Visual tracking via Boolean map representations

Kaihua Zhang, Qingshan Liu, Jian Yang, Ming-Hsuan Yang

Abstract

In this paper, we present a simple yet effective Boolean map based representation that exploits connectivity cues for visual tracking. We describe a target object with histogram of oriented gradients and raw color features, of which each one is characterized by a set of Boolean maps generated by uniformly thresholding their values. The Boolean maps effectively encode multi-scale connectivity cues of the target with different granularities. The fine-grained Boolean maps capture spatially structural details that are effective for precise target localization while the coarse-grained ones encode global shape information that are robust to large target appearance variations. Finally, all the Boolean maps form together a robust representation that can be approximated by an explicit feature map of the intersection kernel, which is fed into a logistic regression classifier with online update, and the target location is estimated within a particle filter framework. The proposed representation scheme is computationally efficient and facilitates achieving favorable performance in terms of accuracy and robustness against the state-of-the-art tracking methods on the OTB50 and VOT2016 benchmark datasets.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

Object tracking is a fundamental problem in computer vision and image processing with numerous applications. Despite significant progress in past decades, it remains a challenging task due to large appearance variations caused by illumination changes, partial occlusion, deformation, as well as cluttered background. To address these challenges, a robust representation, especially under a pixel filter framework, plays a critical role for the success of a visual tracker, and attracts much attention in recent years [1–12].

Numerous representation schemes have been developed for visual tracking based on holistic and local features. Lucas and Kanade [13] leverage holistic templates based on raw pixel values to represent target appearance. Matthews et al. [14] design an effective template update scheme that uses stable information from the first frame for visual tracking. In [15] Henriques et al. propose a correlation filter based template (trained with raw intensity) for visual tracking with promising performance. Zhang et al. [16] propose a multi-expert restoration scheme to address the drift problem in tracking, in which each base tracker leverages an explicit feature map representation via quantizing the CIE LAB color channels of spatially sampled image patches. To deal with appearance changes, subspace learning based trackers have been proposed. Black and Jepson [17] develop a pre-learned view-based eigenbasis representation for visual tracking. However, the pre-trained representation cannot adapt well to significant target appearance variations. In [18] Ross et al. propose an incremental update scheme to learn a low-dimensional subspace representation. Recently, numerous tracking algorithms based on sparse representation have been proposed. Mei and Ling [19] devise a dictionary of holistic intensity templates with target and trivial templates, and then find the location of the object with minimal reconstruction error via solving an $\ell_1$ minimization problem. Zhang et al. [20] formulate visual tracking as a multi-task sparse learning problem, which learns particle representations jointly. In [21] Wang et al. introduce $\ell_1$ regularization into the eigen-reconstruction to develop an effective representation that combines the merits of both subspace and sparse representations.

In spite of demonstrated success of exploiting global representations for visual tracking, existing methods are less effective in dealing with heavy occlusion and large deformation as local visual cues are not taken into account. Consequently, local representations are developed to handle occlusion and deformation. Adam et al. [22] propose a fragment-based tracking method that divides a target object into a set of local regions and represents each region with a histogram. In [23] He et al. present a locality sensitive histogram for visual tracking by considering the contributions of local regions at each pixel, which can model target appearance
well. Babenko et al. [24] formulate the tracking task as a multiple instance learning problem, in which Haar-like features are used to represent target appearance. Hare et al. [25] pose visual tracking as a structure learning task and leverage Haar-like features to describe target appearance. In [26] Henriques et al. propose an algorithm based on a kernel correlation filter (KCF) to describe target templates with feature maps based on histogram of oriented gradients (HOG) [27]. This method has been shown to achieve promising performance on the recent tracking benchmark dataset [28] in terms of accuracy and efficiency. Kwon and Lee [29] present a tracking method that represents target appearance with a set of local patches where the topology is updated to account for large shape deformation. Jia et al. [30] propose a structural sparse representation scheme by dividing a target object into some local image patches in a regular grid and using the coefficients to analyze occlusion and deformation.

Hierarchical representation methods that capture holistic and local object appearance have been developed for visual tracking [31–34]. Zhong et al. [31] propose a sparse collaborative appearance model for visual tracking in which both holistic templates and local representations are used. Li and Zhu [32] extend the KCF tracker [26] with a scale adaptive scheme and effective color features. In [33], Wang et al. demonstrate that a simple tracker based on logistic regression with a representation composed of HOG and raw color channels performs favorably on the benchmark dataset [28]. Ma et al. [34] exploit features from hierarchical layers of a convolutional neural network and learn an effective KCF which takes account of spatial details and semantics of target objects for visual tracking.

In biological vision, it has been suggested that object tracking is carried out by attention mechanisms [35,36]. Global topological structure such as connectivity is used to model tasks related to visual attention [37,38]. However, all aforementioned representations do not consider topological structure for visual tracking.

In this work, we propose a Boolean map based representation (BMR) that leverages connectivity cues for visual tracking. One case of connectivity is the enclosure topological relationship between the (foreground) figure and ground which defines the boundaries of figures. Recent gestalt psychological studies suggest that the enclosure topological cues play an important role in figure-ground segregation and have been successfully applied to saliency detection [39] and measuring objectness [40,41]. The proposed BMR scheme characterizes target appearance by concatenating multiple layers of Boolean maps at different granularities based on uniformly thresholding HOG and color feature maps. The fine-grained Boolean maps capture locally spatial structural details that are effective for precise localization and coarse-grained ones which encode much global shape information to account for significant appearance variations. The Boolean maps are then concatenated and normalized to a BMR scheme that can be just approximated by an explicate feature map. We learn a logistic regression classifier using the BMR scheme and online update to estimate target locations within a particle filter framework. The effectiveness of the proposed algorithm is demonstrated on the OTB50 [28] and VOT2016 [42] tracking benchmark datasets against the state-of-the-art approaches.

The main contributions of this work are summarized as follows:

- We demonstrate that the connectivity cues can be effectively used for robust visual tracking.
- We show that the BMR scheme can be approximated as an explicit feature map of the intersection kernel which can find a nonlinear classification boundary via a linear classifier. In addition, it is easy to train and detect for robust visual tracking with this approach.
- The proposed tracking algorithm based on the BMR scheme performs favorably against the state-of-the-art trackers in terms of accuracy and robustness to initializations based on the OTB50 and VOT2016 benchmark datasets.

2. Tracking via Boolean map representations

We present the BMR scheme and a logistic regression classifier with online update for visual tracking.

2.1. Boolean map representation

The proposed image representation is based on recent findings of human visual attention [43] which shows that momentary conscious awareness of a scene can be represented by Boolean maps. The Boolean maps are concerned with center-surround contrast that mimic the sensitivity of neurons either to dark centers on bright surrounds or vice versa [44]. Specifically, we exploit the connectivity cues inside a target measured by the Boolean maps which can be used for separating the foreground object from the background effectively [37,39–41]. As demonstrated in Fig. 1, the connectivity inside a target can be well captured by the Boolean maps at different scales.

Neurobiological studies have demonstrated that human visual system is sensitive to color and edge orientations [45] which provide useful cues to discriminate the foreground object from the background. In this work, we use color features in the CIE LAB color space and HOG features to represent objects. Note that we employ these feature representations due to their extremely efficient computation and well discriminative capability. Besides, the deep convolutional neural networks (CNN) features extracted by R-CNN [46], faster R-CNN [47], or very deep CNNs [34] can also be exploited to represent the targets, but their computational complexity will be very high. To extract the perceptually uniform color features, we first normalize each sample \( x \) to a canonical size \((32 \times 32)\) in our experiments, and then subsample it to a half size to reduce appearance variations, and finally transform the sample into the CIE LAB color space, denoted as \( \Phi^{\text{col}}(x) = \mathbb{R}^{n_{\text{col}} \times d} \) \((n_{\text{col}} = 16\) in this work). Furthermore, we leverage the HOG features to capture edge orientation information of a target object, denoted as \( \Phi^{\text{hog}}(x) = \mathbb{R}^{2 \times n_{\text{col}} \times d} \) \((n_{\text{col}} = 4\) in this work). Fig. 1 demonstrates that most color and HOG feature maps of the target own center-surrounded patterns that are similar to biologically plausible architecture of primates in [44]. We normalize both \( \Phi^{\text{col}}(x) \) and \( \Phi^{\text{hog}}(x) \) to range from 0 to 1, and concatenate \( \Phi^{\text{col}}(x) \) and \( \Phi^{\text{hog}}(x) \) to form a feature vector \( \phi(x) = \mathbb{R}^{d \times 1} \) with \( d = 3n_{\text{col}}^2 + 31n_{\text{col}}^2 \). The feature vector is rescaled to \([0, 1]\) by

\[
\phi(x) = \frac{\phi(x) - \min(\phi(x))}{\max(\phi(x)) - \min(\phi(x))},
\]

(1)

where \( \max(\cdot) \) and \( \min(\cdot) \) denotes the maximal and minimal operators, respectively.

Next, \( \phi(x) \) in (1) is encoded into a set of vectorized Boolean maps \( B(x) = \{b_i(x)\}_{i=1}^C \) by

\[
b_i(x) = \begin{cases} 
1, & \phi(x) \geq \theta_i, \\
0, & \text{otherwise}. 
\end{cases}
\]

(2)

where \( \theta_i \sim U(0, 1) \) is a threshold drawn from a uniform distribution over \([0, 1]\), and the symbol \( \geq \) denotes elementwise inequality. In this work, we set \( \theta_i = i/C \) that is simply sampled at a fixed-step size \( \delta = 1/C \), and a fixed-step sampling is equivalent to the uniform sampling in the limit \( \delta \to 0 \) [39]. Hence, we have \( b_i(x) \geq b_j(x) \geq \ldots \geq b_0(x) \). It is easy to show that

\[
0 \leq \phi_b(x) - \frac{1}{C} \sum_{j=1}^C b_{ij}(x) < \delta,
\]

(3)
where $\phi_k$ and $b_{jk}$ are the k-th entries of $\phi$ and $b$, respectively.

Proof. Without loss of generality, we assume that $i\delta \leq \phi_k(x) < (1+i)\delta$, $i=0,\ldots,c$. As such, we have $b_{jk}(x)=1$ for all $j \leq i$ because $b_1(x) \geq b_2(x) \geq \ldots \geq b_c(x)$, and $b_{jk}(x)=0$ for $j>i$. Therefore, we have $\frac{1}{c}\sum_{j=1}^{c}b_{jk}(x) = i\delta$, and $0 \leq \phi_k(x) - \frac{1}{c}\sum_{j=1}^{c}b_{jk}(x) < (1+i)\delta - i\delta = \delta$.

In (3), when $\delta \to 0$ (i.e., $\theta_t \sim U(0,1)$) we have

$$
\phi_k(x) = \frac{1}{c}\sum_{j=1}^{c}b_{jk}(x).
$$

(4)

In this work, we set $\delta = 0.25$. Although (4) may not be strictly satisfied, empirical results show that most distinct structures in $\phi(x)$ can be reconstructed as demonstrated in Fig. 2. Furthermore, the reconstructed representations contain more connected structures than the original ones (see the ones highlighted in yellow in Fig. 2), which shows that the Boolean maps facilitate capturing global geometric information of target objects.

Based on (4), to measure the similarity between two samples $x$ and $y$, we use the intersection function [48]

$$
\mathcal{I}(\phi(x),\phi(y)) = \frac{1}{c}\sum_{k=1}^{d}\min(\phi_k(x),\phi_k(y))
$$

$$
= \frac{1}{c}\sum_{k=1}^{d}\min\left(\frac{1}{c}\sum_{j=1}^{c}b_{jk}(x),\frac{1}{c}\sum_{j=1}^{c}b_{jk}(y)\right)
$$

$$
= \frac{1}{c}\sum_{k=1}^{d}\sum_{j=1}^{c}\min(b_{jk}(x),b_{jk}(y))
$$

$$
= \frac{1}{c}\sum_{k=1}^{d}b_{jk}(x)b_{jk}(y)
$$

$$
= \langle b(x),b(y) \rangle,
$$

where $\tilde{b} = |b_1^\top \ldots b_c^\top|^\top / \sqrt{c}$.

To avoid favoring larger input sets [48], we normalize $\mathcal{I}(\phi(x),\phi(y))$ in (5) and define the kernel $\mathcal{K}(\phi(x),\phi(y))$ as

$$
\mathcal{K}(\phi(x),\phi(y)) = \frac{\mathcal{I}(\phi(x),\phi(y))}{\sqrt{\mathcal{I}(\phi(x),\phi(x))\mathcal{I}(\phi(y),\phi(y))}}
$$

$$
= \langle b(x),b(y) \rangle,
$$

where $b(.)$ is an explicit feature map function. In this work, the feature map function is defined by

$$
b(x) = \frac{\tilde{b}(x)}{||\tilde{b}(x)||_2^2},
$$

(7)

where $||·||_2$ is an $\ell_2$ norm operator. We use $b(x)$ to train a linear classifier, which is able to address the nonlinear classification problem in the feature space of $\phi$ for visual tracking with favorable performance. The proposed tracking algorithm based on BMR is summarized in Algorithm 1.

2.2. Learning linear classifier with BMRs

We pose visual tracking as a binary classification problem with local search, in which a linear classifier is learned in the Boolean map feature space to separate the target from the background. Specifically, we use a logistic regressor to learn the classifier for measuring similarity scores of samples.

Let $l_i(x_i) \in \mathbb{R}^2$ denote the location of the $i$-th sample at frame $t$. We assume that $l_i(x_i)$ is the object location, and densely draw samples $D^\theta = \{x||l_i(x_i) - l_i(x_k)|| < \alpha\}$ at a search radius $\alpha$ centered at the current object location, and label them as positive samples. Next, we uniformly sample some patches from set $D^\beta = \{x||l_i(x_i) - l_i(x_k)|| < \beta\}$, and label them as negative samples. After representing these samples with BMRs, we obtain
Algorithm 1 BMR.

**Input:** Normalized image patch \( x \);

1. Compute feature vector \( \phi(x) \) in (1);
2. for all entries \( \phi_i(x), i = 1, \ldots, d \) of \( \phi(x) \), do
3. \( \delta = \delta : \delta = 1 - \delta \), do
4. if \( \phi_i(x) > \delta \), then
5. \( b_i(x) = 1 \);
6. else
7. \( b_i(x) = 0 \);
8. end if
9. end for
10. \( \bar{b}(x) = [b_1(x), \ldots, b_d(x)]^T \); /\Normalization/
11. \( \bar{b}(x) \leftarrow \bar{b}(x)/|\bar{b}(x)| \); //Normalization;

**Output:** BMR \( b(x) \).

Output: \( b(x) \).

\[ \ell_t(w) = \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-y_i^t w^T b(x_i^t))), \] (8)

\[ f(x) = \frac{1}{1 + \exp(-w^T b(x))}. \] (9)

We use a gradient descent method to minimize \( \ell_t(w) \) by iterating

\[ w \leftarrow w - \frac{\partial \ell_t(w)}{\partial w}. \] (10)

\[ \frac{\partial \ell_t(w)}{\partial w} = \frac{1}{n} \sum_{i=1}^{n} b_i(x_i^t) y_i^t \exp(-y_i^t w^T b(x_i^t)). \]

In this work, we use the parameter \( w_{t+1} \) obtained at frame \( t+1 \) to initialize \( w \) in (10) and iterate 20 times for online updates, which is a simple online update strategy that takes into account the useful temporal information from the former frames.

2.3. Proposed tracking algorithm

We estimate the target states sequentially within a particle filter framework. Given the observation set \( O_t = \{o_i\}_{i=1}^{n_o} \) up to frame \( t \), the target state \( s_t \) is obtained by maximizing the posteriori probability

\[ \hat{s}_t = \arg\max_{s_t} p(s_t | O_t) \propto p(o_t | s_t) p(s_t | O_{t-1}), \] (11)

where \( p(s_t | O_{t-1}) \propto \int p(s_t | s_{t-1}) p(s_{t-1} | O_{t-1}) ds_{t-1} \). \( s_t = [x_t, y_t, s_t] \) is the target state with translations \( x_t \) and \( y_t \), and scale \( s_t \). \( p(s_t | s_{t-1}) \) is a dynamic model that describes the temporal correlation of the target states in two consecutive frames, and \( p(o_t | s_t) \) is the observation model that estimates the likelihood of a state given an observation. In the proposed algorithm, we assume that the target state parameters are independent and modeled by three scalar Gaussian distributions between two consecutive frames, i.e., \( p(s_t | s_{t-1}) = N(s_t | s_{t-1}, \Sigma) \), where \( \Sigma = \text{diag} (\sigma_x, \sigma_y, \sigma_t) \) is a diagonal covariance matrix whose elements are the standard deviations of the target state parameters. In visual tracking, the posterior probability \( p(s_t | O_t) \) in (11) is approximated by a finite set of particles \( \{s_t\}_{i=1}^{n_p} \) that are sampled with corresponding importance weights \( \{\pi_i^t\}_{i=1}^{n_p} \), where \( \pi_i^t \propto p(o_t | s_t^i) \). Therefore, (11) can be approximated as

\[ \hat{s}_t = \arg \max_{s_t} p(o_t | s_t^i) p(s_t^i | \hat{s}_{t-1}). \] (12)

In our method, the observation model \( p(o_t | s_t^i) \) is defined as

\[ p(o_t | s_t^i) \propto f(x_t). \] (13)

where \( f(x_t) \) is the logistic regression classifier defined by (9).

To adapt to target appearance variations while preserving the stable information that helps prevent the tracker from drifting to background, we update the classifier parameters \( w \) in a conservative way. We update \( w \) by (10) only when the confidence of the target falls below a threshold \( \rho \). This ensures that the target states always have high confidence scores and alleviate the problem of including noisy samples when updating classifier [33]. The main steps of the proposed algorithm are summarized in Algorithm 2.

Algorithm 2 BMR-based Tracking.

**Input:** Target state \( \hat{s}_{t-1} \), classifier parameter vector \( w_t \);

1. Sample \( n_p \) candidate particles \( \{s_t^i\}_{i=1}^{n_p} \) with the motion model \( p(s_t^i | s_{t-1}^i) \) in (12);
2. For each particle \( s_t^i \), extract the corresponding image patch \( x_t^i \), and compute the BMR \( b(x_t^i) \) by Algorithm 1, and compute the corresponding observation model \( p(o_t | s_t^i) \) by (13);
3. Estimate the optimal state \( s_t^i \) by (12), and obtain the corresponding image patch \( x_t^i \);
4. If \( f(x_t^i) < \rho \), then
5. Update \( w_t \) by iterating (10) until convergence, and set \( w_{t+1} \leftarrow w_t \);
6. else
7. \( w_{t+1} \leftarrow w_t \);
8. end if

**Output:** Target state \( \hat{s}_t \) and classifier parameter vector \( w_{t+1} \).

3. Experimental results

We first present implementation details of the proposed algorithm, and discuss the dataset and metrics for performance evaluation. Next, we analyze the empirical results using widely-adopted metrics. We present ablation study to examine the effectiveness of each key component in the proposed BMR scheme. Finally, we show and analyze some failure cases.

3.1. Implementation details

All images are resized to a fixed size of 240 x 320 pixels [33] for experiments and each patch is resized to a canonical size of 32 x 32 pixels. In addition, each canonical patch is subsampled to a half size with \( n_{col} = 16 \) for color representations. The HOG features are extracted from the canonical patches that supports both gray and color images, and the sizes of HOG feature maps are the same as \( n_{col} \times n_{col} = 31 \times 31 \) (as implemented in http://github.com/pdollar/toolbox).

For grayscale videos, the original image patches are used to extract raw intensity and HOG features, and the feature dimension \( d = 4 \times 4 \times 31 \times 16 \times 16 = 752 \). For color videos, the image patches are transformed to the CIE LAB color space to extract raw color features, and the original RGB image patches are used to extract HOG features. The corresponding total dimension \( d = 4 \times 4 \times 31 \times 16 \times 16 \times 3 = 1264 \). The number of Boolean maps is set to \( c = 4 \), and the total dimension of BMRs is \( 3d = 2256 \) for gray videos, and 3792 for color videos, and the sampling step \( \delta = 1/c = 0.25 \). The search radius for positive samples is set to \( \alpha = 3 \). The inner search radius
for negative samples is set to 0.3min(w, h), where w and h are the weight and height of the target, respectively, and the outer search radius β = 100, where the search step is set to 5, which generates a small subset of negative samples. The target state parameter set for particle filter is set to [σ_1, σ_2] = [6, 6.001], and the number of particles is set to N_p = 400. The confidence threshold is set to ρ = 0.9. All parameter values are fixed for all sequences and the source code will be made available to the public. More results and videos are available at http://kaihuazhang.net/bmr/bmr.htm.

3.2. Datasets and evaluation metrics

(1) The OTSB50 Benchmark: For performance evaluation, we use the tracking benchmark dataset and code library [28] which includes 29 trackers and 50 fully-annotated videos. In addition, we also add the corresponding results of 6 most recent trackers including DLT [49], DSSST [50], KCF [26], TGPR [51], MEEM [16], and HCF [34]. For detailed analysis, the sequences are annotated with 11 attributes based on different challenging factors including low resolution (LR), in-plane rotation (IPR), out-of-plane rotation (OPR), scale variation (SV), occlusion (OCC), deformation (DEF), background clutter (BC), illumination variation (IV), motion blur (MB), fast motion (FM), and out-of-view (OV).

We quantitatively evaluate the trackers with success and precision plots [28]. Given the tracked bounding box B_t and the ground truth bounding box B_g, the overlap score is defined as score = Area(B_t ∩ B_g) / Area(B_t ∪ B_g). Hence, 0 ≤ score ≤ 1 and a larger value of score means a better performance of the evaluated tracker. The success plot demonstrates the percentage of frames with score > t through all threshold t ∈ [0, 1]. Furthermore, the area under curve (AUC) of each success plot serves as a measure to rank the evaluated trackers. On the other hand, the precision plot shows the percentage of frames whose tracked locations are within a given threshold distance (i.e., 20 pixels in [28]) to the ground truth. Both success and precision plots are used in the one-pass evaluation (OPE), temporal robustness evaluation (TRE), and spatial robustness evaluation (SRE) where OPE reports the average precision or success rate by running the trackers through a test sequence with initialization from the ground truth position, and TRE as well as SRE measure a tracker’s robustness to initialization with temporal and spatial perturbations, respectively [28]. We report the OPE, TRE, and SRE results. For presentation clarity, we only present the top 10 algorithms in each plot.

(2) The VOT2016 Benchmark: This dataset includes 60 sequences with polygon annotations for each frame. It contains the evaluated results of 70 trackers including recently published state-of-the-art ones. Three primary measures are employed to analyze the tracking performance including accuracy, robustness, and expected average overlap (EAO). The accuracy is the average overlap between the estimated and ground truth bounding boxes during successful tracking periods while the robustness measures how many times the tracker loses the target during tracking. The EAO estimates the average overlap that a tracker is expected to attain on a large collection of short-term sequences with the same visual properties as the given dataset. For presentation clarity, we only present the top 6 algorithms in terms of EAO.

3.3. Empirical results of OTSB50

(1) Overall Performance: Fig. 3 shows overall performance of the top 10 trackers in terms of success and precision plots. The BMR-based tracking algorithm ranks first on the success rate of all OPE, and second based on TRE and SRE. Furthermore, the BMR-based method ranks third based on the precision rates of OPE, TRE, and SRE. Overall, the proposed BMR-based tracker performs favorably against the state-of-the-art methods in terms of all metrics except for MEEM [16] and HCF [34]. The MEEM tracker exploits a multi-expert restoration scheme to handle the drift problem, which combines a tracker and the historical snapshots as experts. In contrast, even using only a logistic regression classifier without using any restoration strategy, the proposed BMR-based method performs well against MEEM in terms of most metrics (i.e., the success rates of the BMR-based method outperform the MEEM scheme while the precision rates of the BMR-based method are comparable to the MEEM scheme), which shows the effectiveness of the proposed representation scheme for visual tracking. In addition, the HCF method is based on deep learning, which leverages complex hierarchical convolutional features learned off-line from a large dataset and correlation filters for visual tracking. Notwithstanding, the proposed BMR-based algorithm performs comparably against HCF in terms of success rates on all metrics.

(2) Attribute-based Performance: To demonstrate the strength and weakness of BMR, we further evaluate the 35 trackers on videos with 11 attributes categorized by [28].

Tables 1 and 2 summarize the results of success and precision scores of OPE with different attributes. Among them, the BMR-based method ranks within top 3 with most attributes. Specifically, with the success rate of OPE, the BMR-based method ranks first on 4 out of 11 attributes while second on 6 out of 11 attributes. In the sequences with the BC attribute, the BMR-based method ranks third and its score is close to the MEEM scheme that ranks second (0.555 vs. 0.569). For the precision scores of OPE, the BMR-based method ranks second on 4 out of 11 attributes and third on 3 out of 11 attributes. In the sequences with the OV attribute, the BMR-based tracker ranks first, and for the videos with the IPR and BC attributes, the proposed tracking algorithm ranks fourth with comparable performance to the third-rank DSST and KCF methods.

Tables 3 and 4 show the results of TRE with different attributes. The BMR-based method ranks within top 3 with most attributes. In terms of success rates, the BMR-based method ranks first on 2 attributes, second on 3 attributes and third on 6 attributes. In terms of precision rates, the BMR-based tracker ranks third on 7 attributes, and first and second on the OV and OCC attributes, respectively. Furthermore, for other attributes such as LR and BC, the BMR-based tracking algorithm ranks fourth but its scores are close to the results of MEEM and KCF that rank third (0.581 vs. 0.598, and 0.772 vs. 0.776).

Tables 5 and 6 show the results of SRE with different attributes. In terms of success rates, the rankings of the BMR-based method are similar to those based on TRE except for the IPR and OPR attributes. Among them, the BMR-based tracker ranks third based on SRE and second based on TRE. Furthermore, although the MEEM method ranks higher than the BMR-based tracker in most attributes, the differences of the scores are within 1%. In terms of precision rates, the BMR-based algorithm ranks within top 3 with most attributes except for the LR, DEF, and IV attributes.

The AUC score of success rate measures the overall performance of each tracking method [28]. Fig. 3 shows that the BMR-based method achieves better results in terms of success rates than that precision rates in terms of all metrics (OPE, SRE, TRE) and attributes. The tracking performance can be attributed to two factors. First, the proposed method exploits a logistic regression classifier with explicit feature maps, which efficiently determines the nonlinear decision boundary through online training. Second, the online classifier parameter update scheme in [10] facilitates recovering from tracking drift.

Fig. 4 shows sampled tracking results from six long sequences (each with more than 1000 frames). The total number of frames of these sequences is 11,134 that accounts for about 38.4% of the total number of frames (about 29,000) in the benchmark, and hence the performance on these sequences plays an important
Fig. 3. Success and precision plots of OPE, TRE, and SRE by the top 10 trackers. The trackers are ranked by the AUC scores (shown in the legends) when the success rates are used, or precision cores at the threshold of 20 pixels.

### Table 1
Success score of OPE with 11 attributes. The number after each attribute name is the number of sequences. The red, blue and green fonts indicate the best, second and third performance.

| Attribute | BMR | HCF [34] | MEEM [16] | KCF [26] | DSST [50] | TGPR [51] | SCM [31] |
|-----------|-----|----------|----------|----------|----------|----------|--------|
| LR (4)    | 0.409 | 0.557 | 0.360 | 0.310 | 0.352 | 0.470 | 0.279 |
| IPR (31)  | 0.557 | 0.582 | 0.536 | 0.497 | 0.532 | 0.479 | 0.458 |
| OPR (39)  | 0.590 | 0.587 | 0.558 | 0.496 | 0.494 | 0.485 | 0.470 |
| SV (28)   | 0.586 | 0.531 | 0.498 | 0.427 | 0.451 | 0.418 | 0.518 |
| OCC (29)  | 0.615 | 0.606 | 0.552 | 0.513 | 0.480 | 0.484 | 0.487 |
| DEF (19)  | 0.594 | 0.626 | 0.560 | 0.533 | 0.474 | 0.510 | 0.448 |
| BC (21)   | 0.555 | 0.623 | 0.569 | 0.533 | 0.492 | 0.522 | 0.450 |
| IV (25)   | 0.551 | 0.560 | 0.533 | 0.494 | 0.506 | 0.484 | 0.473 |
| MB (12)   | 0.559 | 0.616 | 0.641 | 0.499 | 0.458 | 0.434 | 0.298 |
| FM (17)   | 0.559 | 0.578 | 0.555 | 0.461 | 0.433 | 0.396 | 0.296 |
| OV (6)    | 0.616 | 0.575 | 0.606 | 0.550 | 0.490 | 0.442 | 0.361 |

### Table 2
Precision scores of OPE with 11 attributes. The number after each attribute name is the number of sequences. The red, blue and green fonts indicate the best, second and third performance.

| Attribute | BMR | HCF [34] | MEEM [16] | KCF [26] | DSST [50] | TGPR [51] | SCM [31] |
|-----------|-----|----------|----------|----------|----------|----------|--------|
| LR (4)    | 0.517 | 0.897 | 0.490 | 0.379 | 0.534 | 0.538 | 0.305 |
| IPR (31)  | 0.776 | 0.868 | 0.800 | 0.725 | 0.790 | 0.675 | 0.597 |
| OPR (39)  | 0.819 | 0.869 | 0.840 | 0.730 | 0.732 | 0.678 | 0.618 |
| SV (28)   | 0.803 | 0.880 | 0.783 | 0.680 | 0.740 | 0.620 | 0.672 |
| OCC (29)  | 0.846 | 0.877 | 0.799 | 0.749 | 0.725 | 0.675 | 0.640 |
| DEF (19)  | 0.802 | 0.881 | 0.846 | 0.741 | 0.657 | 0.691 | 0.586 |
| BC (21)   | 0.742 | 0.885 | 0.797 | 0.725 | 0.694 | 0.717 | 0.578 |
| IV (25)   | 0.742 | 0.844 | 0.766 | 0.729 | 0.741 | 0.671 | 0.594 |
| MB (12)   | 0.755 | 0.844 | 0.715 | 0.650 | 0.603 | 0.537 | 0.339 |
| FM (17)   | 0.758 | 0.790 | 0.742 | 0.602 | 0.562 | 0.493 | 0.333 |
| OV (6)    | 0.773 | 0.695 | 0.727 | 0.649 | 0.533 | 0.505 | 0.429 |
Table 3  
Success scores of TRE with 11 attributes. The number after each attribute name is the number of sequences. The red, blue and green fonts indicate the best, second and third performance.

| Attribute | BMR  | HCF  | MEEM | KCF  | DSST | TGPR | SCM  |
|-----------|------|------|------|------|------|------|------|
| LR (4)    | 0.444| 0.520| 0.424| 0.382| 0.403| 0.443| 0.304|
| IPR (31)  | 0.562| 0.591| 0.558| 0.520| 0.515| 0.514| 0.453|
| OPR (39)  | 0.578| 0.595| 0.572| 0.531| 0.507| 0.523| 0.480|
| SV (28)   | 0.564| 0.544| 0.517| 0.488| 0.473| 0.468| 0.496|
| OCC (29)  | 0.585| 0.610| 0.566| 0.547| 0.519| 0.520| 0.502|
| DEF (19)  | 0.599| 0.631| 0.611| 0.571| 0.548| 0.577| 0.515|
| BC (21)   | 0.575| 0.631| 0.577| 0.530| 0.518| 0.530| 0.469|
| IV (25)   | 0.555| 0.597| 0.564| 0.528| 0.529| 0.518| 0.475|
| MB (12)   | 0.537| 0.594| 0.553| 0.493| 0.472| 0.483| 0.290|
| FM (17)   | 0.516| 0.560| 0.542| 0.456| 0.429| 0.461| 0.282|
| OV (6)    | 0.593| 0.557| 0.581| 0.539| 0.505| 0.440| 0.344|

Table 4  
Precision scores of TRE with 11 attributes. The number after each attribute name is the number of sequences. The red, blue and green fonts indicate the best, second and third performance.

| Attribute | BMR  | HCF  | MEEM | KCF  | DSST | TGPR | SCM  |
|-----------|------|------|------|------|------|------|------|
| LR (4)    | 0.581| 0.750| 0.589| 0.501| 0.574| 0.602| 0.350|
| IPR (31)  | 0.767| 0.851| 0.802| 0.728| 0.725| 0.716| 0.581|
| OPR (39)  | 0.759| 0.859| 0.826| 0.749| 0.719| 0.728| 0.617|
| SV (28)   | 0.769| 0.840| 0.787| 0.727| 0.717| 0.676| 0.633|
| OCC (29)  | 0.791| 0.854| 0.788| 0.758| 0.726| 0.705| 0.633|
| DEF (19)  | 0.798| 0.889| 0.854| 0.757| 0.723| 0.765| 0.635|
| BC (21)   | 0.772| 0.874| 0.793| 0.776| 0.697| 0.721| 0.600|
| IV (25)   | 0.747| 0.851| 0.792| 0.729| 0.727| 0.683| 0.585|
| MB (12)   | 0.720| 0.785| 0.724| 0.626| 0.597| 0.607| 0.323|
| FM (17)   | 0.681| 0.738| 0.710| 0.578| 0.532| 0.582| 0.302|
| OV (6)    | 0.719| 0.692| 0.692| 0.643| 0.587| 0.514| 0.371|

Table 5  
Success scores of SRE with 11 attributes. The number after each attribute name is the number of sequences. The red, blue and green fonts indicate the best, second and third performance.

| Attribute | BMR  | HCF  | MEEM | KCF  | DSST | TGPR | SCM  |
|-----------|------|------|------|------|------|------|------|
| LR (4)    | 0.532| 0.488| 0.474| 0.289| 0.328| 0.332| 0.254|
| IPR (31)  | 0.457| 0.537| 0.494| 0.450| 0.460| 0.438| 0.399|
| OPR (39)  | 0.510| 0.536| 0.514| 0.445| 0.439| 0.455| 0.396|
| SV (28)   | 0.524| 0.492| 0.463| 0.401| 0.413| 0.396| 0.438|
| OCC (29)  | 0.524| 0.543| 0.510| 0.445| 0.434| 0.449| 0.398|
| DEF (19)  | 0.492| 0.566| 0.516| 0.469| 0.434| 0.504| 0.358|
| BC (21)   | 0.500| 0.569| 0.517| 0.483| 0.451| 0.483| 0.387|
| IV (25)   | 0.486| 0.516| 0.490| 0.442| 0.446| 0.438| 0.389|
| MB (12)   | 0.503| 0.565| 0.513| 0.425| 0.389| 0.420| 0.266|
| FM (17)   | 0.504| 0.534| 0.518| 0.415| 0.381| 0.412| 0.269|
| OV (6)    | 0.578| 0.526| 0.575| 0.455| 0.426| 0.391| 0.335|

Table 6  
Precision scores of SRE with 11 attributes. The number after each attribute name is the number of sequences. The red, blue and green fonts indicate the best, second and third performance.

| Attribute | BMR  | HCF  | MEEM | KCF  | DSST | TGPR | SCM  |
|-----------|------|------|------|------|------|------|------|
| LR (4)    | 0.476| 0.818| 0.513| 0.377| 0.542| 0.501| 0.305|
| IPR (31)  | 0.704| 0.839| 0.752| 0.667| 0.703| 0.648| 0.546|
| OPR (39)  | 0.732| 0.828| 0.774| 0.666| 0.889| 0.669| 0.547|
| SV (28)   | 0.725| 0.832| 0.732| 0.632| 0.696| 0.599| 0.598|
| OCC (29)  | 0.735| 0.815| 0.730| 0.662| 0.671| 0.649| 0.540|
| DEF (19)  | 0.684| 0.835| 0.757| 0.677| 0.630| 0.715| 0.475|
| BC (21)   | 0.702| 0.851| 0.734| 0.693| 0.655| 0.698| 0.521|
| IV (25)   | 0.677| 0.809| 0.707| 0.652| 0.681| 0.630| 0.509|
| MB (12)   | 0.686| 0.807| 0.691| 0.567| 0.532| 0.561| 0.309|
| FM (17)   | 0.685| 0.748| 0.694| 0.545| 0.505| 0.544| 0.308|
| OV (6)    | 0.719| 0.644| 0.690| 0.533| 0.504| 0.451| 0.386|
Fig. 4. Screenshots sampled results from six long sequences *sylyvester*, *mhyang*, *dog1*, *lemming*, *liquor*, and *doll*.

role in performance evaluation. For clear presentation, only the results of the top performing BMR, HCF, and MEEM methods are shown. In all sequences, the BMR-based tracker is able to track the targets stably over almost all frames. However, the HCF scheme drifts away from the target objects after a few frames in the *Sylvester* (#1178, #1285, #1345) and *lemming* (#386, #1137, #1336) sequences. The MEEM method drifts to background when severe occlusions happen in the *liquor* sequence (#508, #728, #775). To further demonstrate the results over all frames clearly, Fig. 5 shows the plots in terms of overlap score of each frame. Overall, the BMR-based tracker performs well against the HCF and MEEM methods in most frames of these sequences.

3.4. Empirical results of VOT2016

Table 7 summarizes the results of the top 6 trackers among the 70 evaluated trackers on the VOT2016 benchmark ranked by EAO. Due to the diverse characteristics of the VOT2016 dataset, the top trackers come from either the correlation filter methods (CCOT [52], Staple [53]) or the deep ConvNets (TCNN [54], SSAT [55], MLDF [56]). Our proposed BMR ranks fifth with the EAO score 0.2928. As reported by Kristan et al. [42], a strict state-of-the-art bound is 0.251 under the EAO metric, and the trackers whose EAO scores exceed this bound can be regarded as state-of-the-art. Therefore, our BMR can be considered as state-of-the-art. Note that all the top four trackers are based on ConvNets, whose tracking speeds are lower than ours. For fair comparison, we run the codes provided by the authors on the VOT2016 dataset and perform speed measurements on a single CPU with Intel Core i7 3.4 GHz and 16 G RAM standard desktop. Further, our BMR tracker achieves the second best overlap score of accuracy of 0.55 with the first accuracy rank while maintaining a competitive third-placed robustness rank. Noted that although HCF outperforms our BMR in terms of TRE and SRE on OTB50, it performs worse than our BMR on VOT2016 with a EAO score of only 0.22.

Fig. 6 shows the results of the evaluated trackers in terms of overlap scores and failures for each sequence, respectively. Among the 60 sequences, our BMR achieves competitive results among which it ranks first on 12 sequences, second on 16 sequences, and third on 9 sequences in term of overlap score while achieving less failures with first place on 28 sequences, second place on 16 sequences, and third place on 2 sequences.
Fig. 5. Overlap score plots of six long sequences shown in Fig. 4.

Table 7
Results of the evaluated trackers on the VOT2016 benchmark in terms of EAO, accuracy rank (A-Rank), overlap, robustness rank (R-Rank), failures, and tracking speed. The red ranks first, the blue ranks second and the green ranks third.

| Tracker | EAO   | A-Rank | Overlap | R-Rank | Failures | fps  |
|---------|-------|--------|---------|--------|----------|------|
| CCOT    | 0.3279| 1      | 0.54    | 1      | 51.00    | 0.53 |
| TCNN    | 0.3217| 1      | 0.55    | 2      | 57.33    | 1.15 |
| SSAT    | 0.3178| 1      | 0.58    | 2      | 62.47    | 0.74 |
| MLDF    | 0.3076| 3      | 0.49    | 1      | 50       | 2.12 |
| BMR     | 0.2928| 1      | 0.55    | 2      | 67.40    | 5.26 |
| Staple  | 0.2924| 1      | 0.55    | 6      | 80.00    | 15.25|
| HCF     | 0.2200| 6      | 0.374   | 3      | 85.00    | 1.25 |
3.5. Analysis of BMR

(1) Computational Complexity Analysis: As in [57], we employ the fast HOGs whose computational complexity is approximated as \( O(6n^2) \) with the size of the image patch \( n = 32 \) in our experiments. Afterwards, the computational complexity of binarizing the HOG maps and image color channels is about \( O(3 \times 31n^2/64 + 3 \times 3n^2/4) = O(237n^2/64) \). Thus, the total computational complexity of extracting features for each sample is about \( O(621/64n^2) \). Moreover, we have 50 positive samples and 100 negative samples in training and have 400 samples in test. Therefore, the total computational complexity of extracting features from these samples is about \( O(5, 336n^2) \). Furthermore, the computational complexity of updating the parameters via (10) is about \( O(150 \times 20 \times 237n^2/64) \approx O(11, 109n^2) \). So, the upper bound of the total computational complexity for each frame is about \( O(5, 336n^2 + 11, 109n^2) = O(16, 445n^2) \) when assuming that the parameters are updated in each frame. In summary, we set \( n = 32 \) in our experiments, and the total computational complexity of our method is about \( 16, 445n^2 = 16, 839, 680 \) multiplying operations.

In the VOT2016 challenges, all of the leading deep learning based methods including CCOT, TCNN, SSAT, MLDF and HCF shown in Table 7 leverage the pre-trained VGG network architecture [58] to extract multi-layer convolutional features. Their computational complexity of extracting features can be approximated as \( O(\sum_{i=1}^{l-1} w_i^2 c_i p_i^2 c_{i+1}) \), where \( l \) denotes the number of layers, the size of the \( i \)-th layer filter is \( w_i \times w_i \), \( c_i \) denotes the number of the \( i \)-th layer feature channels, and the size of the \( i \)-th layer feature map is \( p_i \times p_i \). The VGG net [58] sets \( p_1 = 224 \), \( c_1 = 64 \), \( c_2 = 128 \) and \( w_1 = 3 \), and hence only the computational complexity of the first-layer is \( w_1^2 c_1 p_1^2 c_2 = 3, 699, 376 \) multiplying operations, which are much larger than the total computational complexity of our method 16,839,680 multiplying operations.

(2) Ablative Studies: To demonstrate the effectiveness of BMRs, we eliminate the component of Boolean maps in the proposed tracking algorithm and only leverage the LAB+HOG representations for visual tracking. In addition, we use the KCF as a baseline as it adopts the HOG representations as the proposed tracking method. Fig. 7 shows quantitative comparisons on the OTB50 dataset. Without using the proposed Boolean maps, the AUC score of success rate in OPE of the proposed method is reduced by 7.5%. For TRE and SRE, the AUC scores of the proposed method are reduced by 3.9% and 4.7%, respectively, without the component of Boolean maps. It is worth noticing that the proposed method, without us-

![Fig. 6. Results of overlap and failures for each sequence in the VOT2016 benchmark.](image-url)
Fig. 7. Success and precision plots of OPE, TRE, and SRE for BMR, BMR only with LAB+HOG representations, and KCF (KCF is used as a baseline for comparisons).

Fig. 8. Number of frames for each sequence in OTB50 that need to update \( w \). The number after each sequence name is the number of frames in that sequence.

(3) Parameter Analysis: From the above Section (1) Computational Complexity Analysis, we can observe that updating the parameters via (10) for each frame takes about \( 11,10^2 \) multiplying operations, and hence updating for each frame is time-consuming. Fortunately, as illustrated by Fig. 8 that shows the number of updated frames for each sequence in OTB50, most of frames in a sequence do not trigger the update procedure.

Moreover, we conduct extensive experiments on OTB50 and VOT2016 to analyze the effect of the number of particles. Fig. 9 shows the results of AUC and EAO scores vs. the number of particles for our method on OTB50 and VOT2016, respectively. We can observe that when the number of particles reaches 200, the AUC score reaches 0.607 and the EAO score reaches 0.2925, and both the AUC and EAO scores do not change much after the particle number reaches 200. Therefore, we can set the particle number = 200, which can further reduce the computational complexity of our method by half.

3.6. Failure cases

Fig. 10 shows failed results of the proposed BMR-based method in two sequences singer2 and motorRolling. In the singer2 sequence, the foreground object and background scene are similar due to the dim stage lighting at the beginning (#10, #25). The HCF, MEEM and proposed methods all drift to the background. Furthermore, as the targets in the motorRolling sequence undergo from 360-degree
Acknowledgment

This work is supported in part by Natural Science Foundation of Jiangsu Province under grant BK20151529 and under grant BK20170040, in part by and Six talent peaks project in Jiangsu Province under Grant R2017L07, in part by NSFC under grand 61532009, in part by the NSFC under Grant Nos. U1713208 and 61472187, the 973 Program No.2014CB349303, and Program for Changjiang Scholars.

4. Conclusions

In this paper, we propose a Boolean map based representation which exploits the connectivity cues for visual tracking. In the BMR scheme, the HOG and raw color feature maps are decomposed into a set of Boolean maps by uniformly thresholding the respective channels. These Boolean maps are concatenated and normalized to form a robust representation, which approximates an explicit feature map of the intersection kernel. A logistic regression classifier with the explicit feature map is trained in an online manner that determines the nonlinear decision boundary for visual tracking. Extensive evaluations on two tracking benchmark datasets demonstrate the proposed tracking algorithm performs favorably against the state-of-the-art algorithms in terms of accuracy and robustness.

References

[1] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, A.V.D. Hengel, A survey of appearance models in visual object tracking, ACM Trans. Intel. Syst. Technol. 4 (4) (2013) 58.
[2] L. Zhang, J. Varadarajan, P. Nagarathnam Suganthan, N. Ahuja, P. Moulin, Robust visual tracking using oblique random forests, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5589–5598.
[3] T. Zhang, C. Xu, M.-H. Yang, Multi-task correlation particle filter for robust object tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1, 2017, p. 3.
[4] K. Moshgi, S.-i. Maeda, S. Oba, H. Skibbe, Y.-z. Li, S. Ishii, An occlusion-aware particle filter tracker to handle complex and persistent occlusions, Comput. Vis. Image Underst. 150 (2016) 81–94.
[5] L. Zhang, P.N. Suganthan, Robust visual tracking via co-trained Kernelized correlation filters, Pattern Recognit. 69 (2017) 82–93.
[6] S. Chan, X. Zhou, J. Li, S. Chen, Adaptive compressive tracking based on locality sensitive histograms, Pattern Recognit. 72 (2017) 517–531.
[7] S. Ono, D. Suhonov, S. Avidan, Best-buddies tracking, arXiv:1611.00148 (2016).
[8] K. Zhang, H. Song, Real-time visual tracking via online weighted multiple instance learning, Pattern Recognit. 46 (1) (2013) 397–411.
[9] S. Zhang, H. Yao, X. Sun, X. Lu, Sparse coding based visual tracking: review and experimental comparison, Pattern Recognit. 46 (7) (2013) 1772–1788.
[10] B. Zhong, H. Yao, S. Chen, R. Ji, T.-J. Chin, H. Wang, Visual tracking via weakly supervised learning from multiple imperfect oracles, Pattern Recognit. 47 (3) (2014) 1395–1410.

Fig. 9. Effect of number of particles for BMR on OTB50 and VOT2016.

Fig. 10. Failure cases of the BMR-based tracker in the singer2 and motorRolling sequences. The results of HCF and MEEM are also illustrated.
Kaihua Zhang is a professor in the School of Information and Control, Nanjing University of Information Science & Technology, Nanjing, China. He received the B.S. degree in Technology and Science of Electronic Information from Ocean University of China (OUC) in 2006, the M.S. degree in Signal and Information Processing from the University of Science and Technology of China (USTC) in 2009 and Ph.D. degree from the Department of Computing in the Hong Kong Polytechnic University in 2013. From Aug. 2009 to Aug. 2010, he worked as a Research Assistant in the Department of Computing, The Hong Kong Polytechnic University. His research interests include image segmentation, level sets, and visual tracking.

Qingshan Liu is a Professor with the School of Information and Control, Nanjing University of Information Science and Technology, Nanjing, China. He received the Ph.D. degree from the National Laboratory of Pattern Recognition, Chinese Academic of Science, Beijing, China, in 2003, and the M.S. degree from the Department of Auto Control, Southeast University, Nanjing, in 2000. He was an Assistant Research Professor with the Department of Computer Science, Computational Biomedicine Imaging and Modeling Center, Rutgers, The State University of New Jersey, New Brunswick, NJ, USA, from 2010 to 2011. Before he joined Rutgers University, he was an Associate Professor with the National Laboratory of Pattern Recognition, Chinese Academic of Science, and an Associate Researcher with the Multimedia Laboratory, Chinese University of Hong Kong, Hong Kong, from 2004 and 2005. He was a recipient of the President Scholarship of the Chinese Academy of Sciences in 2003. His current research interests are image and vision analysis, including face image analysis, graph and hypergraph-based image and video understanding, medical image analysis, and event-based video analysis.

Jian Yang received the BS degree in mathematics from the Xuzhou Normal University in 1995. He received the MS degree in applied mathematics from the Changsha Railway University in 1998 and the Ph.D. degree from the Nanjing University of Science and Technology (NUST), on the subject of pattern recognition and intelligence systems in 2002. In 2003, he was a postdoctoral researcher at the University of Zaragoza. From 2004 to 2006, he was a Postdoctoral Fellow at Biometrics Centre of Hong Kong Polytechnic University. From 2006 to 2007, he was a Postdoctoral Fellow at Department of Computer Science of New Jersey Institute of Technology. Now, he is a professor in the School of Computer Science and Technology of NUST. He is the author of more than 80 scientific papers in pattern recognition and computer vision. His journal papers have been cited more than 2000 times in the ISI Web of Science, and 4000 times in the Web of Scholar Google. His 2DPCA paper published in TPAMI 2004 has been cited more than 2000 in Scholar Google. His research interests include pattern recognition, computer vision and machine learning. Currently, he is an associate editor of Pattern Recognition Letters and IEEE Trans. Neural Networks and Learning Systems, respectively.

Ming-Hsuan Yang is a professor in Electrical Engineering and Computer Science at University of California, Merced. He received the Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign in 2000. Prior to joining UC Merced in 2008, he was a senior research scientist at the Honda Research Institute working on vision problems related to humanoid robots. He coauthored the book Face Detection and Gesture Recognition for Human-Computer Interaction (Kluwer Academic 2001) and edited special issue on face recognition for Computer Vision and Image Understanding in 2003, and a special issue on real world face recognition for IEEE Transactions on Pattern Analysis and Machine Intelligence. Yang served as an associate editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence from 2007 to 2011, and is an associate editor of the International Journal of Computer Vision, Image and Vision Computing and Journal of Artificial Intelligence Research. He received the NSF CAREER award in 2012, the Senate Award for Distinguished Early Career Research at UC Merced in 2011, and the Google Faculty Award in 2009. He is a senior member of the IEEE and the ACM.