Robust Object Tracking With Discrete Graph-Based Multiple Experts

Jiatong Li, Chenwei Deng, *Senior Member, IEEE*, Richard Yi Da Xu, Dacheng Tao, *Fellow, IEEE*, and Baojun Zhao

***Abstract*— Variations of target appearances due to illumination changes, heavy occlusions, and target deformations are the major factors for tracking drift. In this paper, we show that the tracking drift can be effectively corrected by exploiting the relationship between the current tracker and its historical tracker snapshots. Here, a multi-expert framework is established by the current tracker and its historical trained tracker snapshots. The proposed scheme is formulated into a unified discrete graph optimization framework, whose nodes are modeled by the hypotheses of the multiple experts. Furthermore, an exact solution of the discrete graph exists giving the object state estimation at each time step. With the unary and binary compatibility graph scores defined properly, the proposed framework corrects the tracker drift via selecting the best expert hypothesis, which implicitly analyzes the recent performance of the multi-expert by only evaluating graph scores at the current frame. Three base trackers are integrated into the proposed framework to validate its effectiveness. We first integrate the online SVM on a budget algorithm into the framework with significant improvement. Then, the regression correlation filters with hand-crafted features and deep convolutional neural network features are introduced, respectively, to further boost the tracking performance. The proposed three trackers are extensively evaluated on three data sets: TB-50, TB-100, and VOT2015. The experimental results demonstrate the excellent performance of the proposed approaches against the state-of-the-art methods.**

***Index Terms*— Object tracking, discrete graph, dynamic programming, support vector machine, correlation filter, convolutional neural network.**

Manuscript received June 10, 2016; revised November 17, 2016 and January 9, 2017; accepted February 23, 2017. Date of publication March 23, 2017; date of current version April 11, 2017. This work was supported in part by the International Graduate Exchange Program of Beijing Institute of Technology (BIT), in part by the Dual Doctoral Degree Program between University of Technology, Sydney and BIT, in part by the National Natural Science Foundation of China under Grant 61301090, and in part by the Australian Research Council under Project FT-130101457, Project DP-140102164, and Project LP-150100671. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jianfei Cai. *(Corresponding author: Chenwei Deng)*

J. Li is with the School of Information and Electronics, Beijing Institute of Technology, Beijing 100081, China, and also with the Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia (e-mail: ljt_bit@bit.edu.cn; jiatong.li-3@student.uts.edu.au).

C. Deng and B. Zhao are with the School of Information and Electronics, Beijing Institute of Technology, Beijing 100081, China (e-mail: cwdeng@bit.edu.cn; zbj@bit.edu.cn).

R. Y. D. Xu is with the Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia (e-mail: yi-da.xu@uts.edu.au).

D. Tao is with the School of Information Technologies, The University of Sydney, Darlington, NSW 2008, Australia (e-mail: dacheng.tao@sydney.edu.au).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2017.2686601

I. Introduction

VISUAL object tracking is one of the fundamental problems among numerous research topics in computer vision. A common tracking scenario is to track the unknown object given only the initial bounding box of the target. This problem is a challenging task due to target deformations, illumination variations, abrupt motions, occlusions and background clutters.

To handle tracking failures caused by the above mentioned factors, a commonly used strategy is to design an online model that evolves forward to adapt to the variations. The main drawback of this method is that the online model tends to drift with the time passing by. The drift happens even more easily due to large appearance changes of the object, abrupt motions and heavy occlusions. To tackle the model drift problem, many methods propose to use tracker ensembles (or multiple experts) which are composed of more than one base trackers to determine the target position [1]–[5]. One strategy is to establish a tracker pool and choose the most appropriate tracker each frame to make the best decision [1]–[3]. Others use multiple experts working parallel to better discriminate the target from the background [4], [5].

One of the representative work is that in [6], Zhang *et al.* propose to use the multi-expert restoration scheme to address the model drift problem, where an entropy based loss function is defined to determine whether the current tracker is reliable and should be restored to the historical tracker. The proposed tracking framework adopts online SVM as the base tracker, which shows very robust performance. The work in [6] indicates that the historical trained tracker snapshots can be used to correct the model drift.

Furthermore, we show that the effect of historical tracker snapshots for drift correction is more obvious for regression correlation filter, whose response map is with less ambiguity than that of online SVM used in [6]. As shown in Fig. 1, during the tracking process, the objects go through significant illumination changes, appearance variations and occlusions. Therefore, the current tracker (whose response map framed by the green bounding box) tends to drift. However, it is observed that the target location can be accurately estimated by most of the historical tracker snapshots. For example, in sequence *Box*, after the object having been occluded, the current tracker is distracted by the background, but its three past snapshots are all able to identify the true target. The same phenomenon is shown in sequence *Tiger1* and *Coke*. Besides, as illustrated in sequence *Shaking*, we will show that by designing the

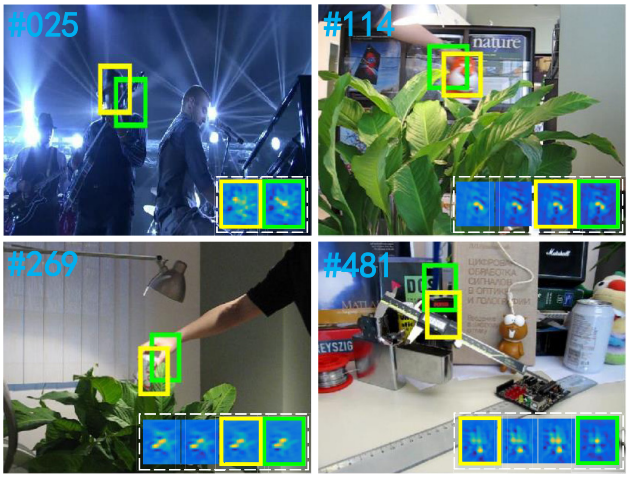

Fig. 1. Four typical sequences (*Shaking*, *Tiger1*, *Coke*, *Box*) show the importance of exploiting the historical tracker snapshots. The embedded figures at the bottom-right of the sequence image are the response maps of correlation filter for the current tracker and its historical tracker snapshots (listed in chronological order from left to right). The historical tracker snapshots are stored at the pre-defined frame intervals. The green bounding box is the tracking result for the current tracker, while the yellow box is the selected tracker snapshot by the multi-expert framework. The number at the top-left corner is the frame count.

appropriate snapshot selection criteria, early period tracking drifts can also be avoided.

The above discussions are the main motivations of our work. During the tracking process, single tracker is easy to overfit when there is challenging tracking scenarios. However, the above moments are relatively short during the whole tracking process. On the other hand, the diversity of target appearance is usually limited, and cannot be varying significantly all the time without restoring to is past appearance. Therefore, sometimes the past tracker snapshots are capable of recognizing the object better than the current tracker, which is natural to rescue tracking failures. As a result, in this paper, we exploit the historical tracker snapshots and show that tracking performance can be effectively promoted by exploiting the relationship between the current tracker and its historical snapshots.

In this work, we aim to establish a multi-expert framework constituted by the current tracker and its past trained tracker snapshots. Therefore, the major issues are:

- How to model the relationship among the multiple experts?
- How to define the criteria to select the best hypothesis given by the multiple experts?
- Since object tracking demands high efficiency, how to get the solution of the framework without sacrificing the computation load?

To solve the above issues, in this paper, we propose a graph optimization framework to model the relationship of the multiple experts. The graph nodes are modeled by the hypotheses of the multiple experts. As each single expert tends to be confident to its own decision, we define the score of the graph in a semi-supervised learning manner. The proposed method is capable of automatically detecting and correcting the tracker drift by finding the optimal path with the highest score in the graph. With the efficient solver of dynamic programing, our method can implicitly analyze the trajectories and reliabilities of the multi-expert by only computing their scores in the current frame, so as to effectively correct the tracker drift with high efficiency.

The proposed framework is applied to three base trackers as special cases. We first introduce the online SVM on a budget [6], [7] as the base tracker, and show our framework can significantly improve its performance. Moreover, we adopt the regression correlation filter with hand-crafted features and CNN features respectively as the base tracker to further boost the tracking performance, which learns the temporal context correlation of the target, and an efficient scale adaptive scheme is introduced to handle the object scale changes on-the-fly. On the widely used TB-50 benchmark [8], the proposed methods show significant improvement against other state-of-the-art methods. The proposed trackers are further tested on the new TB-100 [9] and VOT2015 [10] dataset, which also shows its excellent performance.

In summary, the contributions of this paper are as follows:

- A historical snapshots multi-expert (HSME) framework is introduced to handle the tracker drift problem. We propose to use the unified discrete graph algorithm to model the multiple experts, where an exact solution exists, which can be solved efficiently by dynamic programming without sacrificing the computation load. Furthermore, we extend the expert loss function based on [6] as the graph unary score, which can describe the consistency and ambiguity of the multi-expert in a unified formulation;
- We integrate three base trackers, online SVM and correlation filter (hand-craft feature and CNN feature) into our HSME framework, and propose the trackers named HSME-SVM, HSME-CF and HSME-deep respectively. The proposed three trackers are extensively evaluated on three big datasets, and show excellent performance against state-of-the-art methods with high efficiency.

The rest of the paper are organized as follows. After reviewing the related work in Section II, we introduce the proposed method in Section III. The implementations and method analysis for the proposed trackers are detailed in Section IV. The experimental results are presented in Section V. Finally, we conclude the paper.

II. Related Work

Visual tracking has been extensively studied [11]. Recent public available benchmarks and evaluation methods have also accelerated the development of this field [8], [9], [12], [13]. In the following, we summarize the related work from two angles of views, and discuss their major differences from our work.

A. Tracking-by-Detection

Tracking-by-detection methods play a key role among numerous recent trackers. Under this framework, a discriminative classifier is trained to classify the foreground and background features [14]–[17]. For instance, Avidan [14]

integrates the SVM classifier into the optical flow to establish the online target discriminative model. Babenko *et al.* [15] introduce multiple instance learning to collect positive and negative samples into bags to avoid model drift. In [16], random projection is used to reduce feature dimension which achieves real-time tracking. Particularly, Hare *et al.* [17] use structured output SVM and trains samples with structured labels, which shows excellent performance in the tracking benchmark [8]. Others, such as in [18], context information is also added to promote tracking performance.

Over the last few years, correlation filter based tracking methods have attracted great attention due to their high efficiency [19]–[23]. Since Bolme *et al.* [19] propose a minimum output sum of squared error filter for tracking, correlation filter began to re-attract attention as a commonly used method in signal processing. After that, Henriques *et al.* [20] propose to use circular image patches as dense samples to train the correlation filter in kernel space with low computation load. The above methods are based on gray-level feature. The work is further extended to multi-channel feature in the KCF tracking algorithm [21]. In [22], color attributes are added to the framework of [20], and an adaptive dimension reduction technique is proposed, which demonstrates the importance of color feature in visual tracking. Other extended work, such as in [23]–[25], the scale variation, adaptive update, even long short term memory scheme are added to the correlation filter tracker. More recently, a complementary feature integration scheme is introduced in correlation filter [26], which shows excellent performance.

Since convolutional neural networks (CNNs) have achieved great success in computer vision [27]–[29], there is a rising trend for introducing CNN models into visual tracking task. In [30], a stacked autoencoder is used to learn the tracker representation. On the other hand, Hong *et al.* [31] adopt a learned saliency map on pre-trained CNN to predict the target state. In [32], CNN feature is integrated into the correlation filter framework, and target location is estimated hierarchically to improve the tracking robustness. Zhang *et al.* [33] propose a two-layer convolutional network with a lightweight structure for visual tracking without off-line training. In [34], a biologically inspired tracker is proposed based on the analysis of visual cognitive mechanism. Furthermore, in [35], a sequential online training method is proposed to overcome overfitting when fine-tuning the pre-trained deep models for visual tracking.

B. Tracker Ensemble (Multi-Expert)

Some tracking algorithms adopt tracker ensemble to achieve more robust tracking performance. For example, Kwon [1] decomposes traditional Bayesian recursive framework into basic models, and uses the MCMC sampling to integrate them. In [2], the proposed method not only samples the target state space but also the tracker space to handle challenging tracking scenes. Kalal *et al.* [4] address the long term tracking problem by designing two complementary experts, one estimates missed detections and the other estimates false alarms, apart from this, a re-detection scheme is designed to achieve long term tracking. In [3], a sparsity-based collaboration of discriminative and generative modules are proposed. In MEEM [6] and SME [36], multiple experts including historical tracker snapshots are used to handle the model drift problem, which shows high tracking efficiency. They both use a disagreement threshold to decide whether the current tracker drifts, and then select the most reliable expert according to their accumulated scores. Another multi-expert selection strategy can be found in [37]. In [37], each expert first moves forward and then backward to get pairs of trajectories, the most reliable expert is selected by analyzing their pair-wise geometry similarity.

Our work is mostly close to MEEM [6] and SME tracker [36], but with significant differences summarized as follows. Firstly, in [6] and [36], multi-expert is established in a heuristic form. With the multi-expert framework, tracker drift is first detected by the handcrafted condition and then corrected by the expert selection scheme. In our work, the above two steps are merged and modeled as the unified discrete graph optimization framework, and the tracker drift is corrected by finding the exact optimal solution. Secondly, in [6] and [36], tracker drift is corrected aggressively by replacing the current tracker by the historical snapshot, which means totally discarding the recent tens and even hundreds of frames update. While our framework is capable of correcting the tracker drift promptly both before or after the large disagreement among the multiple experts, which is more flexible and stable. Thirdly, our method is a general framework. We test the proposed framework in three general base trackers to validate its effectiveness.

III. Proposed Method

In this section, we first introduce the discrete graph optimization model, and its solver of dynamic programming. Then we show how to model the multiple experts by the graph optimization framework, as well as how to define the graph scores. At last, we introduce the base trackers.

A. Relational Graph and Hypothesis Graph

In the computer vision problem, many issues can be formulated as follows [38], [39]. Assuming there is a set of entities $V = \{v^{(n)}|_{n=1}^{N}\}$, where each entity can be in one of the following states $S = \{s^{(m)}|_{m=1}^{M}\}$, with the unary score function $\phi(s^{(m)}|v^{(n)}; v^{(n)} \in V,\ s^{(m)} \in S)$, which represents the likelihood that entity $v^{(n)}$ is in state $s^{(m)}$. Moreover, there is a binary compatible function $\psi(s^{(k)}|v^{(i)}, s^{(l)}|v^{(j)}; v^{(i)}, v^{(j)} \in V,\ s^{(k)}, s^{(l)} \in S)$, which denotes the compatibility for that entity $v^{(i)}$ in state $s^{(k)}$ and entity $v^{(j)}$ in state $s^{(l)}$. A natural objective is to assign the states for all the entities so that all of them are with highest unary scores as well as are most compatible with each other. The above issue can be regarded as a discrete optimization problem presented by the relational graph and hypothesis graph.

1) Relational Graph: A relational graph $\mathcal{G}_r = (\mathcal{V}_r, \mathcal{E}_r)$ describes the relationship of a set of entity nodes $\{v_r^{(i)}|_{i=1}^{|\mathcal{V}_r|}\}$ and relationship of pair-wise nodes represented by the edge $e \in \mathcal{E}_r$. For example, Fig. 2(a) shows a typical single branch relational graph.

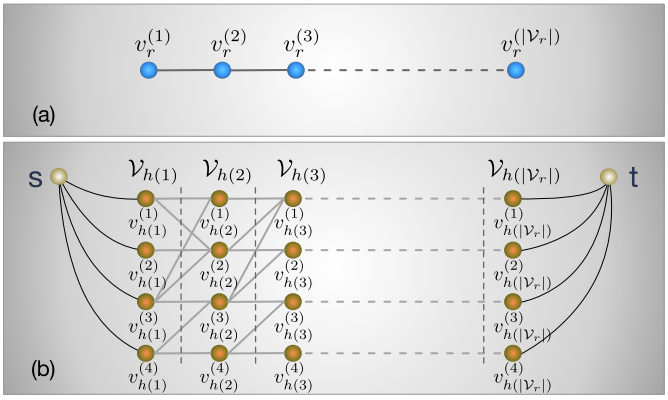

Fig. 2. Graph illustration. (a) Relational Graph for single branch tree. (b) Hypothesis graph generated according to (a) assuming each entity of relational graph has four possible states. The brown balls denote the hypothesis nodes and the lines between balls represent the edges of the hypothesis graph. Note this is only the illustration and not all the edges are drawn for conciseness.

2) Hypothesis Graph: With a relational graph $\mathcal{G}_r$, the corresponding hypothesis graph $\mathcal{G}_h = (\mathcal{V}_h, \mathcal{E}_h)$ can be established. For each entity node $v_r^{(i)} \in \mathcal{V}_r$, a set of hypothesis nodes $\mathcal{V}_{h(i)} = \{v_{h(i)}^{(k)}|_{k=1}^{|\mathcal{V}_{h(i)}|}\}$ are proposed, where $\mathcal{V}_h = \bigcup_{i=1}^{|\mathcal{V}_r|} \mathcal{V}_{h(i)}$. It means that the hypothesis nodes represent the possible states of each entity node. And the hypothesis edges can be denoted by $\mathcal{E}_h = \{(v_{h(i)}^{(k)}, v_{h(j)}^{(l)}); v_{h(i)}^{(k)} \in \mathcal{V}_{h(i)}, v_{h(j)}^{(l)} \in \mathcal{V}_{h(j)}\}$, which are built according to the structure of relational graph $\mathcal{G}_r = (\mathcal{V}_r, \mathcal{E}_r)$, where $(v_r^{(i)}, v_r^{(j)}) \in \mathcal{E}_r$. Similarly, each hypothesis node has its unary score ϕ, and each pair-wise hypothesis edge has its binary compatible score ψ. An illustration of hypothesis graph can be found in Fig. 2(b).

The goal is to select one hypothesis node for each entity node, so that to maximize the summation of their unary and binary compatible scores. Therefore, given a hypothesis graph $\mathcal{G}_h$ building upon a relational graph $\mathcal{G}_r$, the objective function for a set of hypothesis nodes $v_h = \{v_{h(i)}|_{i=1}^{|\mathcal{V}_r|}; v_{h(i)} \in \mathcal{V}_h\}$ is:

$$\mathcal{F}(v_h) = \sum_{v_{h(i)} \in \mathcal{V}_h} \phi(v_{h(i)}) + \beta \sum_{(v_{h(i)}, v_{h(j)}) \in \mathcal{E}_h} \psi(v_{h(i)}, v_{h(j)}), \tag{1}$$

where β is a parameter to trade off the unary and binary weights. The goal is to find an optimal set of nodes to maximize the above objective function, i.e. $v_h^* = \arg\max_{v_h}(\mathcal{F}(v_h))$.

This general problem of discrete optimization is NP-hard, however, if the entities of the relational graph are connected in a tree structure, the problem can be solved efficiently by dynamic programming in polynomial time [38]. Furthermore, in this paper, the proposed problem can be abstracted as a degenerate tree structure (single branch tree as shown in Fig. 2). Now we show how to use dynamic programming to get the solution $v_h^* = \{v_{h(i)}^*|_{i=1}^{|\mathcal{V}_r|}; v_{h(i)}^* \in \mathcal{V}_h\}$ that maximizing the objective function of Eq. 1. Given a relational graph of single branch tree, and let $\mathcal{S}_i(k)$ denotes the maximum accumulated unary and binary score of hypothesis node $v_{h(i)}^{(k)}$ for the first ith entity, Eq. 1 can be solved by dynamic programming through the recursive equations:

$$\mathcal{S}_i(k) = \phi(v_{h(i)}^{(k)}) + \max_l(\mathcal{S}_{i-1}(l) + \beta\psi(v_{h(i-1)}^{(l)}, v_{h(i)}^{(k)})). \tag{2}$$

Once the $\mathcal{S}_i(k)$ has been computed, the optimal solution can be obtained by setting $v_{h(|\mathcal{V}_r|)}^* = \arg\max_k \mathcal{S}_{|\mathcal{V}_r|}(k)$ and tracing back by decreasing i:

$$v_{h(i)}^* = \arg\max_{v_{h(i)}^{(k)}}(\mathcal{S}_i(k) + \psi(v_{h(i)}^{(k)}, v_{h(i+1)}^*)). \tag{3}$$

The above dynamic programming algorithm runs in $\mathcal{O}(|\mathcal{V}_r|K^2)$, where K is the maximum number of hypothesis nodes for each entity in $\mathcal{V}_r$.

B. Multi-Expert Framework Modeled by Discrete Graph

1) Graph for Multiple Experts: We aim to utilize the trained tracker snapshots to build the multiple experts framework. Given a base tracker which updates every frame, we store the update record of the base tracker at intervals of certain frames as the tracker snapshot, i.e. the expert (In the following, we do not differentiate tracker snapshot from expert). Let $\mathcal{T}_t$ denotes the tracker snapshot (expert) trained up to time t. Until time T, we have an expert ensemble $\mathbf{E} := \{\mathcal{T}_{t_1}, \mathcal{T}_{t_2}, \ldots, \mathcal{T}_T\}$, where $\mathcal{T}_T$ represents the tracker at the current time.

Using the discrete graph introduced in Section III-A, we model the evolution of the multi-expert as a single branch relational graph, with the entity node represents the expert ensemble. At each time step, there are $|\mathbf{E}|$ possible object state hypotheses, where each hypothesis is given by a single expert in the expert ensemble. Then the corresponding hypothesis graph is built whose node is the hypothesis given by each expert. As a result, the best hypothesis for each frame is selected according to the path that achieves the highest score from the beginning to the current frame, which can be solved by dynamic programming indicated in Eq. 2. An illustration can be found in Fig. 3, where the expert ensemble contains maximum 4 experts, and discards the oldest expert when its number exceeds 4. The notations for the graph unary score in Fig. 3 are the same in those of Section III-A. In the following, we show how to define the unary and binary graph scores.

2) Unary Graph Score: It is very important to define the unary score and binary compatible score of the graph. As each expert, especially the expert that is prone to drift, tends to be more confident to their own predictions, the expert score cannot be defined simply based on the likelihood given by the response value. Inspired by [6], the unary score is formulated by the label instead of the response values of the experts. In specific, the expert ensemble proposes several target candidates each time. For each instance in the target candidates, the label is uncertain (be positive or negative), which can be regarded as the semi-supervised partial label learning problem [40], [41]. In the partial label learning problem, training data set is denoted by $\mathcal{D} = \{\mathbf{x}_i, \mathbf{z}_i\}$, where $\mathbf{x}_i$ is the data sample, and $\mathbf{z}_i$ indicates the possible label set that contains the true label of $\mathbf{x}_i$.

We aim to define the unary score to describe the consistency and the ambiguity of the expert relative to the others simultaneously. Motivated by [6] and [41], the semi-supervised learning

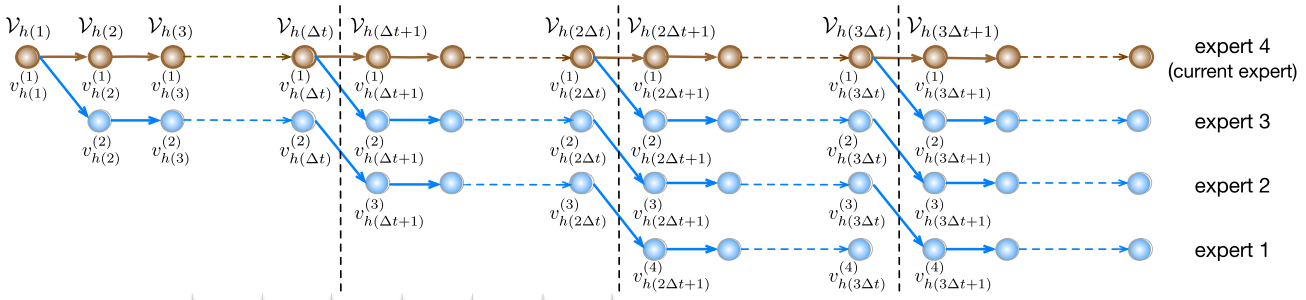

Fig. 3. Illustration for the multi-expert framework modeled by discrete graph. The multi-expert ensemble includes the current tracker represented by the brown nodes and the historical snapshots colored by blue nodes. The historical tracker snapshot is stored at intervals of Δt frames, which is denoted by the black dash line, and the oldest expert is discarded if the expert number exceeds the maximum (the figure illustrates maximum 4 experts). Arrow direction denotes the same expert, and the graph edge exists in pair-wise adjacent nodes.

problem is solved in a MAP framework that maximizes the log posterior probability of the model parameterized by θ:

$$\mathcal{P}(\theta, \eta; \mathcal{D}) = \mathcal{L}(\theta; \mathcal{D}) - \eta H_{emp}(Y|X, Z; \mathcal{D}, \theta), \tag{4}$$

where $\mathcal{L}(\theta; \mathcal{D})$ is the log likelihood of the model parameterized by θ, and $H_{emp}(Y|X, Z; \mathcal{D}, \theta)$, similar to that in [6], is the empirical conditional entropy prior conditioned on the training data X and their possible label set Z. The scalar η is the regularization coefficient to control the trade-off between the likelihood and the prior.

In [6], the score definition only adopts the entropy to favor the expert with low ambiguity. The entropy score is suitable for the base tracker whose response map is relatively ambiguous, for instance, the grid searching methods or the discriminative methods with binary classifier. However, when using the dense searching methods or the regression base trackers, only adopting the entropy score may not work well, which will be validated in the experimental part. Therefore, in the following, we use Eq. 4 to define the unary graph score, where the log likelihood term describes the consistency of the expert relative to the others, and the entropy prior represents the expert ambiguity to the target candidates provided by the expert ensemble.

At each time step, the expert ensemble $\mathbf{E}$ proposes a set of possible target candidate set $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, which are the image patches corresponding to all the local maxima given by the multi-expert. The label of the sample in the data set is denoted by $y_i = (l_i, \omega_i)$, where $l_i \in \mathbb{R}^2$ is the coordinate location of sample $\mathbf{x}_i$ and $\omega_i \in \{1, 0\}$ indicates whether the sample belongs to the foreground or background. Thus the label $Y = \{y_1, y_2, \ldots, y_n\}$ of the candidate set X resides in a very high dimension $\mathcal{Y} \in (\mathbb{R}^2 \times \{1, 0\})^n$. Now we adapt Eq. 4 by applying it to data X and the possible label set Z:

$$\mathcal{P}(X, Z, \theta_{\mathcal{T}}) = \mathcal{L}(\theta_{\mathcal{T}}; X, Z) - \eta H(Y|X, Z; \theta_{\mathcal{T}}). \tag{5}$$

For the likelihood term $\mathcal{L}(\theta_{\mathcal{T}}; X, Z)$, we further extract the global maxima $X_g = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{|\mathbf{E}|}\}$ of the multi-expert from X since these samples are more representative for describing the relationship of the experts. In semi-supervised learning paradigms, graph-based method is one class of typical ways to model the relationship of the data (the graph mentioned here is independent of our graph based multi-expert framework), where each node represents a sample in the data set and the edge denotes the strength proportional to the similarity of the pair-wise data [42]. In the graph-based method, the edges can be described by the affinity matrix W. In our problem, the affinity matrix describes the labels instead of data, which is based on the Gaussian function:

$$W_{mn} = \begin{cases} \exp(-\dfrac{\|l_m - l_n\|^2}{2\sigma^2}), & \text{if } m \neq n \\ 0, & otherwise, \end{cases} \tag{6}$$

where $m, n \in \{1, 2, \ldots, |\mathbf{E}|\}$, and $\sigma > 0$ is the parameter specified for bandwidth of the Gaussian function. Here, we omit the label element ω since X_g is the global maxima set for the experts. Intuitively, if the labels of the data are more similar, the likelihood should be higher. Moreover, in order to differentiate the expert that is more consistent with the others from its counterparts, we define the likelihood as: $\mathcal{C}(\theta_{\mathcal{T}}; X, Z) = \sum_{n=1}^{|\mathbf{E}|} W_{mn}$. Afterward, the log likelihood in Eq. 5 is obtained by calculating the natural logarithm of $\mathcal{C}(\theta_{\mathcal{T}}; X, Z)$.

In actual tracking scenario, the expert tends to be ambiguous due to continuous target appearance variations, especially when there are background clutters, heavy occlusions and abrupt motions. The entropy prior in Eq. 5 is used to give penalty to the expert ambiguity. The entropy can be evaluated as:

$$H(Y|X, Z; \theta_{\mathcal{T}}) = -\sum_{Y \in \mathcal{Y}} P(Y|X, Z, \theta_{\mathcal{T}}) \log P(Y|X, Z, \theta_{\mathcal{T}}). \tag{7}$$

As the entropy is computed in a very large label space $\mathcal{Y}$, a space narrowing strategy should be applied for efficiency. Here, we use the entropy term similar to [6], but with different space narrowing scheme that will be detailed in Section IV.

3) Binary Compatible Graph Score: According to the motivation of proposed framework, the design of binary compatible score should satisfy two criteria: a) Let the multi-expert framework favors the current tracker when the object is tracked stably, i.e. it's unnecessary to switch expert frequently under this situation; b) The same expert is regarded as compatible in order to get the accumulative score of the

same expert. Therefore, according to criterion a), we use the Gaussian function to define the binary compatible score for node m and node n based on the pair-wise label similarity: $\rho_{mn} = \exp(-\frac{\max\{0, \|l_m - l_n\| - r\}^2}{\tau^2})$, where l_m and l_n are the positions given by the global maxima of expert m and expert n respectively, and r is the threshold that has the same effect of "hinge loss", which means historical trackers are regarded as compatible with the current tracker if their label similarity is less than r. To achieve criterion b), for the same expert, we set the binary score as $\max(\rho_{mn})$.

C. Base Trackers for Multi-Expert Framework

As described in Section III-B, the graph score is based on the target candidate set X and its possible labels, therefore the proposed framework is compatible to most of the base trackers that are able to provide the response map for the target, such as the grid searching based tracking-by-detection methods or the dense searching based correlation filter methods. Due to the high efficiency and the impressive performance of SVM trackers [6], [17] and correlation filter trackers [21], [23], we choose the online SVM on a budget algorithm and the regression correlation filter (CF) as special cases for the proposed framework respectively. Here, we propose three Historical Snapshots Multi-Expert (HSME) trackers, HSME-SVM for online linear SVM, and HSME-CF, HSME-deep for regression correlation filter with hand-crafted features and CNN features respectively.

1) Base Tracker of Online Linear SVM: The online SVM base tracker is inspired by the online SVM training strategy in [6] and [7], which adopt the compact prototype sets to make an approximation to offline SVM. Furthermore, a prototype merge scheme is introduced to keep a budget of the model size. In this paper, we adopt the same base tracker (including the features) as in [6], which can compare the two frameworks directly. More details can be referred to [6] and [7].

2) Base Tracker of Regression Correlation Filter: We use the ridge regression model to learn the correlation of the temporal target context [21], [43]. By taking all the 2D circular shifts of image patch into consideration, the model produces less ambiguous response map than the binary classifier, which is more favorable for the likelihood in Eq. 5 to take effect. For the accuracy and low computational cost purpose, we train the CF in the dual space as indicated in [21]. It is worth noting that by deriving the CF in the dual space, multi-channel CF can be computed by only element-wise operations, which avoids the matrix inversions as in [43] and [44].

For the hand-crafted features, we employ the HOG feature described in [45]. Besides the HOG feature which puts more emphasis on the object shape, we further add the color feature to promote the tracker performance. Here we apply color attribute feature to map the RGB values to the probabilistic 11 dimensional color representation [46].

In order to deal with the scale variation of the target, we adopt a scale adaptive method to our tracker. Let $M_t \times N_t$ denotes the searching window size at time t, we first establish a target pyramid by cropping image patches, all of which are centered at the target position of time $t-1$, each of size $(1+as)M_t \times (1+as)N_t$, where a is a constant scalar of the scale factor, and $s \in \{\lfloor -\frac{N_s-1}{2} \rfloor, \lfloor -\frac{N_s-3}{2} \rfloor, \ldots, \lfloor \frac{N_s-1}{2} \rfloor\}$ is the scale index. Then all the N_s image patches are resized to the target template size. After that, the response map of each cropped image patch can be evaluated by CF, all of which constitute the response pyramid. Finally, the accurate scale index is indicated by the maximum of the response pyramid, as well as the translation (which should multiply by its ratio relative to the template size).

3) Base Tracker of CNN Feature: For HSME-deep tracker, we introduce the pre-trained CNN feature into the regression correlation filter. Unlike [31] and [35], which fine-tune the CNN models online, here, we only use the pre-trained model of VGG-Net with 19 layers as the feature extractor [28]. Inspired by [32], we use *conv3, conv4 and conv5* layers. Higher layers contain more semantic information while lower layers pay more attention to spatial details. In order to integrate the power of different layers effectively, each layer is convolved with the dual correlation filter to generate a response map. Since the response map of each layer is of different size caused by maxpooling, they are resized to the template size with bilinear interpolation. The final response map is obtained by stacking all the response maps with different weights [32].

In addition, the pre-trained CNN features put more emphasis on invariance representation of semantics with loss of spatial information, during the experiments, it is found that establishing the scale pyramid using CNN feature as in Section III-C.2 cannot estimate target scale accurately but with high computation load. Therefore, we train a separate CF only using HOG feature to estimate the target scale, while using CF of CNN feature to estimate target translation. We find this method works well in practice.

IV. Implementations and Method Analysis

A. Tracking by Multi-Expert Framework

Under the multi-expert framework modeled by discrete graph, object tracking is conducted as follows. The snapshots in the expert ensemble $\mathbf{E}$ are stored chronologically at intervals of Δt frames. For efficiency purpose, maximum $N_{\mathbf{E}}$ experts (including the current tracker) are maintained. If the number of experts exceeds $N_{\mathbf{E}}$, the oldest expert is discarded. The update procedure of the multi-expert framework is shown in Fig. 3. At each frame, the expert ensemble proposes the potential target candidates X. After that, unary scores of the proposed framework are calculated by Eq. 5, and binary compatible scores are computed according to Sec. III-B.3. The best expert hypothesis is then selected by figuring out the path with the highest score defined by Eq. 2, and the target state is decided by the best expert. Note that only expert $\mathcal{T}_T$ is updated, so the whole algorithm has low computation.

For HSME-SVM, only target translation is considered. For HSME-CF and HSME-deep, target scale is estimated according to the method described in Section III-C.2 and III-C.3 repectively. Generally, scale variation is much smaller than that of translation. For computation efficiency, the scale estimation is only conducted on expert $\mathcal{T}_T$ (current tracker). We find this strategy very effective in practice. Note that in HSME-deep, the CNN feature extraction takes the major part of computation

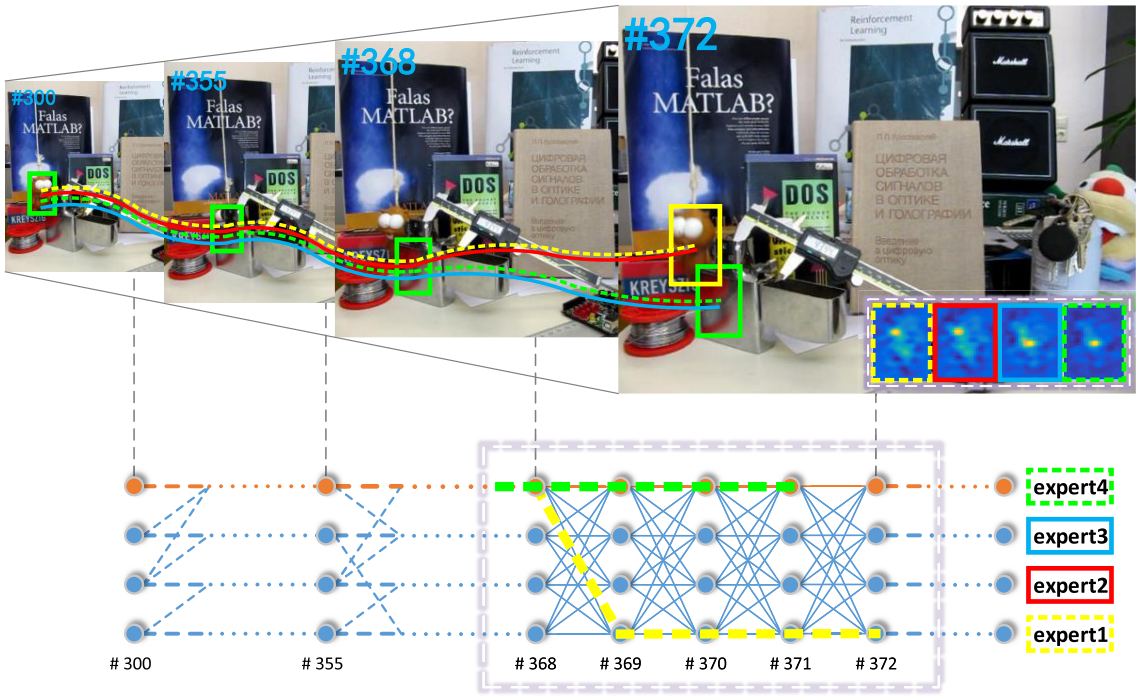

Fig. 4. Illustrations for Drift Correction by the proposed graph based multi-expert framework. Four experts are used, and they are stored in chronological order whose trajectories are colored by yellow, red, blue and green respectively (The green one denotes the current tracker). The current tracker tends to drift, and the multi-expert framework corrects the drift by selecting the historical tracker hypothesis colored by yellow. The corresponding expert selection results are shown in the graph. Note the framework finds the disagreement in frame 368 and corrects the drift in frame 372. The response maps of the experts in frame 372 are shown at the bottom-right corner of that image.

load. However, this forward pass process is operated only once at each frame, since the CNN feature extractor and CF classifier is separated, and only CF is updated at each time step.

In order to obtain the target candidate set X, we first use the non-maxima suppression method to get the local maxima of the multi-expert response maps, and the target candidates are the image patches corresponding to the local maxima whose response value is greater than the pre-defined threshold ε. The samples are then processed by hierarchical clustering according to their spatial positions. Samples of the same cluster are merged at their mean positions to avoid heavily overlap.

The whole HSME tracking algorithm flowchart is summarized in the table below.

B. Multi-Expert Framework Efficiency

In this section, we analyze how the proposed framework correct the drift and its advantages over heuristic multi-expert framework. As a typical example shown in Fig. 4, the current tracker tends to drift after the object being occluded, and its trajectory colored by green begins to be attracted by the distractor. However, the drift can be detected by the multi-expert framework. After calculating the path of the highest score in Frame 372, the true object trajectory is recognized, and the drift is corrected by the historical tracker. In [6] and [36], the best expert is selected with the highest accumulated score for numbers of the latest frames, where the frame number is a pre-defined parameter, while our proposed framework omits this parameter setting.

Algorithm 1 HSME Tracker

input : Initial target bounding box $\mathbf{b}_1$
output: Estimated target state $\mathbf{b}_t = (\hat{x}_t, \hat{y}_t, \hat{s}_t)$

$\mathbf{E} \leftarrow \mathcal{T}_1$;
Initialize the multi-expert graph as in Fig. 3;
for $t = 1 : N_{frames}$ **do**
  Get the target candidate set X proposed by $\mathbf{E}$;
  for $\mathcal{T} \in \boldsymbol{E}$ **do**
    if $\mathcal{T} = \mathcal{T}_t$ **then**
      Build the target pyramid at $(\hat{x}_{t-1}, \hat{y}_{t-1})$;
      Compute the response pyramid, estimate the target location $(x_{\mathcal{T}_t}.y_{\mathcal{T}_t})$ and scale $\hat{s}_t$;
    else
      Get the response map and estimate the target location $(x_{\mathcal{T}}, y_{\mathcal{T}})$;
    Evaluate the graph unary and binary score according to Section III-B2 and Section III-B3;
  Select $\mathcal{T}^* \in \mathbf{E}$ according to Section III-A2;
  Output the target state $\mathbf{b}_t = (x_{\mathcal{T}_t^*}, y_{\mathcal{T}_t^*}, \hat{s}_t)$;
  if $\mod(t, \Delta t) == 0$ **then**
    $\mathbf{E} \leftarrow \mathbf{E} \cup \mathcal{T}_t$;
    discard the oldest expert when $|\mathbf{E}| > N_{\mathbf{E}}$;
  Re-train the current expert $\mathcal{T}_t$;
  Update the multi-expert graph as in Fig. 3;

To be more specific, under the proposed framework, the multiple experts usually give very similar hypotheses of the target state when the object is tracked stably. Therefore, in

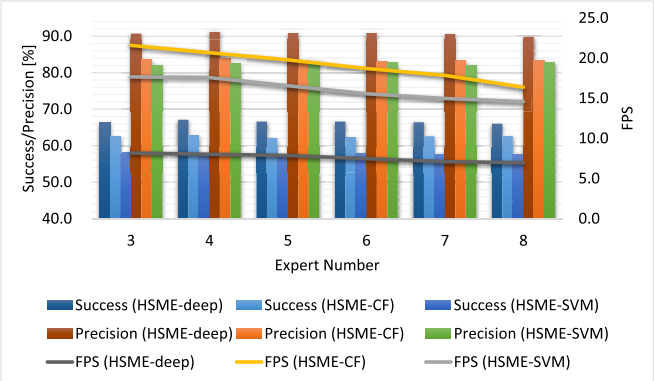

Fig. 5. The success/precision scores and FPS of HSME-deep, HSME-CF and HSME-SVM when the expert number varies. The bar plots show the success/precision scores of the three trackers with y-axis on the left, while the curve plots show the FPS of the three trackers with y-axis on the right.

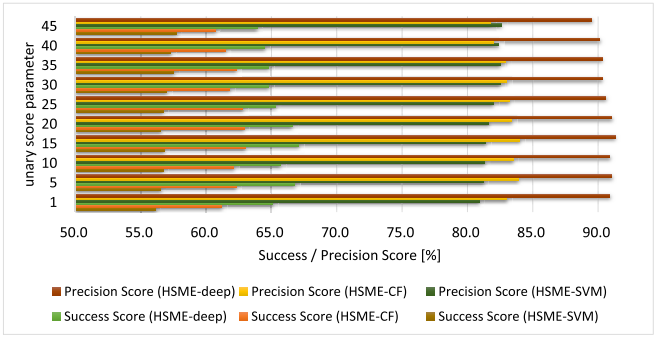

Fig. 6. The success/precision scores of HSME-deep, HSME-CF and HSME-SVM when the unary score parameter varies.

this situation, the multi-expert gives object trajectories approximate to the current tracker. However, when there is disagreement among the expert ensemble, the multi-expert framework is capable of selecting the most reliable object trajectory implicitly by estimating all their recent hypotheses. Moreover, with the dynamic programming solver, the above advantage can be obtained by only evaluating the expert scores of the current frame. Assuming there is $|\mathbf{E}|$ experts at each frame, the computation complexity takes about $\mathcal{O}(|\mathbf{E}|^2)$ time. Note the backward tracing step in Eq. 3 is not necessary in real tracking process.

C. Multi-Expert Framework Robustness

In this section, the robustness of the proposed framework is tested by analyzing the three main parameters that may influence the tracking performance.

One of the main parameters of the proposed framework is the number of experts $N_{\mathbf{E}}$. Since the historical tracker snapshots can be thought of the memory storing the target and the background information, intuitively, increasing the number of experts will enlarge the information content. On the other hand, increasing the expert number will result in efficiency decline. Therefore, we conduct the experiments to explore how expert number influences the tracking performance. We test the tracking performance of HSME-deep, HSME-CF and HSME-SVM on TB-50 dataset using the OPE evaluation [8], and plot their success and precision scores jointly with their frames per second (FPS) variation. We test the expert number from 3 to 8, and the experimental results are shown in Fig. 5. From the figure, it is observed that tracking performances are relatively stable with the expert number varies. However, the FPS criterion falls more significantly when the expert number increases. The FPS of HSME-deep declines more gently than HSME-CF and HSME-SVM since most of the time is taken on CNN feature extraction, and the time for forward passes on CNN is the same as the expert number increases. Moreover, the performance scores reach the highest from number of 3 to number of 4, and drop slightly in number of 5. The scores begin to increase from number of 5, and improve gently afterwards. Taken the FPS into consideration. we set the expert number as 4 in this work since both the performance and efficiency are satisfactory at this point.

The second important parameter is the trade-off coefficient η of the unary graph score in Eq. 5. The parameter η indicates what the weight for the likelihood and entropy prior should be to describe the reliability of multi-expert composed by a specified base tracker. Intuitively, the importance of the two terms in Eq. 5 should not be the same since different base tracker presents different property. The experiment for unary score parameter η is conducted, and Fig. 6 shows the performance of the three proposed trackers when η increases from 1 to 45. From the test results, it is observed that with η increasing, the scores of correlation filter based ensemble trackers (HSME-deep and HSME-CF) first rise, and reach highest when η is around 15, then decline slightly when η becomes bigger. It also can be seen that when the weight of entropy prior becomes very high (η is larger than 35), the performances of HSME-deep and HSME-CF decline obviously. However, the performance of HSME-SVM rises monotonously when η grows. This is because that the energy of response map estimated by correlation filter tracker is focused on the peaks, so that the likelihood term plays a more important role than entropy prior, and a proper trade-off of the two terms presents satisfactory tracking performance. On the other hand, the response map given by the grid search based SVM tracker is more ambiguous than that of correlation filter, which favors the entropy prior to describe the reliability of the base tracker. As a result, we choose $\eta = 15$ for HSME-deep and HSME-CF, and $\eta = 45$ for HSME-SVM.

The third parameter is the variance σ of the affinity matrix in Eq. 6, which constitutes the likelihood term in the graph score. In HSME-deep and HSME-CF, this parameter describes the similarity of the expert hypotheses. In order to test whether σ shows different influence with different target size, we define the likelihood variance factor f that is the proportional to the target size s, which we set it as the multiples of 3σ for convenience according to the 3-sigma rule of Gaussian distribution, i.e. $s = 3\sigma \times f$. Fig. 7 shows the success and precision scores varies as f varies from 0.2 to 3. Note the highest score is $f = 1$. Then the scores decline with f increases. Therefore, we set $f = 1$.

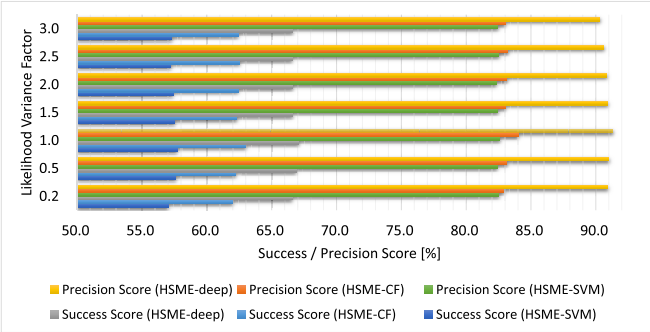

Fig. 7. The success/precision scores of the proposed three trackers when the likelihood variance factor varies. Refer to Section IV-C for details.

V. Experiments

In this section, we evaluate the proposed three trackers HSME-deep, HSME-CF and HSME-SVM on three large datasets, the first is the 50 sequences Object Tracking Benchmark (TB-50) [8], the second dataset TB-100 [9] extends TB-50 to 100 sequences, the third dataset is the 60 sequences VOT2015 Challenge dataset [10], [13]. Moreover, we also compare the proposed trackers to their base trackers to validate the effectiveness of the proposed framework.

A. Experimental Setup

1) Experimental Platform and Metrics: The proposed trackers, HSME-deep, HSME-CF and HSME-SVM are implemented in Matlab&C, where HSME-deep is based on MatConvNet toolbox [47]. HSME-CF and HSME-SVM run at roughly 20 fps and 15 fps respectively on the 3.20GHz CPU with 8GB RAM. HSME-deep runs about 8 fps with NVIDIA Tesla K20c 5GB GPU. Although with multiple experts, the proposed trackers are with high efficiency, mostly due to the efficiency of the dynamic programming solver and the low computation load of the base trackers, especially the correlation filter.

In the TB-50 and TB-100 datasets, the quantitative analysis is illustrated on two evaluation plots: (i) the success plot and (ii) the precision plot. The success plot is based on the bounding box overlap metric, and shows the percentage of successful frames at the overlap threshold varies from 0 to 1. The ranking is according to the area under curve (AUC) score. The precision plot shows the ratio of frames whose center location error (CLE) is within a given threshold. The VOT2015 dataset contains 60 short term challenging sequences. The sequences are annotated using rotated bounding box in order to provide highly accurate ground truth, which is different from those of TB-50 and TB-100 annotated by rectangles. The VOT2015 dataset is evaluated by two criteria: (i) accuracy and (ii) robustness. The accuracy measures the overlap between the tracking result and ground truth. The robustness measures how many times the tracker loses the target.

2) Parameter Setup: The parameters setup of the multi-expert framework is as follows. The max number of experts $N_{\mathbf{E}}$ is set to 4. The frame interval Δt is set to 50, and we find the performance is not sensitive to tens of frames such as 30 or 40. The trade-off parameter of the graph score in Eq. 1 is set to $\beta = 1$ for simplicity. Let the template target size denoted by s. We set the cutoff distance of the hierarchical clustering by $s/2$. The parameter σ for computing the likelihood in Eq. 6 is $s/3$. Note that all the response maps of the expert ensemble (including each layer of response pyramid generated by the current expert) are of the same template size. Therefore the above parameters are not influenced by target size. The trade-off parameter η in Eq. 5 is set to 15 for HSME-deep / HSME-CF, and 45 for HSME-SVM. The candidate selection threshold $\varepsilon = 0.8$.

For the online SVM base tracker, we set parameters the same as those in [6] for fair comparisons in the experiments. For the correlation filter base tracker, only linear kernel is applied, and the padding size and interpolation learning rate are set to 1.8 and 0.01 respectively. The number of target pyramid layers N_s is 9, and the scale factor a is 0.005. The template size is set to the initial target size. For HSME-deep, the weights for stacking response maps are set to 1, 0.5 and 0,02 for *conv5, conv4* and *conv3* respectively similar to [32].

B. TB-50 Dataset

Besides the trackers provided by the benchmark [8], we also compare our methods with many state-of-the-art trackers, including STCT [35], HCT [32], Staple [26], MEEM [6], TGPR [48], CN [22] and KCF [21], where STCT and HCT are based on Convolutional Neural Networks.

1) Overall Performance: According to the evaluation methods by TB-50, the one-pass evaluation (OPE) performance is illustrated in the Success Plot and Precision Plot shown in Fig. 8. As shown in the plot, HSME-deep achieves 67.1% success rate and 91.3% precision rate, both of which rank first among all the compared trackers. HSME-CF gets 63% success rate and 84% precision rate, which ranks only behind the CNN trackers, STCT and HCT. Particularly, MEEM is also based on historical tracker ensemble, and it has the same online SVM base tracker as the proposed HSME-SVM. Compared to MEEM, HSME-SVM improves the performance results significantly by 2.5% of the AUC score and 2.9% of the CLE score, which shows the proposed framework works better than the original framework based on simple accumulated score. The proposed HSME-deep tracker further boosts the performance, and it surpasses HSME-CF and HSME-SVM with large margin, especially exceeds in the CLE score by 8.6% compared to HSME-CF. HCT is also the correlation filter based tracker with CNN feature, which can be regarded as our base tracker. Compared to HCT, HSME-deep improves the success score by 10.9%. The overall plot demonstrates the proposed trackers are effective and promising.

2) Robustness to Initialization: Visual trackers are usually sensitive to different initializations. In this part, we evaluate the robustness to initialization of the proposed methods according to the strategy in [8]. Two evaluation strategies are used: temporal robustness (TRE) and spatial robustness (SRE). The TRE and SRE test results for TB-50 are shown in Fig. 8. From the plot, the proposed HSME-deep ranks first in the two tests

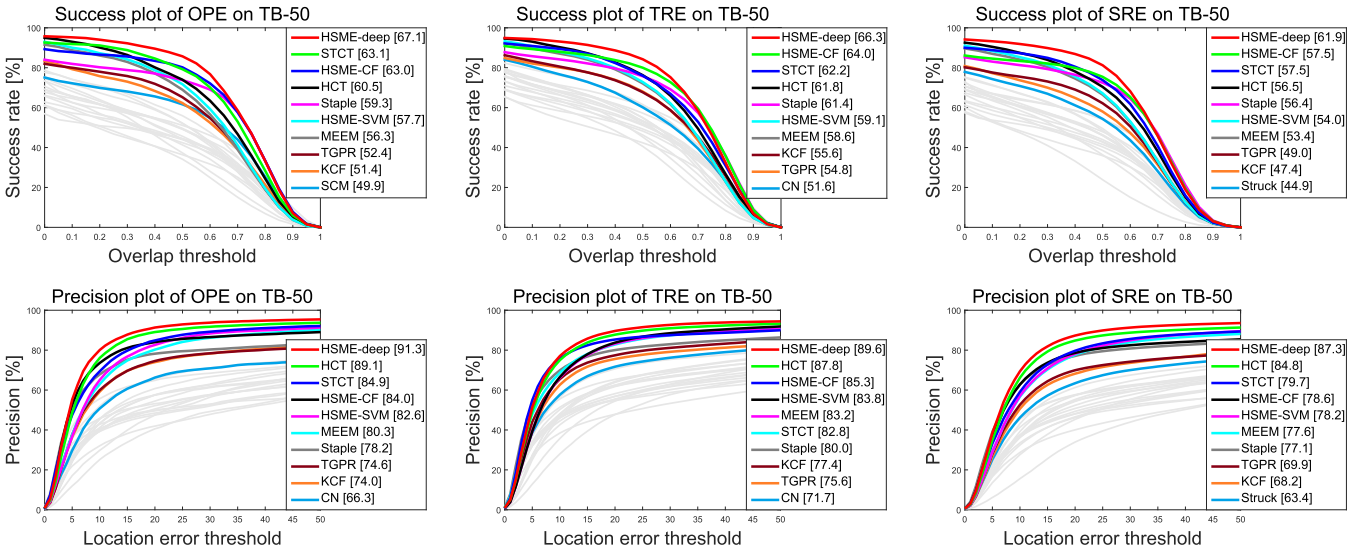

Fig. 8. The success plots and precision plots over TB-50 dataset using one pass evaluation (OPE), temporal robustness evaluation (TRE) and spatial robustness evaluation (SRE). The legend illustrates the area under curve (AUC) for the success plot, and the score of the threshold 20 for the precision plot. Only scores of the top-10 trackers are shown, and the others are plotted in light gray curves.

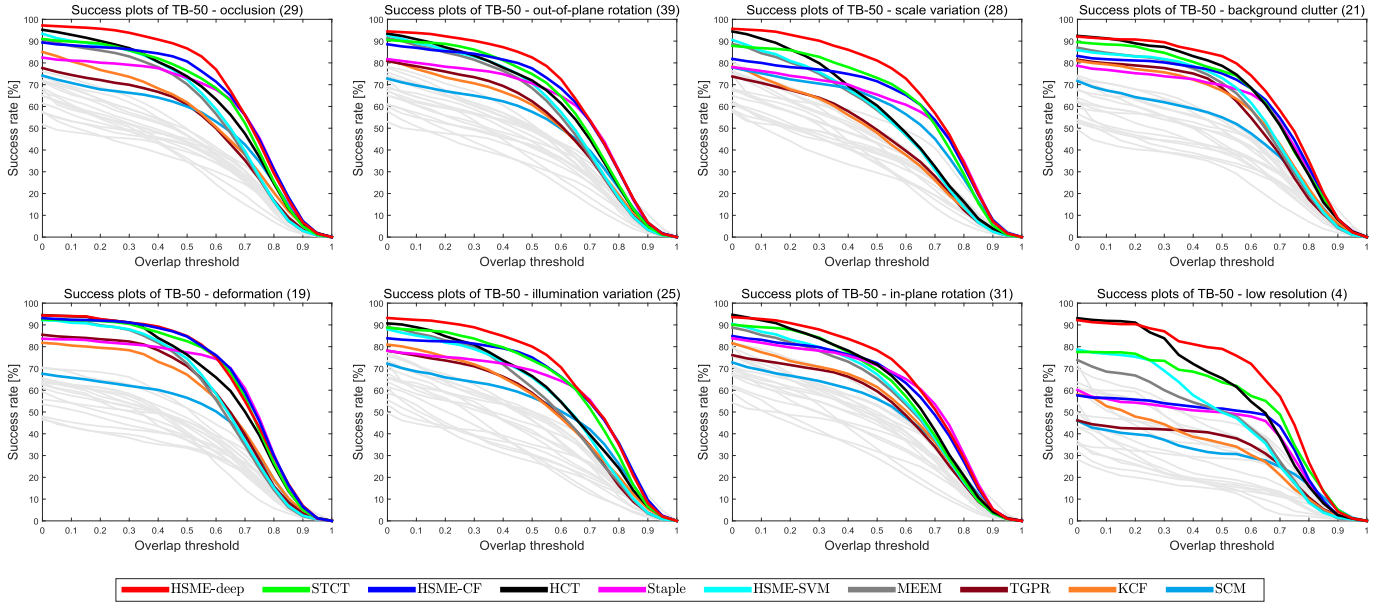

Fig. 9. Success plots for eight attributes of the top-10 performance trackers on TB-50. The test attributes include occlusion (OCC), out-of-plane rotation (OPR), scale variation (SV), background clutter (BC), deformation (DEF), illumination variation (IV), in-plane-rotation (IPR) and low resolution (LR). The number in the brackets of the plot caption indicates the sequence count in TB-50 belongs to that attribute.

for both the success plot and precision plot, and HSME-CF ranks second in both the success rates. It can also be found that HSME-CF is robust to different initialization, which performs even better than the CNN trackers in both success plots.

3) Attribute Performance: In this experiment, the benchmark sequences are divided into 8 main attributes to evaluate the trackers in different scenarios. We choose the top-10 trackers in the OPE test to further evaluate their performance in the attribute test. As described in [8], the AUC score measures the tracker performance more accurately than CLE that is with one threshold, so the success plot is the main analysis evaluation tool. Therefore, we report the eight main attributes of success plots in Fig. 9, whose scores are summarized in Table I.

As illustrated in the plots, the proposed trackers rank first in all the attributes. Particularly, in seven of eight attributes, HSME-deep ranks first, and exceeds the second one with large margin. Among all the attributes, the scale variation performance is improved significantly, which shows our scale scheme is very effective. In addition, HSME-CF also gets more favorable scores than other correlation filter based trackers, Staple [26] and KCF [21], which demonstrates the effectiveness of the multi-expert framework. HSME-SVM performs

TABLE I

SCORES OF THE ATTRIBUTE PLOTS IN FIG. 9. RED, GREEN AND BLUE DENOTE THE BEST, SECOND AND THIRD PERFORMANCE RESPECTIVELY

	HSME-deep	STCT	HSME-CF	HCT	Staple	HSME-SVM	MEEM	TGPR	KCF	SCM
OCC	67.2	61.2	62.8	60.6	58.5	57.5	55.8	48.3	51.4	48.7
OPR	64.7	59.8	61.0	58.7	56.9	56.6	55.4	50.4	49.5	47.0
SV	65.5	59.4	57.4	53.1	54.5	50.6	50.2	42.6	42.7	51.8
BC	65.3	61.0	59.3	62.3	55.7	56.6	55.7	52.9	53.5	45.0
DEF	65.6	64.4	65.8	62.6	60.7	57.9	57.1	55.4	53.4	44.8
IV	64.3	60.2	60.0	56.0	56.1	53.9	52.6	48.2	49.3	47.3
IPR	62.5	56.5	57.2	58.2	57.6	54.7	53.6	47.6	49.7	45.8
LR	62.5	52.5	40.8	55.7	39.5	44.4	41.6	30.1	31.2	27.9

red: rank1, green: rank 2, blue: rank 3

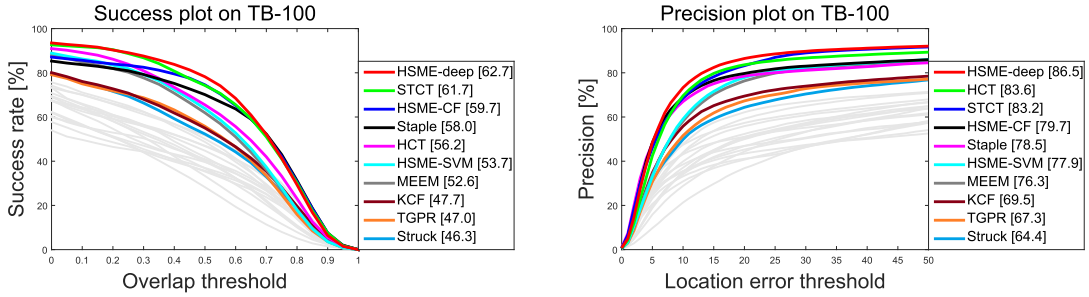

Fig. 10. The success plot and precision plot over TB-100 dataset using one pass evaluation (OPE). Legends for tracker scores are located beside each plot. Only top-10 trackers are colored, and the others are shown in gray.

Fig. 11. Long term sequences snapshots. Sequence name is at the lower left corner.

TABLE II

SUCCESS AND PRECISION SCORES FOR THE PROPOSED TRACKERS AND THEIR BASE TRACKERS. (DARKER CELLS DENOTE HIGHER SCORES)

	HSME-deep	HSME-deep-base	HSME-CF	HSME-CF-base	HSME-SVM	HSME-SVM-base	
OPE	67.1	63.3	63.0	57.4	57.7	55.8	Success
TRE	66.3	64.6	64.0	58.5	59.1	57.4	
SRE	61.9	59.8	57.5	56.5	54.0	52.2	
OPE	91.3	87.1	84.0	80.4	82.6	80.3	Precision
TRE	89.6	87.8	85.3	82.6	83.8	81.2	
SRE	87.3	84.7	78.6	77.1	78.2	75.6	

favorably against the counterpart MEEM, especially in low resolution sequences with significant improvement.

4) Framework Effectiveness Validation: In this experiment, we compare each of our proposed trackers with their base trackers to evaluate the effect of the proposed multi-expert framework. We summarize their success and precision scores in Table II. From the table, both the success and precision scores of the three proposed trackers have significant improvement compared with their base trackers. From the success score perspective, the improvement on CF is higher than that of SVM, which is majorly due to the likelihood term that is more suitable for CF base trackers. Besides, the results of HSME-SVM and HSME-SVM-base demonstrate that the proposed framework pays more attention to correcting the translation estimation error by selecting the reliable expert.

C. TB-100 Dataset

1) Overall Performance: We further compare our methods with the trackers mentioned in Section V-B on the recently introduced TB-100 dataset to validate their performances on relatively large dataset. As illustrated in Fig. 10, the top-10 trackers are colored in the plots, and the remainders are only drawn in light gray color. The evaluation criteria of the success plot and the precision plot are also the previously mentioned AUC and CLE score. As shown in the plots, the proposed HSME-deep ranks first in both the success and precision criteria. And the scores of HSME-CF are only lower than CNN based trackers, STCT and HCT. HSME-SVM also ranks higher than MEEM in both plots.

2) Long Term Sequences: In this section, we test the proposed trackers with the top trackers in the overall performance on the relatively long sequences (at least $>$ 1000 frames). The test sequences are shown in Fig. 11. *Car24* and *Girl2* are the videos of real scenarios, and *RedTeam* is the aerial video, while *Sylvester* and *Box* are normal tracking test sequences. The metrics are the average AUC (in decimal) and CLE

TABLE III
AUC \CLE SCORES FOR LONG TERM SEQUENCES SHOWN IN FIG. 11

Sequence	*Frames*	HSME-deep	HSME-CF	HSME-SVM	STCT	HCT	Staple	MEEM
Car24	3059	0.73 \ 4.32	0.71 \ 5.13	0.43 \ 7.52	0.68 \ 5.39	0.41 \ 7.93	0.43 \ 5.85	0.43 \ 7.48
RedTeam	1918	0.47 \ 3.69	0.46 \ 6.36	0.51 \ 4.06	0.52 \ 4.71	0.46 \ 4.80	0.57 \ 3.04	0.50 \ 4.31
Girl2	1500	0.61 \ 31.73	0.55 \ 33.72	0.60 \ 19.30	0.07 \ 294.07	0.07 \ 118.63	0.11 \ 114.12	0.57 \ 36.10
Sylvester	1345	0.77 \ 4.31	0.69 \ 9.32	0.67 \ 9.33	0.65 \ 9.72	0.65 \ 12.68	0.56 \ 14.16	0.66 \ 10.01
Box	1161	0.76 \ 8.50	0.75 \ 9.73	0.66 \ 12.70	0.72 \ 11.33	0.28 \ 107.23	0.35 \ 90.85	0.63 \ 15.60
Mean	\	0.67 \ 10.51	0.63 \ 12.85	0.57 \ 10.58	0.53 \ 65.04	0.37 \ 50.25	0.40 \ 45.60	0.56 \ 14.7

red: rank1, green: rank 2, blue: rank 3; AUC \ CLE

Fig. 12. Tracking results of the top nine algorithms (HSME-deep, HSME-CF, HSME-SVM, STCT [35], Staple [26], HCT [32], MEEM [6], KCF [21], TGPR [48]) on TB-100 over ten typical sequences. The video illustrations from top-left to bottom-right are *Walking2, Dog1, Box, Skater2, Girl2, Human4, Jogging, Skiing, MotorRolling and Skating1*.

in this test. The results are shown in Table III. From the table, HSME-deep ranks first for both the AUC and CLE mean measurements. It is also observed that for long term sequences, the multi-expert framework is capable of improving the tracking performance, especially in *Girl2*, with frequent occlusions, and deformations, even the CNN trackers, STCT and HCT, fail to track the target, while our trackers with the HSME framework can deal with this situation.

3) Typical Results Analysis: Some typical tracking results of the top trackers are shown in Fig. 12. Among all the test sequences, *Dog1* and *Box* have significant scale variations. *Walking2*, *Girl2*, *Human4*, *Jogging* and *Skating1* go through part or whole occlusion. In addition, *Skating1* also have illumination changes due to the stage light. Some targets in the sequences suffer from frequent appearance variations and non-rigid appearance changes, such as in *Box*, the object has multiple views, and in *Skater2*, the skater has large non-rigid appearance transformations. *Skiing* and *MotorRolling* have severe deformations, which are very hard to track. From *Dog1* and *Box*, we can see that HSME-deep and HSME-CF perform well in handling scale variation. In *Skiing*, where most of the compared trackers lose the target, HSME-deep, HSME-SVM and HCT are capable of catching the object. It is also noted that in *MotorRolling*, only CNN feature based trackers, HSME-deep, STCT and HCT can track the target because of the high rotation deformation. There are also the results to demonstrate the superiority of HSME-SVM over MEEM. In *Girl2*, when an adult walks in front of the girl, MEEM and many other trackers are drifted by the occlusion, while HSME-SVM corrects the drift and continues to track the truth target. The same phenomenon can be observed in the *Walking2* sequence. In *Human4*, MEEM is drifted by the distractor, while HSME-SVM continues to track the target, which also demonstrates the advantage of our framework to that of MEEM. The same conclusion can also be drawn from sequence *Jogging*.

TABLE IV
THE RESULTS OF VOT2015 CHALLENGE DATASET

	Accuracy	Failures	Overall Rank
MDNet	0.56	0.72	10.67
DeepSRDCF	0.53	1.12	14.30
HSME-deep	0.53	1.24	14.33
SRDCF	0.52	1.31	15.34
sPST	0.51	1.29	16.82
Staple	0.52	1.56	17.04
NSAMF	0.49	1.31	17.29
RAJSSC	0.52	1.66	17.42
SO-DLT	0.54	1.81	17.85
SC-EBT	0.52	1.37	18.14
OACF	0.52	1.84	18.18
EBT	0.45	1.05	18.21
HCT	0.47	1.27	18.51
STCT	0.51	1.56	18.59
S3Tracker	0.47	1.80	18.71
LDP	0.45	1.36	18.83
SumShift	0.47	1.75	18.84
AOG	0.47	1.88	19.57
HSME-CF	0.47	2.04	19.65
ASMS	0.46	1.98	20.13
MvCFT	0.48	2.24	20.36
sme	0.48	2.26	20.50
HSME-SVM	0.46	2.24	20.62
RobStruck	0.44	1.97	20.86
SAMF	0.43	2.24	21.63
DSST	0.49	2.31	20.99
Struck	0.44	2.11	21.24
MEEM	0.44	2.41	21.37
G2T	0.43	2.25	21.64
MKCF-plus	0.43	2.33	21.72
DAT	0.44	2.36	21.83
MCT	0.42	2.36	22.00
Dtracker	0.43	2.38	22.04
MUSTer	0.44	2.47	22.72
HMMxD	0.44	2.48	22.82
TGPR	0.45	2.31	23.09
TRIC	0.44	2.34	23.18
KCF	0.43	2.53	25.54

[1]red: rank 1, blue: rank 2, green: rank 3

D. VOT2015 Challenge Dataset

The number of sequences in VOT2015 Challenge Dataset has been enlarged to 60 compared to VOT2013 and VOT2014, whose numbers of sequences are 16 and 25 respectively [10]. Besides the trackers participated in the VOT2015 challenge, we also compare with 3 other state-of-the-art trackers, including STCT [35], HCT [32], Staple [26].

The accuracy and failures, as well as the overall ranking results for all the tested trackers are listed in Table IV. As there is a large number of compared trackers, only the trackers rank higher than KCF [21] are listed. According to the VOT evaluation criteria [13], the overall experimental results are illustrated in accuracy-robustness (AR) ranking plot, as shown in Fig. 13. The AR ranking plot shows average ranking scores of all the sequences for each tracker in the joint accuracy-robustness rank space. The details for the evaluation method can be referred to [10] and [13]. From the table and the plot, it is observed that the proposed HSME-deep ranks higher than most of the compared trackers, and only MDNet (the VOT2015 winner) and DeepSRDCF rank higher than HSME-deep. However, the AR ranking plot shows that the speed of the proposed HSME-deep is about 9 fps, which is much faster than MDNet and DeepSRDCF, because they both rely on iterative optimization operations online. Besides, HSME-CF ranks well in accuracy, even better than CNN tracker HCT. Finally, HSME-SVM ranks higher than MEEM as well.

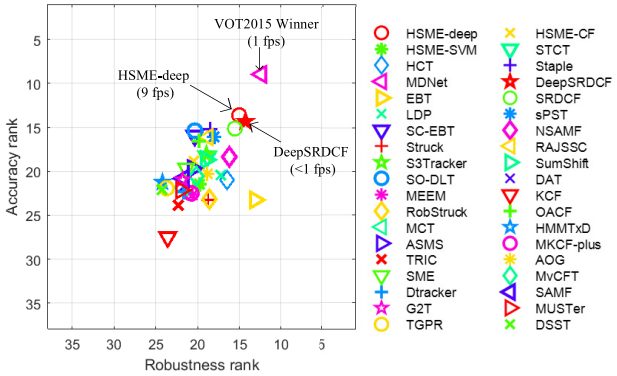

Fig. 13. The AR ranking plot for VOT2015 Dataset. The tracker is better if its legend resides closer to the top-right corner of the plot.

VI. CONCLUSION

In this paper, we propose an effective discrete graph based multi-expert framework to handle the tracker drift problem. The graph nodes are modeled by the hypotheses of the multiple experts, which are composed of the current tracker and the historical trained tracker snapshots. As the drift tracker tends to be more confident to its own estimation, the graph score is defined in a semi-supervised learning manner to describe the relationship of the multiple experts as well as their ambiguity. With the dynamic programming solver, the tracker drift is automatically detected and corrected by analyzing the recent performance of the multi-expert with only evaluating the graph scores of the current frame. Three base trackers, online SVM with a budget, and regression correlation filter (hand-crafted features and CNN features) are integrated into the proposed framework. The experiments are conducted on three large tracking datasets, which demonstrate the proposed trackers perform favorably against state-of-the-art methods.

REFERENCES

[1] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1269–1276.
[2] J. Kwon and K. M. Lee, "Tracking by sampling and integrating multiple trackers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1428–1441, Jul. 2014.
[3] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1838–1845.
[4] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, Jul. 2012.
[5] N. Wang, S. Li, A. Gupta, and D.-Y. Yeung. (2015). "Transferring rich feature hierarchies for robust visual tracking." [Online]. Available: https://arxiv.org/abs/1501.04587
[6] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: Robust tracking via multiple experts using entropy minimization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 188–203.
[7] Z. Wang and S. Vucetic, "Online training on a budget of support vector machines using twin prototypes," *Statist. Anal. Data Mining*, vol. 3, no. 3, pp. 149–169, Jun. 2010.

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

[9] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.

[10] M. Kristan *et al.*, "The visual object tracking VOT2015 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2015, pp. 1–23.

[11] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. Van Den Hengel, "A survey of appearance models in visual object tracking," *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 4, p. 58, Sep. 2013.

[12] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1442–1468, Jul. 2014.

[13] M. Kristan *et al.*, "A novel performance evaluation methodology for single-target trackers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2137–2155, Nov. 2016.

[14] S. Avidan, "Support vector tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 8, pp. 1064–1072, Aug. 2004.

[15] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, Aug. 2011.

[16] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2012, pp. 864–877.

[17] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 263–270.

[18] G. Zhu, J. Wang, C. Zhao, and H. Lu, "Weighted part context learning for visual tracking," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5140–5151, Dec. 2015.

[19] D. S. Bolme, J. R. Beveridge, B. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2544–2550.

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

[21] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

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

[23] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 1–11.

[24] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1430–1438.

[25] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "MUlti-store tracker (MUSTer): A cognitive psychology inspired approach to object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 749–758.

[26] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1401–1409.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1-14.

[29] K. He, X. Zhang, S. Ren, and J. Sun. (2015). "Deep residual learning for image recognition." [Online]. Available: https://arxiv.org/abs/1512.03385

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

[31] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 597–606.

[32] C. Ma, J.-B. Huang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3074–3082.

[33] K. Zhang, Q. Liu, Y. Wu, and M.-H. Yang, "Robust visual tracking via convolutional networks without training," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1779–1792, Apr. 2016.

[34] B. Cai, X. Xu, X. Xing, K. Jia, J. Miao, and D. Tao, "BIT: Biologically inspired tracker," *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1327–1339, Mar. 2016.

[35] L. Wang, W. Ouyang, X. Wang, and H. Lu, "STCT: Sequentially training convolutional networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1373–1381.

[36] J. Li, Z. Hong, and B. Zhao, "Robust visual tracking by exploiting the historical tracker snapshots," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop*, Dec. 2015, pp. 41–49.

[37] D.-Y. Lee, J.-Y. Sim, and C.-S. Kim, "Multihypothesis trajectory analysis for robust visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5088–5096.

[38] P. F. Felzenszwalb and R. Zabih, "Dynamic programming and graph algorithms in computer vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 4, pp. 721–740, Apr. 2011.

[39] H. Ishikawa, "Exact optimization for Markov random fields with convex priors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1333–1336, Oct. 2003.

[40] X. Zhu, J. Lafferty, and R. Rosenfeld, "Semi-supervised learning with graphs," Ph.D. dissertation, Lang. Technol. Inst., School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, 2005.

[41] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 529–536.

[42] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Synth. Lect. Artif. Intell. Mach. Learn.*, vol. 3, no. 1, pp. 1–130, 2009.

[43] H. K. Galoogahi, T. Sim, and S. Lucey, "Multi-channel correlation filters," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3072–3079.

[44] V. N. Boddeti, T. Kanade, and B. V. K. Kumar, "Correlation filters for object alignment," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2291–2298.

[45] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[46] J. van de Weijer, C. Schmid, J. Verbeek, and D. Larlus, "Learning color names for real-world applications," *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1512–1523, Jul. 2009.

[47] A. Vedaldi and K. Lenc, "MatConvNet: Convolutional neural networks for MATLAB," in *Proc. ACM Conf. Multimedia Conf.*, 2015, pp. 689–692.

[48] J. Gao, H. Ling, W. Hu, and J. Xing, "Transfer learning based visual tracking with Gaussian processes regression," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 188–203.

Jiatong Li received the Ph.D. degree from the School of Information and Electronics, Beijing Institute of Technology. He is currently pursuing the Ph.D. degree with the Faculty of Engineering and Information Technology, University of Technology Sydney. His research interests include image processing and computer vision, especially with machine learning and signal processing algorithms.

Chenwei Deng (M'09–SM'15) received the Ph.D. degree in signal and information processing from the Beijing Institute of Technology, Beijing, China, in 2009. He was a Post-Doctoral Research Fellow with the School of Computer Engineering, Nanyang Technological University, Singapore. Since 2012, he has been an Associate Professor and then a Full Professor with the School of Information and Electronics, Beijing Institute of Technology. He has authored or co-authored over 50 technical papers in refereed international journals and conferences, and co-edited one book. His current research interests include video coding, quality assessment, perceptual modeling, feature representation, object recognition, and tracking.

Richard Yi Da Xu is currently the Director of Machine Learning and Data Analytics Laboratory, Global Big Data Technology Centre, University of Technology Sydney. He has been a First Line Researcher in machine learning, data analytics, computer vision, and deep learning and he has authored over 50 peer-reviewed publications, including the IEEE TRANSACTIONS NNLS, the IEEE TRANSACTIONS CYBERNETICS, the IEEE TRANSACTIONS IMAGE PROCESSING, the IEEE TRANSACTIONS KDE, the IEEE SPL, Pattern Recognition, the ACM T. KDD, the AAAI, the IJCAI, and the AI-Statistics. He has co-authored papers with some of the world's best statistical machine learning researchers with Oxford and Cambridge University.

Dacheng Tao (F'15) was a Professor with the Computer Science and Director of Centre for Artificial Intelligence, University of Technology Sydney. He is currently a Professor of Computer Science and ARC Future Fellow with the School of Information Technologies and the Faculty of Engineering and Information Technologies, and the Founding Director of the UBTech Sydney Artificial Intelligence Institute, The University of Sydney. He mainly applies statistics and mathematics to Artificial Intelligence and Data Science. His research results have expounded in one monograph and over 500 publications at prestigious journals and prominent conferences, such as the IEEE T-PAMI, T-NNLS, T-IP, JMLR, IJCV, NIPS, CIKM, ICML, CVPR, ICCV, ECCV, AISTATS, ICDM, and the ACM SIGKDD. His research interests spread across computer vision, data science, image processing, machine learning, and video surveillance. He received several best paper awards, such as the Best Theory/Algorithm Paper Runner Up Award in the IEEE ICDM'07, the Best Student Paper Award in the IEEE ICDM'13, and the 2014 ICDM 10-Year Highest-Impact Paper Award. He received the 2015 Australian Scopus-Eureka Prize, the 2015 ACS Gold Disruptor Award, and the 2015 UTS Vice-Chancellor's Medal for Exceptional Research. He is a Fellow of the OSA, IAPR, and SPIE.

Baojun Zhao received the Ph.D. degree in electromagnetic measurement technology and equipment from the Harbin Institute of Technology, Harbin, China, in 1996. From 1996 to 1998, he was a Post-Doctoral Fellow with the Beijing Institute of Technology, Beijing, China. He is currently a Full Professor, the Vice-Director of the Laboratory and Equipment Management and the Director of the National Signal Acquisition and Processing Professional Laboratory. He has authored or co-authored over 100 publications and received five provincial/ministerial-level scientific and technological progress awards in these fields. His main research interests include image/video coding, image recognition, infrared/laser signal processing, and parallel signal processing.