Multimedia Tools and Applications (2020) 79:4965–4988
https://doi.org/10.1007/s11042-018-7135-6

Object tracking with collaborative extreme learning machines

Haipeng Kuang[1] · **Liang Xun**[2]

Received: 31 October 2018 / Revised: 10 December 2018 / Accepted: 26 December 2018 /
Published online: 21 January 2019

Abstract

We propose a novel collaborative discriminative model based on extreme learning machine (ELM) for object tracking in this paper. In order to represent the object more precisely, we first propose a new collaborative discriminative representation model, which includes both a global discriminative sub-model and a local discriminative sub-model. Different from traditional local representation models, in particular, our local sub-model integrates several classifiers which have structural relations to improve the expression. The global discriminative model represents the appearance comprehensively while the local discriminative sub-model can effectively address occlusions and assist the update. Second, to have better combination of these sub-models, we propose a novel collaboration strategy based on the Kullback-Leibler (KL) distance. The novel strategy can determine the weights of the sub-models adaptively by measuring their KL distances reciprocally. Third, we introduce ELM into tracking and adopt it to build both the global and the local discriminative sub-models simultaneously. Since ELM has a good generalization performance and is robust to the imbalance of the training samples, it is suitable to be used for tracking. Experimental results demonstrate that our method can achieve comparable performance to many state-of-the-art tracking approaches.

Keywords Object tracking · Collaborative model · Extreme learning machine · Kullback-Laibler distance

✉ Haipeng Kuang
kuanghp@163.com

Liang Xun
xunliang@gmail.com

[1] Key Laboratory of Airborne Optical Imaging and Measurement, Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun, 130033, China

[2] Beijing Topmoo Technologies Co., Ltd, Research and Development Plaza, Tsinghua Science Park, Haidian District, Beijing, 100084, China

1 Introduction

Object tracking is a hot topic in computer vision field and has a wide range of applications, such as surveillance, human-computer interface (HCI), video editing, motion analysis, etc. However, it is still a challenging task due to some complex factors, such as occlusions, illumination variations, pose changes of the objects, fast motion, etc. [39, 43].

Recently, more and more works focus on how to build accurate appearance model to represent the object, since a more accurate model seems to lead to more reliable tracking performance. The research of the appearance model has located in the kernel of the tracking. As a whole, the appearance models can be divided into two types. The first is the generative model [1, 6, 21, 27, 32, 37], which mainly uses the information of the object. The second is the discriminative model [2, 3, 51–53], which always formulates tracking as a binary classification problem and takes use of the information of both the target and the background. Moreover, some collaborative models based on the combination of the above models have been proposed as well.

According to the attributes of the features, these models fall into two types: the global model [2, 12, 28, 32] and the local model [1, 3, 17, 24]. The representation with the global model always gives an overall description of the object and can reserve the structure of the object. Yet it is sensitive to occlusions. Conversely, the local model always divides the object into several patches or blocks, which is more robust to occlusions and can fit the changes of the appearance more easily, but it is sensitive to large appearance deformation. Both representation models can be used in either generative or discriminative manner, which generates different tracking methods.

In order to have an accurate representation of the object, in this paper, we propose a novel collaborative representation model, which combines both the global and the local models. In our method, both the global and local sub-models are built as discriminative models. The collaborative model describes the appearance model more precisely and robustly. In our local sub-model, the object is divided into several small patches. As for each patch, we use a corresponding classifier to build the sub-local discriminative model. The final local sub-model is obtained by integrating all of the sub-local models. The global sub-model is built in the similar way as the traditional discriminative model is built. Then the final representation model is built by combining the local sub-model and the global sub-model. Besides, the local discriminative sub-model ia able to detect occlusions and help to update the collaborative model.

How to determine the weights of the sub-models under the collaboration framework is a potential problem, for the representability of the sub-models is not the same. Different from the existing collaborative methods adopting fixed weight parameters, we propose a novel KL distance based collaboration strategy to make the collaboration of the global sub-model and the local sub-model more robust. During tracking, it can be observed that the confidence score map obtained by the classifier is quite similar to the probability distribution. Thus, we employ the KL distance, which is usually used to evaluate the importance of the distribution, to calculate the weights of the sub-models. Because the mutual KL distances between the sub-models are asymmetric, the weights can be determined adaptively.

Since the discriminative models always formulate tracking as a binary classification problem, there is an issue that the positive and the negative samples in tracking are often unbalanced. In this paper, we employ ELM, proposed by Huang [15, 16], as the classifier to build the specific collaborative discriminative model and construct a novel tracking method. ELM is used for single-hidden layer feedforward networks (SLFNs). Huang also proved the good generalization performance of ELM in both theory and practice. ELM has

many advantages, such as robustness to the unbalanced data, powerful classification ability, fast learning speed, etc. Therefore, it is quite suitable to be used for tracking under the tracking-by-detection framework. With the collaborative representation model, we evaluate the performance of our tracker on public benchmark sequences and the results demonstrate that our tracker can achieve state-of-the-art performance.

The main contributions are as the following:

- We propose a novel collaborative discriminative model for tracking, which includes a global sub-model and a local sub-model. In particular, the local sub-model is discriminative that it can address the occlusion problem robustly.
- We propose a novel KL distance based collaboration strategy to combine both the global and the local sub-models. Due to its asymmetry, the KL distance can determine the weights of the sub-model adaptively.
- Based on the novel collaborative formulation, we develop a concrete realization by using ELM as the basic classifier to implement the sub-models, which has the property of non-linearity, robustness to the unbalance of the samples, etc.

The remainder of this paper is organized as follows. In Section 2, we introduce the related work of the global, local and collaborative representations, and give an brief introduction of ELM. In Section 3, we present the proposed collaborative ELM based tracking method, including the collaborative model based on ELM, the KL distance based collaboration strategy, update model, etc. Section 4 displays the experimental results and Section 5 concludes the paper.

2 Related work

2.1 Representation

Representation plays an important role in building a suitable appearance model for object tracking. Both the global and the local representations are widely used in generative or discriminative models based trackers.

Global representation can capture the structure of the object to have a good description, and many techniques, such as subspace learning representation with incremental principal component analysis (PCA) [32], sparse representation [28], compact representation with 3D-DCT [22], etc., have been used in generative models. In the global discriminative model, SVM [2, 50], on-line boosting [12], random forest [33], multiple instance learning [4], structured learning [13] are also introduced to build the discriminative models based on the global representations. Besides, one characteristic that these global discriminative models have in common is that the global representation is implemented by combining local features, such as histogram of oriented gradients (HOG) or Haar features, to improve the robustness. Although the global representation can extract the structural information of the object, they are sensitive to occlusions. Recently, correlation filter [10, 14, 29, 49] and deep learning algorithms [30, 41, 45] are also introduced into tracking. Since the correlation filter can make more full use of the spatial information and deep learning methods have more powerful representation ability, they have achieved good tracking performance. However, most of these methods still use the global information while do not pay enough attentions on the local information, which may affect the tracking performance in complex conditions.

In contrast to the global representation, the local representation describes the object by dividing the object into several patches. Adam et al. [1] build the template with multiple

patches and determine the object's position by voting. However, this method use only the information of the target. Avidan [3] proposes a local discriminative tracking method with low-level features. He takes each pixel as a sample and takes Adaboost as the classifier to judge whether a pixel belongs to the target or the background. The patch strategy is also adopted to build the local sparse appearance model [17, 24, 54]. Due to its powerful ability of addressing occlusion and motion deformation problems, the local representation-based appearance models have attracted more attention recently. For example, Bai et al. [5] take use of the patches of the object to learn a pool of weak classifiers and treat the weight vector as a distribution to construct the classifier ensemble. Although some approaches [42, 46] have attempted to exploit the structures of the local patches, most of the tracking-by-detection-based local models discard the structures of the object, which may lead to the failure of the tracking.

Collaborative model has been also utilized for object tracking [23, 25, 34, 36, 44, 54]. However, most of these methods build the appearance model by combining two discriminative models or a generative model and a discriminative model, but only use the global representation. Besides, some collaborative models make use of both the global and the local models. Sun et al. [35] develop the combination method of scale-invariant feature transform (SIFT) based local description and PCA based global representation methods. However, both of the representations are generative and do not use the background information. Zhong et al. [54] utilize the sparse representation and combine a global sparse model and a local sparse model for tracking. But in their method, the local model is generative and they assign fixed weights to both the sub-models, which may not take into account the adjustment of the weights. Chen et al. [9] propose a hierarchical representation framework for object tracking. They use *cells* to representation local features and integrated them into complex cells to explore various contextual information. However, the hierarchical property is expressed only in the feature level, which can be taken as a specific case with sampling and ensemble of the global model. In our method, we construct a novel collaborative representation model based on the global discriminative sub-model and the local discriminative sub-model, the structural properties of which are in feature level and classifier level simultaneously, to improve the robustness.

2.2 Review of ELM

ELM was proposed by Huang [15] in 2006, which is a novel approach for building the SLFNs. It can be used for both classification and regression with good generalization performance. Hereby, we give a review of ELM briefly.

Assuming that the input samples are $\{\mathbf{x}_j\}$ and the corresponding labels are $\{\mathbf{y}_j\}$, where $\mathbf{x}_j \in \mathbf{R}^n$ and $\mathbf{y}_j \in \mathbf{R}^m$, the standard SLFNs can be modeled as:

$$\sum_{i=1}^{\tilde{N}} \beta_i g(\mathbf{w}_i^T \mathbf{x}_j + \mathbf{b}_i) = \mathbf{y}_j, \qquad j = 1, ..., N, \tag{1}$$

where $\tilde{N}$ is the number of the hidden nodes, $g(\mathbf{x})$ is the activation function, $\mathbf{w}_i$ is the weight vector connecting the ith hidden node and the input nodes, β_i is the weight vector connecting the ith hidden nodes and the output nodes, and $\mathbf{b}_i$ is the bias.

The above (1) can be written as a compact representation:

$$\mathbf{H}\beta = \mathbf{Y}, \tag{2}$$

where $\mathbf{H}_{ji} = g\left(\mathbf{w}_i^T \mathbf{x}_j + \mathbf{b}_i\right)$ is the element of matrix $\mathbf{H}$, $\beta = [\beta_1, \beta_2, ..., \beta_{\tilde{N}}]^T$, and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N]^T$.

Different from the conventional gradient-based solutions of SLFNs, Huang proposed the ELM method, in which the activation function $g(\mathbf{x})$ is set as the infinity differential function, such as the sigmoidal function, and the input weights $\{\mathbf{w}_i\}$ and biases $\{\mathbf{b}_i\}$ are assigned randomly. Once the $\{\mathbf{w}_i\}$ and $\{\mathbf{b}_i\}$ are determined, the training of the SLFN is to find the minimum norm least-squares solution of (2):

$$\hat{\beta} = \mathbf{H}^{+}\mathbf{Y}, \tag{3}$$

where $\mathbf{H}^{+}$ is the Moore-Penrose generalized inverse of $\mathbf{H}$. The detailed proof of this theory is shown in [15]. Since there are no parameters to tune during the training stage, ELM is simple to be realized. It also provides a unified framework for both classification and regression. In the classification problem, it can be trained and used for testing easily and is robust to the imbalance of the input data. Therefore, ELM is quite suitable to be used in tracking-by-detection framework and we employ it to build the appearance model in our method.

3 Collaborative ELMs based tracking

In this section, we first introduce the proposed collaborative representation model and the detailed implementation based on the ELM. With the implemented model, the tracking approach how to determine the final tracking result is displayed. Besides, the occlusion problem is addressed with the collaborative representation and the update strategy is proposed.

3.1 Collaborative representation with ELMs

We propose a collaborative discriminative model to represent the object. The collaborative model contains two parts: the global sub-model and the local sub-model. In our method, both of these two sub-models are discriminative. They both use the information of the target and object. We build classifiers for the global sub-model and local sub-model with ELMs respectively.

First, we introduce the global sub-model ELM_G, which is shown in Fig. 1. Assume the positive sample set is X_G^p and the negative set is X_G^n. Web select the positive and negative samples to build X_G^p and X_G^n according to the distance-based rule. The positive and negative samples and their labels are represented as $\{\mathbf{x}_{Gj}, \mathbf{y}_{Gj}\}(j = 1, ..., N_G)$. In order to improve the robustness of the classifier, ELM adopts another representation for $\mathbf{y}_{Gj}$, in which $\mathbf{y}_{Gj} = [-1, 1]^T$ if $\mathbf{x}_{Gj}$ is positive, and $\mathbf{y}_{Gj} = [1, -1]^T$ if $\mathbf{x}_{Gj}$ is negative. The distance-based rule means that the positive samples are selected near the labeled target, while the negative samples are chosen far away from the target. For example, if $||l(\mathbf{x}_{Gj}) - l(\mathbf{x}_{Gt})||_2 < d_1$, $\mathbf{x}_{Gj}$ will be taken as the positive sample, and if $||l(\mathbf{x}_{Gj}) - l(\mathbf{x}_{Gt})||_2 > d_2$, $\mathbf{x}_{Gj}$ will be considered as the negative sample. Herein, $l(\mathbf{x}_{Gj})$ is the location of $\mathbf{x}_{Gj}$, $l(\mathbf{x}_{Gt})$ is the location of the target, and d_1 and d_2 are two predefined thresholds. In our method, d_1 is set to 2 pixels, while d_2 is set to half of the minimum value of the width and height of the target. Fig. 1a shows how to select the samples. The rectangles with green color are positive while those with blue color are negative. In order to deal with the objects with different sizes conveniently, we

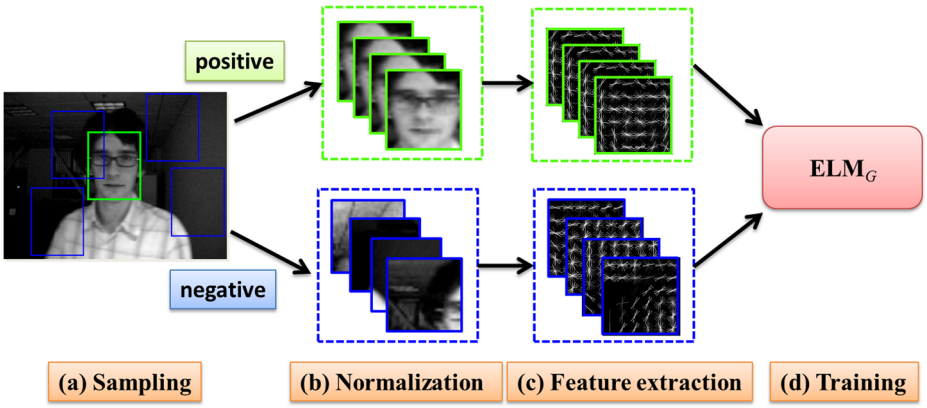


Fig. 1 The global sub-model ELM_G for representation. **a** Select the samples for classification. The regions with green color are taken as positive while the regions with blue color are considered as negative. These samples are selected with the distance-based rule. **b** Normalize the selected samples into a fixed size to simplify the solution. **c** Extract features from the normalized samples. Hereby, the HOG features are extracted. **d** The global sub-model ELM_G is trained based on the positive and negative samples

normalize all of the samples into a fixed size $N_{normG} \times N_{normG}$. With the normalized samples, different feature extraction approaches can be adopted and local features are selected here. After extracting suitable features, we train an ELM classifier to construct the global model. Assuming that the feature corresponding to $\mathbf{x}_{Gj}$ is $\boldsymbol{\phi}(\mathbf{x}_{Gj})$, and its label is $\mathbf{y}_{Gj}$, we assign random value to the weight vector $\mathbf{w}_{Gi}$ and $\mathbf{b}_{Gi}$, and select sigmoid function as the activation function. Then according to (2) and (3), the coefficient $\hat{\beta_G}$ of ELM_G is

$$\hat{\beta}_G = \mathbf{H}_G^{+}\mathbf{Y}_G, \tag{4}$$

where $\mathbf{H}_{Gji} = g\left(\mathbf{w}_{Gi}^T\boldsymbol{\phi}(\mathbf{x}_j) + \mathbf{b}_i\right)$ and $\mathbf{Y}_G = [\mathbf{y}_{G1}, \cdots, \mathbf{y}_{GN_G}]^T$. Thus the ELM_G is implemented with parameters $\{\mathbf{w}_{Gi}, \mathbf{b}_{Gi}\}(i = 1, ..., \tilde{N}_G)$ and $\hat{\beta_G}$.

Next, we introduce the local discriminative sub-model ELM_L, which is shown in Fig. 2. Firstly, we crop the region of interest (ROI) for extracting the positive and negative samples. Instead of using the distance-based rule, in the local sub-model, the region-based rule is adopted, in which the target region is considered as positive, as the green rectangle shows, and the region between the green and blue boundaries is taken as negative. In a similar way to ELM_G, we normalize the ROI into the fixed size $N_{normL} \times N_{normL}$. Note that $N_{normL} = 2N_{normG}$. In order to build the local model, we partition the ROI into several patches $\{\mathbf{p}_l^P\}(l = 1, ..., M_p)$ and $\{\mathbf{p}_l^N\}(l = 1, ..., M_n)$ with compact grids, where $\mathbf{p}_l^P$ is from the target and $\mathbf{p}_l^N$ is from the background. The size of each patch is $N_{norml} \times N_{norml}$. In the local sub-model, each patch region is considered as a sample, and the discriminative model is modeled based on the patches. As Fig. 2b shows, the patches with green color are set to positive while those with blue color are set to negative, which are in the sets X_L^P and X_L^N respectively. In the traditional tracking-by-detection-based local models, a classifier is trained based on the X_L^P and X_L^N. However, this training model neglects the inner structure of the object and mixes the patches. Hereby, we propose a novel local discriminative model ELM_L. Instead of mixing all the positive (negative) patches together to fill X_L^P $\left(X_L^N\right)$, we

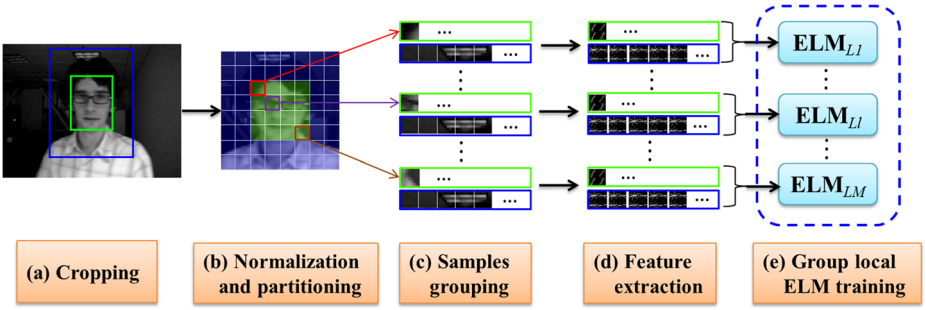


Fig. 2 The local sub-model ELM_L for representation. **a** Select the ROI region with region-based rule by cropping the target and its surrounding region. The region in the green rectangle is positive while the region between the green and blue boundaries is negative. **b** Normalize the ROI into a fixed size and partition the ROI into several patches with fine grids. The patches with green background are positive and the patches with blue background are negative. **c** Divide the patches into several groups. The positive patches in different positions are sent to different groups, and all of the negative patches are sent to all groups respectively. **d** Extract features (HOG) from the groups of samples. **e** Train a group of classifiers with the features based on ELMs. The final local sub-model ELM_L is the combination of the group of ELM_{Ll}

divide the positive (negative) patches into groups. Each group $\mathbf{g}_l$ contains a positive set X_{Ll}^P and a negative set X_{Ll}^N. The X_{Ll}^P has the sample corresponding to $\mathbf{p}_l^P$ of the object, and the X_{Ll}^N contains all the negative patches $\{\mathbf{p}_l^N\}$ with the blue color. It can be seen that the number of positive samples and the number of negative samples are not balanced. After extracting features from the samples, for each group $\mathbf{g}_l$, we build a corresponding local classifier ELM_{Ll}. Denote the weight vectors of ELM_{Ll} as $\{\mathbf{w}_{Lli}\}$ and the biases $\{\mathbf{b}_{Lli}\}(i = 1, ..., \tilde{N}_L)$. By setting $\{\mathbf{w}_{Lli}\}$ and $\{\mathbf{b}_{Lli}\}$ randomly, we obtain the matrix $\mathbf{H}_{Ll}$. We train each ELM_{Ll} model according to (2) and (3) and obtain a group of $\hat{\beta}_{Ll}$. Therefore, the final local sub-model ELM_L is the combination of the group of ELM_{Ll}. In practice, the samples of X_L^P and X_L^N will be organized with the tracking results along with the time axis but with a different update speed, which is shown in Section 3.3.

Then, our collaborative representation model ELM_C is built based on the collaboration of ELM_G and ELM_L. ELM_G gives the global representation with ELM trained by the combination of the local features. ELM_L gives a group of local discriminative ELM classifiers, which have structural relations. With the assigned $\{\mathbf{w}_{Gi}, \mathbf{b}_{Gi}\}(i = 1, ..., \tilde{N}_G)$ of ELM_G, $\{\mathbf{w}_{Lli}, \mathbf{b}_{Lli}\}(i = 1, ..., \tilde{N}_L, l = 1, ..., M)$ of the ELM_L, and the corresponding $\hat{\beta}_G$ and $\{\hat{\beta}_{Ll}\}$, the collaborative model ELM_C can be used to complete the tracking task.

3.2 Particle filter tracking with KL distance based collaboration strategy

With the collaborative representation model ELM_C, we propose a novel tracking algorithm, named the *CET tracker*. We first take the particle filter as the basic motion model to sample several candidate region samples, and calculate the confidence scores of these samples using the ELM_G and ELM_L respectively. Then, we introduce a novel KL distance based strategy to combine the scores together, which assigns weights to these sub-models. At last, the likelihood function is calculated based on the combined scores to determine the final tracking result.

3.2.1 Particle filter

The particle filter has been widely used in tracking field. It always has two steps: the prediction step and the update step,

$$p(S_t|O_{1:t-1}) \propto \int p(S_t|S_{t-1})p(S_{t-1}|O_{1:t-1})dS_{t-1},$$
$$p(S_t|O_t) \propto p(O_t|S_t)p(S_t|O_{1:t-1}), \tag{5}$$

where $\{O_1, O_2, ..., O_t\}$ are observation variables, $p(O_t|S_t)$ is the observation model, and $p(S_t|S_{t-1})$ is the transition model. In our method, the state space S of the target is described with two state variables $\{s_x, s_y\}$, where s_x is the translation in horizontal direction and s_y is the translation in vertical direction. It is easy to add more variables to control the scale or rotation changes if necessary. Hereby, N_p particles are sampled to approximate the state space and the transition model $p(S_t|S_{t-1})$ is supposed to obey the multi-dimensional Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$.

With the above formulation of the particle filter, the main task is to calculate the likelihood function $p(O_t|S_t)$, which can be obtained by sending the particle samples $\{\mathbf{x}_k\}(k = 1, \cdots, N_p)$ into the ELM_C.

3.2.2 Likelihood calculation with the KL distance based collaboration

The likelihood function is acquired by calculating the confidence score of each particle over the ELM_C. As for the global sub-model ELM_G, $\mathbf{x}_k$ is normalized into the same size with the trained samples for the ELM_G, and then we extract the feature $\phi(\mathbf{x}_k)$. By sending $\phi(\mathbf{x}_k)$ into the ELM_G, we obtain

$$\mathbf{y}_{Gk} = \mathbf{h}_{Gk}^T \hat{\beta_G}, \tag{6}$$

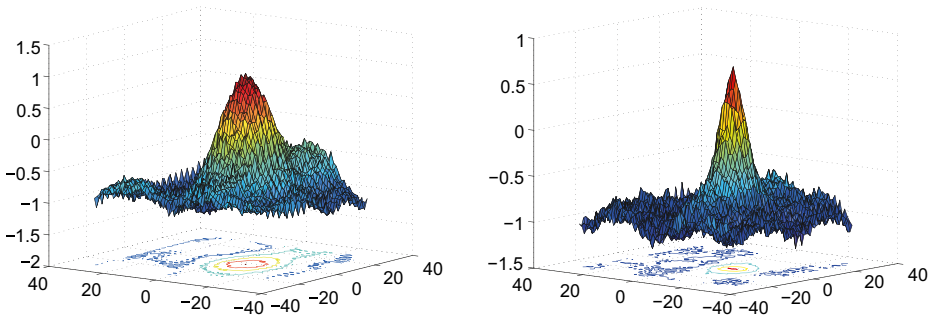
where $\mathbf{h}_{Gk} = \left[g\left(\mathbf{w}_{G1}^T \phi(\mathbf{x}_k) + \mathbf{b}_{G1}\right), \cdots, g\left(\mathbf{w}_{G\tilde{N}}^T \phi(\mathbf{x}_k) + \mathbf{b}_{G\tilde{N}}\right)\right]^T$, $\mathbf{w}_{Gi}$, $\mathbf{b}_{Gi}$ and $\hat{\beta}_G$ are the model parameters of ELM_G, and $\mathbf{y}_{Gk}$ is the vector with dimension size 2. Therefore, as the special case of binary classification of ELM [15], the confidence score of $\mathbf{x}_k$ can be obtained by taking the value of the second element of $\mathbf{y}_{Gk}$:

$$conf_G(k) = \mathbf{y}_{Gk}(2). \tag{7}$$

As for the local sub-model ELM_L, $\mathbf{x}_k$ is divided into several patches $\mathbf{p}_{lk}(l = 1, \cdots, M)$. After taking the similar operation with that in the ELM_G, we send each $\mathbf{p}_{lk}$ of $\mathbf{x}_k$ into the corresponding ELM_{Ll}. Then we get a group of $conf_{Ll}(k)$, according to (7). Since each patch is a fragment of the complete object and represents a part of the structure of the object, we calculate the final confidence score of the ELM_L as

$$conf_L(k) = \frac{1}{M} \sum_{l=1}^{M} conf_{Ll}(k). \tag{8}$$

The ELM_G and ELM_L do not always have the same expression during tracking. Thus, we would like to assign a higher weight to the sub-model with the better expression. Moreover, we have an interesting observation that the confidence map obtained by the candidate samples is very similar to the probability distribution. Figure 3 gives an example of the confidence score maps obtained by the ELM_G and ELM_L over one frame of sequence *davidindoor*. It can be seen that the map shapes of both sub-models are similar to the



(a) Confidence map over ELM_G (b) Confidence map over ELM_L

Fig. 3 Example of the confidence maps obtained over ELM_G and ELM_L (on sequence *davidindoor*) by horizontal and vertical translation. It can be seen that the confidence maps are similar to the probability distributions, but they also have some differences

Gaussian probability distribution, but the *pseu-variance* of the map of ELM_G looks larger than that of ELM_L in that frame. In our option, the smaller the pseu-variance of the map is, the more important the model is, because the steeper confidence map means a better discriminability, possibly leading to more accurate location. Thus, we would like to assign a higher weight to the local sub-model in Fig. 3.

Hereby, we introduce a novel KL distance based strategy to measure the importance of the sub-models. KL distance is an important concept in both probability and information theories, which is used to measure the difference between the probabilities. It has a special property that the KL distance is not symmetric, thus it can determine the weights of the sub-models adaptively according to the expressions of the sub-models. In practice, there often exist two differences between the confidence maps of the sub-models. One is the aforementioned different pseu-variances, and the other one is that the peaks of the maps are not often consistent. Figure 4 shows two examples with different probability distributions. Denote the KL distance of p_2 from p_1 as $D_{KL}(p_1||p_2)$, and that of p_1 from p_2 as $D_{KL}(p_2||p_1)$. In Fig. 4a, if $D_{KL}(p_1||p_2)$ is larger than $D_{KL}(p_2||p_1)$, it implies that p_2 is more important than p_1. Figure 4b gets the same conclusion as well. It can be found that the KL distance can effectively measure the importance of the distributions. Since the KL distance is relative, it can determine the weights by normalization.

Before using KL distance to calculate the weights of the sub-models, we need to scale the confidence scores first to avoid the impact of the negative values. Hereby, we scale the confidence scores of ELM_G with min-max rule to get the $confn_G$:

$$confn_G(k) = \frac{conf_G(k) - \min_i(conf_G(i))}{\max_i(conf_G(i)) - \min_i(conf_G(i))}. \tag{9}$$

Then, scaling the $confn_G(k)$ by dividing the summation, the standard probability can be obtained:

$$p_G(k) = \frac{confn_G(k)}{\sum_{i=1}^{N} confn_G(i)}. \tag{10}$$

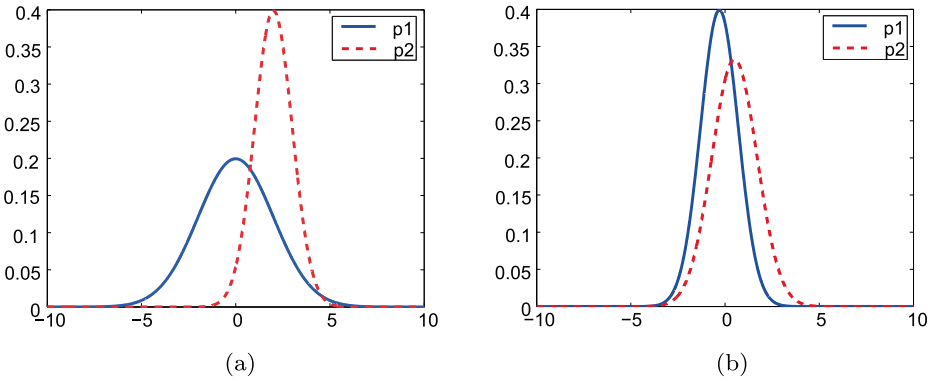


Fig. 4 Two examples of using KL distance to calculate the weights of two different Gaussian distributions. In both **a** and **b**, these two distributions have different means and variances. In **a**, $D_{KL}(p_1 \parallel p_2) = 4.05$ and $D_{KL}(p_2 \parallel p_1) = 1.18$. In **b**, $D_{KL}(p_1 \parallel p_2) = 0.36$ and $D_{KL}(p_2 \parallel p_1) = 0.52$. The weights can be further determined by normalization in inverse proportion according to (12). The probability with a *thinner* shape will be considered better and assigned a larger weight

With the same operation, we can get the normalized confidence score $confn_L$ and probability $p_L(k)$ of the local sub-model. Further, we calculate the KL distance between $p_G(k)$ and $p_L(k)$:

$$\begin{cases} D_{KL}(p_G||p_L) = \sum\limits_k p_G(k) \ln \dfrac{p_G(k)}{p_L(k)}, \\ D_{KL}(p_L||p_G) = \sum\limits_k p_L(k) \ln \dfrac{p_L(k)}{p_G(k)}. \end{cases} \tag{11}$$

The weights α_G and α_L of the sub-models can be defined as

$$\begin{cases} \alpha_G = \dfrac{D_{KL}(p_L||p_G)}{D_{KL}(p_G||p_L) + D_{KL}(p_L||p_G)}, \\ \alpha_L = \dfrac{D_{KL}(p_G||p_L)}{D_{KL}(p_G||p_L) + D_{KL}(p_L||p_G)}. \end{cases} \tag{12}$$

Considering $conf_G(k)$ and $conf_L(k)$, the confidence score of $\mathbf{x}_k$ on the collaborative representation model ELM_C is defined as

$$conf(k) = \alpha_G conf_G(k) + \alpha_L conf_L(k), \tag{13}$$

where α_G and α_L satisfy $\alpha_G + \alpha_L = 1$.

We define the likelihood function $\mathcal{L}(\mathbf{x}_k) = \exp(conf)$, then $p(O_t|S_t) \propto \mathcal{L}(\mathbf{x}_k)$. The optimal candidate sample $\mathbf{x}_{opt}$ and its corresponding state S_t in time t is determined by maximizing a posterior (MAP) rule:

$$\mathbf{x}_{opt} = \arg\max_{\mathbf{x}_k} \mathcal{L}(\mathbf{x}_k). \tag{14}$$

3.3 Model update

In order to cope with the deformation of the object during the tracking process and address the occlusion problem, the appearance model should be updated timely and properly.

The model update is realized by updating the positive and negative samples for both sub-models and retraining all ELMs. We build a positive pool and a negative pool for the positive and negative samples of the global sub-model respectively. For the local sub-model, we build a pool for each positive patch position and a unified pool for all the negative patches. The update of the samples obeys the first-in and first-out (FIFO) rule. Since the surrounding background of the target always changes faster, we update the negative samples all the time by adding new samples into the pool to replace half of earlier ones. Then the key problem is how to update the positive samples.

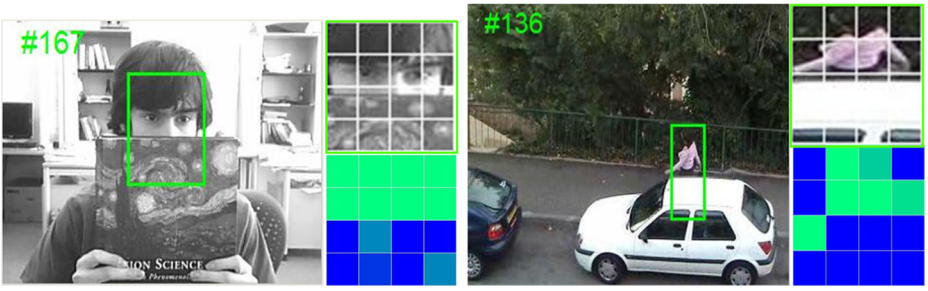
Since the global sub-model is sensitive to the occlusion noises, whether to update the positive samples mainly depends on the local sub-model. For a candidate $\mathbf{x}_k$, each patch $\mathbf{p}_{lk}$ can be classified to be label 1 or -1 by its corresponding ELM_{Ll}. Assuming the output of ELM_{Ll} is $\mathbf{y}_{Llk}$, then the label

$$\hat{\mathbf{y}}_{Llk} = \begin{cases} 1, & if\ \mathbf{y}_{Llk}(2) > \mathbf{y}_{Llk}(1) \\ -1, & otherwise. \end{cases} \tag{15}$$

The samples of each patch pool of the local sub-model are updated separately by their labels $\{\hat{\mathbf{y}}_{Llk}\}$. The samples in the corresponding position will be updated if $\hat{\mathbf{y}}_{Llk} = 1$, and if $\hat{\mathbf{y}}_{Llk} = -1$, they will not be updated for we judge there exists occlusions. The update of samples of the global sub-model depends on the summation

$$\hat{\mathbf{y}}_s = \sum_{l=1}^{M} \hat{\mathbf{y}}_{Llk}. \tag{16}$$

The new sample will be added into the positive pool for updating if $\hat{\mathbf{y}}_s > 0$, and vice versa. This means that, for the obtained optimal candidate sample $\hat{\mathbf{x}}_{opt}$, if there are more negative patches than the positive, we will not sent it to the positive sample pool for updating. Figure 5 gives some examples with our occlusion handling strategy, which indicates that the occlusions can be detected effectively, to help to avoid the wrong update. The detailed tracking process is illustrated in Algorithm 1.



(a) *faceocc2* (b) *woman*

Fig. 5 Some examples of occlusion detection on sequences (a) *faceocc2* and (b) *woman*. For each patch, the darker the green color is, the more possible it is to be positive, while the darker the blue color is, the more possible it is to be negative (or occluded). It can be seen that the occlusions can be effectively detected by our model

Algorithm 1 The CET tracking algorithm.

Input:

Current frame I_t; Previous object state S_{t-1}; Trained sub-models ELM_G and ELM_L.

Output:

The object state S_t in I_t; The new updated sub-models: ELM_G and ELM_L.

1: **If** $t = 1$: **Initialization**.

(1) Set the tracking object manually.
(2) Select training samples set X_G^P and X_G^N according the distance rule, resize them into the fixed size $N_{normG} \times N_{normG}$, and extract corresponding features to train ELM_G, as Fig. 1 shows.
(3) Select training samples set X_L^P and X_L^N according the patch strategy, resize them into the fixed size $N_{normL} \times N_{normL}$, and extract corresponding features to train each ELM_{Ll}. The final ELM_L is the ensemble of $ELM_{Ll}(l = 1, ..., \tilde{N}_L)$, as Fig. 2 shows.

2: **If** $t > 1$:

2.1 Tracking.

(1) Choose the candidate samples $\{\mathbf{x}_k\}$ with particle filter in I_t.
(2) Normalize $\{\mathbf{x}_k\}$ into size $N_{normG} \times N_{normG}$, and extract their features $\{\phi(\mathbf{x}_k)\}$.
(3) Calculate the confidence scores $conf_G(k)$ of each $\mathbf{x}_k$ by ELM_G
(4) Partition each $\mathbf{x}_k$ into patches $\mathbf{p}_{lk}$, extract feature and then send it to the corresponding ELM_{Ll} to obtain $conf_{Ll}(k)$. Calculate the final local confidence $conf_L$ according to (8).
(5) Determine the weights of ELM_G and ELM_L with KL distance. Calculate the final confidence score $conf(k)$ with (13) and then calculate the likelihood of each $\mathbf{x}_k$ with $\mathcal{L}(\mathbf{x}_k) = \exp(conf)$.
(6) Determine the optimal sample $\mathbf{x}_{opt}$ and the corresponding state S_t with (14). Output S_t.

2.2 Update.

(1) Determine whether to update by the tracking result of ELM_L according to (15) and (16). If $\hat{\mathbf{y}}_{Llk} = 1$, update the corresponding local samples; otherwise not update. If $\hat{\mathbf{y}}_s > 0$, update the global samples; otherwise not update.
(2) With the updated global samples and local samples, retrain ELM_G and ELM_L respectively.

3.4 Discussion

3.4.1 Co-training

Co-training framework has been widely used in tracking by different ways [23, 34, 44]. Actually, our collaborative representation follows the co-training framework from another perspective. Both the global sub-model and the local sub-model can be considered as a view

of the appearance model. Since they are both discriminative, the collaborative representation model is a co-trained discriminative model. Therefore, our tracking method possesses the advantages of the co-training framework, which can lead to more robust performance. Another benefit of the co-training is that the collaboration can be taken as the low-pass filter, which can significantly reduce the effect of the noise.

3.4.2 Multi-task learning

Multi-task learning has been successfully applied in detection [7, 8, 31, 48] and used by Zhang et al. to mine the relations of the particles for tracking [47]. In our study, building the local sub-model can be considered as a simplified multi-task learning formulation. Since the target and its surroundings are divided into several patches, to build a classifier for each corresponding patch can be taken as a single task, and the final tracking task is the combination of these tasks. In other words, the tracking task of the object tracking can be decomposed into multiple tasks of tracking of fragments of the object. In this formulation, the classifiers use different positive samples from different fragments of the target and share the negative samples sampled from the surroundings. The multi-task learning formulation makes the combination of local classifiers have the structural property, which can improve the robustness. In addition, the global sub-model can be considered as the regularization to the multi-task local sub-model, which constraints the local tasks to have the same objective.

4 Experiments

4.1 Experimental setup

The initialization of our CET algorithm is as the following. HOG which is with 5-pixel-window size and 9 orientation is extracted as the feature for the samples of both the global and local sub-models, because the HOG with these configurations have been widely applied in object detection and tracking in many methods [14, 52]. Sigmoid function is chosen as the activation function in all ELMs, which presses well in nonlinear feature mapping. ELM_G has 1500 hidden nodes while $ELM_{Ll}(l = 1, ..., M)$ has 500 nodes. The chosen numbers of the nodes are reasonable, for less nodes will degrade the tracking performance in our experiment while more nodes will spend more time which is unnecessary. The buffer depth of the positive pools is 50 while it is 100 for the negative. These numbers are determined experimentally, which can well differentiate the target and the background with good balance. The weight vectors and biases of all ELMs are assigned to random values according to the definition of ELM. The normalization size $Norm_G$ is 32×32 and $Norm_l$ is 8×8, which can adapt to the size of HOG. The number of the particles N_p is set to 400. More particles will increase the computation complexity while fewer particles may not capture the state changes. All these parameters keep the same on all the testing sequences.

We implement our CET tracker and evaluate its performance on OTB-2013 dataset which has 51 sequences [38]. These sequences have various complex conditions, such as severe occlusions, illumination variations, fast motion and blur, pose changes, or slight scale changes, etc. We compare the performance between our tracker and several state-of-the-art trackers, including KCF [14], HMT [40], VTD [19], TLD [18], SCM [54], Struck [13], VTS [20], CXT [11] and ASLA [17]. The parameters of the competing trackers are tuned carefully to achieve their best performance as far as possible.

4.2 Comparison results

We give both the quantitative and qualitative comparison results of these trackers on all sequences. We evaluate the performance quantitatively with four criteria [26]. The first is the average center location error (CLE), which is defined as the average of the errors of the center location and the ground truth in each frame. The second is the average VOC overlap rate (VOR), which is from pascal voc and defined as the average value of the scores, where $score = \frac{area(R_S \cap R_G)}{area(R_S \cup R_G)}$. R_S and R_G are the rectangle boxes of the tracking and ground truth respectively. The third is precision, which is obtained by calculating the ratio of the number of frames in which CLE is smaller than a threshold Th_p and the number of the total frames. The fourth is success rate (SR), which is defined as the ratio of the number of success frames and the total frames. If the VOR in a frame is larger than a predefined threshold Th_s, the tracking is considered successful in that frame. Moreover, the precision plots and success plots can be obtained to demonstrate the overall performance of the tracker, and the area under the curve (AUC) is also used as the evaluation criterion.

First, we compare the overall performance of the competing trackers on all the 51 sequences. Table 1 shows the comparison results on average CLE, average VOR, precision ($Th_p = 20$ pixels) and SR ($Th_s = 0.5$), from which we can find that the proposed CET method obtains the best results among the competing trackers. The average CLE of CET is 31.2 pixels which is smaller than all the competing trackers, while the average VOR of CET is 0.5656 which is the largest among the trackers. The precision with $Th_p = 20$ pixels of CET is 0.775, which outperforms KCF and Struck by 3.3% and 12% respectively. In addition, CET also outperforms KCF and HMT by about 5% in SR which is the best among the competing trackers. The precision plots and the success plots are displayed in Fig. 6. It can also be observed that the overall performance of CET is better than most existing state-of-the-art trackers in the OTB-2013 dataset.

Next, we evaluate the performance of the trackers in different conditions, including occlusion, deformation, scale variation, background clutter, according to the attributes of the sequences. The average precision plots and success plots are shown in Fig. 7, from which we explain the details of the performance of the proposed CET tracker.

Occlusions Figure 8a shows the precision plots and success plots of the competing trackers in the condition of occlusions. It can be observed that, the proposed CET tracker performs second best precision and the best AUC. As mentioned above, CET uses the collaborative

Table 1 The comparison results of average CLE (in pixel), average VOR, Precision ($Th_p = 20$) and SR ($Th_s = 0.5$) results of CET and several famous trackers in the benchmark

Method	Average CLE	Average VOR	Precision (20)	SR (0.5)
CET	31.2	0.5656	0.775	0.674
KCF	35.3	0.5216	0.742	0.625
HMT	39.9	0.5278	0.736	0.624
Struck	50.5	0.4771	0.656	0.559
SCM	54.1	0.5052	0.649	0.616
TLD	48.1	0.4404	0.608	0.521
ASLA	73.0	0.4384	0.532	0.511
CXT	68.4	0.4292	0.575	0.492
VTD	47.4	0.4184	0.576	0.493
VTS	50.7	0.4189	0.575	0.496

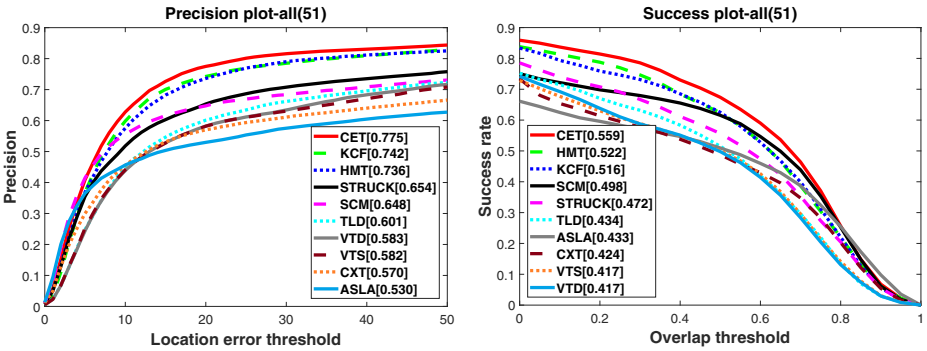


Fig. 6 Precision plots and success plots obtained by CET and several famous trackers in the benchmark. The values in the square brackets represent the precision with $Th_p = 20$ pixels on precision plots and the area under the curve (AUC) on success plots, respectively

model with extreme learning machine to improve the representation ability. Specifically, it makes use of the local model to control the update, which can reduce the impact of the occlusions.

Deformations Figure 8b displays the precision plots and success plots on the sequences with deformations. It can be seen that the plots obtained by CET are much better than the competing methods. The collaborative appearance model based on extreme learning and the on-line update model can adapt to the appearance changes caused by deformation of the object, which makes CET outperform in the condition of deformation.

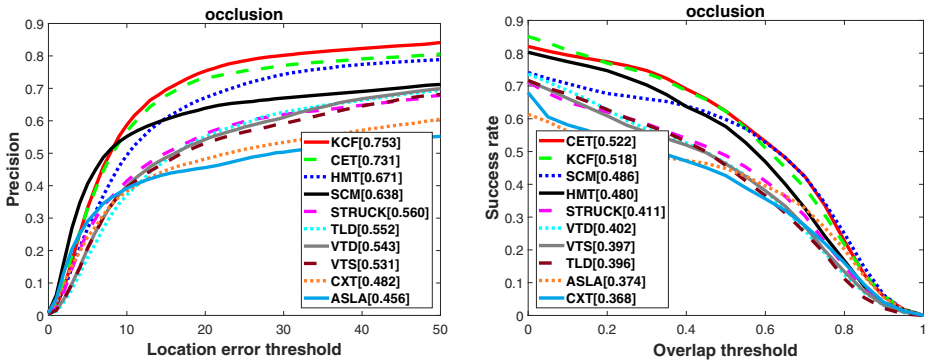
Out-of-plane variations Figure 8c. displays the comparison results of precision plots and success plots in the condition of out-of-plane variations. We can find that CET outperforms most of the others on both plots. Benefiting from the collaborative model and the update mode, CET can get good result in this condition as well.

Scale variations Figure 8d demonstrates the precision plots and success plots on sequences with scale variations. It can be found that CET gets the highest precision and the second best AUC. Since our CET method takes the fixed size bounding box for representation, it does not work as well as SCM, but it still outperforms the rest competing trackers.

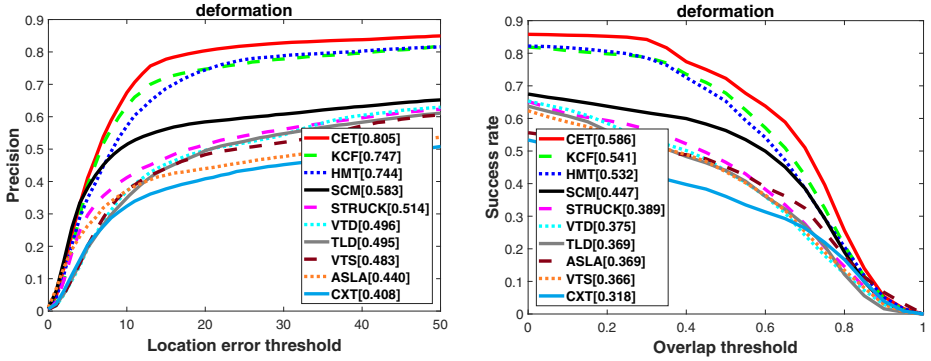
Fast motion Figure 8g demonstrates that CET can obtain significant advantage in the condition of fast motion. Since the particle filter used in our method is good at dealing with nonlinear motion and the collaborative model is discriminative, the CET method achieves desirable performance when the objects move fast.

Background clutter in Fig. 8h displays the comparison results in the condition of background clutter. We can see that CET is superior to the competing trackers. Since the collaborative model can use both the global and local information and both of the submodels are realized based on extreme learning machine which has powerful discriminability, our CET method can build more accurate appearance model and works well on the sequences with background clutter.

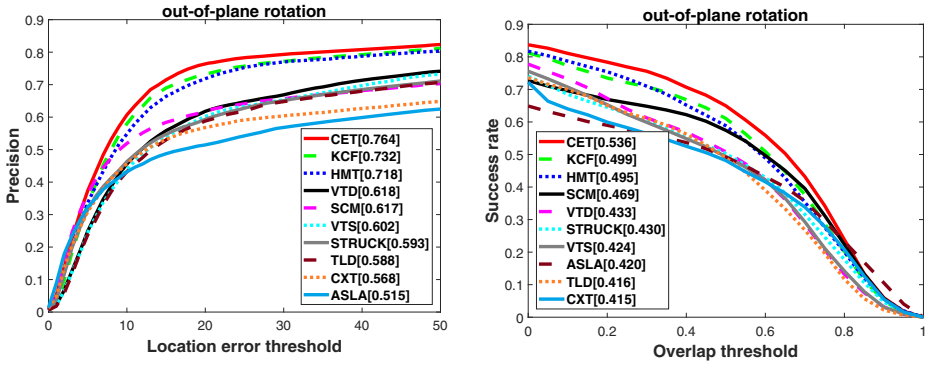
We also evaluate the performance of the CET tracker and competing trackers (e.g. KCF, HMT and Struck) qualitatively and display some tracking result examples on the key frames



(a) Precision and success plots of occlusion attribute.



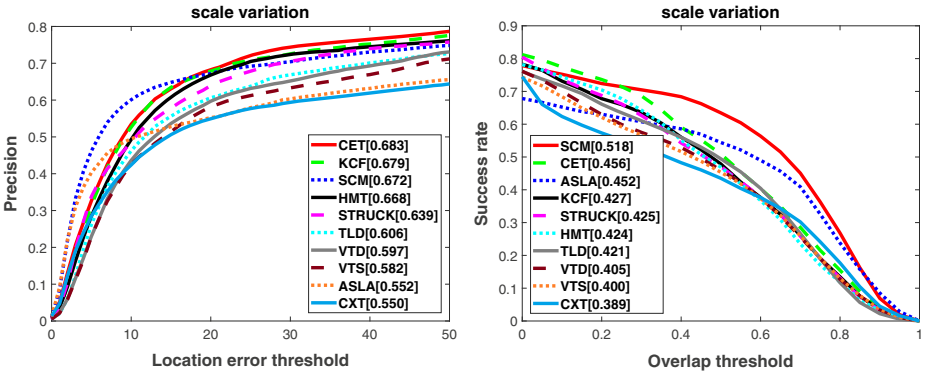
(b) Precision and success plots of deformation attribute.



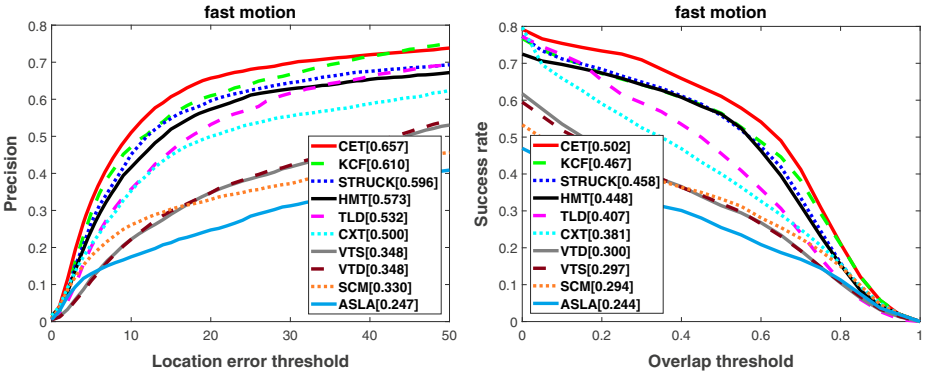
(c) Precision and success plots of out-of-plane attribute.

Fig. 7 Precision plots and success plots of CET and the competing trackers on the sequences with different attributes

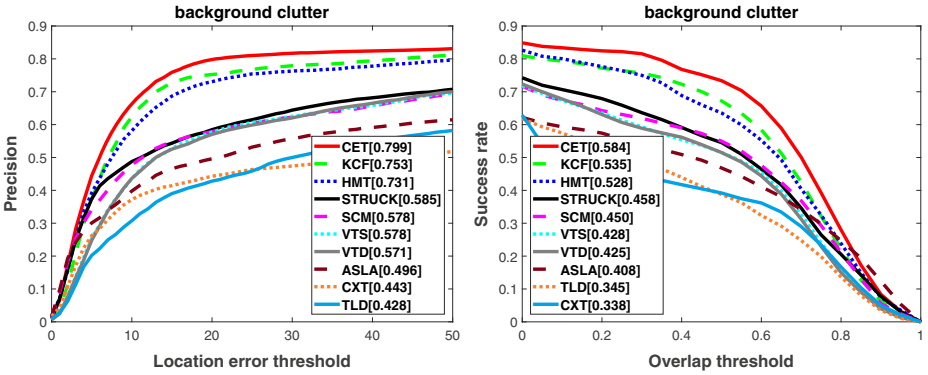
in Figs. 9 and 10. Sequence *basketball* has severe deformation, illumination changes and disturbance of the similar object. It can be found that only CET and KCF complete tracking successfully. There are different occlusions on *SUV*, and fast motion and out-of-rotation on



(a) Precision and success plots of scale variation attribute.



(b) Precision and success plots of fast motion.



(c) Precision and success plots of background clutter.

Fig. 8 Precision plots and success plots of CET and the competing trackers on the sequences with different attributes

tiger1. Since CET uses the local model to judge the occlusion part and use the collaborative model for representation, it successfully tracks the target while both the HME and Struck drift. *Shaking* has fast deformation and serious illumination changes. The HOG feature used in CET is robust to illumination changes while the online update can make CET adapt to the pose changes. Moreover, we can also observe that CET successfully locate the target in *football1* together with HMT and Struck, while KCF drift on this sequence. Because the huge occlusion on *jogging2*, only CET can complete the tracking while the occlusion makes all of the competing trackers drift.

4.3 Role analysis of the submodels

We investigate the role of the global sub-model ELM_G and the local sub-model ELM_L for the complete tracking framework. In our method, we utilize the KL distance to adaptively determine the weights of each sub-model for the collaboration. To evaluate its contribution of the global and the local submodels, we implement the trackers based on only ELM_G and only ELM_L, and represent them as CET_g and CET_l, respectively. In practice, these comparison trackers can be obtained by manually assigning different values to α_G and α_L.

Figure 11 indicates the overall performance comparison between the collaborative model with ELM and the submodels. It can be observed that the standard CET based on collaborative model with KL distance significantly outperforms the trackers CET_g and CET_l. The precision at $Th_p = 20$ pixels of CET is 0.775, which outperforms CET_g and

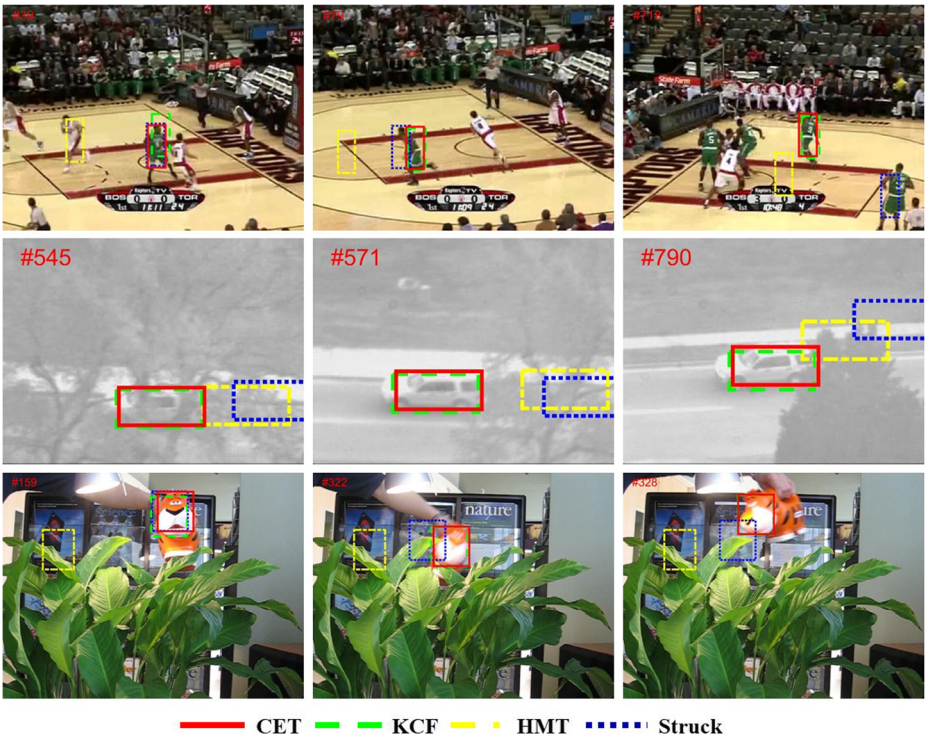


Fig. 9 Examples of the comparison tracking results on some representative frames. Top to down: *basketball*, *SUV*, *tiger1*

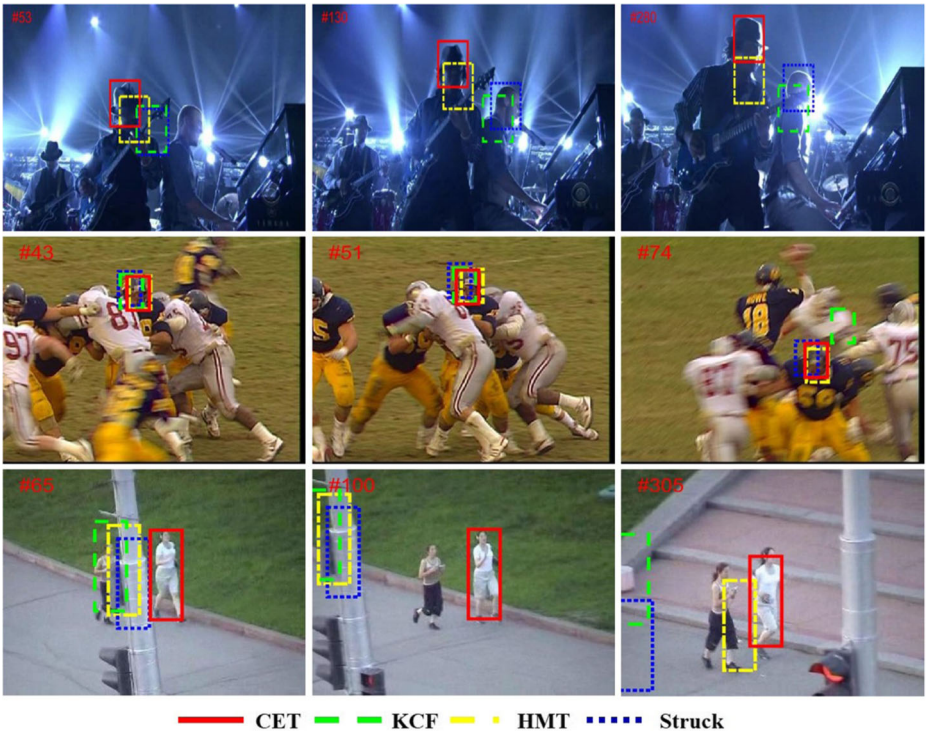


Fig. 10 Examples of the comparison tracking results on some representative frames. Top to down: *shaking*, *football1*, *jogging2*

CET_l by about 9.4% and 17% respectively. CEL also gets the similar result on AUC criterion, as the success plots in Fig. 11b illustrates. Further, we also demonstrate the comparison results in three representative sequences, which are shown in Fig. 12. We can observe that the CLE tracker with both ELM_G and ELM_L expresses much better than the submodels in

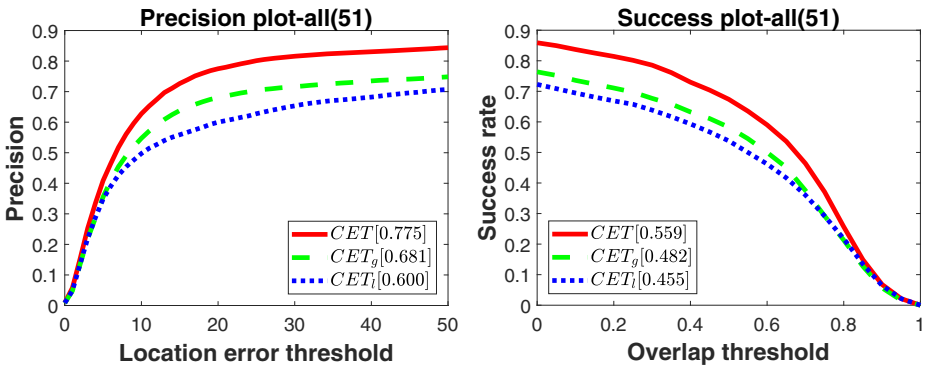
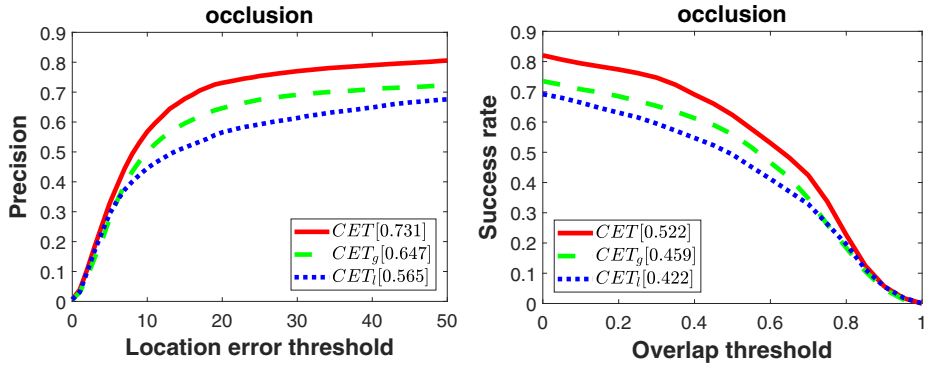
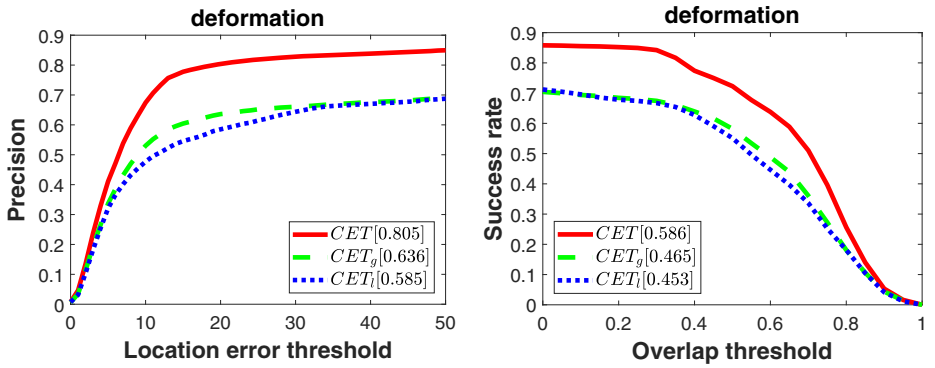


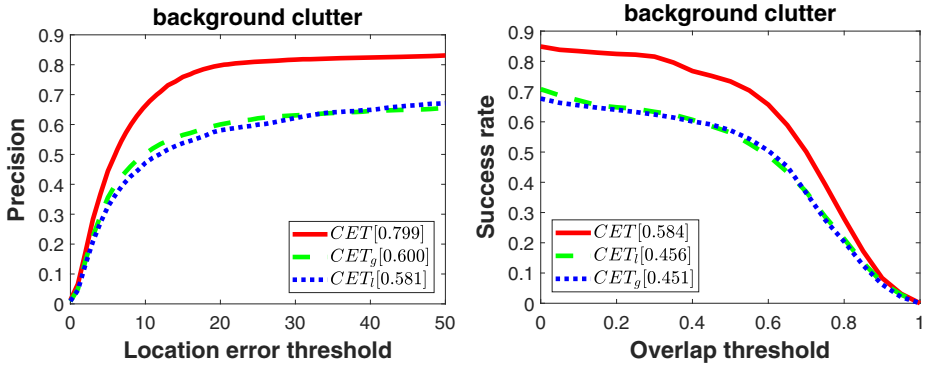
Fig. 11 Precision plots and success plots obtained by CET and submodels in the benchmark. The values in the square brackets represent the precision with $Th_p = 20$ pixels on precision plots and the area under the curve (AUC) on success plots, respectively



(a) Precision and success plots of occlusion attribute.



(b) Precision and success plots of deformation attribute.



(c) Precision and success plots of background clutter attribute.

Fig. 12 Precision plots and success plots of CET and the submodels on the sequences with different attributes

these conditions as well. Besides, CET_g have better results than CET_l, indicating that the global information plays an important role to retain the accuracy of the appearance model. The quantitative analysis is conducted on some sequences including *basketball*, *singer*2 and *woman*, and the results are shown in Fig. 13. On *basketball*, only the collaborative

model can complete the tracking successfully while either of the trackers with only ELM_G or ELM_L loses the target. On $singer2$ with cluttered background, CET_g drifts but both the CET and CET_l get the good tracking result because ELM_L can improve the discriminability in the local region. $Tiger$ has out-of-plane rotation and fast motion during the tracking process, which makes CET_l fail. Because the ELM_g can effectively preserve the global information, both CET and CET_g can complete the tracking on that sequence.

The ELM_G mainly takes use of the combination of the local features. However, the ELM_L is the ensemble of several local classifiers with structural configuration. Therefore, they capture the structural property of the object's appearance from the feature and the classifier levels respectively. The collaborative representation based on the adaptive combination of the global and the local sub-models can improve the tracking robustness significantly.

5 Conclusion

In this paper, we develop a novel collaborative representation model for object tracking. This model is constructed based on a global discriminative sub-model and a local discriminative

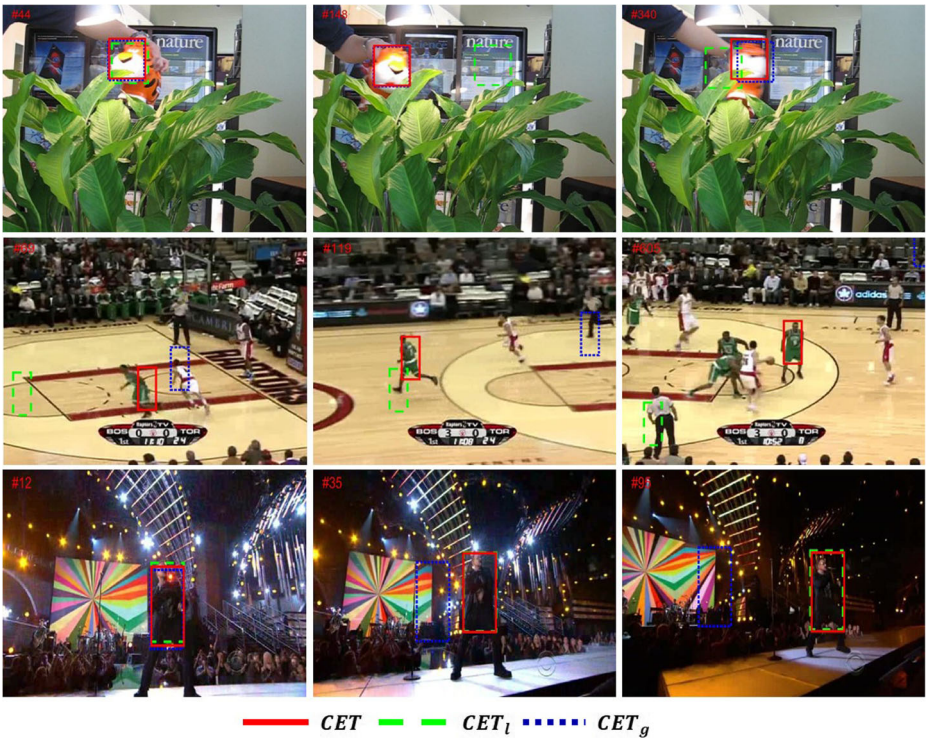


Fig. 13 Examples of the comparison tracking results on some representative frames. Top to down: *shaking*, *football1*, *jogging2*

sub-model, where the global sub-model captures the structure of the local features while the local sub-model builds a structured ensemble of the local classifiers. Both the global sub-model and the local sub-model are discriminative, the combination of which significantly improves the robustness. Moreover, we propose a novel KL distance based strategy to measure the importance of the sub-models, and determine their weights dynamically, which makes the combination more accurate and robust. In addition, the ELM algorithm is utilized to implement both of the sub-models and the novel CET tracking approach is realized. We compare the CET with many other famous trackers on several public sequences and the experimental results show that the CET can achieve the state-of-the-art performance. Furthermore, our collaborative representation model is a framework, which various feature extraction methods and classifiers can be embedded in.

References

1. Adam A, Rivlin E, Shimshoni I (2006) In: Proceedings of the IEEE conference computer vision and pattern recognition (CVPR), vol 1, pp 798–805
2. Avidan S (2004) IEEE Trans Pattern Anal Mach Intell 26(8):1064
3. Avidan S (2007) IEEE Trans Pattern Anal Mach Intell 29(2):261. https://doi.org/10.1109/TPAMI.2007.35
4. Babenko B, Yang MH, Belongie S (2011) IEEE Trans Pattern Anal Mach Intell 33(8):1619. https://doi.org/10.1109/TPAMI.2010.226
5. Bai Q, Wu Z, Sclaroff S, Betke M, Monnier C To appear in ICCV2013
6. Baojie Fan YT, Cong Y (2017) J Electron Imaging 26:26. https://doi.org/10.1117/1.JEI.26.1.013007
7. Chang X, Ma Z, Lin M, Yang Y, Hauptmann AG (2017) IEEE Trans Image Process 26(8):3911
8. Chang X, Yu YL, Yang Y, Xing EP (2017) IEEE Trans Pattern Anal Mach Intell 39(8):1617
9. Chen D, Yuan Z, Wu Y, Zhang G, Zheng N To appear in ICCV2013
10. Danelljan M, Bhat G, Khan FS, Felsberg M (2017) In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), pp 6931–6939
11. Dinh TB, Vo N, Medioni G (2011) In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1177–1184, https://doi.org/10.1109/CVPR.2011.5995733
12. Grabner H, Grabner M, Bischof H (2006) In: Proceedings of the British machine vision conference, vol 1, pp 47–56
13. Hare S, Saffari A, Torr PHS (2011) In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 263–270
14. Henriques JF, Caseiro R, Martins P, Batista J (2015) IEEE Trans Pattern Anal Mach Intell 37(3):583
15. Huang GB, Zhu QY, Siew CK (2006) Neurocomputing 70(1):489
16. Huang GB, Zhou H, Ding X, Zhang R (2012) IEEE Trans Syst Man Cybern B Cybern 42(2):513
17. Jia X, Lu H, Yang MH (2012) In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1822–1829
18. Kalal Z, Mikolajczyk K, Matas J (2012) IEEE Trans Pattern Anal Mach Intell 34(7):1409
19. Kwon J, Lee KM (2010) In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1269–1276, https://doi.org/10.1109/CVPR.2010.5539821
20. Kwon J, Lee KM (2011) In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 1195–1202
21. Li H, Shen C, Shi Q (2011) In: Proceedings of the IEEE conference computer vision and pattern recognition (CVPR), pp 1305–1312. https://doi.org/10.1109/CVPR.2011.5995483
22. Li X, Dick A, Shen C, van den Hengel A, Wang H IEEE Transactions on Pattern Analysis & Machine Intelligence (to be published). Early Access
23. Liu R, Cheng J, Lu H (2009) In: 2009 IEEE 12th international conference on computer vision, IEEE, pp 1459–1466

24. Liu B, Huang J, Kulikowski C, Yang L (2013) IEEE Trans Pattern Anal Mach Intell 35(12):2968
25. Lu H, Zhou Q, Wang D, Xiang R (2011) In: 2011 IEEE international conference on automatic face & gesture recognition and workshops (FG 2011), IEEE, pp 539–544
26. Luka C, Matej K, Ales L Technical report 10-2013
27. Mei X, Ling H (2009) In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 1436 –1443. https://doi.org/10.1109/ICCV.2009.5459292
28. Mei X, Ling H (2011) IEEE Trans Pattern Anal Mach Intell 33(11):2259. https://doi.org/10.1109/TPAMI.2011.66
29. Mueller M, Smith N, Ghanem B (2017) In: Proceedings of the IEEE conference on computer vision pattern recognition (CVPR), pp 1396–1404
30. Nam H, Han B (2016) In: The IEEE conference on computer vision and pattern recognition (CVPR), pp 4293–4302
31. Peng X, Schmid C (2016) In: European conference on computer vision (Springer), pp 744–759
32. Ross D, Lim J, Lin R, Yang M (2008) Int J Comput Vis 77(1):125
33. Saffari A, Leistner C, Santner J, Godec M, Bischof H (2009) In: 2009 IEEE 12th international conference on computer vision workshops (ICCV workshops), IEEE, pp 1393–1400
34. Santner J, Leistner C, Saffari A, Pock T, Bischof H (2010) In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 723–730. https://doi.org/10.1109/CVPR.2010.5540145
35. Sun L, Liu G (2011) IEEE Trans Circuits Syst Video Technol 21(4):408. https://doi.org/10.1109/TCSVT.2010.2087815
36. Tang F, Brennan S, Zhao Q, Tao H (2007) In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 1–8. https://doi.org/10.1109/ICCV.2007.4408954
37. Wang Y, Luo X, Ding L, Hu S (2018) Multimed Tools Appl 77(23):31447. https://doi.org/10.1007/s11042-018-6198-8
38. Wu Y, Lim J, Yang M (2013) In: IEEE conference on computer vision and pattern recognition (CVPR), IEEE, pp 2411–2418
39. Yang H, Shao L, Zheng F, Wang L, Song Z (2011) Neurocomputing 74(18):3823
40. Yang J, Zhang S, Zhang L (2016) J Electron Imaging 25(5):053006
41. Yang H, Zhong D, Liu C, Song K, Yin Z (2018) J Electron Imaging 27:27. https://doi.org/10.1117/1.JEI.27.2.023008
42. Yao R, Shi Q, Shen C, Zhang Y, van den Hengel A (2013) In: IEEE conference on computer vision and pattern recognition (CVPR)
43. Yilmaz A, Javed O, Shah M (2006) Acm Computing Surveys (CSUR) 38(4):13
44. Yu Q, Dinh T, Medioni G (2008) Comput Vis–ECCV 2008:678–691
45. Yun S, Choi J, Yoo Y, Yun K, Choi JY (2017) In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), pp 1349–1358
46. Zhang L, van der Maaten L (2013) In: 2013 IEEE conference on computer vision and pattern recognition (CVPR), IEEE, pp 1838–1845
47. Zhang T, Ghanem B, Liu S, Ahuja N (2012) In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 2042–2049
48. Zhang G, Liu J, Li H, Chen YQ, Davis LS (2017) IEEE Signal Process Lett 24(11):1666
49. Zhang S, Lu W, Xing W, Zhang L (2018) IEEE transactions on cybernetics. Early acccess 1:14. https://doi.org/10.1109/TCYB.2018.2868782
50. Zhang S, Lu W, Xing W, Zhang L (2018) Pattern Recogn 84:112
51. Zhang S, Sui Y, Yu X, Zhao S, Zhang L (2015) Pattern Recogn 48(8):2474
52. Zhang S, Zhao S, Sui Y, Zhang L (2015) IEEE Trans Image Process 24(12):5723
53. Zhao J, Zhang W, Cao F (2018) Multimed Tools Appl 77(23):30969. https://doi.org/10.1007/s11042-018-6132-0
54. Zhong W, Lu H, Yang MH (2012) In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1838–1845

Haipeng Kuang is a researcher in Changchun Institute of Optics Fine Mechanics and Physics. He received his B.S. degree from the Jilin University of Technology in 1994, and his M.S. degree from the University of the Chinese Academy of Sciences in 2000. He received his Ph.D. degree from the University of the Chinese Academy of Sciences in 2008. His current research interests include video analysis, optical imaging and mapping techniques for aerial remote.

Liang Xun received the B.S. adn M.S. degrees from Tsinghua University in 2003 and 2006 respectively. He is currently an engineer in Beijing Topmoo Technologies Co., Ltd. His research interests include image processing, pattern recognition and computer vision.