Unifying Temporal Context and Multi-Feature With Update-Pacing Framework for Visual Tracking

Yuefang Gao, Zexi Hu, Henry Wing Fung Yeung, Yuk Ying Chung, *Member, IEEE*, Xuhong Tian, and Liang Lin

***Abstract*— Model drifting is one of the knotty problems that seriously restricts the accuracy of discriminative trackers in visual tracking. Most existing works usually focus on improving the robustness of the target appearance model. However, they are prone to suffer from model drifting due to the inappropriate model updates during the tracking-by-detection. In this paper, we propose a novel update-pacing framework to suppress the occurrence of model drifting in visual tracking. Specifically, the proposed framework first initializes an ensemble of trackers, each of which updates the model in a different update interval. Once the forward tracking trajectory of each tracker is determined, the backward trajectory will also be generated by the current model to measure the difference with the forward one, and the tracker with the smallest deviation score will be selected as the most robust tracker for the remaining tracking. By performing such self-examination on trajectory pairs, the framework can effectively preserve the temporal context consistency of sequential frames to avoid learning corrupted information. To further improve the performance of the proposed method, a multi-feature extension framework is also proposed to incorporate multiple features into the ensemble of the trackers. The extensive experimental results obtained on large-scale object tracking benchmarks demonstrate that the proposed framework significantly increases the accuracy and robustness of the underlying base trackers, such as DSST, Struck, KCF, and CT, and achieves superior performance compared with the state-of-the-art methods without using deep models.**

***Index Terms*— Object tracking, model drifting, trajectory selection, multi-feature, temporal context.**

I. Introduction

VISUAL tracking is the task of learning an arbitrary target, which is generally an unknown object located within a rectangular bounding box in the first frame, and then predicting the location of the selected target in the subsequent frames. Although numerous object tracking algorithms have been proposed over the past decade [1]–[4], developing a robust and accurate visual tracker remains a challenging problem because of variations in appearance and shape, illumination changes, occlusion, background clutter and abrupt motion, to name a few. Based on the difference of the underlying state inference for each video frame, these tracking algorithms can be roughly classified as filtering-based visual object tracking approaches and tracking-by-detection methods [5].

The filtering-based visual object tracking approaches extensively use the probabilistic state-space model, which can be separated into the measurement model and the system model [6]–[8]. The system model describes the transition of the state of the target object from one frame to the next frame, whereas the measurement model provides the approximated position of the target object given the current information of the state. Two representative types of filtering-based trackers, i.e., Kalman filter [9] and particle filter [10], have achieved positive tracking performance. However, such approaches are prone to mismatch between the underlying model description and reality.

Tracking-by-detection is another approach that attempts to directly approximate the posterior distribution of the target object state using a discriminative model based on the information from the current frame and the tracking results of the last frames [11]–[14]. In contrast to the filtering-based approaches, the tracking-by-detection methods do not have the drawback of mismatch between the underlying model description and reality. However, such approaches heavily rely on the correctness of the target appearance model; thus, these approaches suffer from model drifting during inappropriate model updates. Model drifting originates from the target appearance model being updated using training samples that contain undesired background information. As more samples are used for the update, the accumulated error in the model will cause the model to appear more similar to the background rather than the foreground, thus causing the object tracker to drift to the background.

To address the above issue, various methods have been developed in recent works. For example, a number of approaches focus on making improvements to the model update [15]–[17]. A few strategies consist of maintaining an additional detector and correcting the tracker's prediction when an error occurs [18]–[20]. These methods assume that the tracker is unaware of drifting occurrence; thus, a different detection has to be deployed to correct the drifted

Manuscript received October 12, 2017; revised October 29, 2018; accepted February 22, 2019. Date of publication March 5, 2019; date of current version April 3, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61702196 and in part by the Science and Technology Planning Project of Guangdong Province, China, under Grant 2017A020208041. This paper was recommended by Associate Editor W.-C. Siu. *(Corresponding author: Xuhong Tian.)*

Y. Gao and X. Tian are with the College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China (e-mail: tianxuhong@scau.edu.cn).

Z. Hu, H. W. F. Yeung, and Y. Y. Chung are with the School of Computer Science, The University of Sydney, Sydney, NSW 2006, Australia.

L. Lin is with School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China.

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

Digital Object Identifier 10.1109/TCSVT.2019.2902883

bounding box in the case of tracking failure. This tactic has been proven to provide a significant improvement in the tracker's robustness. However, it suffers from the drawback of needing to design a different algorithm and the increase in computation for maintaining such a detector. Moreover, some works employ sophisticated components, i.e., correlation filters, convolutional neural networks, global and local models, to provide more information to trackers at different occasions [21]–[24]. Furthermore, quite a few trackers utilize ensemble post-processing strategies to improve the overall tracking by selecting the best by the trackers' bounding box outputs [13], [25]–[29]. However, the ensemble strategy cannot improve the tracking performance if the trackers in the ensemble have a low deviation in the tracking output. In contrast to the aforementioned methods, we introduce a novel universal update-pacing tracking framework to alleviate the model drifting problem via integrating both temporal context and multi-feature.

In this paper, a novel update-pacing framework based on the ensemble post-processing strategy is proposed to mitigate the problem of model drifting that adversely affects discriminative visual trackers during model updates. The framework cooperates with the existing discriminative trackers to provide guided updates in proper occasions, utilizing temporal context and alleviates model corruption resulting from false updates. To this end, an ensemble of trackers, which are initialized from a base tracker, is employed in the proposed framework. These trackers are adaptively updated with different paces, and a set of forward and backward trajectories are generated for each of the trackers. Then, the best tracker is selected based on the robustness score computed using the forward and backward trajectory pairs. By performing such self-examination on trajectory pairs, the proposed framework can effectively leverage the temporal context of sequential frames to avoid learning corrupted information. Since the framework leverages the tracking trajectories only, it is universal to cooperate with the existing trackers. Hence, we go further to incorporate multiple features, which are essentially multiple trackers and complementary to each other, into the ensemble of trackers for achieving higher accuracy and robustness. The experimental results obtained on the CVPR13 [30], OTB50 [1] and OTB100 [1] visual tracking datasets demonstrate that the proposed framework enhances the trackers, which obtains comparable results with not only each of the base trackers but also the state-of-the-art trackers.

The main contributions of this work are three-fold. i) We propose a simple but universal update-pacing tracking framework to mitigate the model drifting problem, which effectively exploits the temporal context of sequential frames with an ensemble of trackers to avoid learning corrupted information. ii) A multi-feature extension of the framework is further developed, which allows our approach to leverage multiple complementary features in the ensemble of trackers to further improve performance. iii) The proposed framework can be implemented in most of the trackers that can be decomposed into tracking and updating components, regardless of the trackers' nature.

The remainder of this paper is organized as follows. Section 2 presents a summary of the works related to our research. In Section 3, we provide a detailed description of the proposed method. The experiment setup and detailed qualitative and quantitative results are discussed in Section 4. Conclusions are drawn in Section 5.

II. Related Work

A considerable amount of research has addressed the challenge of visual tracking. However, we present only a few methods that are closely related to this work, namely, discriminative trackers, correlation filter trackers, and ensemble post-processing in the following.

A. Discriminative Trackers

Discriminative approaches consist of training a classifier online or offline and predicting whether image patches are the target, separating them from the background, while the image patches come from a different part of a frame, generally surrounding the location of the target in the previous frame. For example, Hare *et al.* [31] proposed a structured SVM classifier to mitigate the effect of mis-labeling samples. In [32], multiple instance learning was used to avoid the error-prone, hard-labeling process. Another discriminative method proposed by Kalal *et al.* [33] employed a set of structural constraints to guide the sampling process of a boosting classifier. In [34], a discriminative reverse sparse representation model with weighted multitask learning was designed for tracking. A recent study formulated the tracking process as a ranking problem [35] by using the PageRank algorithm, which is a well-known webpage ranking algorithm by Google. Moreover, others attempted to use a hash algorithm with locality sensitive histograms [36] and complementary learner [37] for visual tracking. In contrast to the existing discriminative trackers, we develop an ensemble of trackers with paced updates and a trajectory selection strategy to reduce the risk of model drifting in this work.

B. Correlation Filter Trackers

Correlation filters have recently become increasingly popular due to the development of many highly accurate trackers [24], [38]–[45]. The interest in the correlation filter originated from the MOSSE tracker [40], which is a high-speed tracker that is robust to variations in lighting, scale, pose, and non-rigid deformation. Subsequent research extended the correlation filter from a single-channel to multi-channel [39] and proposed learning the invariance-discriminative power spectra of various features using a multi-kernel correlation filter [41]. In [42], three sparsity-related loss functions were designed to further promote the robustness of the correlation filter learning. Simultaneously, the MOCA tracker utilized MC-HOG features and a saliency proposal to mitigate the problem of model drifting [43]. The Staple tracker combined the correlation filter-based approach using the HOG features with the traditional ridge regression framework and complementary cues to achieve both fast and accurate tracking

performance [24]. Later, an adaptively weighted correlation filter method was developed that used more reliable information during the model update to improve the tracking performance [44]. Moreover, Liu *et al.* [45] proposed a structural correlation filter method that introduced a part-based tracking strategy into the correlation filter tracker to handle partial occlusions, preserve the object structure and to capture outlier parts. In the recently proposed trackers, a novel formulation for training a continuous convolution filter was introduced to enable the efficient integration of multi-resolution deep feature maps [38], [46] proposed the LGCF tracker with local-global correlation filter and the tracker ECO [47] using a combination of factorized convolution operator, compact generative model and effective update strategy achieved the state-of-the-art performance on a publicly available benchmark dataset. Recent works show a tendency to shift from correlation filter-based trackers towards convolutional/recurrent neural network based trackers [48], [49] or a combination of both [46], [47], [50]–[52]. In this work, we extend the correlation filter-based tracker using a novel generic update-pacing framework.

C. Ensemble Post-Processing

The ensemble post-processing strategy is the utilization of multiple trackers to provide a more stable tracking outcome [25], [53]. This strategy implemented based on the idea that the performance of a single tracker can be very volatile, and therefore, tracking stability can increase by determining the tracking output based on the results of multiple trackers. Many trackers that employ the ensemble post-processing framework are related to our work. For example, in [28], Santner *et al.* developed a model update strategy for discriminative trackers. This approach combined three trackers, a simple template model, an optical flow-based mean shift tracker and an online random forest in a cascade, to obtain a more stable update result. Later, Bailer *et al.* [54] used a fusion approach with the merit that it only required the frame-based tracking results in the form of the target object's bounding box as input. The method was based on the concept of attention, i.e., the result that maximized the attraction of all the trackers was chosen to be the final tracking result. Moreover, [55] aimed to aggregate the results of the trackers using a crowdsourcing setting. It formulated the tracking problem as the inference of an unknown target trajectory jointly with a hidden reliability measure for each object tracker using a factorial hidden Markov model. Zhang *et al.* [13] used entropy minimization to selectively update the target model and utilized tracker restoration to correct model drifting. Recently, Lee *et al.* [27] proposed a multi-hypothesis trajectory analysis (MTA) approach to use an ensemble of the Struck trackers with different features to address the model drifting problem. The MTA framework used geometric similarity, the cyclic weight, and the appearance similarity from the forward and backward trajectories to decide the robustness of a tracker. This method achieved a positive result when tested on a publicly available benchmark.

The work proposed in this paper is similar to the MTA framework in the process of generating the forward and backward trajectories and the computation of the robustness score for determining the best tracker. However, the proposed framework makes an innovative change in the initiation of trackers based on a predefined interval with paced updates. The extension of the proposed framework to multi-feature outperforms the MTA framework by a significant margin. Moreover, the framework is an extension of the previous work (Multiple Trajectories of Single Tracker, MTS) by Hu *et al.* [26]. In addition to [26], further experiments for MTS are presented on the impact of tracker number and interval. Moreover, more base trackers, i.e. DSST [56] and CT [57], are integrated into MTS to demonstrate its advantage. Furthermore, we propose a novel Multiple Trajectories of Multiple Trackers (MTM) algorithm to exploit different types of base trackers while utilizing temporal context simultaneously.

III. The Multi-Trajectory Update-Pacing Tracking Framework

In this section, we present a detailed description of our update-pacing tracking framework that takes advantage of paced update and trajectory selection to learn temporal context to alleviate model drifting.

A. Overview of the Framework

The proposed framework is based on the observation that the degradation of the target model occurs randomly along the video as the target object changes its appearance, which could be either temporary or permanent. In the former situation, aggressive model updates are undesirable since such adaptiveness to temporary changes could cause the tracker to drift away from the true target if excessive background noise appears in the bounding box or the target resumes its former appearance in the following frames. However, in the latter situation, model updates are encouraged because a conservative update strategy cannot promptly reflect the appearance changes on the true target and will also lead to drifting to the background.

A simple solution to this problem is to guide the model update process by suppressing updates upon the detection of a temporary appearance change and encouraging updates upon the detection of a permanent change. However, such an approach generally requires defining threshold values, while it is difficult for a tracker to define such values automatically. To address this issue, we develop a novel framework that uses paced update and trajectory comparison rather than a threshold to guide the tracker to avoid updating in the undesired period with negative information, while updating in the appropriate occasion.

1) Paced Update: Prior to the start of the tracking process, an ensemble $\boldsymbol{E}$ of n trackers is first initialized,

$$\boldsymbol{E} = \{\Gamma_1, \Gamma_2, \Gamma_3, ..., \Gamma_n\} \tag{1}$$

where Γ_i denotes the i-th tracker, which is a copy of the base tracker Γ_{base}. Although each of the n trackers begins as an identical copy of the base tracker Γ_{base}, they will run and behave independently in the tracking process.

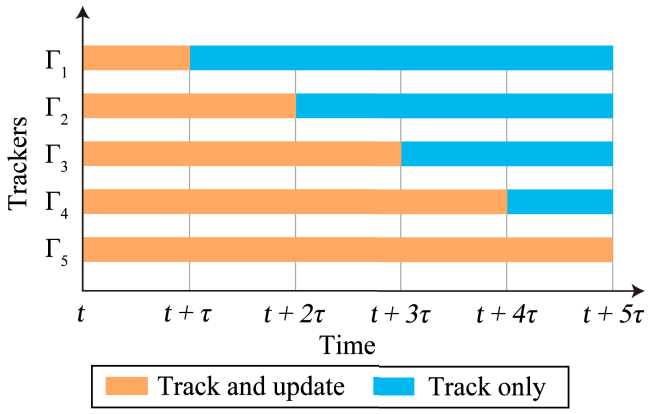


Fig. 1. Paced update: n is set to 5 in this figure. The resulting number of trackers is 5 and the total length of the sequence is 5τ. τ denotes the length of each interval.

After initialization, all the trackers begin to track forward from the first frame and have different paces in their updates. The video sequence is separated into sub-sequences of pre-defined length τ, which is also the length of each tracking interval. Assume that the tracking process starts at frame t; then, at frame $t+\tau$, the ensemble completes tracking and updates in the first interval $[t, t+\tau]$. In the second interval $[t+\tau, t+2\tau]$, all the trackers will continue tracking and be updated except Γ_1. In other words, Γ_1 will continue tracking but not be updated in this interval. In a similar manner, only the trackers $\Gamma_3, \Gamma_4, ..., \Gamma_n$ will be updated in interval $[t+2\tau, t+3\tau]$, and only the trackers $\Gamma_4, \Gamma_5, ..., \Gamma_n$ will be updated in interval $[t+3\tau, t+4\tau]$. Briefly, after every interval τ, one of the trackers will stop updating permanently. This process continues until reaching the last tracker Γ_n in the ensemble. The last tracker Γ_n will always update throughout the entire interval $[t, t+n\tau]$.

Fig. 1 presents a graphical description of this update process. Ensemble $\boldsymbol{E}$ can be considered to cover all the possibilities of updating during these intervals. The trajectories are yielded by $\boldsymbol{\Gamma_i}$, from the frame t to $t+n\tau$ in the current interval, rewritten as from frame t_1 to t_2, which is denoted by

$$\overrightarrow{\boldsymbol{X}^{\boldsymbol{i}}_{\boldsymbol{t_1:t_2}}} = \{\overrightarrow{x^i_{t_1}}, \overrightarrow{x^i_{t_1+1}}, \overrightarrow{x^i_{t_1+2}}, ..., \overrightarrow{x^i_{t_2}}\} \tag{2}$$

where $\overrightarrow{x^i_t}$ is prediction of Γ_i at frame t.

After the ensemble of trackers has arrived at frame t_2, the trackers in $\boldsymbol{E}$ will track backward in $[t_2, t_1]$ starting with the final predicted location at t_2 as the initial ground-truth bounding box. From frame t_2 through frame t_1, the backward trajectories are calculated for each tracker, denoted as

$$\overleftarrow{\boldsymbol{X}^{\boldsymbol{i}}_{\boldsymbol{t_1:t_2}}} = \{\overleftarrow{x^i_{t_2}}, \overleftarrow{x^i_{t_2-1}}, \overleftarrow{x^i_{t_2-2}}, ..., \overleftarrow{x^i_{t_1}}\} \tag{3}$$

where $\overleftarrow{x^i_t}$ is the bounding box predicted backward by Γ_i at frame t and $\overleftarrow{x^i_{t_2}} = \overrightarrow{x^i_{t_1}}$. Updating is enabled in the entire process of backward tracking to allow the trackers to behave like ordinary trackers initialized with different bounding boxes. If a tracker correctly locates the target when tracking forward, then the produced forward and backward trajectories are supposed to be equal, i.e., $\overrightarrow{x_t} = \overleftarrow{x_t}$. A deviation from equality signals inconsistency in the tracking trajectories offers information on the robustness of the trackers.

2) Trajectory Selection: A total of n pairs of forward and backward trajectories from the interval $[t_1, t_2]$ are obtained for each tracker Γ_i in the ensemble $\boldsymbol{E}$. Trajectory analysis on the trajectory pairs is the key to quantify the performance of each tracker. To this end, the method in [27] is employed as the criterion to measure the robustness of each tracker.

The first step of the analysis is checking cyclicity. As shown in Fig. 2, both the Trajectory Pair 1 and Trajectory Pair 2 display cyclicity in Fig. 2(a) and Fig. 2(b) correspondingly, as indicated by the success of the tracker in recovering the target during backward tracking, reaching the initial location of the forward trajectory to form a cycle. An example of acyclicity is given by Trajectory Pair 3 in Fig. 2(c), where the tracker is unable to reach the initial location through backward tracking. Trackers with non-cyclic trajectories indicate tracking failure and are immediately discarded, whereas those producing cyclic trajectories are accepted for further examination.

The two accepted trajectory pairs are measured by the distances between their forward and backward trajectories. A larger gap between the forward and backward trajectories indicates higher inconsistency of the corresponding tracker, which could be attributed to a drastic but temporary change in target appearance. As shown in Fig. 2(a), the backward trajectory matches the forward trajectory more accurately due to the significantly smaller distance between $\overrightarrow{x_t}$ and $\overleftarrow{x_t}$, compared to the distance between the forward trajectory and backward trajectory in Fig. 2(b), suggesting that Trajectory Pair 1 is more reliable.

Geometric and appearance similarities between forward and backward trajectories are taken into account for a more accurate comparison. At frame t, they are defined as

$$\zeta_t = exp\left(-\frac{\left\|\overrightarrow{x_t} - \overleftarrow{x_t}\right\|^2}{\sigma_1^2}\right) \tag{4}$$

$$\phi_t = exp\left(-\frac{\left\|K \cdot (P(\overleftarrow{x_t}) - Q))\right\|^2}{4\omega h \sigma_2^2}\right) \tag{5}$$

where K is a Gaussian weight mask, "$\cdot$" is the pixel-wise weight multiplication, $P(\boldsymbol{x})$ is the image patch of the bounding box $\boldsymbol{x}$, Q is the compared image patch given by the ground-truth object image in the first frame, and ω and h are the width and height of the bounding box respectively.

Finally, the robustness score $\boldsymbol{\psi}$ can be obtained by combining (4) and (5), which has been defined in (6)

$$\boldsymbol{\psi}_{t_1:t_2} = \chi \sum_{t=t_1}^{t_2} \zeta_t \phi_t \tag{6}$$

where χ is the trajectory weight. Cyclic trajectories will be set to a large trajectory weight, e.g., 10^6, to make their score substantially larger than their non-cyclic counterparts.

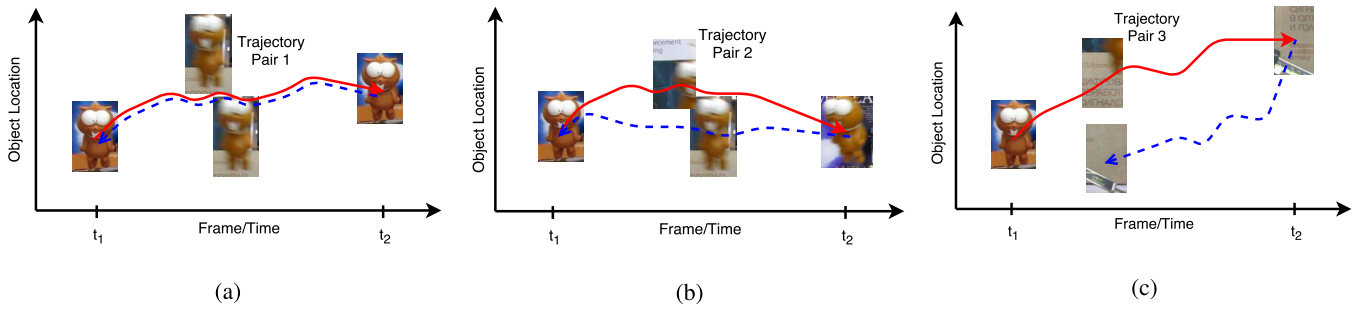


Fig. 2. Selection of trajectories: three different cases illustrating the difference between cyclic and non-cyclic trajectory pairs and the degree of matching among cyclic trajectory pairs. (a) Cyclic trajectory pair with low distance from each other. (b) Cyclic trajectory pair with high distance from each other. (c) Acyclic trajectory pair.

Algorithm 1 Multi-Trajectory Update-Pacing Framework

Input: frames $\{I_t\}$, $t \in [1, T]$, tracker number n, interval τ, base tracker Γ_{base}, initial bbox b_1.

Output: bounding box predictions $\{b_t\}$, $t \in [2, T]$.

$t \leftarrow 1$
while $t < T$ **do**
  Initialize $\{\Gamma_1, \Gamma_2, ..., \Gamma_n\}$ from Γ_{base}.
  $\boldsymbol{E} \leftarrow \{\Gamma_1, \Gamma_2, ..., \Gamma_n\}$.
  $t_1 \leftarrow t, t_2 \leftarrow min(t + n\tau, T)$
  for each $\tau_i \in E$ **do**
    τ_i tracks forward and backward in interval $[t_1, t_2]$
    through Paced Update and obtain
    the trajectory pair $\overleftarrow{\boldsymbol{X}^i_{t_1:t_2}}$ & $\overrightarrow{\boldsymbol{X}^i_{t_1:t_2}}$.
  end for
  Select the best trajectory $\{\overrightarrow{\boldsymbol{X}^*_{t_1:t_2}}\}$ through Trajectory
  Selection with pairs $\overleftarrow{\boldsymbol{X}^i_{t_1:t_2}}$ & $\overrightarrow{\boldsymbol{X}^i_{t_1:t_2}}, i \in 1, 2, ..., n$.
  Select Γ^* that generates $\{\overrightarrow{\boldsymbol{X}^*_{t_1:t_2}}\}$.
  $[b_{t_1}, b_{t_2}] \leftarrow \{\overrightarrow{\boldsymbol{X}^*_{t_1:t_2}}\}$
  $\Gamma_{base} \leftarrow \Gamma_*$
  $t \leftarrow t_2 + 1$
end while

B. Implementation

An abstract of the main flow of the proposed multi-trajectory update-pacing framework is presented in the pseudocode in Algorithm 1. The pipeline of the proposed framework is provided below.

1) Prior to tracking, the number of trackers n and the interval length τ need to be specified. The set of trackers $\boldsymbol{E} = \{\Gamma_1, \Gamma_2, \Gamma_3, ..., \Gamma_n\}$ will be initialized based on the configuration.

2) According to the configuration, an interval has a length of τ, and the total interval has a length of $n\tau$. The trackers run in parallel after initialization. When each interval finishes, a tracker within the ensemble $\boldsymbol{E}$ is selected without replacement, updating is disabled on this tracker until the total interval $n\tau$ is reached (Section III.A.1).

3) When the total interval $n\tau$ is reached, the forward trajectories are obtained for each tracker, and we track backward to compute the backward trajectories. The similarity of the two trajectories is evaluated using the trajectory selection criterion (Section III.A.2) to find the best tracking solution.

4) All the trackers will be reinitialized on the current location predicted by the best tracker.

5) Steps (2)–(4) are repeated until the end of the video.

It is a general belief that when multiple trackers are initialized in the tracking process, the tracking speed will decrease proportional to the number of trackers. However, the implementation of the proposed framework only maintains a running tracker in the ensemble while cloning and saving a snapshot at every interval. This strategy decreases the computation because only one tracker is needed to track and update, and the others just track during the entire process. Moreover, since the proposed framework evaluates the robustness of the trackers only through their trajectories, it is possible to implement it on most of the existing trackers as long as there is an independent update process, regardless of the trackers' nature. Hence, this can be regarded as a universal framework for cooperation among the trackers.

C. Extension to Multi-Feature

A visual tracking framework that used forward and backward trajectories with an ensemble of trackers was first proposed in MTA [27]. MTA initialized three Struck trackers separately with the following features: a 192-dimensional Haar-like feature, a 768-dimensional CIELAB color histogram and an illumination-invariant feature, and it achieved positive performance.

The framework proposed in this paper can also be extended to multi-feature in a formulation similar to MTA. In the extended multi-feature framework, rather than initializing only multiple trackers according to the intervals with paced updates, we also utilize three trackers with different feature representations. HOG feature with 32 bins is adopted as the basic feature. In addition to the HOG feature, a LAB colorspace feature is employed to extract the local color information of the target object. To increase the robustness, a histogram of local intensity on the brightness channel with a transformed channel applying a non-parametric local rank transformation [58] is also adopted. Moreover, the locally assembled binary feature [59] is included in the set of features to further enhance performance.

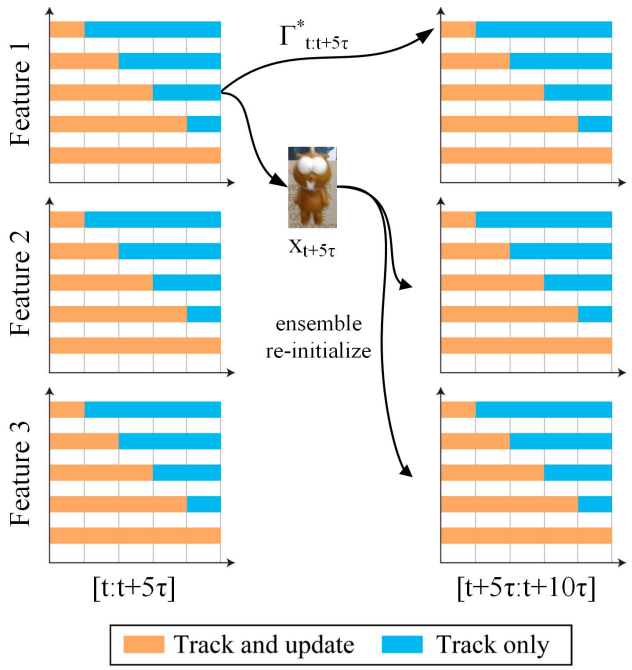


Fig. 3. Multi-feature extension: In the case where $\tau = 5$, 5 trackers are initialized for each feature prior to the start of a tracking interval. There are 15 trackers in total in the ensemble E. Trackers that have overlap less than 80% overlap with the best solution at the end of the interval will be reinitialized.

In the multi-feature framework, the hyper-parameter is no longer confined to the number of trackers n and the length of intervals τ. Rather, there is an additional hyper-parameter that dictates the features employed by different trackers. Prior to the start of tracking, an ensemble of trackers with different features and different update paces are initialized for tracking. For instance, five trackers and three features imply the initialization of fifteen trackers. Similar to that with a single feature, the forward and backward tracking trajectories of the trackers are compared using the criteria mentioned in part A for obtaining the best tracking output. A graphic description is presented in Fig. 3. The extended framework will be tested and compared with the MTA framework, and the results from the following section demonstrate that our extended framework outperforms MTA by a significant margin.

The implementation of the multi-feature framework requires some modification to the pipeline in Section III.B. First, it is necessary to further specify the different features employed by the trackers in Step 1. A set of trackers will be initialized for each feature. Second, in Step 4, if a tracker generates a solution with an overlap of more than 0.8 with the best tracking solution, then the tracker will not be reinitialized. This is in contrast to the single-feature framework in which all trackers will be initialized. This implementation method is adopted because it has been shown to increase the tracking performance.

IV. Experiments and Results

A. Dataset and Evaluation Metrics

The proposed framework with a single feature is named MTS, whereas that with multiple features is named MTM. The proposed method is implemented in C++. The source code will be made available on `https://github.com/huzexi`. The framework is evaluated on the benchmarks CVPR13 [30], OTB50 [1] and OTB100 [1], which consist of testing sequences in different challenging conditions, namely, IV(illumination variation), SV(scale variation), OCC(occlusion), DEF(deformation), MB(motion blur), FM(fast motion), IPR(in-plane rotation), OPR(out-of-plane rotation), OV(out-of-view), BC(background clutters) and LR(low resolution). The performance criteria are given by the precision rate (PR) measured as the area under the curve of the precision plot and the success rate (SR) measured as the area under the curve of the success plot. PR is an indicator of the accuracy of a tracker, and SR is an indicator of the robustness. For each image sequence, the tracked target is annotated with the bounding box in the first frame, and the ground-truth data are used to compute the precision.

B. Experiment on MTS

In this experiment, four commonly recognized outstanding trackers are incorporated into MTS, which are Struck [31], DSST [56], CT [57] and KCF [39], as selected from the existing representative trackers. The experiment will perform a one-pass evaluation (OPE) and the integrated trackers will be compared with the original trackers. For the proposed MTS framework with Struck, DSST, CT and KCF, the number of trackers n are set to 7, 5, 17, 10 and the length of interval τ are set to 10, 25, 5, 25, respectively. All of the hyperparameters of trackers remain the same in the three datasets.

The goal of this experiment is to test the performance change in the base tracker after adopting the MTS framework. The precision plots and the success plots of the tested trackers are presented in Fig. 4. Detailed reports on the experiment in different challenges are presented in Table I, Table II on CVPR13, Table III, Table IV on OTB50, Table V, Table VI on OTB100, and some tracking screenshots are presented in Fig. 5.

As shown in these tables, the tracking performance has significantly improved for most of the datasets. For CVPR13 dataset, MTS has improved the overall tracking performance by at least 1.07% and 0.17% and up to 13.11% and 12.61% for precision and success, respectively. Similar improvement can also be observed in OTB50 and OTB100 datasets. For the challenging conditions of occlusion, out-of-plane rotation and out-of-view, MTS results in a large and consistent increase for most testing algorithms.

Fig. 5(a) presents some examples illustrating how the proposed MTS framework improves the tracking accuracy and robustness of the underlying Struck tracker. The green bounding box is the result of MTS+Struck, while the red bounding box is the result of Struck. The screenshots from the benchmark videos *Liquor* and *Jogging* show that our MTS framework is able to recover from heavy occlusion on the target object, i.e., the target bottle is covered by a similar object in the video *Liquor*, and the person is occluded by the lamp post in the video *Jogging*. The explanation for this success is that the original Struck tracker is updated with

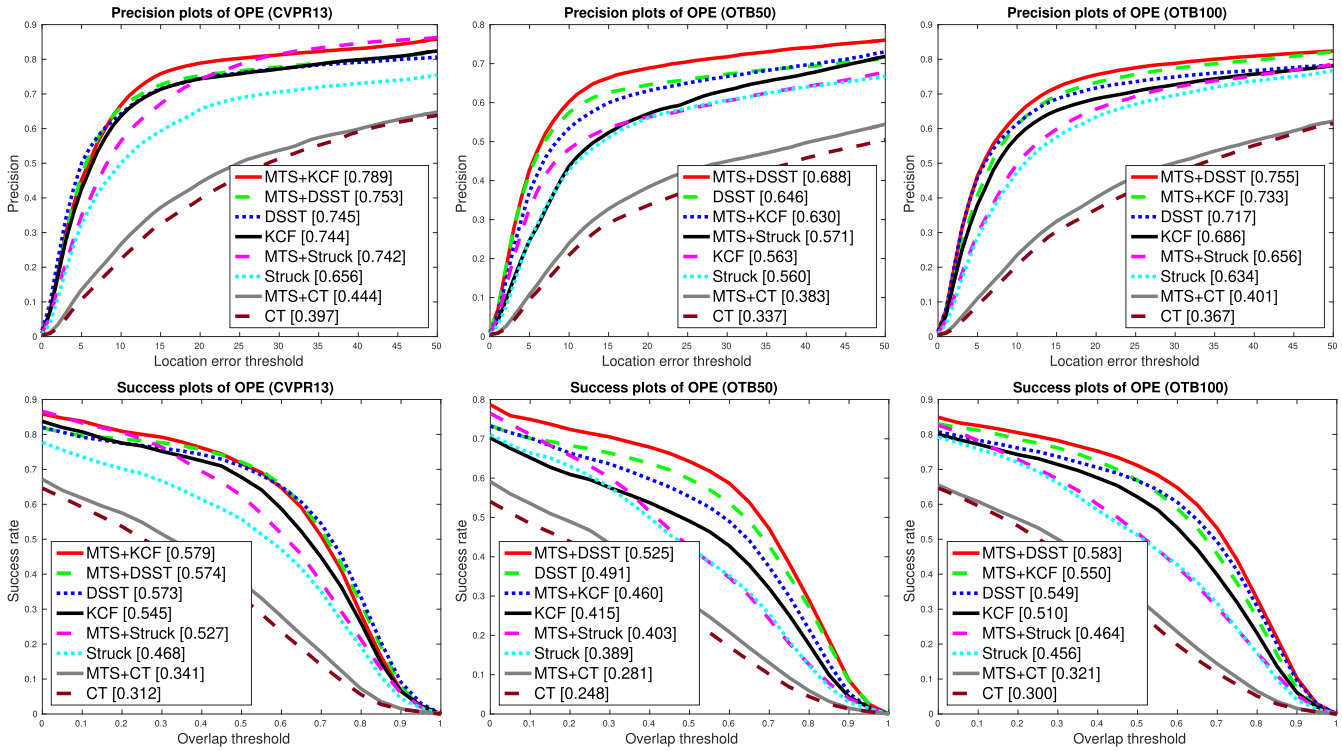


Fig. 4. Precision and success plots of our MTS framework on the CVPR13 [30], OTB50 [1] and OTB100 [1] benchmarks.

TABLE I

COMPARISON OF THE PR/SR SCORES OBTAINED WITH THE OPE METHOD UNDER THE CVPR13 BENCHMARK. NUMBERS IN PARENTHESIS IN THE FIRST COLUMN REFER TO THE NUMBER OF SEQUENCES WITH THE CORRESPONDING CHALLENGE. THE HIGHEST SCORES ARE SET BOLD IN EVERY TEST

	CT	MTS+CT	Struck	MTS+Struck	DSST	MTS+DSST	KCF	MTS+KCF
IV (25)	0.322/0.287	0.382/0.317	0.585/0.436	0.679/0.495	0.670/0.533	0.662/0.519	0.669/0.500	**0.727/0.539**
OPR (39)	0.400/0.311	0.421/0.325	0.600/0.427	0.716/0.500	0.714/0.539	0.742/**0.564**	0.726/0.531	**0.769**/0.560
SV (28)	0.387/0.276	0.439/0.306	0.639/0.416	0.699/0.461	0.723/**0.560**	0.705/0.547	0.717/0.524	**0.750**/0.553
OCC (29)	0.399/0.322	0.468/0.356	0.588/0.427	0.734/0.513	0.696/0.530	0.753/0.566	0.796/0.558	**0.831/0.580**
DEF (19)	0.503/0.396	0.562/0.437	0.560/0.419	0.693/0.509	0.696/**0.529**	0.703/0.514	0.711/0.487	**0.743**/0.503
MB (12)	0.295/0.261	0.330/0.293	0.554/0.441	**0.666/0.539**	0.533/0.452	0.552/0.447	0.607/0.458	0.665/0.501
FM (17)	0.282/0.274	0.337/0.291	0.612/0.467	0.632/**0.511**	0.547/0.454	0.564/0.457	0.612/0.457	**0.659**/0.493
IPR (31)	0.387/0.301	0.386/0.302	0.592/0.425	0.672/0.482	0.717/0.546	0.734/**0.569**	0.710/0.527	**0.739**/0.554
OV (6)	0.261/0.308	0.374/0.344	0.527/0.444	0.580/0.491	0.623/0.529	0.647/0.551	0.730/0.599	**0.801/0.630**
BC (21)	0.437/0.344	0.467/0.352	0.588/0.447	0.661/0.498	**0.738/0.567**	0.643/0.490	0.571/0.460	0.641/0.486
LR (4)	0.187/0.152	0.296/0.190	0.552/0.383	**0.557/0.453**	0.499/0.406	0.368/0.298	0.460/0.361	0.517/0.410
Overall	0.397/0.312	0.444/0.341	0.656/0.468	0.742/0.527	0.745/0.573	0.753/0.574	0.744/0.545	**0.789/0.579**

TABLE II

COMPARISON OF THE PERCENTAGE CHANGE IN PR/SR SCORES WITH OR WITHOUT MTS UNDER THE CVPR13 BENCHMARK

	MTS+CT	MTS+Struck	MTS+DSST	MTS+KCF
IV (25)	**18.63%**/10.45%	16.07%/**13.53%**	-1.19%/-2.63%	7.98%/7.24%
OPR (39)	5.25%/4.50%	**19.33%/17.10%**	3.92%/4.64%	5.59%/5.18%
SV (28)	**13.44%/10.87%**	9.39%/10.82%	-2.49%/-2.32%	4.40%/5.24%
OCC (29)	17.29%/10.56%	**24.83%/20.14%**	8.19%/6.79%	4.21%/3.79%
DEF (19)	11.73%/10.35%	**23.75%/21.48%**	1.01%/-2.84%	4.31%/3.18%
MB (12)	11.86%/12.26%	**20.22%/22.22%**	3.56%/-1.11%	8.72%/8.58%
FM (17)	**19.50%**/6.20%	3.27%/9.42%	3.11%/0.66%	7.13%/**7.30%**
IPR (31)	-0.26%/0.33%	**13.51%/13.41%**	2.37%/4.21%	3.92%/4.87%
OV (6)	**43.30%/11.69%**	10.06%/10.59%	3.85%/4.16%	8.86%/4.92%
BC (21)	6.86%/2.33%	**12.41%/11.41%**	-12.87%/-13.58%	10.92%/5.35%
LR (4)	**58.29%/25.00%**	0.91%/18.28%	-26.25%/-26.60%	11.03%/11.95%
Overall	11.84%/9.29%	**13.11%/12.61%**	1.07%/0.17%	5.70%/5.87%

false information during occlusion and can therefore no longer recover the true target in the following frames. Meanwhile, the proposed MTS framework, which utilizes multiple trackers with paced updates, is able to prevent drifting by selecting the uncontaminated tracker. A similar situation applies to the other videos, which experience temporary target lost due to

TABLE III

COMPARISON OF THE PR/SR SCORES OBTAINED WITH THE OPE METHOD UNDER THE OTB50 BENCHMARK. NUMBERS IN PARENTHESIS IN THE FIRST COLUMN REFER TO THE NUMBER OF SEQUENCES WITH THE CORRESPONDING CHALLENGE. THE HIGHEST SCORES ARE SET BOLD IN EVERY TEST

	CT	MTS+CT	Struck	MTS+Struck	DSST	MTS+DSST	KCF	MTS+KCF
IV (22)	0.318/0.236	0.362/0.281	0.470/0.345	0.600/0.426	0.676/0.521	**0.716/0.546**	0.556/0.417	0.656/0.481
OPR (32)	0.355/0.266	0.406/0.297	0.499/0.350	0.576/0.401	0.590/0.442	**0.645/0.488**	0.540/0.406	0.621/0.447
SV (38)	0.319/0.222	0.360/0.254	0.537/0.363	0.554/0.381	0.641/0.485	**0.693/0.534**	0.561/0.407	0.611/0.443
OCC (29)	0.378/0.264	0.425/0.287	0.513/0.355	0.578/0.400	0.594/0.444	0.651/**0.494**	0.593/0.421	**0.677**/0.464
DEF (23)	0.390/0.297	0.398/0.310	0.472/0.341	0.476/0.359	0.575/0.429	**0.579/0.435**	0.475/0.332	0.543/0.361
MB (19)	0.302/0.225	0.298/0.257	0.535/0.411	0.502/0.391	0.563/0.451	**0.662/0.527**	0.498/0.390	0.582/0.449
FM (25)	0.282/0.242	0.306/0.260	0.537/0.399	0.486/0.380	0.579/0.462	**0.623/0.502**	0.504/0.397	0.558/0.430
IPR (29)	0.351/0.269	0.370/0.294	0.529/0.378	0.546/0.394	0.621/0.461	**0.690/0.519**	0.540/0.411	0.589/0.445
OV (11)	0.387/0.281	0.422/0.288	0.501/0.346	0.578/0.416	0.582/0.448	**0.674/0.520**	0.490/0.365	0.655/0.466
BC (20)	0.303/0.247	0.408/0.307	0.483/0.359	0.561/0.422	0.647/0.502	**0.696/0.530**	0.477/0.384	0.583/0.447
LR (8)	0.433/0.201	0.405/0.190	0.694/0.333	0.738/0.367	0.748/0.480	0.756/**0.505**	0.768/0.443	**0.816**/0.481
Overall	0.337/0.248	0.383/0.281	0.560/0.389	0.571/0.403	0.646/0.491	**0.688/0.525**	0.563/0.415	0.630/0.460

TABLE IV

COMPARISON OF THE PERCENTAGE CHANGE IN PR/SR SCORES WITH OR WITHOUT MTS UNDER THE OTB50 BENCHMARK

	MTS+CT	MTS+Struck	MTS+DSST	MTS+KCF
IV (22)	13.84%/19.07%	**27.66%/23.48%**	5.92%/4.80%	15.24%/13.31%
OPR (32)	14.37%/11.65%	**15.43%/14.57%**	9.32%/10.41%	13.04%/9.17%
SV (38)	**12.85%/14.41%**	3.17%/4.96%	8.11%/10.10%	8.18%/8.13%
OCC (29)	12.43%/8.71%	**12.67%/12.68%**	9.60%/11.26%	12.41%/9.27%
DEF (23)	2.05%/4.38%	0.85%/5.28%	0.70%/1.40%	**12.52%/8.03%**
MB (19)	-1.32%/14.22%	-6.17%/-4.87%	**17.58%/16.85%**	14.43%/13.14%
FM (25)	8.51%/7.44%	-9.50%/-4.76%	7.60%/**8.66%**	**9.68%**/7.67%
IPR (29)	5.41%/9.29%	3.21%/4.23%	**11.11%/12.58%**	8.32%/7.64%
OV (11)	9.04%/2.49%	15.37%/20.23%	15.81%/16.07%	**25.19%/21.67%**
BC (20)	**34.65%/24.29%**	16.15%/17.55%	7.57%/5.58%	18.18%/14.09%
LR (8)	-6.47%/-5.47%	**6.34%/10.21%**	1.07%/5.21%	5.88%/7.90%
Overall	**13.65%/13.31%**	1.96%/3.60%	6.50%/6.92%	10.63%/9.78%

TABLE V

COMPARISON OF THE PR/SR SCORES OBTAINED WITH THE OPE METHOD UNDER THE OTB100 BENCHMARK. NUMBERS IN PARENTHESIS IN THE FIRST COLUMN REFER TO THE NUMBER OF SEQUENCES WITH THE CORRESPONDING CHALLENGE. THE HIGHEST SCORES ARE SET BOLD IN EVERY TEST

	CT	MTS+CT	Struck	MTS+Struck	DSST	MTS+DSST	KCF	MTS+KCF
IV (38)	0.310/0.264	0.347/0.295	0.582/0.432	0.669/0.481	0.710/0.562	**0.764/0.607**	0.656/0.497	0.717/0.551
OPR (63)	0.386/0.311	0.413/0.316	0.593/0.425	0.682/0.471	0.684/0.508	0.731/**0.549**	0.684/0.500	**0.736**/0.535
SV (64)	0.364/0.277	0.397/0.294	0.598/0.400	0.612/0.406	0.697/0.523	**0.755/0.580**	0.646/0.473	0.700/0.517
OCC (49)	0.380/0.309	0.418/0.321	0.564/0.409	0.673/0.469	0.635/0.490	0.708/**0.549**	0.669/0.486	**0.738**/0.525
DEF (44)	0.433/0.351	0.448/0.354	0.538/0.393	0.606/0.440	0.632/0.472	**0.665/0.497**	0.593/0.428	0.642/0.451
MB (29)	0.275/0.257	0.304/0.300	0.575/0.463	0.477/0.388	0.585/0.485	**0.695/0.567**	0.588/0.447	0.645/0.517
FM (39)	0.287/0.279	0.322/0.294	0.609/0.458	0.524/0.408	0.599/0.485	**0.661/0.539**	0.613/0.468	0.642/0.499
IPR (51)	0.386/0.313	0.398/0.314	0.620/0.443	0.661/0.461	0.693/0.512	**0.737/0.551**	0.671/0.500	0.710/0.527
OV (14)	0.328/0.276	0.390/0.289	0.517/0.377	0.579/0.440	0.601/0.472	**0.716/0.565**	0.580/0.441	0.711/0.525
BC (31)	0.378/0.305	0.401/0.315	0.562/0.424	0.640/0.480	0.732/0.564	**0.744/0.578**	0.605/0.477	0.670/0.517
LR (9)	0.398/0.179	0.360/0.169	0.679/0.316	0.746/0.361	0.767/0.465	0.743/0.483	0.793/0.473	**0.837/0.510**
Overall	0.367/0.300	0.401/0.321	0.634/0.456	0.656/0.464	0.717/0.549	**0.755/0.583**	0.686/0.510	0.733/0.550

fast motion, i.e., *Shaking* and *Trellis*. Some improvement can also be observed in the video *Singer*1 with significant illumination variation. Since the Struck tracker is originally poor in adapting to scale variation, the main improvement is in accuracy in terms of precision rather than robustness in terms of overlap ratio.

As shown in Fig. 5(b), in contrast to the Struck tracker, the KCF tracker not only gains accuracy under the proposed MTS framework, but also substantially improves the robustness. The green bounding box is the result of MTS+KCF, while the red bounding box is the result of KCF. In videos such as *Basketball* and *Liquor*, there is a tendency for the KCF tracker to classify the background area in the image as a part of the target. However, KCF under the MTS framework does not have such problem. The enlargement of the bounding box to the surroundings is most likely a consequence of the background information update in the target template. The problem is avoided by the proposed method since the determination for the best tracker includes the criteria of appearance similarity indicated by equation (5) in addition to geometric similarity indicated by equation (4). For similar trajectory pairs, the one with a better fitting bounding box within the interval will be selected as the best tracker. Similar to Struck, KCF+MTS outperforms KCF in situations such as heavy occlusion, i.e., *Basketball*, *Coke* and *Liquor*, and fast motion, i.e., *Freeman*1. It also performs well under poor illumination in the video *Cardark*.

(a) (b)

Fig. 5. Tracking screenshots of (a) MTS+Struck vs Struck, and (b) MTS+KCF vs KCF. The sequences are as follows: (a) liquor, jogging, shaking, singer1 and Trellis; (b) basketball, cardark, coke, freeman1 and liquor.

C. Parameter Variation

The previous experiment has illustrated how the proposed MTS framework improves the tracking accuracy and robustness of the base trackers. However, the determination of parameters such as the number of trackers n and the tracking speed is not explained. In this experiment, the MTS framework with KCF as the base tracker is tested under the OTB100 dataset on different combination of parameters, i.e., the number of trackers n and the length of the tracking intervals τ. The parameters are selected using grid search, with some additional combinations. The impact of the variation in parameters on the tracking accuracy and robustness will be discussed in this section.

Table VII presents the relationship between the number of trackers and the length of the tracking intervals with the tracking accuracy, robustness and speed measured in terms of precision, overlap and frames per second (FPS), respectively. As shown in this table, our proposed framework tends to produce better results when the tracking interval is set to be longer. A longer length of the interval can likely provide more information for evaluating the quality of the trajectory, thereby providing higher tracking performance. Moreover, it is easily observed that the number of trackers is inversely related to the tracking speed due to the increase in computation cost as more trackers are initialized and updated. The reduction in speed is not linear and can be explained by the optimization in the implementation of the multi-tracker framework.

The best combination of parameters is 10 trackers and 25 frame intervals, with the results of 0.733 in precision and 0.550 in overlap. However, such high performance can only be achieved in 13.6 FPS, which is far from real time. With a slight reduction in tracking performance, we can obtain 0.717 in precision and 0.541 in overlap with 22.98 FPS when tracker number is 4 and interval is 20, which is close to real time. Moreover, most other parameter configurations reach a precision level higher than 0.686 and overlap higher than 0.510, compared to the original KCF tracker. The results indicate that the proposed MTS framework can have positive performance guarantee.

D. Experiment on MTM

A similar experiment is conducted on the proposed MTM tracker, which is the multi-feature variation of the proposed MTS tracker. The features described in Section III. C will be used in this experiment.

TABLE VI
COMPARISON OF THE PERCENTAGE CHANGE IN PR/SR SCORES WITH OR WITHOUT MTS UNDER THE OTB100 BENCHMARK

	MTS+CT	MTS+Struck	MTS+DSST	MTS+KCF
IV (38)	11.94%/**11.74%**	**14.95%**/11.34%	7.61%/8.01%	8.51%/9.80%
OPR (63)	6.99%/1.61%	**15.01%/10.82%**	6.87%/8.07%	7.07%/6.54%
SV (64)	**9.07%**/6.14%	2.34%/1.50%	8.32%/**10.90%**	7.71%/8.51%
OCC (49)	10.00%/3.88%	**19.33%/14.67%**	11.50%/12.04%	9.35%/7.43%
DEF (44)	3.46%/0.85%	**12.64%/11.96%**	5.22%/5.30%	7.63%/5.10%
MB (29)	10.55%/16.73%	-17.04%/-16.20%	**18.80%/16.91%**	8.84%/13.54%
FM (39)	**12.20%**/5.38%	-13.96%/-10.92%	10.35%/**11.13%**	4.52%/6.21%
IPR (51)	3.11%/0.32%	**6.61%**/4.06%	6.35%/**7.62%**	5.49%/5.12%
OV (14)	18.90%/4.71%	11.99%/16.71%	**19.13%/19.70%**	18.42%/16.00%
BC (31)	6.08%/3.28%	**13.88%/13.21%**	1.64%/2.48%	9.70%/7.74%
LR (9)	-9.55%/-5.59%	**9.87%/14.24%**	-3.13%/3.87%	5.26%/7.25%
Overall	**9.26%**/7.00%	3.47%/1.75%	5.30%/6.19%	6.41%/**7.27%**

TABLE VII
TRACKING RESULTS ON OTB100 WITH DIFFERENT NUMBERS OF TRACKERS AND INTERVAL RANGES

Tracker	Interval	Precision	Overlap	FPS
2	5	0.676	0.518	**25.37**
2	10	0.688	0.531	24.87
2	15	0.699	**0.534**	24.55
2	20	0.690	0.523	24.53
2	25	**0.708**	0.538	24.49
4	5	0.684	0.529	**23.65**
4	10	0.701	0.533	23.35
4	15	0.712	0.540	23.16
4	20	**0.717**	**0.541**	22.98
4	25	0.708	0.536	23.09
6	5	0.705	0.538	**22.40**
6	10	0.711	0.537	22.10
6	15	**0.707**	**0.541**	22.11
6	20	0.699	0.534	22.13
6	25	0.705	0.533	22.06
8	5	0.703	0.533	18.57
8	10	0.705	0.538	18.45
8	15	0.669	0.517	18.63
8	20	**0.730**	**0.551**	18.58
8	25	0.707	0.535	**18.82**
10	5	0.717	0.547	13.12
10	10	0.709	0.537	13.02
10	15	0.715	0.540	13.04
10	20	0.721	0.545	13.40
10	25	**0.733**	**0.550**	**13.60**

TABLE VIII
FEATURE ANALYSIS: MTM/MTS TRACKERS AND THEIR CORRESPONDING FEATURE REPRESENTATIONS

Tracker Name	HOG	Intensity	LAB
MTM	✔	✔	✔
MTM_12	✔	✔	
MTM_13	✔		✔
MTM_23		✔	✔
MTS_1	✔		
MTS_2		✔	
MTS_3			✔

All hyperparameters remain the same in the three evaluation datasets, CVPR13 [30], OTB50 [1] and OTB100 [1], of which tracker number is 10 and interval is 20. MTM will be compared with several trackers, including the base trackers, e.g. DSST [56], KCF [39], Struck [31], and the recently proposed trackers, i.e. MEEM [13], LCT [19], KCFDP [60], Staple [24], TGPR [61] and MTA [27]. It is worth mentioning that MEEM and MTA are based on the idea of maintaining an ensemble of model snapshots, which have the same mechanism as ours. LCT, KCFDP and Staple, similar to ours, are based on the idea of exploiting an extra component to assist the correlation filter tracker. The precision plots and success plots are presented in Fig. 6, and some tracking screenshots are presented in Fig. 7.

As shown in Fig. 6, the proposed MTM algorithm separately achieves 0.852, 0.712, 0.781 in precision and 0.625, 0.522, 0.585 in overlap on three datasets, which are significantly higher than the results of most other trackers. Even though MTM is not the best performing tracker in precision on OTB100, it is within 1% margin compared to the performance of the best tracker. The superiority of our MTM method should be attributed to the highly adjustability of multiple features in the ensemble of trackers. Moreover, Fig. 7 shows that the MTM performs well on videos with illumination variation and background clutter. The result can be attributed to the adoption of the histogram of local intensity, which enhances the robustness of the tracker to illumination variation, and the LAB colorspace features for color representation, which provides a positive appearance model of the target object.

E. Ablation Analysis on Different Feature Combinations

As shown in the previous subsection, the multi-feature property of the MTM framework has significantly increased the tracking performance. This subsection presents further ablation analysis on the impact of adopting different features on tracking performance under the OTB100 dataset [1].

The aim of this experiment is to show the performance of each feature representation under different challenging conditions and to prove that the combination of multiple features can achieve better tracking performance than utilizing a single feature alone. To this end, we separate the proposed feature representations from Section III.C into groups of two and compare with the proposed MTM tracker that uses different features and the MTS tracker that uses only one of the features.

The three features are HOG, intensity and LAB. The configurations of the tested trackers are presented in Table VIII. The number of trackers n and the tracking interval τ are set to be 10 and 20 respectively. Holding these two parameters

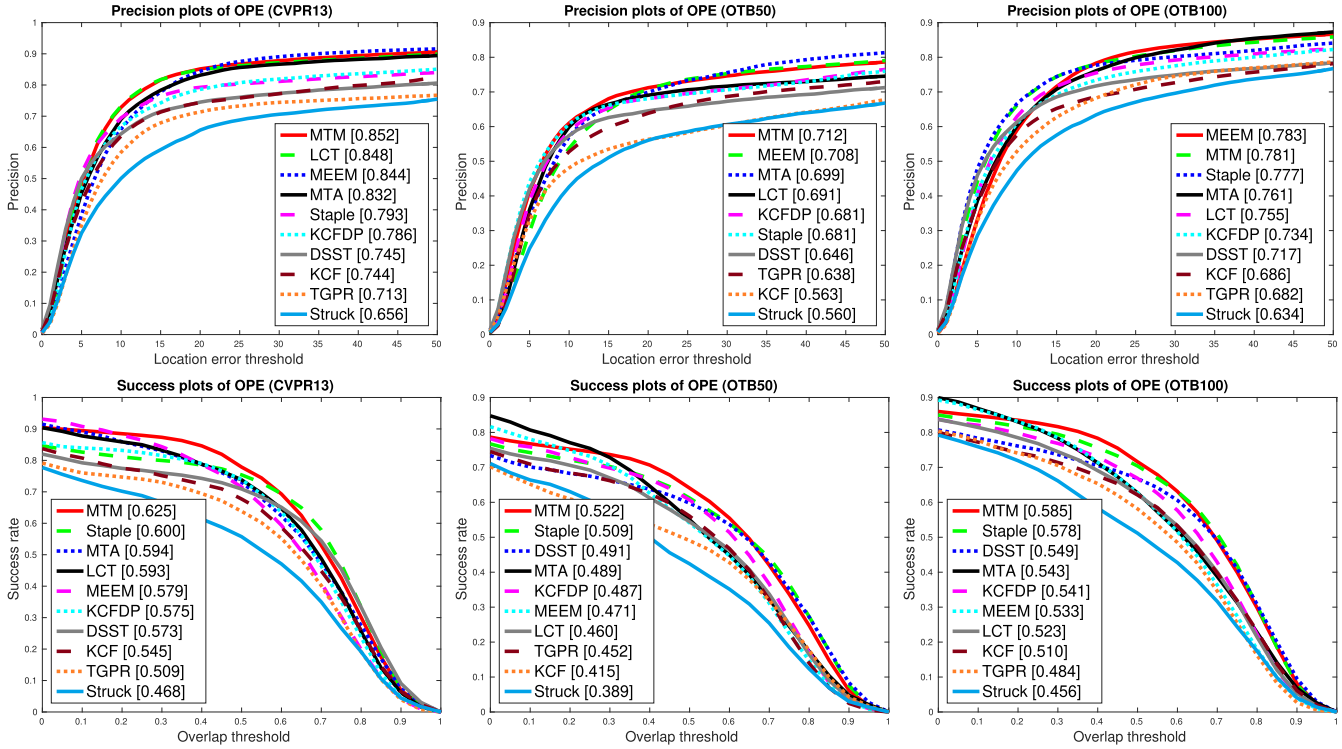


Fig. 6. Precision and success plots of our proposed MTM tracker on the CVPR13 [30], OTB50 [1] and OTB100 [1] benchmarks.

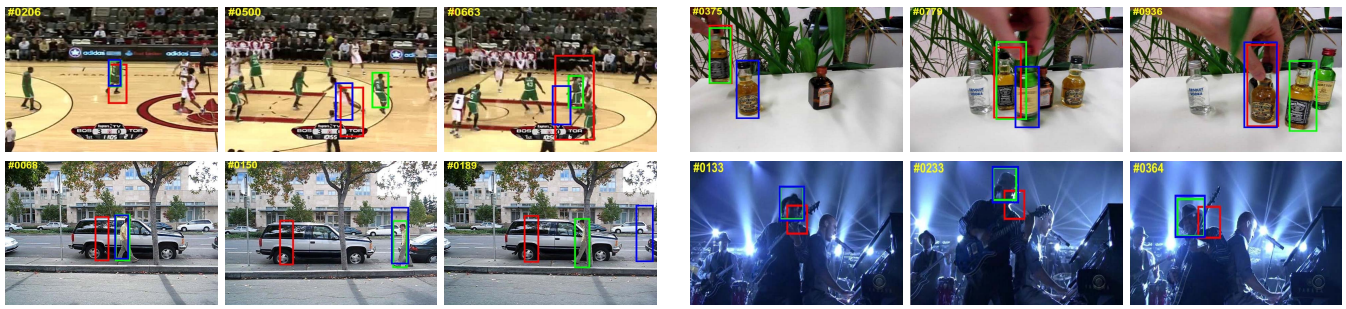


Fig. 7. Tracking screenshots of original KCF (red), MTM (green) and MTA (blue). The sequences are basketball, david3 on Column (a) and liquor, shaking on Column (b).

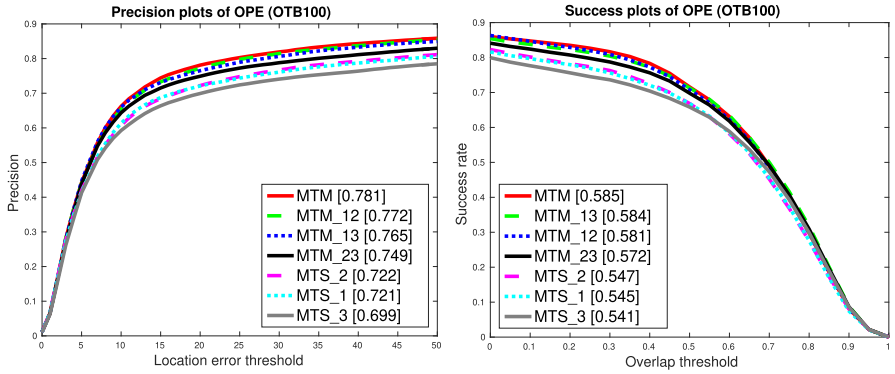


Fig. 8. Precision and success plots of our proposed MTM tracker with different feature combinations under OTB100 [1].

constant enables us to evaluate the tracker performance due to the adoption of different feature combinations. Note that when the number of features in the MTM tracker is 1, the tracker is essentially MTS. In addition, the MTS_1 tracker is the same as the MTS+KCF tracker because the original KCF tracker uses HOG as the only feature.

TABLE IX

THE PR/SR SCORES OF THE 3 MTS TRACKERS ON THE OTB100 DATASET USING OPE. THE FIRST COLUMN PRESENTS THE CHALLENGE CATEGORY WITH THE NUMBER OF VIDEOS IN THE PARENTHESIS. THE BEST PERFORMANCE SCORES IN EACH CHALLENGE ARE RECORDED IN BOLD

	MTS_1	MTS_2	MTS_3
IV (38)	**0.704**/0.530	0.682/0.522	0.670/**0.536**
OPR (63)	**0.709**/0.519	**0.709**/0.522	0.702/**0.531**
SV (64)	0.681/0.509	**0.690/0.518**	0.653/0.504
OCC (49)	**0.702/0.520**	0.696/0.518	0.661/0.506
DEF (44)	**0.625/0.454**	0.620/**0.454**	0.616/0.450
MB (29)	**0.631**/0.494	0.621/**0.497**	0.593/0.491
FM (39)	0.649/0.506	**0.659/0.514**	0.607/0.488
IPR (51)	**0.703/0.524**	0.698/0.523	0.683/**0.526**
OV (14)	0.626/0.468	0.649/0.485	**0.665/0.511**
BC (31)	**0.674/0.516**	0.660/0.503	0.639/0.505
LR (9)	0.779/0.496	0.796/0.521	**0.826/0.550**
Overall	0.721/0.545	**0.722/0.547**	0.699/0.541

TABLE X

THE PR/SR SCORES OF THE 3 MTM TRACKERS WITH 2 FEATURES AND 1 MTM TRACKER WITH 3 FEATURES ON THE OTB100 DATASET USING OPE. THE FIRST COLUMN PRESENTS THE CHALLENGE CATEGORY WITH THE NUMBER OF VIDEOS IN THE PARENTHESIS. THE BEST PERFORMANCE SCORES IN EACH CHALLENGE ARE RECORDED IN BOLD

	MTM_12	MTM_13	MTM_23	MTM
IV (38)	0.733/0.568	**0.749/0.597**	0.732/0.579	0.734/0.567
OPR (63)	0.787/0.574	0.790/**0.590**	0.772/0.578	**0.798**/0.577
SV (64)	0.737/**0.550**	0.719/0.546	0.700/0.533	**0.740**/0.547
OCC (49)	0.756/0.562	0.747/0.562	0.721/0.548	**0.780/0.571**
DEF (44)	**0.701**/0.506	0.700/**0.508**	0.676/0.491	0.696/**0.508**
MB (29)	0.661/**0.532**	0.627/0.520	0.625/0.508	**0.664**/0.529
FM (39)	0.653/0.515	0.632/0.508	0.621/0.496	**0.664/0.525**
IPR (51)	0.741/0.550	0.755/0.563	0.725/0.549	**0.774/0.566**
OV (14)	0.714/0.540	**0.774/0.581**	0.758/0.563	0.718/0.548
BC (31)	0.762/0.574	0.776/**0.597**	0.777/0.592	**0.793/0.597**
LR (9)	0.815/0.544	**0.845/0.569**	0.837/0.564	0.797/0.530
Overall	0.772/0.581	0.765/0.584	0.749/0.572	**0.781/0.585**

TABLE XI

THE PERCENTAGE CHANGE OF THE PR/SR SCORES OF THE 3 MTM TRACKERS WITH 2 FEATURES AGAINST 1 MTM TRACKER WITH 3 FEATURES ON THE OTB100 DATASET USING OPE. THE HIGHEST VALUES IN EACH CHALLENGE ARE RECORDED IN BOLD

	MTM vs MTM_12	MTM vs MTM_13	MTM vs MTM_23
IV (38)	0.14%/**-0.18%**	-2.00%/-5.03%	**0.27%**/-2.07%
OPR (63)	1.40%/**0.52%**	1.01%/-2.20%	**3.37%**/-0.17%
SV (64)	0.41%/-0.55%	2.92%/0.18%	**5.71%/2.63%**
OCC (49)	3.17%/1.60%	4.42%/1.60%	**8.18%/4.20%**
DEF (44)	-0.71%/0.40%	-0.57%/0.00%	**2.96%/3.46%**
MB (29)	0.45%/-0.56%	5.90%/1.73%	**6.24%/4.13%**
FM (39)	1.68%/1.94%	5.06%/3.35%	**6.92%/5.85%**
IPR (51)	4.45%/2.91%	2.52%/0.53%	**6.76%/3.10%**
OV (14)	**0.56%/1.48%**	-7.24%/-5.68%	-5.28%/-2.66%
BC (31)	**4.07%/4.01%**	2.19%/0.00%	2.06%/0.84%
LR (9)	**-2.21%/-2.57%**	-5.68%/-6.85%	-4.78%/-6.03%
Overall	1.17%/0.69%	2.09%/0.17%	**4.27%/2.27%**

Fig. 8 displays the precision plots and the success plots of all seven trackers described in Table VIII. As shown in this figure, all trackers that utilize two feature representations perform better than those with only one feature representation. The trackers with one feature have a precision range of 0.699 to 0.722 and a success range of 0.541 to 0.547, whereas the trackers with two features have a precision range of 0.749 to 0.772 and a success range of 0.572 to 0.584. A further increase in performance can be observed when all features are used, providing a precision of 0.781 and 0.585. It can be observed that under the proposed MTM framework, higher tracking performance can be achieved by combining different feature components.

In addition to the above comparison, Table IX and Table X display the performances of the seven trackers on different challenges. As shown in Table IX, MTS_2 performs well in scale variation and fast motion, MTS_3 performs well in handling out of view and low resolution, and MTS_1 performs well in occlusion, deformation, in-plane rotation and background clutter. Moreover, it can be observed that the gap between the MTS_3 and MTS_1 trackers is very large in motion blur and fast motion, suggesting that a further increase in performance could be achieved by utilizing both features.

Table XI shows that percentage increases in performance after adopting additional feature. It can be observed that the overall contribution of each feature is more or less similar,

i.e., 1% to 4% increase in precision and 1% to 2% increase in overlap. However, if we examine the performance in each challenge, it can be observed that each feature contributes to different areas. For example, the intensity feature contributes the most to improve tracking performance in occlusion, fast motion and motion blur. The LAB feature works best in occlusion and background clutter since color information preserves well under these challenges. The HOG feature has the best performance under occlusion, motion blur, fast motion and in-plane rotation, which is mainly due to the ability of the HOG feature to capture structural information of the target. Therefore, HOG can compensate for the poor performance of the intensity-based feature during an appearance change, while the other features can help classify the target position during a structural change.

V. Conclusion

This paper has proposed a novel universal update-pacing framework by initializing multiple trackers with different update paces to mitigate the problem of model drifting during tracking. This novel tracking framework utilizes the temporal information provided by the trajectories of the trackers during a predefined interval to select the best tracker from the ensemble. The experimental results demonstrate that it is capable of largely increasing the accuracy and robustness of the base tracker under the optimal parameter settings of tracker number and interval length. Experiments using different parameter settings confirm that this framework is able to run in real time while retaining a positive performance gain. The extension of the framework to multi-feature can further enhance the performance of the base tracker.

References

[1] 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.

[2] 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.

[3] C. Li, X. Sun, X. Wang, L. Zhang, and J. Tang, "Grayscale-thermal object tracking via multitask Laplacian sparse representation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 4, pp. 673–681, Apr. 2017.

[4] R. Tao, E. Gavves, and A. W. M. Smeulders, "Siamese instance search for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1420–1429.

[5] Q. Gan, Q. Guo, Z. Zhang, and K. Cho. (2015). "First step toward model-free, anonymous object tracking with recurrent neural networks." [Online]. Available: https://arxiv.org/abs/1511.06425

[6] 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.

[7] S. Cheng, Y. Cao, J. Sun, and G. Liu, "Visual tracking with online incremental deep learning and particle filter," *Int. J. Signal Process., Image Process. Pattern Recognit.*, vol. 8, no. 12, pp. 107–120, 2015.

[8] G. Liu, C. Z. Chen, H. W. F. Yeung, Y. Y. Chung, and W.-C. Yeh, "A new weight adjusted particle swarm optimization for real-time multiple object tracking," in *Proc. Int. Conf. Neural Inf. Process.* Kyoto, Japan: Springer, 2016, pp. 643–651.

[9] S.-K. Weng, C.-M. Kuo, and S.-K. Tu, "Video object tracking using adaptive Kalman filter," *J. Vis. Commun. Image Represent.*, vol. 17, no. 6, pp. 1190–1208, Dec. 2006.

[10] M. Z. Islam, C.-M. Oh, and C.-W. Lee, "Real time moving object tracking by particle filter," in *Proc. Int. Symp. Comput. Sci. Appl.*, Oct. 2008, pp. 347–352.

[11] S. Avidan, "Ensemble tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 2, pp. 261–271, Feb. 2007.

[12] S. Zhang, Y. Sui, S. Zhao, and L. Zhang, "Graph-regularized structured support vector machine for object tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 6, pp. 1249–1262, Jun. 2017.

[13] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: Robust tracking via multiple experts using entropy minimization," in *Proc. Eur. Conf. Comput. Vis.* Zurich, Switzerland: Springer, 2014, pp. 188–203.

[14] W. Zuo, X. Wu, L. Lin, L. Zhang, and M.-H. Yang, "Learning support correlation filters for visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 5, pp. 1158–1172, May 2019.

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

[16] H.-U. Kim, D.-Y. Lee, J.-Y. Sim, and C.-S. Kim, "SOWP: Spatially ordered and weighted patch descriptor for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3011–3019.

[17] H. Possegger, T. Mauthner, and H. Bischof, "In defense of color-based model-free tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2113–2120.

[18] T. Yang, B. Li, and M. Q.-H. Meng, "Robust object tracking with reacquisition ability using online learned detector," *IEEE Trans. Cybern.*, vol. 44, no. 11, pp. 2134–2142, Nov. 2014.

[19] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-term correlation tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5388–5396.

[20] N. L. Baisa, D. Bhowmik, and A. Wallace, "Long-term correlation tracking using multi-layer hybrid features in sparse and dense environments," *J. Vis. Commun. Image Represent.*, vol. 55, pp. 464–476, Aug. 2018.

[21] 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. (CVPR)*, Jun. 2015, pp. 749–758.

[22] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5000–5008.

[23] Y. Sui, G. Wang, Y. Tang, and L. Zhang, "Tracking completion," in *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands: Springer, 2016, pp. 194–209.

[24] 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. (CVPR)*, Jun. 2016, pp. 1401–1409.

[25] N. Wang, J. Shi, D.-Y. Yeung, and J. Jia, "Understanding and diagnosing visual tracking systems," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3101–3109.

[26] Z. Hu, Y. Gao, D. Wang, and X. Tian, "A universal update-pacing framework for visual tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*. Sep. 2016, pp. 1704–1708.

[27] 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. (CVPR)*, Jun. 2015, pp. 5088–5096.

[28] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, "PROST: Parallel robust online simple tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 723–730.

[29] J. Li, Z. Hong, and B. Zhao, "Robust visual tracking by exploiting the historical tracker snapshots," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Dec. 2015, pp. 604–612.

[30] 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.

[31] S. Hare *et al.*, "Struck: Structured output tracking with kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2096–2109, Oct. 2016.

[32] 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.

[33] 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.

[34] Y. Yang, W. Hu, W. Zhang, T. Zhang, and Y. Xie, "Discriminative reverse sparse tracking via weighted multitask learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 5, pp. 1031–1042, May 2017.

[35] C. Gong, K. Fu, A. Loza, Q. Wu, J. Liu, and J. Yang, "Pagerank tracker: From ranking to tracking," *IEEE Trans. Cybern.*, vol. 44, no. 6, pp. 882–893, Jun. 2014.

[36] S. He, R. W. H. Lau, Q. Yang, J. Wang, and M.-H. Yang, "Robust object tracking via locality sensitive histograms," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 5, pp. 1006–1017, May 2017.

[37] R. Shi, G. Wu, W. Kang, Z. Wang, and D. D. Feng, "Visual tracking utilizing robust complementary learner and adaptive refiner," *Neurocomputing*, vol. 260, pp. 367–377, Oct. 2017.

[38] H. Fan and J. Xiang, "Robust visual tracking via local-global correlation filter," in *Proc. 31St Conf. Artif. Intell.*, 2017, pp. 4025–4031.

[39] 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.

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

[41] M. Tang and J. Feng, "Multi-kernel correlation filter for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3038–3046.

[42] Y. Sui, Z. Zhang, G. Wang, Y. Tang, and L. Zhang, "Real-time visual tracking: promoting the robustness of correlation filter learning," in *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands: Springer, 2016, pp. 662–678.

[43] G. Zhu *et al.*, "Mc-hog correlation tracking with saliency proposal," in *Proc. 30th Conf. Artif. Intell.*, 2016, pp. 3690–3696.

[44] Z. Cui, S. Xiao, J. Feng, and S. Yan, "Recurrently target-attending tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1449–1458.

[45] S. Liu, T. Zhang, X. Cao, and C. Xu, "Structural correlation filter for robust visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4312–4320.

[46] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands: Springer, 2016, pp. 472–488.

[47] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6638–6646.

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

[49] H. Fan and H. Ling, "SANet: Structure-aware network for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*. Jul. 2017, pp. 2217–2224.

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

[51] Y. Song, C. Ma, L. Gong, J. Zhang, R. W. Lau, and M.-H. Yang, "CREST: Convolutional residual learning for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*. Oct. 2017, pp. 2574–2583.

[52] H. Fan and H. Ling, "Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 5487–5495.

[53] J. Kwon and K. M. Lee, "Tracking by sampling trackers," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1195–1202.

[54] C. Bailer, A. Pagani, and D. Stricker, "A superior tracking approach: Building a strong tracker through fusion," in *Proc. Eur. Conf. Comput. Vis.* Zurich, Switzerland: Springer, 2014, pp. 170–185.

[55] C. Bailer, A. Pagani, and D. Stricker, "Ensemble-based tracking: Aggregating crowdsourced structured time series data," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 107–185.

[56] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf.* Nottingham, U.K.: BMVA Press, Sep. 2014, pp. 1–5.

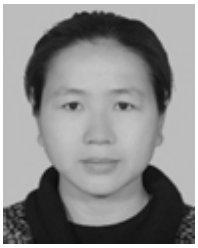
[57] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Proc. Eur. Conf. Comput. Vis.* Florence, Italy: Springer, 2012, pp. 864–877.

[58] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *Proc. Eur. Conf. Comput. Vis.* Stockholm, Sweden: Springer, 1994, pp. 151–158.

[59] S. Yan, S. Shan, X. Chen, and W. Gao, "Locally assembled binary (LAB) feature with feature-centric cascade for fast and accurate face detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–7.

[60] D. Huang, L. Luo, M. Wen, Z. Chen, and C. Zhang, "Enable scale and aspect ratio adaptability in visual tracking with detection proposals," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Sep. 2015, pp. 185.1–185.12.

[61] J. Gao, H. Ling, W. Hu, and J. Xing, "Transfer learning based visual tracking with Gaussian processes regression," in *Proc. Eur. Conf. Comput. Vis.* Zurich, Switzerland: Springer, 2014, pp. 188–203.



Yuefang Gao received the B.S. degree from Xinyang Normal University in 2000, the M.S. degree from the Hefei University of Technology in 2003, and the Ph.D. degree from the South China University of Technology in 2009, all in computer science. She is currently an Associate Professor with the College of Mathematics and Informatics, South China Agricultural University, China. Her research interests include computer vision and machine learning.

Zexi Hu received the B.S. and M.S. degrees in computer science from South China Agricultural University, China. He is currently pursuing the master's degree with the School of Computer Science, The University of Sydney, Australia. His research interests include computer vision and machine learning.

Henry Wing Fung Yeung received the M.Phil. degree in engineering and IT from The University of Sydney, where he is currently pursuing the Ph.D. degree in engineering and IT. His research interests include light-field image processing and machine learning.

Yuk Ying Chung (S'05–M'17) received the B.S. degree in computing and information systems from the University of London, U.K., in 1995, and the Ph.D. degree in computer engineering from the Queensland University of Technology, Australia, in 2000. From 1999 to 2001, she was a Lecturer with La Trobe University, Melbourne, Australia. Since 2001, she has been with the School of Computer Science, The University of Sydney, Australia. Her research interests include image and video processing, virtual reality, deep neural network, and data mining.

Xuhong Tian received the B.S. degree in applied physics from Tsinghua University in 1990, the M.S. degree in computer science from the Central China University of Science and Technology in 1999, and the Ph.D. degree in computer science from the South China University of Technology in 2007. He is currently a Professor with the College of Mathematics and Informatics, South China Agricultural University, Guangzhou, China. His research interests include computer vision and pattern recognition. He is a member of ACM.

Liang Lin was a Post-Doctoral Fellow with the University of California at Los Angeles from 2008 to 2010. He led the SenseTime R&D Team to develop cutting-edge and deliverable solutions on computer vision, data analysis and mining, and intelligent robotic systems from 2016 to 2018. He is currently a Full Professor with Sun Yat-sen University. He has authored or co-authored more than 100 papers in top-tier academic journals and conferences (e.g., 15 papers in TPAMI/IJCV and over 60 papers in CVPR/ICCV/NIPS/IJCAI). He is a Fellow of IET. He was a recipient of the Annual Best Paper Award from *Pattern Recognition* (Elsevier) in 2018, the Best Paper Diamond Award in the IEEE ICME 2017, the Best Paper Runners-Up Award in the ACM NPAR 2010, the Google Faculty Award in 2012, the Best Student Paper Award in the IEEE ICME 2014, and the Hong Kong Scholars Award in 2014. He served as the Area/Session Chair for numerous conferences, such as CVPR, ICME, ACCV, and ICMR. He has been serving as an Associate Editor for the IEEE TRANSACTIONS HUMAN-MACHINE SYSTEMS, *The Visual Computer*, and *Neurocomputing*.