IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received November 14, 2020, accepted December 13, 2020, date of publication December 22, 2020,
date of current version January 6, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3046510

Tracking With CNN Based Correlation Filters on Spherical Manifolds

MINGKE ZHANG[1], XUANDE ZHANG[1], AND LONG XU[2]
[1]School of Electronic Information and Artificial Intelligence, Shaanxi University of Science and Technology, Xi'an 710021, China
[2]Key Laboratory of Solar Activity, National Astronomical Observatories, Chinese Academy of Sciences, Beijing 100101, China

Corresponding author: Xuande Zhang (zhangxuande@sust.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61871260.

ABSTRACT The correlation filters tracker allows features from multiple channels. The fusion of features by simply summing over them in an Euclidean space would destroy the inherent geometry among multiple features, resulting in the lose of their phase information which is crucial to tracking task. To provide a better fusion of features from multiple layers of a convolutional neural network (CNN) in the classical CNNs based correlation filters algorithm, we introduce spherical manifolds and computing intrinsic mean on spherical manifolds in the article, so that fusion of features and online update of filter kernels can be implemented over a spherical manifolds. In addition, we introduce a random projection method, imposed on CNN features before feature fusion to compress features for the sake of reducing computational complexity and modeling complexity. Extensive experiments on OTB-50 dataset demonstrate that the proposed algorithm outperforms state-of-the-art methods with respect to both precision and success rate.

INDEX TERMS Deep features, fusion of multiple kernels, object tracking, random projection, spherical manifolds.

I. INTRODUCTION

The mainly task of tracking is estimating the location of a visual target in each frame of an image sequence. It has various practical applications, especially for human-machine interactions, visual surveillance and unmanned control systems [1]–[3]. Despite significant progress has been achieved in recent years, object tracking is still one of the most challenging problems in computer vision owing to factors, such as partial occlusion, deformation, scale variations, illumination variation, background clutter, in-plane/out-of-plane rotations and motion blur [4]–[6].

In recent years, Correlation filter (CF) based discriminative algorithms have gained high attention owning to high accuracy of object tracking and low computational complexity [7]. To estimate an object's translation in spatial domain efficiently, the classical CF method exploits the dense sampling in the target region via a circulant shift matrix, by which the tracking algorithm can allow an element-wise operation in the Fourier domain, and the target location is inferred by searching for the maximum value of the response map. Meanwhile, CF is proposed to process multi-channel hand-crafted features, and shows the better performance in object tracking [4], [8].

With the rapid development of deep learning, lots of tracking algorithms [9]–[12] based on the features extracted from the deep convolutional neural networks (CNNs) were proposed, and achieved great improvements compared with traditional methods. It has been shown that the CNNs-based trackers perform well against methods using hand-crafted features, such as Scale Invariant Feature Transform(SIFT) [13], HOG [14] and color histogram. For CNNs based correlation filters, the response map of a layer with D channels is computed by

$$r = \mathcal{F}^{-1}\Big(\frac{\sum_{i=1}^{D}\hat{\mathbf{x}}_i^{*}\cdot\hat{\mathbf{z}}_i}{\sum_{i=1}^{D}\hat{\mathbf{x}}_i^{*}\cdot\hat{\mathbf{x}}_i+\lambda}\hat{\mathbf{y}}\Big) = \mathcal{F}^{-1}\Big(\frac{\hat{K}^{\mathbf{xz}}}{\hat{K}^{\mathbf{xx}}+\lambda}\hat{\mathbf{y}}\Big) \quad (1)$$

where $\hat{\mathbf{x}} \in \mathbb{C}^{P\times Q\times D}$ is the target object represented by CNN features from multiple channels, $\hat{\mathbf{y}} \in \mathbb{C}^{P\times Q}$ denotes the Gaussian shape label matrix, $\hat{\mathbf{z}} \in \mathbb{C}^{P\times Q\times D}$ denotes the candidate object. They are all in Fourier domain. The operator $\mathcal{F}^{-1}$ denotes the inverse Fast Fourier Transform(FFT).

In the above CNNs features approaches, an essential step is computing kernels (i.e., the numerator and denominator in the Eq. 1, we denote the numerator as cross

The associate editor coordinating the review of this manuscript and approving it for publication was Hiu Yung Wong.

correlation kernel, denote denominator as autocorrelation kernel). To construct the kernel, the CF algorithm allows simply summing over every kernel built by single channel feature in the Fourier domain, which is equivalent to fuse the kernels of the multi-channel features.

Since visual tracking only focus on the trajectory of the target object, phase information plays a significant role. It is well known that an object's translation is only determined by the phase information of the response map, the corresponding amplitude information is redundant. Adapting the fusion by Eq. 1, we will introduce the unwanted energy information of each feature, which may have no advantage on the tracking performance.

To address this issue, spherical manifolds is introduced to fuse the kernels from multiple channels over an unit hypersphere, allowing a substitution of the mean in a hypersphere for the mean in the normal Euclidean space. Thus, phase information of the features from multiple channels is kept, benefiting object tracking task. However, computing the mean in a hypersphere concerns an iterative process, so computational load would be expensive in case of high dimensional features. To reduce computational complexity, rand projection algorithm is introduced to compress the dimension of CNNs features before feature fusion.

The main contributions of this article are listed in the following items:

1) We propose a novel method to generate random projection matrix based on spectral graph theory, which avoids the high time-consuming operation for Gram-Schmidt orthogonalization during rand projection process.
2) We propose a novel CNNs features based CF algorithm on spherical manifolds.
3) We propose an online update strategy of CF kernels based on geometry of spherical manifolds.
4) We carry out extensive experiments on a large-scale benchmark dataset to demonstrate the effectiveness of the proposed algorithm in comparisons to the state-of-the-art trackers.

II. RELATED WORKS

This work is closely related to CF algorithms. In this section, a brief overview concerning CF tracking is presented.

In 2010, Bolme *et al.* first proposed the Minimum Output Sum of Square Error (MOSSE) [15] tracking algorithm. The MOSSE algorithm uses intensity features which is single channel for object representation to train the filter, converts the convolution into the product in the frequency domain, and achieves a tracking speed with 669 frames per second. In 2012, Henriques *et al.* proposed the Exploiting the Circulant Structure of Tracking-by-detection with Kernels(CSK) algorithm [8] based on cyclic structure and ridge regression, which achieved remarkable results at high processing speeds. However, CSK algorithm uses intensity features like MOSSE, which is insufficient in feature selection. In 2014, Danelljan *et al.* proposed the Color Name(CN) tracking algorithm [16], they used color attribute features which is multiple channels instead of intensity features to achieve better tracking results. In 2015, Henriques *et al.* [4] proposed a multi-channel version of the tracker which used HOG features allowing the representation of objects through 31 dimensional features. Having considered the unwanted boundary effects which can severely degrade the performance of the tracking model, Danelljan *et al.* proposed Spatially Discriminative Regularized Correlation Filters(SRDCF) algorithm [7] in the same year.

Due to the rapid development of deep CNNs, in 2016, several trackers [12], [17], [18] based on Siamese networks were introduced, and became a new research hotspot because of their simplicity and competitive performance. In 2017, Valmadre *et al.* [19] improved the Fully-Convolutional Siamese Networks(SiamFC) [12] tracker by integrating discriminative correlation filters into the Siamese framework. For the methods which focus on integrating convolutional features from a fixed pre-trained deep network, in 2015,2016,2017, Danelljan *et al.* proposed Deep SRDCF algorithm [9], Continuous Convolution Operator Tracker(C-COT) algorithm [20] and Efficient Convolution Operator(ECO) tracking algorithm [11], these algorithms were all based on the deep CNNs features and achieved a remarkable performance compared with those based on the hand-crafted features. In 2016, Ma *et al.* designed an effective correlation filters tracker Hierarchical Convolutional Features for Visual Tracking(HCFT) [21] on each CNN layer, and obtained the target location from the multi-level response maps in a coarse-to-fine fashion. In 2018, they proposed a robust tracker HCFT* [22], in which they applied the classifier to two types of region proposals for scale estimation and target redetection from tracking failures. In 2016, Qi *et al.* combined several weak CNNs trackers from numerous convolutional layers into a stronger one named Hedged Deep Tracking(HDT) algorithm [23], which introduces an improved Hedge algorithm by considering historical performance of weak trackers. In 2018, Qi *et al.* introduce Siamese network to improve the original HDT algorithm by defining the loss of each weak tracker for the proposed hedge method [24]. It has been proved that the CNN model is very suitable for developing robust appearance model in the tracking task, due to its powerful ability on feature extraction [25].

Generally speaking, CF based on hand-crafted features allows less accurate or robust when faced with more complex scenarios, due to the fact that hand-crafted features are usually specially designed for certain aim, while it can run at a high speed. On the other hand, CF using CNNs features can achieve a great progress in terms of accuracy and robustness, but it has a limitation in the tracking speed.

III. PREVIEW OF THE CNN BASED CORRELATION FILTERS

The CNN based Correlation Filters (TCCF) [10] employedthe same hedge method to HDT [23], additionally equipped-witha scale estimation module. It is the baseline of our

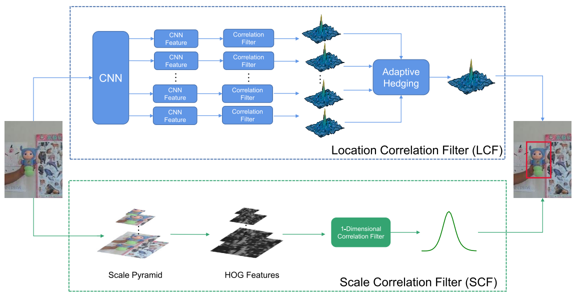


FIGURE 1. The architecture of the TCCF tracker.

TABLE 1. The parameter settings and feature map output size of VGG-16.

Layer	Kernel Size	Output Size
Input		224*224*3
Conv 1-1	3*3	224*224*64
Conv 1-2	3*3	224*224*64
Pool 1	2*2	112*112*64
Conv 2-1	3*3	112*112*128
Conv 2-2	3*3	112*112*128
Pool 2	2*2	56*56*128
Conv 3-1	3*3	56*56*256
Conv 3-2	3*3	56*56*256
Conv 3-3	3*3	56*56*256
Pool 3	2*2	28*28*256
Conv 4-1	3*3	28*28*512
Conv 4-2	3*3	28*28*512
Conv 4-3	3*3	28*28*512
Pool 4	2*2	14*14*512
Conv 5-1	3*3	14*14*512
Conv 5-2	3*3	14*14*512
Conv 5-3	3*3	14*14*512
Pool 5	2*2	7*7*512
fc 1	1*1	1*1*4096
fc 2	1*1	1*1*4096
fc 3	1*1	1*1*1000

proposedtracker, consistingof a Local CorrelationFilter (LCF) and a Scale CorrelationFilter (SCF) as demonstrated in Fig. 1 [10]. For the LCF, CNN features from multiple layers of the pre-trained VGG-16 [26] are used to represent the target appearance. The parameter settings of VGG-16 are given in Table.1.

A. LOCATION ESTIMATION

Use $\mathbf{x}^k \in \mathbb{R}^{P\times Q\times D}$ denotes the feature map extracted from the k-th convolutional layer, the regression target $\mathbf{y}$ is defined by a 2D Gaussian shape label matrix. Let $\hat{\mathbf{x}}^k = \mathcal{F}(\mathbf{x}^k)$ and $\hat{\mathbf{y}} = \mathcal{F}(\mathbf{y})$, where $\mathcal{F}$ denotes the Discrete Fourier Translation (DFT). In the Fourier domain, the k-th deserved filter can be computed by

$$\hat{\mathbf{f}}^k = arg\min_{\hat{\mathbf{f}}} \| \sum_{i=1}^{D} \hat{\mathbf{f}}_i \cdot \hat{\mathbf{x}}_i^k - \hat{\mathbf{y}}\|_2^2 + \lambda \sum_{i=1}^{D} \|\hat{\mathbf{f}}_i\|_2^2 \tag{2}$$

The solution to Eq.2 can be given by

$$\hat{\mathbf{f}}_i^k = \frac{\hat{\mathbf{y}} \cdot \hat{\mathbf{x}}_i^{k*}}{\sum_{i=1}^{D} \hat{\mathbf{x}}_i^{k*} \cdot \hat{\mathbf{x}}_i^k + \lambda} = \hat{\alpha}^k \hat{\mathbf{x}}_i^{k*} \tag{3}$$

where $\cdot$ denote element-wise product, and the division is also performed element-wise. The $\hat{\mathbf{x}}_i^{k*}$ denote the complex conjugation of $\hat{\mathbf{x}}_i^k$, $\hat{\alpha}^k$ has the form as

$$\hat{\alpha}^k = \frac{\hat{\mathbf{y}}}{\sum_{i=1}^{D} \hat{\mathbf{x}}_i^{k*} \cdot \hat{\mathbf{x}}_i^k + \lambda} = \frac{\hat{\mathbf{y}}}{\hat{K}^{\mathbf{x}^k\mathbf{x}^k} + \lambda} \tag{4}$$

Given the testing data $\mathbf{z}^k$, having transformed the data to the Fourier domain by $\hat{\mathbf{z}}^k = \mathcal{F}(\mathbf{z}^k)$, we can compute the response maps by

$$r^k = \mathcal{F}^{-1}(\hat{\alpha}^k \sum_{i=1}^{D} \hat{\mathbf{z}}_i^k \cdot \hat{\mathbf{x}}_i^{k*}) = \mathcal{F}^{-1}(\hat{K}^{\mathbf{x}^k\mathbf{z}^k} \hat{\alpha}^k) \tag{5}$$

Then the k-th tracker outputs the target position with the largest response

$$(x^k, y^k) = arg\max_{x,y} r^k(x, y) \tag{6}$$

Considering that there are K convolutional layers will be used, the final location (x', y') is obtained by

$$(x', y') = \sum_{k=1}^{K} w_k(x_k, y_k) \tag{7}$$

where $w_k \geq 0$ and $\sum_{k=1}^{K} w_k = 1$, which can be determined according to the performance of each tracker [23].

B. SCALE ESTIMATION AND MODEL UPDATE

According to the Discriminatiive Scale Space Tracker(DSST), the SCF can be implemented after the location estimation, which is independent of the translation filters.

Here, HOG feature is used, a brief comment on the HOG for scale estimation can be found in [27]. For the SCF, a set of scale factors are predefined $\{\alpha_j = \theta^{\lceil \frac{j}{2} - j \rceil} | j = 1, 2, \ldots, J, \theta > 1\}$. Given a training sample, J image patches are cropped around the estimated target position, for a scale factor α_j, the corresponding image patch has the size of $\alpha_j P \times \alpha_j Q$, where $P \times Q$ is the size of the target in the previous frame. After reshaping these scaled image patches, a one dimension Correlation Filter will be applied, the deserved scale can be obtain.

For more details about the scale estimation and model update, refer to [10], [27].

IV. PROPOSED TRACKER

A. FUSION METHOD OF KERNELS BASED ON SPHERICAL MANIFOLD GEOMETRY

Since visual tracking focus on the trajectory of the target object, which is only determined by the phase of the response map. Therefore, to remove the impact of amplitude of each kernel, we consider a linear kernel with normalized energy, which can be expressed as:

$$\hat{K}_i^{\mathbf{x}\mathbf{x}'} = \frac{\hat{\mathbf{x}}_i^* \cdot \hat{\mathbf{x}}_i'}{\|\hat{\mathbf{x}}_i^* \cdot \hat{\mathbf{x}}_i'\|_2} \qquad i = 1, 2, \cdots, D \tag{8}$$

where $\|\cdot\|_2$ denotes the L_2 norm, D is the number of the channels. Since $\|\hat{K}_i^{\mathbf{x}\mathbf{x}'}\|_2 = 1$, a natural geometry structure is introduced to these kinds of the kernels with unit norm, i.e., they are embedded on a unit hypersphere. Taking the geometry of spherical manifolds into account, we need to implement the summations in Eq. 1 in a spherical space, allowing a substitution of the mean in a hypersphere for that in the normal Euclidean space.

1) SPHERICAL MANIFOLDS AND OPERATORS ON SPHERICAL MANIFOLDS

A low-dimension spherical manifolds is embedded in a high-dimension Euclidean space. Let $\mathbb{C}^N$ be a N-dimensional Euclidean space, a unit sphere is given by $S^{N-1} = \{p \in \mathbb{C}^N | \|p\|_2 = 1\}$. Two basic operators, Log and Exp, are defined for spherical manifolds.

To project a point $q \in S^{N-1}$ to the tangent space of $p \in S^{N-1}$, denoted by T_pS^{N-1}, a Log operator is defined as [28]

$$Log_p q = q - \langle p, q\rangle p \tag{9}$$

To project $g \in T_pS^{N-1}$ into the spherical manifolds, an Exp operator is defined as [28]

$$Exp_p g = pcos(\|g\|_2) + g\frac{sin(\|g\|_2)}{\|g\|_2} \tag{10}$$

From Eq. 9 and Eq. 10, we can know $Log_p q \in T_pS^{N-1}$, $Exp_p g \in S^{N-1}$.

2) FUSION METHOD OF KERNELS BASED ON SPHERICAL MANIFOLD GEOMETRY

Let $\{p_1, p_2, \cdots, p_n\} \subset S^{N-1}$ denotes points on a spherical manifolds, we compute the intrinsic mean to fuse them together, which is defined as

$$\bar{p} = arg \min_{p \in S^{N-1}} \sum_{i=1}^{n} d(p, p_i)^2 \tag{11}$$

where $d(\cdot,\cdot)$ denotes Riemannian distance on S^{N-1}. It indicates that the intrinsic mean is a point on spherical manifolds which has the minimum sum-of-squared distance to all of given points. In this sense, the intrinsic mean is similar to the clustering center in clustering methods.

From Eq. 11, computing $\bar{p}$ is an optimization problem, by minimizing a sum-of-squared distance function as

$$f(p) = \frac{1}{2n}\sum_{i=1}^{n} d(p, p_i)^2 \tag{12}$$

In addition, given an assumption that all points are confined in a strongly convex neighborhood, Karcher [29] shown that the gradient of Eq. 11 was given by

$$\nabla f(p) = -\frac{1}{n}\sum_{i=1}^{n} Log_p p_i \tag{13}$$

The minimum of $f(p)$ is obtained as the gradient $\nabla f(p) = 0$, namely the stationary point. According to Eq. 11, intrinsic mean has the minimum sum-of-squared distance to $\{p_1, p_2, \cdots, p_n\} \subset S^{N-1}$, so we have the following equation

$$\sum_{i=1}^{n} Log_{\bar{p}} p_i = 0 \tag{14}$$

Algorithm 1 The Fusion Method of Kernels based on Spherical Manifold Geometry

Input: features $\hat{\mathbf{x}}$, $\hat{\mathbf{x}}'$, $\varepsilon > 0$, L.
Output: $\hat{K}^{\mathbf{xx}'}$

```
1) Compute each kernel by Eq. 8.
2) Compute initial value $\hat{K}_0^{\mathbf{xx}'}$ of iteration by Eq. 17.
3) for $m = 1 : L$
4)     Project $D$ kernels onto the tangent space of $\hat{K}_{m-1}^{\mathbf{xx}'}$ by Eq. 9.
5)     Compute $\hat{K}_{m-1}^{\mathbf{xx}'*}$ by Eq. 18.
6)     Project $\hat{K}_{m-1}^{\mathbf{xx}'*}$ back to the spherical manifold $\hat{K}_m^{\mathbf{xx}'}$ by Eq. 10.
7)     if $\|\hat{K}_{m-1}^{\mathbf{xx}'*}\|_2 \leq \varepsilon$ then
8)         break.
9)     end
10) end
```

which means

$$Exp_{\bar{p}}(\frac{1}{n}\sum_{i=1}^{n} Log_{\bar{p}} p_i) = \bar{p} \tag{15}$$

Thus, computing $\bar{p}$ is to find the stationary point of Eq. 15, which is an iterative process as [30]

$$\bar{p}_{m+1} = Exp_{\bar{p}_m}(\frac{1}{n}\sum_{i=1}^{n} Log_{\bar{p}_m} p_i) \tag{16}$$

where m denotes the number of iterations. Although the convergence of this process is not guaranteed in a general manifold, it is well behaved on the hypersphere [31].

Considering the kernels in Eq. 8, we can apply the same iterative process to fuse them under the constraint of the spherical manifold structure, the algorithm is given in Algorithm 1.

The initial value of iteration $\hat{K}_0^{\mathbf{xx}'}$ can be formulated as:

$$\hat{K}_0^{\mathbf{xx}'} = \frac{\sum_{i=1}^{D} \hat{\mathbf{x}}_i^* \cdot \hat{\mathbf{x}}_i'}{\| \sum_{i=1}^{D} \hat{\mathbf{x}}_i^* \cdot \hat{\mathbf{x}}_i' \|_2} \tag{17}$$

The summation of vector in tangent space will be gotten

$$\hat{K}_{m-1}^{\mathbf{xx}'*} = \frac{1}{D}\sum_{i=1}^{D} Log_{\hat{K}_{m-1}^{\mathbf{xx}'}} \hat{K}_i^{\mathbf{xx}'} \tag{18}$$

B. RANDOM PROJECTION FOR THE DIMENSION REDUCTION

As mentioned in the introduction, in visual tracking using the features extracted by the fine-tuning pre-trained CNNs, the dimension of these features is usually very high, computing the mean in a hypersphere will be time-consuming. To conduct a more efficient calculation, we need reduce the dimension of the CNNs features. Here, the random projection is considered.

Random projection is a powerful means of dimensionality reduction, where the original high-dimensional data is projected into a lower-dimensional subspace using a random matrix. Unlike most transform-based dimensionality-reduction techniques which are highly data dependent, random projection is data-independent and computationally more efficient than other widely used dimensionality reduction methods, such as principal component analysis and maximum noise fraction transform. Random projection has been applied to various areas and demonstrated good performance, including information retrieval, machine learning, remote sensing, and so on [32]–[35].

Random projection is composed of random matrix creation and matrix multiplication [34]. The projection matrix is usually chosen as an orthogonal matrix. For a high-dimentional feature, Gram-Schmidt orthogonalization would be very time-consuming [33]. In this work, spectral graph theory is employed to generate a random orthogonal matrix very fast. Given a undirected, connected and weighted graphs $g = \{v, \varepsilon, \mathbf{W}\}$, where v is a finite set of vertices with $|v| = D$, ε is a set of edges, and $\mathbf{W} \in \mathbb{R}^{D\times D}$ is a weighted adjacency matrix. The non-normalized graph Laplacian is defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{W} \tag{19}$$

where $\mathbf{D}$ is a diagonal matrix with $D_{ii} = \sum_j W_{ij}$.

As the graph Laplacian $\mathbf{L}$ is a real symmetric matrix, it has a complete set of orthonormal eigenvectors, From $\mathbf{L}$, an orthogonal projection matrix can be generated as the following processings.

1) Generate a random vector $\mathbf{R} = [r_1, r_2, \cdots, r_D]^T$ with independent identically distributed random variables, each element is uniformly distributed on [0, 1].
2) Calculate the matrix $\mathbf{W} = \mathbf{R}\mathbf{R}^T$, then get the graph Laplacian $\mathbf{L}$ by Eq.19.
3) Decompose the graph Laplacian $\mathbf{L}$ as $\mathbf{U}\Lambda\mathbf{U}^T$. where $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_D]$ is a matrix of the Laplacian eigenvectors, obviously, $\mathbf{u}_i$ satisfies $\|\mathbf{u}_i\|_2 = 1$ and $\sum_j u_{ij} = 0 \quad i = 2, 3, \cdots, D$.
4) Calculate the sparse measure of each eigenvector by
$$S_i = \|\mathbf{u}_i\|_1 = \sum_{l=1}^{D} |u_{il}| \qquad i \neq 1 \tag{20}$$
5) Choose the eigenvectors with the top $d(d \ll D)$ sparse measures to construct the random projection matrix
$$\mathbf{p} = [\mathbf{u}_{(1)}, \mathbf{u}_{(2)}, \cdots, \mathbf{u}_{(d)}] \in \mathbb{R}^{D\times d} \tag{21}$$
where $S_{(1)} \geq S_{(2)} \geq \cdots \geq S_{(d)} \geq S_{(D-1)}$.

After getting the projection matrix, we can implement the random projection by Algorithm 2 [36].

C. UPDATING CORRELATION FILTERS ON THE GEOMETRY OF SPHERICAL MANIFOLDS

In a video sequence, the target can often change appearance by changing its rotation, pose or lighting conditions. The strategy to update filter plays an important role during the tracking process.

Algorithm 2 The projection algorithm

```
Input: the feature $\mathbf{X} \in \mathbb{R}^{P\times Q\times D}$, projection matrix $\mathbf{P} \in \mathbb{R}^{D\times d}$  $(d \ll D)$.
Output: the feature $\mathbf{Y}$ after random projection.
  1) Reformate the feature into a feature matrix denoted as $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_D] \in \mathbb{R}^{L\times D}$  $L = P \times Q$.
  2) Compute the average feature by $\bar{\mathbf{X}} = \frac{1}{D}\sum_{i=1}^{D} \mathbf{X}_i$.
  3) Centralize the features in $\mathbf{X}'$ by $\mathbf{X}'_i = \mathbf{X}_i - \bar{\mathbf{X}}(i = 1, 2, \cdots, D)$.
  4) Project the centered features into the random subspace by $\mathbf{Y}' = \mathbf{X}'\mathbf{P} = [\mathbf{y}'_1, \mathbf{y}'_2, \cdots, \mathbf{y}'_d]$.
  5) Obtain a reduced set of features by $\mathbf{Y} = [\mathbf{y}'_1 + \bar{\mathbf{x}}, \mathbf{y}'_2 + \bar{\mathbf{x}}, \cdots, \mathbf{y}'_d + \bar{\mathbf{x}}]$.
  6) Reformat $\mathbf{Y}$ into a tensor $P \times Q \times d$.
  7) end
```

Algorithm 3 The update strategy of filter

```
Input: $\hat{K}^{\mathbf{x}_{t-1}\mathbf{x}_{t-1}}$, $\hat{K}^{\mathbf{x}_t\mathbf{x}_t}$, learning rate $\eta$.
Output: $\hat{K}'^{\mathbf{x}_t\mathbf{x}_t}$
  1) Project $\hat{K}^{\mathbf{x}_t\mathbf{x}_t}$ onto the tangent space of $\hat{K}^{\mathbf{x}_{t-1}\mathbf{x}_{t-1}}$ by Eq. 9 and get $\hat{K}^{\mathbf{x}_t\mathbf{x}_t *}$.
  2) Project $\eta\hat{K}^{\mathbf{x}_t\mathbf{x}_t *}$ back to the spherical manifolds by Eq. 10 and get the Updated autocorrelation kernel $\hat{K}'^{\mathbf{x}_t\mathbf{x}_t}$.
  3) Compute the Updated filter by using Eq. 24.
  4) end
```

The classical correlation filters algorithm allows the filter to be updated by weighting them in the normal Euclidean space:

$$\hat{\alpha}'_t = (1-\eta)\hat{\alpha}_{t-1} + \eta\hat{\alpha}_t \tag{22}$$

where t is the number of frames, $\eta > 0$ is the learning rate.

Since the autocorrelation kernel (i.e., the denominator in the Eq. 1) has been embedded in a spherical manifold, update strategy can be divided into two steps: update the autocorrelation kernel and then compute the filter.

$\hat{K}'^{\mathbf{x}_t\mathbf{x}_t}$ can be updated as:

$$\hat{K}'^{\mathbf{x}_t\mathbf{x}_t} = Exp_{\hat{K}^{\mathbf{x}_{t-1}\mathbf{x}_{t-1}}}(\eta Log_{\hat{K}^{\mathbf{x}_{t-1}\mathbf{x}_{t-1}}}\hat{K}^{\mathbf{x}_t\mathbf{x}_t}) \tag{23}$$

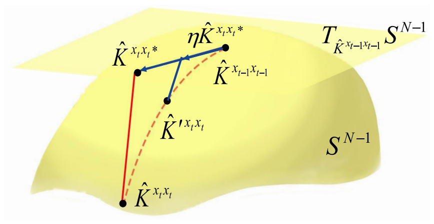
where $\hat{K}^{\mathbf{x}_{t-1}\mathbf{x}_{t-1}}$ denotes the autocorrelation kernel of $t-1$ frame, $\hat{K}^{\mathbf{x}_t\mathbf{x}_t}$ denotes the autocorrelation kernel of t frame, $\eta > 0$ is learning rate, this procedure is illustrated in Fig. 2.

Hence, the updated filter $\hat{\alpha}_t$ can be learnt as below:

$$\hat{\alpha}_t = \frac{\hat{\mathbf{y}}}{\hat{K}'^{\mathbf{x}_t\mathbf{x}_t} + \lambda} \tag{24}$$

D. WHOLE FRAMEWORK FOR THE PROPOSED TRACKER

Putting above the algorithms together, we will first obtain the weak tracker of k-th convolutional layer and its corresponding target position estimation as Algorithm 4.



$\hat{K}^{x_t x_t *}$
$\eta\hat{K}^{x_t x_t *}$
$T_{\hat{K}^{x_{t-1}x_{t-1}}} S^{N-1}$
$\hat{K}^{x_{t-1}x_{t-1}}$
$\hat{K}'^{x_t x_t}$
S^{N-1}
$\hat{K}^{x_t x_t}$

FIGURE 2. Illustration of the procedure for updating the autocorrelation kernel on the hypersphere.

Algorithm 4 Tracking algorithm

```
Input: the initial state of the target.
Output: the estimated location of target $y_t$ of the $k$-th weak tracker in the $t$-th frame.
1)  Initialize the image patch $x_1$ and get CNNs feature $\mathbf{x}_1^k \in \mathbb{C}^{P\times Q\times D}$.
2)  Compute the dimensionality reduction features $\hat{\mathbf{x}}_1^{kr} \in \mathbb{C}^{P\times Q\times d}$ by Algorithm 2.
3)  Compute kernel $\hat{K}^{\mathbf{x}_1^{kr}\mathbf{x}_1^{kr}}$ of training samples $\hat{\mathbf{x}}_1^{kr}$ by Algorithm 1.
4)  Learn a filter $\hat{\alpha}_1^k$ by Eq. 4.
5)  for $t = 2 : N$
6)      Generate target candidate region $z_t$ and get CNNs feature $\mathbf{z}_t^k \in \mathbb{C}^{P\times Q\times D}$ according to the location of $y_{t-1}$.
7)      Compute the dimensionality reduction features $\hat{\mathbf{z}}_t^{kr} \in \mathbb{C}^{P\times Q\times d}$ by Algorithm 2.
8)      Compute $\hat{K}^{\mathbf{x}_{t-1}^{kr}\mathbf{z}_t^{kr}}$ by Algorithm 1.
9)      Compute a response map over the candidate regions by Eq. 5.
10)     Compute the estimated location by the candidate with the maximum filter response.
11)     Update the filter by Algorithm 3.
12) end
```

The final location estimation can be achieved by using weighted sum of the target positions of the weak trackers, which can be found in [10]. Meanwhile, the scale CF can be implemented after the location estimation, For more details about the scale estimation, refer to [27]. An overview of our overall architecture is given in Fig. 3.

V. EXPERIMENTAL RESULTS

The proposed algorithm is based on the *TCCF*, in which some modules are added. We denote *TCCF* with our graph based Random projection as $TCCF_{rp}$, and denote $TCCF_{rp}$ on the spherica manifolds as $TCCF_{rps}$.

The $TCCF_{rp}$ and $TCCF_{rps}$ is implemented in MATLAB 2016a with Caffe framework [37] and runs on an Intel(R) Core (TM) i7-8750H CPU @2.20GHz 2.21GHz CPU and a NVIDIA GeForce GTX 1070 GPU. According to the HDT, VGG-Net adopts very small convolutional filters (3Ã-3 pixel size), the feature maps from shallower layers have limited representation strength [23], therefore, for the VGG-16 [26], the deep part, including conv4-1, conv4-2, conv4-3 and conv5-1, conv5-2, conv5-3 convolutional layers, is used to extract features for location estimation [10], the dimension of the original deep features from each layer is 512, while, the combined features is 40 after random projection. The initial response weights of CNNs layer are 1, 0.5, 0.5, 1, 0.5 and 0.5, $J = 33$ and $\theta = 1.02$ in the scale estimation.

TABLE 2. The features and experimental setup of proposed algorithm and baseline trackers.

	Features	Scale
DCF	HOG	No
DCF_s	HOG	No
$TCCF$	Deep Features	Yes
$TCCF_{rp}$	Deep Features + Dimension Reduction	Yes
$TCCF_{rps}$	Deep Features + Dimension Reduction	Yes

In addition, To show more advantage in the fusion of the kernels, we also apply the the proposed fusion method to classical Dual Correlation Filter (DCF) [4], denoted as DCF_s. The proposed algorithm DCF_s is implemented in MATLAB 2016a and runs on the same CPU with the standard parameters provided by the authors.

We perform the experiments on OTB-50 [38] benchmark and compare with baseline trackers and the state-of-the-art methods. The features and experimental setup of proposed algorithm and baseline trackers are listed in Table. 2.

A. QUANTITATIVE ANALYSIS

Here we provide a quantitative comparison of our approach with the state-of-the-art trackers. Two criteria for evaluations of tracking performance are used on the OTB 50 benchmark: the precision plot and the success plot. The precision plot measures center location Euclidean distance between estimated position of the target and ground-truth, plotted over a range of thresholds. The success plot contains the percentage of frames over a range of overlap precision thresholds between estimated position box of the target and ground-truth box.

Overall performance evaluation: Fig. 4 show the results of precision plot and success plot, respectively. $TCCF_{rps}$ has performed the bast in both precision and success rate on OTB 50. As for the *TCCF* tracker, the proposed algorithm $TCCF_{rps}$ has the outperformance by 9.6% and 13.4%. The $TCCF_{rp}$ tracker also bring a gain in success and precision both compare with TCCF even using the features with less than one tenth of the dimensions of the original deep features.

As for the HOG features, DCF_s compared with DCF tracker, the above two performances are improved 2.6% and 1.4% respectively.

Attribute-based evaluation: We also compare three trackers for every attribute, namely illumination variation(IV), scale variation(SV), occlusion(OCC), deformation(DEF), motion blur(MB), fast motion(FM), in-plane rotation(IPR), out-of-plane rotation(OPR), out-of-view(OV), background clutters(BC) and low resolution(LR). The results are shown in Table. 3. Fig. 5 and 6 show the results of distance precision and success rate over eight attributes respectively.

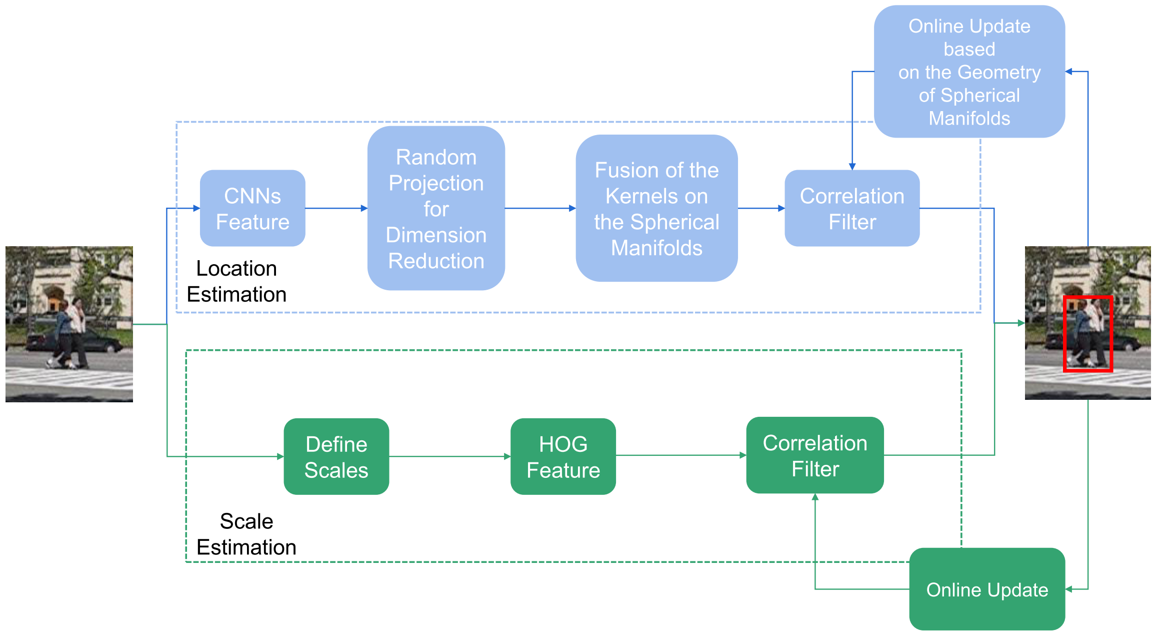

Online Update based on the Geometry of Spherical Manifolds
CNNs Feature
Random Projection for Dimension Reduction
Fusion of the Kernels on the Spherical Manifolds
Correlation Filter
Location Estimation
Define Scales
HOG Feature
Correlation Filter
Scale Estimation
Online Update

FIGURE 3. Full architecture of the tracking algorithm, which consists of three proposed sub-modules: random projection, fusion of the kernels and online update based on the geometry of spherical manifolds.

TABLE 3. The distance precision scores of our tracker and baseline trackers on different attributes.

	$TCCF$	$TCCF_{rp}$	$TCCF_{rps}$
IV	0.771	0.778	0.856
SV	0.798	0.815	0.852
OCC	0.729	0.789	0.869
DEF	0.727	0.800	0.885
MB	0.709	0.710	0.810
LR	0.718	0.689	0.764
FM	0.691	0.680	0.784
IPR	0.775	0.805	0.862
OPR	0.769	0.821	0.885
OV	0.529	0.515	0.758
BC	0.815	0.817	0.874

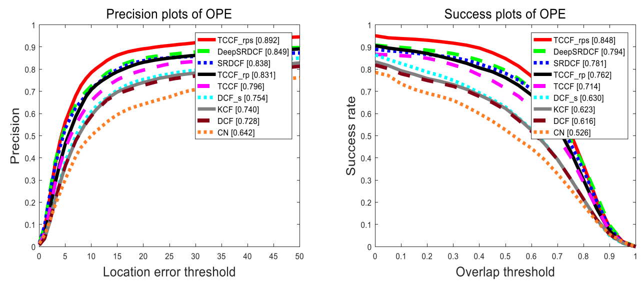


FIGURE 4. Comparison among the proposed algorithm, several baselines and state-of-the-art methods on success and precision.

Shown from the results of the attribute-based evaluation, for the trackers using CNNs feature, $TCCF_{rps}$ achieves the best performance among the all trackers in these 11 attributes. Compared DCF_s with DCF and KCF, the proposed DCF_s has a overall outperformance in the three trackers using HOG feature. It can be seen that the performance is indeed improved, as the difference in the energy of each channel feature is eliminated.

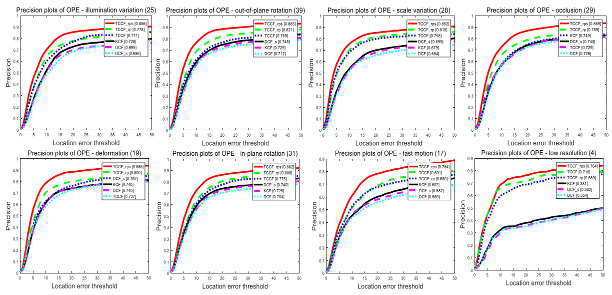


FIGURE 5. Attribute-based comparison with baseline trackers on precision.

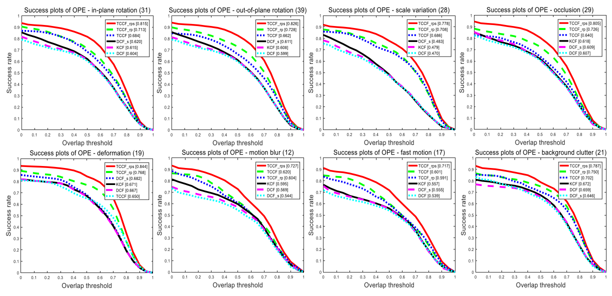


FIGURE 6. Attribute-based comparison with baseline trackers on success.

B. QUALITATIVE ANALYSIS

We present some tracking results in Fig. 7, where challenging frames among 51 image sequences are selected, including Ironman, Shaking, Jogging2, Lemming, Subway, Couple, MotorRolling and Coke. In most complex scenes, our algorithm can locate the target more accurately than baseline algorithms, which shows that it is helpful for the improvement on tracking to introduce the geometric structure of each kernel component.

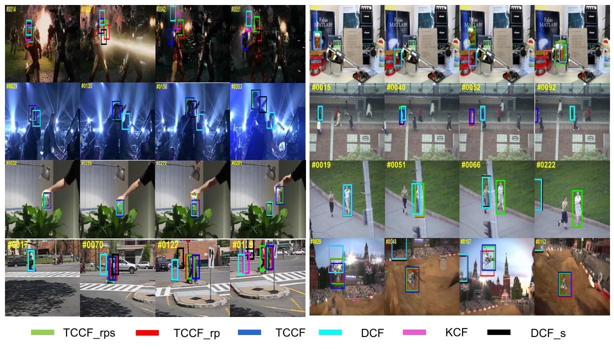


FIGURE 7. A visualization of the tracking results of proposed trackers and state-of-the-art visual trackers.

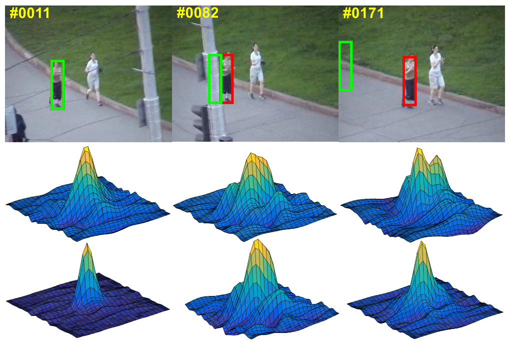


FIGURE 8. Tracking result and response maps of DCF_s (red box, third row) and DCF (green box, second row).

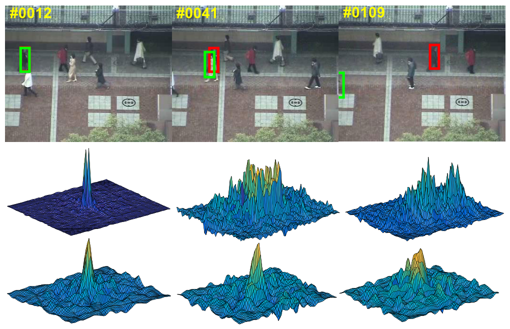


FIGURE 9. Tracker result and response maps of $TCCF_{rps}$ (red box, third row) and $TCCF$ (green box, second row).

For example of the trackers using HOG feature, Fig. 8 shows the results and response maps of DCF_s and DCF in the jogging video sequence. It can be seen that the pedestrian is occluded by the background. DCF_s can keep a robust tracking of the target in the video sequence, while DCF fail.

For example of the trackers using CNNs feature, video sequences subway characterized by OCC, BC and DEF, which is shown in Fig. 9, It can be also seen that $TCCF_{rps}$ can also perform tracking successfully.

It is well known that to estimate the target location more accurately, the response map need to be sharper at the peak. Shown from the Fig. 8 and Fig. 9, DCF_s and $TCCF_{rps}$ have less distraction in response map than DCF and TCCF due to the geometry constraint of spherical manifolds.

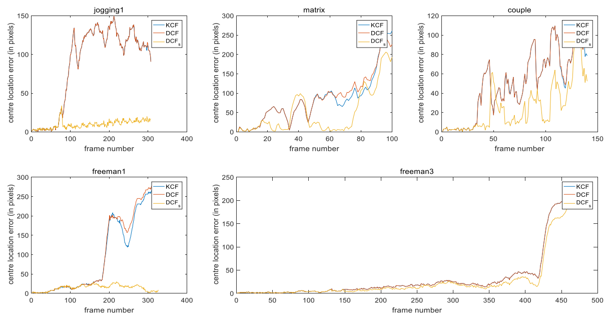


FIGURE 10. A frame-by-frame comparison of center location error with DCF_s and baseline trackers on example sequences.

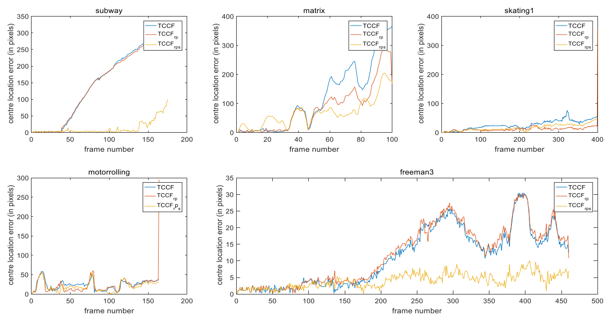


FIGURE 11. A frame-by-frame comparison of center location error with $TCCF_{rps}$ and baseline trackers on example sequences.

TABLE 4. Tracking speed of the different algorithms with hand-crafted features.

with hand-crafted features	DCF	DCF_s	CN
fps	280	150	105

TABLE 5. Tracking speed of the different algorithms with CNN features.

with CNN features	$TCCF$	$TCCF_{rp}$	$TCCF_{rps}$	$DeepSRDCF$
fps	2.5	5	2	2

In addition, a frame-by-frame comparison of our proposed algorithm on example sequences is presented, showing the center location error in pixels. The results came from the trackers using HOG feature and CNNs feature are presented in Fig. 10 and Fig. 11, respectively. It can be seen that our tracker can still identify the target stably when approaching the end of the video sequence.

We also compare the mean frames per second (FPS) of each method, the result is shown in Table. 4 and Table. 5. It can be seen from Table. 4 that despite some decline in frames per second, our tracker still operates beyond real-time. Meanwhile, it also demonstrates from Table. 5 that dimensional reduction by random projection is essential to enhance the computational efficiency.

VI. CONCLUSION

In this article, we propose a novel correlation filtering tracking algorithm based on spherical manifold geometry. By embedding each kernel on the spherical manifold, we have a substitution of the mean in a hypersphere for the mean in the normal Euclidean space, by which a fusion of multiple kernels is achieved. To maintain consistency of our approach, a corresponding online update strategy is also proposed based on the geometry of spherical manifolds. The experiments show that fusion kernels with the normalized amplitude through the geometry of spherical manifolds does have a contribution to dealing with these challenges more effectively. It has been proved that our approach is generic and can be extended to many CF based tracking methods with multiple channels features.

REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, p. 13, 2006.
[2] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1442–1468, Jul. 2014.
[3] T. Vojir, J. Noskova, and J. Matas, "Robust scale-adaptive mean-shift for tracking," *Pattern Recognit. Lett.*, vol. 49, pp. 250–258, Nov. 2014.
[4] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.
[5] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S. L. Hicks, and P. H. S. Torr, "Struck: Structured output tracking with kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2096–2109, Oct. 2016.
[6] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–577, May 2003.
[7] M. Danelljan, G. Hger, and F. S. Khan, "Learning spatially regularized correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2018, pp. 4904–4913.
[8] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 702–715.
[9] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Dec. 2015, pp. 621–629.
[10] Q. Liu, B. Liu, and N. Yu, "TCCF: Tracking based on convolutional neural network and correlation filters," in *Proc. Int. Conf. Image Graph.*, vol. 10666, Dec. 2017, pp. 316–327.
[11] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "Eco: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 6931–6939.
[12] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional Siamese networks for object tracking," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2016, pp. 850–865.
[13] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
[14] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2005, pp. 886–893.
[15] D. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2544–2550.
[16] M. Danelljan, F. S. Khan, M. Felsberg, and J. Van De Weijer, "Adaptive color attributes for real-time visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1090–1097.
[17] R. Tao, E. Gavves, and A. W. M. Smeulders, "Siamese instance search for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1420–1429.
[18] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 FPS with deep regression networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 749–765.
[19] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2805–2813.
[20] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Proc. Eur. Conf. Comput. Vis.(ECCV)*, Oct. 2016, pp. 472–488.
[21] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3074–3082.
[22] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Robust visual tracking via hierarchical convolutional features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 11, pp. 2709–2723, Nov. 2019.
[23] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M.-H. Yang, "Hedged deep tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4303–4311.
[24] Y. Qi, S. Zhang, L. Qin, Q. Huang, H. Yao, J. Lim, and M.-H. Yang, "Hedging deep features for visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 5, pp. 1116–1130, May 2019.
[25] P. Li, D. Wang, L. Wang, and H. Lu, "Deep visual tracking: Review and experimental comparison," *Pattern Recognit.*, vol. 76, pp. 323–338, Apr. 2018.
[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556
[27] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf.*, Sep. 2014, pp. 1–11.
[28] D. V. Arnold, "On the use of evolution strategies for optimization on spherical manifolds," in *Proc. Int. Conf. Parallel Problem Solving Nature*, vol. 8672, 2014, pp. 882–891.
[29] H. Karcher, "Riemannian center of mass and mollifier smoothing," *Commun. Pure Appl. Math.*, vol. 30, no. 5, pp. 509–541, Sep. 1977.
[30] R. C. Wilson and E. R. Hancock, "Spherical embedding and classification," in *Proc. Joint IAPR Int. Conf. Structural*, vol. 6218, 2010, pp. 589–599.
[31] P. T. Fletcher, C. Lu, S. M. Pizer, and S. Joshi, "Principal geodesic analysis for the study of nonlinear statistics of shape," *IEEE Trans. Med. Imag.*, vol. 23, no. 8, pp. 995–1005, Aug. 2004.
[32] T. Takiguchi, J. Bilmes, M. Yoshii, and Y. Ariki, "Evaluation of random-projection-based feature combination on speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2010, pp. 2150–2153.
[33] Q. Du and J. E. Fow, "On the performance of random-projection-based dimensionality reduction for endmember extraction," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2010, pp. 1277–1280.
[34] J. Wang, *Geometric Structure of High-Dimensional Data and Dimensionality Reduction*. Berlin, Germany: Springer, 2012.
[35] V. Menon, Q. Du, and J. E. Fowler, "Hadamard-Walsh random projection for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 9, Jul. 2016, pp. 1275–1279.
[36] A. E. Brouwer and W. H. Haemers, *Spectra of Graphs*. New York, NY, USA: Springer, 2012.
[37] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 675–678.
[38] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.

MINGKE ZHANG received the B.S. degree in electronic information science and technology from the Xi'an University of Posts and Telecommunications, Xi'an, Shaanxi, China, in 2017. She is currently pursuing the M.S. degree in software engineering with the Shaanxi University of Science and Technology, China. Her research area is visual tracking.

XUANDE ZHANG was born in Ningxia, China, in 1979. He received the B.S. degree in computational mathematics from Ningxia University, Ningxia, in 2000, and the M.S. and Ph.D. degrees in applied mathematics from Xidian University, Xi'an, China, in 2006 and 2013, respectively. He is currently a Professor of electrical and information engineering with the Shaanxi University of Science and Technology. His research interests include numerical analysis, machine learning, and its application in computer vision.

LONG XU received the M.S. degree in applied mathematics from Xidian University, Xi'an, China, in 2002, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. He held a postdoctoral position at the Department of Computer Science, City University of Hong Kong, and the Department of Electronic Engineering, The Chinese University of Hong Kong, from 2009 to 2012. From 2013 to 2014, he held a postdoctoral position at the School of Computer Engineering, Nanyang Technological University, Singapore. He is currently with the Key Laboratory of Solar Activity, National Astronomical Observatories, Chinese Academy of Sciences. His research interests include image/video processing, solar radio astronomy, wavelet, machine learning, and computer vision.