Learning Scale-Adaptive Tight Correlation Filter for Object Tracking

Shunli Zhang®, Wei Lu, Weiwei Xing, and Li Zhang

Abstract—In this paper, we propose a novel tracking method by formulating tracking as a correlation filtering as well as a ridge regression problem. First, we develop a tight correlation filter-based tracking framework from the signal detection perspective. In this formulation, the correlation filter is set as the same size as the target, which can make full use of the relations of the adjacent image patches and effectively exclude the influence of the background. Specifically, we point out that the novel correlation filter model can be regarded as the ridge regression model which takes into account the different importance of the samples and has the consistent objective with tracking. Second, we focus on the scale variation problem in tracking. By making use of the spatial structure of the correlation filter, the multiscale filter banks can be generated via interpolation to handle the scale estimation problem easily. Third, we present a novel distance importance-based confidence calculation model to determine the final tracking result, which not only makes use of the fine discriminability of the correlation filter but also takes the distance importance of the candidate samples into account to alleviate the impact of similar distractors. Experimental results demonstrate that our method is superior to several state-of-the-art trackers and many other correlation filter-based methods in the benchmark datasets.

Index Terms—Correlation filter, object tracking, ridge regression, scale adaptation.

I. INTRODUCTION

OBJECT tracking is a hot topic in computer vision. Due to its wide applications in motion analysis, video surveillance, human–computer interface, etc., it has attracted much attention recently. However, because of various complex factors, for example, occlusion, deformation, illumination variation, scale variation, fast motion, and background clutter, tracking is still a challenging task [1]–[3].

A typical tracking system often has three modules, that is, the appearance model, the motion model, and the update model. The appearance model, which plays a crucial role, can be divided into two types. The first is the generative model [4]–[9], which only uses the information of the target. The second is the discriminative model [10]–[13], which often formulates tracking as a binary classification problem and is known as the tracking-by-detection method. Since the latter model adopts the information of both the background and the target, it seems to offer more accurate and robust representation.

However, there are still some potential issues in existing tracking-by-detection methods. First, it is pointed out that the objectives of the tracker and the classifier are not consistent in many methods, where the objective of the tracker is to find the optimal location of the target precisely while the objective of the classifier is to predict the labels of the samples accurately [14]. Second, the training samples are often undersampled randomly and assigned to equal weights. This sampling strategy does not make full use of the spatial constraints around the target and ignores the different importance of the samples [15]. Besides the above issues, we also find that the high spatial correlations of adjacent image patches do not get enough attention, which increases the complexity of sampling, training, and detection. Furthermore, the size of the tracker is often fixed and the inner structure of the tracker is often neglected, so that the scale variation is not well addressed. These issues may be important factors affecting the tracking performance. Although some of these issues have been pointed out by some researchers, they have not been taken into account in a unified framework.

Recently, many correlation filter-based tracking methods have been proposed and achieved great success. These methods formulate tracking as a ridge regression problem, which can make full use of the spatial information and solve the downsampling issue in the binary classification model. Besides, the correlation filter-based methods transfer the ridge regression to a correlation filter problem by the circulant structure assumption, which can be realized by fast Fourier transformation (FFT) with high efficiency. Although these methods obtain both high tracking accuracy and fast tracking speed, there still exist some issues which degrade the performance of the trackers. For example, the correlation filter in [15]–[18] is obtained based on the circulant structure assumption, where the samples are represented by rectangle boxes larger than the true target and are implicitly generated by cyclic shift of the original target. As Fig. 1(a) shows, the samples with
horizontal translation $dx = 20$, $dx = 40$, and $dx = 60$ pixels are obtained by cyclic shift of the original sample with $dx = 0$. However, it should be noted that these samples do not fit the real scene, as the sample with $dx = 60$ pixels displays. In addition, in order to make the samples satisfy the circulant structure assumption, a bounding box used to represent the sample should be much larger than the target and a Hanning window must be utilized to deal with the discontinuity of the image margin, which leads to two additional problems. As Fig. 1(a) displays, the large bounding box contains much more background information which may not represent the target appearance accurately and brings more challenges to scale adaption and robustness to background changes. Besides, the Hanning window also changes the appearance of the training patch, adding the representation errors, as Fig. 2 illustrates.

Fig. 1. Issue of the generated training samples for learning the correlation filters in methods [15]–[18]. Note that the samples in (a) are virtual while the samples in (b) are real.

Fig. 2. Issue of the training patch in traditional correlation filter-based methods. Note that the training patch changes the information far away from the center in the traditional correlation filter (a), while it retains the original information of the image in our formulation (b).

In this paper, we propose a novel tracking method by learning a scale-adaptive tight correlation filter, attempting to address the above issues in tradition tracking-by-detection methods and in conventional correlation filter-based methods. First, we formulate tracking as a correlation filtering problem from the signal detection perspective and learn a tight correlation filter as the tracker, which does not need the prior assumption existing in other correlation filter-based methods. By taking the target as the signal of interest to detect, the objective of filtering is to find the optimal response corresponding to the target, which is consistent with the objective of tracking. Since the filter is set as the same size as the target, it is tight and can effectively reduce the effect of the background, as Fig. 1(b) shows. The samples corresponding to the correlation filter are real rather than virtual, which improves the representation accuracy. In addition, we establish the relationship between the proposed correlation filtering formulation and the ridge regression model from the practical physics background, which enriches the tracking-by-detection framework. Different from the binary classification formulation which assigns discrete labels to samples, in our method, we would like to build a ridge regression model to assign continuous values to samples. Specifically, we point out that the correlation filter is a specific realization of the ridge regression, and the correlation filter corresponds to the appearance model of the tracking-by-detection framework.

Second, we present a multiscale strategy to handle the scale estimation problem adaptively by using the spatial structure of the correlation filter. In our method, the learned correlation filter has the same size with the target. Since the filter retains the spatial structure in the 2-D plane, the multiscale filter banks can be approximately implemented based on the interpolation technique, which can be used for multiscale filtering. Then, the target scale can be estimated adaptively by exploring the optimal response on multiple scales heuristically. The experimental results in the benchmark datasets demonstrate that our method can achieve desirable tracking performance and outperform many state-of-the-art trackers, especially in the condition of scale variation.

Third, we develop a novel location method in the motion model by making use of both the correlation filter and the distance importance weight of the candidate samples. In most traditional tracking-by-detection and correlation filter-based methods, the confidence score is obtained by only the appearance model, for example, the classifier or the correlation filter, and the candidate samples are assigned to the same importance. However, in this condition, the similar distractor, which is far away from the target, may cause the tracking drift. In this paper, we consider the impact of the distance corresponding to each candidate sample as well as the confidence obtained by the correlation filter. In other words, the final confidence map is the combination of the distance importance score and the filtering result, which can effectively reduce the impact of the similar distractors.

The main contributions of this paper are three-fold.

1) We formulate tracking as a tight correlation filtering problem instead of the circulant structural assumption-based correlation filtering. In this formulation, the size of the correlation filter is set the same as the target, which can effectively avoid the effect of the virtual samples and reduce the influence of the background information. Specifically, the zero padding and cropping strategy are introduced to efficiently realize the learning of the filter and tracking in the Fourier domain, which does not need the circulant assumption and can effectively address the boundary effect issue.

2) We approximately construct the multiscale filter banks for scale adaption based on the interpolation strategy and dense feature extraction. Since the multiscale filters are directly built in the filter domain, our method may locate the target more precisely than many other scale
adaptive methods which realize scale adaption by scaling the images.

3) We present a novel location strategy to determine the tracking result which considers both the filtering result and the distance importance of the candidate samples. On one hand, the filtering results obtained by the multiscale filter banks reflect the basic location information from the appearance model. On the other hand, the distance importance of the candidate samples is taken into account, which can effectively alleviate the impact of the similar distractors.

The remainder of this paper is organized as follows. In Section II, we review the tracking-by-detection methods, and talk about the correlation filter-based tracking methods. Section III formulates tracking as a correlation filtering problem and explains it from the perspective of ridge regression. The detailed realization of the proposed method is shown in Section IV and the experimental results are displayed in Section V. Section VI concludes this paper.

II. RELATED WORK

A. Tracking-by-Detection

Tracking-by-detection methods often formulate tracking as a binary classification problem [10], [19]–[24]. Thus, different classifiers, for example, support vector machine and boosting, have been applied to build the trackers. However, there are some potential issues in most of the above methods, for example, the neglect of the different importance of the samples, the inconsistency of the objectives, and the undersampling strategy, which may degrade tracking performance. To address the above issues, Babenko et al. [11] introduced the online multiple-instance learning into tracking. They replace the positive and negative samples by the positive and negative bags to train the classifier, alleviating the drift caused by the classification inaccuracies and sample ambiguity to some degree. Hare et al. [14] applied the structured output learning in tracking to address the inconsistency of the objectives of tracking and classification. In their method, the importance of the samples is measured implicitly. Possegger et al. [25] presented a discriminative object model capable of differentiating the object of interest from the background. This method relies on standard color histograms and is used to identify and suppress distracting regions in advance. Different from the above formulations, in this paper, the correlation filter can be considered as a novel ridge regression model, which can measure the importance of the samples explicitly and adopt a dense sampling strategy to make use of the spatial constraints. Besides, the objectives of the regression and tracking are consistent, providing a model with finer discriminability.

Some other strategies which aim to build accurate appearance models have also been introduced to improve tracking performance. Wang et al. [26] proposed the multicue-based tracking method, which incorporates optical flow, color, and depth clues simultaneously. However, this method is based on the assumption that different features can provide complementary supporting information, which still belongs to the generative model. Fang et al. [27] presented the part-based online tracking method with geometry constraints and attention selection, which utilizes a part-based structure to construct an adaptive appearance model together with the attentional sample weighting and a two-stage motion model. Fang et al. [28] proposed an online tracking method by exploiting the spatiotemporal saliency for template sampling. By making use of the hash code learning, the useful relationship between the positive and negative templates can be preserved and matching can be efficiently conducted. The latter two methods can be considered as the discriminative models, which utilize the positive and negative samples-based classification models. In our method, although we mainly adopt the histogram of the oriented gradients (HOGS) feature for representation, the correlation filtering formulation can better use spatial information by dense sampling, which can build a more accurate discriminative appearance model.

B. Correlation Filter-Based Tracking

The correlation filter has achieved success in many domains [16], [29], [30]. Recently, some tracking methods based on the correlation filter have been proposed. Bolme et al. [31] proposed a minimum output sum of a squared error filter for tracking, which first models the appearance adaptively by learning the correlation filter but does not make full use of the spatial constraints. Henriques et al. [15] developed the circulant structure of tracking-by-detection with kernels tracker (CSK) by exploiting the circulant structure of the local image patch and learning a kernelized least squares classifier. Note that the size of the learned filter is much larger than the target due to the circulant structure assumption. They further propose the kernelized correlation filter tracker (KCF) [16] by utilizing the HOG feature instead of the intensity, which significantly improves CSK. Danelljan et al. [32] developed the adaptive color attributes tracker (CN) by adding the color attribute to augment the feature used by CSK. Recently, some researchers [33]–[36] propose new tracking methods by combining both the deep convolutional neural networks and correlation filter, which can further improve tracking performance. However, most of these methods are based on the circulant structure assumption, and use a rectangle with much larger size to represent the object, that is, the filters are not tight. Although we also employ the concept of correlation filter, we stress that our formulation is different from the above methods. We argue that the circulant structure assumption in those methods is not often satisfied, especially in the condition of complex backgrounds. Moreover, when the target moves, the changes of its surrounding background may be larger than itself, making the assumption invalid. In our method, we model tracking from the signal detection viewpoint, which does not need the above assumption. We build a new correlation filter framework by setting the filter as the same size as the target, which is tight and can exclude the effect of the background as far as possible.

Some researchers also extend the KCF method to the scale-adaptive versions by introducing the multiscale strategy. Danelljan et al. [17] presented discriminative scale space
tracking (DSST) and attempt to address the scale variation under the correlation filter framework by learning separate filters for translation and scale estimation, respectively, but the scale estimation depends on the accuracy of the position. Li and Zhu [18] proposed scale adaptive with a multiple features tracking approach (SAMF) which takes the color-naming feature to improve the representation and employs bilinear interpolation to generate image representations in multiple scales. Both of these two methods realize the scale adaption by interpolation in the image domain and extract features many times. Further, Zhang et al. [37] presented the fast detection proposal method into a correlation filter tracker and uses the EdgeBoxes to realize the scale and aspect ratio adaption. Different from the above methods, the proposed scale adaption in our method directly works in the filter domain, which makes use of the inner structure of the filter. Besides, because the image region is not scaled, our method may provide a more precise location.

III. FORMULATION

A. Correlation Filtering Formulation

Different from most of the traditional correlation filter-based tracking methods which are obtained from the regression formulation in the mathematical domain, we formulate tracking as a correlation filtering problem from the perspective of signal detection which has a practical physics background. By taking the image patch as an input signal and the tracker as a filter, tracking can be modeled as a correlation filtering process to detect the signal of interest, where the largest response value corresponds to the target. To make the objectives of tracking and filtering consistent, it is necessary to design a specific response function, whose value is largest in the target position and becomes smaller with the increasing distance to the target. In practice, the correlation filter can be learned based on the known image patch and its response, and applied for filtering in the testing image.

As Fig. 3 illustrates, the proposed tracking framework based on correlation filtering includes two components, that is, learning the filter and tracking by filtering. In the learning stage [Fig. 3(a)], denote the training region as $f$ which centers at the target, and the response function is $g$. Hereby, the Gaussian function is regarded as the response function. The correlation filter $h$ is learned according to $f$ and $g$. Note that the $h$ is tight and its size is $v \times w$, which is the same as the target, and the size of $f$ is $V \times W$. Based on the minimum output squared error criterion, the objective function of learning the required correlation filter is

$$\min_{h} \|f \otimes h - g\|^2 + \lambda \|h\|^2$$

where $\otimes$ denotes the correlation operation, and $\lambda$ is the tradeoff parameter.

$$\hat{g} = \mathcal{F}^{-1}(\mathcal{F}(f) \otimes \hat{h}^*)$$

where $\mathcal{F}$ denotes the FFT, $\mathcal{F}^{-1}$ denotes the inverse FFT (IFFT), and $\hat{h}$ is the optimal tracking result.

However, the correlation operation in (1) makes it hard to optimize the objective function directly. Therefore, we introduce the zero padding strategy to turn the linear convolution into the circulant convolution. According to the Parseval’s identity [39], the circulant convolution can be realized by FFT efficiently. After padding zeros in $h$ and $f$, we can get the augmented $\hat{h}$ and $\hat{f}$. Then, (1) can be rewritten in the Fourier domain

$$\min_{\hat{h}} \|\hat{f} \otimes \hat{h}^* - \hat{g}\|^2 + \lambda \|\hat{h}\|^2$$

B. Ridge Regression Explanation

We formulate tracking as a ridge regression problem from the tracking-by-detection perspective and establish the relationship between the novel ridge regression formulation and
the above correlation filtering model. Different from the binary classification using the random undersampling manner, hereby, we utilize a dense sampling strategy. Assume that the position of the target in a frame is \( p_{s0} \) and \( p_{o0} \), the position of the \( i \)th sample is \( p_{si} \) and \( p_{oi} \), the size of the target is \( v \times w \), and the size of the sampling region is \( V \times W \). The training samples will be cropped by sliding the sampling window around the target, following \( \|p_{si} - p_{s0}\| < (W - w)/2 \) and \( \|p_{oi} - p_{o0}\| < (V - v)/2 \).

As Fig. 4 shows, the importance of the samples is different because they have different distances to the target. It is natural to assign a higher value to the sample which is closer to the target, following the vectorizing operation. By (8) and (9), if we set \( w = \text{vec}(h) \), it can be observed that \( y_i = g_i \). Therefore, the correlation filtering is a specific realization of the ridge regression model and accords with the tracking-by-detection framework. The relationship of these two formulations is also retained when multichannel features are used.

Specifically, two benefits can be acquired from the correlation filtering formulation. On one hand, correlation filtering can greatly decrease computational complexity. The complexity of sampling and matrix inversion in ridge regression is \( O((V-v)(W-w)vw) \) and \( O(D^3) \), respectively, where \( D \) is the dimension size of the feature of a sample. However, for the filtering formulation, the complexity of FFT and IFFT is both \( O(L_vL_w \log(L_vL_w)) \). In practice, because the relations of the adjacent image patches are not fully exploited in regression, there exist many redundant operations in sample selection and feature extraction. On the other hand, the correlation filter works in the 2-D plane with a spatial structure. Thus, it can be used to deal with the scale variation problem by scaling itself adaptively.

### IV. REALIZATION

Based on the above correlation filtering formulation, hereby, we introduce how to complete the tracking process in detail. The complete tracking framework is shown in Fig. 5. First, we learn the practical correlation filter by introducing several techniques. Specifically, we introduce the zero padding and cropping strategy to make the fast realization by FFT available. Multichannel features are extracted to improve the discriminability, and the multiscale filter banks are built based on the interpolation strategy to realize scale adaption. Next, the learned multichannel multiscale filters are used to track the target in the next frame, where the optimal position and scale are determined by the distance importance-based model. Further, based on the tracking result in the current frame, a new group of filters is built and used to update the system filter banks. At last, the filtering and update steps are iterated to complete the tracking.

#### A. LEARNING THE CORRELATION FILTER

1) Implementation With Zero Padding and Cropping: As we have mentioned in Section III-A, the first step is to
construct the tight correlation filter. We employ a zero padding strategy to turn the correlation into circulant convolution which can be efficiently solved.

The reasons why the zero padding and cropping strategy are adopted to solve the optimization problem in (1) are as follows. First, the zero padding and cropping strategy is consistent with the tight correlation filter and is able to help to realize the fast realization by FFT. In order to address the issue of virtual samples in the traditional correlation filter, we have presented the tight correlation filter method to make all of the training samples be selected from the real image region. However, if we still use the circular assumption by adding a Hanning window to the image region instead of the zero padding and cropping strategy to conduct FFT, the samples near the margin of the image region are heavily affected by the Hanning window and they will decrease the accuracy of filter learning, as Fig. 2 displays. Second, the zero padding and cropping strategy can ensure the feasibility and accuracy of the fast realization based on FFT. According to the convolutional theorem, the circulant convolution can be fast realized by FFT. However, correlation filtering corresponds to linear convolution which cannot be directly realized by FFT, because the image region and the filter do not have the same size, and the dot product in the Fourier domain cannot be conducted. One effective way to address this issue is to convert the linear convolution to circulant convolution by the zero padding strategy. By adding zeros in the margin of both image region and the filter to make them have the same size, the FFT operation can be executed. Based on the zero padding strategy, linear convolution is equal to the circulant convolution [39]. In addition, the zero padding and cropping strategy can effectively avoid the aliasing problem in circulant convolution without zero padding.

The details of the zero padding and cropping strategy are shown in Fig. 6. As pointed out before, correlation can be considered as linear convolution, which is equal to circulant convolution after padding zeros in the margins of both the image patch and the filter to make them have the same size [39]. According to the Parsevals identity [39], the circulant convolution can be realized by FFT efficiently. As Fig. 6(a) shows, the training region \( f \) is padded by zeros in its edge, generating the augmented \( \tilde{f} \) with size \( L_v \times L_w \), where \( L_v = V + v - 1 \) and \( L_w = W + w - 1 \). Then, \( \tilde{f} \) can be transformed into the Fourier domain successfully. The response map \( g \) can be easily extended to the augmented version because the values out of the margin of \( g \) are close to zero based on the Gaussian function. Fig. 6(b) shows how to obtain the final correlation filter. After \( \tilde{h} \) is achieved by (3), we can obtain the augmented \( \tilde{h} \) by IFFT, in which the aliasing effect can be observed. In practice, the standard filter can be obtained by cropping out the up-left corner region of \( \tilde{h} \). The cropping manner is simple but effective and has been successfully used in the image deblurring domain [40]. Thus, we utilize the cropping strategy to avoid complex computation.

Two benefits can be brought by learning in the Fourier domain with the introduced zero padding and cropping strategy and without exploiting the circulant structure. First, the samples corresponding to the zero padding strategy for FFT can be set to be the same size as the target, which can avoid the effect of the background. Second, employing the zero padding strategy for FFT can effectively address the unwanted boundary effects caused by the circulant structure assumption. Because the boundary effects may lead to inaccurate representation and reduce the useful region, using zero padding to alleviate these effects may improve the accuracy of the tracking results.

2) Multichannel Filter: In the aforementioned formulation, we introduce the correlation filtering framework with only a single-channel feature. To improve the representation capability, we would like to extend the single-channel feature to multichannel, for example, HOG, generating several layers of feature planes. Assume the dimension size of the feature at each point is \( d \), then the feature plane in the \( l \)th layer can be denoted as \( f^l \), where \( l \in \{1, 2, \ldots, d\} \). Correspondingly, as Fig. 7 shows, the multichannel filter banks can be learned according to the multichannel features, similar to [29] and [41]. The objective function in (1) becomes

\[
\min_{\mathbf{h}} \left\| \sum_{l=1}^{d} f^l \otimes h^l - g \right\|^2 + \lambda \sum_{l=1}^{d} \left\| h^l \right\|^2 \tag{10}
\]

where \( h^l \) is the filter in the \( l \)th layer of \( \mathbf{h} \). Similar to solving (1) and (2), based on the augmented \( \tilde{f} \) by padding zeros, Parsevals identity, and FFT, the objective function can be rewritten in the Fourier domain

\[
\min_{\mathbf{h}} \left\| \sum_{l=1}^{d} \tilde{f}^l \otimes \tilde{h}^{l*} - \tilde{g} \right\|^2 + \lambda \sum_{l=1}^{d} \left\| \tilde{h}^{l*} \right\|^2 \tag{11}
\]
can be approximately constructed based on the interpolation structure of the correlation filter, the multiscale filters in multiple scales and use a heuristic method to determine the scale factor with $s$. Then, we can complete the tracking in the current frame. The size of the filter bank in scale $s$ is $a^sv \times a^sw$. Therefore, the bicubic interpolation is utilized to construct the filter banks in different scales. After the scale augmentation, the final filter banks can be denoted as $\{h_l^t\}$ with $s \in \{-S, \ldots, S\}$ and $l \in \{1, \ldots, d\}$. In practice, the system filter banks used for tracking are acquired based on the frame-by-frame update and interpolation, which is shown in Section IV-C.

### B. Tracking and Location

Based on the constructed correlation filter, we conduct tracking and propose a distance importance-based confidence map to determine the location of the target, which not only makes use of the filtering result but also takes the distance importance of the candidate samples into account.

Denote the filter banks from frames $\{1, \ldots, t - 1\}$ as $\{H_{l-1}^t\}$. Then, we can complete the tracking in the current frame $t$. First, we crop the image region in frame $t$, which centers in the position of the target of frame $t - 1$, $p_{t-1}$, and with width $W$ and height $V$. In practice, the cropping operation corresponds to sampling the candidate samples in the regression model. Second, dense multichannel features are extracted from the candidate region and a series of feature planes $\{f^l\}$ is obtained. Third, $\{f^l\}$ and $\{H_{l-1}^t\}$ are extended to $\{f^l\}$ and $\{H_{l-1}^t\}$ by padding zeros. Fourth, we apply the filtering operation by employing $\{H_{l-1}^t\}$ in $\{f^l\}$ separately for $s \in \{-S, \ldots, S\}$ in each layer. Then, the response map in scale $s$ is obtained by

$$g'(s) = \sum_{l=1}^{d} F^{-1}(F(f^l) \odot F(H_{l-1}^{l-1} s)) a^{2s}$$

where $a^{2s}$ is used to normalize the response values in different scales.

As we have pointed out above, each point in the confidence map corresponds to a unique candidate sample. It should be noted that the importance of the candidate samples is different. It is assumed that the target cannot move too far in the consecutive frames, so that the sample closer to the tracking result in the last frame should be assigned to a larger importance weight while the one which is farther away should have a smaller weight. Therefore, we further calculate the importance weight map $d$ of candidate samples by the Gaussian function

$$d(p) = \exp \left(\frac{\|p - p_{t-1}\|^2}{\sigma^2_{\lambda}}\right)$$

where $p$ denotes the position of the candidate sample, $p_{t-1}$ is the position of the tracking result in the last frame, $d(p)$ is the weight element of $d$ corresponding to the position $p$, and $\sigma^2_{\lambda}$ is the weight parameter. By embedding the weight map into the confidence map in (13), we can obtain the final confidence map

$$g''(s) = g'(s) \odot d.$$  

Because the response map $g''$ is cubic based on a position vector $p$ and a scale variable $s$, we search for the maximum response value over the whole position and scale space to locate the target

$$p_{t}, s_{\text{opt}} = \arg \max_{(p,s)} g''$$

where $p_{t}$ and $s_{\text{opt}}$ represent the predicted optimal position and scale of the target, respectively, and $\{(p,s)\}$ represents the candidate position and scale space.

### C. Update

To cope with the appearance changes of the target during tracking, the correlation filter banks should be updated once the tracking result is obtained. Therefore, we adopt an incremental update strategy. Assume the multichannel filter banks in frame $t - 1 (t > 1)$ are $[H_{l-1}^{l}]$, where $H_{l-1}^{l0} = h_l^t$ and $H_{l-1}^{l}(s \neq 0)$ is obtained by interpolation. Once the tracking result in frame $t$ is obtained, we can learn a new correlation filter bank $[h_l^t]$ in frame $t$ with the new tracking result according to the determined position and the scale of the target, and then utilize it to update $[H_{l-1}^{l}]$. Since $h_l^t$ is learned in the optimal scale in the current frame, we resize $[H_{l-1}^{l}]$ to make it fit the same scale first. Assuming that the learning rate is $\theta$, the updated filter bank $[H_{l-1}^{l}]$ can be obtained by the linear combination of $[H_{l-1}^{l}]$ and $[h_l^t]$

$$H_{l-1}^{l0} = (1 - \theta)H_{l-1}^{l0} + \theta h_l^t, \quad l = 1, \ldots, d.$$  

Then, the filter banks $\{H_{l}^{l0}\}$ in other scales can be obtained by interpolation as well, used for filtering and detection in frame $t + 1$. The iteration process is summarized in Algorithm 1.
Algorithm 1 SCFR Tracking: Iteration in Frame t

Input:
Frame \( I_t \); Previous object position \( p_{t-1} \) and size \( v \times w \); Filter banks \( \{H^j_{t-1}\} \).

Output:
Object position \( p_t \) and updated size \( v \times w \); Updated filter banks \( \{H^j_t\} \).

1: Filtering and Tracking.
(1) Crop the candidate image patch at \( p_{t-1} \) from \( I_t \) and extract the multichannel features \( \{f^j\} \);
(2) Augment \( \{f^j\} \) and \( \{H^j_{t-1}\} \) by padding zeros;
(3) Calculate the distance weighed response \( g^j(s) \) by Eqn. 13, Eqn. 14 and Eqn. 15;
(4) Determine the optimal position \( p_t \) and scale \( s_{opt} \) by Eqn. 16;
(5) Determine the object size by \( a^{p_{opt}}v \times a^{s_{opt}}w \).

2: Learning and Update.
(1) Crop the training patch at \( p_t \) from \( I_t \) and extract and augment the multichannel features \( \{f^j\} \);
(2) Learn the filter bank \( \{h^j_t\} \) in \( I_t \) by Eqn. 12;
(3) Resize the filter bank \( \{H^j_{t-1}\} \) to scale \( s_{opt} \), and updated \( \{H^j_t\} \) by Eqn. 17;
(4) Generate \( \{H^j_t\} \) in other scales \( (s \neq 0) \) by interpolation.

V. EXPERIMENTS

A. Experimental Setup
The configuration of our tracker is as follows. Since the HOG feature [42] with 5-pixel window size and 9 orientations obtains excellent performance on object detection and tracking in many methods [16], [43], [44], it is adopted as the multichannel feature and is densely sampled in our method. The sizes of the cropped regions for training and detection are experimentally set to 3 times the target size and adaptively adjusted, since a larger region size will significantly increase the computation complexity while a smaller size may get better scale adaption but will greatly increase the computation complexity and will make the tracker less robust to similar distractors while a smaller \( \sigma^2_d \) will decrease the accuracy of the filtering result. All parameters are fixed on the experiments.

Four criteria are used to evaluate the performance of the tracker. The first is the average center location error (CLE), which is defined as the average of the errors of the obtained target center position and the ground truth. The second is precision, which is computed by the ratio of the number of frames whose CLE is smaller than a threshold \( \text{Th}_{\text{p}} \) and the number of total frames. The third is the average Pascal VOC overlap rate (VOR). VOR is defined as the average of Score = ([area(\( R_S \cap R_G \))]) / ([area(\( R_S \cup R_G \)))], where \( R_S \) and \( R_G \) represent the rectangles of the tracking result and ground truth in one frame, respectively [45]. The fourth is success rate (SR). If the VOR in a frame is larger than a predefined threshold \( \text{Th}_s \), the tracking is considered successful in that frame. Then, SR is defined as the ratio of the number of success frames and the total frames. By assigning different values to \( \text{Th}_{\text{p}} \) and \( \text{Th}_s \), the precision plots and success plots can be obtained to demonstrate the overall performance of the tracker. Moreover, the area under the curve can be used as the evaluation criterion as well.

B. Comparison With State-of-the-Art Trackers on OTB2013
Denoting the proposed scale-adaptive correlation filter and ridge regression-based tracker as SCFR, then we evaluate its performance in the OTB2013 dataset and compare it with the top several trackers in the ranking list of the OTB2013 [3], including structured output tracking with kernels (Struck) [14], tracker by sparsity-based collaborative model (SCM) [23], tracking-learning-detection (TLD) [46], tracker with adaptive structural local sparse appearance model (ASLA) [47], context tracker (CTX) [48], visual tracking decomposition (VTD) [49], and visual tracker sampler (VTS) [50], and some recent famous trackers efficient convolution operators for tracking (ECO) [34], action-decision networks for visual tracking (ADNet) [51], multidomain convolutional neural networks for visual tracking (MDNet) [52], and tracking via multiple experts using entropy minimization (MEEM) [44].

A larger \( \sigma^2_d \) may make the tracker less robust to similar distractors while a smaller \( \sigma^2_d \) will decrease the accuracy of the filtering result. All parameters are fixed on the experiments.

First, we compare the overall performance of the competing trackers on all 51 sequences. Table I shows the comparison results on average CLE, average VOR, precision (\( \text{Th}_{\text{p}} = 20 \)), and SR (\( \text{Th}_s = 0.5 \)) results of SCFR and the 11 famous trackers in the OTB2013.

<table>
<thead>
<tr>
<th>Method</th>
<th>Average CLE</th>
<th>Average VOR</th>
<th>Precision (20)</th>
<th>SR (0.5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCFR</td>
<td>29.3</td>
<td>0.022</td>
<td>0.835</td>
<td>0.764</td>
</tr>
<tr>
<td>ECO</td>
<td>17.2</td>
<td>0.710</td>
<td>0.915</td>
<td>0.870</td>
</tr>
<tr>
<td>SCM</td>
<td>14.5</td>
<td>0.661</td>
<td>0.832</td>
<td>0.834</td>
</tr>
<tr>
<td>ASLA</td>
<td>6.8</td>
<td>0.816</td>
<td>0.932</td>
<td>0.896</td>
</tr>
<tr>
<td>MEEM</td>
<td>24.2</td>
<td>0.566</td>
<td>0.806</td>
<td>0.699</td>
</tr>
<tr>
<td>Strick</td>
<td>30.7</td>
<td>0.476</td>
<td>0.654</td>
<td>0.537</td>
</tr>
<tr>
<td>VTS</td>
<td>51.0</td>
<td>0.420</td>
<td>0.583</td>
<td>0.498</td>
</tr>
<tr>
<td>VTS</td>
<td>51.0</td>
<td>0.420</td>
<td>0.582</td>
<td>0.501</td>
</tr>
</tbody>
</table>
trackers. From Table I, we can find that our SCFR method achieves the desirable results on all 4 criteria. The average CLE of SCFR is 29.3 pixels, which is larger than ECO, ADNet, MDNet, and MEEM, but smaller than the remaining competing trackers. The average VOR of SCFR is 0.622, which is only lower than ECO, ADNet, and MDNet, and is better than many other trackers as well. For the competing trackers, Struck and SCM obtain the best precision and is better than many other trackers as well. For the competition, ADNet, MDNet, and MEEM, but smaller than the remaining trackers only use the traditional handcrafted features, they perform better than the others. Our SCFR method is interior to ECO, ADNet, and MDNet, but outperforms all of the remaining trackers without deep features. The precision plots and the success plots are shown in Fig. 9, which shows the average precision and SR on all 51 sequences under different thresholds. It can be observed that the performance of SCFR is better than most existing state-of-the-art trackers without deep features in the OTB2013 dataset as well.

Next, we evaluate the performance of the trackers under different conditions by the results on the sequences with different attributes. The average precision plots and success plots are illustrated in Fig. 10, from which we can find that our SCFR acquires better performance than many trackers in most conditions, including occlusion, deformation, scale variation, background clutter, etc. Therefore, we explain the mechanism that makes SCFR achieve good performance.

Occlusions: Fig. 10(a) shows the precision plots and success plots of the competing trackers on 29 sequences in the condition of occlusions. It can be observed that when $T_{th} = 20$ pixels and $T_{th} = 0.5$, SCFR outperforms SCM which performs best among the competing trackers in the OTB2013 ranking list by more than 10% on both precision and SR, and is only weaker than ECO, ADNet, MDNet, and MEEM. As mentioned before, SCFR follows the tracking-by-detection framework, and it has better discriminability to the background including occlusions. Besides, the multichannel feature is extracted pixel wise, and the HOG feature works in the local manner. Thus, SCFR is robust to partial occlusions.

Deformations: Fig. 10(b) gives the precision plots and success plots of the competing trackers on 19 sequences with deformations. It can be seen that the plots obtained by SCFR are much better than many other methods. This is because the learned correlation filter can build an accurate appearance model and the online update model can adapt to the deformation of the object to retain the accuracy of the appearance model, which makes SCFR have a desirable expression in the condition of deformation.

Out-of-Plane Variations: The comparison results of precision plots and success plots of the competing trackers in the condition of out-of-plane variations are displayed in Fig. 10(c). We can find that both of these plots achieved by SCFR significantly outperform most of the others. On one hand, the filter-based appearance model can provide finer discriminability. On the other hand, the incremental update strategy can learn that the appearance changes timely.

Scale Variations: Fig. 10(d) demonstrates the precision plots and success plots of the competing trackers on 28 sequences with scale variations. Our SCFR also has significant advantages, and it ranks fourth on both plots. A highlight of SCFR is that it utilizes a multiscale strategy based on interpolation to deal with the scale changes of the object. By the precision and success plots on the sequences with scale attributes, we can find that the proposed scale-adaptive correlation filter works well and outperforms SCM and ASLA which employ particle filters to address scale variations.

Fast Motion: The precision plots and success plots in Fig. 10(g) demonstrate that SCFR can acquire robust results when the objects move fast. In our formulation, we utilize the bounding box with the same size as the target for representation, which can effectively deal with the background changes caused by fast motion. Combined with the filter-based appearance model, SCFR performs well when the objects move fast.

Background Clutter: As the comparison results in Fig. 10(h) display, in the condition of background clutter, SCFR achieves the sixth best results. Since the filter exploits the relations of the target and the background, it provides a more accurate appearance model which has finer discriminability. Therefore, SCFR works well when there is background clutter.

C. Comparison With Other Correlation Filter-Based Trackers

We also compare the performance of our tracker with the related filter-based tracking methods, including KCF [16], CSK [15], CN [32], STC [37], DSST [17], and SAMF [18], all of which have publicly available codes. Different from the above trackers mentioned in Section II, SCFR models tracking from the signal detection perspective and accords with the tracking-by-detection framework. The correlation filter in our method is learned with the same size as the target, which can alleviate the effect of the fast-changing background and does not need the circulant structure assumption for FFT. The SCFR method can address the scale variations adaptively by building multiscale filter banks.

Fig. 11 illustrates the evaluation of the overall performance of these trackers. For the precision plots, SCFR, KCF, SAMF, and DSST have similar performance and significantly
outperform the rest trackers. Moreover, for the success plots, SAMF and DSST with scale adaption perform better than the trackers using fixed-size bounding boxes for representation. However, SCFR has better performance than SAMF and DSST. In practice, this benefits much from the success in the condition of scale variation and fast motion. On one hand, the proposed multiscale strategy in the filter domain can adaptively handle the scale variation issue, which expresses better than the scale adaption in image domain of SAMF and DSST. On the other hand, the bounding box with the same size as the target can well exclude the influence of the fast-changing background.

As Fig. 12(a) shows that because of the proposed scale adaption module, SCFR significantly outperforms the other trackers on both the average precision plots and average success plots of all 28 sequences with scale changes. For the competing trackers, KCF, CN, and CSK all make use of the fixed-size bounding box for representation, while STC, SAMF, and DSST use different scale adaption strategies. We can see that the scale adaption module of SCFR works well. Fig. 12(b) demonstrates the comparison results in the condition of fast motion. It can be observed that SCFR performs much better than the other methods, which exhibits that setting the bounding box to be the same size as the target can effectively decrease the distraction from the background. Most of the competing trackers employ a much larger bounding box for representation to satisfy the circulant structure assumption, which may be more easily affected by the fast-changing background.

D. Analysis of Multiscale Filters
To address the scale adaption issue, we propose the multiscale filter banks by making use of the spatial structure and interpolation. Besides, we set the parameter corresponding to the number of scales as 5, which means the multiscale filter
banks are generated by 5 scales. Therefore, to better explore the impact of the multiscale filter banks, we construct another 3 trackers with the multiscale filters in different scales, including 7 scales, 3 scales, and 1 scale. Note that the tracker with only 1 scale corresponds to the tracker without scale adaption.

We run these trackers in the OTB2013 dataset and the comparison results are displayed in Fig. 13. It can be found that when 5 scales are utilized, the SCFR tracker can get the best precision plot and success plot. We can also observe that the trackers with 3 scales and 7 scales express some weaker than the tracker with 5 scales. For example, the tracker with 5 scales outperforms that with 7 scales by about 1% on both precision and AUC score. Specifically, when compared with the tracker without multiscale filter banks, the tracker with 5 scales has significant advantages. Specifically, the tracker with 5 scales outperforms the tracker with only 1 scale and without scale adaption by about 6% and 7% on the precision and AUC score, respectively, indicating that the multiscale filter banks can effectively improve tracking performance by dealing with the scale variation problem. Commonly, the performance of the tracker with more scales can get better tracking results in theory, which can better deal with the scale change problem. However, the filter banks with more scales will cost more time.

We construct a new tracker which only uses the filtering result without the distance importance map to determine the location. We compare the new tracker without importance weight with the standard SCFR which retains the distance importance module in the OTB2013 dataset. The comparison results are shown in Fig. 14, from which we can observe that performance of the tracker with distance importance significantly outperforms that without distance importance. Note that the precisions of the tracker without distance importance are 0.735 and 0.558, which are about 10% and 6% lower than that with distance importance. This indicates that the performance of the tracker is greatly improved by considering the distance importance. In practice, during the tracking process, there exist many similar distractors which may make the tracking drift. However, by calculating the confidence map weighted by the distance importance, the impact caused by the similar distractors which are far away from the target can be effectively alleviated, and the tracking performance can be improved.

F. Evaluation in More Datasets

Besides the OTB2013 dataset in which the SCFR tracker has achieved good results, we also evaluate the performance of our SCFR tracker in more datasets, including the OTB-100 dataset [53] and VOT2016 dataset [54], to explore the effect of the settings of the tracker.

Evaluation in the OTB-100 Dataset: We further evaluate the performance of SCFR in the OTB-100 dataset which is the extension version of OTB2013. Besides the original sequences in OTB2013, more sequences are added up to 100 in the OTB-100 dataset. Therefore, we compare SCFR with several famous tracking methods, including ECO [34], MDNet [52],
Fig. 15. Precision plots and success plots of SCFR and the competing trackers in the OTB-100 dataset.

ADNet [51], ASLA [47], SCM [23], locally orderless tracking [55], CSK [15], CXT [48], OAB [20], multitask sparse learning tracking [56], color-based probabilistic tracking [57], local sparse appearance model and K-selection tracking [58], Kernel MeanShift [59], and the multiple instance learning tracker (MIL) [11] in this dataset. The comparison results of precision plots and success plots are shown in Fig. 15, from which we can obtain the similar conclusion with that in OTB2013. It can be observed that the precision at $T_{th} = 20$ pixels obtained by SCFR is 0.744 and the AUC score is 0.549, which ranks fourth and third, respectively. It can be found that the trackers with deep features also perform better than the others, but SCFR almost outperforms all of the trackers without deep features. The overall performance of SCFR is similar to MEEM, worse than ECO and MDNet, and better than the remaining trackers.

**Evaluation in the VOT2016 Dataset:** We also evaluate our SCRF method in the VOT2016 dataset. There are 65 sequences in the VOT2016 dataset, which captures deformations, occlusions, illumination changes, abrupt motion changes, etc. Specifically, the VOT framework uses two weakly correlated metrics, the accuracy and the robustness, for a standardized analysis, where the accuracy is defined as the average bounding box overlap and the robustness denotes the number of reinitializations. The evaluation is conducted by ranking analysis based on these two criteria. Hereby, we compare our SCFR method with 18 famous tracking methods, including the scale-and-state aware tracker, multilevel deep feature tracker, correlation filters with convolutional features tracker extended with response information failure detection, best displacement flow [60], scalable kernel correlation filter with sparse feature integration [61], dynamic feature selection tracker [62], optical-flow clustering tracker, DSST [17], adaptive color tracker [63], MDNet-N, guided correlation filter, anchor template tracker, transfer learning-based visual tracking with Gaussian processes regression [64], adaptive regression target DSST, MIL [11], DFT [65], incremental learning for robust visual tracking [5], and the anchor template tracker (NCC) [66] in the VOT2016 dataset, and show the comparison results in Table II following the evaluation framework of VOT. The comparison results are obtained in baseline manner with initialization in the ground truth. From Table II, we can find that the average ranking of our SCFR tracker is 3.8 on accuracy which outperforms many competing trackers. SCFR acquires the robustness ranking 8.52 and expected overlap 0.1722, both of which rank in the middle among the trackers. Specifically, the ranking in accuracy of SCFR is higher than that in robustness, which guides us to improve the robustness of SCFR in the future.

**G. Limitations**

As mentioned above, SCFR acquires good precision and SR results in the benchmark datasets, especially in the conditions of scale variations and fast motions. However, we should note that SCFR fails on some very challenging sequences, by which we analyze the limitations of SCFR.

Fig. 16 demonstrates three failure examples on some representative sequences. On *ironman*, the target moves so fast, together with heavy deformation, illumination variation, and scale changes, that tracking begins to drift in the early stage (e.g., in Frame 28). On the other hand, the fast, heavy deformation of consecutive frames makes the learned filters quite different, which will degrade the robustness of the appearance model by incremental learning. On *lemming*, heavy occlusion lasts for a long time, which accumulates the errors of the learned appearance model. To handle the issue of long-term full occlusions, the occlusion judging strategy and motion prediction model can be considered to improve the tracking performance.

In our MATLAB platform on PC with an Intel i7 CPU 3.4 GHz, the running speed of our standard SCFR tracker is 2.0 frames/s, and the speed of the tracker without scale adaption is 9.1 frames/s, both of which are faster than MDNet, SCM, ALSA, VTD, VTS, etc., but slower than ECO, MEEM, KCF, Struck, TLD, etc. Although SCFR has not achieved real-time performance up to now, we can speed it up by employing parallel strategy in several modules of the model in the future. On one hand, because the multiscale filters are implemented by
interpolation and used to realize the tracking separately, they can work in a parallel manner and each filter can be set in a single thread which can reduce the running time to only about $1/S$ times of the original filter banks’ cost time, where $S$ is the number of scales. On the other hand, the computation on the multichannel features can also be divided into single channels. By assigning a work thread to each channel and conducting the FFT operation on all channels at the same time, the running speed of SCFR can be further improved. Because the parallel strategy does not decrease the tracking accuracy, it may be a feasible and effective way to improve the tracking speed of SCFR.

VI. Conclusion

In this paper, we propose a novel tracking method based on the correlation filtering formulation as well as ridge regression formulation. From the signal detection perspective, the relations of the neighboring sample patches are fully exploited and a tight correlation filter is learned to represent the tracker without the circulant structure assumption. By making use of the spatial structure of the filter, the multiscale filter banks can be implemented to handle the challenging scale variation problem adaptively. Moreover, a distance importance-based confidence calculation model is presented to determine the final tracking result. Experimental results demonstrate that the proposed method can outperform many state-of-the-art trackers in the benchmark datasets. Further, we would like to introduce deep features for better tracking performance and improve the running speed of our tracker to satisfy practical applications.

REFERENCES
