IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 12, 2018, accepted November 11, 2018, date of publication November 16, 2018,
date of current version December 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2881723

Visual Tracking Based on Correlation Filter and Robust Coding in Bilateral 2DPCA Subspace

B. K. SHREYAMSHA KUMAR, (Student Member, IEEE), M.N.S. SWAMY, (Fellow, IEEE), AND M. OMAIR AHMAD, (Fellow, IEEE)
Center for Signal Processing and Communications, Department of Electrical and Computer Engineering, Concordia University, Montreal, QC H3G 1M8, Canada

Corresponding author: B. K. Shreyamsha Kumar (s_bidare@ece.concordia.ca)

This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada, in part by the Regroupement Stratégique en Microsystèmes du Québec (ReSMiQ), and in part by the Ministère de l'Éducation, de l'Enseignement Supérieur et de la Recherche (MEESR) du Québec.

ABSTRACT The success of correlation filters in visual tracking has attracted much attention in computer vision due to their high efficiency and performance. However, they are not equipped with a mechanism to cope with challenging situations like scale variations, out-of-view, and camera motion. With the aim of dealing with such situations, a collaborative scheme of tracking based on the discriminative and generative models is proposed. Instead of finding all the affine motion parameters of the target by the combined likelihood of these models, the correlation filters, based on discriminative model, are used to find the position of the target, whereas 2D robust coding in a bilateral 2DPCA subspace, based on generative model, is used to find the other affine motion parameters of the target. Further, a 2D robust coding distance is proposed to differentiate the candidate samples from the subspace and used to compute the observation likelihood in the generative model. In addition, it is proposed to generate a robust occlusion map from the weights obtained during the residual minimization and a novel update mechanism of the appearance model for both the correlation filters and bilateral 2DPCA subspace is proposed. The proposed method is evaluated on the challenging image sequences available in the OTB-50, VOT2016, and UAV20L benchmark datasets, and its performance is compared with that of the state-of-the-art tracking algorithms. In contrast to OTB-50 and VOT2016, the dataset UAV20L contains long duration sequences with additional challenges introduced by both the camera motion and the view points in three dimensions. Quantitative and qualitative performance evaluations on three benchmark datasets demonstrate that the proposed tracking algorithm outperforms the state-of-the-art methods.

INDEX TERMS Visual tracking, weighted least squares, principle component analysis (PCA), bilateral 2DPCA (B2DPCA), occlusion map, correlation filters.

I. INTRODUCTION

In the last two decades, visual object tracking has seen a flurry of research due to its wide range real-life applications including vehicle navigation, robotics, human behavior analysis, action recognition, human computer interaction, video indexing and retrieval, medical imaging, security and surveillance [1], [2]. In spite of much progress in the last two decades, it still remains a challenging problem due to the complexity in target searching as well as intrinsic (e.g., pose changes, shape deformation) and extrinsic (e.g., varying viewpoints, rotation and scaling due to camera motion, illumination changes, occlusions, cluttered and moving backgrounds) object appearance variations [3], [4]. These appearance variations should be handled carefully for a reliable tracking performance for which the appearance model should adapt to the intrinsic appearance variations and be robust enough for extrinsic appearance variations.

In the literature, based on the representation scheme used to model the appearance of the object, the tracking algorithms are categorized into either generative or discriminative methods. Generative methods extract information only from the target region to model the object appearance and search for a region that is most similar to the target model. These methods are based on templates [5]–[7], local patches/fragments [8]–[12], subspace models [13]–[17] or local subspace models [18], [19]. Since the generative methods consider information from the target region alone for object appearance, they are not efficient in cluttered environments, but they achieve higher generalization with limited data. On the other hand, discriminative methods extract information not only from the target region, but also from the background to differentiate the target from the background (e.g., using boosting algorithms [20], semi-supervised learning [21] and support vector machines [22]). In contrast, the discriminative methods perform better if the training set is large due to its capability of differentiating the target from the background. The advantages of these individual methods are exploited by collaborating both generative and discriminative methods for object appearance model in [23]–[26]. Similar to these tracking algorithms, the proposed method also collaborates both these methods to improve the tracking performance, but in a different manner.

Recently, Ma *et al.* [27] proposed a visual tracking algorithm using hierarchical convolutional features for target representation, where both the semantics and fine-grained details are simultaneously exploited to handle large appearance variations. Similar to [27], Qi *et al.* [28] proposed a hedged deep tracking, where the features from different convolutional layers are convolved with the corresponding correlation filters to generate response maps that are combined by adaptive hedging to obtain the tracking result. In [29], Danelljan *et al.* proposed efficient continuous convolution operators for visual tracking, where the multiple continuous convolution filters are learned jointly using multi-resolution deep feature maps, and the object is tracked by finding the maximum of the final continuous confidence output function. Nam and Han [30] proposed a multi-domain network for visual tracking by learning generic target representations from multiple annotated sequences, considering each sequence as a separate domain, and then the object is tracked by a binary classifier constructed using shared layers of pre-trained CNN with a classification layer. The deep learning-based trackers are required to train using large-scale data specialized for visual tracking covering a wide range of variations in the target and background in order to exploit the power of deep representation. Due to lack of a-priori information of the object, most of the deep learning-based trackers [27]–[29] have adapted pre-trained deep networks, which are learned for classification task, to obtain the deep features for visual tracking. But the deep feature representation thus obtained are not effective due to the fundamental difference between classification and tracking. Moreover, the feature representations obtained by the pre-trained deep networks may be less discriminative for tracking specific objects. The combination of features from different layers for better representation of the object from that of the background increases the computational complexity and memory required to learn and update the object appearance model for tracking. This is in addition to the existing complexity and memory required in training to obtain large set of hyper parameters of the deep networks. The deep learning-based trackers perform well; however, this superior performance is achieved at the cost of excessive complexity and memory requirements. In contrast to deep learning-based methods, non deep learning-based methods do not rely on any external source of information or pre-trained model to obtain the features, but instead use the features obtained from the object itself to learn and update the object appearance model for tracking.

The correlation filter-based visual tracking has achieved great success due to its high efficiency and performance, and hence, has attracted much attention among research community. Despite its good performance, there are still some issues, such as scale variations and out-of-view problems, that need to be addressed. On the other hand, visual tracking based on particle filters perform well in occlusions, noisy and cluttered background with robust appearance model and more particle samples. However, it suffers from high computational cost as the number of particles increases, and tracking failure when the objects move rapidly with high speed and large accelerations. Also, the predicted state may not be correct if the sampled particles do not cover the object states well.

In this paper, a scheme with collaboration of discriminative and generative models for visual tracking is proposed. The position of the target is found using correlation filters, which are based on discriminative model, and other affine motion parameters of the target using 2D robust coding (2DRC) in a bilateral 2DPCA (B2DPCA) subspace, which is based on generative model. This is motivated by the idea that the discriminative capability of the tracker plays an important role while finding the location of the target rather than while finding the other affine motion parameters of the target. On the other hand, the generative capability of the tracker plays a prominent role while finding the other affine motion parameters of the target. In order to find the position of the target, it is proposed to use two correlation filters each with its own target appearance and learned coefficients' model. Then, in the generative model, 2DRC is introduced into the B2DPCA reconstruction to develop an iterative reweighted coding algorithm. The introduction of the robust coding takes care of non-Gaussian or non-Laplacian noise and avoids the effect of outliers (e.g. occluded or corrupted pixels) while computing the projection coefficients of the B2DPCA projection matrices. Also, a 2DRC distance measure is introduced to find the similarity between the candidate and the subspace, and is used to find the observation likelihood. Further, it is proposed to use the weights computed during the process of residual minimization to capture the occlusion information, thereby generating an occlusion map. In addition, a novel appearance model update mechanism is proposed for both the correlation filters and the B2DPCA subspace. Experiments conducted on three popular benchmark datasets and comparison with the state-of-the-art tracking methods bear out the competency and effectiveness of the proposed method for visual tracking.

The paper is organized as follows. Section II gives the background information and reviews the related work available in the literature. The object representation based on the B2DPCA projection matrices and 2D robust coding is explained in Section III followed by the proposed tracking algorithm in Section IV. Experimental results for the three popular benchmark datasets are discussed in Section V followed by concluding remarks in Section VI.

II. BACKGROUND AND RELATED WORK

A. CORRELATION FILTERS

Recently, correlation filters have attained significance in visual tracking due to its computational efficiency and tracking accuracy. In [31], adaptive correlation filters are learned to model the target appearance by minimizing the output sum of the squared error. Henriques *et al.* [32] have exploited the circulant structure of adjacent image patches in a kernel space based on intensity features, and HOG features [33] for visual tracking. Also, Danelljan *et al.* [34] have exploited adaptive color attributes in, and in [35] adaptive multi-scale correlation filters have been used to handle scale variations of the object. In [36], Zhang *et al.* have incorporated the circulant property of target template to improve sparse based trackers.

The Kernelized Correlation Filters (KCF) [33] employ numerous negative samples to enhance the discriminative capability of the tracking-by-detection scheme by exploiting the structure of the circulant matrix for computational efficiency. In KCF, the object appearance is modeled using correlation filter $\mathbf{w}$ trained on an image patch $\mathbf{x}$ of $M \times N$ pixels, where all the circular shifts of $\mathbf{x}_{m,n}, (m, n) \in \{0, 1, ..., M-1\} \times \{0, 1, ..., N-1\}$ are generated as training samples with Gaussian function label $\mathbf{y}_{m,n}$. The optimal weights $\mathbf{w}$ are then obtained as

$$\mathbf{w} = \arg\min_{\mathbf{w}} \sum_{m,n} |\langle \phi(\mathbf{x}_{m,n}), \mathbf{w}\rangle - \mathbf{y}_{m,n}|^2 + \xi \|\mathbf{w}\|^2 \tag{1}$$

where ϕ denotes the mapping to a kernel space and ξ is a regularization parameter. By using the Fourier transform, the objective function in Eq. (1) is minimized as $\mathbf{w} = \sum_{m,n} \boldsymbol{\alpha}_{m,n}\phi(\mathbf{x}_{m,n})$, and the coefficient $\boldsymbol{\alpha}$ is given by

$$\boldsymbol{\alpha} = \mathcal{F}^{-1}\left(\frac{\mathcal{F}(y)}{\mathcal{F}\left(\langle\phi(\mathbf{x}), \phi(\mathbf{x})\rangle\right) + \xi}\right) \tag{2}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote, respectively, the Fourier transform and its inverse. The learned coefficients $\hat{\boldsymbol{\alpha}}$ and the target appearance model $\hat{\mathbf{x}}$ along with an image patch $\mathbf{z}$ cropped out in the new frame are used to find the response map as

$$\bar{\mathbf{y}} = \mathcal{F}^{-1}\left(\mathcal{F}(\hat{\boldsymbol{\alpha}}) \odot \mathcal{F}\left(\langle\phi(\mathbf{z}), \phi(\hat{\mathbf{x}})\rangle\right)\right) \tag{3}$$

where $\odot$ is the Hadamard product. The maximal value of the response map $\bar{\mathbf{y}}$ gives the position of the target.

B. PCA SUBSPACE

In incremental visual tracking (IVT) [13], the object is represented by a low dimensional PCA subspace, which is learned and updated efficiently to adapt the appearance variations of the object. Further, incremental B2DPCA has been used for object appearance model in visual tracking based on maximum likelihood estimation (MLE) [37]. Wang *et al.* [3] and Wang and Lu [16] have exploited the strength of both the subspace and sparse representations by introducing l_1 regularization into the PCA and B2DPCA reconstruction, respectively. Even though the subspace-based trackers [13], [37] are robust to in-plane rotation, illumination variation, scale change and pose change, they are sensitive to partial occlusion due to their underlying assumption that the residual is Gaussian distributed with small variances. This is not valid as the residual cannot be modeled with small variances during partial occlusion. On the other hand, if the residual is assumed to be Laplacian distributed, then it aims to handle outliers. To account for non-Gaussian or non-Laplacian residual, 2D robust coding along with the B2DPCA subspace for object representation for visual tracking is presented in the next section.

III. OBJECT REPRESENTATION BASED ON B2DPCA PROJECTION MATRICES AND 2D ROBUST CODING

In this paper, the strengths of both the B2DPCA subspace learning and the 2D robust coding are exploited for object appearance modeling. Weighted least squares are introduced into B2DPCA reconstruction, thus avoiding the very complex l_1-norm constraint on the projection coefficients. The object appearance is modeled by two separate B2DPCA projection matrices as [38]

$$\bar{\mathbf{Y}} = \mathbf{U}\mathbf{Z}\mathbf{V}^T \tag{4}$$

where $\bar{\mathbf{Y}} \in \mathbb{R}^{d_l \times d_r} = \mathbf{Y} - \boldsymbol{\mu}$ represents the centered image observation matrix, $\boldsymbol{\mu} \in \mathbb{R}^{d_l \times d_r}$ represents mean matrix, $\mathbf{U} \in \mathbb{R}^{d_l \times k_l}$ and $\mathbf{V} \in \mathbb{R}^{d_r \times k_r}$ represent orthogonal left- and right-projection matrices, respectively, $\mathbf{Z} \in \mathbb{R}^{k_l \times k_r}$ denotes the projection coefficients, $d_l \times d_r$ the size of the observation matrix, and k_l and k_r are the number of B2DPCA left- and right-projection basis vectors, respectively. Given a set of image observations $\mathcal{Y} = \{\mathbf{Y}_1, ..., \mathbf{Y}_K\}$, the projection matrices $\mathbf{U}, \mathbf{V}$ are computed using [37], and then, the projection coefficient is computed as $\mathbf{Z}_i = \mathbf{U}^T\bar{\mathbf{Y}}_i\mathbf{V}$. As the target templates are coherent in [6] and [7], the coding coefficients should be sparse and hence, there is a requirement of l_1-norm constraint on the coding coefficients. But in the proposed method, the projection coefficients are not sparse due to the orthogonality of the B2DPCA projection matrices and hence, it is not required to impose complex l_1-norm constraint on the projection coefficients. This is in contrast to [39], where the unnecessary complex l_1-norm constraint is imposed on the projection coefficients in spite of using the B2DPCA projection matrices.

Expressing the left- and right-projection matrices as $\mathbf{U} = [\mathbf{u}_1; \mathbf{u}_2; ...; \mathbf{u}_{d_l}]$ and $\mathbf{V} = [\mathbf{v}_1; \mathbf{v}_2; ...; \mathbf{v}_{d_r}]$ respectively, where the vectors $\mathbf{u}_i$ and $\mathbf{v}_j$ are the i-th and j-th rows of $\mathbf{U}$ and $\mathbf{V}$, respectively, and denoting the coding residual matrix as $\mathbf{E} = \bar{\mathbf{Y}} - \mathbf{U}\mathbf{Z}\mathbf{V}^T$, each element of the residual matrix $\mathbf{E}$ is written as $e_{ij} = \bar{y}_{ij} - \mathbf{u}_i\mathbf{Z}\mathbf{v}_j^T$. Assume that the residuals $e_{11}, ..., e_{d_l d_r}$ are independently and identically distributed (i.i.d) according to some probability density function $f_\theta(e_{ij})$, where θ denotes the parameter set that characterizes the probability distribution. Then, maximizing the likelihood function $L_\theta(e_{11}, ..., e_{d_l d_r}) = \prod_{j=1}^{d_r}\prod_{i=1}^{d_l} f_\theta(e_{ij})$ is equivalent to minimizing the objective function: $-\ln L_\theta(e_{11}, ..., e_{d_l d_r}) = \sum_{j=1}^{d_r}\sum_{i=1}^{d_l} \rho_\theta(e_{ij})$, where $\rho_\theta(e_{ij}) = -\ln f_\theta(e_{ij})$. From this discussion, MLE of $\mathbf{Z}$, referred to as 2D robust coding (2DRC), can be formulated as the following minimization problem

$$\min_{\mathbf{Z}} \sum_{j=1}^{d_r}\sum_{i=1}^{d_l} \rho_\theta(e_{ij}) = \min_{\mathbf{Z}} \sum_{j=1}^{d_r}\sum_{i=1}^{d_l} \rho_\theta(\bar{y}_{ij} - \mathbf{u}_i\mathbf{Z}\mathbf{v}_j^T) \qquad (5)$$

Now, MLE of $\mathbf{Z}$ can be obtained by solving Eq. (5), but the problem is as to how to find the distribution ρ_θ (or f_θ). Explicitly taking f_θ as Gaussian or Laplacian distribution is simple, but not effective during occlusion. Based on the assumptions on ρ_θ specified in [40], the above minimization problem is transformed into a weighted least squares (WLS) problem, given by

$$\min_{\mathbf{Z}} \|\mathbf{A}^{\frac{1}{2}} \odot (\bar{\mathbf{Y}} - \mathbf{U}\mathbf{Z}\mathbf{V}^T)\|_F^2 \qquad (6)$$

where $\mathbf{A} \in \mathbb{R}^{d_l \times d_r}$ is a weight matrix to model different types of noise, and its element a_{ij} is the weight assigned to each pixel of the observed image $\mathbf{Y}$ depending on the value of the residual e_{ij}, and $\odot$ is the Hadamard product. Since the weight matrix $\mathbf{A}$ is unknown and needs to be estimated, WLS in Eq. (6) is a local approximation of RC in Eq. (5). Therefore, the RC minimization procedure can be converted to an iterative coding problem with $\mathbf{A}$ being updated using the residuals in the previous iteration. Since the distribution ρ_θ is unknown, it is difficult to find the weight matrix $\mathbf{A}$. Thus, a logistic function given by

$$a_{ij} = \frac{\exp\left(\delta[\beta - e_{ij}^2]\right)}{1 + \exp\left(\delta[\beta - e_{ij}^2]\right)}, \qquad (7)$$

where δ controls the decreasing rate from 1 to 0, and β controls the location of the demarcation point, is chosen as the weight function, as it satisfies the following properties: (1) weight assigned to each pixel of the observed image $\mathbf{Y}$ depends on the corresponding value of the residual $\mathbf{E}$ and (2) the weight function has higher capability to classify inliers and outliers [40]. This weight function is bounded in [0,1] and adaptively assigns low weights to the outliers (usually the pixels with large residuals) to reduce their effect on the estimation of the projection coefficients $\mathbf{Z}$ so that the sensitivity to outliers can be greatly reduced. Even though the methods in [39], [41], and [42] use robust sparse coding, they differ from the proposed method in a number of ways. The methods in [41] and [42] use a target template-based appearance model, and hence, use l_1-norm constraint on the coding coefficients, and are computationally complex. Also, they differ in the way the observation model is updated. Even though the appearance model of [39] is based on B2DPCA, it imposes l_1-norm constraint on the projection coefficients thereby increasing the computational complexity. Further, its weight function and the observation model update mechanism are different from that of the proposed method.

The minimization problem in (6) can be solved by estimating the projection coefficients $\mathbf{Z}_{opt}$ and the weight matrix $\mathbf{A}$ iteratively using Eqs. (8) and (7), respectively, and is referred to as iteratively reweighted coding (IRC) algorithm, which is summarized in Algorithm 1.

$$\mathbf{Z}_{opt} = \mathbf{U}^T(\mathbf{A} \odot \bar{\mathbf{Y}})\mathbf{V} \qquad (8)$$

The IRC algorithm is terminated when the following criterion is satisfied:

$$\frac{\|\mathbf{A}^k - \mathbf{A}^{k-1}\|_F}{\|\mathbf{A}^{k-1}\|_F} < \psi\,, \qquad (9)$$

where ψ is a small positive scalar constant.

Algorithm 1 IRC Algorithm for Computing $\mathbf{Z}_{opt}$ and $\mathbf{A}$

```
Input: Centered image observation matrix $\bar{\mathbf{Y}}$, left- and right-projection matrices $\mathbf{U}$ and $\mathbf{V}$, previous weight matrix $\mathbf{A}_{t-1}$ corresponding to the tracking result at time $t-1$, constants $\delta$ and $\beta$
Initialize $k = 0$ and $\mathbf{A}^k = \mathbf{A}_{t-1}$
Compute basis coefficients $\mathbf{Z}^k = \mathbf{U}^T(\mathbf{A}^k \odot \bar{\mathbf{Y}})\mathbf{V}$
Iterate
    $k \leftarrow k + 1$
    Compute residual $\mathbf{E}^k = \bar{\mathbf{Y}} - \mathbf{U}\mathbf{Z}^{k-1}\mathbf{V}^T$
    Compute the weights using
    $a_{ij}^k = \frac{\exp\left(\delta[\beta - (e_{ij}^k)^2]\right)}{1 + \exp\left(\delta[\beta - (e_{ij}^k)^2]\right)};\ i = 1, 2, .., d_l;\ j = 1, 2, .., d_r$

    Recompute $\mathbf{Z}^k = \mathbf{U}^T(\mathbf{A}^k \odot \bar{\mathbf{Y}})\mathbf{V}$
Until convergence or termination
Output: Basis coefficients $\mathbf{Z}_{opt}$, weight matrix $\mathbf{A}$
```

IV. PROPOSED TRACKING ALGORITHM

Most of the collaborative methods [23]–[26] find all the affine motion parameters of the target by combining the individual scores of both the generative and discriminative models, whereas the proposed method uses the discriminative model to find the target location (x_t, y_t) and the particle filter-based generative model for the remaining affine parameters of the target such as scale and aspect ratio. This is based on the intuition that the discriminative capability of the tracker plays a prominent role while finding the location of the target rather than while finding the other affine motion parameters of the target. On the other hand, the generative capability of the tracker plays a prominent role while finding the other affine motion parameters of the target. The proposed method of tracking is summarized in Algorithm 2 and its block diagram is shown FIGURE 1.

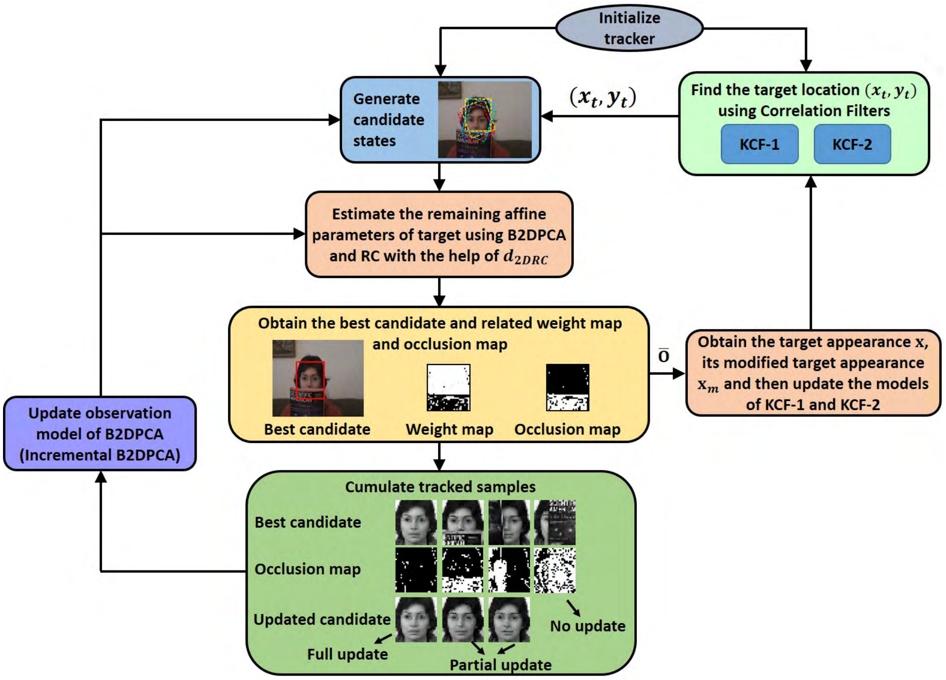

Initialize tracker
Generate candidate states
(x_t, y_t)
Find the target location (x_t, y_t) using Correlation Filters
KCF-1
KCF-2
Estimate the remaining affine parameters of target using B2DPCA and RC with the help of d_{2DRC}
Obtain the best candidate and related weight map and occlusion map
Best candidate
Weight map
Occlusion map
$\bar{o}$
Obtain the target appearance x, its modified target appearance x_m and then update the models of KCF-1 and KCF-2
Update observation model of B2DPCA (Incremental B2DPCA)
Cumulate tracked samples
Best candidate
Occlusion map
Updated candidate
No update
Full update
Partial update

FIGURE 1. Block diagram of the proposed method.

A. TARGET LOCATION ESTIMATION USING CORRELATION FILTERS

In this paper, it is proposed to use two correlation filters, KCF-1 and KCF-2, each with its own target appearance $\hat{\mathbf{x}}_k$ and learned coefficients model $\hat{\boldsymbol{\alpha}}_k$, where $k = 1, 2$ indicates which correlation filter the model belongs to. An image patch $\mathbf{z}^t$ of new window size, which is estimated in the previous frame $t-1$, is cropped out from the previous target position (x_{t-1}, y_{t-1}) in the current frame t and then resized to the initial window size in order to preserve the consistency of the object representation in the scale space. This patch along with their respective models $\hat{\mathbf{x}}_k^{t-1}$ and $\hat{\boldsymbol{\alpha}}_k^{t-1}$ is used to find the response maps $\overline{\mathbf{y}}_k^t$ of the two correlation filters using Eq. (3). The resulting response maps $\overline{\mathbf{y}}_k^t$ are energy normalized to scale the peak value according to the total energy in the respective response map. This helps to normalize low/high peak values when the entire response map is low/high due to the image characteristics such as illumination [43]. Finally, the location of the maximum value of the response map $\widetilde{\mathbf{y}}^t$ that is computed employing Eq. (10) is used to find the position of the target (x_t, y_t).

$$\widetilde{\mathbf{y}}^t = \begin{cases} \overline{\mathbf{y}}_1^t, & \text{if } \max(\overline{\mathbf{y}}_1^t) > \max(\overline{\mathbf{y}}_2^t) \\ \overline{\mathbf{y}}_2^t, & \text{otherwise} \end{cases} \qquad (10)$$

B. TARGET STATE ESTIMATION USING B2DPCA AND 2DRC

In the generative model of the proposed method, the affine parameters are estimated using a Markov model with hidden state variables [44]. Let $\mathbf{s}_t$ denote a state variable describing the affine motion parameters of a target at time t. Given a set of image observations $\mathcal{Y}_t = \{\mathbf{Y}_1, ..., \mathbf{Y}_t\}$ at time t, the posterior probability is inferred recursively by the Bayesian theorem:

$$p(\mathbf{s}_t|\mathcal{Y}_t) \propto p(\mathbf{Y}_t|\mathbf{s}_t) \int p(\mathbf{s}_t|\mathbf{s}_{t-1})\, p(\mathbf{s}_{t-1}|\mathcal{Y}_{t-1})\, d\mathbf{s}_{t-1} \qquad (11)$$

where $p(\mathbf{s}_t|\mathbf{s}_{t-1})$ represents the dynamic model, and $p(\mathbf{Y}_t|\mathbf{s}_t)$ represents the observation model. In this work, an affine transformation with four parameters is adopted to model the target state $\mathbf{s}_t = (\theta_t, s_t, \alpha_t, \phi_t)$, where $\theta_t, s_t, \alpha_t, \phi_t$ denote the rotation angle, scale, aspect ratio and skew direction at time t, respectively. The dynamic model describes the target motion between two consecutive frames and is modeled by Gaussian distribution assuming the affine parameters to be independent, i.e., $p(\mathbf{s}_t|\mathbf{s}_{t-1}) = \mathcal{N}(\mathbf{s}_t; \mathbf{s}_{t-1}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ denotes a diagonal covariance matrix whose elements are the variances of the affine parameters. These four affine parameters of the target state $\mathbf{s}_t$ along with location parameters (x_t, y_t) obtained from KCF for the current frame are used to crop a sub-image from the current frame and then normalized to the size $w \times h$. The dynamic model randomly selects M samples of the state variable $\mathbf{s}_t$ given the state at $t-1$, which are used to generate the target candidates $\mathbf{Y}_t^m$, where $m = 1, 2, ..., M$. The optimal state of the tracked target $\mathbf{s}_t$ is determined by the following MAP estimation:

$$\hat{\mathbf{s}}_t = \arg\max_{\mathbf{s}_t^m} p(\mathbf{Y}_t^m|\mathbf{s}_t^m)\, p(\mathbf{s}_t^m|\mathbf{s}_{t-1}), \qquad m = 1, 2, ..., M \qquad (12)$$

where $\mathbf{s}_t^m$ denotes the m-th sample of the state $\mathbf{s}_t$, and $\mathbf{Y}_t^m$ represents the image sample observed by $\mathbf{s}_t^m$.

For some of the vision applications, like visual tracking, in addition to an accurate estimation of the coefficients, it also requires a distance metric to find the similarity between a noisy observation and the dictionary or the subspace [13], [45], [46]. In general, the distance metric is inversely proportional to the maximum joint likelihood with respect to the coefficient $\mathbf{Z}$ [13], [46],

$$d(\mathbf{Y}; \mathbf{U}, \mathbf{V}, \boldsymbol{\mu}) \propto -\ln \max_{\mathbf{Z}} p(\mathbf{Y}, \mathbf{Z}) \propto -\ln \max_{\mathbf{Z}} p(\mathbf{Y}|\mathbf{Z})\, p(\mathbf{Z})$$

Assuming a uniform prior, the distance metric is written as

$$d(\mathbf{Y}; \mathbf{U}, \mathbf{V}, \boldsymbol{\mu}) \propto -\ln \max_{\mathbf{Z}} \exp\left(-\frac{1}{2}\|\overline{\mathbf{Y}} - \mathbf{U}\mathbf{Z}\mathbf{V}^T\|_F^2\right)$$

This distance metric is sensitive to occlusion/outliers as it considers the occluded/outlier pixels for the similarity measurement. In order to make the distance metric robust to occlusion/outliers, it should give less importance to the reconstruction error due to the occluded pixels or outliers, and hence, in this paper, a new 2DRC distance, defined as

$$d_{2DRC}(\mathbf{Y}; \mathbf{U}, \mathbf{V}, \boldsymbol{\mu}) = \|\mathbf{A}^{\frac{1}{2}}\, \mathbf{E}\|_F^2 + \lambda \|1 - \mathbf{A}^{\frac{1}{2}}\|_F^2 \qquad (13)$$

where $\mathbf{E} = \overline{\mathbf{Y}} - \mathbf{U}\mathbf{Z}\mathbf{V}^T$ and λ is a penalty constant, is proposed.

In visual tracking based on Bayesian inference framework, the confidence of each particle is given by its observation likelihood, and in the proposed method, it is defined as

$$p(\mathbf{Y}_t^m|\mathbf{s}_t^m) = \exp\left(-\frac{1}{\gamma}\, d_{2DRC}(\mathbf{Y}_t^m; \mathbf{U}, \mathbf{V}, \boldsymbol{\mu})\right) \qquad (14)$$

where γ is a constant. As the observation likelihood in the proposed method considers the distance metric $d_{2DRC}(\mathbf{Y}; \mathbf{U}, \mathbf{V}, \boldsymbol{\mu})$, the effect of occlusion/outliers on the likelihood is reduced thereby making the likelihood robust to occlusion/outliers.

Finally, the observation models are adapted to handle the appearance change of the target by incrementally updating both the KCF models $(\hat{\mathbf{x}}_1, \hat{\boldsymbol{\alpha}}_1, \hat{\mathbf{x}}_2, \hat{\boldsymbol{\alpha}}_2)$, and the B2DPCA subspace model $(\mathbf{U}, \mathbf{V}, \boldsymbol{\mu})$, as discussed in the next subsection.

C. OBSERVATION MODEL UPDATE

The update of the observation model is very much essential to handle the appearance variations of the object, but the update with imprecise samples will cause tracking drift due to the model degradation. Therefore, the imprecise samples should be avoided during the model update.

1) B2DPCA

The appearance variations of the object are handled by incrementally updating the B2DPCA projection matrices $\mathbf{U}$ and $\mathbf{V}$, and the mean $\boldsymbol{\mu}$. In order to avoid update with imprecise samples having occlusion/outliers, it is very important to extract the occlusion/outliers information from the tracking results. As the occluded/outlier pixels have low weights in the proposed method, the occlusion/outliers information is extracted from the weights. This is unlike in SPT [3], where it is extracted from the coefficients of the trivial templates. In the proposed method, a pixel is considered either noisy or occluded, if the corresponding weight $a_{ij} < 0.5$ while generating a binary occlusion map $\mathbf{O}$. Using the weight matrix $\mathbf{A}$, the occlusion map $\mathbf{O}$, with an entry of unity indicating outlier and an entry of zero indicating inlier pixel, is generated according to the following rule:

$$\mathbf{O}_{ij} = \begin{cases} 1, & \text{if } a_{ij} < 0.5 \qquad i = 1, 2, .., d_l \\ 0, & \text{otherwise} \qquad j = 1, 2, .., d_r \end{cases} \qquad (15)$$

Usually the occluded region is a large connected area compared to that of random noises or object appearance variations, which are comparatively very small. Hence, to retain the large connected area, and to fill the small hole between the regions and to remove the small regions, morphological operations and connected component analysis are performed on the occlusion map. This updated occlusion map $\hat{\mathbf{O}}$ is used to find the occlusion ratio τ, which is the ratio of the number of ones in $\hat{\mathbf{O}}$ to the total number of elements in $\hat{\mathbf{O}}$. Now, with the help of two thresholds, τ_1 and τ_2, the occlusion ratio τ is used to decide whether the tracked sample is utilized fully, or partially, or not utilized at all, in the observation model update. In the absence of occlusion (when $\tau < \tau_1$), the tracked sample is used directly for the model update (full update). During a partial occlusion (when $\tau_1 < \tau < \tau_2$), the occluded pixels in the tracked sample are replaced with the corresponding pixels from the previously updated mean $\boldsymbol{\mu}$ to get an updated sample, which is free from occlusion, and is used in a model update. Otherwise, the tracked sample is not used for the model update due to severe occlusion (when $\tau > \tau_2$). These updated new observations are accumulated and used to update the observation model $(\mathbf{U}, \mathbf{V}, \boldsymbol{\mu})$ by incremental subspace learning [37].

2) CORRELATION FILTERS

In the proposed method, the model of each correlation filter consists of its own target appearance $\hat{\mathbf{x}}_k$ and the learned coefficients model $\hat{\boldsymbol{\alpha}}_k$, where $k = 1, 2$ indicates which correlation filter the model belongs to. In order to preserve the consistency of the object representation in the scale space, the optimal state $\hat{\mathbf{s}}_t$ of the tracked target, obtained from Eq. (12), is used to find the new target and window sizes, and then an image is cropped out from the current frame corresponding to the new window size and position $(\hat{x}_t, \hat{y}_t)$, and it is resized to the initial window size to obtain the target appearance $\mathbf{x}^t$. The model of the first correlation filter KCF-1 is updated by the linear interpolation, given by

$$\mathcal{F}(\hat{\boldsymbol{\alpha}}_1^t) = \begin{cases} (1-\eta)\mathcal{F}(\hat{\boldsymbol{\alpha}}_1^{t-1}) + \eta\mathcal{F}(\boldsymbol{\alpha}_1^t), & \text{if } \tau \le \tau_{KCF} \\ \mathcal{F}(\hat{\boldsymbol{\alpha}}_1^{t-1}), & \text{otherwise} \end{cases}$$
$$\mathcal{F}(\hat{\mathbf{x}}_1^t) = \begin{cases} (1-\eta)\mathcal{F}(\hat{\mathbf{x}}_1^{t-1}) + \eta\mathcal{F}(\mathbf{x}^t), & \text{if } \tau \le \tau_{KCF} \\ \mathcal{F}(\hat{\mathbf{x}}_1^{t-1}), & \text{otherwise} \end{cases} \qquad (16)$$

where $\boldsymbol{\alpha}_1^t$ is the learned coefficient obtained from Eq. (2) using the target appearance $\mathbf{x}^t$ at time t, η is the learning rate

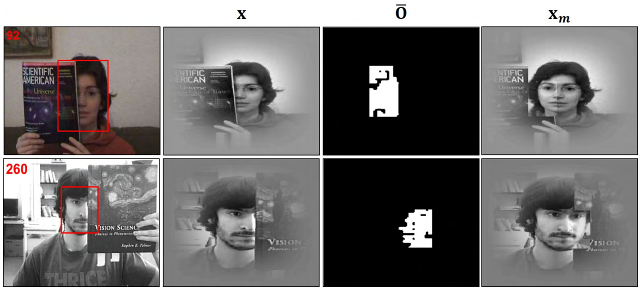


FIGURE 2. Some representative cases of *Faceocc1* (#92) and *Faceocc2* (#260) sequences showing the resized occlusion map $\overline{\mathbf{O}}$ generated from 2DRC in B2DPCA subspace and the modified target appearance $\mathbf{x}_m$.

parameter and τ_{KCF} is a threshold. Note that both the target appearance model $\hat{\mathbf{x}}_1^t$ and the learned coefficients model $\hat{\boldsymbol{\alpha}}_1^t$ of KCF-1 are not updated when the occlusion ratio $\tau > \tau_{KCF}$. This prevents the model from getting degraded during severe occlusion, and hence, the tracking drift. Further, with the help of occlusion map $\hat{\mathbf{O}}$ and the previous target appearance model $\hat{\mathbf{x}}_2^{t-1}$ of KCF-2, the modified target appearance $\mathbf{x}_m^t$ is obtained from the target appearance $\mathbf{x}^t$ as

$$\mathbf{x}_m^t = \overline{\mathbf{O}} \odot \hat{\mathbf{x}}_2^{t-1} + (1 - \overline{\mathbf{O}}) \odot \mathbf{x}^t \tag{17}$$

where $\overline{\mathbf{O}}$ is the resized occlusion map obtained from $\hat{\mathbf{O}}$ to match the matrix dimensions of $\hat{\mathbf{x}}_2^{t-1}$. Note that the modified target appearance $\mathbf{x}_m^t$ is free from the occlusion only inside the target region but not outside, since the occlusion maps $\mathbf{O}$ and $\hat{\mathbf{O}}$ are obtained only for the target region in a generative appearance model of B2DPCA as observed in FIGURE 2. Now, the models of KCF-2 are updated as

$$\begin{aligned} \mathcal{F}(\hat{\boldsymbol{\alpha}}_2^t) &= (1-\eta)\mathcal{F}(\hat{\boldsymbol{\alpha}}_2^{t-1}) + \eta\mathcal{F}(\boldsymbol{\alpha}_2^t) \\ \mathcal{F}(\hat{\mathbf{x}}_2^t) &= (1-\eta)\mathcal{F}(\hat{\mathbf{x}}_2^{t-1}) + \eta\mathcal{F}(\mathbf{x}_m^t) \end{aligned} \tag{18}$$

where $\boldsymbol{\alpha}_2^t$ is the learned coefficients obtained from the modified target appearance $\mathbf{x}_m^t$ using Eq. (2). By employing the modified target appearance $\mathbf{x}_m^t$ instead of $\mathbf{x}^t$ in both the learned coefficients $\boldsymbol{\alpha}_2^t$ computation and the target appearance model $\hat{\mathbf{x}}_2^t$ update, the models of KCF-2 are prevented from degradation due to occlusion.

V. EXPERIMENTAL RESULTS

The proposed algorithm is implemented in MATLAB and its performance is evaluated using 50, 60 and 20 challenging sequences available in the OTB-50 [4], VOT2016 [47] and UAV20L [48] datasets,[1] respectively, by following their evaluation protocols. These sequences cover most of the real-life challenging situations in object tracking, such as pose variation, complex background, motion blur due to fast movement, varying lighting conditions, scale change, low contrast and heavy occlusion. In addition to these challenges, UAV20L dataset captured by the camera mounted on UAV contains long duration sequences with large variations both in camera motion and the view points in three dimensions, resulting in out-of-view or full occlusion, in-plane and out-of-plane rotations of the object with respect to camera axes.

Algorithm 2 Proposed Tracking Algorithm

```
Input: Target object is labeled in the first frame, and its
       initial state and location are $\mathbf{s}_1$ and $(x_1, y_1)$, respectively.
An image patch $\mathbf{x}^1$ is cropped out from the target object
location $(x_1, y_1)$ in the first frame and use it to initialize
both $\hat{\mathbf{x}}_1^1$ and $\hat{\mathbf{x}}_2^1$.
Compute the coefficients $\boldsymbol{\alpha}_1^1$ using Eq. (2), and initialize
$\hat{\boldsymbol{\alpha}}_1^1$ and $\hat{\boldsymbol{\alpha}}_2^1$ with $\boldsymbol{\alpha}_1^1$.
for $t > 1$ do
    An image patch $\mathbf{z}^t$ of new window size is cropped
    out from the position $(x_{t-1}, y_{t-1})$ in frame $t$, and then
    resized to match the initial window size.
    Compute $\overline{\mathbf{y}}_k^t$ from Eq. (3) using $\hat{\mathbf{x}}_k^{t-1}$, $\hat{\boldsymbol{\alpha}}_k^{t-1}$ and $\mathbf{z}^t$,
    where $k = 1, 2$.
    Compute $\widetilde{\mathbf{y}}^t$ using Eq. (10) after energy normalization
    of the response maps $\overline{\mathbf{y}}_k^t$, and the location of its max-
    imum value is used to find the position of the target
    $(x_t, y_t)$.
    Sample $M$ candidate states $\{\mathbf{s}_t^1, \mathbf{s}_t^2, ..., \mathbf{s}_t^M\}$ from $\mathbf{s}_{t-1}$.
    Extract the candidate sample $\mathbf{Y}_t^m$ from the candidate
    state $\mathbf{s}_t^m$ and position $(x_t, y_t)$, $\forall\, m = 1, 2, ..., M$.
    For all the candidate samples $\mathbf{Y}_t^m$, compute $\mathbf{Z}_t^m$ and
    $\mathbf{A}_t^m$ according to Algorithm 1.
    Find the optimal state of the tracked target $\hat{\mathbf{s}}_t$ using
    Eqs. (13), (14) and (12).
    The observation models of KCF $(\hat{\mathbf{x}}_1, \hat{\boldsymbol{\alpha}}_1, \hat{\mathbf{x}}_2, \hat{\boldsymbol{\alpha}}_2)$ and
    B2DPCA $(\mathbf{U}, \mathbf{V}, \boldsymbol{\mu})$ are updated incrementally for
    every one and five frames, respectively, as described
    in section IV-C.
end for
Output: Target state $\hat{\mathbf{s}}_t$ and position $(x_t, y_t)$ at time $t$
```

The parameters of both of the correlation filters are set the same as in KCF [33]. In the proposed method for B2DPCA representation, each image observation is resized to 32×32 pixels, and $k_l = 4$ left- and $k_r = 4$ right-projection basis vectors are used in all the experiments. Considering the trade-off between the effectiveness in tracking and the computational efficiency, 400 particles are sampled using a particle filter and the parameter $\boldsymbol{\Sigma}$ of the particle filter is set to $\boldsymbol{\Sigma} = (0.01, 0.0, 0.005, 0)^2$. The positive scalar ψ used to terminate the IRC algorithm in Eq. (9) is set to 0.1. The penalty constant λ used in the computation of d_{2DRC} in Eq. (13) is set to 0.1. The B2DPCA observation model is incrementally updated for every 5 frames, and the occlusion ratio thresholds τ_1 and τ_2 used in B2DPCA observation model update are set to 0.1 and 0.6, respectively. The occlusion ratio threshold τ_{KCF} used in model update of KCF-1 (Eq. (16)) is set to 0.6. In both the correlation filters and B2DPCA,

[1] There are 18 and 14 sequences with more than 500 frames, and 7 and 2 sequences with more than 1000 frames in OTB-50 and VOT2016 datasets, respectively. In UAV20L dataset, there are 11 sequences with more than 2500 frames and 4 sequences with more than 4000 frames totaling 58670 frames for 20 sequences.

the cell size of 4 × 4 pixels with 9 orientations are adopted for extracting the HOG feature.

The computation of the two parameters β and δ used in the weight function (Eq. (7)) of the IRC algorithm is explained below. In order to compute the value of β, which controls the location of the demarcation point, the coding residuals e_{ij} at all locations are assumed to be in the "normal range" and follow the Gaussian distribution in the absence of occlusions in the tracked sample. But during occlusions, they will probably exceed the "normal range" at the occluded locations. Hence, by knowing the "normal range" of the residuals e_{ij} in the initial frames (from 2 to F) of the respective sequence, the value of β is computed as

$$\beta = \frac{1}{F-1}\sum_{f=2}^{F}\left[\text{mean}(\mathbf{E}_f) + c_1\,\text{std}(\mathbf{E}_f)\right]^2 \tag{19}$$

where c_1 is a constant and F is the frame number at which the number of the left- and right-projection basis vectors k_l and k_r become 4 for the first time. Note that the residual matrix $\mathbf{E}$ is vectorized while computing the mean and standard deviation in Eq. (19). The first frame is manually labeled, due to which all the elements of the residual will be zero, and hence, the first frame is not considered in the computation of β. Now, the square of the residual e_{ij} that is larger than the computed β will be considered as occluded/outlier and the value of weight a_{ij} will be less than 0.5. Further, the parameter δ, which controls the rate of the weight between 1 and 0 is computed using $\delta = c_2/\beta$, where c_2 is a constant. The constants c_1 and c_2 are set as 2 and 7, respectively, for all the sequences. In the proposed method, the IRC algorithm starts functioning after the number of the left- and right-projection basis vectors k_l and k_r become 4 for the first time. At the end of frame F, the number of the left- and right-projection basis vectors k_l and k_r are 4, and then, the parameters β and δ are computed, which are then used by the IRC algorithm from frame $F+1$ onwards for the calculation of weights in Eq. (7).

The performance of the proposed method is evaluated against several recent state-of-the-art algorithms for comparison. The algorithms considered are visual tracking via least soft-threshold squares (LSST) [46], weighted residual minimization in PCA subspace for visual tracking (WRMPCA) [14], visual tracking via bilateral 2DPCA and robust coding (B2DPCA) [17], visual tracking via discriminative low-rank learning (DLR) [49], visual tracking via weighted local cosine similarity (WLCS) [50], locally weighted inverse sparse tracker (LWIST) [51], robust object tracking via probability continuous outlier model (PCOM) [52], visual tracking by learning a deep compact image representation (DLT) [53], tracking via structured discriminative dictionary learning (DDL) [54], ACT [34], KCF [33], visual tracking via locally structured Gaussian process regression (LSGPR) [55] and discriminative low-rank tracking (DSL) [56]. Note that the codes of all the trackers are downloaded from the respective authors' website and evaluated on the OTB-50, VOT2016 and UAV20L benchmark sequences for a fair comparison with the proposed method except DDL [54], DLR [49], LSGPR [55] and DSL [56] as the codes of these trackers are not available. As the OTB-50 results of these trackers are available on the respective authors' website, they have been used to compare with that of the proposed method.

A. PERFORMANCE EVALUATION ON OTB-50

Generally, the performance of a tracker in a given frame is evaluated using two frame-based metrics, namely, overlap rate (OR) and center location error (CLE). Based on these two basic metrics, Wu *et al.* [4] and Kristan *et al.* [47] have derived other performance measures to analyze the performance of some existing trackers on their benchmark datasets, OTB-50 and VOT2016, respectively.

The performance of a tracker for a given sequence is evaluated using the *success rate* and the *precision score* on OTB-50 dataset [4]. The former is the ratio of successful frames whose OR is larger than a given threshold value to the total number of frames in a sequence, whereas the later is the percentage of frames whose CLE is less than a given threshold distance of the ground truth. By using these two metrics with multiple thresholds, two curves are obtained showing how the threshold value affects the *success rate* and the *precision score*, and are, respectively, called *success plot* and *precision plot*, for a given sequence. Further, these *success* and *precision plots* are averaged over all the sequences to obtain the overall *success* and *precision plots*, respectively. In order to quantify the overall performance of a tracker, the area under curve (AUC) of the *success plot* or the *precision plot* for the threshold of 20 pixels, is employed [4].

The proposed method is evaluated on the OTB-50 benchmark [4] consisting of 50 sequences with fully annotated attributes and compared with the state-of-the-art tracking algorithms using one-pass evaluation (OPE). In OPE, the tracker is initialized with the ground truth object location in the first frame and then allowed to run through the rest of the sequence, and the average *precision score* or *success rate* is reported at the end. Table 1 shows the performance comparison of the proposed method in terms of the *precision scores* for the threshold of 20 pixels with that of the other state-of-the-art trackers for different attributes such as illumination variation (IV), out-of-plane rotation (OPR), scale variation (SV), occlusion (Occ), deformation (Def), motion blur (MB), fast motion (FM), in-plane rotation (IPR), out-of-view (OV), background clutter (BC) and low resolution (LR). It can be observed from Table 1 that KCF-B2DPCA_HOG outperforms the other trackers in all the challenging attributes except LR, where the variant of the proposed method with gray features (KCF-B2DPCA_Gray) performs the best. The LR images suffer from lack of details due to which the efficient representation of target is not possible with HOG features resulting in inferior performance to that of gray features. The same reason holds good even for KCF with HOG and gray features, where KCF_Gray performs better than KCF_HOG does for the LR sequences. Further,

TABLE 1. Precision scores of the proposed method with that of the compared trackers for different attributes of OTB-50. (red, bold), (violet, underline) and (blue, italic) indicate first, second and third rankings, respectively.

Precision Score	IV	OPR	SV	Occ	Def	MB	FM	IPR	OV	BC	LR
KCF-B2DPCA_HOG	**0.748**	**0.760**	**0.749**	**0.794**	**0.747**	**0.686**	**0.667**	**0.732**	**0.730**	**0.753**	0.496
KCF-B2DPCA_Gray	0.427	0.514	0.514	0.508	0.483	0.454	0.419	0.499	0.352	0.481	**0.507**
KCF_HOG [33]	0.728	0.729	0.679	0.749	0.740	0.650	0.602	0.725	0.650	**0.753**	0.381
KCF_Gray [33]	0.448	0.541	0.492	0.505	0.480	0.394	0.441	0.552	0.358	0.503	0.396
ACT [34]	0.575	0.645	0.598	0.619	0.605	*0.550*	*0.480*	0.675	0.434	0.629	*0.405*
LSGPR [55]	0.521	0.571	0.586	0.627	0.579	0.312	0.336	0.535	0.455	0.570	0.391
DSL [56]	0.627	0.691	0.686	0.655	0.676	0.372	0.411	0.638	0.413	0.620	0.312
DLT [53]	0.534	0.561	0.590	0.574	0.563	0.453	0.446	0.548	0.444	0.495	0.396
DDL [54]	0.650	*0.726*	*0.693*	0.688	*0.695*	0.332	0.362	*0.679*	0.354	0.673	0.346
WRMPCA [14]	0.427	0.477	0.528	0.459	0.412	0.243	0.236	0.475	0.311	0.434	0.280
DLR [49]	*0.659*	*0.726*	0.717	*0.699*	0.691	0.420	0.458	0.672	*0.511*	*0.662*	0.341
B2DPCARC [17]	0.307	0.420	0.417	0.400	0.434	0.199	0.232	0.414	0.276	0.382	0.130
LWIST [51]	0.343	0.384	0.403	0.404	0.353	0.151	0.187	0.337	0.307	0.374	0.302
WLCS [50]	0.329	0.330	0.383	0.357	0.299	0.154	0.213	0.303	0.228	0.331	0.288
LSST [46]	0.415	0.470	0.533	0.413	0.450	0.271	0.252	0.461	0.243	0.426	0.133
PCOM [52]	0.368	0.461	0.486	0.444	0.383	0.229	0.220	0.460	0.305	0.401	0.278

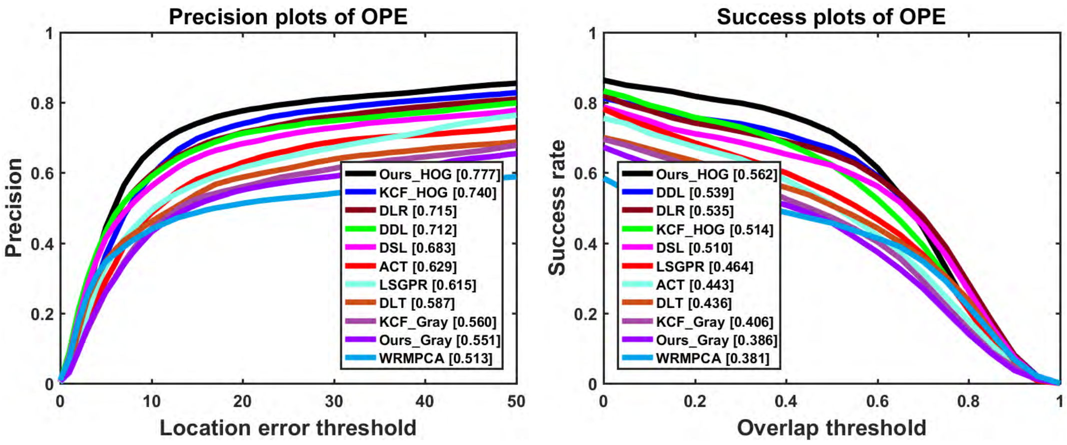


FIGURE 3. Overall performance evaluation of the proposed method on OTB-50 (50 videos with 51 target objects) using *precision* and *success plots* of OPE. The *precision score* for the threshold of 20 pixels and AUC of the *success plot* are used to rank the trackers in the respective plots and their values have been shown along with the tracker name.

KCF-B2DPCA_HOG outperforms KCF_HOG in all the attributes especially in SV, FM, OV and LR attributes by 10.3%, 10.8%, 12.3% and 30.1%, respectively.

The *precision* and *success plots* of OPE for the various trackers averaging over the OTB-50 benchmark sequences are shown in FIGURE 3. To rank the tracker, the *precision score* for the threshold of 20 pixels is used in the *precision plot*, whereas AUC is used in the *success plot*, and their values are shown along with the tracker name. From FIGURE 3, it is observed that the proposed method (KCF-B2DPCA_HOG) outperforms the state-of-the-trackers KCF_HOG, DLR, DDL, DSL, ACT, LSGPR, DLT, KCF_Gray and WRMPCA by 5%, 8.6%, 9.1% 13.7%, 23.5%, 26.3%, 32.3%, 38.7% and 51.4%, respectively, in terms of the *precision score*. Similarly in terms of the *success scores*, KCF-B2DPCA_HOG outperforms DDL, DLR, KCF_HOG, DSL, LSGPR, ACT, DLT, KCF_Gray and WRMPCA, by 4.2%, 5%, 9.3%, 10.2%,21.1%, 26.8%, 28.9%, 38.4% and 47.5%, respectively.

TABLE 2. Performance comparison of the variants of the proposed method in terms of precision score and AUC on OTB-50.

Variants/Metric	Precision Score	AUC
KCF	0.740	0.514
Proposed_KCF-1	0.758	0.550
Proposed_KCF-2	0.732	0.532
Proposed	0.777	0.562

Variants of the Proposed Method: The performance comparison of the variants of the proposed method in terms of the *precision score* and AUC on OTB-50 are given in Table 2. The methods, Proposed_KCF-1 and Proposed_KCF-2, use only one correlation filter, KCF-1 and KCF-2, respectively, to find the location of the target. It is observed from this table that the Proposed_KCF-1 performs better than KCF and Proposed_KCF-2 do in terms of the *precision score* and AUC. Further, the Proposed_KCF-2 performs better than KCF does in terms of AUC but not in terms of the *precision score*, where

TABLE 3. Accuracy rank and average overlap comparison of the proposed method with that of the compared trackers for different attributes of VOT2016. (red, bold), (violet, underline) and (blue, italic) indicate first, second and third rankings, respectively.

Accuracy	label camera motion		label empty		label illum. change		label motion change		label occlusion		label size change		Mean		Weighted mean		Pooled	
Trackers	A-Rank	Overlap	A-Rank	Overlap	A-Rank	Overlap	A-Rank	Overlap	A-Rank	Overlap	A-Rank	Overlap	A-Rank	Overlap	A-Rank	Overlap	A-Rank	Overlap
KCF-B2DPCA_HOG	**1.00**	0.48	**1.00**	**0.51**	**1.00**	*0.62*	**1.00**	0.42	**1.00**	*0.40*	**1.00**	**0.42**	**1.00**	**0.47**	**1.00**	**0.46**	**1.00**	**0.47**
KCF-B2DPCA_Gray	**1.00**	0.43	**1.00**	*0.50*	9.00	0.43	**1.00**	0.39	2.00	0.34	6.00	0.32	3.33	0.40	3.33	0.41	**1.00**	0.43
KCF_HOG [33]	**1.00**	**0.50**	**1.00**	0.49	9.00	0.44	**1.00**	*0.41*	**1.00**	**0.43**	4.00	0.36	2.83	0.44	2.83	0.45	**1.00**	0.46
KCF_Gray [33]	**1.00**	0.42	**1.00**	0.49	9.00	0.44	**1.00**	0.40	2.00	0.35	6.00	0.31	3.33	0.40	3.33	0.41	**1.00**	0.43
ACT [34]	2.00	*0.46*	**1.00**	0.50	9.00	0.44	**1.00**	**0.43**	**1.00**	0.41	*5.00*	0.35	3.17	0.43	3.17	0.44	**1.00**	0.45
WRMPCA [14]	*3.00*	0.40	**1.00**	0.49	4.00	0.52	**1.00**	0.37	*4.00*	0.31	**1.00**	0.37	*2.33*	0.41	*2.33*	0.41	**1.00**	0.42
B2DPCARC [17]	*3.00*	0.42	**1.00**	0.47	*3.00*	0.54	4.00	0.36	2.00	0.35	**1.00**	0.37	*2.33*	0.42	*2.33*	0.41	**1.00**	0.42
WLCS [50]	*3.00*	0.41	**1.00**	0.50	**1.00**	0.63	*6.00*	0.34	2.00	0.33	**1.00**	0.40	*2.33*	0.43	*2.33*	0.42	**1.00**	0.43
LWIST [51]	*3.00*	0.42	**1.00**	0.49	**1.00**	**0.65**	*6.00*	0.34	2.00	0.34	**1.00**	*0.41*	*2.33*	*0.44*	*2.33*	0.42	**1.00**	0.43
PCOM [52]	*3.00*	0.41	**1.00**	0.49	*3.00*	0.57	**1.00**	0.36	**1.00**	0.36	**1.00**	0.41	1.67	0.43	1.67	0.42	**1.00**	0.43
DLT [53]	**1.00**	0.44	**1.00**	0.50	**1.00**	0.58	**1.00**	0.39	**1.00**	0.38	**1.00**	0.42	**1.00**	0.45	**1.00**	*0.44*	**1.00**	*0.45*
LSST [46]	*3.00*	0.40	**1.00**	0.44	2.00	0.55	7.00	0.32	2.00	0.34	**1.00**	0.37	2.67	0.40	2.67	0.39	2.00	0.40

TABLE 4. Robustness rank and average failures comparison of the proposed method with that of the compared trackers for different attributes of VOT2016. (red, bold), (violet, underline) and (blue, italic) indicate first, second and third rankings, respectively.

Robustness	label camera motion		label empty		label illum. change		label motion change		label occlusion		label size change		Mean		Weighted mean		Pooled	
Trackers	R-Rank	Failures	R-Rank	Failures	R-Rank	Failures	R-Rank	Failures	R-Rank	Failures	R-Rank	Failures	R-Rank	Failures	R-Rank	Failures	R-Rank	Failures
KCF-B2DPCA_HOG	**1.00**	66.93	**1.00**	35.07	*3.00*	10.07	2.00	64.13	**1.00**	25.47	**1.00**	*33.33*	*1.50*	*39.17*	*1.50*	*47.29*	2.00	*158.60*
KCF-B2DPCA_Gray	3.00	73.87	**1.00**	36.27	2.00	9.33	**1.00**	*57.80*	**1.00**	*24.93*	*4.00*	41.20	2.00	40.57	2.00	49.79	2.00	167.53
KCF_HOG [33]	**1.00**	*68.00*	2.00	39.00	*3.00*	10.00	**1.00**	57.00	**1.00**	25.00	**1.00**	32.00	*1.50*	38.50	*1.50*	47.08	2.00	157.00
KCF_Gray [33]	**1.00**	72.00	**1.00**	**35.00**	**1.00**	7.00	**1.00**	57.00	**1.00**	24.00	2.00	36.00	1.17	38.50	1.17	47.74	**1.00**	160.00
ACT [34]	**1.00**	**62.93**	**1.00**	*36.00*	**1.00**	**5.00**	**1.00**	**47.00**	**1.00**	**23.00**	**1.00**	**26.93**	**1.00**	**33.48**	**1.00**	**41.93**	**1.00**	**140.00**
WRMPCA [14]	7.00	106.60	*7.00*	52.47	*3.00*	10.47	*7.00*	91.93	*7.00*	38.20	*4.00*	48.33	5.83	58.00	5.83	71.51	7.00	242.60
B2DPCARC [17]	8.00	126.93	8.00	62.20	5.00	10.67	8.00	100.67	8.00	43.93	9.00	59.20	7.67	67.27	7.67	83.66	8.00	278.73
WLCS [50]	9.00	164.73	12.00	89.53	*3.00*	9.13	10.00	124.13	*7.00*	46.67	9.00	60.87	8.33	82.51	8.33	105.86	12.00	355.27
LWIST [51]	8.00	131.27	10.00	81.13	*3.00*	*8.53*	8.00	115.13	*7.00*	43.07	8.00	55.27	7.33	72.40	7.33	91.28	9.00	307.53
PCOM [52]	8.00	126.80	*7.00*	57.67	2.00	8.87	*7.00*	102.27	*7.00*	41.20	7.00	53.60	6.33	65.07	6.33	81.51	8.00	273.67
DLT [53]	*4.00*	94.73	**1.00**	42.20	*3.00*	10.07	2.00	66.53	2.00	35.40	*4.00*	43.33	2.67	48.71	2.67	60.06	*6.00*	201.47
LSST [46]	9.00	135.60	10.00	77.87	5.00	11.40	8.00	115.20	*7.00*	45.60	10.00	61.47	8.17	74.52	8.17	93.11	10.00	315.20

KCF performs better. The combination of the two correlations filters, KCF-1 and KCF-2, further enhances the performance of the proposed method both in terms of the *precision score* and AUC.

B. PERFORMANCE EVALUATION ON VOT2016

In VOT2016, the performance of a tracker is analyzed using accuracy (A) and robustness (R). The accuracy is the average overlap between the predicted and ground truth bounding boxes during successful tracking periods, whereas the robustness measures the number of times the tracker fails to track. In VOT2016, whenever a tracker predicts a bounding box with zero overlap with the ground truth, a failure is detected and the tracker is re-initialized. The per-frame accuracy is obtained as an average over these runs. Averaging per-frame accuracies gives per-sequence accuracy, whereas the per-sequence robustness is computed by averaging the failure rates over different runs [47].

The average accuracy and robustness are used to evaluate the performance of the proposed method using the VOT2016 benchmark [47] consisting of 60 sequences, which are per-frame annotated with several visual attributes. Further, the tracking results are ranked according to the accuracy and robustness performance metrics, and are named as accuracy rank (A-Rank) and robustness rank (R-Rank), respectively. Table 3 shows the A-Rank and overlap comparison of KCF-B2DPCA with that of the recent state-of-the-art tracking algorithms averaging over the VOT2016 sequences having challenging situations such as camera motion, illumination change, motion change, occlusion and size change. Likewise, Table 4 shows the R-Rank and failures comparison of the proposed method (KCF-B2DPCA) averaging over the same challenging sequences. Also, the respective measures with different averaging methodologies are shown in the last six columns of these two tables. The averages of the attributes in an equal or weighted manner are denoted as mean and weighted mean, and the per-frame averaging of the super-sequence obtained by concatenating all of the sequences as pooled. Note that as the trackers with statistically equivalent results are merged while ranking, the different trackers may have the same A-Rank and R-Rank [47]. The best three results are shown in (red, bold), (violet, underline) and (blue, italic) fonts for better comparison of KCF-B2DPCA with the other state-of-the-art trackers. From Table 3, it is observed that in terms of the overlap KCF-B2DPCA_HOG ranks first for the attribute empty, size change and averages of attributes (Mean, Weighted mean and Pooled), and stands second and third for the attributes camera motion and motion change, and for the attributes illumination change and occlusion, respectively. Also, KCF-B2DPCA_Gray ranks third for the attribute empty. Further, in terms of the overlap, KCF_HOG ranks first and second for the attributes camera motion and occlusion, and for the attributes Weighted mean, Pooled, respectively. On the other hand, DLT ranks second and third for the attributes empty, size change and Mean, and for the attributes Weighted mean and Pooled, respectively. Also, ACT ranks

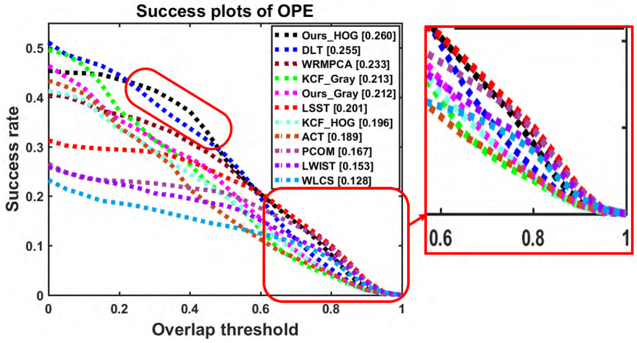


FIGURE 4. **Overall performance evaluation of the proposed method on UAV20L (20 sequences) using *success plots* of OPE. AUC of the *success plot* are used to rank the trackers and their values have been shown along with the tracker name.**

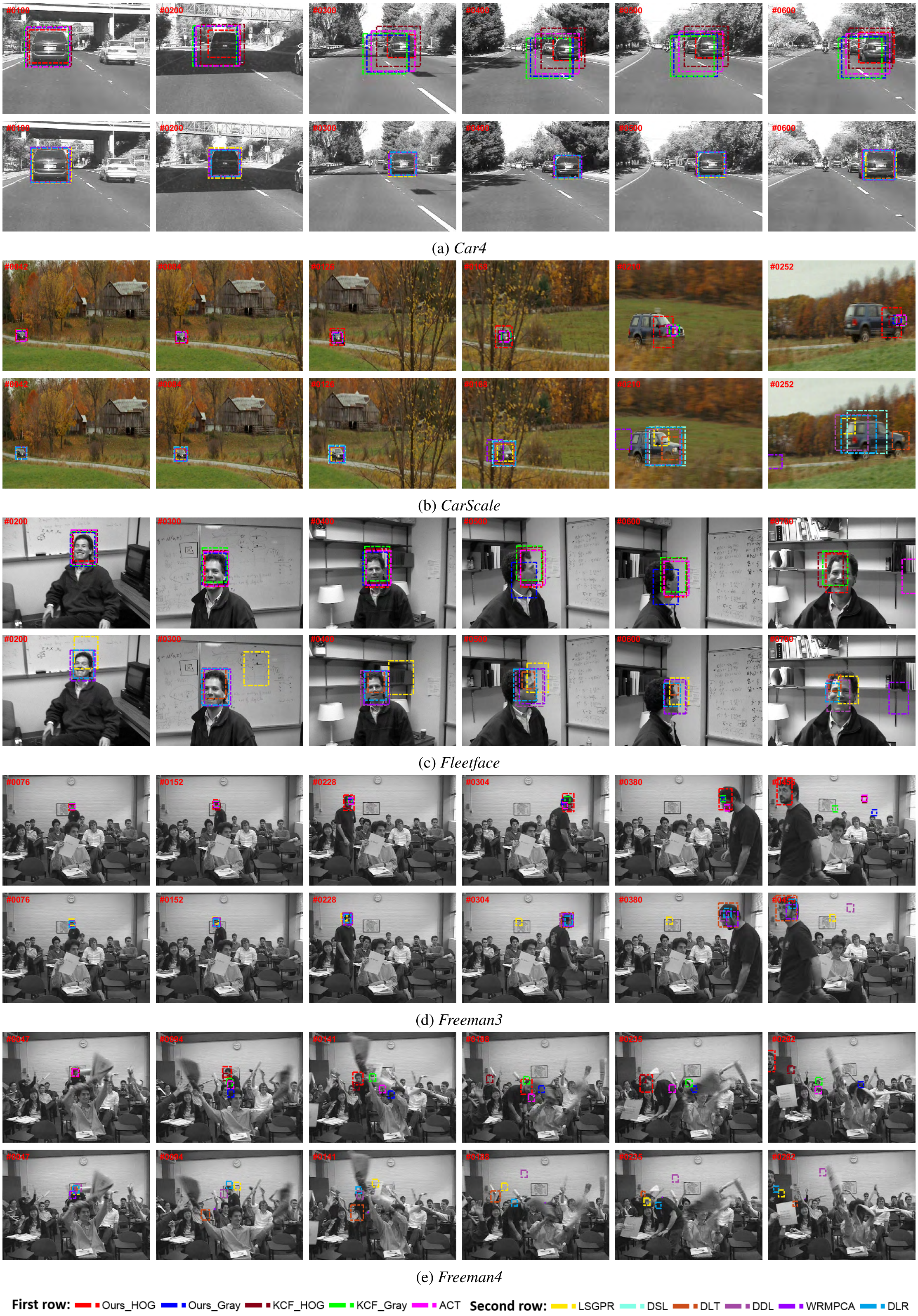
first, second and third in terms of overlap, for the attributes motion change and occlusion and for the attribute camera motion, respectively. In addition, LWIST ranks first and third for the attributes illumination change and size change and for the attribute Mean, respectively. Also, WLCS and KCF_HOG ranks second and third for the attributes illumination change and motion change, respectively. Further, it is observed from Table 4 that ACT ranks first in all the attributes in terms of failures except empty in which KCF_Gray ranks first. On the other hand, KCF_HOG ranks second in the attributes motion change, size change, Mean, Weighted mean and Pooled. Further, KCF-B2DPCA_HOG ranks second and third in the attributes camera motion and empty, and in the attributes size change, Mean, Weighted mean and Pooled, respectively. Also, KCF-B2DPCA_Gray and LWIST rank third in the attributes motion change and occlusion, and in the attribute illumination change, respectively. It is also noted that even though the overall performance of ACT is good in terms of Robustness, it is not so in terms of Accuracy. Similarly, even though the overall performance of the proposed method is inferior in terms of Robustness, it outperforms in terms of Accuracy.

C. PERFORMANCE EVALUATION USING UAV20L

As in OTB-50 evaluation, the *success plot* is used to evaluate the performance of the proposed method on UAV20L. The *success plot* of OPE for various trackers averaged over UAV20L benchmark sequences is shown in FIGURE 4. From FIGURE 4, it is observed that the proposed KCF-B2DPCA_HOG exhibits the best performance in terms of AUC followed by DLT, WRMPCA, KCF_Gray, KCF-B2DPCA_Gray, LSST, KCF_HOG, ACT, PCOM, LWIST and WLCS in the that order. Even though the *success rate* of DLT is more than that of KCF-B2DPCA_HOG for overlap threshold < 0.2, the latter method performs better than DLT does, when the overlap threshold approaches unity (this can be observed in the zoomed window of FIGURE 4). Even though KCF-B2DPCA_HOG performs better for challenges that are considerably difficult, it loses the target, as the other trackers do, for challenges that are extremely difficult or for sequences where the target undergoes several changes simultaneously, especially in long duration tracking. Further from the *success plots* of OPE in FIGURE 3 and FIGURE 4, it is observed that the performance of all the trackers deteriorates for UAV20L dataset compared to that with OTB-50 dataset due to the additional complexity and challenges in the sequences of UAV20L that are introduced by the camera motion and view point variations in three dimensions along with long duration of the sequences. Hence, tracking on the sequences of UAV20L is a difficult challenge and there is much room for improvement desired in long duration tracking.

D. QUALITATIVE EVALUATION

For qualitative evaluation of the trackers, some tracking results on a subset of the OTB-50 benchmark sequences are obtained, and shown in FIGURE 5 and FIGURE 6; FIGURE 5 shows the results for the five sequences, *Car4*, *CarScale*, *Fleetface*, *Freeman3* and *Freeman4*, whereas FIGURE 6 the results for the sequences *Jogging-1*, *Lemming*, *Singer1*, *Suv* and *Trellis*. The tracking results of the various trackers on the six exemplar image frames are shown for each selected sequence. The six frames are selected at regular intervals without any bias. For each of the sequences in FIGURE 5 and FIGURE 6, the first row shows the tracking results of the proposed methods, KCF-B2DPCA_HOG and KCF-B2DPCA_Gray, with that of KCF_HOG, KCF_Gray and ACT, and the second row shows the results of the remaining trackers, LSGPR, DSL, DLT, DDL, WRMPCA and DLR. The proposed KCF-B2DPCA_HOG tracker successfully tracks the target in all the frames of the *Car4*, *CarScale*, *Fleetface*, *Freeman3*, *Freeman4*, *Jogging-1*, *Singer1*, *Suv* and *Trellis* sequences, which contain most of the real-time challenges such as pose change, partial occlusion, illumination change, scale change and out-of-plane rotation. This indicates the strong capabilities of the proposed method in handling these challenges. Even though KCF-B2DPCA_HOG performs better for the challenges that are considerably difficult, it loses the target for challenges that are extremely difficult or for the sequences where the target undergoes several changes simultaneously. This can be observed in the last image of the *Lemming* sequence, one of the longest and most challenging sequences, where the target undergoes severe occlusion, scale change, out-of-plane rotation, motion blur, fast motion either individually or simultaneously. In addition to KCF-B2DPCA_HOG, other methods have also failed in the *Lemming* sequence but at different frames of the sequence. For example, in frame #0666, all the trackers, DDL, DLR, ACT, KCF_HOG, KCF_Gray, LSGPR, DSL, WRMPCA and KCF-B2DPCA_Gray, have failed except KCF-B2DPCA_HOG and DLT, whereas in #0888, all the trackers have started to track the target again except DDL, LSGPR, DLT and WRMPCA. So, none of the trackers has tracked the target through the entire *Lemming* sequence successfully. Even though some trackers fail


First row: Ours_HOG Ours_Gray KCF_HOG KCF_Gray ACT
Second row: LSGPR DSL DLT DDL WRMPCA DLR

(a) *Car4*

(b) *CarScale*

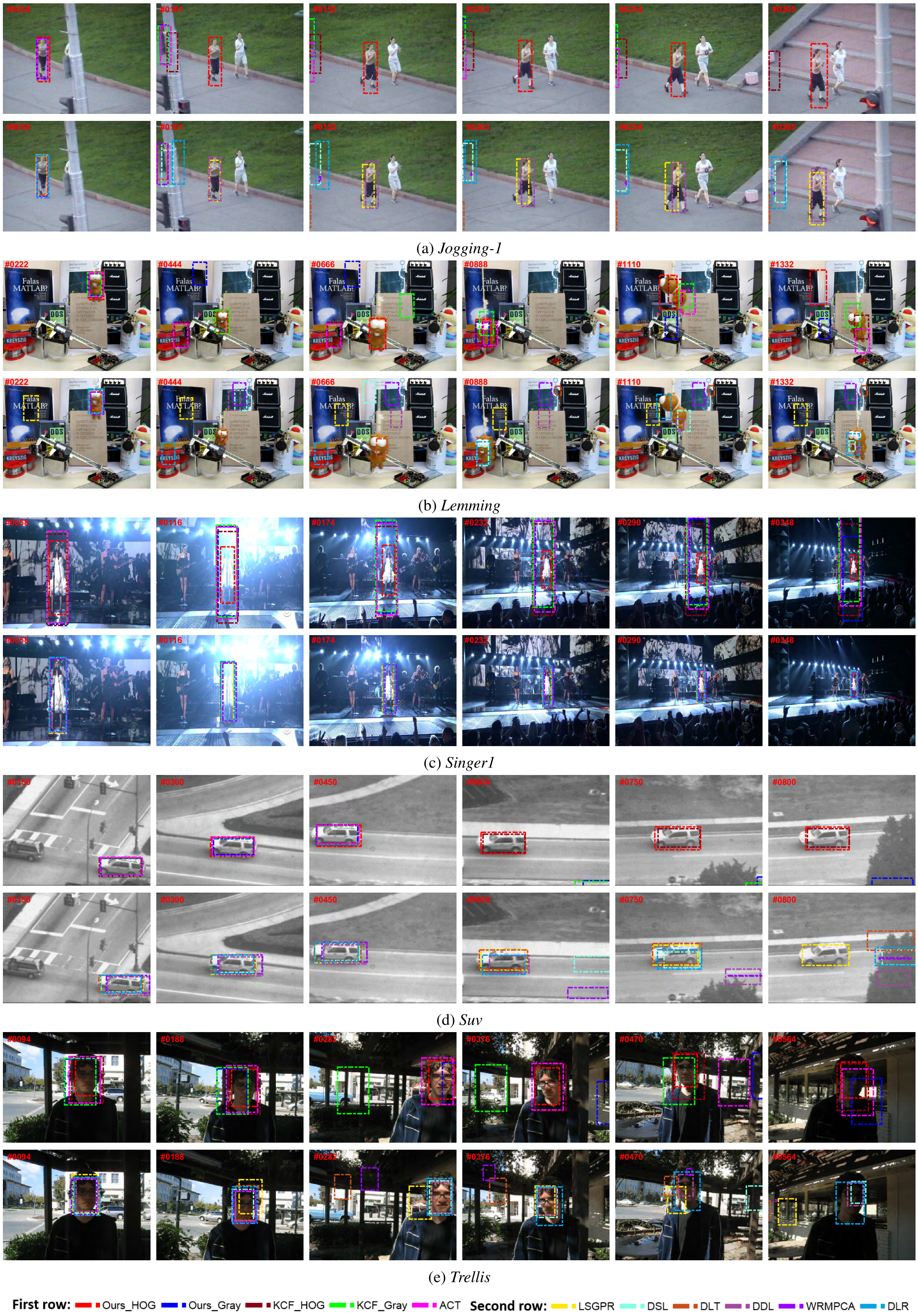
(c) *Fleetface*

(d) *Freeman3*

(e) *Freeman4*

FIGURE 5. Examples of tracking results of the compared methods on five OTB-50 sequences, (a) *Car4*, (b) *CarScale*, (c) *Fleetface*, (d) *Freeman3* and (e) *Freeman4*.

in some frames, they are able to track the object once again by chance as the object reappears at the same location due to camera pan or due to repetitive motion of the object. All the trackers perform well in the *Car4* sequence except KCF_Gray, KCF-B2DPCA_Gray and ACT, where the tracker drifts away slightly but with imprecise estimation of


#0222
#0444
#0666
#0888
#1110
#1332
#0116
#0174
#0232
#0290
#0348
#0300
#0450
#0750
#0800
#0094
#0188
#0282
#0376
#0470
#0564
First row: ▬ Ours_HOG ▬ Ours_Gray ▬ KCF_HOG ▬ KCF_Gray ▬ ACT Second row: ▬ LSGPR ▬ DSL ▬ DLT ▬ DDL ▬ WRMPCA ▬ DLR

(a) *Jogging-1*

(b) *Lemming*

(c) *Singer1*

(d) *Suv*

(e) *Trellis*

FIGURE 6. Examples of tracking results of the compared methods on five OTB-50 sequences, (a) *Jogging-1*, (b) *Lemming*, (c) *Singer1*, (d) *Suv* and (e) *Trellis*.

scale. In the *CarScale* sequence, all the methods track the target, but fail to estimate the scale information effectively, except WRMPCA that drifts away towards the end of the sequence. Similar observations can be made even in the *Fleetface* sequence, where WRMPCA and ACT fail towards the end of the sequence. KCF-B2DPCA_HOG and DLT estimate both the scale and location of the target effectively in *Freeman3*, whereas DSL, WRMPCA and DLR fail to estimate the scale of the target. The only method that tracks the target effectively until the end in the *Freeman4* sequence is KCF-B2DPCA_HOG. Further, in the *Jogging-1* sequence, LSGPR, DDL and KCF-B2DPCA_HOG track the object successfully, whereas the other trackers fail to track the object after a few initial frames. In *Singer1*, even though all the methods track the target to some extent, they fail to estimate the scale accurately except DDL, DSL, WRMPCA, DLT, DLR, LSGPR and KCF-B2DPCA_HOG, which estimate the scale more precisely than the others. Also, KCF_HOG, KCF-B2DPCA_HOG and LSGPR track the object successfully in *Suv* except DDL, DLT, DLR, WRMPCA, DSL, ACT and KCF_Gray, which fail to track the object towards the end of the sequence. Further, KCF_HOG, DLR, DDL, KCF-B2DPCA_HOG, KCF-B2DPCA_Gray and ACT successfully track the target in *Trellis*, but fail to estimate the scale precisely except KCF-B2DPCA_HOG. Thus, from these qualitative analyses, it is observed that the proposed KCF-B2DPCA_HOG tracker performs favorably in most of the challenging sequences.

VI. CONCLUSION

In this paper, a collaborative tracking algorithm based on the discriminative and generative models has been proposed. The correlation filters, based on the discriminative model, have been used to find the target position, and a robust coding in the B2DPCA subspace appearance model, based on generative framework, has been used to find the remaining affine motion parameters of the target. This is motivated by the idea that the discriminative capability of the tracker plays a important role while finding the location of the target rather than while finding the other affine motion parameters of the target. On the other hand, the generative capability of the tracker plays a prominent role while finding the other affine motion parameters of the target. Also, the robust coding (RC) has been extended to 2D residuals, to account for non-Laplacian or non-Gaussian noise, and introduced into B2DPCA reconstruction. In addition, a 2D robust coding (2DRC) distance metric has been introduced to find the candidates having appearance similar to that of the subspace and used to compute the observation likelihood in the generative model. Further, a robust occlusion map has been generated from the weights obtained during the residual minimization and used to obtain occlusion-free observation samples, which are then accumulated for the B2DPCA appearance model update. The occlusion map thus obtained has also been used in the appearance model update of both the correlation filters in different ways to avoid the degradation of their appearance models. Extensive experiments have been conducted on three popular tracking benchmark datasets, namely, OTB-50, VOT2016 and UAV20L, to analyze the performance of the proposed method. Quantitative and qualitative performance of the proposed method has been compared with that of several recent state-of-the-art algorithms using these benchmark datasets, and it has been shown that the proposed method outperforms the state-of-the-art methods. Finally, it needs to be pointed out that despite the fact that the UAV20L dataset is extremely challenging in that it contains long duration sequences with large variations both in the camera motion and view points in three dimensions, the proposed method performs well and better than other state-of-the-art methods do, when it is applied to the sequences of this dataset.

ACKNOWLEDGMENT

The authors would like to thank the authors of [33], [34], [46], and [48]–[56]who have made their codes/results available for comparison with the proposed method.

REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, pp. 1–45, 2006.

[2] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, "Recent advances and trends in visual tracking: A review," *Neurocomputing*, vol. 74, no. 18, pp. 3823–3831, Nov. 2011.

[3] D. Wang, H. Lu, and M.-H. Yang, "Online object tracking with sparse prototypes," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 314–325, Jan. 2013.

[4] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 2411–2418.

[5] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–577, May 2003.

[6] X. Mei and H. Ling, "Robust visual tracking using ℓ_1 minimization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Sep./Oct. 2009, pp. 1436–1443.

[7] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust ℓ_1 tracker using accelerated proximal gradient approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn. (CVPR)*, Jun. 2012, pp. 1830–1837.

[8] B. K. S. Kumar, M. N. S. Swamy, and M. O. Ahmad, "Structural local DCT sparse appearance model for visual tracking," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2015, pp. 1194–1197.

[9] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1822–1829.

[10] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, New York, NY, USA, Jun. 2006, pp. 798–805.

[11] J. Yang, R. Xu, J. Cui, and Z. Ding, "Robust visual tracking using adaptive local appearance model for smart transportation," *Multimedia Tools Appl.*, vol. 75, no. 24, pp. 17487–17500, Dec. 2016.

[12] B. K. S. Kumar, M. N. S. Swamy, and M. O. Ahmad, "Visual tracking using structural local DCT sparse appearance model with occlusion detection," *Multimedia Tools and Applications*. Cham, Switzerland: Springer, 2018, pp. 1–24.

[13] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 125–141, 2008.

[14] B. K. S. Kumar, M. N. S. Swamy, and M. O. Ahmad, "Weighted residual minimization in PCA subspace for visual tracking," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2016, pp. 986–989.

[15] H. Wang, H. Ge, and S. Zhang, "Object tracking via 2DPCA and ℓ_2-regularization," *J. Elect. Comput. Eng.*, vol. 2016, Jul. 2016, Art. no. 7975951.

[16] D. Wang and H. Lu, "Object tracking via 2DPCA and ℓ_1-regularization," *IEEE Signal Process. Lett.*, vol. 19, no. 11, pp. 711–714, Nov. 2012.

[17] B. K. S. Kumar, M. N. S. Swamy, and M. O. Ahmad, "Visual tracking via bilateral 2DPCA and robust coding," in *Proc. IEEE Can. Conf. Elect. Comput. Eng. (CCECE)*, May 2016, pp. 1–4.

[18] P. Qu, "Visual tracking with fragments-based PCA sparse representation," *Int. J. Signal Process., Image Process. Pattern Recognit.*, vol. 7, no. 2, pp. 23–34, Feb. 2014.

[19] M. Sun, D. Du, H. Lu, and L. Zhang, "Visual tracking with astructured local model," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 2855–2859.

[20] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proc. CVPR*, Jun. 2009, pp. 983–990.

[21] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Proc. Eur.Conf. Comput. Vis. (ECCV)*, Oct. 2008, pp. 234–247.

[22] F. Wang, J. Zhang, Q. Guo, P. Liu, and D. Tu, "Robust visual tracking via discriminative structural sparse feature," in *Proc. Chin. Conf. Image Graph. Technol.*, Jun. 2015, pp. 438–446.

[23] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparse collaborative appearance model," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2356–2368, May 2014.

[24] C. Xie, J. Tan, P. Chen, J. Zhang, and L. He, "Collaborative object tracking model with local sparse representation," *J. Vis. Commun. Image Represent.*, vol. 25, no. 2, pp. 423–434, 2014.

[25] B. Zhuang, L. Wang, and H. Lu, "Visual tracking via shallow and deep collaborative model," *Neurocomputing*, vol. 218, pp. 61–71, Dec. 2016.

[26] H. Zhang, F. Tao, and G. Yang, "Robust visual tracking based on structured sparse representation model," *Multimedia Tools Appl.*, vol. 74, no. 3, pp. 1021–1043, 2015.

[27] C. Ma, J. Huang, X. Yang, and M. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Jun. 2015, pp. 3074–3082.

[28] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M.-H. Yang, "Hedged deep tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4303–4311.

[29] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2016, pp. 472–488.

[30] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4293–4302.

[31] D. Bolme, J. Beveridge, B. Draper, and Y. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Sep. 2010, pp. 2544–2550.

[32] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2012, pp. 702–715.

[33] 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.

[34] M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn. (CVPR)*, Jun. 2014, pp. 1090–1097.

[35] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Sep. 2014, pp. 1–11.

[36] T. Zhang, A. Bibi, and B. Ghanem, "In defense of sparse tracking: Circulant sparse tracker," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3880–3888.

[37] T. Wang, I. Y. H. Gu, and P. Shi, "Object tracking using incremental 2D-PCA learning and ML estimation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2007, pp. I-933–I-936.

[38] H. Kong, L. Wang, E. K. Teoh, X. Li, J.-G. Wang, and R. Venkateswarlu, "Generalized 2D principal component analysis for face image representation and recognition," *Neural Netw.*, vol. 18, nos. 5–6, pp. 585–594, Jul. 2005.

[39] M.-X. Jiang, M. Li, and H.-Y. Wang, "Visual object tracking based on 2DPCA and ML," *Math. Problems Eng.*, vol. 2013, May 2013, Art. no. 404978.

[40] M. Yang, L. Zhang, J. Yang, and D. Zhang, "Robust sparse coding for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn. (CVPR)*, Jun. 2011, pp. 625–632.

[41] J. Yan and M. Tong, "Weighted sparse coding residual minimization for visual tracking," in *Proc. Vis. Commun. Image Process. (VCIP)*, Nov. 2011, pp. 1–4.

[42] M. Jiang, H. Wang, and B. Wang, "Robust visual tracking based on maximum likelihood estimation," *Int. J. Digit. Content Technol. Appl.*, vol. 6, no. 22, pp. 467–474, Dec. 2012.

[43] S. A. Siena, "Improving the design and use of correlation filters in visual tracking," Ph.D. dissertation, Dept. Elect. Comput. Eng., Carnegie Mellon Univ., Pittsburgh, PA, USA, 2017.

[44] M. Isard and A. Blake, "CONDENSATION—Conditional density propagation for visual tracking," *Int. J. Comput. Vis.*, vol. 29, no. 1, pp. 5–28, Aug. 1998.

[45] M. J. Black and A. D. Jepson, "EigenTracking: Robust matching and tracking of articulated objects using a view-based representation," *Int. J. Comput. Vis.*, vol. 26, no. 1, pp. 63–84, 1998.

[46] D. Wang, H. Lu, and M.-H. Yang, "Robust visual tracking via least soft-threshold squares," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 9, pp. 1709–1721, Sep. 2016.

[47] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, and R. Pflugfelder, "The visual object tracking VOT2016 challenge results," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2016, pp. 777–823.

[48] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for UAV tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2016, pp. 445–461.

[49] Y. Sui, Y. Tang, L. Zhang, and G. Wang, "Visual tracking via subspace learning: A discriminative approach," *Int. J. Comput. Vis.*, vol. 126, no. 5, pp. 515–536, 2018.

[50] D. Wang, H. Lu, and C. Bo, "Visual tracking via weighted local cosine similarity," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 1838–1850, Sep. 2015.

[51] D. Wang, H. Lu, Z. Xiao, and M.-H. Yang, "Inverse sparse tracker with a locally weighted distance metric," *IEEE Trans. Image Process.*, vol. 24, no. 9, pp. 2646–2657, Sep. 2015.

[52] D. Wang, H. Lu, and C. Bo, "Fast and robust object tracking via probability continuous outlier model," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5166–5176, Dec. 2015.

[53] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 809–817.

[54] Y. Sui, G. Wang, L. Zhang, and M.-H. Yang, "Exploiting spatial-temporal locality of tracking via structured dictionary learning," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1282–1296, Mar. 2018.

[55] Y. Sui and L. Zhang, "Visual tracking via locally structured Gaussian process regression," *IEEE Signal Process. Lett.*, vol. 22, no. 9, pp. 1331–1335, Sep. 2015.

[56] Y. Sui, Y. Tang, and L. Zhang, "Discriminative low-rank tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3002–3010.

B. K. SHREYAMSHA KUMAR received the B.E. degree in electronics and communication engineering from Bangalore University, India, in 2000, and the M.Tech. degree in industrial electronics from the National Institute of Technology Karnataka, Surathkal, India, in 2004. He is currently pursuing the Ph.D. degree in electrical and computer engineering with Concordia University, Montreal, QC, Canada. He has been a Research Associate with the Signal Processing Group, Concordia University, since 2012. Before joining Concordia, he was with the Central Research Laboratory (A Corporate Research Facility of Bharat Electronics, India) as a member (Research Staff) from 2004 to 2012. He is a recipient of the R&D Excellence Award conferred by Bharat Electronics. His research interests include visual tracking, computer vision, image fusion, image denoising, image encryption, and document image processing. He has served as a reviewer for several peer-reviewed journals and major conferences.

M.N.S. SWAMY (S'59–M'62–SM'74–F'80) received the B.Sc. degree (Hons.) in mathematics from the University of Mysore, Mysore, India, in 1954, the Diploma degree in electrical communication engineering from the Indian Institute of Science, Bengaluru, India, in 1957, and the M.Sc. and Ph.D. degrees in electrical engineering from the University of Saskatchewan, Saskatoon, SK, Canada, in 1960 and 1963, respectively. He was conferred with the title of Honorary Professor by the National Chiao Tong University, Hsinchu, Taiwan, in 2009. He is currently a Research Professor with the Department of Electrical and Computer Engineering, Concordia University, Montreal, QC, Canada, where he served as the Founding Chair of the Department of Electrical Engineering from 1970 to 1977, and the Dean of engineering and computer science from 1977 to 1993. During that time, he developed the faculty into a research-oriented one, from what was primarily an undergraduate faculty. Since 2001, he has been the Concordia Chair (Tier I) in signal processing. He has also taught at the Department of Electrical Engineering, Technical University of Nova Scotia, Halifax, NS, Canada, the University of Calgary, Calgary, AB, Canada, and the Department of Mathematics, University of Saskatchewan. He has published in the areas of number theory, circuits, systems, and signal processing, and holds five patents. He has co-authored nine books and five book chapters. He was a Founding Member of Micronet, a Canadian Network of Centers of Excellence from 1990 to 2004, and also a Coordinator of Concordia University. He is a fellow of the Institute of Electrical Engineers, U.K, the Engineering Institute of Canada, the Institution of Engineers, India, and the Institution of Electronic and Telecommunication Engineers, India. He was inducted to the Provosts Circle of Distinction for career achievements in 2009. He is a recipient of many IEEE-CAS Society awards, including the Education Award in 2000, the Golden Jubilee Medal in 2000, and the 1986 Guillemin-Cauer Best Paper Award. He has served as the Program Chair for the 1973 IEEE Circuits and Systems (CAS) Symposium, the General Chair for the 1984 IEEE CAS Symposium, the Vice Chair of the 1999 IEEE CAS Symposium, and a member of the Board of Governors of the CAS Society. He has been the Editor-in-Chief of the journal *Circuits, Systems and Signal Processing* (CSSP) since 1999. Recently, CSSP has instituted the Best Paper Award in his name. He has served as the Editor-in-Chief for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I from 1999 to 2001, and an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS from 1985 to 1987. He has served the IEEE in various capacities, such as the President Elect in 2003, the President in 2004, the Past-President in 2005, the Vice President (publications) from 2001 to 2002, and the Vice President in 1976.

M. OMAIR AHMAD (S'69–M'78–SM'83–F'01) received the B.Eng. degree in electrical engineering from Sir George Williams University, Montreal, QC, Canada, and the Ph.D. degree in electrical engineering from Concordia University, Montreal. From 1978 to 1979, he was a Faculty Member with the New York University College, Buffalo, NY, USA. In 1979, he joined the faculty of Concordia University as an Assistant Professor of computer science. Subsequently, he joined the Department of Electrical and Computer Engineering, Concordia University, where he was the Chair of the Department from 2002 to 2005, and is currently a Professor. He was a Founding Researcher of Micronet, a Canadian Network of Centers of Excellence, from 1990 to 2004. He is also the Concordia University Research Chair (Tier I) in multimedia signal processing. He has authored in the area of signal processing and holds four patents. His current research interests include the areas of image and speech processing, biomedical signal processing, watermarking, biometrics, video signal processing and object detection and tracking, deep learning techniques in signal processing, and fast signal transforms and algorithms. In 1988, he was a member of the Admission and Advancement Committee of the IEEE. He was a recipient of numerous honors and awards, including the Wighton Fellowship from the Sandford Fleming Foundation, an induction to Provosts Circle of Distinction for Career Achievements, and the Award of Excellence in Doctoral Supervision from the Faculty of Engineering and Computer Science, Concordia University. He was a Guest Professor at Southeast University, Nanjing, China, and the Local Arrangements Chairman of the 1984 IEEE International Symposium on Circuits and Systems. He has served as the Program Co-Chair for the 1995 IEEE International Conference on Neural Networks and Signal Processing, the 2003 IEEE International Conference on Neural Networks and Signal Processing, and the 2004 IEEE International Midwest Symposium on Circuits and Systems. He was the General Co-Chair of the 2008 IEEE International Conference on Neural Networks and Signal Processing. He is the Chair of the Montreal Chapter IEEE Circuits and Systems Society. He was an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS Part I: FUNDAMENTAL THEORY AND APPLICATIONS from 1999 to 2001.