. Considering this, four mathematical modeling forms of correlation filter object tracking from the literature were listed, and the equivalence of the four modeling forms was theoretically proven. Then, the fast solution of the correlation filter was discussed from the perspective of the diagonalization property of the circulant matrix and the convolution theorem. In addition, we delved into the difference between the one-dimensional and two-dimensional correlation filter responses as well as the reasons for their generation. Numerical experiments were conducted to verify the proposed perspectives. The results showed that the filters calculated based on the diagonalization property and the convolution property of the cyclic matrix were completely equivalent. The experimental code of this paper is available at <https://github.com/110500617/Correlation-filter/tree/main>.

Keywords: correlation filter; object tracking; diagonalization of circulant matrix; convolution operator; correlation operator

1. Introduction

Object tracking [1–3] technology has become a research hotspot in the field of computer vision [4] and it is widely employed in intelligent traffic management [5,6], unmanned aerial vehicle tracking [7,8], and human-computer interactions [9,10]. Correlation filter object tracking algorithms [11–15] have gained increasing attention in the field of tracking, owing to their excellent tracking performance and efficiency. These methods have become mainstream for visual tracking [16–20].

The correlation operator is a signal processing operator that is used to measure signal similarity. Thus, it is widely employed in the field of object tracking. For example, the correlation operator was first introduced into the field of object tracking by Bolme et al. [21] in 2010. In 2015, Henriques et al. [11] proposed a correlation filter model in the form of a circulant matrix to train a classifier through intensive sampling by the cyclic shift. In 2016, Bertinetto et al. [22] introduced the correlation operator in a two-branches weight-shared deep learning network and proposed a SiamFC, a fully convolutional Siamese network. In 2017, Galoogahi et al. [23] proposed a background-aware correlation filter (BACF) in the form of vector multiplication, which cleverly avoids the boundary effect existing in the correlation filter tracking method. In 2020, Li et al. [24] proposed a correlation filter model in the form of convolution operations, which uses local and global information of response graphs to achieve adaptive spatio-temporal regularization. In 2022, Song et al. [25] proposed a Transformer tracker with cyclic shifting window attention, which is calculated by correlation operator. In 2024, Chen et al. [26] regarded the correlation operator as the convolution operation and proposed an asymmetrical background-aware correlation filter for object tracking by exploring the shape information of the object. In 2024, Chen et al. [27] introduced the deep-convolutional-neural-network-based features in correlation filter framework to further improve the tracking performance of BACF.

The correlation filter object tracking method in the form of circulant matrix utilizes a cyclic shift matrix [28,29] to generate many virtual samples, thereby expanding the sample richness to improve algorithm performance. Specifically, the algorithm pulls the training samples into row vectors, and a matrix with a row circulant structure is subsequently formed via a continuous cyclic shift. The filter was designed using this matrix. There are two drawbacks in directly solving the correlation filter in the spatial domain: 1) The spatial domain operation involves the inversion of a large circulant matrix, resulting in high computational complexity; and 2) the matrix formed by the cyclic shift contains a large amount of redundant information, which will occupy a large amount of storage while calculating the filter. Therefore, the property that the circulant matrix can be diagonalized by the Fourier transform matrix is invoked [30–32] to transform the correlation operation into the entry-wise multiplication operation in the frequency domain to avoid the inverse operation of the large spatial matrix. Notably, the single sample in the frequency domain replaces the virtual sample generated by the cyclic shift, effectively reducing the complexity and storage requirements of the correlation operation.

The discrete convolution operation [33] is important in signal processing. In a discrete convolution operation, the signal is reversed and shifted. This moving signal is multiplied entry-wise with another stationary signal and summed to obtain the convolution result. The difference between the correlation and convolution operations is that the correlation operation does not perform the reverse operation on the moving signal. Rather, the correlation operation directly moves the signal. Therefore, the correlation operation is a special type of convolution. Given the convolution operation, the translation, multiplication, and summation calculations of the spatial domain can be transformed into a frequency domain entry-wise multiplication operation based on the convolution theorem [33] and Parseval's

theorem [34,35] to avoid the high storage and computation requirements involved with moving the signal in the spatial domain. Researchers have understood the correlation object tracking framework from the perspective of convolution.

The two approaches previously described (the diagonalization of the circulant matrix [36,37] and the transformation of the correlation operator into a convolution) yield the same form of computation, namely, the calculation of the correlation operation via the frequency domain entry-wise multiplication operation, albeit from different perspectives. Hence, there must be a close internal relationship among the different mathematical modeling approaches of the CFs. With the improvement and perfection of correlation filter tracking theory [38–42], various forms of object tracking algorithms have been proposed. Based on a mathematical modeling perspective, correlation filter object tracking algorithms can be specifically classified into four forms: Correlation operations [21], vector multiplication operations [23], circulant matrix operations [29], and convolution operations [24]. These four modeling methods are expressed differently but are essentially equivalent.

The motivation of this paper is to sort out four mathematical modeling methods for the correlation filter object tracking algorithm by exploring the properties of circulant matrix, convolution, and correlation operations. First, we review the definitions of these four modeling methods. Then, the internal relations of the four modeling methods are discussed in detail. Based on the properties and relationships among circulant matrix, convolution, and correlation operations, two fast correlation filter calculation methods are proposed. Both theoretical derivation and experimental results prove the equivalence of the two methods. Numerical experiments verified the proposed viewpoint. In addition, Most existing studies on the correlation filter [16,23] investigated filter calculation in the form of a one-dimensional filter. Recently, few studies have presented a solution to the correlation filter in the form of a two-dimensional matrix [26,37]. Thus, we further discuss the relationship and difference between the one-dimensional and two-dimensional filters.

The main contributions of this study are as follows. 1) We comprehensively describe the definitions of the circulant matrix, convolution, and correlation operations and then theoretically prove the four theorems of the circulant matrix. Based on these theorems, the relationships of four modeling approaches for the correlation filter are further discussed. 2) The fast calculation of the correlation filter is discussed from two perspectives: the diagonalization property of the circulant matrix and the convolution theorem. The multiplication and inversion operations of the large-scale matrix are transformed into entry-wise multiplication and entry-wise division operations of the vector to improve the efficiency of the filter solution. 3) We convert a one-dimensional correlation filter into a two-dimensional correlation filter, present the calculation flow of the two filter methods, analyze the differences and connections between the two filter methods, and discuss the reasoning behind these relationships.

The rest of this paper is organized as follows. In Section 2, we present the definitions of the three operations of correlation, circulant matrix, and convolution; argue the four theorems of the circulant matrix; and discuss the relationship among the three operations in depth. In Section 3, we enumerate the four forms of correlation filter tracking modeling. In Section 4, we present the solution to the filter from the perspectives of the diagonalization of circulant matrix and the convolution theorem. In Section 5, we discuss the differences and connections between one-dimensional and two-dimensional filters in detail. In Section 6, we present the verification of the viewpoints presented in this study through numerical experimentation and response plots to verify the equivalence of the two methods for solving the filter. Finally, in Section 7, we draw conclusions and present the outlook for future work.

2. Theoretical basis of the correlation filter

2.1. Definition of the three operations

2.1.1. Definition of the column-vector-based circulant matrix

Suppose the first column vector of the matrix is $\mathbf{x} = (x_0, x_1, x_2, \dots, x_{N-1})^T \in \mathbb{R}^{N \times 1}$, where the superscript T denotes the transpose operation. \mathbf{x} is cyclically shifted by one bit to obtain the second-column vector $\mathbf{v} = (x_{N-1}, x_0, x_1, \dots, x_{N-2})^T \in \mathbb{R}^{N \times 1}$ of the column-vector-based circulant matrix. The N column vectors are obtained after N cyclic shifts. These vectors form the column circulant

$$\text{matrix } \mathbf{C}(\mathbf{x}) = \begin{pmatrix} x_0 & x_{N-1} & x_{N-2} & \cdots & x_1 \\ x_1 & x_0 & x_{N-1} & \cdots & x_2 \\ x_2 & x_1 & x_0 & \cdots & x_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{N-1} & x_{N-2} & x_{N-3} & \cdots & x_0 \end{pmatrix} \in \mathbb{R}^{N \times N}.$$

Similarly, the vector $\mathbf{x}^T = (x_0, x_1, x_2, \dots, x_{N-1}) \in \mathbb{R}^{1 \times N}$ is cyclically shifted N times as the base vector to obtain N row vectors. These vectors form the row-vector-based circulant matrix

$$\mathbf{C}(\mathbf{x}^T) = \begin{pmatrix} x_0 & x_1 & x_2 & \cdots & x_{N-1} \\ x_{N-1} & x_0 & x_1 & \cdots & x_{N-2} \\ x_{N-2} & x_{N-1} & x_0 & \cdots & x_{N-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1 & x_2 & x_3 & \cdots & x_0 \end{pmatrix} \in \mathbb{R}^{N \times N}.$$

The patches obtained by the traditional correlation filter through N cyclic shifts form a circulant matrix. Among the samples generated by the cyclic shift operation, only the first row represents the real sample.

2.1.2. Definition of the discrete convolution operator

Discrete convolution is given by

$$(\mathbf{x} * \mathbf{h})(n) = \sum_{m=0}^{N-1} \mathbf{x}(m)\mathbf{h}(n-m), \quad (1)$$

where $(\mathbf{x} * \mathbf{h}) \in \mathbb{R}^{N \times 1}$ and $(\mathbf{x} * \mathbf{h})(n)$ are the n th element of the vector $\mathbf{x} * \mathbf{h}$, $*$ is a one-dimensional convolution operation, and the signals $\mathbf{x} = (x_0, x_1, \dots, x_{N-1})^T \in \mathbb{R}^{N \times 1}$ and $\mathbf{h} = (h_0, h_1, \dots, h_{N-1})^T \in \mathbb{R}^{N \times 1}$ satisfy the periodic boundary conditions. Notably, in the correlation filter tracking framework, the \mathbf{h} is the correlation filter, while in the Siamese tracking framework, \mathbf{h} is the test sample, $n = 0, 1, \dots, N-1$.

2.1.3. Definition of the correlation operation

The correlation operation is defined as

$$(\mathbf{x} \star \mathbf{h})(n) = \sum_{m=0}^{N-1} \mathbf{x}(m)\mathbf{h}(n+m) = \mathbf{x}^T \mathbf{h}[\Delta\tau_n], \quad (2)$$

where \star is a one-dimensional correlation operator and $\mathbf{h}[\Delta\tau_n] = \text{circshift}(\mathbf{h}, n)$, $\text{circshift}(\mathbf{h}, n)$ denotes the cyclic shift operator that shifts the signal by the n ($n=0, 1, \dots, N-1$) step.

2.2. Diagonalization theorem of circulant matrix

A circulant-matrix structure can effectively capture the motion characteristics of an object and provide accurate prediction information during tracking. However, there is redundancy in circulant matrix data, resulting in a large number of computations when operating in the spatial domain. To solve this problem, the computational complexity must be reduced using **Theorem 1**.

Theorem 1: If the column-vector-based circulant matrix $\mathbf{C}(\mathbf{x}) = \begin{pmatrix} x_0 & x_{N-1} & x_{N-2} & \cdots & x_1 \\ x_1 & x_0 & x_{N-1} & \cdots & x_2 \\ x_2 & x_1 & x_0 & \cdots & x_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{N-1} & x_{N-2} & x_{N-3} & \cdots & x_0 \end{pmatrix}$ is

known, then the discrete Fourier transform matrix is

$$\mathbf{F}_N = \begin{pmatrix} 1 & 1 & \cdots & 1 & 1 \\ 1 & e^{-j\frac{2\pi \times 1 \times 1}{N}} & \cdots & e^{-j\frac{2\pi \times (N-2) \times 1}{N}} & e^{-j\frac{2\pi \times (N-1) \times 1}{N}} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & e^{-j\frac{2\pi \times 1 \times (N-2)}{N}} & \cdots & e^{-j\frac{2\pi \times (N-2) \times (N-2)}{N}} & e^{-j\frac{2\pi \times (N-1) \times (N-2)}{N}} \\ 1 & e^{-j\frac{2\pi \times 1 \times (N-1)}{N}} & \cdots & e^{-j\frac{2\pi \times (N-2) \times (N-1)}{N}} & e^{-j\frac{2\pi \times (N-1) \times (N-1)}{N}} \end{pmatrix}, \text{ the inverse Fourier transform matrix is,}$$

$$\mathbf{F}_N = \begin{pmatrix} 1 & 1 & \cdots & 1 & 1 \\ 1 & e^{j\frac{2\pi \times 1 \times 1}{N}} & \cdots & e^{j\frac{2\pi \times (N-2) \times 1}{N}} & e^{j\frac{2\pi \times (N-1) \times 1}{N}} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & e^{j\frac{2\pi \times 1 \times (N-2)}{N}} & \cdots & e^{j\frac{2\pi \times (N-2) \times (N-2)}{N}} & e^{j\frac{2\pi \times (N-1) \times (N-2)}{N}} \\ 1 & e^{j\frac{2\pi \times 1 \times (N-1)}{N}} & \cdots & e^{j\frac{2\pi \times (N-2) \times (N-1)}{N}} & e^{j\frac{2\pi \times (N-1) \times (N-1)}{N}} \end{pmatrix}, \text{ and } \hat{\mathbf{x}} = \mathbf{F}_N \mathbf{x} \text{ is the one-dimensional Fourier}$$

transform of the vector $\mathbf{x} = (x_0, x_1, x_2, \dots, x_{N-1})^T \in \mathbb{R}^{N \times 1}$. Then, we obtain

$$\mathbf{F}_N \mathbf{C}(\mathbf{x}) \mathbf{F}_N^{-1} = \mathbf{Diag}(\hat{\mathbf{x}}) = \begin{pmatrix} \hat{x}_0 & 0 & \cdots & 0 \\ 0 & \hat{x}_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \hat{x}_{N-1} \end{pmatrix}, \text{ where } \mathbf{Diag} \text{ is the operator that stacks the column}$$

vectors onto the diagonal of the diagonal matrix.

Proof: In $\mathbf{C}(\mathbf{x}) \mathbf{F}_N^{-1}$, the first row of $\mathbf{C}(\mathbf{x})$ multiplied by the $k+1$ th ($k=0, 1, \dots, N-1$) column of \mathbf{F}_N^{-1} is denoted as $f_{(0,k)}$. Then, we have

$$f_{(0,k)} = \frac{1}{N} (x_0 + x_{N-1} e^{kj\frac{2\pi \times 1}{N}} + x_{N-2} e^{kj\frac{2\pi \times 2}{N}} + \cdots + x_1 e^{kj\frac{2\pi \times (N-1)}{N}}). \quad (3)$$

Using the Euler relation $e^{-2\pi kj} = \cos(-2\pi k) + j\sin(-2\pi k) = 1$, we have

$$e^{kj\frac{2\pi}{N}} = e^{kj\frac{2\pi}{N}} \cdot e^{-2\pi kj} = e^{kj\frac{2\pi}{N} - 2\pi kj} = e^{-kj\frac{2\pi(N-1)}{N}}. \quad (4)$$

Likewise, we have

$$e^{kj\frac{2\pi(N-1)}{N}} = e^{kj\frac{2\pi(N-1)}{N}} \cdot e^{-2\pi kj} = e^{kj\frac{2\pi(N-1)}{N} - 2\pi kj} = e^{-kj\frac{2\pi}{N}}. \quad (5)$$

According to the period invariance of the complex signals, Eq (3) can be rewritten as

$$\begin{aligned} f_{(0,k)} &= \frac{1}{N} (x_0 + x_{N-1} e^{-kj\frac{2\pi(N-1)}{N}} + \dots + x_1 e^{-kj\frac{2\pi \times 1}{N}}) \\ &= \frac{1}{N} (x_0 + x_1 e^{-kj\frac{2\pi \times 1}{N}} + x_2 e^{-kj\frac{2\pi \times 2}{N}} \dots + x_{N-1} e^{-kj\frac{2\pi(N-1)}{N}}) \\ &= \frac{1}{N} \hat{\mathbf{x}}_k, \end{aligned} \quad (6)$$

where $\hat{\mathbf{x}}_k$ is the $k+1$ th ($k=0,1,\dots,N-1$) element of $\hat{\mathbf{x}} = \mathbf{fft}_1(\mathbf{x}) = \mathbf{F}_N \mathbf{x}$ (\mathbf{fft}_1 is a one-dimensional fast Fourier transform operator), that is, the $k+1$ th row of \mathbf{F}_N multiplied by the vector \mathbf{x} .

The second row of $\mathbf{C}(\mathbf{x})$ multiplied by the $k+1$ th ($k=0,1,\dots,N-1$) column of \mathbf{F}_N^{-1} is denoted as $f_{(1,k)}$. The second row of $\mathbf{C}(\mathbf{x})$ is the right-shifted signal of the first row of $\mathbf{C}(\mathbf{x})$, we have:

$$\begin{aligned} f_{(1,k)} &= \frac{1}{N} (x_1 + x_0 e^{kj\frac{2\pi \times 1}{N}} + \dots + x_2 e^{kj\frac{2\pi(N-1)}{N}}) \\ &= \frac{1}{N} (x_0 e^{kj\frac{2\pi \times 1}{N}} + x_1 + x_2 e^{kj\frac{2\pi(N-1)}{N}} + \dots + x_{N-1} e^{kj\frac{2\pi \times 2}{N}}) \\ &= \frac{1}{N} (x_0 e^{kj\frac{2\pi \times 1}{N}} + x_1 + x_2 e^{-kj\frac{2\pi \times 1}{N}} + \dots + x_{N-1} e^{-kj\frac{2\pi(N-2)}{N}}) \\ &= \frac{e^{kj\frac{2\pi \times 1}{N}}}{N} (x_0 + x_1 e^{-kj\frac{2\pi \times 1}{N}} + x_2 e^{-kj\frac{2\pi \times 2}{N}} + \dots + x_{N-1} e^{-kj\frac{2\pi(N-1)}{N}}) \\ &= e^{kj\frac{2\pi \times 1}{N}} f_{(0,k)}. \end{aligned} \quad (7)$$

Thus, we have

$$\begin{aligned} \mathbf{C}(\mathbf{x}) \mathbf{F}_N^{-1} &= \begin{pmatrix} f_{(0,0)} & f_{(0,1)} & \dots & f_{(0,N-1)} \\ f_{(1,0)} & f_{(1,1)} & \dots & f_{(1,N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ f_{(N-1,0)} & f_{(N-1,1)} & \dots & f_{(N-1,N-1)} \end{pmatrix} = \frac{1}{N} \begin{pmatrix} \hat{\mathbf{x}}_0 & \hat{\mathbf{x}}_1 & \dots & \hat{\mathbf{x}}_{N-1} \\ e^{j\frac{2\pi \times 0 \times 1}{N}} \hat{\mathbf{x}}_0 & e^{j\frac{2\pi \times 1 \times 1}{N}} \hat{\mathbf{x}}_1 & \dots & e^{j\frac{2\pi \times (N-1) \times 1}{N}} \hat{\mathbf{x}}_{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ e^{j\frac{2\pi \times 0 \times (N-1)}{N}} \hat{\mathbf{x}}_0 & e^{j\frac{2\pi \times 1 \times (N-1)}{N}} \hat{\mathbf{x}}_1 & \dots & e^{j\frac{2\pi \times (N-1) \times (N-1)}{N}} \hat{\mathbf{x}}_{N-1} \end{pmatrix} \\ &= \frac{1}{N} \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & e^{j\frac{2\pi}{N}} & \dots & e^{j\frac{2\pi(N-1)}{N}} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{j\frac{2\pi(N-1)}{N}} & \dots & e^{j\frac{2\pi(N-1)^2}{N}} \end{pmatrix} \begin{pmatrix} \hat{\mathbf{x}}_0 \\ \hat{\mathbf{x}}_1 \\ \vdots \\ \hat{\mathbf{x}}_{N-1} \end{pmatrix} = \mathbf{F}_N^{-1} \mathbf{Diag}(\hat{\mathbf{x}}). \end{aligned} \quad (8)$$

Therefore, we have

$$\mathbf{F}_N \mathbf{C}(\mathbf{x}) \mathbf{F}_N^{-1} = \mathbf{F}_N \mathbf{F}_N^{-1} \mathbf{Diag}(\hat{\mathbf{x}}) = \mathbf{Diag}(\hat{\mathbf{x}}). \quad (9)$$

Hence, **Theorem 1** is proven.

According to the diagonalization theorem of the column circulant matrix, the diagonalization theorem of the row circulant matrix is derived as follows.

Theorem 2: If the row-vector-based circulant matrix $\mathbf{C}(\mathbf{x}^T) = \begin{pmatrix} x_0 & x_1 & x_2 & \cdots & x_{N-1} \\ x_{N-1} & x_0 & x_1 & \cdots & x_{N-2} \\ x_{N-2} & x_{N-1} & x_0 & \cdots & x_{N-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1 & x_2 & x_3 & \cdots & x_0 \end{pmatrix}$ is

known, let $\mathbf{F}_n = \frac{\mathbf{F}_N}{\sqrt{N}}$ denote the normalized discrete Fourier transformation (DFT) matrix. Then, the row-vector-based circulant matrix satisfies $\mathbf{C}(\mathbf{x}^T) = \mathbf{F}_n \mathbf{Diag}(\hat{\mathbf{x}}) \mathbf{F}_n^H$ (where \mathbf{F}_n^H denotes the result of the conjugate transpose on \mathbf{F}_n).

Proof: The diagonalization theorem of the row-vector-based circulant matrix can be proven using the diagonalization theorem of the column-vector-based circulant matrix. According to **Theorem 1**, we simultaneously transpose both sides of the equation to obtain

$$\mathbf{C}(\mathbf{x}^T) = (\mathbf{F}_N^T) (\mathbf{Diag}(\hat{\mathbf{x}}))^T (\mathbf{F}_N^{-1})^T = (\mathbf{F}_N^T) (\mathbf{Diag}(\hat{\mathbf{x}}))^T \left(\frac{\mathbf{F}_N^*}{N} \right)^T, \quad (10)$$

where $\mathbf{F}_N^{-1} = \frac{1}{N} \mathbf{F}_N^*$. By decomposing N into $\sqrt{N} \cdot \sqrt{N}$, we have

$$\mathbf{C}(\mathbf{x}^T) = \left(\frac{\mathbf{F}_N^T}{\sqrt{N}} \right) (\mathbf{Diag}(\hat{\mathbf{x}}))^T \left(\frac{\mathbf{F}_N^*}{\sqrt{N}} \right)^T. \quad (11)$$

As the normalized DFT matrix $\mathbf{F}_n = \frac{\mathbf{F}_N}{\sqrt{N}}$ satisfies $\mathbf{F}_n^T = \mathbf{F}_n$ and $\mathbf{F}_n^H = \mathbf{F}_n^{-1}$, we obtain

$$\mathbf{C}(\mathbf{x}^T) = \mathbf{F}_n (\mathbf{Diag}(\hat{\mathbf{x}}))^T \mathbf{F}_n^{-1} = \mathbf{F}_n (\mathbf{Diag}(\hat{\mathbf{x}})) \mathbf{F}_n^H. \quad (12)$$

Hence, **Theorem 2** is proven.

2.3. Relationship between convolution and circulant matrices

Directly performing operations with the circulant matrix in the spatial domain leads to high computational complexity. However, another function of the correlation filter tracking algorithm is to utilize the relationship between the convolution and circulant matrices. The convolution theorem can transform it into an entry-wise operation in the frequency domain to circumvent large matrix multiplication and inverse operations, effectively reducing the number of operations and improving the computational efficiency.

Theorem 3: By multiplying the column-vector-based circulant matrix $\mathbf{C}(\mathbf{x})$ by a signal $\mathbf{h} = (h_0, h_1, \dots, h_{N-1})^T \in \mathbb{R}^{N \times 1}$, then we have

$$\mathbf{C}(\mathbf{x})\mathbf{h} = \begin{pmatrix} x_0 & x_{N-1} & x_{N-2} & \cdots & x_1 \\ x_1 & x_0 & x_{N-1} & \cdots & x_2 \\ x_2 & x_1 & x_0 & \cdots & x_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{N-1} & x_{N-2} & x_{N-3} & \cdots & x_0 \end{pmatrix} \begin{pmatrix} h_0 \\ h_1 \\ h_2 \\ \vdots \\ h_{N-1} \end{pmatrix} = \mathbf{x} * \mathbf{h}. \quad (13)$$

Proof: By multiplying both sides of Eq (9) with $\hat{\mathbf{h}}$ simultaneously, we have

$$\mathbf{F}_N \mathbf{C}(\mathbf{x}) \mathbf{F}_N^{-1} \hat{\mathbf{h}} = \text{Diag}(\hat{\mathbf{x}}) \hat{\mathbf{h}}. \quad (14)$$

Then

$$\mathbf{F}_N \mathbf{C}(\mathbf{x}) \mathbf{h} = \hat{\mathbf{x}} \odot \hat{\mathbf{h}}, \quad (15)$$

where \odot denotes the entry-wise multiplication operation.

According to the convolution theorem, the entry-wise multiplication of the spectrum of the two signals is equal to that of the spatial convolution signal, we have

$$\hat{\mathbf{x}} \odot \hat{\mathbf{h}} = \mathbf{F}_N(\mathbf{x} * \mathbf{h}). \quad (16)$$

By combining Eqs (15) and (16), we obtain

$$\mathbf{F}_N \mathbf{C}(\mathbf{x}) \mathbf{h} = \mathbf{F}_N(\mathbf{x} * \mathbf{h}). \quad (17)$$

Then, it is seen that

$$\mathbf{C}(\mathbf{x}) \mathbf{h} = \mathbf{x} * \mathbf{h}. \quad (18)$$

Hence, **Theorem 3** is proven.

According to the convolution theorem, spatial convolution can be calculated in the frequency domain. The specific calculation method is

$$\mathbf{x} * \mathbf{h} = \text{real} \left(\text{ifft}_1 \left(\hat{\mathbf{x}} \odot \hat{\mathbf{h}} \right) \right), \quad (19)$$

where **real** is the real part-taking operator and **ifft₁** is a one-dimensional inverse-Fourier transform operator.

Comment 1: The calculation of $\mathbf{x} * \mathbf{h}$ can be expressed as $\mathbf{C}(\mathbf{x})\mathbf{h}$ if directly calculated in the spatial domain. The memory space occupied by $\mathbf{C}(\mathbf{x})\mathbf{h}$ is denoted by $N^2 + N$, and its multiplication complexity is denoted by $\mathcal{O}(N^2)$. Notably, $\mathbf{x} * \mathbf{h}$ can be calculated in the frequency domain using the convolution theorem, that is, $\mathbf{x} * \mathbf{h} = \text{real} \left(\text{ifft}_1 \left(\hat{\mathbf{x}} \odot \hat{\mathbf{h}} \right) \right)$. The memory space occupied

by $\text{real}(\text{ifft}_1(\hat{x} \odot \hat{h}))$ is denoted by $4N$, and its multiplication complexity is denoted by $\mathcal{O}(8N \log_2 N + 4N)$ (It includes two fast Fourier transforms, one inverse Fourier transform and the multiplication of complex numbers in the frequency domain, where two fast Fourier transforms require $2N \log_2 N$ multiplications with complex number and real number, involving $4N \log_2 N$ floating-point number multiplications. One fast inverse Fourier transform requires $N \log_2 N$ complex number multiplications, involving $4N \log_2 N$ floating-point number multiplications, and the number of floating-point number multiplications required for the multiplication of N complex numbers in the frequency domain is $4N$).

Table 1 presents the occupied memory space and the complexity of the floating-point multiplication operation of the $C(\mathbf{x})\mathbf{h}$ and $\text{real}(\text{ifft}_1(\hat{x} \odot \hat{h}))$ operations. For example, when N is 4, the $C(\mathbf{x})\mathbf{h}$ operation occupies $N^2 + N|_{N=4} = 4^2 + 4 = 20$ floating-point units, and the number of floating-point multiplications is $N^2|_{N=4} = 4^2 = 16$. The $\text{real}(\text{ifft}_1(\hat{x} \odot \hat{h}))$ operation occupies $4N|_{N=4} = 4 \times 4 = 16$ floating-point units, and the number of floating-point multiplications is $8N \log_2 N + 4N|_{N=4} = 8 \times 4 \times \log_2 4 + 4 \times 4 = 80$. When N is 256, the $C(\mathbf{x})\mathbf{h}$ operation occupies $N^2 + N|_{N=256} = 256^2 + 256 = 65792$ floating-point units, and the number of floating-point multiplications is $N^2|_{N=256} = 256^2 = 65536$. By contrast, the $\text{real}(\text{ifft}_1(\hat{x} \odot \hat{h}))$ operation occupies $4N|_{N=256} = 4 \times 256 = 1024$ floating-point units, and the number of floating-point multiplications is $8N \log_2 N + 4N|_{N=256} = 8 \times 256 \times \log_2 256 + 4 \times 256 = 17408$. The results show that when the image size is large, the inverse operation of the large circular matrix in the spatial domain can be transformed into the entry-wise multiplication operation in the frequency domain according to **Theorem 3** and the convolution theorem to effectively reduce the number of solving operations.

Table 1. Memory footprint and computational complexity analysis of the one-dimensional operation.

Operation (image size $\mathbb{R}^{\sqrt{N} \times \sqrt{N}}$)	Memory space occupied/floating-point unit	Complexity of floating-point multiplication operation/time
$C(\mathbf{x})\mathbf{h}$ operation	$N^2 + N$	$\mathcal{O}(N^2)$
$\text{real}(\text{ifft}_1(\hat{x} \odot \hat{h}))$ operation	$4N$	$\mathcal{O}(8N \log_2 N + 4N)$

Theorem 4: The row-vector-based circulant matrix satisfies $C(\mathbf{x}^T)\mathbf{h} = C(\bar{\mathbf{x}})\mathbf{h} = \bar{\mathbf{x}} * \mathbf{h}$.

Proof: By observing each column of the row-vector-based circulant matrix

$$C(\mathbf{x}^T) = \begin{pmatrix} x_0 & x_1 & x_2 & \cdots & x_{N-1} \\ x_{N-1} & x_0 & x_1 & \cdots & x_{N-2} \\ x_{N-2} & x_{N-1} & x_0 & \cdots & x_{N-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1 & x_2 & x_3 & \cdots & x_0 \end{pmatrix},$$

we find that each column of $C(\mathbf{x}^T)$ is obtained by the cyclic shift of the previous column. The first column $\bar{\mathbf{x}} = (x_0, x_{N-1}, \dots, x_1)^T \in \mathbb{R}^{N \times 1}$ of this matrix is the reverse

signal of $\mathbf{x} = (x_0, x_1, \dots, x_{N-1})^T \in \mathbb{R}^{N \times 1}$ (e.g., if $\mathbf{x} = [1, 2, 3, 4]^T$, then $\bar{\mathbf{x}} = [1, 4, 3, 2]^T$). Then, we have: $C(\mathbf{x}^T) = C(\bar{\mathbf{x}})$. Combining this result with Eq (13), we obtain

$$C(\mathbf{x}^T)\mathbf{h} = C(\bar{\mathbf{x}})\mathbf{h} = \bar{\mathbf{x}} * \mathbf{h}. \quad (20)$$

Hence, **Theorem 4** is proven.

Comment 2: We can regard \mathbf{h} in $\bar{\mathbf{x}} * \mathbf{h}$ as a static signal and $\bar{\mathbf{x}}$ as a dynamic signal. According to the definition of the discrete convolution, the dynamic signal should be reversed as $\bar{\bar{\mathbf{x}}}$, which is equal to \mathbf{x} . Then, the reverse signal $\bar{\bar{\mathbf{x}}}$ should be cyclically shifted as the shifted vectors. The vector formed by the inner product of these shifted vectors with the static signal is the result of discrete convolution. The shifted vectors can be stacked into the row-vector-based circulant matrix $C(\mathbf{x}^T)$. Thus, $\bar{\mathbf{x}} * \mathbf{h} = C(\mathbf{x}^T)\mathbf{h}$.

Comment 3: $\bar{\mathbf{x}}$ satisfies $\text{fft}_1(\bar{\mathbf{x}}) = \hat{\mathbf{x}}^*$ (where $\hat{\mathbf{x}}^*$ is the conjugate signal of $\hat{\mathbf{x}}$).

Proof: If the spectral signal $\hat{\mathbf{x}}$ is the Fourier transform of the signal \mathbf{x} , then the spectral elements of \mathbf{x} are expressed as

$$\hat{\mathbf{x}}(k) = \sum_{n=0}^{N-1} \mathbf{x}(n) e^{-kj \frac{2\pi n}{N}} = \sum_{n=1}^{N-1} \mathbf{x}(n) e^{-kj \frac{2\pi n}{N}} + \mathbf{x}(0). \quad (21)$$

Similarly, the spectral elements of $\bar{\mathbf{x}}$ are given by

$$\text{fft}_1(\bar{\mathbf{x}})(k) = \sum_{n=0}^{N-1} \bar{\mathbf{x}}(n) e^{-kj \frac{2\pi n}{N}} = \sum_{n=1}^{N-1} \mathbf{x}(N-n) e^{-kj \frac{2\pi n}{N}} + \mathbf{x}(0), \quad (22)$$

where $\text{fft}_1(\bar{\mathbf{x}})(k)$ is the $k+1$ th ($k = 0, 1, \dots, N-1$) element of $\text{fft}_1(\bar{\mathbf{x}})$.

Based on the time-shifting properties of the discrete Fourier transform and Euler's formula $e^{2\pi k j} = \cos(2\pi k) + j \sin(2\pi k) = 1$, Eq (22) can be rewritten as follows

$$\begin{aligned} \text{fft}_1(\bar{\mathbf{x}})(k) &= \sum_{n=1}^{N-1} \mathbf{x}(N-n) e^{-kj \frac{2\pi n}{N}} e^{2\pi k j} + \mathbf{x}(0) \\ &= \sum_{n=1}^{N-1} \mathbf{x}(N-n) e^{kj \frac{2\pi(N-n)}{N}} + \mathbf{x}(0) \\ &= \sum_{t=1}^{N-1} \mathbf{x}(t) e^{kj \frac{2\pi t}{N}} + \mathbf{x}(0) \Big|_{t=N-n} \\ &= \sum_{t=0}^{N-1} \mathbf{x}(t) e^{kj \frac{2\pi t}{N}}. \end{aligned} \quad (23)$$

Combining Eqs (21) and (23) yields

$$\text{fft}_1(\bar{\mathbf{x}}) = \hat{\mathbf{x}}^*. \quad (24)$$

Hence, the proof is complete.

According to Eq (20), the second type of proof for **Theorem 3** is provided as follows.

Proof:

We observe that $C(\mathbf{x}) = \begin{pmatrix} x_0 & x_{N-1} & x_{N-2} & \cdots & x_1 \\ x_1 & x_0 & x_{N-1} & \cdots & x_2 \\ x_2 & x_1 & x_0 & \cdots & x_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{N-1} & x_{N-2} & x_{N-3} & \cdots & x_0 \end{pmatrix} = C(\bar{\mathbf{x}}^T)$, thereby we obtain

$$C(\mathbf{x})\mathbf{h} = C(\bar{\mathbf{x}}^T)\mathbf{h} = \bar{\mathbf{x}} * \mathbf{h} = \mathbf{x} * \mathbf{h}, \quad (25)$$

where $\bar{\mathbf{x}} = \mathbf{x}$.

2.4. Relationship among the correlation operation, convolution, and circulant matrix

A tracking algorithm based on a correlation filter significantly improves the tracking speed by transforming complex correlation operations in the spatial domain into simple entry-wise multiplication operations in the frequency domain. Utilizing the relationship between the correlation operation and the convolution, we rewrite the correlation operator as the convolution form

$$(\mathbf{x} \star \mathbf{h})(n) = \sum_{m=0}^{N-1} \mathbf{x}(m)\mathbf{h}(n+m) = \sum_{m=0}^{N-1} \mathbf{x}(m)\mathbf{h}[n-(-m)] = (\mathbf{x} * \bar{\mathbf{h}})(n), \quad (26)$$

where $\bar{\mathbf{h}} = (h_0, h_{N-1}, \dots, h_1)^T \in \mathbb{R}^{N \times 1}$ is the reverse signal of $\mathbf{h} = (h_0, h_1, \dots, h_{N-1})^T \in \mathbb{R}^{N \times 1}$, $\bar{\mathbf{h}}$ satisfies one dimensional periodic boundary conditions.

By combining Eqs (20) and (26), the relationship between the correlation operation and the row-vector-based circulant matrix is given by

$$C(\mathbf{x}^T)\mathbf{h} = \overline{\mathbf{x} \star \mathbf{h}}. \quad (27)$$

3. Four modeling forms for correlation filter

3.1. First form of correlation filter tracking modeling: Correlation operation form

A traditional discriminative tracking algorithm distinguishes between an object and its background by training its classifier. The background information and object are used as negative and positive samples, respectively. The candidate sample with the highest response is selected as the prediction result. The correlation filter uses ridge regression to design the filter \mathbf{h} . The regularization term is added to prevent overfitting. The correlation operation form of the correlation filter is shown in Eq (28).

$$E(\mathbf{h}) = \min_{\mathbf{h}} \frac{1}{2} \|\mathbf{y} - \mathbf{x} \star \mathbf{h}\|_2^2 + \frac{\lambda}{2} \|\mathbf{h}\|_2^2, \quad (28)$$

where $\mathbf{x} \in \mathbb{R}^{N \times 1}$ is the column vector form of the object sample after the weighted cosine window and N is the size of the pixels occupied by the object sample. $\mathbf{y} \in \mathbb{R}^{N \times 1}$ is the desired correlation response, $\mathbf{h} \in \mathbb{R}^{N \times 1}$ is the filter, and λ is the balancing parameter, which is utilized to balance the

fidelity term $\frac{1}{2}\|\mathbf{y} - \mathbf{x} \star \mathbf{h}\|_2^2$ and ridge regression regularization term $\frac{\lambda}{2}\|\mathbf{h}\|_2^2$.

3.2. Second form of correlation filter tracking modeling: Vector multiplication form

The vector multiplication form of the correlation filter is given by Eq (29).

$$\begin{aligned} E(\mathbf{h}) &= \min_{\mathbf{h}} \frac{1}{2} \sum_{n=0}^{N-1} (\mathbf{y}(n) - \mathbf{x}^T \mathbf{h}[\Delta\tau_n])^2 + \frac{\lambda}{2} \|\mathbf{h}\|_2^2 \\ &= \min_{\mathbf{h}} \frac{1}{2} \sum_{n=0}^{N-1} (\bar{\mathbf{y}}(n) - \mathbf{h}^T \mathbf{x}[\Delta\tau_n])^2 + \frac{\lambda}{2} \|\mathbf{h}\|_2^2, \end{aligned} \quad (29)$$

where if $\mathbf{r} = C(\mathbf{h}^T)\mathbf{x}$, then $\mathbf{r} \in \mathbb{R}^{N \times 1}$ and $\mathbf{r}(n) = \mathbf{x}^T \mathbf{h}[\Delta\tau_n]$.

3.3. Third form of correlation filter tracking modeling: The circulant matrix operation form

A key focus area of the correlation filter tracking algorithm is improving computational efficiency using the characteristics of the circulant matrix. The circulant matrix operation form of the correlation filter is given by Eq (30):

$$\begin{aligned} E(\mathbf{h}) &= \min_{\mathbf{h}} \frac{1}{2} \|\mathbf{y} - C(\mathbf{h}^T)\mathbf{x}\|_2^2 + \frac{\lambda}{2} \|\mathbf{h}\|_2^2 \\ &= \min_{\mathbf{h}} \frac{1}{2} \|\bar{\mathbf{y}} - C(\mathbf{x}^T)\mathbf{h}\|_2^2 + \frac{\lambda}{2} \|\mathbf{h}\|_2^2, \end{aligned} \quad (30)$$

where $C(\mathbf{h}^T)$ is the row-vector-based circulant matrix. $C(\mathbf{h}^T)$ satisfies $C(\mathbf{h}^T)\mathbf{x} = \bar{\mathbf{h}} * \mathbf{x} = \mathbf{x} \star \mathbf{h}$ and $C(\mathbf{h}^T)\mathbf{x} = \overline{C(\mathbf{x}^T)\mathbf{h}}$.

3.4. Fourth form of correlation filter tracking modeling: Convolution operation form

According to Eq (26), the correlation forms in Eq (28) can be rewritten in the convolutional form of Eq (31), that is

$$E(\mathbf{h}) = \min_{\mathbf{h}} \frac{1}{2} \|\mathbf{y} - \bar{\mathbf{h}} * \mathbf{x}\|_2^2 + \frac{\lambda}{2} \|\bar{\mathbf{h}}\|_2^2, \quad (31)$$

where $*$ denotes the convolution operator that satisfies $\mathbf{x} \star \mathbf{h} = \mathbf{x} * \bar{\mathbf{h}}$.

According to the convolution and Parseval's theorems, Eq (31) can be written in the frequency-domain form as follows

$$E(\mathbf{h}) = \min_{\hat{\mathbf{h}}} \frac{1}{2N} \|\hat{\mathbf{y}} - \hat{\mathbf{h}}^* \odot \hat{\mathbf{x}}\|_2^2 + \frac{\lambda}{2N} \|\hat{\mathbf{h}}^*\|_2^2, \quad (32)$$

where $\hat{\mathbf{h}}$ is the Fourier transform of \mathbf{h} , and $\hat{\mathbf{h}}^*$ is the conjugate signal of $\hat{\mathbf{h}}$.

3.5. Relationship among the four modeling forms of correlation filter object tracking

Based on the above discussion, we determine the relationship among the four mathematical modeling forms of correlation filter tracking as follows

$$\begin{aligned}
 E(\mathbf{h}) &= \min_{\mathbf{h}} \frac{1}{2} \|\mathbf{y} - \mathbf{x} \star \mathbf{h}\|_2^2 + \frac{\lambda}{2} \|\mathbf{h}\|_2^2 \\
 &= \min_{\mathbf{h}} \frac{1}{2} \sum_{n=0}^{N-1} (\mathbf{y}(n) - \mathbf{x}^T \mathbf{h}[\Delta\tau_n])^2 + \frac{\lambda}{2} \|\mathbf{h}\|_2^2 \\
 &= \min_{\mathbf{h}} \frac{1}{2} \|\mathbf{y} - \mathbf{C}(\mathbf{h}^T) \mathbf{x}\|_2^2 + \frac{\lambda}{2} \|\mathbf{h}\|_2^2 \\
 &= \min_{\mathbf{h}} \frac{1}{2} \|\mathbf{y} - \bar{\mathbf{h}} \ast \mathbf{x}\|_2^2 + \frac{\lambda}{2} \|\bar{\mathbf{h}}\|_2^2.
 \end{aligned} \tag{33}$$

Comment 4: $\mathbf{C}(\mathbf{h}^T) \mathbf{x}$ and $\mathbf{C}(\mathbf{x}^T) \mathbf{h}$ are confusing in some studies. There is a reverse relationship between $\mathbf{C}(\mathbf{h}^T) \mathbf{x}$ and $\mathbf{C}(\mathbf{x}^T) \mathbf{h}$, that is, $\mathbf{C}(\mathbf{h}^T) \mathbf{x} = \overline{\mathbf{C}(\mathbf{x}^T) \mathbf{h}}$.

Comment 5: The definition of the correlation operation $\mathbf{x} \star \mathbf{h}$ differs across studies in the literature. If the element of the correlation operator is defined as $(\mathbf{x} \star \mathbf{h})(n) = \mathbf{x}^T \mathbf{h}[\Delta\tau_n]$, we obtain $\mathbf{x} \star \mathbf{h} = \mathbf{C}(\mathbf{h}^T) \mathbf{x} = \overline{\mathbf{C}(\mathbf{x}^T) \mathbf{h}}$.

4. Equivalence of the two filter-solving methods

4.1. Filter solution based on the diagonalization property of the circulant matrix

In the spatial domain, by computing the first-order derivative of \mathbf{h} in Eq (30) and setting it equal to zero, we obtain

$$\frac{dE(\mathbf{h})}{d\mathbf{h}} = \mathbf{C}(\mathbf{x}^T)^H (\mathbf{C}(\mathbf{x}^T) \mathbf{h} - \bar{\mathbf{y}}) + \lambda \mathbf{h} = \mathbf{0}. \tag{34}$$

Then, the spatial domain optimal solution for \mathbf{h} is given by

$$\begin{aligned}
 \mathbf{h} &= \left(\mathbf{C}(\mathbf{x}^T)^H \mathbf{C}(\mathbf{x}^T) + \lambda \right)^{-1} \mathbf{C}(\mathbf{x}^T)^H \bar{\mathbf{y}} \\
 &= \left(\mathbf{C}(\mathbf{x}^T)^T \mathbf{C}(\mathbf{x}^T) + \lambda \right)^{-1} \mathbf{C}(\mathbf{x}^T)^T \bar{\mathbf{y}}.
 \end{aligned} \tag{35}$$

Because the introduction of a circulant matrix generates numerous virtual samples, considerable computation is required. The sample matrix can be transformed into a diagonal matrix for processing based on the diagonalization property of the row-vector-based circulant matrix. This method significantly accelerates the matrix calculations and reduces the computational complexity of directly computing solutions in the spatial domain, that is

$$\begin{aligned}
\mathbf{h} &= (\mathbf{C}(\mathbf{x}^T)^T \mathbf{C}(\mathbf{x}^T) + \lambda)^{-1} \mathbf{C}(\mathbf{x}^T)^T \bar{\mathbf{y}} \\
&= (\mathbf{F}_n \mathbf{Diag}(\hat{\mathbf{x}} \odot \hat{\mathbf{x}}^*) \mathbf{F}_n^H + \lambda \mathbf{F}_n \mathbf{Diag}(\boldsymbol{\delta}) \mathbf{F}_n^H)^{-1} \mathbf{C}(\mathbf{x}) \bar{\mathbf{y}} \\
&= (\mathbf{F}_n \mathbf{Diag}(\hat{\mathbf{x}} \odot \hat{\mathbf{x}}^*) \mathbf{F}_n^H + \lambda \mathbf{F}_n \mathbf{Diag}(\boldsymbol{\delta}) \mathbf{F}_n^H)^{-1} \mathbf{C}(\mathbf{x}) \bar{\mathbf{y}} \\
&= (\mathbf{F}_n (\mathbf{Diag}(\hat{\mathbf{x}} \odot \hat{\mathbf{x}}^*) + \lambda \mathbf{Diag}(\boldsymbol{\delta})) \mathbf{F}_n^H)^{-1} \mathbf{C}(\mathbf{x}) \bar{\mathbf{y}} \\
&= \mathbf{F}_n \mathbf{Diag} \left(\frac{1}{\hat{\mathbf{x}} \odot \hat{\mathbf{x}}^* + \lambda \boldsymbol{\delta}} \right) \mathbf{F}_n^H \mathbf{F}_n \mathbf{Diag}(\hat{\mathbf{x}}^*) \mathbf{F}_n^H \bar{\mathbf{y}} \\
&= \mathbf{F}_n \mathbf{Diag} \left(\frac{\hat{\mathbf{x}}^*}{\hat{\mathbf{x}} \odot \hat{\mathbf{x}}^* + \lambda \boldsymbol{\delta}} \right) \mathbf{F}_n^H \bar{\mathbf{y}} \\
&= \mathbf{C}(\mathbf{u}^T) \bar{\mathbf{y}} \Big|_{u = \text{ifft}_1 \left\{ \frac{\hat{\mathbf{x}}^*}{\hat{\mathbf{x}} \odot \hat{\mathbf{x}}^* + \lambda \boldsymbol{\delta}} \right\}} = \bar{\mathbf{u}} * \bar{\mathbf{y}},
\end{aligned} \tag{36}$$

where $\boldsymbol{\delta}$ is the column vector whose elements are all 1, that is, $\boldsymbol{\delta} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^{N \times 1}$.

According to the convolution theorem, Eq (36) can be transformed into the frequency domain to calculate

$$\mathbf{h} = \text{real} \left(\text{ifft}_1 (\hat{\mathbf{u}}^* \odot \hat{\mathbf{y}}^*) \right) = \text{real} \left(\text{ifft}_1 \left(\frac{\hat{\mathbf{x}} \odot \hat{\mathbf{y}}^*}{\hat{\mathbf{x}} \odot \hat{\mathbf{x}}^* + \lambda \boldsymbol{\delta}} \right) \right). \tag{37}$$

For a new test sample \mathbf{z} , we have

$$\bar{\mathbf{r}} = \mathbf{C}(\mathbf{z}^T) \mathbf{h}, \tag{38}$$

where $\bar{\mathbf{r}}$ is the reverse signal of the spatial domain response \mathbf{r} .

By reversing both sides of Eq (38), the spatial domain response \mathbf{r} is expressed as

$$\mathbf{r} = \overline{\mathbf{C}(\mathbf{z}^T) \mathbf{h}} = \mathbf{C}(\mathbf{h}^T) \mathbf{z}. \tag{39}$$

As $\mathbf{C}(\mathbf{z}^T) \mathbf{h} = \bar{\mathbf{z}} * \mathbf{h}$, Eq (39) can be written in the convolutional form, as shown in Eq (40).

$$\mathbf{r} = \overline{\mathbf{C}(\mathbf{z}^T) \mathbf{h}} = \overline{\bar{\mathbf{z}} * \mathbf{h}} = \mathbf{z} * \bar{\mathbf{h}}. \tag{40}$$

According to Eqs (19) and (24), Eq. (40) can be rewritten as

$$\mathbf{r} = \text{real} \left(\text{ifft}_1 (\hat{\mathbf{h}}^* \odot \hat{\mathbf{z}}) \right). \tag{41}$$

4.2. Filter solution based on the convolution theorem

In the frequency domain, by computing the first-order derivative of $\hat{\mathbf{h}}^*$ in Eq (32) and setting it to zero, that is $\frac{dE(\mathbf{h})}{d\hat{\mathbf{h}}^*} = \frac{2\hat{\mathbf{x}}^* \odot (\hat{\mathbf{h}}^* \odot \hat{\mathbf{x}} - \hat{\mathbf{y}})}{N} + \frac{2\lambda \hat{\mathbf{h}}^*}{N} = \mathbf{0}$, we obtain

$$\hat{\mathbf{h}}^* = \frac{\hat{\mathbf{y}} \odot \hat{\mathbf{x}}^*}{\hat{\mathbf{x}}^* \odot \hat{\mathbf{x}} + \lambda}, \tag{42}$$

where the sign of the division in Eq (42) denotes entry-wise division.

For the new sample \hat{z} , the corresponding spatial response is

$$r = \text{real}\left(\text{ifft}_1\left(\hat{h}^* \odot \hat{z}\right)\right). \quad (43)$$

Comment 6: Eqs (41) and (43) show that the results obtained by solving the filter using the diagonalization property of the row-vector-based circulant matrix and the convolution theorem are completely consistent.

5. Difference and connection between the one-dimensional and two-dimensional filter methods

For the object tracking techniques, the image being processed is a two-dimensional signal, whereas all the signals discussed in the previous section are one-dimensional. Hence, we generalize the one-dimensional signal convolution form in Eq (20) into a two-dimensional convolution form, as shown in Eq (44)

$$\text{Conv}_2(\bar{I}_1, I_2) = \text{mat}\left(\text{Cat}\left(\text{vec}(I_1[\Delta\tau_c, \Delta\tau_r])^T\right)\text{vec}(I_2)\right) = \text{real}\left(\text{ifft}_2\left(\hat{I}_1^* \odot \hat{I}_2\right)\right), \quad (44)$$

where mat is an operator that transforms column vectors into matrices, Cat is an operator that stacks the row vectors into a matrix, vec is an operator that transforms a matrix into a column vector, and Conv_2 is a two-dimensional convolution operator. The elements of the c th row and r th column of $\text{Conv}_2(\bar{I}_1, I_2)$ are $\langle \text{vec}(I_1[\Delta\tau_c, \Delta\tau_r]), \text{vec}(I_2) \rangle = \text{vec}(I_1[\Delta\tau_c, \Delta\tau_r])^T \text{vec}(I_2)$, and $I_1 \in \mathbb{R}^{\sqrt{N} \times \sqrt{N}}$, $I_2 \in \mathbb{R}^{\sqrt{N} \times \sqrt{N}}$, and \sqrt{N} are integers. $r = 0, 1, \dots, \sqrt{N} - 1$ represents the number of row cyclic shifts, and $c = 0, 1, \dots, \sqrt{N} - 1$ represents the number of column cyclic shifts. Moreover, $I_1[\Delta\tau_c, \Delta\tau_r] \in \mathbb{R}^{\sqrt{N} \times \sqrt{N}}$ denotes a matrix obtained by the first cyclic shift I_1 , row-by-row, using r units to obtain an intermediate matrix, and then performing a cyclic shift of the intermediate matrix, column-by-column, using c units. Finally, \bar{I}_1 is the reverse matrix of I_1 , and the specific calculation is as follows: I_1 first reverses the original matrix, row-by-row, to obtain an intermediate matrix. Then, it reverses the intermediate matrix, column-by-column, to obtain \bar{I}_1 . For example, $I_1 = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ is performed,

row-by-row, on the original matrix to perform the reverse operation to obtain the intermediate matrix $I_r = \begin{bmatrix} 1 & 3 & 2 \\ 4 & 6 & 5 \\ 7 & 9 & 8 \end{bmatrix}$. Then, it is carried out on the intermediate matrix I_r , column-by-column, to perform

the reverse operation. This is followed by $\bar{I}_1 = \begin{bmatrix} 1 & 3 & 2 \\ 7 & 9 & 8 \\ 4 & 6 & 5 \end{bmatrix}$. Here,

$\text{mat}\left(\text{Cat}\left(\text{vec}(I_1[\Delta\tau_c, \Delta\tau_r])^T\right)\text{vec}(I_2)\right) \in \mathbb{R}^{\sqrt{N} \times \sqrt{N}}$, $\text{Conv}_2(\bar{I}_1, I_2) \in \mathbb{R}^{\sqrt{N} \times \sqrt{N}}$ is the two-dimensional convolution of images \bar{I}_1 and I_2 , and ifft_2 is a two-dimensional Fourier inverse transform operator.

Comment 7: If $\text{Conv}_2(\bar{I}_1, I_2)$ is calculated in the spatial domain, that is, $\text{mat}\left(\text{Cat}\left(\text{vec}(I_1[\Delta\tau_c, \Delta\tau_r])^T\right)\text{vec}(I_2)\right)$, the occupied memory space is $\mathcal{O}(N^2 + N)$, and the computational multiplicative complexity is $\mathcal{O}(N^2)$. If the convolution theorem is introduced, then the convolution in the spatial domain can be transformed into an entry-wise multiplication operation and a Fourier inverse

transform in the frequency domain, that is, $\text{Conv}_2(\bar{I}_1, I_2) = \text{real}(\text{ifft}_2(\hat{I}_1^* \odot \hat{I}_2))$. The memory space occupied by the algorithm is $4N$, and the computational multiplication complexity is $\mathcal{O}(8N \log_2 N + 4N)$. A detailed calculation process of the two-dimensional filter $\mathbf{R}_{2D} = \text{Conv}_2(I_1, I_2)$

$$(I_1 = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}, I_2 = \begin{bmatrix} 0.1 & 1 & 1 \\ 1 & 0.1 & 1 \\ 1 & 1 & 0.1 \end{bmatrix}) \text{ is listed in Table 2.}$$

Table 2. Two-dimensional filter calculation process.

Element of \mathbf{R}_{2D}	Stage 1: Reverse	Stage 2: Cyclic shift	Stage 3: Multiplication	Stage 4: Summation
$\mathbf{R}_{2D}(1,1)$	$\bar{I}_1 = \begin{bmatrix} 1 & 7 & 4 \\ 3 & 9 & 6 \\ 2 & 8 & 5 \end{bmatrix}$	$\bar{I}_1[\Delta\tau_0, \Delta\tau_0] = \begin{bmatrix} 1 & 7 & 4 \\ 3 & 9 & 6 \\ 2 & 8 & 5 \end{bmatrix}$	$\begin{aligned} & \bar{I}_1[\Delta\tau_0, \Delta\tau_0] \odot I_2 \\ & = \begin{bmatrix} 1 & 7 & 4 \\ 3 & 9 & 6 \\ 2 & 8 & 5 \end{bmatrix} \odot \begin{bmatrix} 0.1 & 1 & 1 \\ 1 & 0.1 & 1 \\ 1 & 1 & 0.1 \end{bmatrix} \\ & = \begin{bmatrix} 0.1 & 7 & 4 \\ 3 & 0.9 & 6 \\ 2 & 8 & 0.5 \end{bmatrix} \end{aligned}$	31.5
$\mathbf{R}_{2D}(1,2)$	$\bar{I}_1 = \begin{bmatrix} 1 & 7 & 4 \\ 3 & 9 & 6 \\ 2 & 8 & 5 \end{bmatrix}$	$\bar{I}_1[\Delta\tau_0, \Delta\tau_1] = \begin{bmatrix} 4 & 1 & 7 \\ 6 & 3 & 9 \\ 5 & 2 & 8 \end{bmatrix}$	$\begin{aligned} & \bar{I}_1[\Delta\tau_0, \Delta\tau_1] \odot I_2 \\ & = \begin{bmatrix} 4 & 1 & 7 \\ 6 & 3 & 9 \\ 5 & 2 & 8 \end{bmatrix} \odot \begin{bmatrix} 0.1 & 1 & 1 \\ 1 & 0.1 & 1 \\ 1 & 1 & 0.1 \end{bmatrix} \\ & = \begin{bmatrix} 0.4 & 1 & 7 \\ 6 & 0.3 & 9 \\ 5 & 2 & 0.8 \end{bmatrix} \end{aligned}$	31.5
$\mathbf{R}_{2D}(1,3)$	$\bar{I}_1 = \begin{bmatrix} 1 & 7 & 4 \\ 3 & 9 & 6 \\ 2 & 8 & 5 \end{bmatrix}$	$\bar{I}_1[\Delta\tau_0, \Delta\tau_2] = \begin{bmatrix} 7 & 4 & 1 \\ 9 & 6 & 3 \\ 8 & 5 & 2 \end{bmatrix}$	$\begin{aligned} & \bar{I}_1[\Delta\tau_0, \Delta\tau_2] \odot I_2 \\ & = \begin{bmatrix} 7 & 4 & 1 \\ 9 & 6 & 3 \\ 8 & 5 & 2 \end{bmatrix} \odot \begin{bmatrix} 0.1 & 1 & 1 \\ 1 & 0.1 & 1 \\ 1 & 1 & 0.1 \end{bmatrix} \\ & = \begin{bmatrix} 0.7 & 4 & 1 \\ 9 & 0.6 & 3 \\ 8 & 5 & 0.2 \end{bmatrix} \end{aligned}$	31.5
$\mathbf{R}_{2D}(2,1)$	$\bar{I}_1 = \begin{bmatrix} 1 & 7 & 4 \\ 3 & 9 & 6 \\ 2 & 8 & 5 \end{bmatrix}$	$\bar{I}_1[\Delta\tau_1, \Delta\tau_0] = \begin{bmatrix} 2 & 8 & 5 \\ 1 & 7 & 4 \\ 3 & 9 & 6 \end{bmatrix}$	$\begin{aligned} & \bar{I}_1[\Delta\tau_1, \Delta\tau_0] \odot I_2 \\ & = \begin{bmatrix} 2 & 8 & 5 \\ 1 & 7 & 4 \\ 3 & 9 & 6 \end{bmatrix} \odot \begin{bmatrix} 0.1 & 1 & 1 \\ 1 & 0.1 & 1 \\ 1 & 1 & 0.1 \end{bmatrix} \\ & = \begin{bmatrix} 0.2 & 8 & 5 \\ 1 & 0.7 & 4 \\ 3 & 9 & 0.6 \end{bmatrix} \end{aligned}$	31.5

Continued on next page

Element of \mathbf{R}_{2D}	Stage 1: Reverse	Stage 2: Cyclic shift	Stage 3: Multiplication	Stage 4: Summation
$\mathbf{R}_{2D}(2,2)$	$\bar{I}_1 = \begin{bmatrix} 1 & 7 & 4 \\ 3 & 9 & 6 \\ 2 & 8 & 5 \end{bmatrix}$	$\bar{I}_1[\Delta\tau_1, \Delta\tau_1] = \begin{bmatrix} 5 & 2 & 8 \\ 4 & 1 & 7 \\ 6 & 3 & 9 \end{bmatrix}$	$\begin{aligned} & \bar{I}_1[\Delta\tau_1, \Delta\tau_1] \odot I_2 \\ & = \begin{bmatrix} 5 & 2 & 8 \\ 4 & 1 & 7 \\ 6 & 3 & 9 \end{bmatrix} \odot \begin{bmatrix} 0.1 & 1 & 1 \\ 1 & 0.1 & 1 \\ 1 & 1 & 0.1 \end{bmatrix} \\ & = \begin{bmatrix} 0.5 & 2 & 8 \\ 4 & 0.1 & 7 \\ 6 & 3 & 0.9 \end{bmatrix} \end{aligned}$	31.5
$\mathbf{R}_{2D}(2,3)$	$\bar{I}_1 = \begin{bmatrix} 1 & 7 & 4 \\ 3 & 9 & 6 \\ 2 & 8 & 5 \end{bmatrix}$	$\bar{I}_1[\Delta\tau_1, \Delta\tau_2] = \begin{bmatrix} 8 & 5 & 2 \\ 7 & 4 & 1 \\ 9 & 6 & 3 \end{bmatrix}$	$\begin{aligned} & \bar{I}_1[\Delta\tau_1, \Delta\tau_2] \odot I_2 \\ & = \begin{bmatrix} 8 & 5 & 2 \\ 7 & 4 & 1 \\ 9 & 6 & 3 \end{bmatrix} \odot \begin{bmatrix} 0.1 & 1 & 1 \\ 1 & 0.1 & 1 \\ 1 & 1 & 0.1 \end{bmatrix} \\ & = \begin{bmatrix} 0.8 & 5 & 2 \\ 7 & 0.4 & 1 \\ 9 & 6 & 0.3 \end{bmatrix} \end{aligned}$	31.5
$\mathbf{R}_{2D}(3,1)$	$\bar{I}_1 = \begin{bmatrix} 1 & 7 & 4 \\ 3 & 9 & 6 \\ 2 & 8 & 5 \end{bmatrix}$	$\bar{I}_1[\Delta\tau_2, \Delta\tau_0] = \begin{bmatrix} 3 & 9 & 6 \\ 2 & 8 & 5 \\ 1 & 7 & 4 \end{bmatrix}$	$\begin{aligned} & \bar{I}_1[\Delta\tau_2, \Delta\tau_0] \odot I_2 \\ & = \begin{bmatrix} 3 & 9 & 6 \\ 2 & 8 & 5 \\ 1 & 7 & 4 \end{bmatrix} \odot \begin{bmatrix} 0.1 & 1 & 1 \\ 1 & 0.1 & 1 \\ 1 & 1 & 0.1 \end{bmatrix} \\ & = \begin{bmatrix} 0.3 & 9 & 6 \\ 2 & 0.8 & 5 \\ 1 & 7 & 0.4 \end{bmatrix} \end{aligned}$	31.5
$\mathbf{R}_{2D}(3,2)$	$\bar{I}_1 = \begin{bmatrix} 1 & 7 & 4 \\ 3 & 9 & 6 \\ 2 & 8 & 5 \end{bmatrix}$	$\bar{I}_1[\Delta\tau_2, \Delta\tau_1] = \begin{bmatrix} 6 & 3 & 9 \\ 5 & 2 & 8 \\ 4 & 1 & 7 \end{bmatrix}$	$\begin{aligned} & \bar{I}_1[\Delta\tau_2, \Delta\tau_1] \odot I_2 \\ & = \begin{bmatrix} 6 & 3 & 9 \\ 5 & 2 & 8 \\ 4 & 1 & 7 \end{bmatrix} \odot \begin{bmatrix} 0.1 & 1 & 1 \\ 1 & 0.1 & 1 \\ 1 & 1 & 0.1 \end{bmatrix} \\ & = \begin{bmatrix} 0.6 & 3 & 9 \\ 5 & 0.2 & 8 \\ 4 & 1 & 0.7 \end{bmatrix} \end{aligned}$	31.5
$\mathbf{R}_{2D}(3,3)$	$\bar{I}_1 = \begin{bmatrix} 1 & 7 & 4 \\ 3 & 9 & 6 \\ 2 & 8 & 5 \end{bmatrix}$	$\bar{I}_1[\Delta\tau_2, \Delta\tau_2] = \begin{bmatrix} 9 & 6 & 3 \\ 8 & 5 & 2 \\ 7 & 4 & 1 \end{bmatrix}$	$\begin{aligned} & \bar{I}_1[\Delta\tau_2, \Delta\tau_2] \odot I_2 \\ & = \begin{bmatrix} 9 & 6 & 3 \\ 8 & 5 & 2 \\ 7 & 4 & 1 \end{bmatrix} \odot \begin{bmatrix} 0.1 & 1 & 1 \\ 1 & 0.1 & 1 \\ 1 & 1 & 0.1 \end{bmatrix} \\ & = \begin{bmatrix} 0.9 & 6 & 3 \\ 8 & 0.5 & 2 \\ 7 & 4 & 0.1 \end{bmatrix} \end{aligned}$	31.5

Table 3. One-dimensional filter calculation process.

Element of r_{1D}	Stage 1: Reverse	Stage 2: Cyclic shift	Stage 3: Multiplication	Stage 4: Summation
$r_{1D}(1)$	$\bar{i}_1 = \begin{pmatrix} 1 \\ 9 \\ 8 \\ 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \end{pmatrix}$	$\bar{i}_1[\Delta\tau_0] = \begin{pmatrix} 1 \\ 9 \\ 8 \\ 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \end{pmatrix}$	$\bar{i}_1[\Delta\tau_0] \odot i_2 = \begin{pmatrix} 1 \\ 9 \\ 8 \\ 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \end{pmatrix} \odot \begin{pmatrix} 0.1 \\ 1 \\ 1 \\ 1 \\ 0.1 \\ 1 \\ 1 \\ 1 \\ 0.1 \end{pmatrix} = \begin{pmatrix} 0.1 \\ 9 \\ 8 \\ 7 \\ 0.6 \\ 5 \\ 4 \\ 3 \\ 0.2 \end{pmatrix}$	36.9
$r_{1D}(2)$	$\bar{i}_1 = \begin{pmatrix} 1 \\ 9 \\ 8 \\ 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \end{pmatrix}$	$\bar{i}_1[\Delta\tau_1] = \begin{pmatrix} 2 \\ 1 \\ 9 \\ 8 \\ 7 \\ 6 \\ 5 \\ 4 \\ 3 \end{pmatrix}$	$\bar{i}_1[\Delta\tau_1] \odot i_2 = \begin{pmatrix} 2 \\ 1 \\ 9 \\ 8 \\ 7 \\ 6 \\ 5 \\ 4 \\ 3 \end{pmatrix} \odot \begin{pmatrix} 0.1 \\ 1 \\ 1 \\ 1 \\ 0.1 \\ 1 \\ 1 \\ 1 \\ 0.3 \end{pmatrix} = \begin{pmatrix} 0.2 \\ 1 \\ 9 \\ 8 \\ 0.7 \\ 6 \\ 5 \\ 4 \\ 0.3 \end{pmatrix}$	34.2
$r_{1D}(3)$	$\bar{i}_1 = \begin{pmatrix} 1 \\ 9 \\ 8 \\ 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \end{pmatrix}$	$\bar{i}_1[\Delta\tau_2] = \begin{pmatrix} 3 \\ 2 \\ 1 \\ 9 \\ 8 \\ 7 \\ 6 \\ 5 \\ 4 \end{pmatrix}$	$\bar{i}_1[\Delta\tau_2] \odot i_2 = \begin{pmatrix} 3 \\ 2 \\ 1 \\ 9 \\ 8 \\ 7 \\ 6 \\ 5 \\ 4 \end{pmatrix} \odot \begin{pmatrix} 0.1 \\ 1 \\ 1 \\ 1 \\ 0.1 \\ 1 \\ 1 \\ 1 \\ 0.4 \end{pmatrix} = \begin{pmatrix} 0.3 \\ 2 \\ 1 \\ 9 \\ 0.8 \\ 7 \\ 6 \\ 5 \\ 0.4 \end{pmatrix}$	31.5
$r_{1D}(4)$	$\bar{i}_1 = \begin{pmatrix} 1 \\ 9 \\ 8 \\ 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \end{pmatrix}$	$\bar{i}_1[\Delta\tau_3] = \begin{pmatrix} 4 \\ 3 \\ 2 \\ 1 \\ 9 \\ 8 \\ 7 \\ 6 \\ 5 \end{pmatrix}$	$\bar{i}_1[\Delta\tau_3] \odot i_2 = \begin{pmatrix} 4 \\ 3 \\ 2 \\ 1 \\ 9 \\ 8 \\ 7 \\ 6 \\ 5 \end{pmatrix} \odot \begin{pmatrix} 0.1 \\ 1 \\ 1 \\ 1 \\ 0.1 \\ 1 \\ 1 \\ 1 \\ 0.5 \end{pmatrix} = \begin{pmatrix} 0.4 \\ 3 \\ 2 \\ 1 \\ 0.9 \\ 8 \\ 7 \\ 6 \\ 0.5 \end{pmatrix}$	28.8

Continued on next page

Element of r_{1D}	Stage 1: Reverse	Stage 2: Cyclic shift	Stage 3: Multiplication	Stage 4: Summation
$r_{1D}(5)$	$\bar{i}_1 = \begin{pmatrix} 1 \\ 9 \\ 8 \\ 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \end{pmatrix}$	$\bar{i}_1[\Delta\tau_4] = \begin{pmatrix} 5 \\ 4 \\ 3 \\ 2 \\ 1 \\ 9 \\ 8 \\ 7 \\ 6 \end{pmatrix}$	$\bar{i}_1[\Delta\tau_4] \odot i_2 = \begin{pmatrix} 5 \\ 4 \\ 3 \\ 2 \\ 1 \\ 9 \\ 8 \\ 7 \\ 6 \end{pmatrix} \odot \begin{pmatrix} 0.1 \\ 1 \\ 1 \\ 1 \\ 0.1 \\ 1 \\ 1 \\ 1 \\ 0.1 \end{pmatrix} = \begin{pmatrix} 0.5 \\ 4 \\ 3 \\ 2 \\ 0.1 \\ 9 \\ 8 \\ 7 \\ 0.6 \end{pmatrix}$	34.2
$r_{1D}(6)$	$\bar{i}_1 = \begin{pmatrix} 1 \\ 9 \\ 8 \\ 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \end{pmatrix}$	$\bar{i}_1[\Delta\tau_5] = \begin{pmatrix} 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \\ 9 \\ 8 \\ 7 \end{pmatrix}$	$\bar{i}_1[\Delta\tau_5] \odot i_2 = \begin{pmatrix} 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \\ 9 \\ 8 \\ 7 \end{pmatrix} \odot \begin{pmatrix} 0.1 \\ 1 \\ 1 \\ 1 \\ 0.1 \\ 1 \\ 1 \\ 1 \\ 0.1 \end{pmatrix} = \begin{pmatrix} 0.6 \\ 5 \\ 4 \\ 3 \\ 0.2 \\ 1 \\ 9 \\ 8 \\ 0.7 \end{pmatrix}$	31.5
$r_{1D}(7)$	$\bar{i}_1 = \begin{pmatrix} 1 \\ 9 \\ 8 \\ 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \end{pmatrix}$	$\bar{i}_1[\Delta\tau_6] = \begin{pmatrix} 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \\ 9 \\ 8 \end{pmatrix}$	$\bar{i}_1[\Delta\tau_6] \odot i_2 = \begin{pmatrix} 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \\ 9 \\ 8 \end{pmatrix} \odot \begin{pmatrix} 0.1 \\ 1 \\ 1 \\ 1 \\ 0.1 \\ 1 \\ 1 \\ 1 \\ 0.1 \end{pmatrix} = \begin{pmatrix} 0.7 \\ 6 \\ 5 \\ 4 \\ 0.3 \\ 2 \\ 1 \\ 9 \\ 0.8 \end{pmatrix}$	28.8
$r_{1D}(8)$	$\bar{i}_1 = \begin{pmatrix} 1 \\ 9 \\ 8 \\ 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \end{pmatrix}$	$\bar{i}_1[\Delta\tau_7] = \begin{pmatrix} 8 \\ 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \\ 9 \end{pmatrix}$	$\bar{i}_1[\Delta\tau_7] \odot i_2 = \begin{pmatrix} 8 \\ 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \\ 9 \end{pmatrix} \odot \begin{pmatrix} 0.1 \\ 1 \\ 1 \\ 1 \\ 0.1 \\ 1 \\ 1 \\ 1 \\ 0.1 \end{pmatrix} = \begin{pmatrix} 0.8 \\ 7 \\ 6 \\ 5 \\ 0.4 \\ 3 \\ 2 \\ 1 \\ 0.9 \end{pmatrix}$	26.1

Continued on next page

Element of r_{1D}	Stage 1: Reverse	Stage 2: Cyclic shift	Stage 3: Multiplication	Stage 4: Summation
$r_{1D}(9)$	$\bar{i}_1 = \begin{pmatrix} 1 \\ 9 \\ 8 \\ 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \end{pmatrix}$	$\bar{i}_1[\Delta\tau_8] = \begin{pmatrix} 9 \\ 8 \\ 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \end{pmatrix}$	$\bar{i}_1[\Delta\tau_8] \odot i_2 = \begin{pmatrix} 9 \\ 8 \\ 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \end{pmatrix} \odot \begin{pmatrix} 0.1 \\ 1 \\ 1 \\ 1 \\ 0.1 \\ 1 \\ 1 \\ 1 \\ 0.1 \end{pmatrix} = \begin{pmatrix} 0.9 \\ 8 \\ 7 \\ 6 \\ 0.5 \\ 4 \\ 3 \\ 2 \\ 0.1 \end{pmatrix}$	31.5

A detailed calculation process of the one-dimensional filter $r_{1D} = \text{vec}(I_1) * \text{vec}(I_2) = i_1 * i_2$, where

$$i_1 = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{pmatrix}, \quad i_2 = \begin{pmatrix} 0.1 \\ 1 \\ 1 \\ 1 \\ 0.1 \\ 1 \\ 1 \\ 1 \\ 0.1 \end{pmatrix}, \quad \text{and } \bar{i}_1[\Delta\tau_n] \text{ is the } \bar{i}_1 \text{ cyclic shift } n \text{ times, which is listed in Table 3.}$$

Table 4 presents the occupied memory space and the complexity of the floating-point multiplication operation by the $\text{Cat}(\text{vec}(I_1[\Delta\tau_c, \Delta\tau_r])^T) \text{vec}(I_2)$ and $\text{real}(\text{ifft}_2(\hat{I}_1^* \odot \hat{I}_2))$ operations. Table 4 shows that the memory footprint and computational complexity of the one-dimensional and two-dimensional operations are completely consistent. The computational complexity of the $\text{real}(\text{ifft}_2(\hat{I}_1^* \odot \hat{I}_2))$ operation is much smaller than that of the $\text{Cat}(\text{vec}(I_1[\Delta\tau_c, \Delta\tau_r])^T) \text{vec}(I_2)$ operation when the image size is large.

Table 4. Memory footprint and computational complexity analysis of the two-dimensional operation.

Operation (image size $\mathbb{R}^{\sqrt{N} \times \sqrt{N}}$)	Memory space occupied/floating-point unit	Complexity of floating-point multiplication operation
$\text{Cat}(\text{vec}(I_1[\Delta\tau_c, \Delta\tau_r])^T) \text{vec}(I_2)$	$N^2 + N$	$\mathcal{O}(N^2)$
$\text{real}(\text{ifft}_2(\hat{I}_1^* \odot \hat{I}_2))$	$4N$	$\mathcal{O}(8N \log_2 N + 4N)$

The one-dimensional and two-dimensional filters are equivalent in estimating the object's position. However, subtle differences exist between the one-dimensional and two-dimensional filter responses owing to the inconsistency in the receptive fields and periodic boundary conditions between the one-dimensional and two-dimensional convolutions. The receptive field of the one-dimensional convolution operation has only one dimension, whereas that of the two-dimensional convolution operation has two dimensions. By contrast, the periodic boundary-filling signal of a two-dimensional

signal is a two-dimensional signal. However, the periodic boundary-filling signal from a two-dimensional image columnized into a one-dimensional signal is a one-dimensional signal. Thus, a difference exists in the data involved in the operation at the same spatial location between the one-dimensional and two-dimensional convolutions. Consequently, subtle differences occur in the results between the two filters.

When we model the correlation filter tracking problem in matrix form, the mathematical formula is modeled as follows

$$E(\mathbf{H}) = \min_{\mathbf{H}} \frac{1}{2} \|\mathbf{Y} - \mathbf{X} \star_2 \mathbf{H}\|_2^2 + \frac{\lambda}{2} \|\mathbf{H}\|_2^2, \quad (45)$$

where \star_2 represents a two-dimensional correlation operator, $(\mathbf{X} \star_2 \mathbf{H})(n, k) = \sum_{m=0}^{\sqrt{N}-1} \sum_{l=0}^{\sqrt{N}-1} \mathbf{X}(m, l) \mathbf{H}(n+m, k+l) = \text{Conv}_2(\bar{\mathbf{H}}, \mathbf{X})$, $\bar{\mathbf{H}}$ is the reversed two-dimensional signal of \mathbf{H} , $\bar{\mathbf{H}}$ satisfies two dimensional periodic boundary conditions, \mathbf{X} , \mathbf{Y} and \mathbf{H} are the matrix form of \mathbf{x} , \mathbf{y} and \mathbf{h} .

Rewrite the above formula into convolution form and we have

$$E(\mathbf{H}) = \min_{\mathbf{H}} \frac{1}{2} \|\mathbf{Y} - \text{Conv}_2(\bar{\mathbf{H}}, \mathbf{X})\|_2^2 + \frac{\lambda}{2} \|\mathbf{H}\|_2^2, \quad (46)$$

where $\bar{\mathbf{H}}$ satisfies $\text{fft}_2(\bar{\mathbf{H}}) = \hat{\mathbf{H}}^*$, where $\hat{\mathbf{H}}$ denotes the spectrum of \mathbf{H} , fft_2 represents a two-dimensional fast Fourier transform operator, and $\hat{\mathbf{H}}^*$ is the conjugate matrix of $\hat{\mathbf{H}}$.

According to the convolution theorem, the above expression can be further arranged into the frequency domain form, namely

$$E(\hat{\mathbf{H}}^*) = \min_{\hat{\mathbf{H}}^*} \frac{1}{2N} \|\hat{\mathbf{Y}} - \hat{\mathbf{H}}^* \odot \hat{\mathbf{X}}\|_2^2 + \frac{\lambda}{2N} \|\hat{\mathbf{H}}^*\|_2^2. \quad (47)$$

Set $\frac{dE(\hat{\mathbf{H}}^*)}{d\hat{\mathbf{H}}^*} = \mathbf{0}$, we have

$$\frac{\hat{\mathbf{X}}^* \odot (\hat{\mathbf{H}}^* \odot \hat{\mathbf{X}} - \hat{\mathbf{Y}}) + \lambda \hat{\mathbf{H}}^*}{N} = \mathbf{0}. \quad (48)$$

Then, the filter in the matrix form is calculated by

$$\hat{\mathbf{H}}^* = \frac{\hat{\mathbf{Y}} \odot \hat{\mathbf{X}}^*}{\hat{\mathbf{X}}^* \odot \hat{\mathbf{X}} + \lambda}. \quad (49)$$

Figure 1(a) shows the training sample, and Figure 1(b) illustrates the desired response. Figure 1(c) shows the test sample obtained by the cyclic shift of the training sample. Figure 1(d) presents the plot of the two-dimensional filter's (the two-dimensional filter is calculated by $\hat{\mathbf{H}}^* = \frac{\hat{\mathbf{Y}} \odot \hat{\mathbf{X}}^*}{\hat{\mathbf{X}}^* \odot \hat{\mathbf{X}} + \lambda}$) response. Figure 1(e) shows the plot of the one-dimensional filter's (the one-dimensional filter is calculated by $\hat{\mathbf{h}}^* = \frac{\hat{\mathbf{y}} \odot \hat{\mathbf{x}}^*}{\hat{\mathbf{x}}^* \odot \hat{\mathbf{x}} + \lambda}$) response. Figure 1(f) shows the difference between the one-dimensional and two-dimensional filter responses. The results show that the one-dimensional and two-dimensional filters are equivalent in positioning but are not completely consistent. When the image is cyclically shifted

to the edge, a significant difference is observed between the one-dimensional and two-dimensional filter responses.

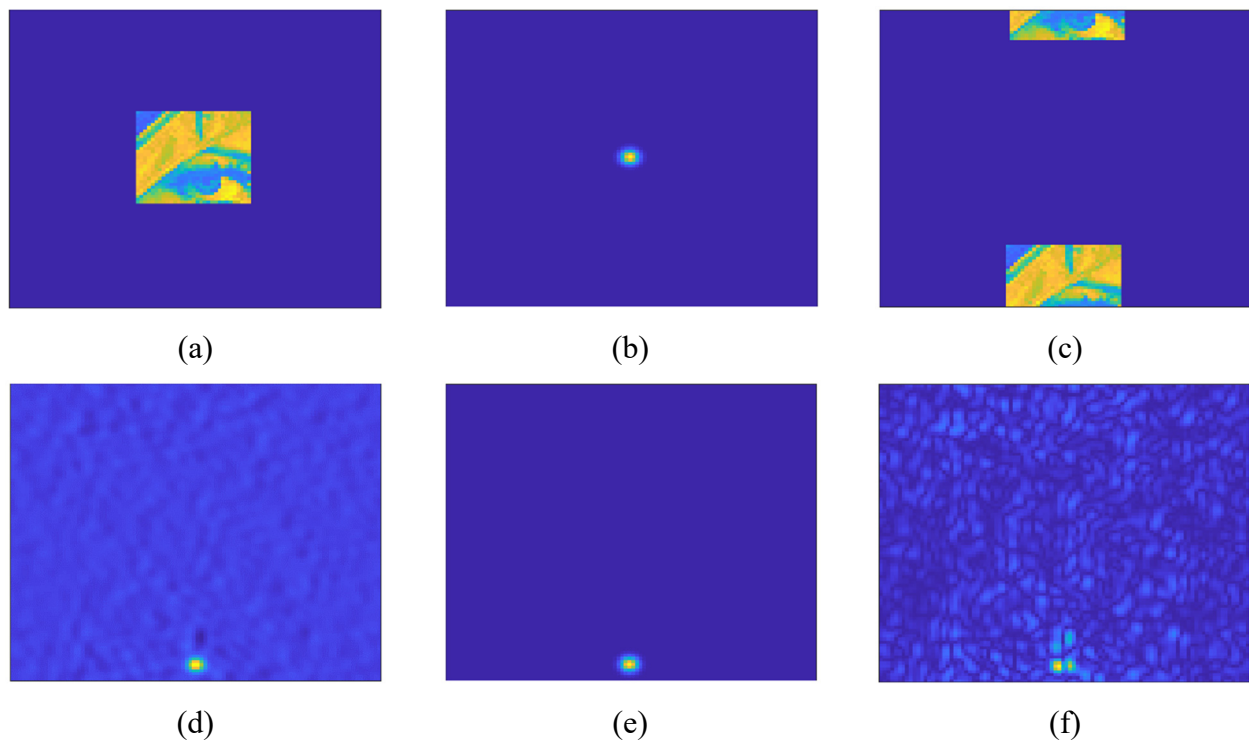


Figure 1. Connections and differences between the one-dimensional and two-dimensional filter responses. (a) Training sample (the size of the sample is 100×100). (b) Desired response. (c) Test sample obtained via the cyclic shift of the training sample. (d) Two-dimensional filter response. (e) One-dimensional filter response. (f) Difference between the two-dimensional and one-dimensional filter responses.

6. Experiments

Experiments were conducted on a computer equipped with an i5-8265 (1.80 GHz) CPU. The proposed viewpoints and the equivalence of the two filter-solving methods were verified through numerical experimentation and by designing response maps, respectively, to ensure the mathematical rigor and scientific validity of the correlation filter object tracking algorithm.

6.1. Numerical experimental verification of *Theorem 1*

To verify **Theorem 1**, let $x = \sin(50\pi t)$. The sampling rate was taken as 100 Hz, and the sampling time is 0–0.99 s, that is, $t = [0, 0.01, 0.02, \dots, 0.99]^T$. The sinusoidal signal was Fourier transformed, and its real and imaginary parts were taken, respectively. Figure 2(a),(c) show the amplitude of the real part of the \hat{x} and the amplitude of the imaginary part of the \hat{x} , respectively. The real and imaginary parts were taken for the diagonal elements of $F_N C(x) F_N^{-1}$. Figure 2(b) shows the amplitude of the real part of the diagonal elements of $F_N C(x) F_N^{-1}$, and Figure 2(d) presents the amplitude of the imaginary part of the diagonal

elements of $F_N C(x) F_N^{-1}$. The $\|F_N C(x) F_N^{-1} - \text{Diag}(\hat{x})\|_2$ in this experiment was 2.83×10^{-14} , where $\|x\|_2$ represents the l_2 norm of x . These results show that the real part map and imaginary part map of both are completely consistent, which proves **Theorem 1**.

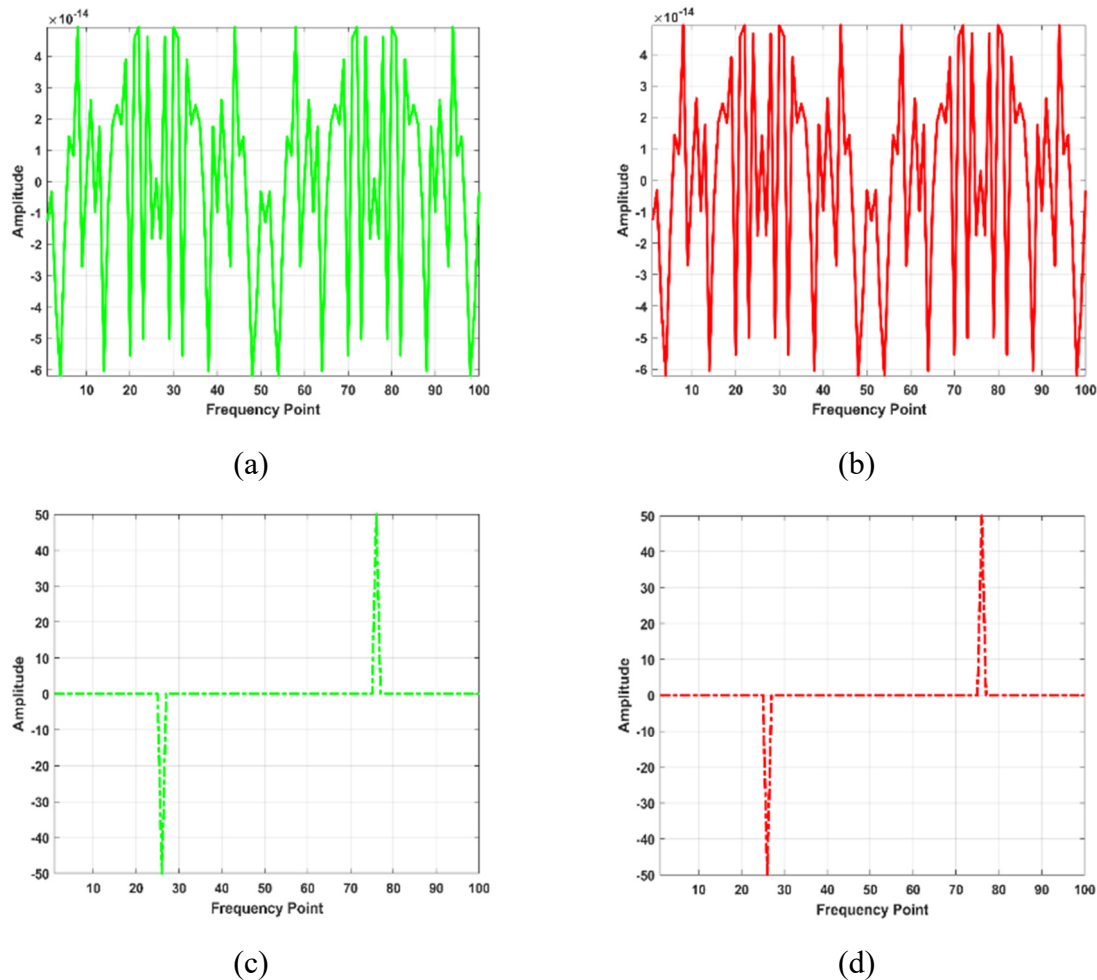


Figure 2. Numerical experimental results for **Theorem 1**. (a) Amplitude of the real part of \hat{x} . (b) Amplitude of the real part of the $F_N C(x) F_N^{-1}$ diagonal element. (c) Amplitude of the imaginary part of \hat{x} . (d) Amplitude of the imaginary part of the $F_N C(x) F_N^{-1}$ diagonal element.

6.2. Numerical experimental verification of **Theorem 2**

To verify **Theorem 2**, let $x = \sin(50\pi t)$. The sampling rate was taken as 100 Hz, and the sampling time is 0–0.99s, that is, $t = [0, 0.01, 0.02, \dots, 0.99]^T$. Figure 3(a) shows the pseudo-color map of the row-vector-based circulant matrix $C(x^T)$, and Figure 3(b) presents the pseudo-color map of the real part of $F_n \text{Diag}(\hat{x}) F_n^H$. The two maps are completely similar. The $\|F_n \text{Diag}(\hat{x}) F_n^H - C(x^T)\|_2$ in this experiment was 2.86×10^{-12} , thereby proving **Theorem 2**.

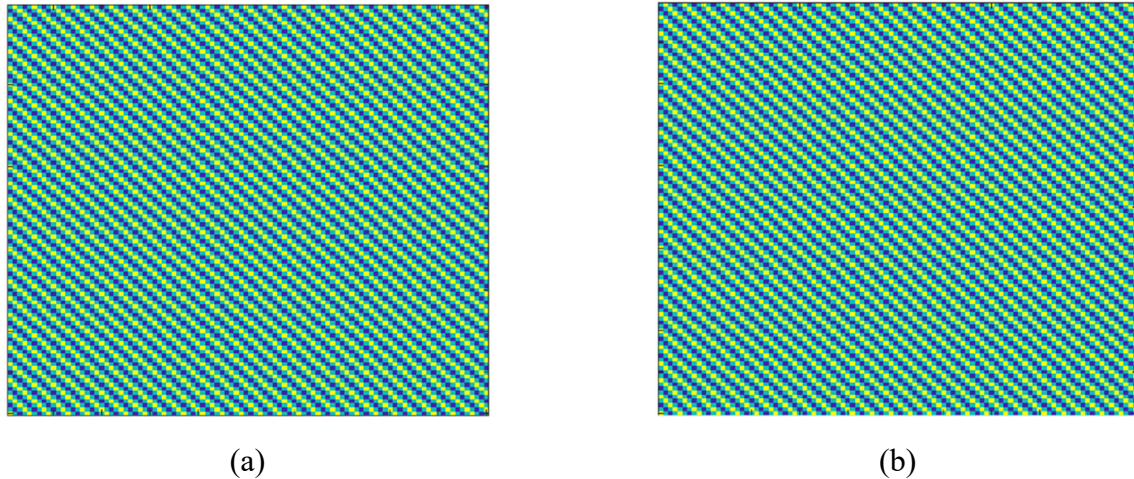


Figure 3. Numerical experimental results for **Theorem 2**. (a) Pseudo-color map of $C(\mathbf{x}^T)$. (b) Pseudo-color map of $F_n \text{Diag}(\hat{\mathbf{x}}) F_n^H$.

6.3. Numerical experimental verification of **Theorem 3**

To verify **Theorem 3**, let $\mathbf{x} = \sin(50\pi t)$ and $\mathbf{h} = \sin(20\pi t)$. The sampling rate was taken as 100 Hz, and the sampling time is 0–0.99 s, that is, $\mathbf{t} = [0, 0.01, 0.02, \dots, 0.99]^T$. Figures 4(a),(b) present the amplitudes of $C(\mathbf{x})\mathbf{h}$ over 1 s and $\text{real}(\text{ifft}_1(\hat{\mathbf{x}} \odot \hat{\mathbf{h}}))$ over 1 s, respectively. The results show that the two amplitude maps are completely consistent. The $\|C(\mathbf{x})\mathbf{h} - \text{real}(\text{ifft}_1(\hat{\mathbf{x}} \odot \hat{\mathbf{h}}))\|_2$ in this experiment was 2.44×10^{-12} . Since $\mathbf{x} * \mathbf{h} = \text{real}(\text{ifft}_1(\hat{\mathbf{x}} \odot \hat{\mathbf{h}}))$, **Theorem 3** is verified.

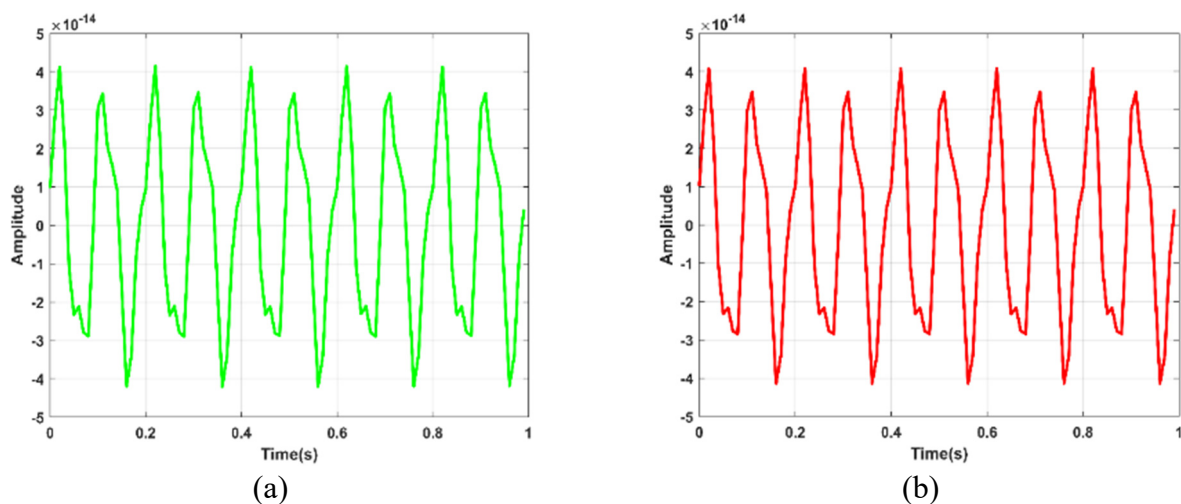


Figure 4. Numerical experimental results of **Theorem 3**. (a) Amplitude of $C(\mathbf{x})\mathbf{h}$ and (b) amplitude of $\text{real}(\text{ifft}_1(\hat{\mathbf{x}} \odot \hat{\mathbf{h}}))$.

6.4. Numerical experimental verification of **Theorem 4**

To verify **Theorem 4**, let $\mathbf{x} = \sin(50\pi t)$ and $\mathbf{h} = \sin(20\pi t)$. The sampling rate was taken as 100 Hz, and the sampling time is 0–0.99 s, that is, $\mathbf{t} = [0, 0.01, 0.02, \dots, 0.99]^T$. Figure 5(a) shows the amplitude of $\mathbf{C}(\mathbf{x}^T)\mathbf{h}$ over 1 s, and Figure 5(b) presents the amplitude of $\text{real}\left(\text{ifft}_1(\hat{\mathbf{x}}^* \odot \hat{\mathbf{h}})\right)$ over 1 s. The results show that the two amplitude maps are completely consistent. The $\left\|\mathbf{C}(\mathbf{x}^T)\mathbf{h} - \text{real}\left(\text{ifft}_1(\hat{\mathbf{x}}^* \odot \hat{\mathbf{h}})\right)\right\|_2$ in this experiment was 2.44×10^{-12} . As $\bar{\mathbf{x}} * \mathbf{h} = \text{real}\left(\text{ifft}_1(\hat{\mathbf{x}}^* \odot \hat{\mathbf{h}})\right)$, **Theorem 4** is verified.

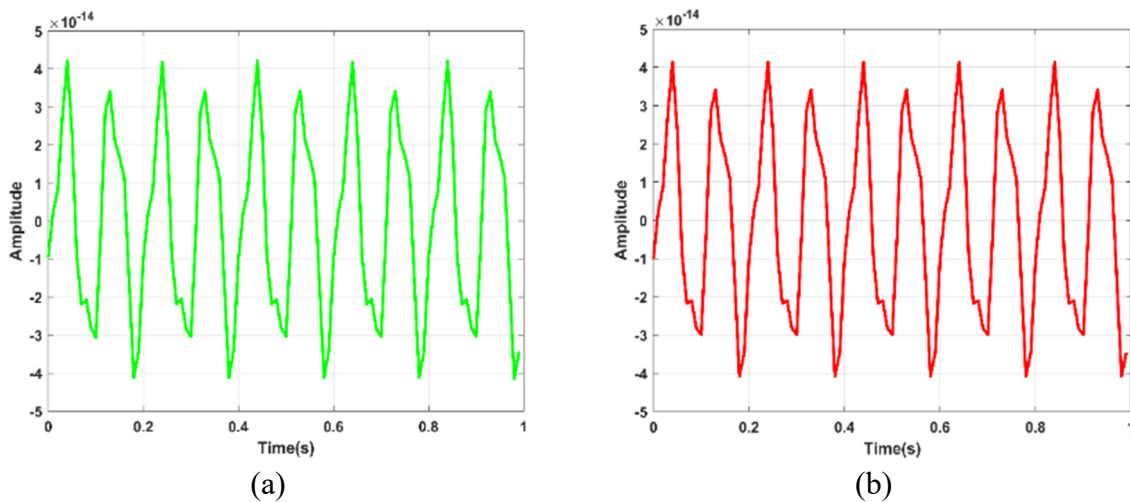


Figure 5 Numerical experimental results for **Theorem 4**. (a) Amplitude of $\mathbf{C}(\mathbf{x}^T)\mathbf{h}$ and (b) amplitude of $\text{real}\left(\text{ifft}_1(\hat{\mathbf{x}}^* \odot \hat{\mathbf{h}})\right)$.

6.5. Numerical experimental verification of Eq (44)

To verify Eq (44), let I_1 be the two-dimensional image signal, as shown in Figure 6(a), and let I_2 be the two-dimensional image signal, as shown in Figure 6(b). Figures 6(c),(d) illustrate the $\text{real}\left(\text{ifft}_2(\hat{I}_1^* \odot \hat{I}_2)\right)$ and $\text{mat}\left(\text{Cat}\left(\text{vec}(I_1[\Delta\tau_c, \Delta\tau_r])^T\right)\text{vec}(I_2)\right)$ responses. The results show that the responses were completely consistent as shown in Figures 6(c),(d). The $\left\|\text{real}\left(\text{ifft}_2(\hat{I}_1^* \odot \hat{I}_2)\right) - \text{mat}\left(\text{Cat}\left(\text{vec}(I_1[\Delta\tau_c, \Delta\tau_r])^T\right)\text{vec}(I_2)\right)\right\|_2$ in this experiment was 1.55×10^{-15} , proving that Eq (44) holds.

6.6. Equivalence verification of the two filter-solving methods

Figure 7(a) shows the base sample, and Figure 7(b) shows the predicted sample obtained via the cyclic shift of the base sample. In addition, Figures 7(c),(d) present the desired correlation response y and the spatial domain response obtained according to Eq (43), respectively. Figures 7(e),(f) show the spatial domain response obtained using Eqs (38) and (39), respectively. The results show that the spatial responses

in Figures 7(d),(f) are completely consistent (i.e., the two filter-solving methods are equivalent).

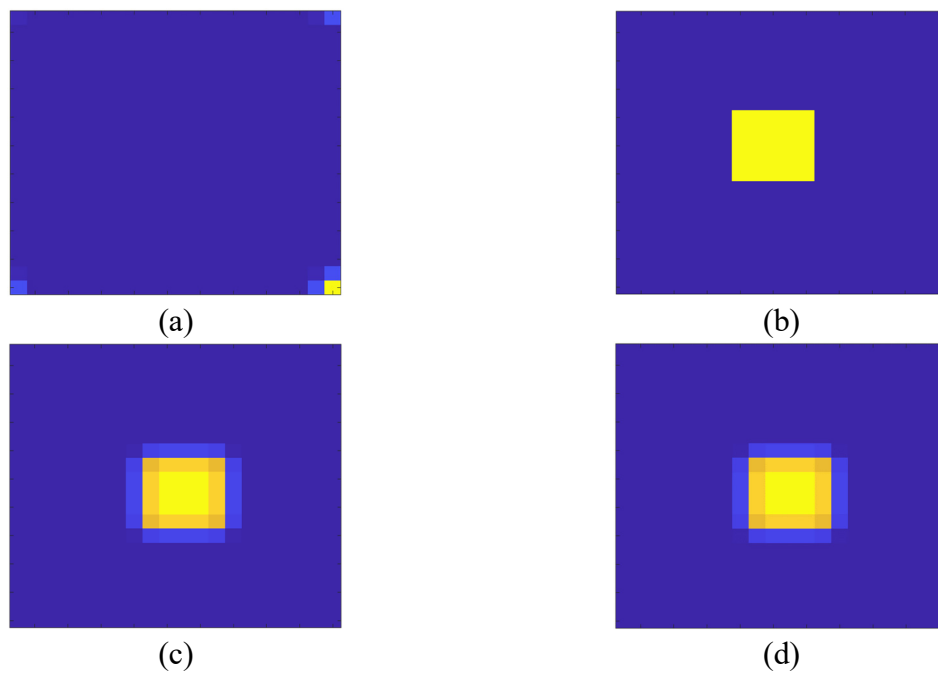


Figure 6. Numerical experimental results to validate Eq (44). (a) I_1 image, (b) I_2 image, (c) $\text{real}\left(\text{iff}_t\left(\hat{I}_1^* \odot \hat{I}_2\right)\right)$ response, and (d) $\text{mat}\left(\text{Cat}\left(\text{vec}\left(I_1\left[\Delta\tau_c, \Delta\tau_r\right]\right)^T\right) \text{vec}\left(I_2\right)\right)$ response.

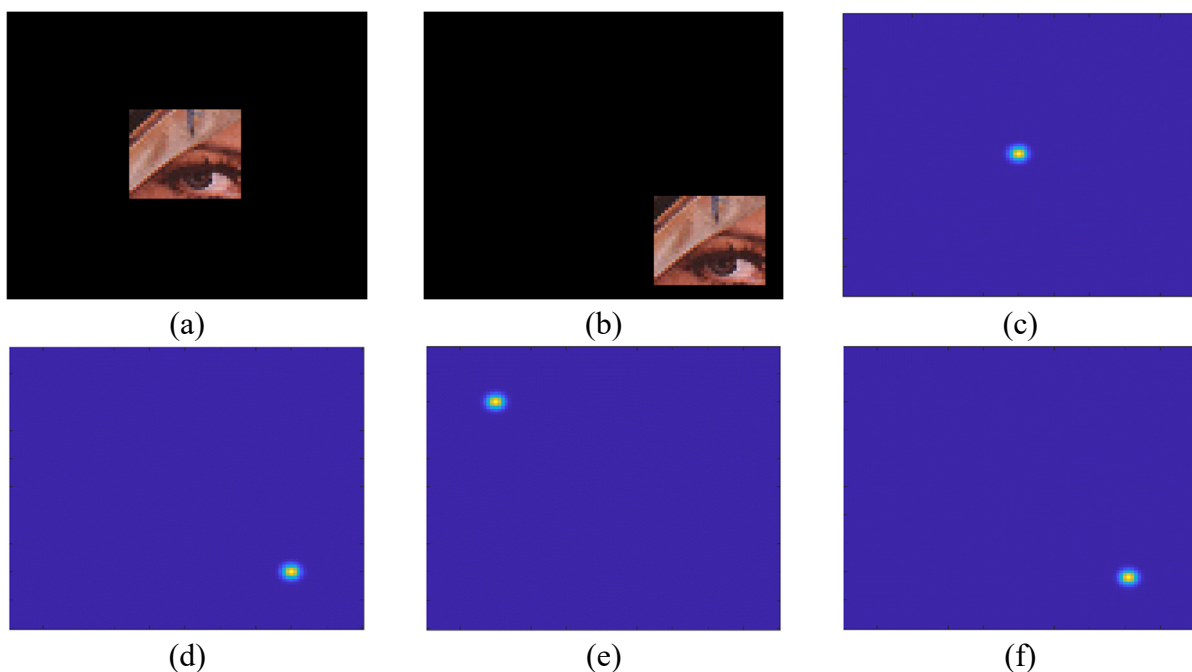


Figure 7. Equivalence experiment of the two filter-solving methods. (a) Base sample. (b) Predicted sample obtained via the cyclic shift of the base sample. (c) Desired correlation response. (d) Spatial domain response based on Eq (43). (e) Spatial domain response based on Eq (38). (f) Spatial domain response based on Eq (39).

Through the diagonalization property of the row-vector-based circulant matrix and the convolution theorem, the two filter-solving methods transform the spatial domain operation into an entry-wise multiplication operation in the frequency domain to circumvent the inverse operations of large matrices. Table 5 lists the running times required to calculate Eqs (35) and (37) for different image sizes, where λ was 0.1. The results indicate that solving the filter in the frequency domain can effectively reduce the operation time and improve the computational efficiency of the filter. The larger the signal size, the more apparent the advantage of solving the filter in the frequency domain.

Table 5. Time consumed to solve in the spatial and frequency domains.

Average time consumed (the number of the experiments is 10, and the image size is $\mathbb{R}^{\sqrt{N} \times \sqrt{N}}$)	$N=256$	$N=1024$	$N=4096$
Time consumed in the spatial domain/s (according to Eq. (35))	0.002429	0.088521	3.248634
Time consumed in the frequency domain/s (according to Eq (37))	0.000965	0.001281	0.001537

7. Conclusions

In this study, we systematically elucidated the theoretical modeling system of the correlation filter. Based on existing literature on correlation filters, four types of mathematical modeling and two types of filter-fast calculation methods were summarized and experimentally proven. The relationship among the four modeling types for correlation filter were discussed in detail. Our conclusions are as follows:

1) We elaborated on the definitions of the circulant matrix, convolution, and correlation operations in the correlation filter and their relationships. The viewpoints and mathematical findings provided in this study can provide useful theoretical support for research in the field of correlated filter object tracking.

2) The diagonalization property of the circulant matrix and the convolution theorem were employed to solve the filter by transforming the spatial-domain operation into an entry-wise multiplication operation in the frequency domain. This approach avoids the inverse operation of large spatial-domain matrices and reduces the computational complexity compared with directly solving in the spatial domain. The experiments showed that the results obtained using the two filter-solving methods were consistent. The proposed fast filter calculation method is critical the efficient implementation of the correlation filter tracking algorithm.

3) We experimentally proved the existence of slight differences between the one-dimensional and two-dimensional filter methods. The main reasons for these differences were discussed in detail. Subsequently, the equivalence of the two filter methods in object positioning was reflected via experimentation to provide a reliable foundation for the engineering realization of the two theoretical methods.

Traditional correlation filter tracking frameworks utilize handcrafted features to distinguish the object and background. The discrimination ability of these features is limited; thus, the application of the correlation filter tracking method in complex scenes has some limitations. As deep learning technology gradually matures, it will provide a correlation filter theoretical framework with more discriminative visual features. Subsequent work should attempt to improve the correlation and deep-learning-based tracking algorithm to improve the overall performance of the tracker. For example, the

computational theory proposed in this paper can be introduced into cyclic shifting attention computation [25] to obtain more efficient computation.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work is supported by the Natural Science Foundation Project of Zhangzhou City (ZZ2023J37), the Principal Foundation of Minnan Normal University (KJ19019), the High-level Science Research Project of Minnan Normal University (GJ19019), Research Project on Education and Teaching of Undergraduate Colleges and Universities in Fujian Province (FBJY20230083), and the Education Research Program of Minnan Normal University (202211).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. S. Javed, M. Danelljan, F. S. Khan, M. H. Khan, M. Felsberg, J. Matas, Visual object tracking with discriminative filters and siamese networks: A survey and outlook, *IEEE Trans. Pattern Anal. Mach. Intell.*, **45** (2023), 6552–6574. <https://doi.org/10.1109/TPAMI.2022.3212594>
2. F. Chen, X. Wang, Y. Zhao, S. Lv, X. Niu, Visual object tracking: A survey, *Comput. Vision Image Understanding*, **222** (2022), 103508. <https://doi.org/10.1016/j.cviu.2022.103508>
3. D. Zhang, Z. Zheng, M. Li, R. Liu, CSART: Channel and spatial attention-guided residual learning for real-time object tracking, *Neurocomputing*, **436** (2021), 260–272. <https://doi.org/10.1016/j.neucom.2020.11.046>
4. F. Gu, J. Lu, C. Cai, Q. Zhu, Z. Ju, RTSformer: A robust toroidal transformer with spatiotemporal features for visual tracking, *IEEE Trans. Hum.-Mach. Syst.*, **54** (2024), 214–225. <https://doi.org/10.1109/THMS.2024.3370582>
5. Y. Qian, L. Yu, W. Liu, A. G. Hauptmann, Electricity: An efficient multi-camera vehicle tracking system for intelligent city, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, (2020), 2511–2519. <https://doi.org/10.1109/CVPRW50498.2020.00302>
6. X. Chen, X. Xu, Y. Yang, Y. Huang, J. Chen, Y. Yan, Visual ship tracking via a hybrid kernelized correlation filter and anomaly cleansing framework, *Appl. Ocean Res.*, **106** (2021), 102455. <https://doi.org/10.1016/j.apor.2020.102455>
7. H. Zhang, Y. Li, H. Liu, D. Yuan, Y. Yang, Feature block-aware correlation filters for real-time UAV tracking, *IEEE Signal Process. Lett.*, **31** (2024), 840–844. <https://doi.org/10.1109/LSP.2024.3373528>

8. X. Wang, D. Zeng, Y. Li, M. Zou, Q. Zhao, S. Li, Enhancing UAV tracking: a focus on discriminative representations using contrastive instances, *J. R.-Time Image Process.*, **21** (2024), 78. <https://doi.org/10.1007/s11554-024-01456-2>
9. C. Zhu, J. Yang, Z. Shao, C. Liu, Vision based hand gesture recognition using 3D shape context, *IEEE/CAA J. Autom. Sin.*, **8** (2021), 1600–1613. <https://doi.org/10.1109/JAS.2019.1911534>
10. M. N. H. Mohd, M. S. M. Asaari, O. L. Ping, B. A. Rosdi, Vision-based hand detection and tracking using fusion of kernelized correlation filter and single-shot detection, *Appl. Sci.*, **13** (2023), 7433. <https://doi.org/10.3390/app13137433>
11. J. F. Henriques, R. Caseiro, P. Martins, J. Batista, High-speed tracking with kernelized correlation filters, *IEEE Trans. Pattern Anal. Mach. Intell.*, **37** (2015), 583–596. <https://doi.org/10.1109/TPAMI.2014.2345390>
12. Y. Li, J. Zhu, A scale adaptive kernel correlation filter tracker with feature integration, in *Computer Vision-ECCV 2014 Workshops*, **8926** (2014), 254–265. https://doi.org/10.1007/978-3-319-16181-5_18
13. M. Danelljan, G. Hager, F. S. Khan, M. Felsberg, Learning spatially regularized correlation filters for visual tracking, in *2015 IEEE International Conference on Computer Vision (ICCV)*, (2015), 4310–4318. <https://doi.org/10.1109/ICCV.2015.490>
14. C. Ma, X. Yang, C. Zhang, M. Yang, Long-term correlation tracking, in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2015), 5388–5396. <https://doi.org/10.1109/CVPR.2015.7299177>
15. M. Danelljan, G. Häger, F. S. Khan, M. Felsberg, Discriminative scale space tracking, *IEEE Trans. Pattern Anal. Mach. Intell.*, **39** (2017), 1561–1575. <https://doi.org/10.1109/TPAMI.2016.2609928>
16. M. Danelljan, G. Bhat, F. Shahbaz Khan, M. Felsberg, ECO: Efficient convolution operators for tracking, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2017), 6931–6939. <https://doi.org/10.1109/CVPR.2017.733>
17. A. Lukezic, T. Vojir, L. C. Zajc, J. Matas, M. Kristan, Discriminative correlation filter with channel and spatial reliability, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2017), 4847–4856. <https://doi.org/10.1109/CVPR.2017.515>
18. Z. Huang, C. Fu, Y. Li, F. Lin, P. Lu, Learning aberrance repressed correlation filters for real-time UAV tracking, in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, (2019), 2891–2900. <https://doi.org/10.1109/ICCV.2019.00298>
19. B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, J. Yan, SiamRPN++: Evolution of siamese visual tracking with very deep networks, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2019), 4277–4286. <https://doi.org/10.1109/CVPR.2019.00441>
20. T. Xu, Z. Feng, X. Wu, J. Kittler, Joint group feature selection and discriminative filter learning for robust visual object tracking, in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, (2019), 7949–7959. <https://doi.org/10.1109/ICCV.2019.00804>
21. D. S. Bolme, J. R. Beveridge, B. A. Draper, Y. M. Lui, Visual object tracking using adaptive correlation filters, in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (2010), 2544–2550. <https://doi.org/10.1109/CVPR.2010.5539960>
22. L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, P. H. Torr, Fully-convolutional siamese networks for object tracking, in *Computer Vision-ECCV 2016 Workshops*, **9914** (2016), 850–865. https://doi.org/10.1007/978-3-319-48881-3_56

23. H. K. Galoogahi, A. Fagg, S. Lucey, Learning background-aware correlation filters for visual tracking, in *2017 IEEE International Conference on Computer Vision (ICCV)*, (2017), 1144–1152. <https://doi.org/10.1109/ICCV.2017.129>
24. Y. Li, C. Fu, F. Ding, Z. Huang, G. Lu, Autotrack: Towards high-performance visual tracking for UAV with automatic spatio-temporal regularization, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2020), 11920–11929. <https://doi.org/10.1109/CVPR42600.2020.01194>
25. Z. Song, J. Yu, Y. P. Chen, W. Yang, Transformer tracking with cyclic shifting window attention, in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2022), 8781–8790. <https://doi.org/10.1109/CVPR52688.2022.00859>
26. Y. Chen, H. Wu, Z. Deng, J. Zhang, H. Wang, L. Wang, et al., Deep-feature-based asymmetrical background-aware correlation filter for object tracking, *Digital Signal Process.*, **148** (2024), 104446. <https://doi.org/10.1016/j.dsp.2024.104446>
27. K. Chen, L. Wang, H. Wu, C. Wu, Y. Liao, Y. Chen, et al., Background-aware correlation filter for object tracking with deep CNN features, *Eng. Lett.*, **32** (2024), 1353–1363.
28. R. M. Gray, Toeplitz and circulant matrices: A review, *Found. Trends Commun. Inf. Theory*, **2** (2006), 155–239. <http://doi.org/10.1561/0100000006>
29. J. F. Henriques, R. Caseiro, P. Martins, J. Batista, Exploiting the circulant structure of tracking-by-detection with kernels, in *Computer Vision-ECCV 2012*, (2012), 702–715. https://doi.org/10.1007/978-3-642-33765-9_50
30. M. E. Kilmer, C. D. Martin, Factorization strategies for third-order tensors, *Linear Algebra Appl.*, **435** (2011), 641–658. <https://doi.org/10.1016/j.laa.2010.09.020>
31. N. Hao, M. E. Kilmer, K. Braman, R. C. Hoover, Facial recognition using tensor-tensor decompositions, *SIAM J. Imaging Sci.*, **6** (2013), 437–463. <https://doi.org/10.1137/110842570>
32. M. E. Kilmer, K. Braman, N. Hao, R. C. Hoover, Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging, *SIAM J. Matrix Anal. Appl.*, **34** (2013), 148–172. <https://doi.org/10.1137/110837711>
33. B. Hunt, A matrix theory proof of the discrete convolution theorem, *IEEE Trans. Audio Electroacoust.*, **19** (1971), 285–288. <https://doi.org/10.1109/TAU.1971.1162202>
34. J. Martinez, R. Heusdens, R. C. Hendriks, A generalized Fourier domain: Signal processing framework and applications, *Signal Process.*, **93** (2013), 1259–1267. <https://doi.org/10.1016/j.sigpro.2012.10.015>
35. A. Iwasaki, Deriving the variance of the discrete Fourier transform test using Parseval’s theorem, *IEEE Trans. Inf. Theory*, **66** (2020), 1164–1170. <https://doi.org/10.1109/TIT.2019.2947045>
36. Q. Hu, H. Wu, J. Wu, J. Shen, H. Hu, Y. Chen, et al., Spatio-temporal self-learning object tracking model based on anti-occlusion mechanism, *Eng. Lett.*, **31** (2023), 1–10.
37. Y. Huang, Y. Chen, C. Lin, Q. Hu, J. Song, Visual attention learning and antiocclusion-based correlation filter for visual object tracking, *J. Electron. Imaging*, **32** (2023), 13023. <https://doi.org/10.1117/1.JEI.32.1.013023>
38. J. Cui, J. Wu, L. Zhao, Learning channel-selective and aberrance repressed correlation filter with memory model for unmanned aerial vehicle object tracking, *Front. Neurosci.*, **16** (2023). <https://doi.org/10.3389/fnins.2022.1080521>

39. C. Fan, H. Yu, Y. Huang, C. Shan, L. Wang, C. Li, SiamON: Siamese occlusion-aware network for visual tracking, *IEEE Trans. Circuits Syst. Video Technol.*, **33** (2023), 186–199. <https://doi.org/10.1109/TCSVT.2021.3102886>
40. W. Hu, Q. Wang, L. Zhang, L. Bertinetto, P. H. S. Torr, SiamMask: A framework for fast online object tracking and segmentation, *IEEE Trans. Pattern Anal. Mach. Intell.*, **45** (2023), 3072–3089.
41. D. Sharma, Z. A. Jaffery, Multiple object tracking through background learning, *Comput. Syst. Sci. Eng.*, **44** (2023), 191–204. <https://doi.org/10.32604/csse.2023.023728>
42. J. Zhang, Y. He, S. Wang, Learning adaptive sparse spatially-regularized correlation filters for visual tracking, *IEEE Signal Process. Lett.*, **30** (2023), 11–15. <https://doi.org/10.1109/LSP.2023.3238277>



AIMS Press

©2024 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)