Robust Visual Tracking Revisited: From Correlation Filter to Template Matching

Fanghui Liu, Chen Gong, *Member, IEEE*, Xiaolin Huang, *Member, IEEE*, Tao Zhou, *Member, IEEE*, Jie Yang, and Dacheng Tao, *Fellow, IEEE*

***Abstract*—In this paper, we propose a novel matching based tracker by investigating the relationship between template matching and the recent popular correlation filter based trackers (CFTs). Compared to the correlation operation in CFTs, a sophisticated similarity metric termed "mutual buddies similarity" (MBS) is proposed to exploit the relationship of multiple reciprocal nearest neighbors for target matching. By doing so, our tracker obtains powerful discriminative ability on distinguishing target and background as demonstrated by both empirical and theoretical analyses. Besides, instead of utilizing single template with the improper updating scheme in CFTs, we design a novel online template updating strategy named "memory filtering" (MF), which aims to select a certain amount of representative and reliable tracking results in history to construct the current stable and expressive template set. This scheme is beneficial for the proposed tracker to comprehensively "understand" the target appearance variations, "recall" some stable results. Both qualitative and quantitative evaluations on two benchmarks suggest that the proposed tracking method performs favorably against some recently developed CFTs and other competitive trackers.**

***Index Terms*—visual tracking, template matching, mutual buddies similarity, memory filtering**

I. Introduction

VISUAL tracking is the problem of continuously localizing a pre-specified object in a video sequence. Although much effort [1], [2], [3], [4] has been made, it still remains a challenging task to find a lasting solution for object tracking due to the intrinsic factors (*e.g.*, shape deformation and rotation in-plane or out-of-plane) and extrinsic factors (*e.g.*, partial occlusions and background clutter).

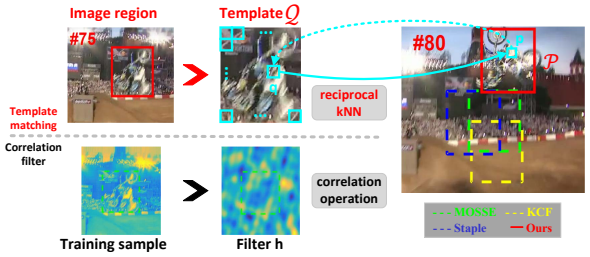


Fig. 1. Illustration of the relationship between our template matching based tracker (top panel) and CFTs (down panel). Our method considers a reciprocal k-NN scheme for similarity computation, and thus performs robustly to drastic appearance variations when compared to representative correlation filter based trackers including MOSSE [5], KCF [6], and Staple [7].

Recently, correlation filter based trackers (CFTs) have made significant achievements with high computational efficiency. The earliest work was done by Bolme *et al.* [5], in which the filter $\mathbf{h}$ is learned by minimizing the total squared error between the actual output and the desired correlation output $\mathcal{Y} = \{\mathbf{y}_i\}_{i=1}^n$ on a set of sample patches $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$. The target location can then be predicted by finding the maximum of the actual correlation response map $\mathbf{y}'$, that is computed as:

$$\mathbf{y}' = \mathcal{F}^{-1}(\hat{\mathbf{x}}' \odot \hat{\mathbf{h}}^*),$$

where $\mathcal{F}^{-1}$ is the inverse Fourier transform operation, and $\hat{\mathbf{x}}'$, $\hat{\mathbf{h}}$ are the Fourier transform of a new patch $\mathbf{x}'$ and the filter $\mathbf{h}$, respectively. The symbol $*$ means the complex conjugate, and $\odot$ denotes element-wise multiplication. In signal processing notation, the learned filter $\mathbf{h}$ is also called a template, and accordingly, the correlation operation can be regarded as a similarity measure. As a result, such CFTs share the similar framework with the conventional template matching based methods, as both of them aim to find the most similar region to the template via a computational similarity metric (*i.e.*, the correlation operation in CFTs or the reciprocal k-NN scheme in this paper) as shown in Fig. 1.

Although much progress has been made in CFTs, they often do not achieve satisfactory performance in some complex situations such as nonrigid object deformations and partial occlusions. There are three reasons as follows. First, by the correlation operation, all pixels within the candidate region $\mathbf{x}'$ and the template $\mathbf{h}$ are considered to measure their similarity. In fact, a region may contain a considerable amount of redundant pixels that are irrelevant to its semantic meaning, and

Manuscript received June 17, 2017; revised November 03, 2017; accepted February 28, 2018. Date of publication xxxx xx, xxxx; date of current version xxxxx xx, xxxx. This work was supported in part by the National Natural Science Foundation of China under Grant 61572315, Grant 6151101179, Grant 61602246, Grant 61603248, in part by 973 Plan of China under Grant 2015CB856004, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20171430, in part by the "Six Talent Peak" Project of Jiangsu Province of China under Grant DZXX-027, in part by the China Postdoctoral Science Foundation under Grant No. 2016M601597, and in part by Australian Research Council Projects under Grant FL-170100117, Grant DP-180103424, Grant DP-140102164, and Grant LP-150100671. This associate editor coordinating the review of this manuscript and approving it for publication was Nilanjan Ray. (*Corresponding author: Jie Yang.*)

F. Liu, X. Huang, T. Zhou and J. Yang are with Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: lfhsgre@outlook.com; xiaolinhuang@sjtu.edu.cn; zhou.tao@sjtu.edu.cn; jieyang@sjtu.edu.cn).

C. Gong is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: chen.gong@njust.edu.cn).

D. Tao is with the UBTECH Sydney Artificial Intelligence Centre and the School of Information Technologies, the Faculty of Engineering and Information Technologies, the University of Sydney, 6 Cleveland St, 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 xx.xxxx/TIP.xxxx.xxxxxxx

these pixels should not be taken into account for the similarity computation. Second, the learned filter, as a single and global patch, often poorly approximates the object that undergoes non-linear deformation and significant occlusions. Consequently, it easily leads to model corruption and eventual failure. Third, most CFTs usually update their models at each frame without considering whether the tracking result is accurate or not.

To sum up, the tracking performance of CFTs is limited due to the direct correlation operation, and the single template with improper updating scheme. Accordingly, we propose a novel template matching based tracker termed TM^3 (**T**emplate **M**atching via **M**utual buddies similarity and **M**emory filtering) tracker. In TM^3, the similarity based reciprocal k nearest neighbor is exploited to conduct target matching, and the scheme of memory filtering can select "representative" and "reliable" results to learn different types of templates. By doing so, our tracker performs robustly to undesirable appearance variations.

A. Related Work

Visual tracking has been intensively studied and numerous trackers have been reviewed in the surveys such as [8], [9]. Existing tracking algorithms can be roughly grouped into two categories: generative methods and discriminative methods. Generative models aim to find the most similar region among the sampled candidate regions to the given target. Such generative trackers are usually built on sparse representation [10], [11] or subspace learning [12]. The rationale of sparse representation based trackers is that the target can be represented by the atoms in an over-complete dictionary with a sparse coefficient vector. Subspace analysis utilizes PCA subspace [12], Riemannian manifold on a tangent space [13], and other linear/nonlinear subspaces to model the relationship of object appearances.

In contrast, discriminative methods formulate object tracking as a classification problem, in which a classifier is trained to distinguish the foreground (*i.e.*, the target) from the background. Structured SVM [14] and the correlation filter [6], [7] are representative tools for designing a discriminative tracker. Structured SVM treats object tracking as a structured output prediction problem that admits a consistent target representation for both learning and detection. CFTs train a filter, which encodes the target appearance, to yield strong response to a region that is similar to the target while suppressing responses to distractors. Since our TM^3 method is based on the template matching mechanism, we briefly review several representative matching based trackers as follows.

Matching based trackers can be cast into two categories: template matching based methods and keypoint matching based approaches. A template matching based tracker directly compares the image patches sampled from the current frame with the known template. The primitive tracking method [15] employs normalized cross-correlation (NCC) [16]. The advantage of NCC lies in its simplicity for implementation and thus it is used in some recent trackers such as the TLD tracker [17]. Another representative template matching based tracker is proposed by Shaul *et al*. [18], which extends the Lucas-Kanade Tracking algorithm, and combines template matching with pixel based object/background segregation to build a unified Bayesian framework. Apart from template matching, keypoint matching trackers have also gained much popularity and achieved a great success. For example, Nebehay *et al*. [19] exploit keypoint matching and optical flow to enhance the tracking performance. Hong *et al*. [20] incorporate a RANSAC-based geometric matching for long-term tracking. In [21], by establishing correspondences on several deformable parts (served as key points) in an object, a dissimilarity measure is proposed to evaluate their geometric compatibility for the clustering of correspondences, so the inlier correspondences can be separated from outliers.

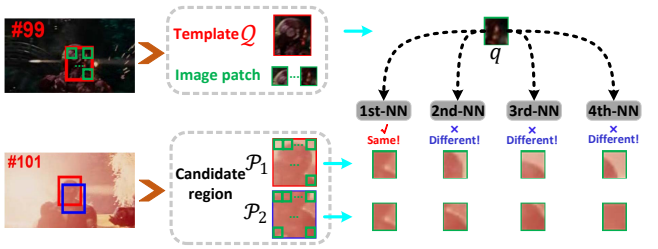


Fig. 2. Illustration of the nearest neighbors relationship of the given image patch q. In our method, a template $\mathcal{Q}$, and two candidate regions $\mathcal{P}_1$ and $\mathcal{P}_2$ are split into a set of small image patches. Specifically, we pick up a small patch q in $\mathcal{Q}$ and then investigate its nearest neighbors (*i.e.*, small image patches) in $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively. We see that two small patches selected as the 1st nearest neighbors of q in $\mathcal{P}_1$ and $\mathcal{P}_2$ are the same; while its corresponding 2nd, 3rd and 4th nearest neighbors in these two regions are very dissimilar.

B. Our Approach and Contributions

Based on the above discussion, we develop a similarity metric called "Mutual Buddies Similarity" (MBS) to evaluate the similarity between two image regions[1], based on the Best Buddies Similarity (BBS) [22] that is originally designed for general image matching problem. Herein, every image region is split into a set of non-overlapped small image patches. As shown in Fig. 1, we only consider the patches within the reciprocal nearest neighbor relationship, that is, one patch p in a candidate region $\mathcal{P}$ is the nearest neighbor of the other one q in the template $\mathcal{Q}$, and vice versa. Thereby, the similarity computation relies on a subset of these "reliable" pairs, and thus is robust to significant outliers and appearance variations. Further, to improve the discriminative ability of this metric for visual tracking, the scheme of multiple reciprocal nearest neighbors is incorporated into the proposed MBS. As shown in Fig. 2, for a certain patch q in the target template $\mathcal{Q}$, only considering the 1-reciprocal nearest neighbor of q is definitely inadequate for a tracker to distinguish two similar candidate regions $\mathcal{P}_1$ and $\mathcal{P}_2$. By exploiting these different 2nd, 3rd and 4th nearest neighbors, MBS can distinguish these candidate regions when they are extremely similar. Moreover, such discriminative ability inherited by MBS is also theoretically demonstrated in our paper.

For template updating, two types of templates Tmpl_r and Tmpl_e are designed in a comprehensive updating scheme. The template Tmpl_r is established by carefully selecting both "representative" and "reliable" tracking results during a long period.

[1]In our paper, an image region can be a template, or a candidate region such as a target region and a target proposal.

Herein, "representative" denotes that a template should well represent the target under various appearances during the previous video frames. The terminology "reliable" implies that the selected template should be accurate and stable. To this end, a memory filtering strategy is proposed in our TM^3 tracker to carefully pick up the representative and reliable tracking results during past frames, so that the accurate template Tmpl_r can be properly constructed. By doing so, the memory filtering strategy is able to "recall" some stable results and "forget" some results under abnormal situations. Different from Tmpl_r, the template Tmpl_e is frequently updated from the latest frames to timely adapt to the target's appearance changes within a short period. Hence, compared to using the single template in CFTs, the combination of Tmpl_e and Tmpl_r is beneficial for our tracker to capture the target appearance variations in different periods.

Extensive evaluations on the Object Tracking Benchmark (OTB) [8] (including OTB-50 and OTB-100) and Princeton Tracking Benchmark (PTB) [23] suggest that in most cases our method significantly improves the performance of template matching based trackers and also performs favorably against the state-of-the-art trackers.

II. The Proposed TM^3 Tracker

Our TM^3 tracker contains two main flows, *i.e.* `Flow-r` (`r` denotes random sampling) and `Flow-e` (`e` is short for EdgeBox). Each flow will undergo four steps, namely *candidate generation*, *fast candidate selection*, *similarity computation*, and *template updating* as shown in Fig. 3.

In `Flow-r` process, at the tth frame, the N_r target regions $\mathcal{R}_t = \{\mathcal{I}(\mathbf{c}_r^i, s_r^i)\}_{i=1}^{N_r}$ are randomly sampled from a Gaussian distribution, where $\mathcal{I}(\mathbf{c}_r^i, s_r^i)$[2] is the ith image region with the center coordinate $\mathbf{c}_r^i = [c_r^{ix}, c_r^{iy}]$ and scale factor s_r^i. To efficiently reduce the computational complexity, a fast k-NN selection algorithm [24] is used to form the refined candidate regions $\mathcal{R}'_t = \{\mathcal{I}(\mathbf{c}_r^i, s_r^i)\}_{i=1}^{N'_r}$ that are composed of N'_r nearest neighbors of the tracking result at the $(t-1)$th frame from $\mathcal{R}_t$. After that, MBS evaluates the similarities between these target regions and the template Tmpl_r, and then outputs $\mathcal{I}(\mathbf{c}_r^*, s_r^*)$ with the highest similarity score.

In `Flow-e` process, at the tth frame, the EdgeBox approach [25] is utilized to generate a set $\mathcal{E}_t = \{\mathcal{I}(\mathbf{c}_e^j, s_e^j)\}_{j=1}^{N_e}$ with N_e target proposals, where $\mathcal{I}(\mathbf{c}_e^j, s_e^j)$ ($\mathcal{I}_e^j$ for simplicity) is the jth image region with the center coordinate $\mathbf{c}_e^j = [c_e^{jx}, c_e^{jy}]$ and scale factor s_e^j. Subsequently, most non-target proposals are filtered out by exploiting the geometry constraint. After that, only a small amount of N'_e potential proposals $\mathcal{E}'_t = \{\mathcal{I}(\mathbf{c}_e^j, s_e^j)\}_{j=1}^{N'_e}$ are evaluated by MBS to indicate how they are similar to the template Tmpl_e. This process generates two tracking cues $\mathcal{I}_e^*$ and $\mathcal{I}_e^{dist}$, which will be further described in Section III-A. Based on the three tracking cues generated by above the two flows, the final tracking result $\mathcal{I}^*$ at the tth frame is obtained by fusing $\mathcal{I}_r^*$, $\mathcal{I}_e^*$ and $\mathcal{I}_e^{dist}$ with details introduced in Section III-B.

[2]Note that the time index t is omitted and we denote $\mathcal{I}(\mathbf{c}_r^i, s_r^i)$ as $\mathcal{I}_r^i$ for simplicity.

Note that in the entire tracking process, illustrated in Fig. 3, similarity computation and template updating are critical for our tracker to achieve satisfactory performance, so we will detail them in Sections II-A and II-B, respectively.

A. Mutual Buddies Similarity

In this section, we detail the MBS for computing the similarity between two image regions, which aims to improve the discriminative ability of our tracker.

1) The definition of MBS: In our TM^3 tracker, every image region is split into a set of non-overlapped 3×3 image patches. Without loss of generality, we denote a candidate region as a set $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^N$, where $\mathbf{p}_i$ is a small patch and N is the number of such small patches. Likewise, a template (Tmpl_r or Tmpl_e) is represented by the set of $\mathcal{Q} = \{\mathbf{q}_j\}_{j=1}^M$ of size M. The objective of MBS is to reasonably evaluate the similarity between $\mathcal{P}$ and $\mathcal{Q}$, so that a faithful and discriminative similarity score can be assigned to the candidate region $\mathcal{P}$.

To design MBS, we begin with the definition of a similarity metric MBP between two patches $\{\mathbf{p}_i \in \mathcal{P}, \mathbf{q}_j \in \mathcal{Q}\}$. Assuming that $\mathbf{q}_j$ is the rth nearest neighbor of $\mathbf{p}_i$ in the set of $\mathcal{Q}$ (denoted as $\mathbf{q}_j = \text{NN}_r(\mathbf{p}_i, \mathcal{Q})$), and meanwhile $\mathbf{p}_i$ is the sth nearest neighbor of $\mathbf{q}_j$ in $\mathcal{P}$ (denoted as $\mathbf{p}_i = \text{NN}_s(\mathbf{q}_j, \mathcal{P})$), then the similarity MBP of two patches $\mathbf{p}_i$ and $\mathbf{q}_j$ is:

$$\text{MBP}(\mathbf{p}_i,\mathbf{q}_j) = \mathrm{e}^{-\frac{rs}{\sigma_1}}, \text{if } \mathbf{q}_j = \text{NN}_r(\mathbf{p}_i, \mathcal{Q}) \wedge \mathbf{p}_i = \text{NN}_s(\mathbf{q}_j, \mathcal{P}), \tag{1}$$

where σ_1 is a tuning parameter. In our experiment, we empirically set it to 0.5. Such similarity metric MBP evaluates the closeness level between two patches by the scheme of multiple reciprocal nearest neighbors. Therefore, MBS between $\mathcal{P}$ and $\mathcal{Q}$ is defined as[3]:

$$\text{MBS}(\mathcal{P}, \mathcal{Q}) = \frac{1}{\min\{M, N\}} \cdot \sum_{i=1}^{N} \sum_{j=1}^{M} \text{MBP}(\mathbf{p}_i, \mathbf{q}_j), \tag{2}$$

One can see that MBS is the statistical average of MBP. Specifically, the similarity metric BBP used in BBS [22] is defined as:

$$\text{BBP}(\mathbf{p}_i, \mathbf{q}_j) = \begin{cases} 1 & \mathbf{q}_j = \text{NN}_1(\mathbf{p}_i, \mathcal{Q}) \wedge \mathbf{p}_i = \text{NN}_1(\mathbf{q}_j, \mathcal{P}); \\ 0 & \text{otherwise.} \end{cases}$$

Herein, the metric BBP can be viewed as a special case of the proposed similarity MBP in Eq. (1) when r and s are set to 1.

As aforementioned, BBS shows less discriminative ability than MBS for object tracking, and next we will theoretically explain the reason. Note that the factor $\frac{1}{\min\{M,N\}}$ defined in Eq. (2) does not have influence on the theoretical result, so we investigate the relationship between BBS and MBS without any factor to verify the effectiveness of such multiple nearest neighbor scheme. To this end, we introduce first-order statistic (mean value) and second-order statistic (variance) of two such metrics to analyse their respective discriminative (or "scatter") ability. We begin with the following statistical definition:

Definition 1. *Suppose two image patches* $\mathbf{p}_i$ *and* $\mathbf{q}_j$ *are randomly sampled from two given distributions* $\Pr\{P\}$ *and*

[3]In the experimental setting, the set sizes of $\mathcal{P}$ and $\mathcal{Q}$ have been set to the same value.

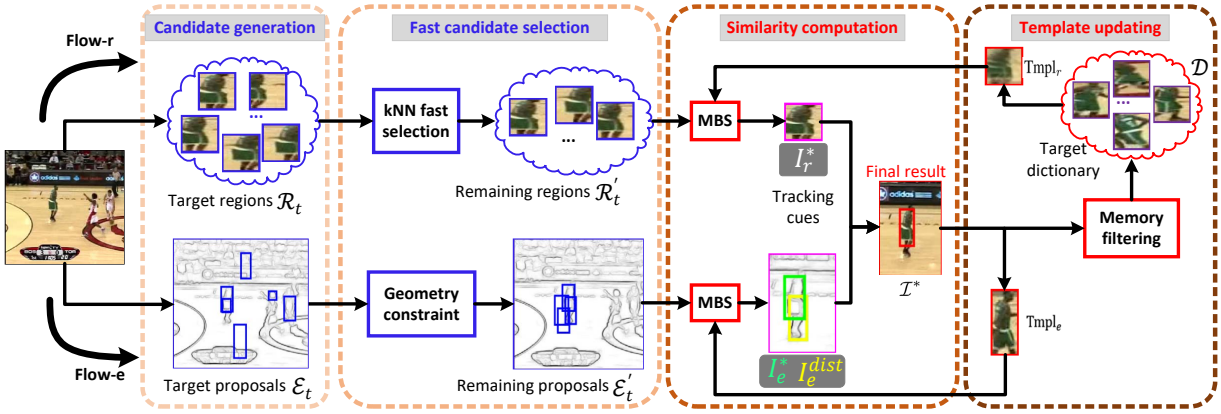


Fig. 3. The diagram of the proposed TM^3 tracker contains four main steps, namely *candidate generation*, *fast candidate selection*, *similarity computation*, and *template updating*. Numerous potential candidate regions $\mathcal{R}_t \bigcup \mathcal{E}_t$ are produced in *candidate generation* step, and then processed by the *fast candidate selection* step to form the refined $\mathcal{R}_t' \bigcup \mathcal{E}_t'$. In *similarity computation* step, MBS is used to evaluate the similarities between these refined potential regions and the templates Tmpl$_r$, Tmpl$_e$, respectively. Such step results in three tracking cues including $\mathcal{I}_r^* = \text{argmax}\,\text{MBS}(\mathcal{R}_t', \text{Tmpl}_r)$, $\mathcal{I}_e^* = \text{argmax}\,\text{MBS}(\mathcal{E}_t', \text{Tmpl}_e)$ with green box, and $\mathcal{I}_e^{dist} = \text{argmin}\, dist(\mathcal{E}_t', \mathcal{I}_r^*)$ with yellow box (the definition of distance metric $dist$ can be found in Section III-A). The final tracking result $\mathcal{I}^*$ is jointly decided by above three cues. Finally, two types of templates Tmpl$_r$ and Tmpl$_e$ are updated via the memory filtering strategy. Above process iterates until all the frames of a video sequence have been processed.

$\Pr\{Q\}$[4] *corresponding to the sets* $\mathcal{P}$ *and* $\mathcal{Q}$*, respectively, and* $\mathbb{E}_{MBS}(\mathcal{P},\mathcal{Q})$ *is the expectation of the similarity score between a pair of patches* $\{\mathbf{p}_i, \mathbf{q}_j\}$ *computed by MBP over all possible pairwise patches in* $\mathcal{P}$ *and* $\mathcal{Q}$*, then we have:*

$$\begin{aligned}\mathbb{E}_{MBS}(\mathcal{P},\mathcal{Q}) &= \int_P\int_Q MBP(\mathbf{p}_i,\mathbf{q}_j)\Pr\{P\}\Pr\{Q\}\mathrm{d}P\mathrm{d}Q\,,\\ \mathbb{E}_{BBS}(\mathcal{P},\mathcal{Q}) &= \int_P\int_Q BBP(\mathbf{p}_i,\mathbf{q}_j)\Pr\{P\}\Pr\{Q\}\mathrm{d}P\mathrm{d}Q\,.\end{aligned} \tag{3}$$

By this definition, the variance $\mathbb{V}_{\text{MBS}}(\mathcal{P},\mathcal{Q})$ and $\mathbb{V}_{\text{BBS}}(\mathcal{P},\mathcal{Q})$ can be easily computed. Formally, we have the following three lemmas. We begin with the simplification of $\mathbb{E}_{\text{MBS}}(\mathcal{P},\mathcal{Q})$ in Lemma 1, and next compute $\mathbb{E}_{\text{MBS}^2}(\mathcal{P},\mathcal{Q})$ and $\mathbb{E}^2_{\text{MBS}}(\mathcal{P},\mathcal{Q})$ to obtain $\mathbb{V}_{\text{MBS}}(\mathcal{P},\mathcal{Q})$ in Lemma 2. Lastly, we seek for the relationship between $\mathbb{E}_{\text{MBS}^2}(\mathcal{P},\mathcal{Q})$ and $\mathbb{E}_{\text{MBS}}(\mathcal{P},\mathcal{Q})$ in Lemma 3. The proofs of these three lemmas are put into Appendix A, B and C, respectively.

Lemma 1. *Let* $F_P(x)$*,* $F_Q(x)$ *be the cumulative distribution functions (CDFs) of* $\Pr\{P\}$ *and* $\Pr\{Q\}$*, respectively, and assuming that each patch is independent of the others, then the multivariate integral in Eq.* (3) *can be represented by Eq.* (5)*, where* $p^+ = p + d(p,q)$*,* $p^- = p - d(p,q)$*,* $q^+ = q + d(p,q)$*, and* $q^- = q - d(p,q)$*.*

Lemma 2. *Given* $\mathbb{E}_{MBS}(\mathcal{P},\mathcal{Q})$ *obtained in Lemma 1, the variance* $\mathbb{V}_{MBS}(\mathcal{P},\mathcal{Q})$ *can be computed by Eq.* (6)*.*

Lemma 3. *The relationship between* $\mathbb{E}_{MBS^2}(\mathcal{P},\mathcal{Q})$ *and* $\mathbb{E}_{MBS}(\mathcal{P},\mathcal{Q})$ *satisfies:*

$$\mathbb{E}_{MBS^2}(\mathcal{P},\mathcal{Q}) > \mathbb{E}_{MBS}(\mathcal{P},\mathcal{Q})\,. \tag{4}$$

By introducing above three auxiliary Lemmas, we formally present Theorem 1 as follows.

Theorem 1. *The relationship between* $\mathbb{V}_{MBS}(\mathcal{P},\mathcal{Q})$ *and* $\mathbb{V}_{BBS}(\mathcal{P},\mathcal{Q})$ *satisfies:*

$$\mathbb{V}_{MBS}(\mathcal{P},\mathcal{Q}) > \mathbb{V}_{BBS}(\mathcal{P},\mathcal{Q}),\ \ if\ \ \mathbb{E}_{MBS}(\mathcal{P},\mathcal{Q}) = \mathbb{E}_{BBS}(\mathcal{P},\mathcal{Q}). \tag{7}$$

Proof. We firstly obtain $\mathbb{V}_{\text{BBS}}(\mathcal{P},\mathcal{Q})$ and then prove that under the condition of $\mathbb{E}_{\text{MBS}}(\mathcal{P},\mathcal{Q}) = \mathbb{E}_{\text{BBS}}(\mathcal{P},\mathcal{Q})$, the variance of $\text{MBS}(\mathcal{P},\mathcal{Q})$ is larger than that of $\text{BBS}(\mathcal{P},\mathcal{Q})$.

Since $\left[\text{BBP}(\mathbf{p}_i,\mathbf{q}_j)\right]^2 = \text{BBP}(\mathbf{p}_i,\mathbf{q}_j)$ is derived from Eq. (II-A1), we have $\mathbb{E}_{\text{BBS}^2}(\mathcal{P},\mathcal{Q}) = \mathbb{E}_{\text{BBS}}(\mathcal{P},\mathcal{Q})$[5] by Definition 1. Based on this, under the condition of $\mathbb{E}_{\text{MBS}}(\mathcal{P},\mathcal{Q}) = \mathbb{E}_{\text{BBS}}(\mathcal{P},\mathcal{Q})$, we have:

$$\begin{aligned}\mathbb{V}_{\text{MBS}} - \mathbb{V}_{\text{BBS}} &= \mathbb{E}_{\text{MBS}^2} - [\mathbb{E}_{\text{MBS}}]^2 - \mathbb{E}_{\text{BBS}^2} + [\mathbb{E}_{\text{BBS}}]^2\\ &= \mathbb{E}_{\text{MBS}^2} - \mathbb{E}_{\text{BBS}^2}\ (\text{Using}\ \mathbb{E}_{\text{MBS}} = \mathbb{E}_{\text{BBS}})\\ &= \mathbb{E}_{\text{MBS}^2} - \mathbb{E}_{\text{BBS}}\ (\text{Using}\ \mathbb{E}_{\text{BBS}^2} = \mathbb{E}_{\text{BBS}})\\ &= \mathbb{E}_{\text{MBS}^2} - \mathbb{E}_{\text{MBS}} > 0\,.\end{aligned} \tag{8}$$

Finally, we obtain $\mathbb{V}_{\text{MBS}}(\mathcal{P},\mathcal{Q}) > \mathbb{V}_{\text{BBS}}(\mathcal{P},\mathcal{Q})$ as claimed, thereby completing the proof. □

Theorem 1 theoretically demonstrates that under the condition of $\mathbb{E}_{\text{MBS}}(\mathcal{P},\mathcal{Q}) = \mathbb{E}_{\text{BBS}}(\mathcal{P},\mathcal{Q})$, the variance of $\text{MBS}(\mathcal{P},\mathcal{Q})$ is larger than that of $\text{BBS}(\mathcal{P},\mathcal{Q})$, which indicates that MBS is able to produce more disperse similarity scores over $\mathcal{P}$ and $\mathcal{Q}$ than BBS when they have the same mean value of similarity scores. Therefore, MBS equipped with the scheme of multiple reciprocal nearest neighbors is more discriminative than BBS for distinguishing numerous candidate regions when they are extremely similar, which can be illustrated in Fig. 4. It can be observed that the similarity score curve of BBS (see blue curve) within No.1∼No.57, No.58∼No.89, and No.90∼No.100 candidate regions is almost flat, which means that BBS cannot tell a difference on these ranges.

[4]Such general definition does not rely on a specific distribution of $\Pr\{P\}$ and $\Pr\{Q\}$. The details of $\mathbb{E}_{\text{BBS}}(\mathcal{P},\mathcal{Q})$ can be found in Eq. (4) in [22].

[5]Note that we denote $\mathbb{E}_{\text{BBS}}(\mathcal{P},\mathcal{Q})$ as $\mathbb{E}_{\text{BBS}}$ by dropping "$(\mathcal{P},\mathcal{Q})$" for simplicity in the following description.

$$\mathbb{E}_{\text{MBS}}(\mathcal{P},\mathcal{Q}) = 1 - \frac{1}{\sigma_1} MN \iint_{-\infty}^{\infty} \left[F_P(q^+) - F_P(q^-)\right]\left[F_Q(p^+) - F_Q(p^-)\right] f_P(p) f_Q(q) \mathrm{d}p\mathrm{d}q + \frac{1}{2\sigma_1^2} M^2 N^2 \iint_{-\infty}^{\infty} \left[F_P(q^+) - F_P(q^-)\right]^2 \left[F_Q(p^+) - F_Q(p^-)\right]^2 f_P(p) f_Q(q) \mathrm{d}p\mathrm{d}q\,. \tag{5}$$

$$\mathbb{V}_{\text{MBS}}(\mathcal{P},\mathcal{Q}) = \frac{1}{\sigma_1^2} M^2 N^2 \iint_{-\infty}^{\infty} \left[F_P(q^+) - F_P(q^-)\right]^2 \left[F_Q(p^+) - F_Q(p^-)\right]^2 f_P(p) f_Q(q) \mathrm{d}p\mathrm{d}q - \frac{1}{\sigma_1^2} M^2 N^2 \left\{ \iint_{-\infty}^{\infty} \left[F_P(q^+) - F_P(q^-)\right]\left[F_Q(p^+) - F_Q(p^-)\right] f_P(p) f_Q(q) \mathrm{d}p\mathrm{d}q \right\}^2 . \tag{6}$$

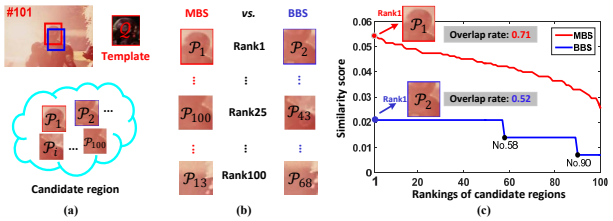


Fig. 4. Illustration of the superiority of MBS to BBS. We aim to decide which of the 100 candidate regions (from $\mathcal{P}_1$ to $\mathcal{P}_{100}$) in (a) mostly matches the target template $\mathcal{Q}$. The overlap rate between the decided target and groundtruth region is particularly observed. We see that the candidate region $\mathcal{P}_1$ with the highest similarity score is selected by MBS; while the inferior result $\mathcal{P}_2$ is picked up by BBS. Consequently, $\mathcal{P}_1$ computed by the MBS achieves higher overlap rate (71%) than $\mathcal{P}_2$ (52%) that is obtained by the BBS. Specifically, in (c), after ranking these candidates according to their similarity scores measured by MBS, we plot their rankings on the horizontal axis and the corresponding similarity scores computed by MBS (red curve) and BBS (blue curve) on the vertical axis.

In contrast, the similarity scores decided by MBS (see red curve) on all candidate regions are totally different and thus are discriminative. By such scheme, more image patches are involved in computing the similarity scores and they yield different responses to candidate regions. Accordingly, MBS can distinguish numerous candidate regions when they are extremely similar, and thus the discriminative ability of our tracker can be effectively improved.

B. Memory Filtering for Template Updating

Here we investigate the template updating scheme in our TM^3 tracker by designing two types of templates Tmpl_r and Tmpl_e. For Tmpl_e, the tracking result in the current frame is directly taken as the template Tmpl_e if its similarity score is larger than a pre-defined threshold 0.5. Note that low threshold would incur in some unreliable results and thus degrade the tracking accuracy; a large one makes the template Tmpl_e difficult to be frequently updated. In other words, Tmpl_e is frequently updated to capture the target appearance change in a short period without error accumulation. In contrast, the template Tmpl_r focuses on the tracking results in history, and it is updated via the memory filtering strategy to "recall" some stable results and "forget" some results under abnormal situations.

1) Formulation of memory filtering: Suppose the tracked target in every frame is characterized by a d-dimensional feature vector, then the tracking results in the latest N_s frames can be arranged as a data matrix $\mathbf{X} \in \mathbb{R}^{N_s \times d}$ where each row represents a tracking result. Similar to sparse dictionary selection [26], memory filtering aims to find a compact subset from N_s tracking results so that they can well represent the entire N_s results. To this end, we define a selection matrix $\mathbf{S} \in \mathbb{R}^{N_s \times N_s}$ of which s_{ij} reflects the expressive power of the ith tracking result $\mathbf{x}_i$ on the jth result $\mathbf{x}_j$, so the norm of the $\mathbf{S}$'s ith row suggests the qualification of $\mathbf{x}_i$ is to represent the whole N_s results. The selection process is fulfilled by solving:

$$\min_{\mathbf{S}} \underbrace{\frac{1}{2}\|\mathbf{X} - \mathbf{X}\mathbf{S}\|_{\text{F}}^2 + \delta \text{Tr}(\mathbf{S}^\top \mathbf{L}\mathbf{S})}_{\triangleq f(\mathbf{S})} + \underbrace{\beta \sum_{i=1}^{N_s} \frac{1}{h_i + \varepsilon} \|\mathbf{S}\|_{1,2}}_{\triangleq g(\mathbf{S})}, \tag{9}$$

where ε is a small positive constant to avoid being divided by zero, and $\|\mathbf{S}\|_{1,2} = \sum_{i=1}^{N_s} \|\mathbf{S}_{i,.}\|_2$ denotes the sum of ℓ_2 norm of all N_s rows. The second term in Eq. (9) is the smoothness graph regularization with the weighting parameter δ. Within this term, the similar tracking results will have a similar probability to be selected. Herein, the Laplacian matrix is defined by $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ is a diagonal matrix with $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$ and $\mathbf{W}$ is the weight matrix defined by the reciprocal nearest neighbors scheme, namely:

$$\mathbf{W}_{ij} = \mathrm{e}^{-\frac{rs}{\sigma_2}} \quad \text{if} \quad \mathbf{x}_i = \mathcal{N}_r(\mathbf{x}_j, \mathbf{X}) \wedge \mathbf{x}_j = \mathcal{N}_s(\mathbf{x}_i, \mathbf{X})\,,$$

where $\sigma_2 = 2$ is the kernel width. The regularization parameter β in Eq. (9) governs the trade-off between the reconstruction error $f(\mathbf{S})$ and the group sparsity $g(\mathbf{S})$ with respect to the selection matrix. Specifically, by introducing the similarity score $h_i = \text{MBS}(\mathbf{x}_i, \text{Tmpl}_r)$ to Eq. (9), the selection matrix $\mathbf{S}$ is weighted by the similarity scores to faithfully represent the "reliable" degrees of the corresponding tracking results.

2) Optimization for memory filtering: The objective function in Eq. (9) can be decomposed into a differentiable convex function $f(\mathbf{S})$ with a Lipschitz continuous gradient and a non-smooth but convex function $g(\mathbf{S})$, so the accelerated proximal gradient (APG) [27] algorithm can be used for efficiently solving this problem with the convergence rate of $\mathcal{O}(\frac{1}{T^2})$ (T

Algorithm 1: Algorithm for memory filtering strategy

Input: data matrix $\mathbf{X} \in \mathbb{R}^{N_s \times d}$ with their corresponding similarity scores $\{h_i\}_{i=1}^{N_s}$, two related regularization parameters: β, δ.
Output: the selected representative result $\mathbf{x}_i$ with the largest value in $\|\mathbf{S}_{i,\cdot}\|_2$.

Set: stopping error ε.
Obtain the Lipschitz constant p_L by Eq. (11).
Initialize $t = 0$ and $l^{(0)} = 1$, $\mathbf{S}^{(0)} = \mathbf{0}$ and two auxiliary matrices $\mathbf{U}_1^{(0)} = \mathbf{U}_2^{(0)} = \mathbf{0}$.
Repeat
  $\mathbf{Z}^{(t+1)} := \mathbf{U}_1^{(t)} - \frac{1}{p_L}\nabla f(\mathbf{U}_1^{(t)})$ by Eq. (12);
  $\mathbf{U}_2^{(t+1)} := \mathbf{S}^{(t)}$ and $\mathbf{S}_i^{(t+1)}$ is obtained by Eq. (13) for $i = 1, 2, \cdots, N_s$;
  $\tau := l^{(t)} - 1$, and $l^{(t+1)} := \frac{1+\sqrt{1+(l^{(t)})^2}}{2}$;
  $\mathbf{U}_1^{(t+1)} := \mathbf{S}^{(t+1)} + \frac{\tau(\mathbf{S}^{(t+1)} - \mathbf{U}_2^{(t+1)})}{t}$;
  $t := t + 1$;
Until $\frac{\|\mathbf{S}^{(t+1)} - \mathbf{S}^{(t)}\|_{1,2}}{\|\mathbf{S}^{i}\|_{1,2}} \leq \varepsilon$;
Output $\mathbf{x}_i$ with the largest value in $\|\mathbf{S}_{i,\cdot}\|_2$.

is the number of iterations). Therefore, we need to solve the following optimization problem:

$$\mathbf{Z}^{(t+1)} = \underset{\mathbf{S}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{S} - \mathbf{Z}^{(t)}\|_{\mathrm{F}}^2 + \frac{1}{p_L} g(\mathbf{S}), \qquad (10)$$

where the auxiliary variable $\mathbf{Z} = \mathbf{S} - \frac{1}{p_L}\nabla f(\mathbf{S})$, and p_L is the smallest feasible Lipschitz constant, which equals to:

$$p_L = \phi(\mathbf{X}^\top\mathbf{X} + \delta(\mathbf{L} + \mathbf{L}^\top)), \qquad (11)$$

where $\phi(\cdot)$ is the spectral radius of the corresponding matrix. The gradient $\nabla f(\mathbf{S})$ is obtained by:

$$\nabla f(\mathbf{S}) = -\mathbf{X}\mathbf{X}^\top + \mathbf{X}^\top\mathbf{X}\mathbf{S} + \delta(\mathbf{L} + \mathbf{L}^\top)\mathbf{S}. \qquad (12)$$

Notice that the objective function in Eq. (10) is separable regarding the rows of $\mathbf{S}$, thus we decompose Eq. (9) into an N_s of group lasso sub-problems that can be effectively solved by a soft-thresholding operator, which is:

$$\mathbf{S}_{i,\cdot} = \mathbf{Z}_{i,\cdot} \max\left\{1 - \frac{\frac{\beta}{p_L(h_i + C)}}{\|\mathbf{Z}_{i,\cdot}\|_2}, 0\right\}, i = 1, 2, \cdots, N_s. \qquad (13)$$

Finally, the algorithm for the memory filtering strategy is summarized in Algorithm 1.

3) Illustration of memory filtering for Tmpl_r: In our tracker, the tracking results of the latest ten frames ($N_s = 10$) are preserved to construct the matrix $\mathbf{X}$, and the ith ($i = 1, 2, \cdots, N_s$) tracking result $\mathbf{T}_i$ with the largest value $\|\mathbf{S}_{i,\cdot}\|_2$ is added into the target dictionary. Furthermore, to save storage space and reduce computational cost, the "First-in and First-out" procedure is employed to maintain the number of atoms N_D in the target dictionary $\mathcal{D} \in \mathbb{R}^{d \times N_D}$. That is, the latest representative tracking result is added, and meanwhile the oldest tracking result is thrown away.

Here, similar to [3], we detail how the template Tmpl_r is represented by such a carefully constructed dictionary. As shown in Fig. 5, in our method, we construct a codebook $\mathbf{U} = \mathcal{D} \cup \mathbf{I}$, where $\mathbf{I}$ represents a set of trivial templates[6]. Subsequently, we select k ($k = 5$ in our experiment) nearest

[6]Each trivial template is formulated as a vector with only one nonzero element.

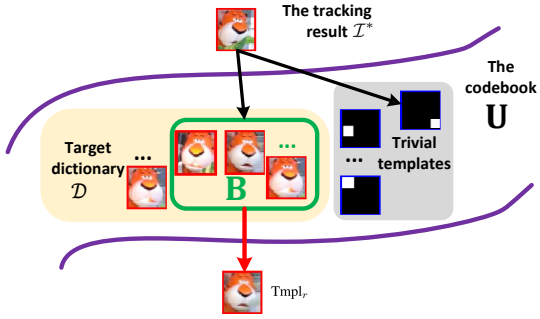


Fig. 5. Illustration of how the template Tmpl_r is represented by the target dictionary $\mathcal{D}$. The tracking results are represented by its k nearest neighbors from the target dictionary $\mathcal{D}$ and a certain amount of trivial templates. Such selected target templates render the construction of the template Tmpl_r.

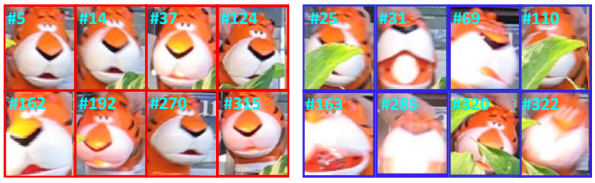


Fig. 6. Examples of some selected (red boxes) and discarded (blue boxes) historical tracking results by our memory filtering strategy. We see that the selected results represent the general appearance of the target, so they are reliable and should be "recalled". The results under abnormal situations (*e.g.* occlusion, incompleteness, and undesirable observed angle) are filtered out and "forgotten" by the memory filtering strategy.

neighbors of the tracking result $\mathcal{I}^*$ from the codebook $\mathbf{U}$, to form the dictionary $\mathbf{B}$. Finally, the template Tmpl_r is reconstructed by a linear combination of atoms in the dictionary $\mathbf{B}$. By doing so, the appearance model can effectively avoid being contaminated when the tracking result $\mathcal{I}^*$ is slightly occluded. Fig. 5 demonstrates that the template Tmpl_r is much more accurate than $\mathcal{I}^*$ because the leaf in front of the target is removed in the template Tmpl_r.

To show the effectiveness of our memory filtering strategy, a qualitative result is shown in Fig. 6. One can see that the memory filtering strategy selects some representative and reliable results (in red), which depict the target appearance in normal conditions, so they can precisely represent the target in general cases. Comparably, some tracking results under drastic appearance changes, severe occlusions and dramatic illumination variations are not incorporated to the template Tmpl_r. This is because these results just temporarily present the abnormal situations of the target, *i.e.*, far away from the target's general appearances. Note that these dramatic appearance variations in a short period can be captured by the template Tmpl_e. Therefore, the combination of two such types of templates effectively decreases the risk of tracking drifts, so that the appearance of interesting target can be comprehensively understood by our TM^3 tracker.

Finally, our TM^3 tracker is summarized in Algorithm 2.

III. Implementation Details

In this section, more implementation details of our method will be discussed.

A. Geometry Constraint

The *fast candidate selection* step aims at designing a fast algorithm to throw away some definitive non-target regions to balance between running speed and accuracy. The obtained region $\mathcal{I}_r^*$ at the tth frame in `Flow-r` process helps to remove numerous definitive non-target regions in `Flow-e` process. To this end, we introduce a distance measure $dist$ [28] between the jth target proposal $\mathcal{I}(\mathbf{c}_e^j, s_e^j)$ ($\mathcal{I}_e^j$ for simplicity) at the tth frame and $\mathcal{I}(\mathbf{c}_r^*, s_r^*)$, that is:

$$dist([\mathbf{c}_e^j, s_e^j], [\mathbf{c}_r^*, s_r^*]) = \\ \|\frac{c_r^{*x} - c_e^{jx}}{w(s_r^* + s_e^j)}, \frac{c_r^{*y} - c_e^{jy}}{h(s_r^* + s_e^j)}, \tau\frac{s_r^* - s_e^j}{s_r^* + s_e^j}\|_2, \tag{14}$$

where $\tau = 5$ is a parameter determining the influence of scale to the value of $dist$, and w, h are the width and height of the final tracking result at the $(t-1)$th frame, respectively. A small $dist$ value indicates that the corresponding image region $\mathcal{I}_e^j$ is very similar to the tracking cue $\mathcal{I}_r^*$ with a high probability. Following the definition of such distance, in our method the top N_e' target proposals with the smallest $dist$ value are retained, so that they can be used in the following step in `Flow-e` process. Therein, two tracking cues including $\mathcal{I}_e^{dist}$ with the smallest $dist$, and the $\mathcal{I}_e^*$ with the highest similarity score are picked up for the fusion step.

B. Fusion of Multiple Tracking Cues

Three tracking cues $\mathcal{I}_r^*$, $\mathcal{I}_e^*$ and $\mathcal{I}_e^{dist}$ are obtained by two main flows as shown in Fig. 3. Here we fuse the above results to the final tracking result based on a confidence level F. For example, the confidence level of $\mathcal{I}_r^*$ is defined as:

$$F(\mathcal{I}_r^*) = \text{MBS}(\mathcal{I}_r^*, \text{Tmpl}_r) + \text{MBS}(\mathcal{I}_r^*, \text{Tmpl}_e) \\ + \text{VOR}([\mathbf{c}_r^*, s_r^*], [\mathbf{c}_e^{dist}, s_e^{dist}]) + \text{VOR}([\mathbf{c}_r^*, s_r^*], [\mathbf{c}_e^*, s_e^*]), \tag{15}$$

where the Pascal VOC Overlap Ratio (VOR) [29] measures the overlap rate between the two bounding boxes A and B, namely $\text{VOR}(A, B) = \frac{area(A \cap B)}{area(A \cup B)}$. The first two terms in Eq. (15) reflect the appearance similarity degree and the last two terms consider the spatial relationship of the two cues. The confidence levels for $F(\mathcal{I}_e^*)$ and $F(\mathcal{I}_e^{dist})$ can be calculated in the similar way. Finally, the tracking cue with the highest confidence level is chosen as the final tracking result $\mathcal{I}^*$.

C. Feature Descriptors

We experimented with two appearance descriptors consisting of color feature and deep feature to represent the target regions and the templates.

Color features: For a colored video sequence, all target regions and the templates are normalized into $36 \times 36 \times 3$ in CIE Lab color space. In the `Flow-e` process, the EdgeBox approach is executed on RGB color space to generate various target proposals $\mathcal{E}_t$. In the two processes, each image region is split into 3×3 non-overlapped small patches, where each patch is represented by a 27-dimensional $(3 \times 3 \times 3)$ feature vector.

Deep features: We adopt the Fast R-CNN [30] with a pre-trained VGG16 model on ImageNet [31] and PASCAL07 [29] for our region-based feature extraction. In our method, the Fast R-CNN network takes the entire image $\mathcal{I}$ and the potential candidate proposals $\mathcal{R}_t' \cup \mathcal{E}_t'$ as input. For each candidate proposal, *the region of interest (ROI) pooling layer* in the network architecture is adopted to exact a 4096-dimensional feature vector from the feature map to represent each image region.

Algorithm 2: The proposed TM^3 tracking algorithm

Input: Initial target bounding box $\mathbf{o}_1 = (x_1, y_1, s_1)$.
Output: Estimated object state $\mathbf{o}_t^* = (\hat{x}_t, \hat{y}_t, \hat{s}_t)$.

Repeat
- Generate candidate target regions $\mathcal{R}_t \bigcup \mathcal{E}_t$;
- `// Flow-r process`
- Obtain potential candidate regions $\mathcal{R}_t'$;
- MBS: Find the optimal region $\mathcal{I}_r^*$ from $\mathcal{R}_t'$;
- `// Flow-e process`
- Obtain potential candidate regions $\mathcal{E}_t'$ by Eq. (14);
- MBS: Obtain $\mathcal{I}_e^*$ and $\mathcal{I}_e^{dist}$ from $\mathcal{E}_t'$;
- Output $\mathbf{o}_t^*$ and $\mathcal{I}^*$ by the fusion step in Eq. (15);
- `// Update Tmpl`$_e$
- **if** *MBS*$(\mathcal{I}^*, \textit{Tmpl}_e) > 0.5$ **then** $\text{Tmpl}_e \leftarrow \mathcal{I}^*$;
- `// Select the representative result`
- **if** $t \bmod 10 = 0$ **then** Obtain $\mathbf{T}_i$ by Algorithm 1;
- `// Update` $\mathcal{D}$ `and Tmpl`$_r$
- **if** *Num*$(\mathcal{D}) = N_D$ **then** Obtain Tmpl_r by $\mathcal{D}$;
- **else if** *Num*$(\mathcal{D}) < N_D$ **then**
  - $\mathcal{D} = \mathcal{D} \cup \mathbf{T}_i$, $\text{Num}(\mathcal{D}) := \text{Num}(\mathcal{D}) + 1$;
- **end**
- **else** "First-in and First-out" procedure for $\mathcal{D}$;

Until *End of video sequence*;

D. Computational Complexity of MBS

The computation of MBS can be divided into two steps: first to calculate the similarity matrix, and second to pick up r (or s) reciprocal nearest neighbors of each image patch based on the similarity matrix as demonstrated in Eq. (1).

Thanks to the reciprocal k-NN scheme, the generated similarity matrix is sparse and the nonzero elements in the matrix almost spread along its diagonal direction. As a result, the average computational complexity of the similarity matrix reduces from $\mathcal{O}(M^2 d)$ to $\mathcal{O}(Md)$, where d is the feature dimension. After that, we pick up r (or s) reciprocal nearest neighbors of each image patch in $\mathcal{P}$ based on the similarity matrix. Due to the exponential decay operator in Eq. (1), there is no sense to consider a large r and s. Hence, we just consider the $c \leq 4$ nearest neighbors of an image patch to accelerate the computation in our experiment. As described in [22], such operation can be further reduced to $\mathcal{O}(Mc)$ on the average. Finally, the overall MBS complexity is $\mathcal{O}(M^2 cd)$.

IV. Experiments

In this section, we compare the proposed TM^3 tracker with other recent algorithms on two benchmarks including OTB [8] and PTB [23]. Moreover, the results of ablation study and parametric sensitivity are also provided.

A. Experimental Setup

In our experiment, the proposed TM^3 tracker is tested on both color feature (denoted as "TM^3-color") and deep feature (denoted as TM^3-deep). Our TM^3-color tracker is implemented in MATLAB on a PC with Intel i5-6500 CPU (3.20 GHz) and

8 GB memory, and runs about 5 fps (frames per second). The proposed TM^3-deep tracker is based on MatConvNet toolbox [32] with Intel Xeon E5-2620 CPU @2.10GHz and a NVIDIA GTX1080 GPU, and runs almost 4 fps.

Parameter settings In our TM^3-color tracker, every image region is normalized to 36×36 pixels, and then split into a set of non-overlapped 3×3 image patches. In this case, $M = N = \frac{36^2}{3^2} = 144$. In the TM^3-deep tracker, each image region is represented by a 4096-dimensional feature vector, namely $M = N = 4096$. In `Flow-r` process, to generate target regions in the tth frame, we draw $N_r = 700$ samples in translation and scale dimension, $\mathbf{x}_i^t = (\mathbf{c}_r^i, s_r^i), i = 1, 2, \cdots, N_r$, from a Gaussian distribution whose mean is the previous target state $\mathbf{x}_*^{t-1}$ and covariance is a diagonal matrix $\Sigma = \mathrm{diag}(\sigma_x, \sigma_y, \sigma_s)$ of which diagonal elements are standard deviations of the sampling parameter vector $[\sigma_x, \sigma_y, \sigma_s]$ for representing the target state. In our experiments, the sampling parameter vector is set to $[\sigma_x, \sigma_y, 0.15]$ where $\sigma_x = \min\{w/4, 15\}$ and $\sigma_y = \min\{h/4, 15\}$ are fixed for all test sequences, and w, h have been defined in Eq. (14). The number of potential proposals $\mathcal{R}_t'$ and $\mathcal{E}_t'$ are set to $N_r' = N_e' = 50$. In `Flow-e` process, we use the same parameters in EdgeBox as described in [25]. The trade-off parameters δ and β in Eq. (9) are fixed to 5 and 10 accordingly; The number of atoms in the target dictionary $\mathcal{D}$ is decided as $N_D = 12$.

B. Results on OTB

1) Dataset description and evaluation protocols: OTB includes two versions, *i.e.* OTB-2013 and OTB-2015. OTB-2013 contains 51 sequences with precise bounding-box annotations, and 36 of them are colored sequences. In OTB-2015, there are 77 colored video sequences among all the 100 sequences. Specifically, considering that the proposed TM^3-color tracker is executed on CIE Lab and RGB color spaces, the TM^3-color tracker is only compared with other baseline trackers on the colored sequences to achieve fair comparison. Differently, our TM^3-deep tracker can handle both colored and gray-level sequences, so it is evaluated on all the sequences in the above two benchmarks.

The quantitative analysis on OTB is demonstrated on two evaluation plots in the one-pass evaluation (OPE) protocol: the success plot and the precision plot. In the success plot, the target in a frame is declared to be successfully tracked if its current overlap rate exceeds a certain threshold. The success plot shows the percentage of successful frames at the overlap threshold varies from 0 to 1. In the precision plot, the tracking result in a frame is considered successful if the center location error (CLE) falls below a pre-defined threshold. The precision plot shows the ratio of successful frames at the CLE threshold ranging from 0 to 50. Based on the above two evaluation plots, two ranking metrics are used to evaluate all compared trackers: one is the Area Under the Curve (AUC) metric for the success plot, and the other is the precision score at threshold of 20 pixels for the precision plot. For details about the OTB protocol, refer to [8].

Apart from the totally 29 and 37 trackers included in OTB-2013 and OTB-2015, respectively, we also compare our tracker with several state-of-the-art methods, including ACFN [1], RaF [2], TrSSI-TDT [33], DLSSVM [14], Staple [7], DST [34], DSST [35], MEEM [36], TGPR [37], KCF [6], IMT [38], LNLT [39], DAT [40], and CNT [41][7]. Specifically, two state-of-the-art template matching based trackers including ELK [18] and BBT [42] are also incorporated for comparison.

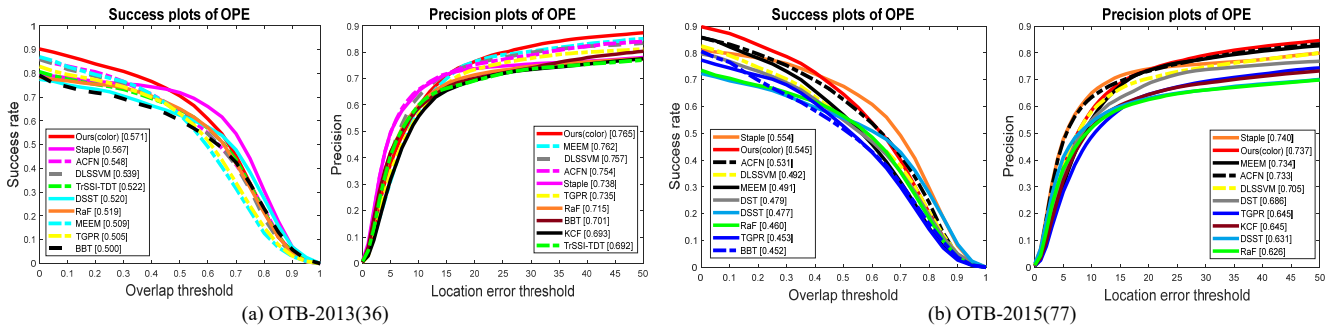
2) Overall performance: Fig. 7 shows the performance of all compared trackers on OTB-2013 and OTB-2015 datasets. On OTB-2013, our TM^3 tracker with color feature achieves 57.1% on average overlap rate, which is higher than the 56.7% produced by a very competitive correlation filter based algorithm Staple. On OTB-2015, it can be observed that the performance of all trackers decreases. The proposed TM^3-color tracker and Staple still provide the best results with the AUC scores equivalent to 54.5% and 55.4%, respectively. Specifically, on these two benchmarks, we see that the competitive template matching based tracker BBT obtains 50.0% and 45.2% on average overlap rate, respectively. Comparatively, our TM^3 tracker significantly improves the performance of BBT with a noticeable margin of 7.1% and 9.3% on OTB-2013 and OTB-2015, accordingly.

We also test our tracker with deep feature and the corresponding performance of these trackers are shown in Fig. 8. Not surprisingly, TM^3-deep tracker boosts the performance of TM^3-color with color feature. It achieves 61.2% and 58.0% success rates on the above two benchmarks, both of which rank first among all compared trackers. On the precision plots, the proposed TM^3-deep tracker yields the precision rates of 82.3% and 79.0% on the two benchmarks, respectively.

The overall plots on the two benchmarks demonstrate that our TM^3 (with colored and deep features) tracker comes in first or second place among the trackers with a comparable performance evaluated by the success rate. It is able to outperform the trackers such as CNN based trackers, correlation filter based algorithms, template matching based approaches, and other representative methods. The favorable performance of our TM^3 tracker benefits from the fact that the discriminative similarity metric, the memory filtering strategy, and the rich feature help our TM^3 tracker to accurately separate the target from its cluttered background, and effectively capture the target appearance variations.

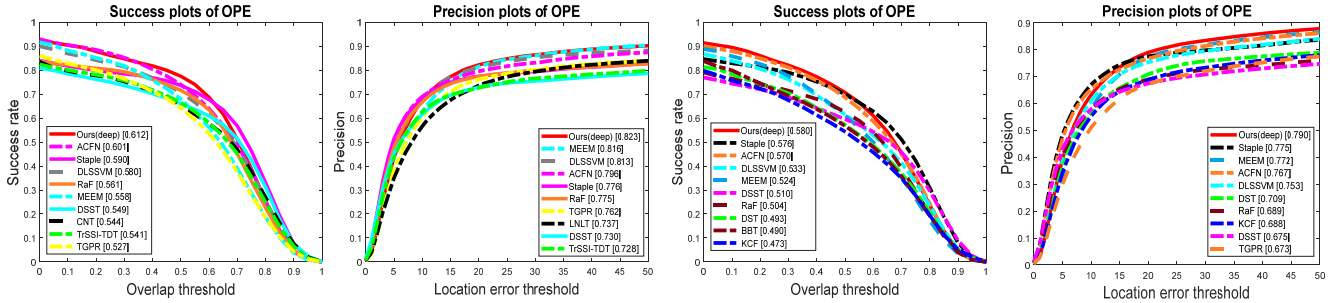
3) Attribute based performance analysis: To analyze the strength and weakness of the proposed algorithm, we provide the attribute based performance analysis to illustrate the superiority of our tracker on four key attributes in Fig. 9. All video sequences in OTB have been manually annotated with several challenging attributes, including *Occlusion* (OCC), *Illumination Variation* (IV), *Scale Variation* (SV), *Deformation* (DEF), *Motion Blur* (MB), *Fast Motion* (FM), *In-Plane Rotation*, *Out-of-Plane Rotation* (OPR), *Out-of-View* (OV), *Background Clutter* (BC), and *Low Resolution* (LR). As illustrated in Fig. 9, our TM^3-deep tracker performs the best on OPR, DEF, IPR, and FM attributes when compared to some representative trackers. The favorable performance of our tracker on appearance variations (*e.g.* OPR, IPR, and DEF) demonstrates the effectiveness of

[7]The implementation of several algorithms *i.e.*, TrSSI-TDT and LNLT is not public, and hence we just report their results on OTB provided by the authors for fair comparisons.



(a) OTB-2013(36) (b) OTB-2015(77)

Fig. 7. Success and precision plots of our color-based tracker TM^3-color and various compared trackers. (a) shows the results on OTB-2013 with 36 colored sequences, and (b) presents the results on OTB-2015 with 77 colored sequences. For clarity, we only show the curves of top 10 trackers in (a) and (b).



(a) OTB-2013(51) (b) OTB-2015(100)

Fig. 8. Success and precision plots of our deep feature based tracker TM^3-deep and various compared trackers. (a) shows the results on OTB-2013 with 51 sequences, and (b) presents the results on OTB-2015 containing 100 sequences. For clarity, we only show the curves of top 10 trackers.

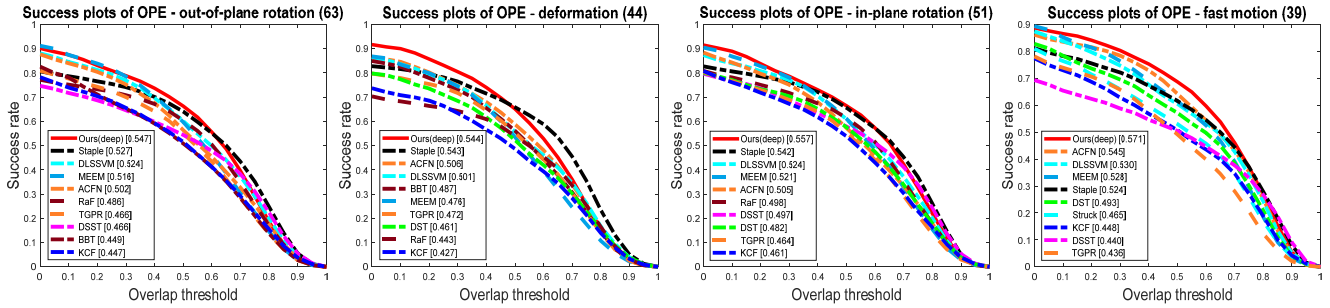


Fig. 9. Attribute-based analysis of our TM^3-deep tracker with four main attribute on OTB-2015(100), respectively. For clarity, we only show the top 10 trackers in the legends. The title of each plot indicates the number of videos labelled with the respective attribute.

the discriminative similarity metric and the memory filtering strategy.

C. Results on PTB

The PTB benchmark database contains 100 video sequences with both RGB and depth data under highly diverse circumstances. These sequences are grouped into the following aspects: target type (human, animal and rigid), target size (large and small), movement (slow and fast), presence of occlusion, and motion type (passive and active). Generally, the human and animal targets including dogs and rabbits often suffer from out-of-plane rotation and severe deformation.

In PTB evaluation system, the ground truth of only 5 video sequences is shared for parameter tuning. Meanwhile, the author make the ground truth of the remaining 95 video sequences inaccessible to public for fair comparison. Hence, the compared algorithms, conducted on these 95 sequences, are allowed to submit their tracking results for performance comparison by an online evaluation server. Hence, the benchmark is fair and valuable in evaluating the effectiveness of different tracking algorithms. Apart from 9 algorithms using RGB data included in PTB, we also compare the proposed tracker with 12 recent algorithms appeared in Section IV-B1. Tab. I shows the average overlap ratio and ranking results of these compared trackers on 95 sequences. The top five trackers are TM^3-deep, TM^3-color, Staple, ACFN, and RaF. The results show that the proposed TM^3-deep tracker again achieves the state-of-the-art performance over other trackers. Specifically, it is worthwhile to mention that our method performs better than CFTs on large appearance variations (*e.g.*, human and animal) and fast movement.

TABLE I
RESULTS ON THE PRINCETON TRACKING BENCHMARK WITH 95 VIDEO SEQUENCES: SUCCESS RATES AND RANKINGS (IN PARENTHESES) UNDER DIFFERENT SEQUENCE CATEGORIZATIONS. THE BEST THREE RESULTS ARE HIGHLIGHTED BY RED, BLUE, AND GREEN, RESPECTIVELY.

Method	Avg. Rank	target type			target size		movement		occlusion		motion type	
		human	animal	rigid	large	small	slow	fast	yes	no	passive	active
TM^3-deep	2.364(1)	0.612(1)	0.672(1)	0.691(1)	0.586(1)	0.502(11)	0.724(1)	0.625(1)	0.526(2)	0.692(4)	0.701(1)	0.551(2)
TM^3-color	3.818(2)	0.551(4)	0.657(2)	0.547(10)	0.535(4)	0.513(9)	0.683(2)	0.597(2)	0.511(3)	0.695(2)	0.646(3)	0.559(1)
Staple [7]	4.909(3)	0.529(5)	0.619(3)	0.553(8)	0.555(3)	0.556(4)	0.652(4)	0.514(4)	0.455(9)	0.690(5)	0.631(5)	0.524(4)
ACFN [1]	5.182(4)	0.574(2)	0.538(8)	0.599(3)	0.505(8)	0.557(3)	0.653(3)	0.504(5)	0.482(7)	0.655(7)	0.603(8)	0.535(3)
RaF [2]	5.818(5)	0.572(3)	0.542(7)	0.557(5)	0.515(7)	0.527(7)	0.582(10)	0.498(6)	0.492(5)	0.706(1)	0.604(7)	0.483(6)
DLSSVM [14]	6.455(6)	0.522(6)	0.584(4)	0.523(16)	0.563(2)	0.559(2)	0.597(6)	0.455(10)	0.458(8)	0.694(3)	0.658(2)	0.433(12)
MEEM [36]	7.455(7)	0.477(10)	0.510(10)	0.556(6)	0.523(5)	0.587(1)	0.610(5)	0.436(12)	0.433(12)	0.644(9)	0.638(4)	0.458(8)
KCF [6]	7.636(8)	0.464(11)	0.519(9)	0.594(4)	0.491(9)	0.547(5)	0.594(7)	0.494(7)	0.417(13)	0.668(6)	0.627(6)	0.480(7)
BBT [42]	9.091(9)	0.422(14)	0.553(5)	0.610(2)	0.452(13)	0.511(10)	0.583(9)	0.521(3)	0.448(10)	0.572(15)	0.575(9)	0.451(10)
DSST [35]	9.909(10)	0.512(7)	0.551(6)	0.472(19)	0.480(10)	0.516(8)	0.591(8)	0.471(8)	0.408(15)	0.651(8)	0.561(11)	0.458(9)
TGPR [37]	11.182(11)	0.484(8)	0.466(16)	0.498(18)	0.519(6)	0.530(6)	0.535(14)	0.459(9)	0.445(11)	0.611(13)	0.521(17)	0.504(5)
DST [34]	11.909(12)	0.436(12)	0.495(11)	0.554(7)	0.416(16)	0.467(12)	0.522(17)	0.413(14)	0.546(1)	0.630(12)	0.546(14)	0.415(15)
DAT [40]	12.364(13)	0.483(9)	0.484(13)	0.545(12)	0.473(12)	0.440(17)	0.543(13)	0.425(13)	0.495(4)	0.577(14)	0.521(18)	0.437(11)
CNT [41]	13.909(14)	0.424(13)	0.455(18)	0.551(9)	0.475(11)	0.459(15)	0.533(15)	0.377(17)	0.484(6)	0.563(16)	0.495(20)	0.421(13)
Struck [43]	14.909(15)	0.354(16)	0.470(14)	0.534(15)	0.450(14)	0.439(18)	0.580(11)	0.390(16)	0.304(19)	0.635(10)	0.544(15)	0.406(16)
IMT [38]	15.000(16)	0.324(17)	0.457(17)	0.545(13)	0.425(15)	0.444(16)	0.530(16)	0.445(11)	0.364(16)	0.536(18)	0.557(12)	0.418(14)
VTD [44]	15.273(17)	0.309(20)	0.488(12)	0.539(14)	0.386(18)	0.462(13)	0.573(12)	0.372(18)	0.283(20)	0.631(11)	0.549(13)	0.385(17)
RGBdet [23]	17.636(18)	0.267(22)	0.409(20)	0.547(11)	0.319(22)	0.460(14)	0.505(20)	0.357(19)	0.348(17)	0.468(20)	0.562(10)	0.342(19)
ELK [18]	17.727(19)	0.386(15)	0.434(19)	0.502(17)	0.352(20)	0.368(20)	0.514(19)	0.395(15)	0.416(14)	0.347(22)	0.528(16)	0.369(18)
CT [45]	19.727(20)	0.311(19)	0.467(15)	0.369(22)	0.390(17)	0.344(22)	0.486(21)	0.315(20)	0.233(23)	0.543(17)	0.421(21)	0.342(20)
TLD [17]	20.273(21)	0.290(21)	0.351(22)	0.444(20)	0.325(21)	0.385(19)	0.516(18)	0.297(22)	0.338(18)	0.387(21)	0.502(19)	0.305(22)
MIL [46]	20.636(22)	0.322(18)	0.372(21)	0.383(21)	0.366(19)	0.346(21)	0.455(22)	0.315(21)	0.256(21)	0.490(19)	0.404(23)	0.336(21)
SemiB [47]	22.818(23)	0.225(23)	0.330(23)	0.327(23)	0.240(23)	0.316(23)	0.382(23)	0.244(23)	0.251(22)	0.327(23)	0.419(22)	0.232(23)
OF [23]	24.000(24)	0.179(24)	0.114(24)	0.234(24)	0.201(24)	0.175(24)	0.181(24)	0.188(24)	0.159(24)	0.223(24)	0.234(24)	0.168(24)

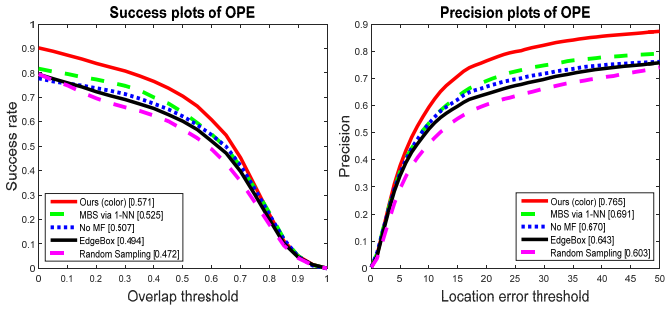


Fig. 10. Verification of four key components in our tracker on OTB-2013(36). "MBS via 1-NN" setting means that only the 1-reciprocal nearest neighbor is utilized for region matching; "No MF" setting denotes that the templates Tmpl_r is also frequently updated as Tmpl_e without the memory filtering strategy; "Edgebox" setting means that the candidate regions are only generated by the EdgeBox approach; "Random sampling" denotes that `Flow-r` process is retained and `Flow-e` process is removed.

D. Ablation Study and Parameter Sensitivity Analysis

In this section, we firstly test the effects of several key components to see how they contribute to improving the final performance, and then investigate the parametric sensitivity of four parameters in the proposed tracker.

1) Key component verification: Several key components includes the scheme of multiple reciprocal nearest neighbors, memory filtering strategy, and the candidate generation scheme. The influence of each component on the final tracking performance is illustrated in Fig. 10.

Firstly, to demonstrate that our multiple reciprocal nearest neighbors scheme is better than simply using one nearest neighbor, we compute MBS by only considering the single nearest neighbor (*i.e.* "MBS via 1-NN"). We see that "1-NN" setting leads to the reduction of 4.6% on average overlap rate when compared with the adopted "MBS" strategy. Therefore, the utilization of multiple nearest neighbors in our tracker enhances the discriminative ability of the existing 1-reciprocal nearest neighbor scheme.

Secondly, to investigate how the memory filtering strategy contributes to improving the final performance, we remove the memory filtering manipulation from our TM^3 tracker (*i.e.* "No MF") and see the performance. The average success rate of such "No MF" setting is as low as 50.7%, with a 6.4% reduction compared with the complete TM^3 tracker. As a result, the memory filtering strategy plays an importance role in obtaining satisfactory tracking performance.

Lastly, to illustrate the effectiveness of two different candidate generation types with the multiple templates scheme, we design two experimental settings: "Edgebox" and "Random Sampling". In "Random Sampling" setting, only `Flow-r` process is retained, which further causes to the invalidation of Tmpl_e and the fusion scheme. In this case, "Random Sampling" directly outputs $\mathcal{I}_r^*$ as the final tracking result without any fusion scheme, and then produces only one template Tmpl_r for updating. As a consequence, the average overlap rate dramatically decreases from the original 57.1% to 47.2% if only `Flow-r` process is used. Likewise, in the "Edgebox" setting, `Flow-e` process is retained and `Flow-r` process is removed, so only Tmpl_r is involved for template updating. Such setting achieves 49.4% success rate and 64.3% precision rate, which are much lower than the original result.

2) Parametric sensitivity: Here we investigate the parametric sensitivity of the number of atoms in $\mathcal{D}$, the kernel width σ_1 in Eq. (1), and two regularized parameters δ and β in Eq. (9). Fig. 11 illustrates that the proposed TM^3 tracker is robust to the variations of these parameters, so they can be easily tuned for practical use.

V. CONCLUSION

This paper proposes a novel template matching based tracker named TM^3 to address the limitations of existing CFTs. The proposed MBS notably improves the discriminative ability

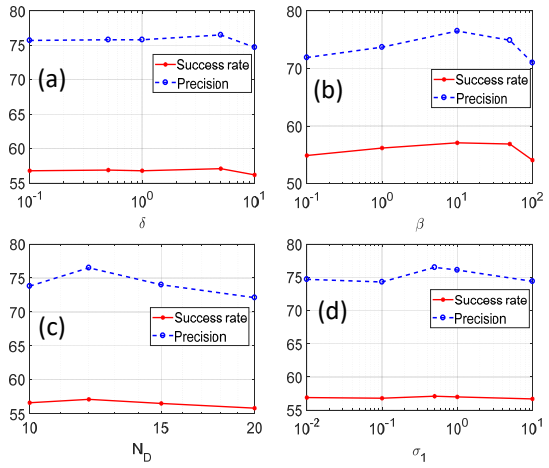


Fig. 11. Tracking performance of success rate and precision versus four varying parameters on OTB-2013(36).

of our tracker as revealed by both empirical and theoretical analyses. Moreover, the memory filtering strategy is incorporated into the tracking framework to select "representative" and "reliable" previous tracking results to construct the current trustable templates, which greatly enhances the robustness of our tracker to appearance variations. Experimental results on two benchmarks indicate that our TM^3 tracker equipped with the multiple reciprocal nearest neighbor scheme and the memory filtering strategy can achieve better performance than other state-of-the-art trackers.

APPENDIX A
THE PROOF OF LEMMA 1

This section aims to simplify the formulation of $\mathbb{E}_{\text{MBS}}(\mathcal{P},\mathcal{Q})$ defined by Definition 1 as follows[8].

Due to the independence among image patches, the integral in Eq. (3) can be decoupled as:

$$\mathbb{E}_{\text{MBS}}(\mathcal{P},\mathcal{Q}) = \int_{p_1}\cdots\int_{p_N}\int_{q_1}\cdots\int_{q_M}\text{MBP}(\mathbf{p}_i,\mathbf{q}_j)\prod_{k=1}^{N}f_P(p_k)\prod_{l=1}^{M}f_Q(q_l)\mathrm{d}P\mathrm{d}Q, \tag{16}$$

where $\mathrm{d}P=\mathrm{d}p_1\cdot\mathrm{d}p_2\cdots\mathrm{d}p_N$, and $\mathrm{d}Q=\mathrm{d}q_1\cdot\mathrm{d}q_2\cdots\mathrm{d}q_M$. By introducing the indictor $\mathbb{I}$, it equals to 1 when $\mathbf{q}_j=\text{NN}_r(\mathbf{p}_i,\mathcal{Q})\wedge\mathbf{p}_i=\text{NN}_s(\mathbf{q}_j,\mathcal{P})$. The similarity $\text{MBP}(\mathbf{p}_i,\mathbf{q}_j)$ in Eq. (1) can be reformulated as:

$$\text{MBP}(\mathbf{p}_i,\mathbf{q}_j)=\exp\left\{-\frac{1}{\sigma_1}\sum_{k=1,k\neq i}^{N}\mathbb{I}\big[d(\mathbf{p}_k,\mathbf{q}_j)\leq d(\mathbf{p}_i,\mathbf{q}_j)\big]\cdot\sum_{l=1,l\neq i}^{M}\mathbb{I}\big[d(\mathbf{q}_l,\mathbf{p}_i)\leq d(\mathbf{p}_i,\mathbf{q}_j)\big]\right\}, \tag{17}$$

which shares the similar formulation of $\text{BBP}(\mathbf{p}_i,\mathbf{q}_j)$ in [22] (see in Eq. (7) on Page 5). And next, by defining:

$$Cp_k=\int_{-\infty}^{\infty}\mathbb{I}[d(\mathbf{p}_k,\mathbf{q}_j)\leq d(\mathbf{p}_i,\mathbf{q}_j)]f_P(p_k)\mathrm{d}p_k\,, \tag{18}$$

[8]Our simplification process of $\mathbb{E}_{\text{MBS}}(\mathcal{P},\mathcal{Q})$ is similar to that of $\mathbb{E}_{\text{BBS}}(\mathcal{P},\mathcal{Q})$. Please refer to Section 3.1 in [22].

and assuming $d(\mathbf{p},\mathbf{q})=\sqrt{(\mathbf{p}-\mathbf{q})^2}=|\mathbf{p}-\mathbf{q}|$, we can rewrite Eq. (18) as:

$$Cp_k=\int_{-\infty}^{\infty}\mathbb{I}\big[\mathbf{p}_k<\mathbf{q}_j^-\vee\mathbf{p}_k>\mathbf{q}_j^+\big]f_P(p_k)\mathrm{d}p_k=F_P(q_j^+)-F_P(q_j^-)\,. \tag{19}$$

Similarly, Cq_l is:

$$Cq_l=\int_{-\infty}^{\infty}\mathbb{I}\big[d(\mathbf{q}_l,\mathbf{p}_i)\leq d(\mathbf{p}_i,\mathbf{q}_j)\big]f_Q(q_l)\mathrm{d}q_l=F_Q(p_i^+)-F_Q(p_i^-)\,. \tag{20}$$

Note that Cp_k and Cq_l only depend on $\mathbf{p}_i$, $\mathbf{q}_j$, and the underlying distributions $f_P(p)$ and $f_Q(q)$. Therefore, $\mathbb{E}_{\text{MBS}}(\mathcal{P},\mathcal{Q})$ can be reformulated as:

$$\mathbb{E}_{\text{MBS}}(\mathcal{P},\mathcal{Q})=\int_{p_i}\int_{q_j}\mathrm{d}p_i\mathrm{d}q_jf_P(p_i)f_Q(q_j)\text{MBP}(\mathbf{p}_i,\mathbf{q}_j)\,.$$

Using the Taylor expansion with second-order approximation $\exp(-\frac{1}{\sigma}x)=1-\frac{1}{\sigma}x+\frac{1}{2\sigma^2}x^2$ and $NCp_k=\sum_{k=1,k\neq i}^{N}\mathbb{I}\big[d(\mathbf{p}_k,\mathbf{q}_j)\leq d(\mathbf{p}_i,\mathbf{q}_j)\big]$ by Eq. (19), and $MCq_l=\sum_{l=1,l\neq i}^{M}\mathbb{I}\big[d(\mathbf{q}_l,\mathbf{p}_i)\leq d(\mathbf{p}_i,\mathbf{q}_j)\big]$ Eq. (20), we have:

$$\begin{aligned}&\mathbb{E}_{\text{MBS}}(\mathcal{P},\mathcal{Q})=\\&1-\frac{1}{\sigma_1}\int_{p_i}\int_{q_j}(MCq_l)\cdot(NCp_k)f_P(p_i)f_Q(q_j)\mathrm{d}p_i\mathrm{d}q_j\\&+\frac{1}{2\sigma_1^2}\int_{p_i}\int_{q_j}(MCq_l)^2\cdot(NCp_k)^2f_P(p_i)f_Q(q_j)\mathrm{d}p_i\mathrm{d}q_j\,.\end{aligned} \tag{21}$$

Finally, after some straightforward algebraic manipulations, the $\mathbb{E}_{\text{MBS}}$ in Eq. (5) can be easily obtained.

APPENDIX B
THE PROOF OF LEMMA 2

This section begins with the formulation of $\mathbb{E}_{\text{MBS}^2}(\mathcal{P},\mathcal{Q})$, and then computes the variance $\mathbb{V}_{\text{MBS}}(\mathcal{P},\mathcal{Q})=\mathbb{E}_{\text{MBS}^2}(\mathcal{P},\mathcal{Q})-\mathbb{E}^2_{\text{MBS}}(\mathcal{P},\mathcal{Q})$ by Lemma 1.

First, the similarity metric $\text{MBP}^2(\mathcal{P},\mathcal{Q})$ is obtained by Eq. (17), namely:

$$\text{MBP}^2(\mathbf{p}_i,\mathbf{q}_j)=\exp\left\{-\frac{2}{\sigma_1}\sum_{k=1,k\neq i}^{N}\mathbb{I}\big[d(\mathbf{p}_k,\mathbf{q}_j)\leq d(\mathbf{p}_i,\mathbf{q}_j)\big]\cdot\sum_{l=1,l\neq i}^{M}\mathbb{I}\big[d(\mathbf{q}_l,\mathbf{p}_i)\leq d(\mathbf{p}_i,\mathbf{q}_j)\big]\right\}. \tag{22}$$

Similar to the above derivation of $\mathbb{E}_{\text{MBS}}(\mathcal{P},\mathcal{Q})$ in Lemma 1, $\mathbb{E}_{\text{MBS}^2}(\mathcal{P},\mathcal{Q})$ can be computed as:

$$\begin{aligned}&\mathbb{E}_{\text{MBS}^2}(\mathcal{P},\mathcal{Q})=1-\\&\frac{2MN}{\sigma_1}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\big[F_P(q^+)-F_P(q^-)\big]\big[F_Q(p^+)-F_Q(p^-)\big]f_P(p)f_Q(q)\mathrm{d}p\mathrm{d}q+\\&\frac{2M^2N^2}{\sigma_1^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\big[F_P(q^+)-F_P(q^-)\big]^2\big[F_Q(p^+)-F_Q(p^-)\big]^2f_P(p)f_Q(q)\mathrm{d}p\mathrm{d}q.\end{aligned} \tag{23}$$

Next, $\mathbb{E}^2_{\text{MBS}}(\mathcal{P},\mathcal{Q})$ can be obtained by Lemma 1 with its second-order approximation, namely:

$$\begin{aligned}
&\mathbb{E}^2_{\text{MBS}}(\mathcal{P},\mathcal{Q}) = 1+ \\
&\frac{M^2N^2}{\sigma_1^2}\left\{\int_{-\infty}^{\infty}\!\!\int_{-\infty}^{\infty}[F_P(q^+)-F_P(q^-)][F_Q(p^+)-F_Q(p^-)]f_P(p)f_Q(q)\mathrm{d}p\mathrm{d}q\right\}^2 \\
&-\frac{2MN}{\sigma_1}\int_{-\infty}^{\infty}\!\!\int_{-\infty}^{\infty}[F_P(q^+)-F_P(q^-)][F_Q(p^+)-F_Q(p^-)]f_P(p)f_Q(q)\mathrm{d}p\mathrm{d}q \\
&+\frac{M^2N^2}{\sigma_1^2}\int_{-\infty}^{\infty}\!\!\int_{-\infty}^{\infty}[F_P(q^+)-F_P(q^-)]^2[F_Q(p^+)-F_Q(p^-)]^2f_P(p)f_Q(q)\mathrm{d}p\mathrm{d}q.
\end{aligned} \tag{24}$$

As a result, Lemma 2 can be easily proved after some straightforward algebraic manipulations on Eq. (23) and Eq. (24).

APPENDIX C
THE PROOF OF LEMMA 3

By Lemma 1 and Eq. (24), $\mathbb{E}_{\text{MBS}^2}(\mathcal{P},\mathcal{Q})$ can be represented by:

$$\begin{aligned}
&\mathbb{E}_{\text{MBS}^2}(\mathcal{P},\mathcal{Q}) = 4\mathbb{E}_{\text{MBS}}(\mathcal{P},\mathcal{Q}) - 3+ \\
&\frac{3MN}{\sigma_1}\int_{-\infty}^{\infty}\!\!\int_{-\infty}^{\infty}[F_P(q^+)-F_P(q^-)][F_Q(p^+)-F_Q(p^-)]f_P(p)f_Q(q)\mathrm{d}p\mathrm{d}q\,.
\end{aligned}$$

Therefore, by Lemma 2 and Eq. (21), we have:

$$\begin{aligned}
&\mathbb{E}_{\text{MBS}^2}(\mathcal{P},\mathcal{Q}) - \mathbb{E}_{\text{MBS}}(\mathcal{P},\mathcal{Q}) = 3\mathbb{E}_{\text{MBS}}(\mathcal{P},\mathcal{Q}) - 3 \\
&+\frac{3MN}{\sigma_1}\int_{-\infty}^{\infty}\!\!\int_{-\infty}^{\infty}[F_P(q^+)-F_P(q^-)][F_Q(p^+)-F_Q(p^-)]f_P(p)f_Q(q)\mathrm{d}p\mathrm{d}q \\
&=\frac{3M^2N^2}{2\sigma_1^2}\int_{-\infty}^{\infty}\!\!\int_{-\infty}^{\infty}[F_P(q^+)-F_P(q^-)]^2[F_Q(p^+)-F_Q(p^-)]^2f_P(p)f_Q(q)\mathrm{d}p\mathrm{d}q \\
&>0\,,
\end{aligned} \tag{25}$$

which completes the proof.

ACKNOWLEDGEMENTS

The authors would like to thank Cheng Peng from Shanghai Jiao Tong University for his work on algorithm comparisons, and also sincerely appreciate the anonymous reviewers for their insightful comments.

REFERENCES

[1] J. Choi, H. Chang, S. Yun, T. Fischer, Y. Demiris, and J. Choi, "Attentional correlation filter network for adaptive visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017.

[2] L. Zhang, J. Varadarajan, P. Suganthan, N. Ahuja, and P. Moulin, "Robust visual tracking using oblique random forests," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017.

[3] F. Liu, C. Gong, T. Zhou, K. Fu, X. He, and J. Yang, "Visual tracking via nonnegative multiple coding," *IEEE Trans. Multimedia*, vol. 19, no. 12, pp. 2680–2691, 2017.

[4] X. Lan, A. Ma, and P. Yuen, "Multi-cue visual tracking using robust feature-level fusion based on joint sparse representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2014, pp. 1194–1201.

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

[6] J. 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, 2015.

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

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

[9] A. Li, M. Lin, Y. Wu, M. Yang, and S. Yan, "NUS-PRO: A new visual tracking challenge," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 335–349, 2016.

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

[11] X. Jia, H. Lu, and M. H. Yang, "Visual tracking via coarse and fine structural local sparse appearance models." *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4555–4564, 2016.

[12] D. Ross, J. Lim, R. Lin, and M. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, pp. 125–141, 2008.

[13] X. Li, W. Hu, Z. Zhang, and X. Zhang, "Robust visual tracking based on an effective appearance model," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2008, pp. 396–408.

[14] J. Ning, J. Yang, S. Jiang, L. Zhang, and M. Yang, "Object tracking via dual linear structured SVM and explicit feature map," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 4266–4274.

[15] P. Sebastian and Y. Voon, "Tracking using normalized cross correlation and color space," in *Proc. Int. Conf. Intell. and Adv. Sys.*, 2007, pp. 770–774.

[16] K. Briechle and U. Hanebeck, "Template matching using fast normalized cross correlation," in *Proc. SPIE Optical Pattern Recognit. XII*, 2001, pp. 95–102.

[17] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, 2012.

[18] S. Oron, A. B., and S. A., "Extended lucas-kanade tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 142–156.

[19] G. Nebehay and R. Pflugfelder, "Consensus-based matching and tracking of keypoints for object tracking," in *Proc. IEEE Winter Conf. Appli. Comput. Vis. (WACV)*, 2014, pp. 862–869.

[20] 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)*, 2015, pp. 749–758.

[21] G. Nebehay and R. Pflugfelder, "Clustering of static-adaptive correspondences for deformable object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015, pp. 2784–2791.

[22] T. Dekel, S. Oron, M. Rubinstein, S. Avidan, and W. Freeman, "Best-buddies similarity for robust template matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015, pp. 2021–2029.

[23] S. Song and J. Xiao, "Tracking revisited using rgbd camera: Unified benchmark and baselines," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2013, pp. 233–240.

[24] T. Zhou, H. Bhaskar, F. Liu, and J. Yang, "Graph regularized and locality-constrained coding for robust visual tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. PP, no. 99, pp. 1–1, 2016.

[25] P. Zitnick, C.and Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 391–405.

[26] A. Krause and V. Cevher, "Submodular dictionary selection for sparse representation," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 567–574.

[27] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization," *SIAM J. Opt.*, 2008.

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

[29] M. Everingham, L. Van, C. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[30] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 1440–1448.

[31] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2009, pp. 248–255.

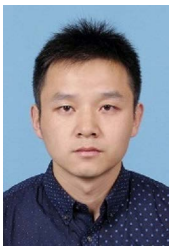
[32] A. Vedaldi and K. Lenc, "Matconvnet: Convolutional neural networks for matlab," in *Proc. ACM Int. Conf. Multimedia*, 2015, pp. 689–692.

[33] W. Hu, J. Gao, J. Xing, C. Zhang, and S. Maybank, "Semi-supervised tensor-based graph embedding learning and its application to visual discriminant tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 1, pp. 172–188, 2017.
[34] J. Xiao, L. Qiao, R. Stolkin, and A. Leonardis, "Distractor-supported single target tracking in extremely cluttered scenes," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 121–136.
[35] M. Danelljan, F. Häger, G., and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. British Mach. Vis. Conf. (BMVC)*, 2014.
[36] 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.
[37] J. Gao, H. Ling, W. Hu, and J. Xing, "Transfer learning based visual tracking with gaussian process regression," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 188–203.
[38] J. Yoon, M. Yang, and K. Yoon, "Interacting multiview tracker," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 5, pp. 903–917, 2016.
[39] B. Ma, H. Hu, J. Shen, Y. Zhang, and F. Porikli, "Linearization to nonlinear learning for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 4400–4407.
[40] H. Possegger, M. Thomas, and B. Horst, "In defense of color-based model-free tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015, pp. 2113–2120.
[41] K. Zhang, Q. Liu, Y. Wu, and M. Yang, "Robust visual tracking via convolutional networks without training," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1779–1792, 2016.
[42] S. Oron, D. Suhanov, and S. Avidan, "Best-buddies tracking," *arXiv:1611.00148*, 2016.
[43] S. Hare, A. Saffari, and P. Torr, "Struck: Structured output tracking with kernels," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2011, pp. 263–270.
[44] J. Kwon and K. Lee, "Visual tracking decomposition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2010, pp. 1269–1276.
[45] K. Zhang, L. Zhang, and M. Yang, "Fast compressive tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 10, pp. 2002–2015, 2014.
[46] B. Babenko, M. 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, 2011.
[47] H. Grabner, C. Leistner, and H. Bischof, "Semi-Supervised Boosting On-line Boosting for Robust Tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2008, pp. 234–247.

Fanghui Liu received the B.S. degree in control science and engineering from Harbin Institute of Technology, Harbin, China, in 2014. He is currently pursuing the Ph.D. degree with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, under the supervision of Prof. Jie Yang. His research areas mainly include computer vision and machine learning with respect to kernel learning, visual tracking, and bayesian learning.

Chen Gong received his dual Ph.D. degree from Shanghai Jiao Tong University (SJTU) and University of Technology Sydney (UTS) in 2016, under the supervision of Prof. Jie Yang and Prof. Dacheng Tao, respectively. Currently, he is a professor in the School of Computer Science and Engineering, Nanjing University of Science and Technology. His research interests mainly include machine learning and data mining. He has published more than 30 technical papers at prominent journals and conferences such as IEEE T-NNLS, IEEE T-IP, IEEE T-CYB, CVPR, AAAI, IJCAI, ICDM, etc. He received the "Excellent Doctorial Dissertation" awarded by Shanghai Jiao Tong University (SJTU) and Chinese Association for Artificial Intelligence (CAAI). He was also enrolled by the "Summit of the Six Top Talents" Program of Jiangsu Province, China.

Xiaolin Huang (S'10-M'12) received the B.S. degree in control science and engineering, and the B.S. degree in applied mathematics from Xi'an Jiaotong University, Xi'an, China in 2006. In 2012, he received the Ph.D. degree in control science and engineering from Tsinghua University, Beijing, China. From 2012 to 2015, he worked as a postdoctoral researcher in ESAT-STADIUS, KU Leuven, Leuven, Belgium. After that he was selected as an Alexander von Humboldt Fellow and working in Pattern Recognition Lab, the Friedrich-Alexander-Universitt Erlangen-Nürnberg, Erlangen, Germany, where he was appointed as a group head. From 2016, he has been an Associate Professor at Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China. In 2017, he has been awarded as "1000-Talent"(Young Program). His current research areas include machine learning, optimization, and their applications.



Tao Zhou received the M.S degree in computer application technology from Jiangnan University, in 2012, and the Ph.D. degree in Pattern Recognition and Intelligent System from the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, in 2016. His current research interests include object detection, visual tracking and machine learning.

Jie Yang received his Ph.D. from the Department of Computer Science, Hamburg University, Germany, in 1994. Currently, he is a professor at the Institute of Image Processing and Pattern recognition, Shanghai Jiao Tong University, China. He has led many research projects (*e.g.*, National Science Foundation, 863 National High Tech. Plan), had one book published in Germany, and authored more than 300 journal papers. His major research interests are object detection and recognition, data fusion and data mining, and medical image processing.

Dacheng Tao (F'15) is Professor of Computer Science and ARC Laureate Fellow in the School of Information Technologies and the Faculty of Engineering and Information Technologies, and the Inaugural Director of the UBTECH Sydney Artificial Intelligence Centre, at the University of Sydney. He mainly applies statistics and mathematics to Artificial Intelligence and Data Science. His research interests spread across computer vision, data science, image processing, machine learning, and video surveillance. His research results have expounded in one monograph and 500+ publications at prestigious journals and prominent conferences, such as IEEE T-PAMI, T-NNLS, T-IP, JMLR, IJCV, NIPS, ICML, CVPR, ICCV, ECCV, ICDM; and ACM SIGKDD, with several best paper awards, such as the best theory/algorithm paper runner up award in IEEE ICDM'07, the best student paper award in IEEE ICDM'13, the distinguished student paper award in the 2017 IJCAI, the 2014 ICDM 10-year highest-impact paper award, and the 2017 IEEE Signal Processing Society Best 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 IEEE, AAAS, OSA, IAPR and SPIE.