

Received December 18, 2018, accepted January 7, 2019, date of publication January 16, 2019, date of current version February 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2892429

Sampling Operator to Learn the Scalable Correlation Filter for Visual Tracking

MINKYU LEE¹, TAEHO KIM¹, YUSEOK BAN², EUNGYEOL SONG¹,
AND SANGYOUN LEE¹, (Member, IEEE)

¹Department of Electrical and Electronic Engineering, Yonsei University, Seoul 03722, South Korea

²Agency for Defense Development, Daejeon 34060, South Korea

Corresponding author: Sangyoun Lee (sylee@yonsei.ac.kr)

This work was partly supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (2016-0-00197, Development of the high-precision natural 3D view generation technology using smart-car multi sensors and deep learning) and Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (2016-0-00562, Emotional Intelligence Technology to Infer Human Emotion and Carry on Dialogue Accordingly).

ABSTRACT The correlation filter is suitable for tracking on account of its low computational complexity and promising performance. However, the number of available training samples is limited to the filter size, and the lack of samples leads to poor generalization. Moreover, spectral leakage degrades the filter quality. In this paper, we, therefore, propose a sampling operator for learning a scalable correlation filter in an enlarged window, whose size is larger than the object size. The scalable filter encodes the sparse frequency representation to reconstruct a larger filter with zeros outside of the object in the spatial domain. The sampling operator, which is composed of windowing and sampling operations, enables learning the scalable filter from a large window, and it suppresses spectral leakage. Our method was evaluated on the OTB-100, TC-128, and UAV-123 datasets and achieved the promising results in terms of precision and success rates.

INDEX TERMS Visual tracking, correlation filter, sampling operator, scalability.

I. INTRODUCTION

Visual tracking has been widely used in numerous computer vision applications, such as surveillance systems [1], key point tracking [2] for augmented reality or three-dimensional (3D) object reconstruction, and object recognition [3]. Various tracking methods [4]–[8] have emerged over the last several years and achieved considerable performance. Nevertheless, it remains challenging to track an object robustly in the presence of illumination variation, scale variation, background clutter, rigid/non-rigid transformation, occlusion, and fast motion.

Recently, trackers have been developed based on the correlation filter (CF) on account of its low computational complexity and promising performance [5], [9]–[17]. In CF, the circularly shifted samples of an object are used as hard negative samples to produce a framework for learning the filter with a convolution operator. According to the Fourier transform property, the convolution operation in the spatial domain is the same as element-wise multiplication in the frequency domain. Consequently, the filter is efficiently learned in the frequency domain with an $O(N)$ complexity, where N is the object area.

However, the number of available samples is limited to the object size, which should be adequately small to avoid overfitting to the background. Since the number of possible cyclic shifts is equal to the object area, an inadequate number of samples leads to poor filter generalization. Another problem is that spectral leakage [18] degrades the filter quality. Both the samples and filter have a discontinuous cyclic boundary, which generates spectral leakage in the frequency domain via the discrete Fourier transform.

A few CF trackers exist that enhance the filter quality with more negative samples than the object size (width×height). Galoogahi *et al.* [19] proposed correlation filters with limited boundaries (CFLB). CFLB learns small filters in enlarged windows using a zero-padding matrix, where small filters are zero-padded to generate large filters in the spatial domain. Then, the filter is estimated by augmented Lagrangian multipliers (ALM). Danelljan *et al.* [20] proposed spatially regularized discriminant correlation filters (SRDCF) to learn large filters in large windows by using a spatial regularizer. Mueller *et al.* [21] proposed the context aware correlation filter (CACF) tracking method. The small filter is learned by several small windows, among which one is the object and

the others comprise the context. Lukežič *et al.* [16] proposed the channel and spatial reliability discriminative correlation filter (CSR-DCF), which is used to weight features in an enlarged search window. Furthermore, Galoogahi *et al.* [22] proposed the background-aware correlation filter (BACF) tracker, which employs background samples as negative samples.

In this paper, we propose a sampling operator to learn a scalable correlation filter in an enlarged window, which includes a large number of samples. To learn the scalable filter, the sampling operator is used in the frequency domain to encode the sparse frequency representation of a filter and to suppress spectral leakage. Here, the term ‘scalable correlation filter’ refers to having capability of scaling filter from an encoded correlation filter in frequency domain while preserving information within interested bandwidth in spatial domain. The similar term ‘scalable’ is addressed from scalable kernel correlation filter (sKCF) [23]. It means that the capability of handling scale changes of an object in a fixed size filter regardless of scale variation by applying adjustable Gaussian window function. The main difference between sKCF and our proposed method is as follow: the window function of sKCF depends on the size variation of an object to learn a fixed-size filter. In other words, there is no resizing of image for scale variation and it might impose information of various scaled objects in fixed size filter. It leads to decreasing discrimination of correlation filter. On the other hand, our window function depends on the size of an object. Therefore, our method leads to consistent and discriminant correlation filter.

In a view of signal processing, the spectral leakage should be carefully treated because it leads to unwanted artifacts on the signal when we try to recover the desired signal. Although many CF-based trackers apply hanning window in search area, its sub-windows still suffer from spectral leakage. CFLB [19] and BACF [22] leverage sub-windows but they do not consider this problem. Therefore, they suffer from undesirable periodic repetition of main lobe. This problem is presented in Sec. IV-C.

The principle of scalable filter representation is derived from the duality of the Fourier transform, where a zero-padded signal in the spatial domain corresponds to ideally sampled frequency components in the frequency domain [24]. The sampling operator is composed of the point sampling operation and windowing operation. To detect the object in an enlarged window, a large filter is reconstructed from the scalable filter by the sampling operator (Fig. 1a). The reconstructed filter is then used to estimate the response map (Fig. 1b). To improve the filter quality, the channel-dependent filter learning framework is adopted.

The point sampling operation reduces the number of filter parameters to provide a better generalized tracking model. The windowing operation filters the band-limited signal in the spatial domain and generates a smooth boundary for scalable frequency representation. Accordingly, the proposed sampling operator enables a filter to learn in a larger window

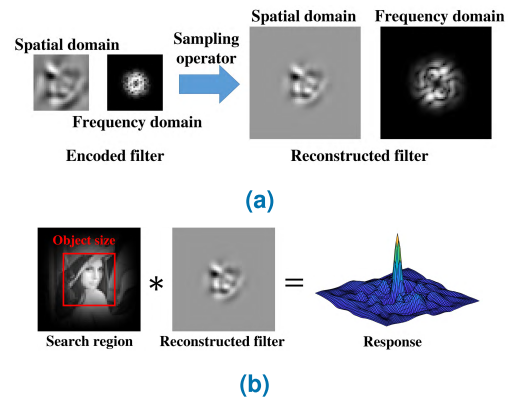


FIGURE 1. Filter reconstruction with sampling operation and its tracking process. (a) Sampling operation. (b) Tracking process with the reconstructed filter.

than the filter size, and it suppresses spectral leakage caused by the discrete Fourier transform in the frequency domain.

The proposed method was evaluated based on success and precision rate on OTB-100, TC-128, and UAV-123 datasets. It achieved promising results in terms of precision and success rate.

II. RELATED WORKS

A. CF LEARNING USING A SAMPLE SIZE GREATER THAN THE OBJECT SIZE

The CF can use only a limited number of samples to utilize its diagonal structure for fast computation. This limitation leads to poor generalization of the tracking model. As mentioned earlier, Galoogahi *et al.* [19] proposed CFLB to reduce the boundary effect. This approach uses a large number of samples simultaneously with implicit zero-padding of the filter in the spatial domain, as well as the alternating direction method of the multiplier (ADMM). Danelljan *et al.* [20] proposed spatial regularization to learn the filter in an enlarged window. Mueller *et al.* [21] proposed the context-aware correlation filter (CACF) to use context samples in filter learning and a diagonal structure for fast computation. Lukežič *et al.* [16] proposed the channel and spatial reliability discriminative correlation filter (CSR-DCF), which uses a weighting feature in an enlarged search window. Furthermore, Galoogahi *et al.* [22] proposed the background-aware correlation filter (BACF) tracker, which is extended by CFLB and learns the background object from a larger window than CFLB.

Our work seems similar to SRDCF and BACF, but there are some differences as follows. The proposed method uses the smaller size filter than SRDCF to reduce overfitting problem due to massive number of trainable parameters [10]. BACF learns the filter alternately in the spatial and frequency domain with dual variables without any consideration of spectral leakage. In contrast, We derive a closed-form solution with the sampling operator, which deals with scaling and suppressing spectral leakage simultaneously for data sampling in the frequency domain. It is described in Sec. IV-C.

B. CF WITH REDUCED SPECTRAL LEAKAGE

Spectral leakage is an unavoidable problem when the correlation filter is learned. Thus, Bolme *et al.* [5] used the Hann window to suppress leakage caused by the discontinuity of the cyclic boundary. This approach generates smoothly connected samples in a circular pattern around the boundary, and it reduces values around the boundary for assigning a high weight to the filter around the object. Generally, CF-based trackers use the Hann window for preprocessing.

C. CF WITH A MULTI-CHANNEL DEPENDENT FILTER

Channel-dependent learning of multi-channel filters is crucial to improve the quality of the filters. For each channel, the discrimination power of filters depends on the feature representation of positive and negative samples. In channel-independent filter estimation, the output response is incorrectly estimated if non-discriminant filters have large peak values in tracking on account of all filters having equal weights in channel-independent learning. The channel-dependent learning of multi-channel filters was introduced by Galoogahi *et al.* [25]. SRDCF [20] also uses channel-dependent learning, and its extension has been widely used and has achieved the state-of-the-art performance [10], [12], [13], [26]. Furthermore, as an extension of CFLB, BACF also adopted channel-dependent learning [22].

D. OTHER CF EXTENSIONS

Except for the above classes, various extensions of the CF tracker exist, including adopting discriminant features, such as the histogram of gradient (HoG) [14], color name (CN) [15], color histogram [9], and deep feature [12], [27], kernel trick [28], scale estimation [11], [15], [29]–[31], training set adaption [10], [26], response adaption [32], [33], part based model [17], [34], [35], continuous extension [13], combined with attention or saliency [16], [36]–[39], and deep learning [40], [41].

III. METHODS

A. SAMPLING OPERATOR

The sampling operator aims to enable learning the scalable filter from an enlarged search window. The idea of the scalable filter is motivated from the sampling theory of the Fourier transform. According to the sampling theory, interleaving zeros at each sample point of the signal in the spatial domain establishes a periodic repetition of the band-limited signal in the frequency domain. This idea provides the key to reconstruct a continuous signal by applying the low-pass filter on a discrete one. This approach strongly inspired our sampling operator. The object patch can be regarded as a band-limited patch in an enlarged window, while the scaling can be deemed as a sampling in the frequency domain.

Our sampling operator is composed of point sampling and windowing. Point sampling is the process of interleaving zero values for each sample or extracting the sample points. Windowing is the process of filtering the band-limited signal

in the spatial domain, which is similar to the low-pass filter in continuous signal reconstruction.

Let $\mathbf{f}_e$ be the single-channel scalable filter vector with size K . Then the original or reconstructed filter $\mathbf{f}_r$ with size N is represented by

$$\mathbf{f}_r = \boldsymbol{\psi} \odot \mathbf{F}^{-1}(\Phi \hat{\mathbf{f}}_e), \quad (1)$$

where $\hat{\cdot}$ is frequency domain representation, $\mathbf{F}$ is the discrete Fourier transform (DFT) [24] matrix, such that $\hat{\mathbf{f}} = \mathbf{F}\mathbf{f}$, $\mathbf{F}^{-1}$ is the inverse DFT, $\odot$ is element-wise multiplication, Φ denotes point sampling matrix with size $N \times K$, and $\boldsymbol{\psi}$ represents the windowing vector with size N at spatial domain. Note that K is integer number smaller than N .

For simplicity, assume that $sK = N$ for a scale factor, s , is positive integer. Then, the point sampling matrix is given by

$$\Phi = [\delta[n] \delta[n-s] \delta[n-2s] \dots \delta[n-(K-1)s]], \quad (2)$$

where $\delta[n]$ is binary vector, where the element is one only if $n = 0$.

B. LEARNING SCALABLE FILTER

In this section, we first derive our scalable filter equation from the general correlation filter. Then, we extend it with multi-channel dependent learning. Finally, optimization and model update methods are described in detail.

1) CLOSED-FORM SOLUTION DERIVATION

The objective of the CF tracker is to estimate the filter, when the convolution of the sample, $\mathbf{x} \in \mathbb{R}^N$, and filter, $\mathbf{f} \in \mathbb{R}^N$ results in the Gaussian-shaped response, $\mathbf{y} \in \mathbb{R}^N$. By the least squares method, the error function is described as:

$$E(\mathbf{f}) = \|\mathbf{x} * \mathbf{f} - \mathbf{y}\|_2 + \lambda \|\mathbf{f}\|_2, \quad (3)$$

where λ is the regularization parameter and $*$ is the circular convolution operator. Our objective is to find the optimal compact filter over a much larger number of samples than the object size.

To derive scalable correlation filter learning, suppose that we use multi-channel features and multi-channel filters, and each filter is totally independent. Then, our objective function is

$$E(\mathbf{f}_r) = \sum_{i=1}^L \|\mathbf{x}_i * \mathbf{f}_{r,i} - \mathbf{y}\|_2 + \lambda \sum_{i=1}^L \|\mathbf{f}_{r,i}\|_2, \quad (4)$$

where $\mathbf{f}_r = [\mathbf{f}_{r,1}^T \mathbf{f}_{r,2}^T \dots \mathbf{f}_{r,L}^T]^T$ is reconstructed multi-channel filters and L is the number of channels. By substituting Eq. (1) into Eq. (4) and modifying the regularization term, we obtain

$$E(\mathbf{f}_e) = \sum_{i=1}^L \|\mathbf{x}_i * (\boldsymbol{\psi} \odot \mathbf{F}^{-1}(\Phi \hat{\mathbf{f}}_{e,i})) - \mathbf{y}\|_2 + \lambda \sum_{i=1}^L \|\mathbf{f}_{e,i}\|_2, \quad (5)$$

where $\mathbf{f}_e = [\mathbf{f}_{e,1}^T \mathbf{f}_{e,2}^T \dots \mathbf{f}_{e,L}^T]^T$ is scalable multi-channel filters.

It is not straightforward to estimate the multi-channel scalable filter, $\mathbf{f}_r$, because various operations are mixed and the computation of circular convolution is inevitable in Eq. (5). To derive the closed form and efficient solution, Eq. (5) should be transformed into the frequency domain:

$$E(\hat{\mathbf{f}}_e) = \sum_{i=1}^L \|\mathbf{D}(\hat{\mathbf{x}}_i)\mathbf{M}\hat{\mathbf{f}}_{e,i} - \hat{\mathbf{y}}\|_2 + \lambda \sum_{i=1}^L \|\hat{\mathbf{f}}_{e,i}\|_2, \quad (6)$$

where $\mathbf{D}(\hat{\mathbf{x}}_i)$ is the diagonalized matrix of $\hat{\mathbf{x}}_i$, $\mathbf{M} = \mathbf{C}(\hat{\boldsymbol{\psi}})\Phi$ is the sampling operation matrix, and $\mathbf{C}(\hat{\boldsymbol{\psi}})$ is the circulant matrix of base $\hat{\boldsymbol{\psi}}$.

Here, the windowing function, $\hat{\boldsymbol{\psi}}$, should be carefully chosen to enable the filters to have smooth and continuous boundaries to suppress spectral leakage. Moreover, it should be adequately sparse to estimate a compacted and scaled filter with low computational complexity. There are various windowing functions, such as a rectangular window, B-spline window, Hann window, and Gaussian window. In our method, the truncated sinc-squared function is selected for DFT of the windowing function, $\hat{\boldsymbol{\psi}}$, because it satisfies the two aforementioned necessary conditions. The truncated sinc-squared function is defined as

$$\hat{\boldsymbol{\psi}}[n] = \text{rect}(t/2)\text{sinc}^2(t)|_{t=(n-N/2)T} \quad (7)$$

where T is the sampling period. In the window function, the number of nonzero values is determined by the sampling period.

In channel-independent learning, we can simplify Eq. (6) by reorganizing it in a matrix multiplication form.

$$E(\hat{\mathbf{f}}_e) = \|\mathbf{A}(\hat{\mathbf{x}})\hat{\mathbf{f}}_e - \hat{\mathbf{z}}\|_2 + \lambda \|\hat{\mathbf{f}}_e\|_2, \quad (8)$$

where

$$\mathbf{A}(\hat{\mathbf{x}}) = \begin{bmatrix} \mathbf{D}(\hat{\mathbf{x}}_1)\mathbf{M} & 0 & \dots & 0 \\ 0 & \mathbf{D}(\hat{\mathbf{x}}_2)\mathbf{M} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{D}(\hat{\mathbf{x}}_L)\mathbf{M} \end{bmatrix} \quad (9)$$

and

$$\hat{\mathbf{z}} = \begin{bmatrix} \hat{\mathbf{y}} \\ \hat{\mathbf{y}} \\ \vdots \\ \hat{\mathbf{y}} \end{bmatrix}. \quad (10)$$

2) EXTENSION TO THE CHANNEL-DEPENDENT CORRELATION FILTER

To effectively utilize the multi-channel feature, we should consider the channel dependency on the weighting channel correlation in a pixel-wise manner and satisfy their sum being equal to $\mathbf{y}$. The equation of the channel-dependent correlation filter is

$$E(\hat{\mathbf{f}}_e) = \left\| \sum_{i=1}^L \mathbf{D}(\hat{\mathbf{x}}_i)\mathbf{M}\hat{\mathbf{f}}_{e,i} - \hat{\mathbf{y}} \right\|_2 + \lambda \sum_{i=1}^L \|\hat{\mathbf{f}}_{e,i}\|_2. \quad (11)$$

In channel-dependent learning, we can simplify Eq. (11) by reorganizing it into matrix multiplication form.

$$E(\hat{\mathbf{f}}_e) = \|\mathbf{A}(\hat{\mathbf{x}})\hat{\mathbf{f}}_e - \hat{\mathbf{y}}\|_2 + \lambda \|\hat{\mathbf{f}}_e\|_2, \quad (12)$$

where

$$\mathbf{A}(\hat{\mathbf{x}}) = [\mathbf{D}(\hat{\mathbf{x}}_1)\mathbf{M} \quad \mathbf{D}(\hat{\mathbf{x}}_2)\mathbf{M} \quad \dots \quad \mathbf{D}(\hat{\mathbf{x}}_c)\mathbf{M}]. \quad (13)$$

3) OPTIMIZATION

To find the optimal filters, we use iterative least squares optimization since the computational complexity of pseudo inverse of $\mathbf{A}(\hat{\mathbf{x}})$ is very expensive. For a single sample, the derivative of Eq. (12) is reduced to

$$\mathbf{A}(\hat{\mathbf{x}})^H \hat{\mathbf{y}} = (\mathbf{A}(\hat{\mathbf{x}})^H \mathbf{A}(\hat{\mathbf{x}}) + \lambda \mathbf{I}) \hat{\mathbf{f}}_e, \quad (14)$$

where ^{H} denotes Hermitian transpose.

Here, we use online preconditioned conjugate gradient optimization, which is applied in [10] and [13], for rapid and accurate filter estimation. For each sequential frame, the previous conjugate direction is partially used. In other words, it has the similar effect of the number of iterations being separately computed for each sequential frame.

4) MODEL UPDATE

For multiple samples, the weighted least squares error is derived from Eq. (12).

$$E(\hat{\mathbf{f}}_e) = \sum_{i=1}^N w_i \|\mathbf{A}(\hat{\mathbf{x}}^i) \hat{\mathbf{f}}_e - \hat{\mathbf{y}}^i\|_2 + \lambda \|\hat{\mathbf{f}}_e\|_2, \quad (15)$$

where w_i is the weight for i -th sample and D is the number of samples. For simplicity, we denote $\mathbf{A}(\hat{\mathbf{x}}^i)$ to $\mathbf{A}_i$. The derivative of Eq. (15) also can be reduced to

$$\left(\sum_{i=1}^D w_i \mathbf{A}_i^H \mathbf{A}_i + \lambda \mathbf{I} \right) \hat{\mathbf{f}}_e = \sum_{i=1}^D w_i \mathbf{A}_i^H \hat{\mathbf{y}}^i. \quad (16)$$

Here, we call $\mathbf{A}_i^H \mathbf{A}_i$ the i -th auto-correlation matrix and $\mathbf{A}_i^H \hat{\mathbf{y}}^i$ the cross-correlation matrix. We denote them as $\mathbf{S}_{xx}$ and $\mathbf{S}_{xy}$, respectively. In Eq. (16), we notice that the left side is the weighted sum of the auto-correlation matrix, and the right side is the weighted sum of the cross-correlation matrix for each sample. Therefore, the update of the auto-correlation and cross-correlation is the cumulative weighted sum.

$$\begin{cases} \tilde{\mathbf{S}}_{xx}^i = (1 - \eta) \tilde{\mathbf{S}}_{xx}^{i-1} + \eta \mathbf{S}_{xx}^i \\ \tilde{\mathbf{S}}_{xy}^i = (1 - \eta) \tilde{\mathbf{S}}_{xy}^{i-1} + \eta \mathbf{S}_{xy}^i \end{cases} \quad (17)$$

where $\tilde{\cdot}$ is online model and η is learning rate.

However, computing the auto-correlation is too expensive because it requires an immense amount of memory, $O(K^2 L^2)$, and this results in slow computation, even when the matrix is very sparse. To increase the speed, we store and use a limited number of samples and then compute the functional gradient for each iteration to satisfy the weighted least squares method. Both methods are used in [10] and [13].

IV. EXPERIMENTS

Our experiments were performed on an Intel i7-7700 3.6 GHz CPU with 24 GB of RAM. The evaluation was performed on OTB-100 [42], TC-128 [43], and UAV-123 [44] datasets. The sequences in these datasets were tagged with various attributes, including illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out-of-view, background clutter, and low resolution.

Two kinds of evaluation measures were used: success and precision plots. The success plot visualizes the rate of frames with smaller intersection over union (IoU) between the ground truth region and tracked region than a threshold. Typically, the area under the curve (AUC) of the success plot is used to rank trackers and it is called success rate. The precision plot visualizes the rate of the frames with smaller Euclidean distance than a threshold. Typically, the rate of the test frame with a 20-pixel threshold was used to rank the trackers and it is called precision rate.

A. IMPLEMENTATION DETAILS AND PARAMETERS

The point sampling matrix was designed for integer-scaled point sampling because of its simplicity in generating a matrix. The sampling matrix was generated by the tensor product of two independent periodic sampling matrixes for each axis. The principle is similar to the Fourier transform. The Fourier transform is a dimension independent operation and has a commutative property for the multi-dimensional Fourier transform. Consequently, the DFT matrix for the multi-dimensional Fourier transform is generated by the tensor product of the DFT matrix for each axis.

For a feature descriptor, We adopted a multi-channel feature that combines HoG and CN features [15]. Some techniques were borrowed from the continuous convolution operator tracker (C-COT) [13]: a square-shaped area search strategy, limiting the maximum and minimum filter area, a sub-grid detection, and an online preconditioned conjugate gradient method.

As common parameters, which are frequently used in CF, we set the learning rate $\eta = 0.014$, and the standard deviation of the desired response $\sigma = \text{area of output} \times 1/15$. We set the search area to 4^2 times the object size with a square-like shape, the maximum filter size was set to 50^2 , and the minimum filter size was 40^2 . In addition, we used object area padding to 0.5 to utilize outline features of the target. For scale estimation, we set the incremental scale step to 1.01 and the number of scale to 7. To generate the sampling matrix, we used the truncated sinc-squared function and set the number of nonzero values of windowing vector to $s \times 4.3$. For optimization, we adopted the online preconditioned conjugate gradient method. Its iteration was 15 in the first frame, and we set it to 3 for the others. For online gradient learning, the forgetting rate of conjugate directions was set to 3. To reduce the computation time, the direct residual computation was applied to conjugate gradient optimization with

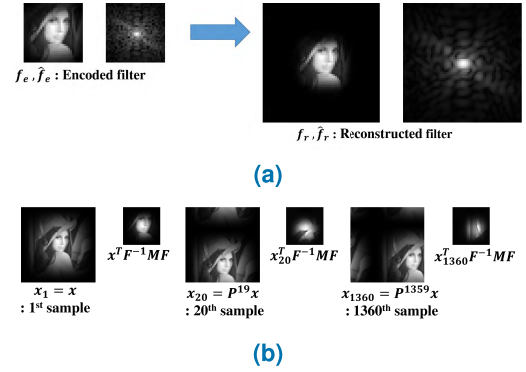


FIGURE 2. Effect of sampling operator for filter and training samples. (a) Sampling operation to the filter. (b) Sampling operation to training samples.

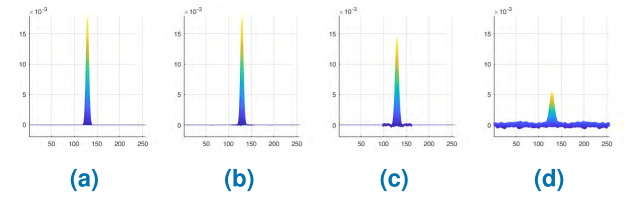


FIGURE 3. X-Z plot of filters in the spatial domain. (a) is the groundtruth Gaussian function. (b), (c), and (d) are filters estimated by ours, BACF, and SRDCF, respectively.

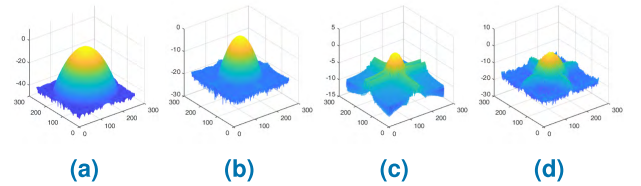


FIGURE 4. Log-magnitude of filters in the frequency domain. (a) is the groundtruth Gaussian function. (b), (c), and (d) are filters estimated by ours, BACF, and SRDCF, respectively.

limited training samples. Moreover, the centro-Hermitian structure was leveraged for faster computation. Both methods were already used in C-COT [13]. The number of stored samples was set to 200.

B. EFFECTS OF THE SAMPLING OPERATOR FOR THE FILTER AND TRAINING SAMPLES

In this section, we show how our sampling operator works for the filter and training samples. From Eq. (6), we know that the output response for the single-channel filter is given by

$$\hat{y} = D(\hat{x})M\hat{f}_e. \quad (18)$$

To visualize the effect of the sampling operator, Eq. (18) is transformed into the spatial domain.

$$y = C(x)F^{-1}MFf_e, \quad (19)$$

where $C(x) = F^{-1}D(\hat{x})F$. Here, $C(x)$ can be represented with the circular permutation matrix P , $C(x) = [xPxP^2x \dots P^N x]^T$.

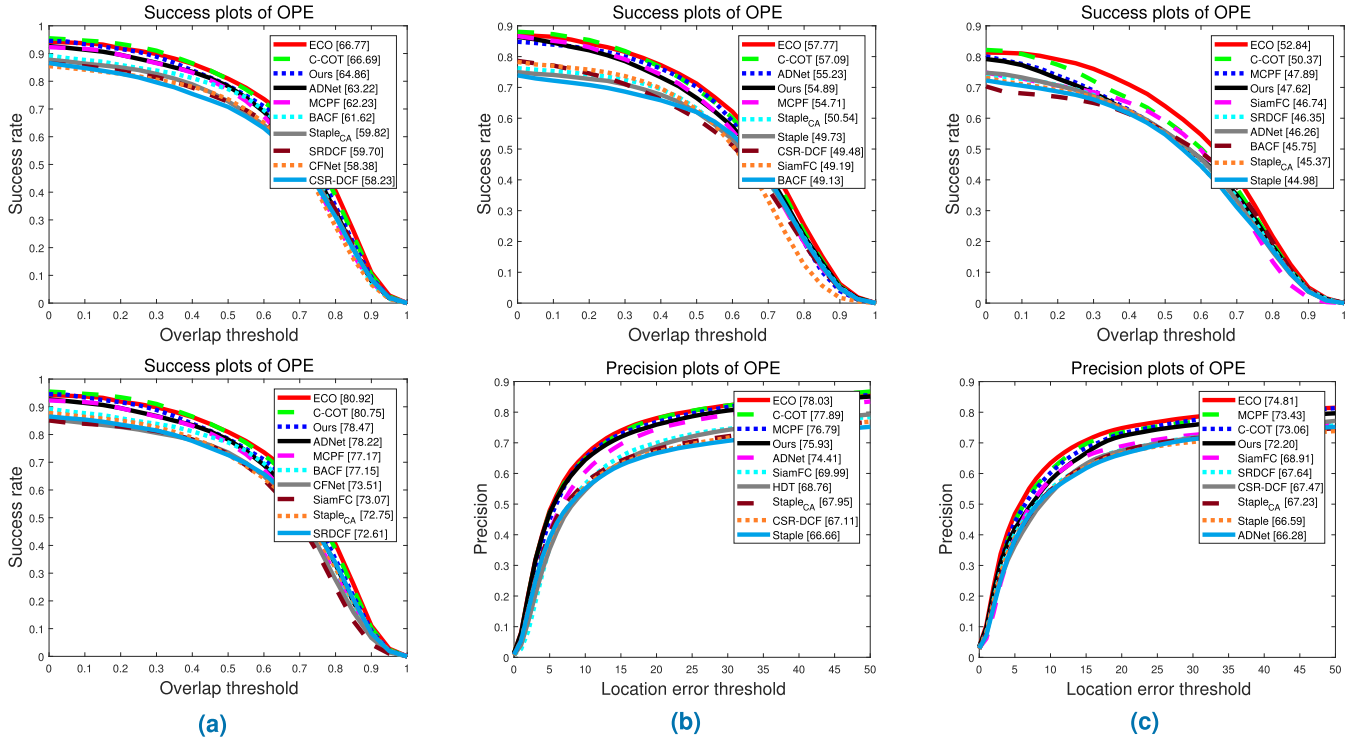


FIGURE 5. Comparison results with state-of-the-art trackers in terms of success plot and precision plot on OTB-100, TC-128, and UAV-123 datasets. Only the top-10 trackers are displayed. Success rate and precision rate are reported in the legend at success plot and precision plot, respectively. (a) OTB-100. (b) TC-128. (c) UAV-123.

In Fig. 2, for better visualization, the same image is used for the filter and training samples. On the filter side, the scalable filter, f_e , is interpolated with our sampling operator to reconstruct a large-sized filter, $f_r = F^{-1}MF_e$, in the frequency domain (Fig. 2a). Meanwhile, the training samples are decimated in the frequency domain and yield a small training patch with a smooth boundary, thereby suppressing spatial leakage, $C(x)F^{-1}MF$ (Fig. 2b).

C. COMPARISON WITH FILTERS OF SRDCF, BACF, AND OURS

We visually compare our proposed filter with those of BACF and SRDCF when the groundtruth filter, g , is known. Here, the response, y , is generated by convolution between input image, x , and groundtruth filter, g . For pair comparison, we set the number of iteration to 32 because the maximum iterations whose parameters were provided from BACF is 32. We choose Gaussian function as a groundtruth filter. Since Fourier transform of Gaussian function is also Gaussian function, it is effective for showing the suppression of spectral leakage. The groundtruth and filters estimated by ours, BACF, and SRDCF are visualized in Fig. 3 and in Fig. 4. Fig. 3 shows X-Z plot of filters in the spatial domain and Fig. 4 shows log-magnitude of filters in the frequency domain.

From the Fig. 3, SRDCF learns poor filter because they have too many parameters to estimate and their hyper-parameters should be carefully tuned [22]. The spectral

TABLE 1. Component comparison between baseline trackers and ours. s is scale factor.

Method	Kernel size	Sampling size	Channel dependency	Feature
CFLB [19]	N	sN	No	Normalized-Pixel
SRDCF [20]	sN	sN	Yes	HoG
MOSSE _{CA} [21]	N	N+4N	No	Normalized-Pixel
SAMF _{CA} [21]	N	N+4N	No	HoG, CN
Staple _{CA} [21]	N	N+4N	No	HoG, Color Histogram
CSR-DCF [16]	N	sN	No	HoG, CN
BACF [22]	N	sN	Yes	HoG
Ours	N	sN	Yes	HoG, CN

leakage leads to sidelobes, which are the convolutions of main lobe and Dirichlet kernel [18]. According to Fig. 4, the proposed method preserves original frequency structure and reduces the effect of spectral leakage in the frequency domain, while BACF seems to suffer from spectral leakage.

D. PERFORMANCE COMPARISON WITH RECENT TRACKERS

1) COMPARISON WITH BASELINE TRACKERS

As baseline trackers, CFLB [19], SRDCF [20], CACF [21], CSR-DCF [16], and BACF [22] were selected since they belong to the same CF extension class as the tracker, which strives to learn filter with the greater number of negative

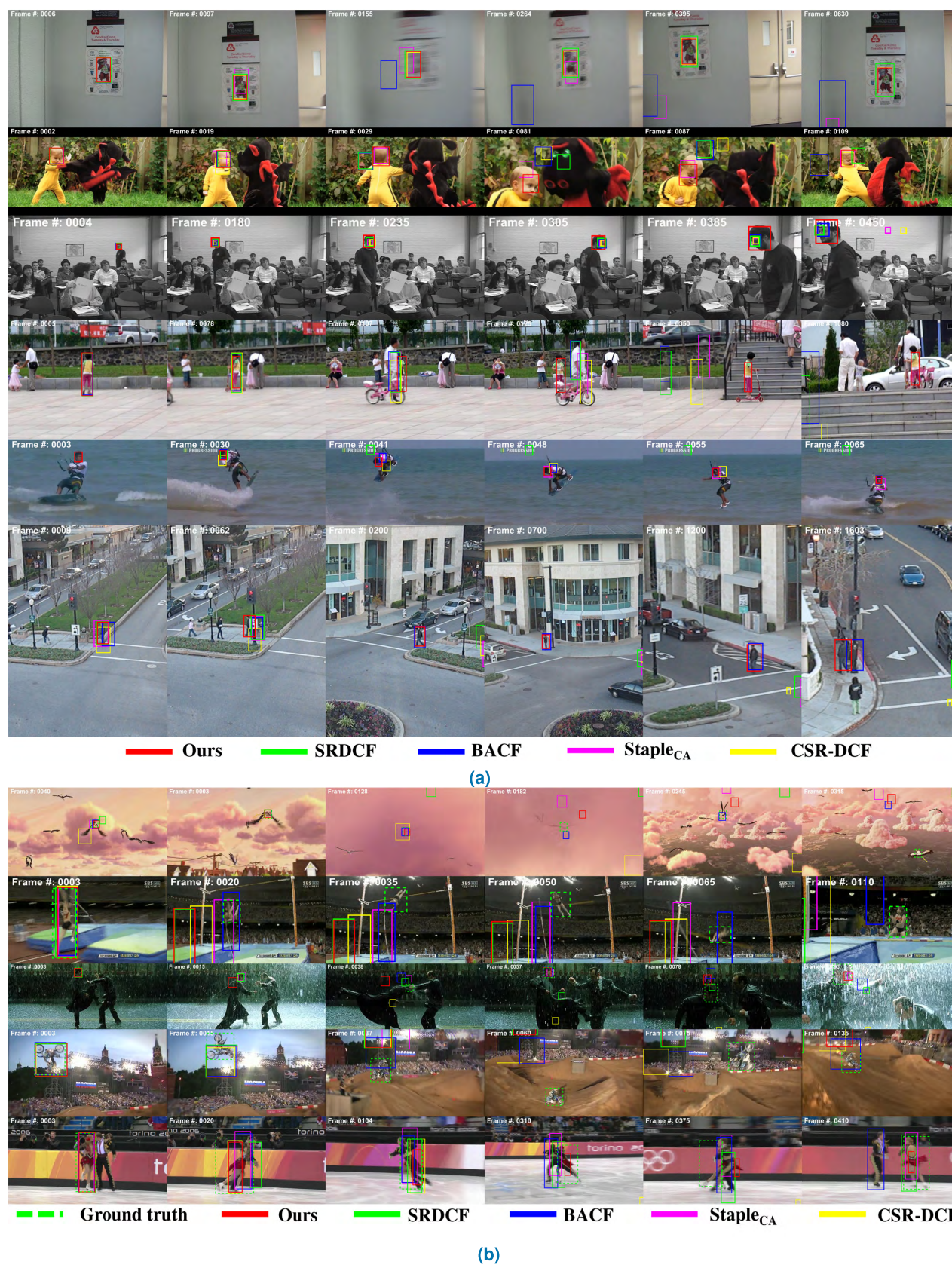
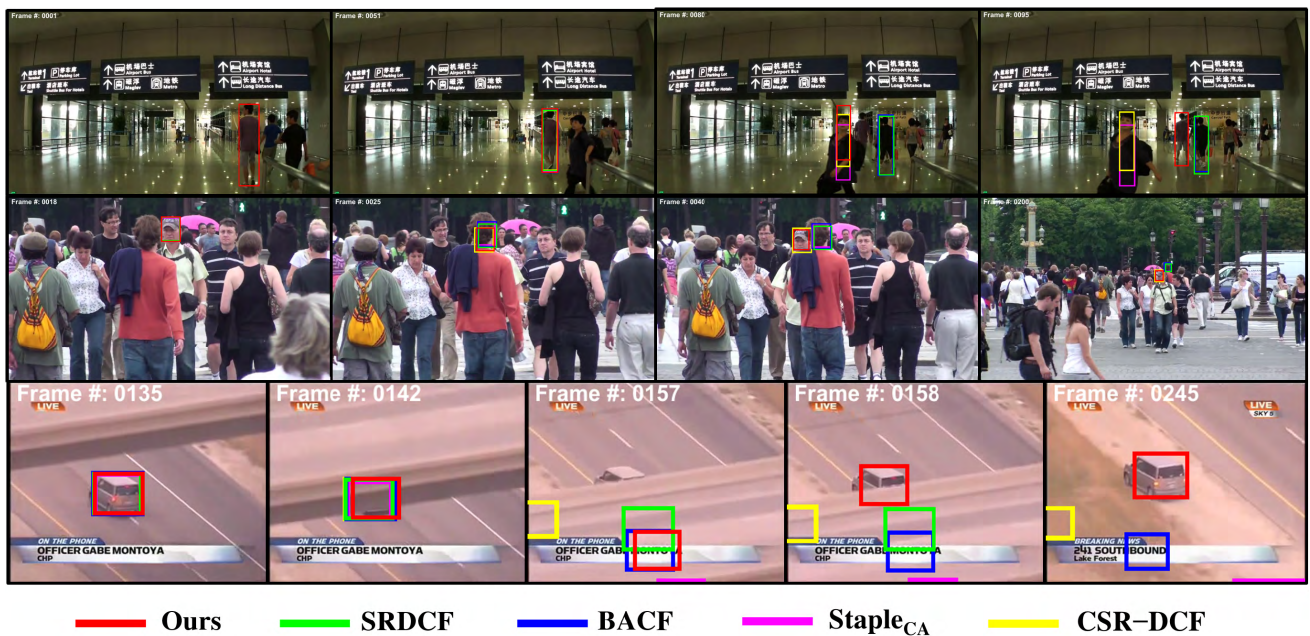


FIGURE 6. Qualitative Results on OTB-100. (a) Tracking results. (b) Failure cases.

TABLE 2. Comparison with baseline trackers in terms of success rate, precision rate, and frame per seconds (FPS) on OTB-100, TC-128, and UAV-123 datasets.

Dataset	Evaluation Method	CFLB [19]	SRDCF [20]	MOSSE _{CA} [21]	SAMF _{CA} [21]	Staple _{CA} [21]	CSR-DCF [16]	BACF [22]	Ours
OTB-100	Success rate (%)	33.29	59.70	44.73	57.34	59.82	58.23	61.62	64.86
	Precision rate (%)	45.74	78.83	59.80	79.08	80.96	78.72	81.46	88.04
	FPS	393.58	8.90	364.26	17.86	41.70	11.81	40.64	7.29
TC-128	Success rate (%)	30.33	48.53	34.99	48.75	50.54	49.48	49.13	54.89
	Precision rate (%)	40.69	66.26	45.84	65.76	67.95	67.11	65.31	75.93
	FPS	495.23	9.89	459.69	16.31	53.05	10.92	42.08	7.56
UAV-123	Success rate (%)	36.00	46.35	34.74	42.00	45.37	44.93	45.75	47.62
	Precision rate (%)	54.99	67.64	54.60	61.42	67.23	67.47	65.33	72.20
	FPS	365.53	12.13	366.87	7.61	52.41	7.83	46.68	7.80

**FIGURE 7.** Qualitative Results on TC-128.

training samples than the object size. CFLB used normalized pixels [5] to learn the single-channel filter, while SRDCF adopted the HoG feature with a channel-dependent correlation. Danelljan *et al.* proved that SRDCF outperformed CFLB under the same condition. Detailed descriptions of each CACF method were addressed in [21]. In CSR-DCF, a concatenation of HoG and CN was used. BACF adopted HoG and channel-dependent learning. The Table 1 summarizes the main differences between the baseline trackers and ours, and Table 2 summarizes the evaluation results. According to Table 2, our method shows better performance than the other methods do.

2) COMPARISON WITH ALL OTHER TRACKERS

In this section, we compare the performance of the proposed approach with those of recent trackers including deep-trackers: baseline trackers [16], [19]–[22], Staple [9],

DSST [29], fDSST [11], HDT [40], C-COT [13], ECO [10], SiamFC [45], CFNet [41], MCPF [46], and ADNet [47]. Fig. 5 shows the success plot and precision plot on OTB-100, TC-128, and UAV-123 datasets. Trackers are ranked with success rate and precision rate.

Compared to those of non-deep-learning based trackers [9], [11], [16], [19]–[22], [29], we achieved the best results with AUC (64.86%) followed by BACF (61.62%) and Staple_{CA} (59.82%) in OTB-100, with AUC (54.89%) followed by Staple_{CA} (50.54%) and Staple (49.73%) in TC-128, and with AUC (47.62%) followed by SRDCF (46.35%) and BACF (45.75%) in UAV-123.

Compared to those of deep-feature-based trackers [10], [13], [46] and deep-neural network-based trackers [40], [41], [45], [47], we achieved promising results, even if we did not take full advantages of deep-learning framework, such as deep features with high representation power in

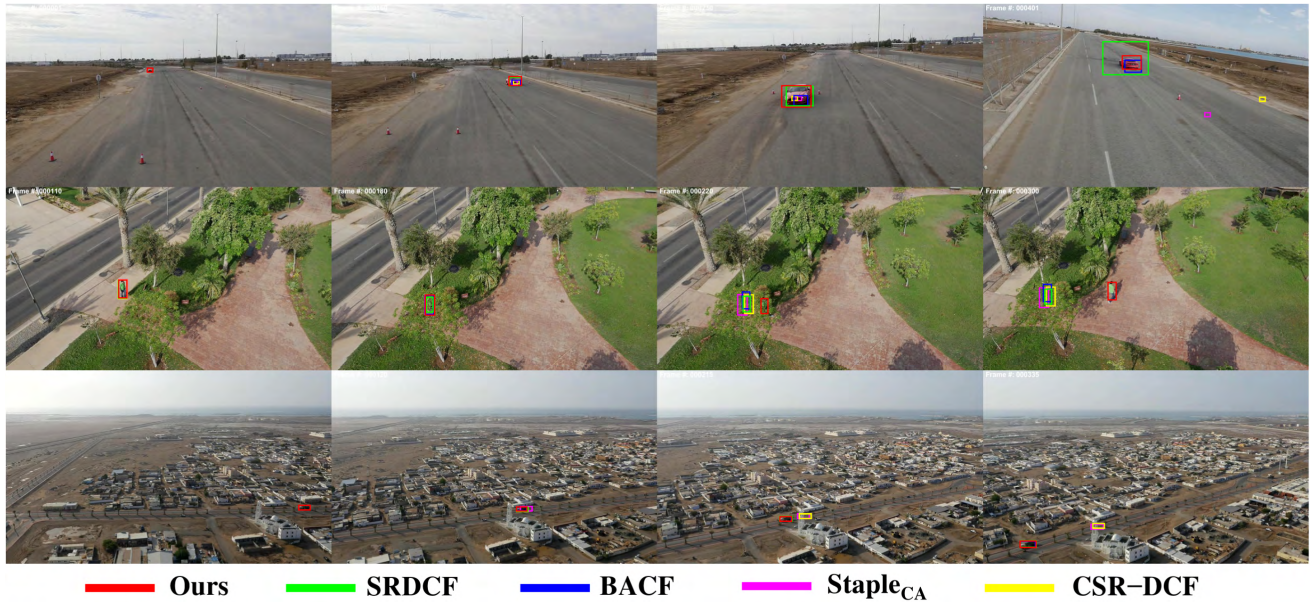


FIGURE 8. Qualitative Results on UAV-123.

classification, and also by using less number of parameters than deep-learning based method. Especially, compared with C-COT and ECO in OTB-100 dataset, our relative AUCs are -1.83% and -1.91% , but, the proposed method is 16-times and 6-times faster than C-COT and ECO, respectively. Moreover, we obtained comparative results in terms of relative AUC against ADNet ($+1.64\%$, -0.34% , $+1.36\%$), MCPF ($+2.63\%$, $+0.18\%$, -0.27%), CFNet ($+6.48\%$, $+12.83\%$, $+4.86\%$), siamFC ($+7.32\%$, $+5.70\%$, $+0.88\%$), and HDT ($+9.39\%$, $+6.84\%$, $+8.75\%$) on OTB-100, TC-128, and UAV-123.

E. QUALITATIVE RESULTS

To compare the qualitative results, we demonstrated the tracking results of the baseline trackers and ours on challenging issues, including occlusion, blur, and deformation, from OTB-100 dataset. Fig. 6a describes the bounding box of the tracking object for each method. Fig. 6a shows that our method is robust to blurring, short-term occlusion, and low resolution. Fig. 7 shows tracking results on TC-128 dataset and Fig. 8 shows tracking results on UAV-123 dataset. Still, there is room for improvement: First, it is easy to overfit the background when the object portion is smaller than the bounding box (see the second row in Fig. 6b). Another limitation is that an inaccurate filter is learned when the objects are frequently occluded (see the first and third rows in Fig. 6b). Since the online training samples, which were occluded, should not have a Gaussian-shaped response as output, it degrades the filter quality. Lastly, significant object deformation, fast scale variation, and occlusion, caused a drift problem (see the first, second, and third rows in Fig. 6b).

V. CONCLUSION

In this paper, we proposed a method to learn scalable correlation filter with a sampling operator in an enlarged window. Our scalable filter encodes a sparse frequency representation to reconstruct a larger filter. The sampling operator enables learning the scalable filter from a large window, and it suppresses spectral leakage. The proposed method achieved promising performance in terms of success and precision rates on OTB-100, TC-128, and UAV-123 datasets.

REFERENCES

- [1] B. Benfold and I. Reid, "Stable multi-target tracking in real-time surveillance video," in *Proc. CVPR*, Jun. 2011, pp. 3457–3464.
- [2] D.-N. Ta, W.-C. Chen, N. Gelfand, and K. Pulli, "SURFTrac: Efficient tracking and continuous object recognition using local feature descriptors," in *Proc. CVPR*, Jun. 2009, pp. 2937–2944.
- [3] S. Gammeter, A. Gassmann, L. Bossard, T. Quack, and L. Van Gool, "Server-side object recognition and client-side object tracking for mobile augmented reality," in *Proc. CVPR Workshop*, Jun. 2010, pp. 1–8.
- [4] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proc. CVPR*, Jun. 2009, pp. 983–990.
- [5] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. CVPR*, Jun. 2010, pp. 2544–2550.
- [6] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *Proc. ICCV*, Nov. 2011, pp. 263–270.
- [7] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proc. CVPR*, Jun. 2012, pp. 1822–1829.
- [8] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, Jul. 2012.
- [9] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. CVPR*, Jun. 2016, pp. 1401–1409.
- [10] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. CVPR*, Jul. 2017, pp. 6931–6939.

- [11] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Discriminative scale space tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1561–1575, Aug. 2017.
- [12] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proc. ICCV Workshop*, Dec. 2015, pp. 621–629.
- [13] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Proc. ECCV*, 2016, pp. 472–488.
- [14] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. ECCV*, 2012, pp. 702–715.
- [15] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proc. ECCV Workshop*, 2014, pp. 254–265.
- [16] A. Lukežič, T. Vojšír, L. C. Zajíč, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," in *Proc. CVPR*, Jul. 2017, pp. 4847–4856.
- [17] A. Lukežič, L. C. Zajíč, and M. Kristan, "Deformable parts correlation filters for robust visual tracking," *IEEE Trans. Cybern.*, vol. 48, no. 6, pp. 849–1861, Jun. 2018.
- [18] F. J. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," *Proc. IEEE*, vol. 66, no. 1, pp. 51–83, Jan. 1978.
- [19] H. K. Galoogahi, T. Sim, and S. Lucey, "Correlation filters with limited boundaries," in *Proc. CVPR*, Jun. 2015, pp. 4630–4638.
- [20] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. ICCV*, Dec. 2015, pp. 4310–4318.
- [21] M. Mueller, N. Smith, and B. Ghanem, "Context-aware correlation filter tracking," in *Proc. CVPR*, Jul. 2017, pp. 1387–1395.
- [22] H. K. Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proc. ICCV*, Oct. 2017, pp. 1144–1152.
- [23] A. S. Montero, J. Lang, and R. Laganière, "Scalable kernel correlation filter with sparse feature integration," in *Proc. ICCV Workshop*, Dec. 2015, pp. 587–594.
- [24] R. G. Lyons, *Understanding Digital Signal Processing*. London, U.K.: Pearson, 2004.
- [25] H. K. Galoogahi, T. Sim, and S. Lucey, "Multi-channel correlation filters," in *Proc. ICCV*, Dec. 2013, pp. 3072–3079.
- [26] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking," in *Proc. CVPR*, Jun. 2016, pp. 1430–1438.
- [27] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. ICCV*, Dec. 2015, pp. 3074–3082.
- [28] 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.
- [29] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. BMVC*, Sep. 2014, pp. 1–11.
- [30] A. S. Montero, J. Lang, and R. Laganière, "Scalable kernel correlation filter with sparse feature integration," in *Proc. ICCV Workshop*, Dec. 2015, pp. 587–594.
- [31] M. Zhang, J. Xing, J. Gao, X. Shi, Q. Wang, and W. Hu, "Joint scale-spatial correlation tracking with adaptive rotation estimation," in *Proc. ICCV Workshop*, Dec. 2015, pp. 595–603.
- [32] A. Bibi, M. Mueller, and B. Ghanem, "Target response adaptation for correlation filter tracking," in *Proc. ECCV*, 2016, pp. 419–433.
- [33] Y. Sui, Z. Zhang, G. Wang, Y. Tang, and L. Zhang, "Real-time visual tracking: Promoting the robustness of correlation filter learning," in *Proc. ECCV*, 2016, pp. 662–678.
- [34] S. Liu, T. Zhang, X. Cao, and C. Xu, "Structural correlation filter for robust visual tracking," in *Proc. CVPR*, Jun. 2016, pp. 4312–4320.
- [35] T. Liu, G. Wang, and Q. Yang, "Real-time part-based visual tracking via adaptive correlation filters," in *Proc. CVPR*, Jun. 2015, pp. 4902–4912.
- [36] J. Choi *et al.*, "Attentional correlation filter network for adaptive visual tracking," in *Proc. CVPR*, Jul. 2017, pp. 4828–4837.
- [37] J. Choi, H. J. Chang, J. Jeong, Y. Demiris, and J. Y. Choi, "Visual tracking using attention-modulated disintegration and integration," in *Proc. CVPR*, Jun. 2016, pp. 4321–4330.
- [38] Z. Cui, S. Xiao, J. Feng, and S. Yan, "Recurrently target-attending tracking," in *Proc. CVPR*, Jun. 2016, pp. 1449–1458.
- [39] Y. Li, J. Zhu, and S. C. H. Hoi, "Reliable patch trackers: Robust visual tracking by exploiting reliable patches," in *Proc. CVPR*, Jun. 2015, pp. 353–361.
- [40] Y. Qi *et al.*, "Hedged deep tracking," in *Proc. CVPR*, Jun. 2016, pp. 4303–4311.
- [41] 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. CVPR*, Jul. 2017, pp. 5000–5008.
- [42] 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.
- [43] P. Liang, E. Blasch, and H. Ling, "Encoding color information for visual tracking: Algorithms and benchmark," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5630–5644, Dec. 2015.
- [44] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for UAV tracking," in *Proc. ECCV*, 2016, pp. 445–461.
- [45] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *Proc. ECCV Workshop*, 2016, pp. 850–865.
- [46] T. Zhang, C. Xu, and M.-H. Yang, "Multi-task correlation particle filter for robust object tracking," in *Proc. CVPR*, Jul. 2017, pp. 4819–4827.
- [47] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Y. Choi, "Action-decision networks for visual tracking with deep reinforcement learning," in *Proc. CVPR*, Jul. 2017, pp. 1349–1358.



MINKYU LEE received the B.S. degree from the Department of Computer Engineering, Kwang-woon University, in 2013. He is currently pursuing the Ph.D. degree in electrical and electronic engineering with Yonsei University, Seoul, South Korea. His current research interests include camera calibration and visual tracking.



TAEHO KIM received the B.S. degree from the Department of Electronic and Electrical Engineering, Yonsei University, Seoul, South Korea, in 2015, where he is currently pursuing the Ph.D. degree in electrical and electronic engineering. His current research interests include image restoration and computer vision via deep learning.



YUSEOK BAN received the B.S. and Ph.D. degrees in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2010 and 2017, respectively. He is currently a Senior Researcher with the Agency for Defense Development, South Korea. His current research interests include pattern recognition and computer vision.



EUNGYEOL SONG received the B.S. degree from the Department of Electronic and Electrical Engineering, Dankook University, in 2012. He is currently pursuing the Ph.D. degree in electrical and electronic engineering with Yonsei University, Seoul, South Korea. His current research interests include hand tracking, 3-D reconstruction, and machine learning.



SANGYOUN LEE (M'04) received the B.S. and M.S. degrees in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 1987 and 1989, respectively, and the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 1999. He is currently a Professor and the Head of electrical and electronic engineering with the Graduate School and the Head of the Image and Video Pattern Recognition Laboratory,

Yonsei University. His current research interests include all aspects of computer vision, with a special focus on pattern recognition for face detection and recognition, advanced driver-assistance systems, and video codecs.

...