



Research paper

Design, Analysis, and Implementation of a New Online Object Tracking Method Based on Sketch Kernel Correlation Filter (SHKCF)

M. Yousefzadeh, A. Golmakani*, G. Sarbishaei

Department of Electrical Engineering, Sadjad University of Technology, Mashhad, Iran.

Article Info

Article History:

Received 08 July 2023
Reviewed 17 August 2023
Revised 11 September 2023
Accepted 11 October 2023

Keywords:

Artificial intelligence
Video analysis
Object tracking
SHKCF
KCF and online tracker

*Corresponding Author's Email
Address:
golmakani@sadjad.ac.ir

Abstract

Background and Objectives: To design an efficient tracker in a crowded environment based on artificial intelligence and image processing, there are several challenges such as the occlusion, fast motion, in-plane rotation, variations in target illumination and Other challenges of online tracking are the time complexity of the algorithm, increasing memory space, and tracker dependence on the target model. In this paper, for the first time, sketch matrix theory in ridge regression for video sequences has been proposed.

Methods: A new tracking object method based on the element-wise matrix with an online training method is proposed including the kernel correlation Filter (KCF), circular, and sketch matrix. The proposed algorithm is not only the free model but also increases the robustness of the tracker related to the scale variation, occlusion, fast motion, and reduces KCF drift.

Results: The simulation results demonstrate that the proposed sketch kernel correlation filter (SHKCF) can increase the computational speed of the algorithm and reduces both the time complexity and the memory space. Finally, the proposed tracker is implemented and experimentally evaluated based on video sequences of OTB50, OTB100 and VOT2016 benchmarks.

Conclusion: The experimental results show that the SHKCF method obtains not only OPE partial evaluation of Out of view, Occlusion and Motion Blur in object accuracy but also achieved the partial evaluation of Illumination Variation, Out of Plane Rotation, Scale Variation, Out of View, Occlusion, In of Plane Rotation, Background Clutter, Fast Motion and Deformation in object overlap which are the first rank compared to the state-the-art works. The result of accuracy, robustness and time complexity are obtained 0.929, 0.93 and 35.4, respectively.

This work is distributed under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>)



Introduction

One of the most important aspects in the machine vision and pattern recognition systems is object tracking which has different applications such as video surveillance with CCTV, medical and military video analysis, human-computer communication, and robotic smart vehicle [1]-[6], [12]. The main task of tracking is that the object location is automatically estimated and scaled in the video sequences [1]. In the last decade, mathematical theories and modelling techniques in object tracking have

been developed. These techniques are based on the dynamic and static learning theory, particle filters, discriminate correlation filters, pattern matching, deep neural networks, etc. [5].

Despite this impressive progress, the existence of an object tracking algorithm that can be adapted to different conditions is still a challenging problem in machine vision. Besides, the majority of most previous approaches have been designed and implemented in simple landscapes. However, the online tracking of real objects in

unpredictable landscapes is still a big challenge because of illumination variations, occlusion, scale variations, and background clutter [6]-[8]. Therefore, developing this new design of online tracker is still needed to open up practical tracking applications. It is worth mentioning that the challenges based on tracking benchmarks can help to assess the accuracy, precision, and robustness of the tracking algorithm instead of the challenges within the real-world environments and online video sequences [9], [11]. For examples, OTB50 and OTB100 are two well-known famous standard benchmarks [5], [12], [13].

Generally, modern trackers are divided into two main categories: the generative and discriminating trackers [9]. The generative tracker method is based on the random variable probability estimating explained as follows. The object is labelled on the first frame. Then, the estimation error between the initial and new samples is calculated. Finally, based on the results of the previous step, the best candidate is generated. This approach extracts two models: an appearance model for the object and another model for the background. The main goal of this tracker is to predict the object location using the maximum similarity of the test sample to the appearance model. The tracker calculates the density of test samples around the object location to increase prediction accuracy. Then the object is selected by a particle filter. The main problem of the generative tracker is its dependency on the labeled datasets. Also, small changes in the background cause object estimation from the test samples to be inaccurate and very erroneous. As a to these challenges, researchers are aiming to use new trackers, such as the average transmission tracker [15], ensemble object tracker [15], fragment-based tracker [16], and sparse representation tracker [17]. However, the discriminating tracker can detect the object within the background regarding an online classified problem [2]. Nowadays, discriminating trackers deploy machine learning and tracking algorithms based on the kernel correlation filter. The main reasons for using the correlation filter-based trackers are their high speed, stability, and accuracy since these trackers use both the object and background information [7], [9], [11]. Despite the advancements of discriminating trackers, the efficiency of the designed tracking algorithms must be evaluated and compared with different objects on the existing benchmarks and challenges (e.g., OTB100 and OTB50).

The robustness and efficiency of the tracker are two main issues in the field of tracking that are highly competing in the literature. As already mentioned, improving the mentioned issues should be evaluated according to standard challenges (e.g., OTB50, OTB100, VOT2019, UAV123, LaSOT and TrackingNet). In this article, we have used two of the most widely used

benchmarks, ot50 and otb100, and all our results are based on these two standard datasets. Also, the speed of the designed algorithm for calculating the object location is another issue that has to be considered within the algorithm evaluation. Recently, there have been attempts to overcome the above problems, such as multiple learning tracker [18], ensemble tracker [19], SVM tracker [8], correlation filter-based tracker, to name a few. It should be noted that the high capability and performance of the trackers based on discriminating correlation filters (DCF), compared to the most up-to-date available tracking algorithms, have been proven in [9], [18], [19]. In fact, the main advantages of DCF trackers are the multidimensional appearance model, circulant matrix, and frequency domain calculations explained in the next section.

In the frequency domain, DCF learning comes with a huge learning cost, such as circular shift samples of the ground-truth object, because a circular shift introduces the unwanted boundary effects. This problem is partially mitigated by additional predefined spatial constraints on the filter coefficients. For example, Danelljan *et al.* [52] introduced spatial regularized differential correlation filters (SRDCF) to reduce boundary effects. It is expected from an object tracker to be spatially penalized by its distance from the object center. In order to generate true positive and negative samples of the model training, Galoogahi *et al.* [50] proposed Learning background-aware correlation filters (BACF) for multiplied the correlation filter directly to binary matrix. In order to these mentioned method Learning Spatial-Temporal Regularized Correlation Filters for Visual Tracking (STRCF) and Visual Tracking via Adaptive Spatially-Regularized Correlation Filters (ASRCF) are also employed in literature for reducing mentioned boundary effect with spatial temporal Regularized and adaptive Spatially-Regularized, respectively.

It should be mention that the appearance model of most DCF-based trackers is developed by a linear interpolation approach. However, these models cannot adapt to overall appearance variances, leading to filter degradation. In order to solve the problem of filter degradation, several approaches have been proposed in the literature, such as the training set management [31], [32] and the temporal constraints, where temporal regularization has been proven to be an effective method.

In order to solve most to challenging problem of unwanted boundary effect, we proposed sketch kernel correlation filters (SHKCF) for real time object tracking. We implemented and compared our approach with state-of-the-art trackers in the OTB 100 benchmarks. The results show that SHKCF performs better than the state-of-the-art trackers in terms of accuracy and computational speed.

The rest of the paper is organized as follows: Section II presents the related works on video sequence trackers. Section III introduces the KCF tracking algorithm. Then, the proposed method based on the kernel ridge regression (KRR), sketch kernel ridge regression (SKRR), and SHKCF video sequence tracking algorithm is analyzed and presented. Section V provides the test results and compares them with the SOTA works regarding the OTB100 and OTB50 datasets.

Related Works

As already mentioned, appearance modeling is one of the most important approaches for Conventional object tracking, which can be classified roughly into discriminative and generative methods [37]. Generative approaches identify an object by learning reference model with the most similar video sequence region, including sparse representation [55], template matching [39], subspace learning [54], the generative methods can provide more accurate performance in a small region and are robust against the object occlusion. But they are sensitive to the same distractions in the object' surrounding region.

During the past decade, trackers based on the well-known regression are investigated and provided well performance [12]. Particularly, the series of the correlation filters-based trackers, KCF [1], SAMF [38], DSST [56], are demonstrated to be the best tracker in accuracy on the challenging of OTB100 [26].

Kernel correlation filter (KCF) is one of the methods used in DCF based trackers that has been highly investigated and developed by researchers in recent years [8]. The KCF method uses a set of patch mappings (positive/negative) and a classification between the object and its surroundings to create a kernel model. Fast Fourier transform (FFT) and the inverse fast Fourier transform (IFFT) are also employed to increase computational speed. It is worth noting that the calculation speed of the correlation filter in the frequency domain is higher than in the time domain. Therefore, all calculations are performed in the frequency domain, and then the IFFT is employed to return them to the time domain. Finally, the KCF algorithm returns the object location as the output [9]. The results in [2], [5], [20] show that the KCF method enhances the object tracking performance compared to the SOTA works in terms of speed and robustness regarding the standard benchmark platforms [8]. However, some KCF method errors against various challenges remain unsolved, such as smart training, speed, drifting, and occultation. To the best of our knowledge, no other KCF object tracker has employed the sketch method. Although the sketch is introduced in Reference [7], its combination with the video sequence tracker has not yet been reported. So, in this paper, we combine the sketch method with the classic KCF tracker

and demonstrate its pros and cons. In this paper, we propose a new method for object tracking based on the kernel correlation filter (KCF) and compare the results with the SOTA algorithms in object tracking scenarios. Our proposed method is based on SHKCF principle, which is proposed for the first time in video sequence tracking. We implement and evaluate our method according to OTB50 and OTB100 benchmarks to validate the results.

The main reason for the development of these algorithms is that the correlation filter calculates the learning coefficients and minimizes energy consumption [21]. This filter is also employed to calculate the variance of the training sample responses [22]. It should be noted that the basic filter-based trackers were designed based on calculations in the time domain. However, Bolme et al. proposed a filter learning method in the frequency domain [23]. In these filters, convolutional operators in the time domain transform into summations and multiplications in the frequency domain leading to a decrease in computational time. Therefore, implementing filters in the frequency domain leads to a very high frame rate, and the temporal complexity of the calculations reduces significantly. Although FFT-based algorithms are very effective and have many applications in signal processing, FFT has many limitations in tracking applications. Recently, the most efficient FFT-based tracker has been proposed by Henriques using the kernel method [1]. Previous algorithms did not consider a clear relationship between nonlinear kernels and Fourier domain parameters. So, they had high computational complexity and had limitations in image processing. This fact motivates Henriques to propose a simple connection between the transfer video patches and the training algorithms. Since 2015, researchers have improved Henriques' tracker through combination with other methods such as circular structure kernel (CSK) tracker, color name (CN) tracker, spatially scaled discriminating tracker, and kernel correlation filter (KCF). The CSK tracker is designed based on light intensity features, but it is not robust to some challenges such as occlusion and deformation [9]. The KCF deployed the density of samples around the object location, minimum kernel squares, and the rotational shifting structure of the video patch for learning [15]. This tracker shows a more accurate and robust performance than the CSK tracker, because the intelligent samples in the correlation filter are trained by both the histogram of gradient directional (HOG) features and the circular shift matrix on the video sequence. In addition to the HOG features, KCF can be combined with the color name feature to promote multi-feature detections. The KCF has significant computational properties leading to online frame per second (FPS) rates. The KCF tracker, similar to the other trackers, can be trained using neural networks and deep learning

algorithms. Recently, [24] proposed a deep learning-based method for estimating the object's location. This method, known as the CCOT method, combines location information with neural network features and correlation filters. It is worth noting that the CCOT tracker wins the VOT2016 challenge. For a fair comparison, we have compared the results of the proposed method with the results of CCOT using OTB100 and OTB50 challenges datasets. We also compared the results with the SOTA works having the same parameters and processing standard platform. Moreover, the KCF tracker can be combined with other mathematical matrices such as a multidimensional matrix with color scale features [25], [26] in a way that the calculations are performed using kernel functions and circulant matrix structure. To develop KCF trackers in the free model, multiple KCF trackers are proposed in [27]. Also, the online classifier Fern tracker is proposed in [28] to solve some other challenges (e.g., occlusion, out of view). To increase accuracy, some methods use the object and its vicinity pixels.

In other words, the KCF tracker intercepts the object features using circular shifts and takes samples in the vicinities of the desired object.

The Traditional KCF Tracker Algorithm

The pattern in the first frame of the video sequence (X) is assumed as the input circulant matrix. This one-dimensional circulant matrix is obtained by the original data set received from continuous frames: $P_x = [x_n, x_1, \dots, x_{n-1}]^T$. Although, this circulant matrix can be extended to its two-dimensional form which considers all circular shifts: $\{P_x^i | i = 0, \dots, n - 1\}$. There are two possible shifts in each direction for this matrix. The matrix generated by the possible circular shifts is called the circulant matrix or data matrix. Therefore, the goal of KCF learning is to train the H filter as follows. Considering the minimum regression error, the KCF classifier is trained according to (1).

$$\text{Arg min}_H \sum_i^n (f(H; P^i x) - Y_i)^2 + \lambda \|H\|_2^2 \quad (1)$$

$$f(H; P^i x) = H^T \Phi(X) \quad (2)$$

where f , H and $\Phi(X)$ are the mapping function, KCF filter and the mapping of the X pattern in the Fourier domain, respectively. In this equation, two patterns are used, the learning pattern (X) and the regression pattern (Y). This method is known as the minimum output sum of squared error (MOSSE) calculation.

The MOSSE calculation is obtained from the maximum values of the Gaussian function with the purpose of the minimal change in the circular shift. It should be noted that the digitalization of the Gaussian function, often termed P^i is achieved in the KCF method based on the

degree of circular shifts. The digital matrix P^i is a matrix of zeros and ones indicating the incorrect and correct data, respectively. The detecting probability of an object is proportional to the number of correct data. Thus, for a detected object, the sum of correct data, is higher than a predefined threshold. The threshold is set by the designer and avoids falling into the local minimum trap. Equation (1) can be rewritten as (3):

$$\mathcal{L}(H) = \text{Arg min}_H \|H^T \Phi_H - Y\|_2^2 + \lambda \|H\|_2^2 \quad (3)$$

where Φ_H , $\lambda \geq 0$ and Y are the mapping of all circular shifts of the X pattern in the Fourier domain, the regularization parameter and the results of the regression object pattern, respectively. Equation (3) shows the cost function depends on the partial mapping function $\Phi(\cdot)$ explained as follow.

If the partial mapping Φ is linear, then $\Phi(X) = X$ where $X = [X_1 X_2 \dots]^T$. The KCF filter function (H) can be written by $H = (X^T X + \lambda I)^{-1} X^T Y$, where I is the identity matrix. The components of the H filter can be obtained from the X pattern using a circulant matrix. In other words, the H filter performs the diagonal matrix calculations using the discrete Fourier transform matrix (DFT). Therefore, the H filter in the Fourier domain with Hadamard Product (element-wise multiplication) is expressed as:

$$\hat{H}^* = \frac{\hat{X}^* \odot \hat{Y}}{\hat{X}^* \odot \hat{X} + \lambda} \quad (4)$$

where \hat{X} and \hat{X}^* are the fast Fourier transform of X and the complex conjugate of X , respectively. It is worth mentioning when several patterns are employed in the training steps, the H filter is combined with all the patterns as [11]:

$$\hat{H}^* = \frac{\sum_{j=1}^m \hat{X}_j^* \odot \hat{Y}_j}{\sum_{j=1}^m \hat{X}_j^* \odot \hat{X}_j + \lambda} \quad (5)$$

Due to the linearity of partial mapping Φ in (2), it is not possible to calculate the multiple features of the objects. If the partial mapping Φ is nonlinear, then the X pattern has various properties such as HOG. Due to the nonlinearity of the mapping, (2) will not have a suitable response. Thus, to solve the nonlinear mapping, the problem is converted to the ridge regression method and the new H is calculated using kernel filter analysis as $H = \Phi^T \alpha$, where $\alpha = (K + \lambda I)^{-1} Y$ and $K = \Phi_X \Phi_X^T$.

The Gaussian kernel function creates multiple features of the X pattern. In [16], it is proved that if the kernel matrix has a constant permutation, the kernel matrix K is circular. Regarding the circularity of the K matrix and DFT of the diagonal property in (2), the coefficient α (frequency domain) can be extracted as [7].

$$\hat{\alpha}^* = \frac{\hat{Y}}{\hat{K}^{XX} + \lambda} \quad (6)$$

For the well-known Gaussian kernel $\hat{K}(x, x') = \exp\left(-\frac{1}{\sigma^2}(\|x - x'\|^2)\right)$, Next

$$\hat{K}^{xx'} = \exp\left\{-\frac{1}{\sigma^2}(\|X\|^2 + \|X'\|^2 - 2F^{-1}(\sum_a \hat{X}_a^* \odot \hat{X}'_a))\right\} \quad (7)$$

where $\hat{K}^{xx'}$, \hat{X} and \hat{Y} are the Gaussian kernel, the object of FFT filter and response, respectively.

A. Kernel Ridge Regression with Circulant Matrix

The idea of the proposed method is to minimize the difference between the output and the object's real location. To create a circulant matrix, we need to store the first column of the input matrix. Since a circulant matrix depends on its first column, a matrix formed by the circular shift method requires less memory [29]. The circulant matrix is shown as (8):

$$C = \begin{bmatrix} C_1 & C_m & C_{m-1} & \dots & C_2 \\ C_2 & C_1 & C_m & \ddots & C_3 \\ C_3 & C_2 & C_1 & \ddots & C_4 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_m & C_{m-1} & C_{m-2} & \dots & C_1 \end{bmatrix} \quad (8)$$

The matrix (8) is expressed in closed form (9):

$$C_{[m]} = \text{cir}[c_j; j \in \{1, 2, \dots, m\}] \quad (9)$$

Besides, the matrix C with two inputs i and j is displayed in the simple closed-form solution of (10):

$$C_{ij} = c_{(i-j) \bmod m} \quad (10)$$

It should be noted that the advantage of a circulant matrix is not only the decreased memory but also calculations in the Fourier domain are faster compared to time-domain convolution. Discrete Fourier transform (DFT) of a circulant matrix is calculated by (11):

$$C = \frac{1}{m} G^* \text{diag}(Gc)G \quad (11)$$

where $C = [C_1, C_2, \dots, C_m]^T$, $G = \left[e^{i\left(\frac{2\pi}{mk\ell}\right)} \right]_{k,t=1}^m$ and G^* are circulant matrix transpose, discrete Fourier matrix and its conjugate, respectively. $\text{diag}(Gc)$ is a diagonal matrix whose diagonal elements are the elements of vector G . Moreover, the computational complexity is decreased from order m^2 to $m \log(m)$ by the proposed method for tracking [9], [18].

B. Complexity Analysis for Circulant Matrix

A circulant matrix $C \in \mathbb{R}^{m \times m}$ [29] is a structured matrix, which is completely defined by its first column so that to reconstruct the entire matrix, need to store the first column, where m is the sketch dimension. The space complexity is $\mathcal{O}(m)$ instead of $\mathcal{O}(m^2)$. Therefore, the space complexity for solving $\mathcal{O}(nm)$. Furthermore, the

circulant matrix can obtain a matrix-vector product ($C * V$, $V \in \mathbb{R}^m$) by the fast Fourier transform (FFT), whose time cost is $\mathcal{O}(m \log(m))$ [44] the time required for the same operation with the unstructured Gaussian sketch. Therefore, the time complexity for solving the sketch matrix-kernel matrix product (SK) in our method is $\mathcal{O}(nm \log(m))$. For details, see [11].

Most importantly, the effectiveness of a circulant matrix whose inputs in the first column are independent and identically distributed (i.i.d.) Gaussian inputs is almost the same as that of an unstructured matrix with i.i.d. Gaussian inputs [42]. Due to the advantages mentioned above, the circulant matrices have attracted extensive attention in some fields: approximation of the kernel matrices [43], [44], kernel selection [45], [46], approximation of the kernel function [41], binary embedding [47], ect. To the best of our knowledge, the circulant matrix based on the random sketch has not been applied to KRR, except for a theoretical justification. The purpose method of using the circulant matrix in our method is different from previous methods.

Proposed Algorithm Implementation Process

Despite recent advances in KCF tracking algorithms, researchers have paid less attention to the learning section of KCF tracking algorithms. Careful design of this part can solve some challenges, such as drifting and speed issues. Therefore, we propose a new method to improve the learning section of the KCF tracking algorithm. First, we take a look into the background of the learning section based on Kernel ridge regression (KRR).

The classical version of the KRR is well-known for solving complex statistical calculations based on the Hilbert transform. The goal of the KRR calculation is to produce an optimal approximation in the data set $\{(x_i, y_i)\}_{i=1}^n$ using the regression model. The mathematical expectation function \mathbb{E} between X and Y denoted by $f^*(x) = \mathbb{E}[Y|X = x]$.

Notice that the KRR method is based on the convex equation as [12]:

$$f = \arg \min_{f \in \mathcal{H}} \left\{ \frac{1}{2n} \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda_n \|f\|_{\mathcal{H}}^2 \right\} \quad (12)$$

The finite dimensions of n are used for optimizing the convex equations as:

$$\alpha = \arg \min_{\alpha \in \mathbb{R}^n} \left\{ \frac{1}{2} \alpha^T K^2 \alpha - \alpha^T \frac{K y}{\sqrt{n}} + \lambda_n \alpha^T K \alpha \right\} \quad (13)$$

$$f(\cdot) = \frac{1}{\sqrt{n}} \sum_{i=1}^n \alpha_i k(\cdot, x_i) \quad (14)$$

where α , $f(\cdot)$, λ_n and K are the quadratic program, estimate function, regularization parameter and kernel function, respectively. Superscript of T in (13) is the transpose operator. It should be noted that the temporal and memory complexities are from the order of $\mathcal{O}(n^3)$

and $\mathcal{O}(n^2)$, respectively, leading to an increase in computational complexity and memory and reducing the efficiency of the KRR algorithm.

A. Sketch Kernel Ridge Regression

Adding the circulant matrix to the KRR algorithm creates a new feature. Although the dimensions of the circulant matrix are small, it still needs to be made smaller to reduce the complex calculation and corresponding time. Since the large dimensions of the circulant matrix highly impact the computational speed, it motivated us to propose a sketch circulant matrix in video sequence trackers. This proposed method is called sketched KRR. It should be noted that the sketched KRR (SHKRR) has been previously introduced in statistical calculations [11]. However, this method is not employed in video sequence tracking yet. The SHKRR reduces the size of the kernel matrix and the speed of numerical calculations. In the SHKCF method, the sketched matrix is displayed by $S \in \mathbb{R}^{m \times n}$. This matrix S is extracted from the pattern vector of X. The kernel matrix is sketched, because the dimensions of the kernel matrix are converted from the order of $n \times n$ to the order of $m \times n$ ($m \ll n$). Generally, (17) has been applied to the SHKRR method [11]. Accordingly, the original KRR method is converted to the estimation of sketch kernel ridge regression (SKRR) by the sketch matrix. Therefore, we add (15) to the learning section of the KCF method and implement it in the Matlab platform, particularly for object tracking. This process is referred to as SHKCF in this paper.

$$\alpha' = \arg \min_{\alpha \in \mathbb{R}^m} \left\{ \frac{1}{2} \alpha'^T (SK)(KS^T) \alpha' - \alpha'^T \frac{SKy}{\sqrt{n}} + \lambda_n \alpha'^T SKS^T \alpha' \right\} \quad (15)$$

It should be noted that this equation is a quadratic sketched program with m dimensions, in which the equations of $(SK^2S^T$ and $SKS^T)$ operate as input m dimensional matrices and the equation SKy is m dimensional vector. To improve the computational, the sketched kernel matrix $SK = [SK_1, \dots, SK_n]$ in the input is calculated in way that parallelization technique is employed across its columns. In addition, this sketching idea can be extended to other kernel approaches regarding other loss functions that Characterizing of its properties is an interesting research subject for future works [44].

In a sliding window technique, the (16) can be evaluated on total of the sub-windows for fast detecting. However, to compute total of the responses simultaneously, one can exploit the circular technique [1].

$$f'(\cdot) = \frac{1}{\sqrt{n}} \sum_{i=1}^n (S^T \alpha')_i k(\cdot, x_i) \quad (16)$$

with

$$\alpha' = (SK^2S^T + 2\lambda_n SKS^T)^{-1} SKy \quad (17)$$

where K and I are the kernel matrix with elements $K_{ij} = k(x_i; x_j)$, and the identity matrix, respectively. It should be noted that parameter α' , as the estimation function, is needed for solving the KCF filter (H) in (1). The main difference between (6) and (13) is that the sketch matrix is integrated into (13). As a result, we will explain how to calculate the sketch matrix in detail for implementing the KCF method in object tracking (OT), and then we will explain the proposed algorithm in a flowchart.

The key to SKRR success is building the effective Sketch matrix in the tracking. Although many methods have been implemented to improve tracking through KRR upgrades, the Sketch circulant matrix has not yet been used in the tracking.

B. Sketch Kernel Ridge Regression Algorithm

As already mentioned, the main part of the proposed method is the sketched matrix (S matrix) in (11) which is added to the learning section of the KCF tracking.

$$S = \frac{1}{\sqrt{m}} DCQ \quad S \in \mathbb{R}^{m \times n}, D \in \mathbb{R}^{m \times m}, Q \in \mathbb{R}^{m \times n} \quad (18)$$

where D is Rademacher mapping between +1 or -1 which is similar to the behavior of the random diagonal matrix with a probability of $\frac{1}{2}$. m is the dimension of the sketch matrix. Q represents the sampling matrix which consists of a random subset of m rows from the $n \times n$ identity matrix.

Proposed algorithm: sketch KRR employing Circulant Matrix

Input: Data set $\{(x_i, y_i)\}_{i=1}^n$

Output: α'

- 1- Initialization: kernel parameters, sketched matrix m, the regularization parameter $\lambda_n \geq \frac{\sqrt{\log(n)}}{2n}$.
- 2- Construction of a random diagonal matrix $D \in \mathbb{R}^{m \times m}$, regarding the Rademacher variable
- 3- Construction of a circulant matrix $C \in \mathbb{R}^{m \times m}$, whose inputs in the first column are obtained from the standard normal distribution.
- 4- Construction of a variable kernel matrix $K' \in \mathbb{R}^{m \times n}$, which is based on the data sample of the Q matrix in Eq. (18).
- 5- Calculate $SK \in \mathbb{R}^{m \times n}$ using FFT, $S(SK)^T$.

Calculate $\alpha' = (SK^2S^T + 2\lambda_n SKS^T)^{-1} SKy$.

The Circulant Kernel Ridge Regression (CKRR) shows our sketch approach. The Gauss kernel is a typical and representative kernel, so in this paper we mainly use Gaussian kernel for experiments. The kernel function (7) is used for all types of feature representations. In the function (9), there are two parameters need to be determined in advance, i.e., σ and λ_n , the kernel

parameter and regularization parameter, respectively.

That the λ_n in our approach meets $\lambda_n = \frac{\sqrt{\log(n)}}{2n}$.

Implementation of the SHKCF tracker is simple, i.e. there is no type of heuristic methods for motion modeling or failure detection. Using video sequence dataset SHKCF tracker can train a model at the object's initial position in the first frame of the video sequence. To obtain some context, the searching window of SHKCF tracker is considered larger than the size of object. SHKCF tracker uses the previous position of the object over the window for each new frame. Then, the position of the object is updated to the one which yielded the greatest possible value. To obtain SHKCF tracker with some memory, SHKCF tracker train a new model in its new position. Finally, this tracker interpolates the value of α' linearly with its values which are updated with the following equations of linear interpolation [57].

$$\alpha'^t = (1 - \theta) * \alpha'^{t-1} + \theta * \alpha' \quad (18)$$

$$X'^t = (1 - \theta) * X'^{t-1} + \theta * X' \quad (19)$$

where α' , X' , t and θ denote the kernelized regularized Ridge regression, the object appearance, the t -th frame and the learning rate, respectively. In fact, this updating strategy can work well when changing of object appearance is very slow or there is no occlusion. To solve this problem, two indicators are introduced for evaluating whether the target is in occlusion challenge and tune the learning rate adaptively. If the object is in occlusion challenge the learning rate is reducing; if else the learning rate is fixed. These indicators are Peak-to-Sidelobe Ratio (PSR) [23] and appearance similarity(d). then we tune the learning rate of $\theta = \gamma * \theta_{in}$, if $d \leq 0.22$ and $PSR \leq 30$, $\theta = \theta_{in}$, otherwise. where γ and θ_{in} are the relative ratio to decrease the learning rate and the initialization value, respectively [57].

Fig. 1 shows the implementation steps of the proposed algorithm based on the sketch kernel. As can be seen in the proposed flowchart, a grayscale image sequence is first received to extract the object features. It should be noted that the coded color features of the object in grayscale image sequence are achieved by replicating the gray-scale image sequence into the its red, green and blue types and then extracting the object feature on this image as usual. The video sequence features are extracted by the HOG method. The circulant matrix is then generated and formulated by the first sequence patch of input data. A circulant matrix is used in the ridge regression kernel evaluated by the normal Gaussian distribution.

In the proposed algorithm, the coefficient α' is trained by calculating the sketched kernel correlation filter. Finally, the evaluation is performed to confirm the correct choice of the object. However, if the target is lost, the feature extraction operation is performed again by

updating the learning section of the algorithm for the current frame. If the error is less than the specified threshold, the evaluation of the tracking operation is correctly determined, and the next frame is extracted.

As already mentioned, the optimal value of α' is trained with the pattern X . Besides, the coefficient α' should be updated in each sequence of the flowchart (Fig. 1) [19], [35], [36]. The object in the next frame is estimated using the H filter in the search area of the current frame. In other words, the function $f(z; H) = H^T \phi(z)$ is applied to the search area, where z is the evaluation data and the mapping ϕ is the latest updated model of the z .

If the partial mapping ϕ is nonlinear, it can easily estimate several features of the object as a function $f(z; \alpha') = \alpha'^T \Phi_x \phi(z)$. It should be noted that linear mapping cannot estimate multiple features. According to the element-wise learning algorithm, the mapping z is calculated by considering the circular shifts element by element. Besides, the filter responses in the frequency domain are calculated as a function $f'(z; \alpha') = (k'^{\tilde{x}z} \odot \alpha')$, where $k'^{\tilde{x}z}$ is the estimate of the kernel matrix in the FFT of $\Phi_x \Phi_z^T$ [33], [35]. Deferent form the earlier type SHKCF tracker [29], it is developed to deal with multiple channels, as input arrays' third dimension. We implement SHKCF tracker by three functions: train (13), detect (14), and kernel correlation (7), which is demonstrated in Fig. 1.

Evaluating the Proposed Method with other Kernel Tracking Algorithms

For a fair comparison, some recent kernel-based video sequence tracking algorithms [29]-[34] are compared to the proposed algorithm. They have been evaluated on the OTB100/50 and VOT2016 challenges video sequence dataset and implemented by the same system (CPU: Core i7 RAM: 8GB, GPU: 4G). Fig. 2 shows the conventional challenges of OTB100/50 object tracking [7], including motion blur (MB), scale variation (SV), out of plane rotation (OPR), illumination variation (IV), occlusion (OCC), in of plane rotation (IPR), out of view (OV), background clutter (BC), low resolution (LR), fast motion (FM), deformation (DEF). For more clarity, three different kinds of one pass evaluation (OPE) are employed: 1- object accuracy, 2- object overlap and 3- qualitative comparison of algorithms. Note that OPE is to evaluate the threshold error of the object location and also overlap with ground truth.

Using a Gaussian kernel, we propose and implement a new object tracker based on the sketch kernelized correlation filter. Then we experimentally test more variants which works on HOG features with 4×4 cell size pixels, in particular variant [42], [53]. It should be noted that we employed adaptation rate (θ_{in}) = 0.02, γ = 0.1,

spatial bandwidth is $S = \frac{1}{\sqrt{m}}DCQ$ (that $m = 1.25\sqrt{\log(n)} \cong 2.17, n = 1024$), feature bandwidth $\sigma = 0.5$, and regularization $\lambda = \frac{\sqrt{\log(n)}}{2n} = 8.47 \times 10^{-4}$ [44], in our SHKCF tracker.

A. Evaluation Methods

To examination the tracker’s robustness the OPE technique is employed. In this way, The OPE evaluation operates on the object trackers just once in a video sequence. To analyze the trackers’ efficiency, plots of precision and success are shown in Fig. 3, Fig. 4 and Fig. 5. The threshold value t_0 differs in range between 0 and 1 for generating result of curves in the success plots. The threshold value of the success rate is bounded to 0.5 for evaluation process. In the other hand, the Euclidean

distance is calculated between the ground truth- and estimated-centers in precision plots, measures center location error (CLE) [12], [31], as follows:

$$CLE = v_{gp} = \sqrt{(l_g - l_p)^2 + (r_g - r_p)^2}, \quad (20)$$

where $(l_g; r_g)$ and $(l_p; r_p)$ are the ground truth center location and the predicted center location of the object in a frame, prospectively. During tracking, the average error metric cannot be used to accurately measure the tracking performance, as the tracker can lose the actual object’s location and the estimated location can be random. Instead, it may be a better performance metric to use the percentage of frames whose estimated location is within the specified threshold distance from the ground truth.

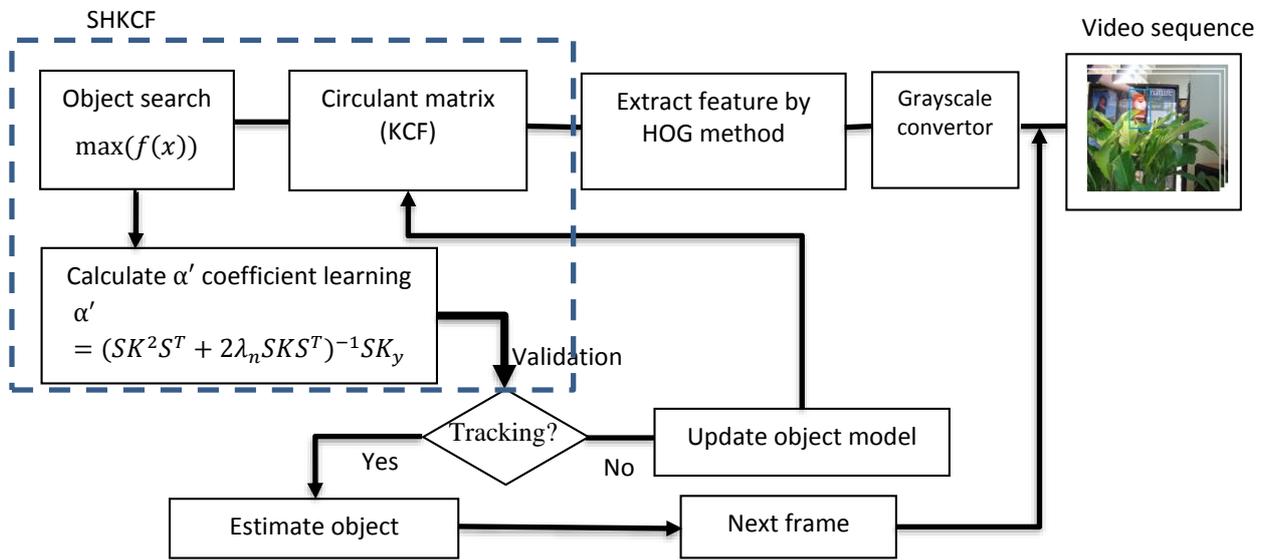


Fig. 1: Flowchart of the proposed tracking algorithm (SHKCF).

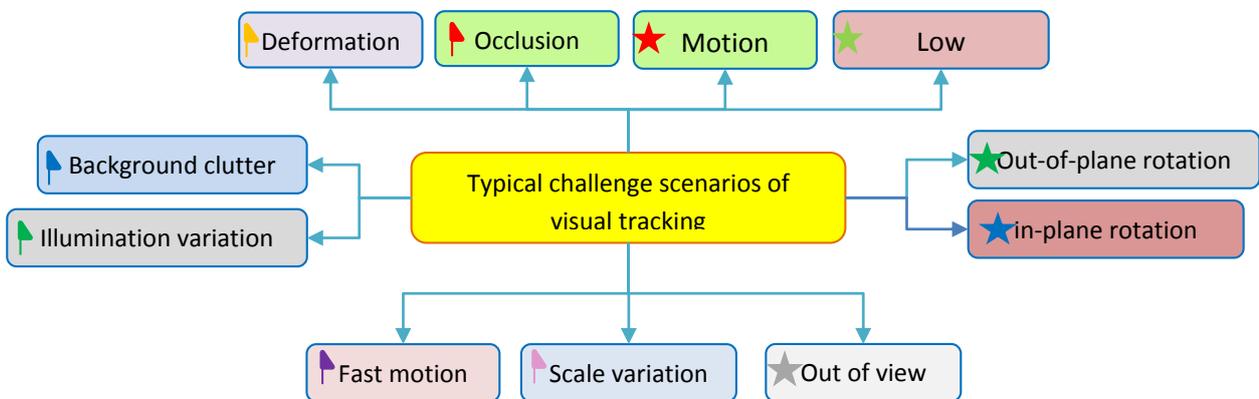


Fig. 2: Conventional challenges of video sequences for goal tracking, from [7].

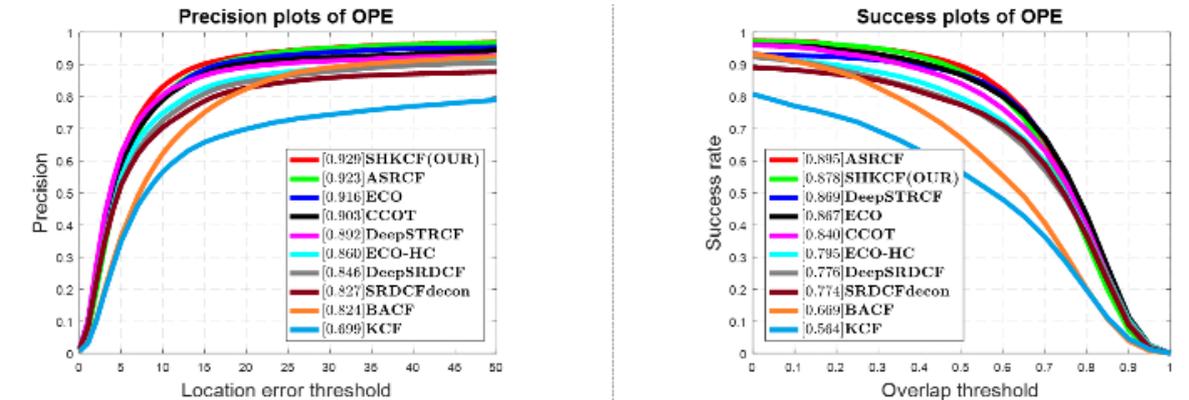


Fig. 3: Success plots (right) and precision plots (left) of our SHKCF tracker against the other ones (ECO [32], CCOT [35], ECO_HC, DeepSRDCF [52], SRDCFdecon [31], KCF [1], ASRCF [59], BACF [50] and DeepSTRCF [52])) on the OTB100 benchmark video sequence.

$$z = \frac{\sum_{n=1}^N \chi(v_{gp}^n)}{N} * 100, \quad (21)$$

$$\chi(v_{gp}^n) = \begin{cases} 1 & \text{if } v_{gp}^n \leq v_{th} \\ 0 & \text{otherwise} \end{cases}, \quad (22)$$

where N is all number of frames. Legends in the plots of precision demonstrate that precision regarding to a threshold of $v_{th} = 20$ pixels. Due to error of object center location just measures pixel difference, precision cannot produce a clear picture of estimated object shape and size. Thus, success plots have been employed as a more robustness measurement. In this way, an overlap score (OS) is computed between the ground truth- and the estimated-bounding box, which is based on area under the curve (AUC) [12], [31], as follows:

$$AUC = o_w = \frac{\text{area}(u_t \cap u_g)}{\text{area}(u_t \cup u_g)}, \quad (23)$$

where u_t , u_g , $|\cdot|$, \cap and \cup are the object bounding box, the ground-truth bounding box, the number of pixels, intersection and union of two regions, respectively. The overlap score is employed to demonstrate whether an object tracking algorithm has been successfully tracked an object in the frame. To demonstrate the successful frames o_w score should have more value than a threshold. Similar to precision, another performance metric is considered for computing of the overlap score percentage, as follows:

$$w = \frac{\sum_{n=1}^N \Gamma(o_w^i)}{N} * 100, \quad (24)$$

$$\Gamma(o_w^i) = \begin{cases} 1 & \text{if } o_w^i \leq t_0 \\ 0 & \text{otherwise} \end{cases}, \quad (25)$$

where N and t_0 are all number of frames and the overlap score threshold, respectively.

B. Quantitative Evaluation

The overall success performance and the precision of all the trackers over OTB100 are plotted in Fig. 3, where overlap precision (OP) metric is applied by computing the bounding box overlaps greater than 0.5 in a video sequence. In addition, we provide the overlap success plots containing the OP metric over a range of thresholds. We compare SHKCF tracker with 8 state-of-the-art trackers and one base tracker, including trackers (i.e. ECO [32], CCOT [35], ECO_HC, DeepSRDCF [52], SRDCFdecon [31], KCF [1], ASRCF [59], BACF [50] and DeepSTRCF [52])). It should be noted that for fair comparison, we use the publicly available codes or results provided by the authors. As can be seen, in Fig. 3 the result of precision plots of our SHKCF tracker have better than other trackers based on correlation filters and the result of success performance of our SHKCF tracker ranks second after ASRCF tracker with a difference of 0.017. To decrease the drifting problem, the distribution of correlation response is modeled in a sketch optimization framework by the SHKCF algorithm, making the object location in each frame more accurate.

For better understanding, overall comprehensive evaluations of nine top trackers are summarized in Table 1. Table 1 demonstrates that our SHKCF tracker has obtained the best results between three kernel correlation filter based trackers and also other type trackers. Comparing to KCF_DP [1], the SHKCF tracker gets a 32.9%, 55.7% and 78.4% improvement for AUC score, OP score and speed, respectively. Comparing to ASRCF [58], the SHKCF tracker achieves a 0.65% and 6.4% improvement for AUC score and speed, respectively. Also, the overall comprehensive evaluation results show that SHKCF promotes the KCF's performance [11] which

employee the identical scale strategy and features as our object tracker.

Table 1: Overall comprehensive evaluations of our SHKCF and other trackers

Tracker	Mean OP (CLE)	AUC (OS)
SHKCF(OUR+ HOG)	<i>0.878</i>	0.929
ASRCF	0.895	<i>0.923</i>
ECO	<u>0.867</u>	<u>0.916</u>
C-COT	0.840	0.903
DeepSTRCF	0.776	0.892
ECO-HC	0.795	0.860
DeepSRDCF	0.776	0.846
SRDCFdecon	0.774	0.827
BACF	0.669	0.824
KCF_DP	0.564	0.699

Table 2 and Table 3 demonstrate OPE partial evaluation of target accuracy based on OTB100 and OTB50 datasets. Basically, the output is based on the area under the curve (AUC).

The evaluation accuracy of the tracking algorithm is between zero and one.

If the output is closer to one, more similarity is obtained between the actual and estimated location of

the object. the evaluation was performed in 1500 different conventional video sequences, as OTB100 and OTB50.

Also, the output is an average number of 100 video sequences (OTB100) in Table 2 and Table 3. Ranking in Table 2 and Table 3 is depicted by green/Bold, red/Italic and blue/underline for first, second and third rank, respectively.

In addition, we demonstrate the tracking speed (FPS) comparison on OTB-2015 dataset in

Table 4 One can see that SHKCF (HOGCN) runs at 22.1 FPS. SHKCF (HOG) using HOG feature performs even faster and obtains a real-time speed of 35.4 FPS, which is 1.6× and 1.13× faster than BACF and STRCF tracker, respectively.

As shown in Table 2, the proposed algorithm wins in IPR, BC and FM challenges, Location error overall and Precision OPE (fps). The results of Table 2 demonstrate that the SHKCF method is a good choice considering object tracking accuracy since it gains one of the best ranks in most challenges.

The results of Table 3 demonstrate that the SHKCF method a lower performance in OCC and MB challenges than modern trackers. Fig. 4 and Fig 5 demonstrate the precision and success plots of the proposed method and other SOTA methods, respectively.

The ground-truth location in the first frame acts as an initial value for evaluating the test sequence.

It is worth mentioning that the algorithms are ranked based on AUC.

Table 2: OPE partial evaluation of object accuracy based on AUC calculations

Tracker	illumination Variation [36][36]	Out-of-Plane [63]	Scale Variation [64]	Motion Blur [30]	Occlusion [48]	In-plane-rotation [52]	Out-of-view [15]	Background Clutter[31]	Low resolution [9]	Fast motion [39]	Deformation [44]
SHKCF(OUR)	<i>0.913</i>	<u>0.915</u>	<i>0.905</i>	0.867	0.889	0.910	0.834	0.943	<u>0.942</u>	0.907	<i>0.916</i>
ASRCF	0.924	0.929	0.906	<u>0.881</u>	<u>0.909</u>	<i>0.897</i>	0.918	<u>0.929</u>	0.999	<i>0.902</i>	0.917
ECO	<u>0.906</u>	<i>0.918</i>	<u>0.892</u>	0.888	0.927	<u>0.887</u>	<i>0.914</i>	<i>0.936</i>	0.882	<u>0.897</u>	0.869
C-COT	0.869	0.903	0.885	<i>0.887</i>	<i>0.921</i>	0.862	<u>0.890</u>	0.862	<i>0.977</i>	0.896	<u>0.871</u>
DeepSTRCF	0.829	0.884	0.871	0.833	0.881	0.830	0.839	0.846	0.881	0.844	0.861
ECO-HC	0.811	0.845	0.832	0.798	0.860	0.802	0.825	0.846	0.888	0.840	0.813
DeepSRDCF	0.766	0.840	0.825	0.817	0.820	0.817	0.787	0.820	0.847	0.831	0.782
SRDCFdecon	0.825	0.803	0.815	0.809	0.779	0.774	0.654	0.839	0.747	0.787	0.754
BACF	0.780	0.811	0.802	0.763	0.758	0.820	0.667	0.771	0.925	0.781	0.805
KCF	0.725	0.689	0.647	0.601	0.642	0.702	0.526	0.715	0.700	0.637	0.629

Table 3: Partial evaluation of OPE object overlap

Tracker	illumination Variation [36]	Out-of-Plane [63]	Scale Variation [64]	Motion Blur [30]	Occlusion [48]	In-plane-rotation [52]	Out-of-view [15]	Background Clutter [31]	Low resolution [9]	Fast motion [39]	Deformation [44]
SHKCF(OUR)	<u>0.867</u>	0.868	0.844	0.860	0.847	0.847	<u>0.794</u>	0.883	0.807	<u>0.857</u>	0.857
ASRCF	0.900	0.889	0.858	0.875	0.882	0.847	0.856	0.907	0.808	0.874	0.868
ECO	0.868	0.863	0.841	0.881	0.865	0.829	0.800	<u>0.853</u>	0.717	0.863	<u>0.831</u>
C-COT	0.815	0.823	0.817	<u>0.862</u>	<u>0.861</u>	0.775	0.791	0.779	0.738	0.838	0.803
DeepSTRCF	0.811	<u>0.864</u>	0.841	0.833	0.856	<u>0.802</u>	0.789	0.812	<u>0.802</u>	0.832	0.829
ECO-HC	0.767	0.764	0.749	0.775	0.787	0.715	0.740	0.792	0.607	0.785	0.764
DeepSRDCF	0.717	0.753	0.754	0.781	0.736	0.734	0.676	0.724	0.713	0.777	0.708
SRDCFdecon	0.764	0.738	0.765	0.796	0.736	0.715	0.663	0.761	0.679	0.755	0.693
BACF	0.621	0.673	0.560	0.728	0.618	0.678	0.617	0.709	0.341	0.684	0.654
KCF	0.560	0.555	0.449	0.558	0.520	0.570	0.492	0.630	0.304	0.557	0.527

Table 4: The FPS results of trackers on OTB-2015. The best three results are shown in green, blue and red fonts, respectively.

	ECO_HC	SRDCF	SRDCFDecon	KCF_DP	STRCF	DeepSRDCF	CCOT	BACF	SHKCF(HOGCN)	SHKCF (HOG)
FPS	15.6	5.8	2.0	16.7	31.5	5.3	0.3	<u>26.7</u>	22.1	35.4

The number in the parentheses specifies how many sequences are used in each challenging scenario. According to the precision of OPE in Fig. 4, our SHKCF algorithm finds the best result results in some challenging scenarios (MB, OCC and OV). Moreover, the experimental result of Fig. 4 shows that our SHKCF method achieves second and third rank in other challenging scenarios. The horizontal and vertical axis in each precision plot of OPE indicates the location error and overlap threshold of the

OT algorithm, respectively (Fig. 4 and Fig. 5). Comparing results between the proposed method and the top ten related works demonstrate that our method has a close or even better performance than other methods. Note that all methods are implemented using the same platform challenges and processor. The final result of the AUC calculation is presented at the bottom right of each precision plot in Fig. 4.

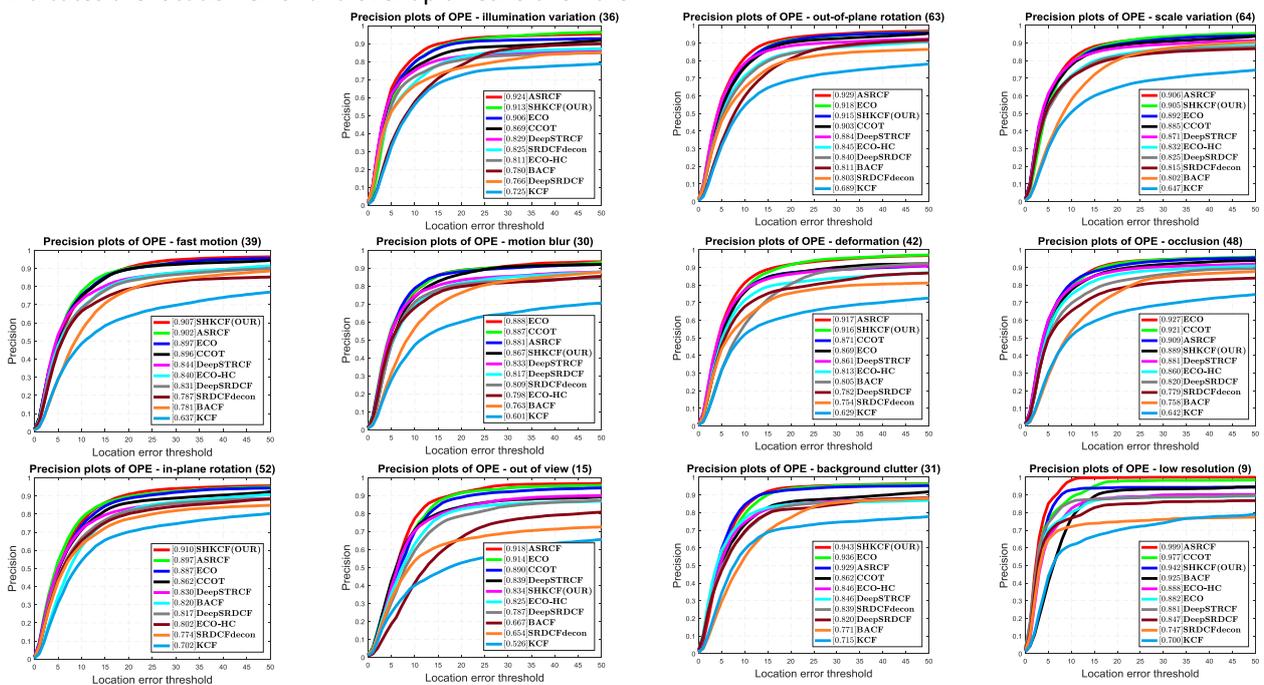


Fig. 4: Precision distance plot of our SHKCF tracker and other trackers (ECO [32], CCOT [35], ECO_HC, DeepSRDCF [52], SRDCFdecon [31], KCF [1], ASRCF [59], BACF [50] and DeepSTRCF [52])) on OTB100 [12] benchmark on eleven different challenges (IV, OPR, SV, FM, MB, Def, OCC, IPR, OV, BC and LR). The legend has score at a threshold of 20 pixels for each object tracker.

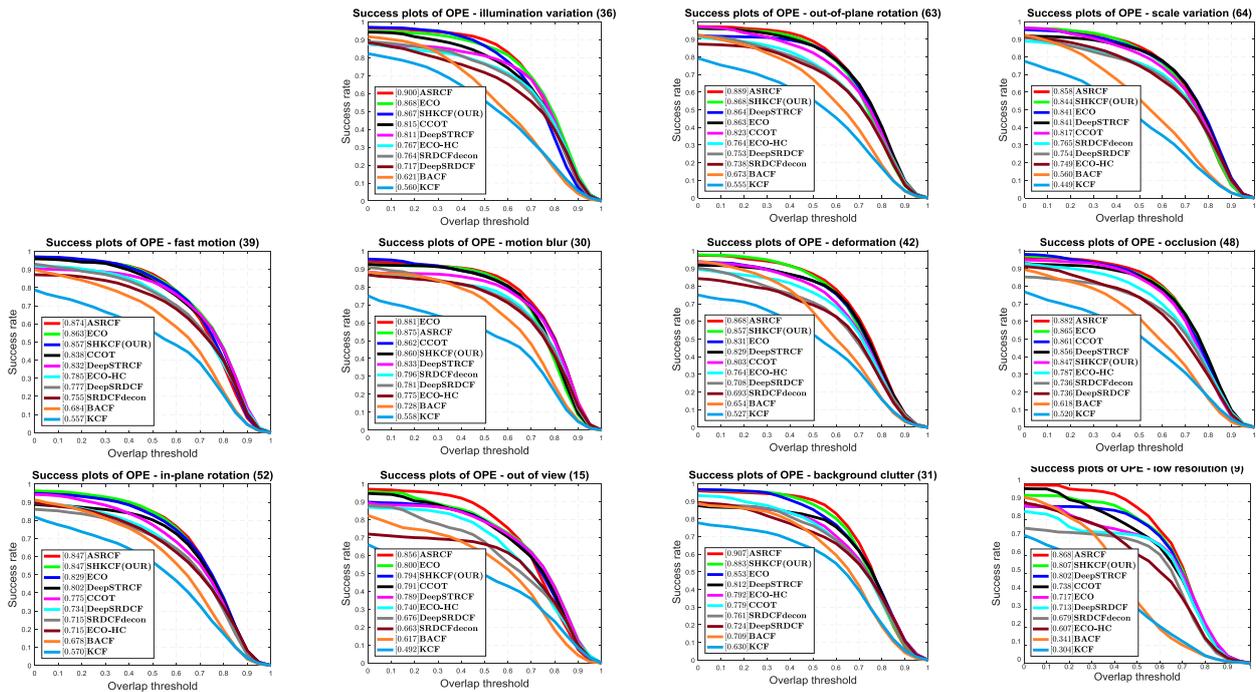


Fig. 5: Overlap success plots of our SHKCF tracker and other trackers (ECO [32], CCOT [35], ECO_HC, DeepSRDCF [52], SRDCFdecon [31], KCF [1], ASRCF [59], BACF [50] and DeepSTRCF [52])) on OTB100 [12] benchmark on eleven different challenges (IV, OPR, SV, FM, MB, Def, OCC, IPR, OV, BC and LR). The legend has score at a threshold 0.5 pixels for each object tracker.

C. Qualitative Evaluation

For more clarity, Fig. 6 shows the result of the qualitative comparisons evaluated between SOTA works (e.g., ECO, CCOT, ECO_HC, DeepSRDCF, SRDCFdecon, KCF, ASRCF, BACF, DeepSTRCF) and SHKCF. In order to cover overall the tracking challenges, random sequences are selected as shown in Fig. 6, which is including Soccer, Matrix, Ironman, Basketball, Bolt, Bird, Deer, Biker and Panda. These samples include several available challenges of the OTB 100 benchmark.

Soccer video sequence sample:

In this sample with 383 frames evaluates on SV, OCC, MB, IPR, OPR and BC challenges. According to experimentally result Fig. 6, in frame 30, all trackers were able to track the target correctly, it should be noted that in its 70th frame SV, MB and IPR challenges are evaluated, and the tracking drift is observed in the evaluation of KCF and DeepSRDCF trackers. In frame 110, four trackers (KCF, BACF, ARSCF and DeepSTRCF) completely lost the target, and five trackers (ECO, C_COT, ECO_HC, SRDCFdecon and DeepSRDCF) partially overlap the target, and only the proposed tracker of SHKCF tracker succeeds in tracking the target. Farther more in frame 345, KCF and DeepSRDCF trackers completely lost the target and the other six trackers and our tracker found the target position and overlapped the target.

Matrix video sequence sample:

In this sample with 91 frames evaluates on IV, SV, OCC, FM, IPR, OPR and BC challenges.

According to Fig. 6, in frame 15, DeepSRDCF and C_COT trackers have partial overlap and the other trackers and our tracker have full overlap. At frame 40, except for the ECO and C_COT trackers which have partial overlap, the all other trackers lost the target. In the next frame of matrix sample, except the ECO and C_COT trackers which have complete overlap, the all other trackers lost the target. In this example, the proposed tracker lost the target from frame 40 onwards, and the tracker could not correct itself in the next frames.

In this sample with 91 frames evaluates on IV, SV, OCC, FM, IPR, OPR and BC challenges. According to Fig. 6, in frame 15, DeepSRDCF and C_COT trackers have partial overlap and the other trackers and our tracker have full overlap. At frame 40, except for the ECO and C_COT trackers which have partial overlap, the all other trackers lost the target. In the next frame of matrix sample, except the ECO and C_COT trackers which have complete overlap, the all other trackers lost the target. In this example, the proposed tracker lost the target from frame 40 onwards, and the tracker could not correct itself in the next frames.

Ironman video sequence sample:

In this sample with 157 frames evaluates on IV, OCC, FM, IPR, OPR, OV and BC challenges. According to Fig. 6, in frame 36, the BACF and DeepSRDCF trackers the target is lost the target, the other trackers have partial overlap, and the proposed tracker has the most overlap with the target. At frame 54, four trackers (BACF, KCF, DeepSRDCF and SRDCFdecon) have lost the target, and one tracker

(DeepSRTCF) has partial overlap and the other trackers, including the proposed SHKCF tracker, have full target overlap.

Basketball video sequence sample:

In this sample with 716 frames evaluates on IV, OCC, FM, IPR, OPR, OV and BC challenges. According to Fig. 6, in frame 30, only the SRDCFdecon tracker lost the target. At frame 650, the ECO tracker with partial occlusion, the target window is larger than the ground-truth and does not overlap completely. At frame 717, the ECO tracker misses the target completely, the SRDCFdecon tracker has partial overlap, and the other trackers track the target correctly.

Bolt video sequence sample:

In these sample with 341 frames evaluates on OCC, OPR and BC challenges. According to Fig. 6, except for the SRDCFdecon tracker, the other trackers found the target correctly.

Bird video sequence sample:

In this sample with 716 frames evaluates on OCC, DEF, FM, IPR, OPR and OV challenges. According to Fig. 6, in frame 20, except for the KCF tracker which has a partial target overlap, the other trackers found the target correctly. At frame 123, three SRDCFdecon, KCF, ECO_HC trackers lost the target. At frame 182, the proposed tracker found the target completely, and the two trackers have partial overlap, and the other trackers lost the target. In frame 391, only two SHKCF and ASRCF trackers found the target correctly.

Deer video sequence sample:

In this sample with 62 frames evaluates on MB, FM, IPR and BC challenges. According to Fig. 6, in frame 20, all trackers track the target correctly. In frame 30, three trackers (DeepSRDCF, ECO and C_COT) lost the target.

At frame 40, only one tracker (C_COT) lost the target and the other trackers found the target correctly. At frame 40, all trackers found the target correctly.

Biker video sequence sample:

In this sample with 133 frames evaluates on OCC, MB, FM, OPR, OV and LR challenges. According to Fig. 6, in frame 68, three trackers (DeepSRDCF, BACF and C_COT) lost the target and other trackers, including the proposed SHKCF tracker, follow the target correctly. In frames 80, 84, and 140, the proposed tracker and five other trackers (SHKCF, ECO_HC, DeepSRTCF, KCF, ECO and SRDCFdecon) found the tracker with proper overlap.

Panda video sequence sample:

In this sample with 991 frames evaluates on DEF, IPR, OPR, OV and LR challenges. According to Fig. 6, in frame 110, the trackers are tracking the target correctly. At frame 213, only one tracker (SRDCFdecon) has missed the target, and other trackers, including the proposed tracker, are tracking the target. At frame 345, the proposed tracker and seven other trackers found the target with proper overlap. At frame 645, three trackers (C_COT, DeepSRDCF and SRDCFdecon) lost the target and the proposed tracker found the target correctly with other modern trackers.

The SHKCF, ASRCF, ECO, the DeepSRDCF, the CCOT and the SRDCFdecon succeeded in object tracking in a clean environment. To highlight the strange of our tracker against other trackers, quality evaluation in the illumination variation of the object, clearly shows that our tracker can only track the object in the "Matrix" challenges. These quantity output results corroborate that our method has better experimental results than other works. As mentioned before, we employ the same parameters and the protocol generated in the OTB100 for all video sequences.

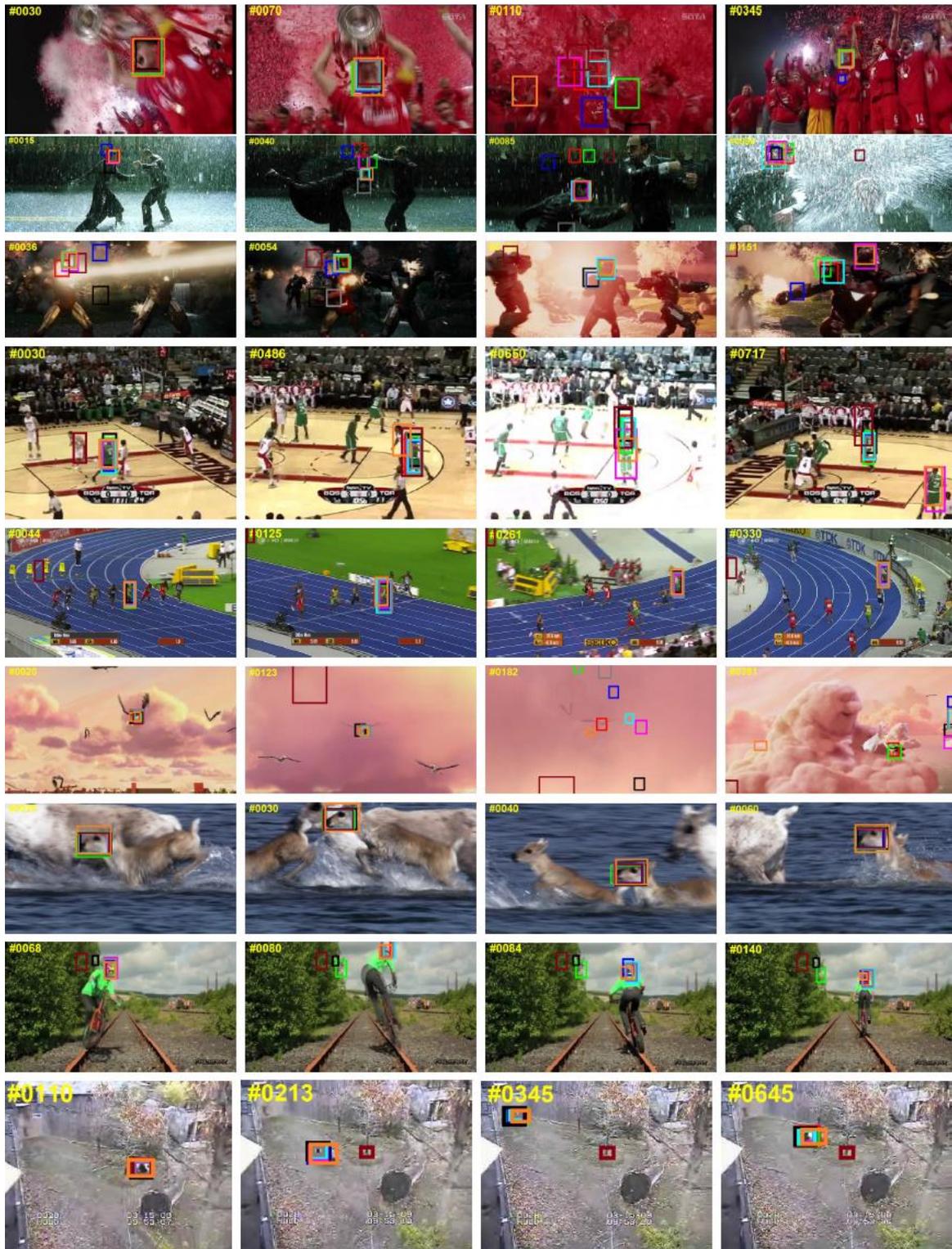
Table 5: Evaluation on VOT2016 benchmark by expected average overlap (EAO), Accuracy and robustness

	ECO	SRDCF	BACF	SRDCFDecon	ECO_HC	DeepSTRCF	SHKCF (OUR)
EAO	0.369	0.249	0.221	0.259	0.329	0.318	<u>0.321</u>
Accuracy	0.52	0.51	0.58	0.51	<u>0.54</u>	0.56	<u>0.54</u>
Robustness	0.78	1.61	1.80	1.49	1.13	<u>0.96</u>	0.89

A. The VOT-2016 Benchmark

We demonstrate the results on VOT2016 [21] benchmark, the VOT2016 benchmark consists of 60 video sequences. We evaluate the trackers of expected average overlap (EAO), accuracy and robustness [31].

The EAO measures the average without-reset overlap of a tracker over several video sequences. The accuracy computes the average overlap ratio between the predicted and the ground-truth box. And the robustness averages the number of tracking failures on the video sequence.



— SHKCF — ASRCF — DeepSRTCF — BACF — ECO — C-COT — ECO-HC — DeepSRDCF — SRDCFdecon — KCF

Fig. 6: Qualitative analysis of trackers SHKCF(OUR), ECO [32], CCOT [35], ECO_HC, DeepSRDCF [52], SRDCFdecon [31], KCF [1], ASRCF [59], BACF [50] and DeepSTRCF [52] over OTB100 [12] benchmark on eleven challenging sequences (from up to down Soccer, Matrix, Ironman, Basketball, Bolt, Bird, Deer, Biker and Panda respectively).

Results and Discussion

We compare SHKCF with state-of-the-art trackers, including ECO [32], SRDCF [52], BACF [50], SRDCFDecon [31], ECO_HC [32] and DeepSTRCF [52]. Table 5 demonstrates the results of different trackers on VOT-2016 dataset.

We can see from Table 5 that SHKCF performs significantly better than the DeepSTRCF, BACF and SRDCF methods in terms of the EAO metric. In addition, SHKCF also performs favorably against DeepSTRCF, BACF and SRDCF by a gain of 2.6%, 4.4% and 2.9% in EAO metric, respectively.

Compared to modern trackers, the SHKCF tracker was ranked second and three in robustness and accuracy, respectively.

Conclusion

In this paper, we have proposed, analyzed, and implemented the sketch kernel correlation filter (SHKCF) for object tracking.

The proposed method improves the basic correlation filter trackers. Although the sketch matrix theory was first proposed in regression, this method was not employed in object tracking scenarios which motivated us to employ it in the KCF basic algorithm. We exploit the learning section of the filter by integrating a new parameter α with original KCF trackers.

To speed up learning and detection, the element-wise matrix is trained by a sketch algorithm. The element-wise matrix is developed by a circulant matrix to sketch method.

Experimental results on OTB100 and OTB50 standard challenges such as MB, SV, OP, IV, OCC, IPR, OV, BC, LR, FM demonstrate that the SHKCF algorithm can further develop the original KCF tracker performance compared to most of the state-of-art works. In addition, the SHKCF method provides optimal optimization for scaling and occlusion problems.

Besides, the proposed algorithm shows that it has better accuracy and robustness compared to other trackers based on in-depth training, ECO, CCOT, ECO_HC, DeepSRDCF, SRDCFdecon, KCF, ASRCF, BACF and DeepSTRCF.

Finally, the experimental results of the quality comparison of our SHKCF method with other SOTA works related to the online CFT demonstrate the better performance of the proposed algorithm. In order to developed our SHKCF method for the object tracking, the BACF algorithm may be incorporated to the Sketch coefficient for the learning model, leading to increase the learning speed.

Therefore, in addition to the advantages the BACF algorithm such as efficient background target modeling,

the mention boundary effect of these new incorporating method of BACF. and SHKCF can solves the local minimum problem BACF.

This new algorithm may perform better in various challenges.

Author Contributions

M. Yousefzadeh, A. Golmakani, and Gh. Sarbishaei designed the experiments. M. Yousefzadeh collected the data. M. Yousefzadeh carried out the data analysis. M. Yousefzadeh, A. Golmakani, and Gh. Sarbishaei interpreted the results and wrote the manuscript.

Acknowledgment

The author gratefully acknowledges the sadjad university A. Golmakani, gh. Sarbishaei for their work on the original version of this document.

Conflict of Interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

Abbreviations

<i>MB</i>	Motion Blur
<i>SV</i>	Scale Variation
<i>OPR</i>	Out of Plane Rotation
<i>IV</i>	Illumination Variation
<i>OCC</i>	Occlusion
<i>IPR</i>	In of Plane Rotation
<i>OV</i>	Out of View
<i>BC</i>	Background Clutter
<i>LR</i>	Low Resolution
<i>FM</i>	Fast Motion
<i>DEF</i>	Deformation
<i>OPE</i>	One Pass Evaluation
<i>CLE</i>	Center Location Error
<i>AUC</i>	Area Under the Curve
<i>KCF</i>	Kernel Correlation Filter

References

- [1] J. F. Henriques, R. Caseiro, P. Martins, J. Batista, "Highspeed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(3): 583–596, 2014.

- [2] P. Zhang, S. Yu, J. Xu, Xinge You, X. Jiang, X. Y. Jing, D. Tao, "Robust visual tracking using multi-frame multi-feature joint modeling," *IEEE Trans. Circuits Syst. Video Technol.*, 29(12): 3673–3686, 2018.
- [3] B. Chen, P. Li, C. Sun, D. Wang, G. Yang, H. Lu, "Multi attention module for visual tracking," *Pattern Recognit.*, 87: 80–93, 2019.
- [4] P. Li, D. Wang, L. Wang, H. Lu, "Deep visual tracking: Review and experimental comparison," *Pattern Recognit.*, 76: 323–338, 2018.
- [5] T. Xu, Z. H. Feng, X. J. Wu, J. Kittler, "Learning low-rank and sparse discriminative correlation filters for coarse-to-fine visual object tracking," *IEEE Trans. Circuits Syst. Video Technol.*, 30(10): 3727–3739, 2019.
- [6] B. Zhang, Z. Li, X. Cao, Q. Ye, C. Chen, L. Shen, A. Perina, R. Jill, "Output constraint transfer for kernelized correlation filter in tracking," *IEEE Trans. Syst. Man Cybern. Syst.*, 47(4): 693–703, 2016.
- [7] H. Wang, H. Xu, Y. Yuan, "High-dimensional expensive multi-objective optimization via additive structure," *Intell. Syst. Appl.*, 14: 200062, 2022.
- [8] C. Du, M. Lan, M. Gao, Zh. Dong, H. Yu, Zh. He, "Real-time object tracking via adaptive correlation filters," *Sensors*, 20(15): 4124, 2020.
- [9] M. Y. Abbass, K. C. Kwon, N. Kim, S. A Abdelwahab, F. E. A. El-Samie, A. A. Khalaf, "A survey on online learning for visual tracking," *Visual Comput.*, 37: 993–1014, 2021.
- [10] X. Zhang, G. S. Xia, Q. Lu, W. Shen, L. Zhang, "Visual object tracking by correlation filters and online learning," *ISPRS J. Photogramm. Remote Sens.*, 140: 77–89, 2018.
- [11] R. Yin, Y. Liu, W. Wang, D. Meng, "Sketch kernel ridge regression using circulant matrix: Algorithm and theory," *IEEE Trans. Neural Networks Learn. Syst.*, 31(9): 3512–3524, 2019.
- [12] B. Yan, H. Zhao, D. Wang, H. Lu, X. Yang, "Skimming-perusal tracking: A framework for real-time and robust longterm tracking," in *Proc. the IEEE/CVF International Conference on Computer Vision*: 2385–2393, 2019.
- [13] S. Avidan, "Ensemble tracking," *IEEE transactions on pattern analysis and machine intelligence*, 29(2):261–271, 2007.
- [14] J. Zheng, Bo. Li, P. Tian, G. Luo, "Robust object tracking using valid fragments selection," in *Proc. Multi Media Modeling: 22nd International Conference, Part I 22*: 738–751, 2016.
- [15] H. Yang, S. Qu, "Online hierarchical sparse representation of multifeature for robust object tracking," *Comput. Intell. Neurosci.*, 330-341, 2016.
- [16] D. T Nguyen, C. D Nguyen, R. Hargraves, L. A Kurgan, K. J. Cios, "mids: Multiple-instance learning algorithm," *IEEE Trans. Cybern.*, 43(1): 143–154, 2012.
- [17] D. Comaniciu, V. Ramesh, P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(5): 564–577, 2003.
- [18] M. Danelljan, G. Hager, F. Shahbaz Khan, M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. the IEEE International Conference on Computer Vision*: 4310–4318, 2015.
- [19] H. K. Galoogahi, A. Fagg, S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proc. the IEEE International Conference on Computer Vision*: 1135– 1143, 2017.
- [20] Q. Guo, R. Han, W. Feng, Zh. Chen, L. Wan, "Selective spatial regularization by reinforcement learned decision making for object tracking," *IEEE Trans. Image Proc.*, 29: 2999–3013, 2019.
- [21] S. Hadfield, R. Bowden, K. Lebeda, "The visual object tracking vot2016 challenge results," *Lect. Notes Comput. Sci.*, 9914: 777–823, 2016.
- [22] M. Danelljan, G. Hager, F. Shahbaz Khan, M. Felsberg, "Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking," in *Proc. the IEEE conference on Computer Vision and Pattern Recognition*: 1430–1438, 2016.
- [23] M. Danelljan, G. Bhat, F. Shahbaz Khan, M. Felsberg, "Eco: Efficient convolution operators for tracking," in *Proc. the IEEE conference On Computer Vision and Pattern Recognition*: 6638–6646, 2017.
- [24] A. Yilmaz, O. Javed, M. Shah, "Object tracking: A survey," *Acm Comput. Surv. (CSUR)*, 38(4): 13–es, 2006.
- [25] X. Mei, H. Ling, Y. Wu, E. Blasch, L. Bai, "Minimum error bounded efficient l1 tracker with occlusion detection," in *Proc. CVPR*: 1257–1264, 2011.
- [26] Sh. Shekhar, N. Hoque, D. K. Bhattacharyya, "Pknnmifs: A parallel knn classifier over an optimal subset of features," *Intell. Syst. Appl.*, 14: 200073, 2022.
- [27] D. A. Ross, J. Lim, et al., "Incremental learning for robust visual tracking," *Int. J. Comput. Vision*, 77(1-3): 125-141, 2008.
- [28] Y. Li, J. Zhu, et al., "A scale adaptive kernel correlation filter tracker with feature integration," in *Proc. ECCV workshops (2)*, 8926: 254–265, 2014.
- [29] M. Danelljan, G. Häger, F. Khan, M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. British Machine Vision Conference, Bmva Press*, 2014.
- [30] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernandez, T. Vojir, G. Hager, G. Nebehay, R. Pflugfelder, "The visual object tracking vot2015 challenge results," in *Proc. the IEEE International Conference on Computer Vision Workshops*: 1–23, 2015.
- [31] H. Lian, F. Zhang, W. Lu, "Randomized sketches for kernel cca," *Neural Networks*, 127: 29–37, 2020.
- [32] A. Mahalanobis, B. V. Kumar, D. Casasent, "Minimum average correlation energy filters," *Appl. Opt.*, 26(17): 3633– 3640, 1987.
- [33] B. V. Kumar, "Minimum-variance synthetic discriminant functions," *JOSA A*, 3(10): 1579–1584, 1986.
- [34] D. S. Bolme, J. R. Beveridge, B. A. Draper, Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*: 2544–2550, 2010.
- [35] S. Liu, D. Liu, G. Srivastava, D. Polap, M. Woźniak, "Overview of correlation filter based algorithms in object tracking," *Complex Intell. Syst.*, 7: 1895–1917, 2020.
- [36] M. Danelljan, A. Robinson, F. Sh. Khan, M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Proc. Computer Vision–ECCV 2016: 14th European Conference*: 472–488, 2016.
- [37] T. Liu, G. Wang, Q. Yang, "Real-time part-based visual tracking via adaptive correlation filters," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*: 4902– 4912, 2015.
- [38] C. Ma, X. Yang, Ch. Zhang, M. H. Yang, "Long-term correlation tracking," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*: 5388–5396, 2015.
- [39] D. S. Bolme, J. R. Beveridge, B. A. Draper, Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*: 2544–2550, 2010.

[40] G. Turin, "An introduction to matched filters," IRE Trans. Inf. Theory, 6(3): 311–329, 1960.

[41] E. Gundogdu, A. A. Alatan, "Good features to correlate for visual tracking," IEEE Trans. Image Process., 27(5): 2526–2540, 2018.

[42] J. F. Henriques, R. Caseiro, P. Martins, J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in Proc. Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Part IV 12: 702–715, 2012.

[43] Y. Yang, M. Pilanci, M. J. Wainwright, "Randomized sketches for kernels: Fast and optimal nonparametric regression," arXiv preprint arXiv:1501.06195, 2017.

[44] L. Čehovin, A. Leonardis, M. Kristan, "Visual object tracking performance measures revisited," IEEE Trans. Image Process., 25(3): 1261–1274, 2016.

[45] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, D. Ramanan, "Object detection with discriminatively trained part-based models," IEEE Trans. Pattern Anal. Mach. Intell., 32(9): 1627–1645, 2009.

[46] Q. He, X. Shao, W. Chen, X. Li, X. Yang, T. Sun, "Adaptive multi-scale tracking target algorithm through drone," IEICE Trans. Commun., 102(10): 1998–2005, 2019.

[47] K. Zhang, J. T Kwok, "Clustered nystrom method for large scale manifold learning and dimension reduction," IEEE Trans. Neural Networks, 21(10): 1576–1587, 2010.

[48] Sh. Xuan, Sh. Li, M. Han, X. Wan, G. S. Xia, "Object tracking in satellite videos by improved correlation filters with motion estimations," IEEE Trans. Geosci. Remote Sens., 58(2): 1074–1086, 2019.

[49] M. Ding, W. H. Chen, L. Wei, Y. F. Cao, Z. Y. Zhang, "Visual tracking with online assessment and improved sampling strategy," IEEE Access, 8: 36948–36962, 2020.

[50] J. P. Sun, E. J. Ding, B. Sun, Zh. Y. Liu, K. L. Zhang, "Adaptive kernel correlation filter tracking algorithm in complex scenes," IEEE Access, 8: 208179–208194, 2020.

[51] G. Zhu, J. Wang, Y. Wu, X. Zhang, H. Lu, "Mchog correlation tracking with saliency proposal," in Proc. the AAAI Conference on Artificial Intelligence, 30, 2016.

[52] M. Danelljan, A. Robinson, F. S. Khan, M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in Proc. Computer Vision–ECCV 2016: 14th European Conference, Part V 14: 472–488, 2016.

[53] X. Li, Q. Liu, Zh. He, H. Wang, Ch. Zhang, W. S. Chen, "A multi-view model for visual tracking via correlation filters," Knowledge-Based Systems, 113: 88–99, 2016.

[54] Y. Zhang, L. Wang, J. Qi, D. Wang, M. Feng, H. Lu, "Structured siamese network for real-time visual tracking," in Proc. the European Conference on Computer Vision (ECCV): 351–366, 2018.

[55] X. Hao, J. Huang, F. Qin, X. Zheng, "Multi-label learning with missing features and labels and its application to text categorization," Intell. Syst. Appl., 14: 200086, 2022.

[56] M. Tang, B. Yu, F. Zhang, J. Wang, "High-speed tracking with multi-kernel correlation filters," in Proc. the IEEE Conference on Computer Vision and Pattern Recognition: 4874–4883, 2018.

[57] P. Dollár, R. Appel, S. Belongie, P. Perona, "Fast feature pyramids for object detection," IEEE Trans. Pattern Anal. Mach. Intell., 36(8): 1532–1545, 2014.

[58] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, J. Yan, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," in

Proc. the IEEE/CVF conference on Computer Vision and Pattern Recognition: 4282–4291, 2019.

[59] K. Dai, D. Wang, H. Lu, C. Sun, J. Li, "Visual tracking via adaptive spatially-regularized correlation filters," in Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition: 4670–4679, 2019.

Biographies



Morteza Yousefzadeh He was born in the city of Mahmutabad-Mazandaran in Iran. He studied B.A. in Noshirvani University of Babol and M.A. in Malek Ashtar University. His work experience includes digital electronic systems and image processing. His specific areas of interest include artificial intelligence, image processing, and image tracking algorithms. He is a

Ph.D. student in Electrical and Electronic Engineering from Sajjad University of Mashhad.

- Email: morteza.usefzadeh@gmail.com
- ORCID: [0009-0006-2611-1616](https://orcid.org/0009-0006-2611-1616)
- Web of Science Researcher ID: NA
- Scopus Author ID: NA
- Homepage: NA



Abbas Golmakani He was born in Mashhad in Iran. He studied B.A. and M.A. in Sharif University of Technology. His work experience includes designing analog, digital and high frequency integrated circuits - IoT systems. His special areas of interest include integrated circuit design, image processing. He studied Ph.D. Electrical and Electronics Engineering at Ferdowsi

University of Mashhad.

- Email: golmakani@sadjad.ac.ir
- ORCID: [0009-0006-6061-4380](https://orcid.org/0009-0006-6061-4380)
- Web of Science Researcher ID: NA
- Scopus Author ID: NA
- Homepage: <https://profile.sadjad.ac.ir/golmakani/>



Ghazaleh Sarbishaei She was born in Mashhad in Iran. She studied B.A. and M.A. in Ferdowsi University of Mashhad. His work experience includes telecommunication systems and signal processing. His special fields of interest include telecommunication systems and signal processing. She studied Ph.D. Electrical and Telecommunication Engineering at Ferdowsi University of

Mashhad.

- Email: gh_sarbisheie@sadjad.ac.ir
- ORCID: [0000-0002-7590-2928](https://orcid.org/0000-0002-7590-2928)
- Web of Science Researcher ID: NA
- Scopus Author ID: NA
- Homepage: <https://profile.sadjad.ac.ir/sarbisheie>

How to cite this paper:

M. Yousefzadeh, A. Golmakani, G. Sarbishaei, "Design, analysis, and implementation of a new online object tracking method based on Sketch Kernel Correlation Filter (SHKCF)," J. Electr. Comput. Eng. Innovations, 12(1): 115-132, 2024.

DOI: [10.22061/jecei.2023.10126.680](https://doi.org/10.22061/jecei.2023.10126.680)

URL: https://jecei.sru.ac.ir/article_1989.html

