# MAT: Multianchor Visual Tracking With Selective Search Region

Zhiwen Fang, Zhiguo Cao, *Member, IEEE*, Yang Xiao, Kaicheng Gong, and Junsong Yuan, *Senior Member, IEEE*

***Abstract*—The core prerequisite of most modern trackers is a motion assumption, defined as predicting the current location in a limited search region centering at the previous prediction. For clarity, the central subregion of a search region is denoted as the tracking anchor (e.g., the location of the previous prediction in the current frame). However, providing accurate predictions in all frames is very challenging in the complex nature scenes. In addition, the target locations in consecutive frames often change violently under the attribute of fast motion. Both facts are likely to lead the previous prediction to an unbelievable tracking anchor, which will make the aforementioned prerequisite invalid and cause tracking drift. To enhance the reliability of tracking anchors, we propose a real-time multianchor visual tracking mechanism, called multianchor tracking (MAT). Instead of directly relying on the tracking anchor inherited from the previous prediction, MAT selects the best anchor from an anchor ensemble, which includes several objectness-based anchor proposals and the anchor inherited from the previous prediction. The objectness-based anchors provide several complementary selective search regions, and an entropy-minimization-based selection method is introduced to find the best anchor. Our approach offers two benefits: 1) selective search regions can increase the chance of tracking success with affordable computational load and 2) anchor selection introduces the best anchor for each frame, which breaks the limitation of solo depending on the previous prediction. The extensive experiments of nine base trackers upgraded by MAT on four challenging datasets demonstrate the effectiveness of MAT.**

***Index Terms*—Anchor proposal, anchor selection, multianchor visual tracking, object tracking, selective search region.**

Manuscript received 9 August 2020; accepted 12 November 2020. Date of publication 31 December 2020; date of current version 1 July 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61702182, Grant 61876211, Grant U1913602, and Grant 61502187; in part by the Hunan Provincial Natural Science Foundation of China under Grant 2018JJ3254; in part by the National Key Laboratory Open Fund of China under Grant 6142113180211; and in part by the National Science Foundation under Grant CNS-1951952. This article was recommended by Associate Editor W. Hu. *(Corresponding author: Zhiguo Cao.)*

Zhiwen Fang is with the School of Biomedical Engineering, Southern Medical University, Guangzhou 510515, China, also with the School of Energy and Mechanical-Electronic Engineering, Hunan University of Humanities, Science and Technology, Loudi 417000, China, and also with the Guangdong Provincial Key Laboratory of Medical Image Processing, Southern Medical University, Guangzhou 510515, China (e-mail: fzw310@gmail.com).

Zhiguo Cao, Yang Xiao, and Kaicheng Gong are with the National Key Laboratory of Science and Technology on Multi-Spectral Information Processing, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: zgcao@hust.edu.cn; yangxiao@hust.edu.cn; kaichenggong@hust.edu.cn).

Junsong Yuan is with the Computer Science and Engineering Department, University at Buffalo, State University of New York, Buffalo, NY 14260 USA (e-mail: jsyuan@buffalo.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2020.3039341.

Digital Object Identifier 10.1109/TCYB.2020.3039341

## I. Introduction

Visual tracking plays a crucial role in intelligent surveillance, human–computer interaction, and robot vision systems. In the past decades, single-target trackers have been widely studied based on a smoothness motion assumption [1]–[6]. It assumes that trackers can predict the target location in a sole search region centering at the previous prediction. However, this assumption is often violated [7]. For example, a low-quality prediction shown as the red solid box in Fig. 1(a) may lead to an invalid search region shown as the red dash box in Fig. 1(b). For clarity, we denote the central subregion of a search region as a tracking anchor. Once the previous prediction is not reliable as the tracking anchor in the current frame, trackers depending solely on the previous prediction will miss the chance to adjust their search regions and fall into a background region.

Generally, there are two ways to mitigate this problem. The first way is to design fine tracking models, which can improve the quality of predictions (i.e., tracking anchors in successive frames). Many tracking models based on SVMs [1]; boosting [8], [9]; random forest [10], [11]; kernel ridge regression [2]; and deep learning [12]–[19] have been optimized continually. Meanwhile, multitracker-based methods [7], [20], [21] are also proposed to enhance the tracking capability. However, it is difficult for tracking models to simultaneously handle multiple tracking challenges, such as partial occlusion and object deformation. Thus, unbelievable tracking anchors are inevitable. The second way is to enlarge the search range, so as to weaken the influence of low-quality tracking anchors. EBT [22] enlarges its search range to the whole image. But arbitrarily enlarging the search range will introduce more distractors, which possibly increase the risk of tracking drift [23], [24]. In order to decrease the distractors, EBT [22] adopts an instance-special objectness estimation based on EdgeBoxes [25] to propose candidate samples over the entire image. Nevertheless, it has high time consumption because an elaborate object proposal method [25] is expected to obtain accurate samples over the entire image.

We argue that if the search range is enlarged in view of discovering multiple selective local search regions rather than arbitrarily using the whole image, object tracking will be more robust with affordable computational load. In this article, we propose a multianchor visual tracking mechanism with selective search region discovery. For short, it is denoted as multianchor tracking (MAT), which can discover selective search regions by means of anchor proposals. An example is

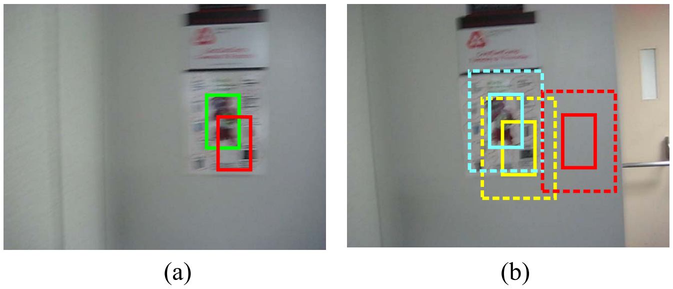


Fig. 1. Illustration of the multianchor visual tracking mechanism. (a) In frame $t-1$, the ground truth (i.e., a bear) is marked in the green box. The red box is a low-quality prediction caused by fast motion and motion blur. (b) In frame $t$, the solid boxes and the dash boxes represent the tracking anchors and their corresponding search regions respectively. The red tracking anchor is inherited from the prediction in frame $t-1$. The blue and yellow anchors are proposed according to an objectness estimation in frame $t$. It can be seen that multiple anchors can increase the probability of tracking success while the red anchor is inadequate for tracking. This figure is best viewed in color.

shown in Fig. 1. The red solid box in frame $t$ is the tracking anchor inherited from the prediction in frame $t-1$. In such a case, the red tracking anchor is a bad anchor because the target is out of its search region. Tracking drift will occur due to the one-fold adoption of the red anchor in frame $t$. The introduced blue search region based on a proposed anchor can provide a supplementary search region, which can effectively avoid the potential tracking drift caused by the red anchor.

Aiming to discover selective search regions in each frame, we introduce a real-time object proposal method to propose tracking anchors. This method includes three main steps. First, using color histograms of the previous predictions, a histogram weight vector is learned to estimate a color histogram score of each pixel in the current frame. To restrain the interference of background and remote distractors, we define a pixel-level object adobe mask and a Gaussian mask, respectively. The pixel-level object adobe mask is calculated to label potential object pixels, because object adobes can locate potential object parts [26]. After obtaining the pixel-level histogram score filtered by the two masks, the objectness score of a region is computed through accumulating pixel-level histogram scores in the region. Using a sliding window [27] with the size of the previous prediction, we can obtain dense objectness scores of all regions. Finally, the regions with high objectness scores are proposed as tracking anchors by nonmaximal suppression [27]. Due to the favorable simplicity of our representation, the procedure of proposing tracking anchors can be done efficiently.

Next, we hope to find the best tracking anchor from a tracking anchor ensemble, including anchor proposals and the anchor inherited from the previous prediction. Ideally, the best tracking anchor should have the ability to provide the search region where a base tracker can obtain high tracking scores with low ambiguity. We design an entropy-regularized loss function as the anchor selection criterion. The entropy regularization term is designed to measure the tracking ambiguity [7], [28]. In the search region centering at the best tracking anchor, the target location is predicted using the base tracker.

To implement a base tracker[1] in our multianchor visual tracking mechanism, we choose nine base trackers: 1) ECO-HC [13], 2) STAPLE [4], 3) MEEM [7], 4) KCF [2], 5) STC [29], 6) CA_DCF [24], 7) CA_MOSSE [24], 8) HCF [12], and 9) UDT [30]. The reasons of choosing these trackers are detailed in the experimental section. Extensive experiments on standard benchmark datasets (OTB100 [31], Temple Color [32], UAV123 [33], and VOT2018 [34]) are carried out to evaluate the effectiveness of multiple anchors and the anchor selection. The experimental results show that the multianchor visual tracking mechanism performs well. For example, on the Temple Color dataset, the distance precision (DP) improvements of STAPLE [4] and HCF [12] are near 9% and 6%, respectively. Moreover, MAT using real-time base trackers can also meet the real-time requirement.

Overall, the main contributions of this article include the following.

1) We propose a MAT mechanism with selective search region discovery, denoted as MAT. Different from the traditional trackers based on a hypothetical valid search region, MAT selectively increases the search range based on anchor proposals and allows a base tracker to choose the best one for predicting the target location. It can facilitate numerous kinds of tracking paradigms.
2) A real-time object proposal method is introduced for visual tracking to propose tracking anchors and a minimum loss criterion based on entropy regularization is provided as the selection criterion.

The source code of this work is published online.[2]

The remainder of this article is organized as follows. The related work is introduced in Section II. An overview of multianchor visual tracking is introduced in Section III. Then, tracking anchor proposals and tracking anchor selection are illustrated in Sections IV and V, respectively. Experimental results and discussions are shown in Section VI. Section VII concludes this article.

## II. Related Work

Great achievements in single-target visual tracking have been made over past decades [30], [35]–[54]. In this section, we first review two main solutions for alleviating this problem caused by the motion assumption: 1) tracking model optimization and 2) search range enlargement. Then, relevant object proposal methods for visual tracking are summarized.

*Tracking Model Optimization:* Traditionally, most methods incline to define high-performance tracking models to improve the prediction quality. In these methods, trackers founded on the generative model locate an object by measuring the similarity between the object and its reference model [40]–[42], [55]. Trackers relying on the discriminative model introduce a large

[1] In this article, we pay more attention to lightweight trackers. Due to the spoofability and equipment dependency of the convolutional neural network (CNN), lots of classical lightweight trackers are still widely used. The other reason is that the CNN-based methods with different depths have different receptive fields at the last prediction layer, which needs fine-tuned sampling parameters in the procedure of our tracking anchor selection according to the characteristics of different CNN-based trackers.

[2] https://github.com/fzw310/MAT-tracking.git

variety of suitable learning models into visual tracking, such as structured output SVMs [1]; boosting [8], [9]; online random forest [10], [11]; kernel ridge regression [2], [56]; multiple instance learning [57]; and deep learning [12]–[19]. In addition, several multitracker-based methods [7], [20], [21], [58] are proposed to enhance the tracking capability. However, these trackers share the common problem with single-anchor-based trackers, that is, they would be unable to rectify their tracking anchors if the previous prediction is unsuitable as the tracking anchor in the current frame.

*Search Range Enlargement:* Another approach is to enlarge the radius of a search region, so as to weaken the influence of the low-quality tracking anchor. But arbitrarily expanding search range introduces more distractors and possibly increases the risk of tracking drift [24]. Lately, tracking methods [59]–[61] based on particle filtering have been widely researched. The particle filter in a large search scale is employed to vote for the target location and estimate the object scale on the basis of importance sampling [62]. Nevertheless, the initial random sampling is blind to the target information [22] and the computational complexity of the resampling procedure is high [63]. In [60] and [61], the anchors provided by particles focus on reducing the computing load rather than selectively enlarge the search region. However, the workload is still heavy (e.g., 1.8 fps for ten particles). Differently, Zhu *et al.* [22] used modified EdgeBoxes [25] to propose candidate samples over the entire image. The modified EdgeBoxes aims at improving the ranking quality with high computational complexity.

*Object Proposal Methods for Visual Tracking:* In previous works, object proposal methods efficiently promote high detection rate with a few proposals [25]–[27], [64]. Enlightened by the high detection rate, several visual tracking methods [16], [22], [65]–[67] have tried object proposal methods as a supportive cue. In [65], a linear combination of tracking model scores and adaptive objectness scores based on BING [27] is utilized to compute the final scores. Huang *et al.* [66] treated Edgeboxes [25] as a post-processing step to enhance the adaptability to size variety. Zhu *et al.* [22] modified Edgeboxes [25] to rerank proposals based on a separate classifier. For visual tracking, dynamic objectness [67] based on the motion saliency is proposed to obtain the clusters of similarly moving target points. Inspired by [64], Li *et al.* [16] designed a region proposal network to propose the samples according to the ground truth object annotation in the first frame. However, due to the deficiency of update mechanism, these object proposal methods cannot adapt to the appearance changes. In [54], a three-step proposal method based on Faster R-CNN [64] is introduced to generate candidates. However, the initial proposals depend on the general category information rather than the appearance information of tracking targets, which would lead to tracking drift caused by ambiguous category information in challenging scenes. In addition, the time consumption of [25] and [26] is high and the application of the GPU-based object proposal method [16], [54], [68] will be restricted due to special equipments. Drawing support from updatable color histograms, we propose an objectness method to propose tracking anchors, which can discover selective research regions at real-time speed.

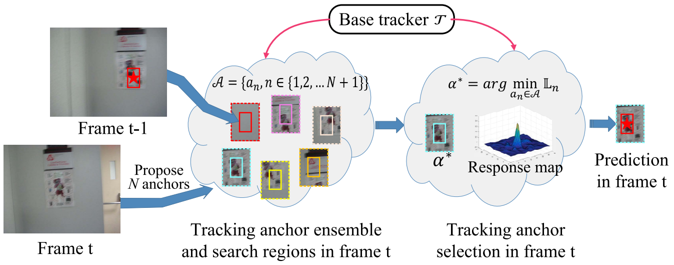

Base tracker $\mathcal{T}$
Frame t-1
Frame t
Propose
$N$ anchors
$\mathcal{A} = \{a_n, n \in \{1,2,\ldots N+1\}\}$
$\alpha^* = arg \min_{a_n \in \mathcal{A}} \mathbb{L}_n$
$\alpha^*$
Response map
Prediction
in frame t
Tracking anchor ensemble
and search regions in frame t
Tracking anchor
selection in frame t

Fig. 2. Multianchor visual tracking with selective search region. The solid boxes, the dash boxes and the red boxes with a red star inside represent tracking anchors, search regions, and predicted results, respectively. 1) Build a tracking anchor ensemble $\mathcal{A}$ based on color histograms in frame $t$. 2) Select the best tracking anchor $a^*$ according to a loss function $\mathbb{L}$ from $\mathcal{A}$. 3) Predict the target location in the search region centering at $a^*$.

## III. Overview

A good tracking anchor should have the ability of leading a tracker to building a reliable search region that contains the real target. Obviously, the smaller the distance between the tracking anchor and the real target, the higher the probability of covering the real target by the search region. According to the motion assumption, traditional methods [1]–[5] simply adopt the previous prediction as the tracking anchor in the current frame. It assumes that trackers can predict the target location in the current frame around the previous result. However, the previous prediction is not always reliable as they suffer from multiple tracking challenges, such as partial occlusion, fast motion, and object deformation [7]. These challenges will make the tracking anchor far from the real target so the corresponding search region is invalid. In addition, traditional tracking methods lack the capability of adjusting their tracking anchors while the anchors are not reliable.

Instead of using the tracking anchor inherited from the previous prediction directly, we propose to discover multiple tracking anchors and select the best one for each frame. The proposed mechanism is denoted as multianchor visual tracking. Fig. 2 is the visual representation and Algorithm 1 illustrates the execution flow of MAT. To find the tracking anchors that have a small distance to the real target, we formulate the tracking anchor discovery as an object proposal problem because object proposal methods can coarsely locate proto-objects efficiently [22], [26], [27]. In frame $t$, tracking anchor candidates are collected by the sliding window method and ranked according to the objectness score $s(\Upsilon)$, which is computed from the pixels included in a candidate region $\Upsilon$. Using nonmaximal suppression [27], the top-N candidates are selected as the anchor proposals depending on their objectness scores $s(\Upsilon)$. The formal definition of $s(\Upsilon)$ and the details of proposing anchors will be discussed in Section IV.

After gathering anchor proposals, an anchor ensemble $\mathcal{A}$ is built to contain these proposed anchors and the anchor inherited from the previous prediction. Intuitively, the best one can be selected from the anchor ensemble according to their objectness scores $s(\Upsilon)$. Whereas, object proposal methods may introduce some false-positive proposals with high objectness scores [69]. Therefore, it is not reliable to treat the proposed anchor with the highest objectness score as the

```
Algorithm 1 MAT: Multi-Anchor Tracking
Input: (1) frame $t$; (2) prediction in frame $t-1$; (3) foreground color histogram $\mathcal{H}_{t-1}(F)$ and background color histogram $\mathcal{H}_{t-1}(B)$; (4) base tracker $\mathcal{T}$
Output: (1) prediction in frame $t$
Start
1. Update $\mathcal{H}_t(F)$ and $\mathcal{H}_t(B)$ using $\mathcal{H}_{t-1}(F)$ and $\mathcal{H}_{t-1}(B)$ by Eqn. 11.
2. Learn a histogram weight vector $\omega$ by Eqn. 9.
3. Calculate the objectness scores of all pixels in frame $t$ by Eqn. 1, 3 and 4, and obtain an objectness score map.
4. Propose top-$N$ tracking anchors from the objectness map using Non-Maximal Suppression.
5. Build an anchor ensemble $\mathcal{A}$ including $N$ anchor proposals at frame $t$ and the prediction at frame $t-1$.
6. Select the best tracking anchor $a^*$ based on $\mathcal{T}$ from $\mathcal{A}$ by Eqn. 12 and 13.
7. Predict the location using $\mathcal{T}$ in the search region centering at $a^*$, and update the model of $\mathcal{T}$.
End
```

best tracking anchor. So as to further select the best tracking anchor, a more effective evaluation criterion is needed. We choose the best tracking anchor in the hope that a base tracker can achieve high tracking scores with low ambiguity in the search region centering at the best anchor. Given a base tracker $\mathcal{T}$, the size of each search region is decided by the parameter of $\mathcal{T}$. A loss function $\mathbb{L}$ is assigned to each anchor, including a log likelihood term and an entropy regularization term. The log likelihood term favors large tracking scores, and the entropy regularization term prefers low ambiguity (i.e., without multiple peak tracking scores). The anchor with the minimal loss will be selected as the best anchor in frame $t$, and then the base tracker $\mathcal{T}$ can predict the target location in the corresponding search region. Tracking anchor selection will be illustrated in Section V.

## IV. Proposing Tracking Anchor

To propose tracking anchors, it is important to assign an objectness score $s(\Upsilon)$ to a candidate region $\Upsilon$. Fig. 3 illustrates the major steps of obtaining the objectness score. Because the color information is insensitive to shape variations and can be extracted efficiently, the color histograms are selected to compute the objectness score $s(\Upsilon)$. Aiming to mine anchors for visual tracking, a histogram weight vector $\omega$ is learned based on the color histograms of previous predictions. Then, given a pixel $u$ characterized by a color histogram feature $\psi(u)$, we can obtain $u$'s color histogram score using $\psi(u)$ weighted by $\omega$ [i.e., (1)]. The score reflects the likelihood of belonging to the tracking target. To reduce the interference of background pixels and remote false object pixels, a pixel-level object adobe mask and a Gaussian mask are defined for pixel $u$, respectively. Weighted by the two masks, the color histogram scores of all pixels in a candidate region $\Upsilon$ are accumulated as the objectness score $s(\Upsilon)$ [i.e., (4)]. Over the entire image, all tracking anchor candidates can be obtained by a sliding window with the size of the previous prediction. Their corresponding objectness scores form an objectness score map, and nonmaximal suppression [27] is applied to propose top-N tracking anchors from this map.

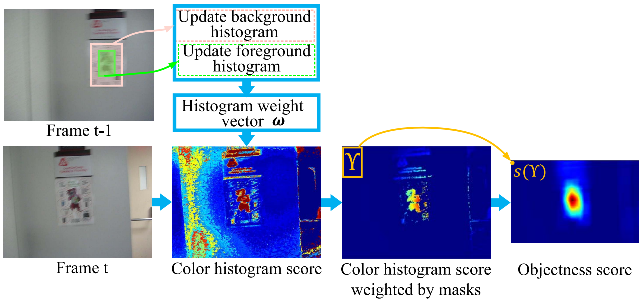

Update background histogram
Update foreground histogram
Histogram weight vector $\omega$
Frame t-1
Frame t
Color histogram score
$\Upsilon$
Color histogram score weighted by masks
$s(\Upsilon)$
Objectness score

Fig. 3. Procedure of calculating the objectness score. The foreground and background bounding boxes are marked in green and pink, respectively.

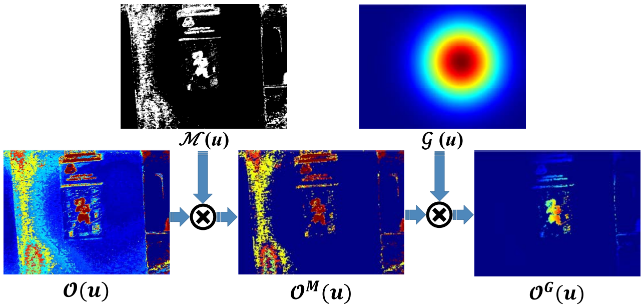


Fig. 4. Illustration of the color histogram score weighted by an object adobe mask $\mathcal{M}(u)$ and a Gaussian mask $\mathcal{G}(u)$.

### A. Objectness Score Estimation

Here, we will introduce the objectness score $s(\Upsilon)$ in detail. Given a $K$-dimension histogram feature $\psi(u)$ of pixel $u$, the color histogram score can be calculated as

$$\mathcal{O}(u) = \omega^T \psi(u) = \sum_{i=1}^{K} \omega^i \cdot \psi^i(u) \tag{1}$$

where $i$ is the dimension index and $\omega$ is a $K$-dimension histogram weight vector learned based on the color histograms of previous predictions. The procedure of calculating $\omega$ will be analyzed in Section IV-B.

According to (1), all pixels have color histogram scores shown as $\mathcal{O}(u)$ in Fig. 4. Obviously, nontarget pixels with nonzero scores will impact the objectness estimation. So it is hoped to highlight potential pixels belonging to the tracking target and restrain nontarget pixels. Adobe Boxes [26] introduces that object adobes can highlight the potential object parts and achieve high performance of discovering potential objects. In [26], an object adobe is defined as the salient superpixel, whose distance from the background is larger than that from the foreground. However, the high time consumption of extracting superpixels is not suitable for visual tracking. Thus, we define a pixel-level object adobe mask as

$$\mathcal{M}(u) = \begin{cases} 1, & D(\psi(u), \mathcal{H}(B)) > D(\psi(u), \mathcal{H}(F)) \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $\mathcal{M}(u) = 1$ means that pixel $u$ is labeled as an object adobe pixel, $\mathcal{H}(Z), Z \in \{F, B\}$ are the $K$-dimension histograms of the foreground $F$ and background $B$ regions, and $D(\psi(u), \mathcal{H}(Z))$ are the histogram intersection distances [70] between the histogram feature $\psi(u)$ of pixel $u$ and $\mathcal{H}(Z)$.

Filtered by $\mathcal{M}(u)$, the histogram score $\mathcal{O}(u)$ is transformed to $\mathcal{O}^M(u)$ shown as Fig. 4. We can see that most nontarget noises are restrained and a large number of potential pixels of the target are maintained. To further restrain remote false object pixels, $\mathcal{O}^M(u)$ is weighted by a Gaussian mask $\mathcal{G}(u)$ with a standard deviation $\sigma$. Because of the high reliability of the annotated bounding box in the first frame [65], $\sigma$ is set to the maximum of width and height of the initialized bounding box in the first frame [22]. Moreover, the centre of the Gaussian mask is decided by the centre of the previous prediction. A sample of the weighted color histogram score $\mathcal{O}^G(u)$ is provided in Fig. 4. $\mathcal{O}^G(u)$ can be computed as

$$\mathcal{O}^G(u) = \mathcal{O}^M(u) \times \mathcal{G}(u) = \mathcal{O}(u) \times \mathcal{M}(u) \times \mathcal{G}(u). \quad (3)$$

After obtaining $\mathcal{O}^G(u)$ of each pixel, we scan over the image to get dense tracking anchor candidates using the sliding-window method. The size of candidates is equal to the size of the previous prediction. A candidate region $\Upsilon$ is scored with an accumulating function of $\mathcal{O}^G(u)$ in $\Upsilon$ as

$$s(\Upsilon) = \frac{1}{|\Upsilon|} \sum_{u \in \Upsilon} \mathcal{O}^G(u) \quad (4)$$

where $s(\Upsilon)$ is the objectness score of $\Upsilon$, and $| * |$ denotes the cardinality of $*$. Using nonmaximal suppression [27], we select top-$N$ locations, which correspond to the positions of tracking anchors in the source image. The size of tracking anchors is the same as the size of the previous prediction. Especially, our multianchor visual tracking can handle scale variation while the base tracker takes a multiscale approach to implement the prediction.

### B. Calculating the Histogram Weight Vector $\omega$

*1) Calculating $\omega$:* In this section, we will introduce the procedure of calculating the $K$-dimension histogram weight vector $\omega$. Given a foreground region $F$ and its surrounding region $B$, linear regression [4] is applied to each pixel independently over $F$ and $B$ for per-frame using objective as

$$\ell(\omega, F, B)=\frac{1}{|F|}\sum_{u\in F}\left(\omega^T\psi(u)-1\right)^2+\frac{1}{|B|}\sum_{u\in B}\left(\omega^T\psi(u)\right)^2 \quad (5)$$

where $\psi(u)$ is the $K$-dimension histogram feature of pixel $u$. Let $b(u)$ present the bin assigned to the color components of pixel $u$. Thus, $\psi(u)$ is a sparse vector that is 1 at index $b(u)$ and 0 everywhere else. Equation (5) can be represented with per feature dimension as

$$\ell(\omega, F, B) = \sum_{i=1}^{K}\left[\frac{C^i(F)}{|F|}\cdot\left(\omega^i - 1\right)^2 + \frac{C^i(B)}{|B|}\cdot\left(\omega^i\right)^2\right] \quad (6)$$

where $C^i(Z) = |\{u \in Z : b(u) = i\}|$ represents the number of pixels in the region $Z \in \{F, B\}$ for which $b(u) = i$. The solution is computed by setting the derivative of (6) to be 0, and the result can be written as

$$\omega^i = \frac{\eta^i(F)}{\eta^i(F) + \eta^i(B)} \quad (7)$$

where $\eta^i(Z) = ([C^i(Z)]/|Z|)$ is the proportion of pixels in a region $Z \in \{F, B\}$ for which the $i$th dimension of the histogram feature is nonzero. $\omega^i$ is set to 0 while $\eta^i(F)$ and $\eta^i(B)$ are both equal to 0. The expression $\eta^i(Z) = ([C^i(Z)]/|Z|)$ also is the $i$th dimension of the normalized color histogram of region $Z$. Thus, $\eta^i(F)$ and $\eta^i(B)$ can be replaced by $\mathcal{H}^i(F)$ and $\mathcal{H}^i(B)$ as

$$\omega^i = \frac{\mathcal{H}^i(F)}{\mathcal{H}^i(F) + \mathcal{H}^i(B)} \quad (8)$$

where $i$ is the dimension index.

*2) Updating $\omega$:* Because visual tracking is an online application, (8) will be further represented with the frame index for frame $t$ as

$$\omega_t^i = \frac{\mathcal{H}_t^i(F)}{\mathcal{H}_t^i(F) + \mathcal{H}_t^i(B)} \quad (9)$$

where $t$ and $i$ are the frame indices and the dimension index, respectively. To adapt to the appearance changes, a linear interpolation [2]–[4], [12] is used to update $\mathcal{H}_t^i(F)$ and $\mathcal{H}_t^i(B)$ in (9) as

$$\begin{aligned}\mathcal{H}_t^i(F) &= (1-\beta)\mathcal{H}_{t-1}^i(F) + \beta\overline{\mathcal{H}}_{t-1}^i(F)\\ \mathcal{H}_t^i(B) &= (1-\beta)\mathcal{H}_{t-1}^i(B) + \beta\overline{\mathcal{H}}_{t-1}^i(B)\end{aligned} \quad (10)$$

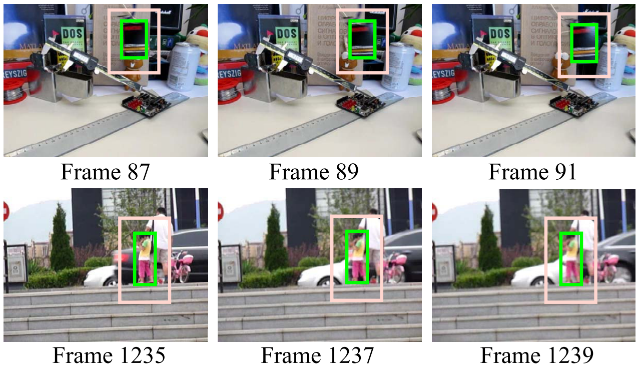
where $\beta$ is an update rate, $\overline{\mathcal{H}}_{t-1}(F)$ and $\overline{\mathcal{H}}_{t-1}(B)$ are the color histograms of foreground and background (relative to the predicted location) at frame $t-1$, respectively. Because the color information of an object is generally more robust than that of the background, we adopt heterogeneous update frequencies $\Delta t_F$ and $\Delta t_B$ for $\mathcal{H}_t^i(F)$ and $\mathcal{H}_t^i(B)$, respectively. Examples are shown in Fig. 5. It can be observed that compared to the foreground, the color information of the background is prone to mutability within a short time. Thus, we redefine (10) as

$$\begin{aligned}\mathcal{H}_t^i(F) &= \begin{cases}(1-\beta)\mathcal{H}_{t-1}^i(F) + \beta\overline{\mathcal{H}}_{t-1}^i(F), & \textit{if } t|\Delta t_F\\ \mathcal{H}_{t-1}^i(F), & \text{otherwise}\end{cases}\\ \mathcal{H}_t^i(B) &= \begin{cases}(1-\beta)\mathcal{H}_{t-1}^i(B) + \beta\overline{\mathcal{H}}_{t-1}^i(B), & \textit{if } t|\Delta t_B\\ \mathcal{H}_{t-1}^i(B), & \text{otherwise}\end{cases}\end{aligned} \quad (11)$$

where $t|\Delta t_Z, Z \in \{F, B\}$ represents that $t$ can be divisible by $\Delta t_Z$, and $\Delta t_Z$, $Z \in \{F, B\}$ is the update interval parameter for $\mathcal{H}_t^i(Z)$, $Z \in \{F, B\}$. $\Delta t_B$ will be smaller than $\Delta t_F$. The infrequent update of $\mathcal{H}_t^i(F)$ is useful for reducing overfitting to the recent foreground [13], which can overcome short-term occlusion to some extent. Moreover, the small value of $\Delta t_B$ is used to handle the background mutability.

## V. Tracking Anchor Selection

In the multianchor visual tracking mechanism, each frame has $N+1$ tracking anchors except the first frame. The anchor ensemble is denoted as $\mathcal{A}= \{a_n, n \in (1, 2, \ldots N+1)\} : N$ anchor proposals and the anchor inherited from the previous


Frame 87 Frame 89 Frame 91

Frame 1235 Frame 1237 Frame 1239

Fig. 5. Examples of the mutability of background regions within a short time. The foreground and background bounding boxes are marked in green and pink, respectively. In the first row, the background region around the tracked object (i.e., a black box) changes due to a fast object moving. The second row is an example of a stationary object (i.e., a girl) with its changing background caused by a moving black car.

prediction. A loss $\mathbb{L}_n$ is assigned to $a_n$ and the best tracking anchor $a^*$ is determined by

$$a^* = \arg\min_{a_n \in \mathcal{A}} \mathbb{L}_n. \tag{12}$$

Centering at the best tracking anchor $a^*$, a search region is built to predict the target location. Thus, a proper loss function is desired for the multianchor visual tracking mechanism to select the best tracking anchor from the anchor ensemble. In this work, a minimum loss criterion $\mathbb{L}_n$ based on entropy regularization is introduced for our task.

Because the process of calculating $\mathbb{L}_n$ of each tracking anchor $a_n$ is the same, the subscript $n$ will be omitted in the rest of this section for conciseness, for example, $a_n \to a$. We assume that a base tracker $\mathcal{T}$ is given, and its own model keeps being updated. The size of a search region is decided by the parameter of $\mathcal{T}$. Before introducing $\mathbb{L}$, we define two sets for the tracking anchor $a$: 1) an instance bag $X$ and 2) a possible label set $\Gamma$, including the ground-truth label of $X$.

In the search region centering at anchor $a$, we use the base tracker $\mathcal{T}$ to obtain a response map, which consists of dense tracking scores. Then, in the search region, an instance bag $X = \{x_m, m \in (1, 2, \ldots M)\}$ is built by $M$ samples with the top-$M$ tracking scores, which are extracted using nonmaximal suppression [27] in the response map. $x_m$ is a candidate image patch, which is labeled by $\phi_m = (y_m, l_m)$. $y_m \in (0, 1)$ represents the background(0)/foreground(1) label, and $l_m$ is the 2-D location of $x_m$. The ground-truth label of $X$ can be represented as $\Phi = \{\phi_m, m \in (1, 2, \ldots M)\}$. It is assumed that only one image patch in $X$ will be treated as the target for visual tracking, and the true label $\Phi$ should be contained in a possible label set [7], [28]. Therefore, we build a small possible set $\Gamma = \{\Phi_m, m \in (1, 2, \ldots M)\}$, where the ground-truth label $\Phi$ of $X$ must be included. For each $\Phi_m = \{(y_m^j, l_m), j \in (1, 2, \ldots M)\}$, $y_m^j$ is 1 when $j = m$ and others are 0.

Based on the instance bag $X$ and the small possible set $\Gamma$, the loss function $\mathbb{L}$ in (12) is defined as

$$\mathbb{L}(X, \Gamma) = -L(a; X, \Gamma) + \gamma H(\Phi|X, \Gamma; a) \tag{13}$$

where $L(a; X, \Gamma)$ is the log likelihood term that favors $\mathcal{T}$'s response map with a large peak score in the search region centering at $a$, $H(\Phi|X, \Gamma; a)$ is the entropy term that prefers the search region with low ambiguity (i.e., $\mathcal{T}$'s response map without multiple large peak scores), and the scalar $\gamma$ is the tradeoff parameter. $L(a; X, \Gamma)$ and $H(\Phi|X, \Gamma; a)$ are, respectively, computed by

$$L(a; X, \Gamma) = \max_{\Phi \in \Gamma} \log P(\Phi|X; a), \tag{14}$$

$$H(\Phi|X, \Gamma; a) = \sum_{\Phi \in \Gamma} P(\Phi|X, \Gamma; a) \log P(\Phi|X, \Gamma; a). \tag{15}$$

Following the assumptions [7] that $\phi^m = (y^m, l^m)$ only depends on $x^m$ and $P(l^m|y^m, x^m) = P(l^m|y^m)$, the graphical model can be described as $x^m \to y^m \to l^m$. It means that the information about the location $l^m$ is provided by the appearance of the image patch $x^m$ only through the appearance-based posterior $P(y^m|x^m; a)$ and the motion prior $P(l^m|y^m)$. Thus, $P(\Phi|X; a)$ is decomposed as

$$\begin{aligned} P(\Phi|X; a) &= \prod_m P\big(\phi^m|x^m; a\big) = \prod_m P\big(y^m, l^m|x^m; a\big) \\ &= \prod_m P\big(l^m|y^m\big) P\big(y^m|x^m; a\big) \end{aligned} \tag{16}$$

where $P(y^m = 1|x^m; a)$ is decided by the tracking score of $\mathcal{T}$ and $P(y^m = 0|x^m; a) = 1 - P(y^m = 1|x^m; a)$; $P(l^m|y^m = 1)$ is provided by the local Gaussian distribution of $\mathcal{T}$ and $P(l^m|y^m = 0)$ is a uniform distribution.

Then, $P(\Phi|X, \Gamma; a)$ is calculated as

$$P(\Phi|X, \Gamma; a) = \frac{P(\Phi|X; a)}{\sum_{\Phi^* \in \Gamma} P(\Phi^*|X; a)}. \tag{17}$$

## VI. Experiments

To validate the effectiveness of the multianchor visual tracking mechanism (MAT for short), we estimate MAT using nine base trackers on four popular object tracking benchmarks (OTB100 [31], Temple Color [32], UAV123 [33], and VOT2018 [34]), which contain hundreds of challenge sequences with various attributes.

The experimental results are organized as follows. In Section VI-A, the details about the evaluation methodology, the base trackers, and the parameter settings are introduced. We give the comparison experiments on the OTB100, Temple Color, and UAV123 datasets in Section VI-B. Section VI-C is for the VOT2018 dataset. The qualitative evaluation is presented in Section VI-D. Then, the parameter sensitivity investigation is illustrated in Section VI-E and the ablation analysis is shown in Section VI-F. Section VI-G provides the potential analysis of MAT. In Section VI-H, we analyze the time consumption. Section VI-I provides a discussion about failure cases. The attribution-based evaluation is available in Section VI-J.

### A. Implementation Details

*Evaluation Methodology:* On the OTB100 dataset, the Temple Color dataset and the UAV123 dataset, we evaluate all trackers according to two measures, which are DP and overlap

TABLE I

DP(%)/AUC(%) Comparison Between the Base Trackers and the Corresponding Counterparts Improved by MAT Mechanism (MAT_Tracker) on the OTB100, Temple Color, and UAV123 Datasets. ΔDP and ΔAUC Represent the Change of DP and AUC, Respectively. ↑ Represents the Increased Value

| Dataset | Tracker | DP / AUC | MAT_tracker | DP / AUC | ΔDP/ΔAUC |
|---|---|---|---|---|---|
| OTB100 | ECO_HC [13] | 84.0 / 63.2 | MAT_ECO_HC | 86.4 / 65.0 | 2.4 ↑ / 1.8 ↑ |
| | STAPLE [4] | 78.4 / 57.9 | MAT_SATPLE | 84.3 / 61.8 | 5.9 ↑ / 3.9 ↑ |
| | MEEM [7] | 77.5 / 53.1 | MAT_MEEM | 78.2 / 53.9 | 0.7 ↑ / 0.8 ↑ |
| | KCF [2] | 69.6 / 47.7 | MAT_KCF | 75.0 / 51.1 | 5.4 ↑ / 3.4 ↑ |
| | STC [29] | 40.8 / 27.7 | MAT_STC | 47.4 / 36.3 | 6.6 ↑ / 8.6 ↑ |
| | CA_DCF [24] | 74.3 / 51.1 | MAT_CA_DCF | 77.9 / 52.8 | 3.6 ↑ / 1.7 ↑ |
| | CA_MOSSE [24] | 59.8 / 44.7 | MAT_CA_MOSSE | 63.2 / 46.8 | 3.4 ↑ / 2.1 ↑ |
| | HCF [12] | 83.2 / 55.8 | MAT_HCF | 85.5 / 57.3 | 2.3 ↑ / 1.5 ↑ |
| | UDT [30] | 76.6 / 59.1 | MAT_UDT | 77.6 / 60.0 | 1.0 ↑ / 0.9 ↑ |
| Temple Color | ECO_HC [13] | 74.0 / 55.4 | MAT_ECO_HC | 76.0 / 57.3 | 2.0 ↑ / 1.9 ↑ |
| | STAPLE [4] | 66.7 / 49.7 | MAT_STAPLE | 75.6 / 55.9 | 8.9 ↑ / 6.2 ↑ |
| | MEEM [7] | 70.9 / 50.0 | MAT_MEEM | 72.0 / 51.3 | 1.1 ↑ / 1.3 ↑ |
| | KCF [2] | 55.8 / 38.7 | MAT_KCF | 63.8 / 44.9 | 8.0 ↑ / 6.2 ↑ |
| | STC [29] | 41.5 / 26.8 | MAT_STC | 47.7 / 34.0 | 6.2 ↑ / 7.2 ↑ |
| | CA_DCF [24] | 59.2 / 42.4 | MAT_CA_DCF | 64.1 / 46.3 | 4.9 ↑ / 3.9 ↑ |
| | CA_MOSSE [24] | 45.8 / 35.0 | MAT_CA_MOSSE | 55.0 / 40.5 | 9.2 ↑ / 5.5 ↑ |
| | HCF [12] | 70.5 / 48.4 | MAT_HCF | 76.1 / 52.2 | 5.6 ↑ / 3.8 ↑ |
| | UDT [30] | 67.9 / 51.4 | MAT_UDT | 75.9 / 56.6 | 8.0 ↑ / 5.2 ↑ |
| UAV123 | ECO_HC [13] | 72.2 / 50.1 | MAT_ECO_HC | 72.7 / 50.6 | 0.5 ↑ / 0.5 ↑ |
| | STAPLE [4] | 66.6 / 45.0 | MAT_STAPLE | 69.4 / 47.2 | 2.8 ↑ / 2.2 ↑ |
| | MEEM [7] | 61.0 / 38.9 | MAT_MEEM | 62.8 / 40.6 | 1.8 ↑ / 1.7 ↑ |
| | KCF [2] | 52.3 / 33.1 | MAT_KCF | 60.8 / 37.8 | 8.5 ↑ / 4.7 ↑ |
| | STC [29] | 47.6 / 31.8 | MAT_STC | 51.8 / 34.7 | 4.2 ↑ / 2.9 ↑ |
| | CA_DCF [24] | 57.5 / 36.2 | MAT_CA_DCF | 61.4 / 38.6 | 3.9 ↑ / 2.4 ↑ |
| | CA_MOSSE [24] | 54.6 / 34.7 | MAT_CA_MOSSE | 60.6 / 38.1 | 6.0 ↑ / 3.4 ↑ |
| | HCF [12] | 65.4 / 39.6 | MAT_HCF | 67.8 / 41.2 | 2.4 ↑ / 1.6 ↑ |
| | UDT [30] | 67.0 / 47.9 | MAT_UDT | 69.5 / 49.0 | 2.5 ↑ / 1.1 ↑ |

success. DP measures the center error between the center $c_t$ of the tracking bounding box and the center $c_{gt}$ of the ground-truth bounding box as $\text{dis} = ||c_t - c_{gt}||_2$. A threshold of 20 pixels is commonly used to rank the trackers. In the precision plot, the center error threshold in pixel distance is varied along the $x$-axis and the percentage of correct prediction according to the threshold is plotted on the $y$-axis. The overlap success is calculated using the intersection over union (IoU) of the tracker bounding box $b_t$ and the ground truth bounding box $b_{gt}$ as $\text{IoU} = [|b_t \cup b_{gt}|/|b_t \cap b_{gt}|]$, where $| * |$ represents the area of $*$. In the success plot, the overlap threshold is varied along the $x$-axis and the percentage of correct prediction according to the threshold is plotted on the $y$-axis. As suggested in [31] and [33], the area under the curve (AUC) is used to evaluate the qualification of tracker bounding box $b_t$. On the VOT2018 dataset, we adopt the expected average overlap (EAO) to rank trackers, which combines the accuracy and failure values in a principled manner. All trackers are implemented using MATLAB/C++ on i7-6600K 4.0-GHz CPU and GeForce GTX TITAN X GPU.

*Introduction About Base Trackers:* The proposed method is implemented in a base tracker, and select the tracking anchor based on the motion model and the observation model of the base tracker. The base trackers using dense prediction maps, such as most correlation-filter-based trackers and partial tracking-by-detection trackers, would be suitable for the proposed multianchor visual tracking mechanism. Facing other categories of trackers, the anchor selection method should be redesigned according to the characteristics of different trackers. In the experiments, because the base trackers emerge in an endless stream, ECO_HC [13], STAPLE [4], MEEM [7], KCF [2], STC [29], CA_DCF [24], CA_MOSSE [24], HCF [12], and UDT [30] are carefully chosen for testing. The main reasons are as follows.

1) ECO_HC is a high-performance tracker with handcrafted features. It is used to evaluate the effectiveness of the proposed mechanism with high-performance trackers.
2) STAPLE uses the HOG and color features to track targets. In the procedure of proposing tracking anchors, the color information is also used. Thus, we use STAPLE to check whether color-based anchors can enhance the performance of color-based trackers.
3) KCF and STC are classical methods with high speed, but their performance is not good enough currently. We try to improve their performance by the proposed mechanism.
4) CA_DCF and CA_MOSSE are utilized to verify that the proposed method can further promote the performance of trackers, whose ability has been improved by other methods (i.e., context-aware).
5) MEEM is a tracking-by-detection tracker.
6) HCF and UDT are delegates of GPU-based trackers.

Because the source codes of some trackers implemented in different computers will achieve different performance to some extent [13], all base trackers and their MAT versions will be rerun in our computer for a fair comparison. For clarity, trackers using multiple tracking anchors are denoted as MAT_ECO_HC, MAT_STAPLE, MAT_MEEM, MAT_KCF, MAT_STC, MAT_CA_DCF, MAT_CA_MOSSE, MAT_HCF, and MAT_UDT, respectively.

*Parameter Settings:* All base trackers are implemented with their published original codes. In all experiments, $N = 3$ anchors are proposed. For anchor proposals, we set $\beta = 0.04$, $\Delta t_F = 3$, and $\Delta t_B = 1$ in (11). The number $K$ of RGB-color

TABLE II

EAO/*A*/*R* COMPARISON BETWEEN THE BASE TRACKERS AND THE CORRESPONDING COUNTERPARTS IMPROVED BY MAT MECHANISM (MAT_TRACKER) ON THE VOT2018 DATASET. ΔEAO, ΔA AND ΔR REPRESENT THE CHANGE OF EAO, *A*, AND *R*, RESPECTIVELY. ↑ AND ↓ REPRESENT THE INCREASED AND DECREASED VALUE. FOR ΔEAO AND ΔA, ↑ MEANS BETTER PERFORMANCE, AND ↓ REPRESENTS MORE ROBUSTNESS FOR ΔR

| Tracker | EAO / A / R | MAT_tracker | EAO / A / R | ΔEAO / ΔA / ΔR |
|---|---|---|---|---|
| ECO_HC [13] | 0.20 / 0.50 / 0.51 | MAT_ECO_HC | 0.22 / 0.52 / 0.46 | 0.02 ↑ / 0.02 ↑ / 0.05 ↓ |
| STAPLE [4] | 0.18 / 0.53 / 0.67 | MAT_STAPLE | 0.24 / 0.52 / 0.44 | 0.06 ↑ / 0.01 ↓ / 0.23 ↓ |
| MEEM [7] | 0.18 / 0.46 / 0.58 | MAT_MEEM | 0.19 / 0.45 / 0.55 | 0.01 ↑ / 0.01 ↓ / 0.03 ↓ |
| KCF [2] | 0.10 / 0.45 / 1.23 | MAT_KCF | 0.14 / 0.45 / 0.78 | 0.04 ↑ / 0.00 ↑ / 0.45 ↓ |
| STC [29] | 0.08 / 0.35 / 1.26 | MAT_STC | 0.12 / 0.43 / 1.08 | 0.04 ↑ / 0.08 ↑ / 0.18 ↓ |
| CA_DCF [24] | 0.11 / 0.44 / 1.05 | MAT_CA_DCF | 0.15 / 0.47 / 0.77 | 0.04 ↑ / 0.03 ↑ / 0.28 ↓ |
| CA_MOSSE [24] | 0.14 / 0.40 / 0.82 | MAT_CA_MOSSE | 0.16 / 0.42 / 0.70 | 0.02 ↑ / 0.02 ↑ / 0.12 ↓ |
| HCF [12] | 0.17 / 0.47 / 0.63 | MAT_HCF | 0.22 / 0.47 / 0.42 | 0.05 ↑ / 0.00 ↑ / 0.21 ↓ |
| UDT [30] | 0.17 / 0.47 / 0.60 | MAT_UDT | 0.18 / 0.48 / 0.54 | 0.01 ↑ / 0.01 ↑ / 0.06 ↓ |

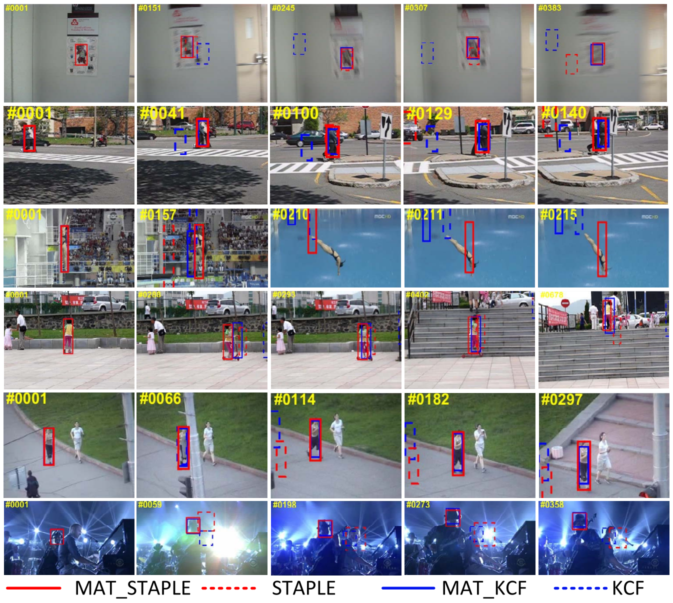


Fig. 6. Qualitative evaluation of STAPLE [4], KCF [2], and their corresponding counterparts improved by MAT on six challenging sequences. From top to bottom, the sequences are *blurOwl*, *couple*, *diving*, *girl2*, *jogging*, and *shaking*, respectively.

histogram bins is 32 × 32 × 32. For anchor selection, we use $M = 10$ samples for each search region and $\gamma = 45$ in (13). Practically, in order to achieve better performance, these parameters can be fine tuned according to the characteristics of different base trackers.

### B. Quantitative Evaluation on the OTB100, Temple Color, and UAV123 Datasets

In this section, we analyze MAT on the OTB100, Temple Color, and UAV123 datasets. Because the color information is used to build the objectness map, selective tracking anchors are only proposed in color sequences (i.e., 75 color sequences on the OTB100 dataset and all sequences on the Temple Color and UAV123 datasets). The results are listed in Table I. It can be observed as follows.

1) Our tracking anchors effectively improve the performance of base trackers. The reason is that multianchor-based trackers will increase the probability of tracking success through selectively enlarging the search range by means of anchor proposals.
2) In [24], a context-aware technique is utilized to optimize the tracker model, which can promote the tracking performance. When CA_DCF [24] and CA_MOSSE [24] are adopted in the multianchor visual tracking mechanism, their results are further improved. It demonstrates that, though tracking models have been optimized, the unsatisfactory predictions cannot be avoided. Multiple tracking anchors can help them to decrease the risk of tracking drift.

### C. Quantitative Evaluation on the VOT2018 Dataset

The VOT2018 includes 60 sequences and the performance is estimated according to EAO, Accuracy and Robustness. Because all base trackers are run with their published original codes rather than their updated codes on the OTB dataset, the Temple Color dataset and the UAV123 dataset, the results on the VOT2018 are also obtained from their published original codes. The details are provided in Table II. For ΔEAO and ΔA, ↑ means better performance and ↓ represents more robustness for ΔR. It can be observed that the proposed multianchor mechanism generally improves the performance of base tackers on the VOT2018 dataset.

### D. Qualitative Evaluation

Fig. 6 shows some tracking results of STAPLE [4], KCF [2], and their corresponding counterparts upgraded by MAT on six challenging sequences. From top to bottom, the sequences are *blurOwl*, *couple*, *diving*, *girl2*, *jogging*, and *shaking*, which are respectively, under the attributes of out-of-plane rotation (*girl2* and *shaking*), in-plane rotation (*diving* and *shaking*), motion blur (*blurOwl*), deformation (*diving* and *couple*), and occlusion (*jogging* and *girl2*). It can be seen that:

1) MAT-based trackers have good performance in these challenging sequences. For example, the dash boxes in *blurOwl* sequence drift away from the real target locations because of fast motion and motion blur shown as frame 151 and frame 307. However, MAT_STAPLE and MAT_KCF can locate the target accurately relying on the anchor proposals;
2) The performance of STAPLE [4] is better than that of KCF [2] under motion blur (*blurOwl*) and deformation (*couple*), because STAPLE [4] combines the color and structure information together. However, once STAPLE [4] loses the target in frame 307 of

TABLE III
DP(%)/AUC(%) COMPARISON WITHIN MAT_STAPLE USING DIFFERENT PARAMETERS. THE BEST VALUES ARE HIGHLIGHTED BY BOLD

| Parameter | Tracker | DP / AUC |
|---|---|---|
| K | MAT_SATPLE (K=8) | 79.7 / 59.2 |
| | MAT_SATPLE (K=16) | 82.0 / 59.8 |
| | MAT_SATPLE (K=32) | **84.3 / 61.8** |
| | MAT_SATPLE (K=64) | 83.6 / 61.1 |
| | MAT_SATPLE (K=128) | 80.0 / 59.2 |
| M | MAT_SATPLE (M=5) | 83.0 / 61.3 |
| | MAT_SATPLE (M=10) | **84.3 / 61.8** |
| | MAT_SATPLE (M=15) | 83.3 / 61.2 |
| | MAT_SATPLE (M=20) | 82.4 / 61.0 |
| $\beta$ | MAT_SATPLE ($\beta$=0.02) | 83.2 / 61.5 |
| | MAT_SATPLE ($\beta$=0.04) | **84.3 / 61.8** |
| | MAT_SATPLE ($\beta$=0.06) | 82.5 / 60.6 |
| | MAT_SATPLE ($\beta$=0.08) | 83.3 / 61.4 |
| $\triangle t_F / \triangle\ t_B$ | MAT_SATPLE ($\triangle t_F = 1, \triangle t_B = 1$) | 82.1 / 60.6 |
| | MAT_SATPLE ($\triangle t_F = 1, \triangle t_B = 3$) | 81.8 / 59.8 |
| | MAT_SATPLE ($\triangle t_F = 3, \triangle t_B = 3$) | 83.1 / 61.1 |
| | MAT_SATPLE ($\triangle t_F = 3, \triangle t_B = 1$) | **84.3 / 61.8** |

TABLE IV
DP(%)/AUC(%) COMPARISON WITHIN MAT_STAPLE USING DIFFERENT ABLATION ITEMS. *w/o* REPRESENTS "WITHOUT." THE BEST VALUES ARE HIGHLIGHTED BY BOLD

| Ablation item | Tracker | DP / AUC |
|---|---|---|
| Mask ($\mathcal{M}(u)$ and $\mathcal{G}(u)$) | MAT_SATPLE | **84.3 / 61.8** |
| | MAT_SATPLE w/o $\mathcal{M}(u)$ | 83.8 / 61.5 |
| | MAT_SATPLE w/o $\mathcal{G}(u)$ | 80.1 / 59.1 |
| | SATPLE | 78.4 / 57.9 |
| Loss ($Likelihood$ and $Entropy$) | MAT_SATPLE | **84.3 / 61.8** |
| | MAT_SATPLE w/o $L$ | 82.7 / 60.7 |
| | MAT_SATPLE w/o $E$ | 82.5 / 61.3 |
| | SATPLE | 78.4 / 57.9 |

*blurOwl*, it would be hard to relocate the target again. Comparatively, MAT_SATPLE and MAT_KCF have a high probability of relocating the target. As the red solid box (MAT_STAPLE) in frame 210 of *diving* and the blue box (MAT_KCF) in frame 288 of *girl2*, the predicted locations are wrong. MAT-based trackers successfully relocate the target in the successive frames. The reason is that the proposed anchors can weaken the dependence on the previous prediction;

3) In the *jogging* sequence, a running person is heavily occluded. STPALE [4] and KCF [2] fall into the background region. Oppositely, MAT_STAPLE and MAT_KCF can catch the real target after the target reappears due to the multiple anchor mechanism and the infrequent update of foreground histograms;
4) Rotation (out-of-plane rotation and in-plane rotation) in a cluttered background often leads to tracking drift shown as the *shaking* sequence. The main reason is that more appropriate anchors mean more opportunities. The anchors, which are proposed by a color-based objectness method, are valuable supplements to the anchor inherited from the previous prediction. The base trackers generally adopt the gradient feature and CNN feature. Thus, these two kinds of tracking anchors can complement each other effectively;
5) MAT-based trackers can handle the challenge under the attribute of scale variation while the base tracker takes a multiscale approach to obtain the prediction. In the *couple* and *jogging* sequences, MAT_STAPLE can deal with scale variation as STAPLE has the multiscale mechanism. Furthermore, the best tracking anchor selected by MAT can help MAT_STAPLE achieve correct scale estimations in the cluttered background shown as the *shaking* sequence.

### *E. Parameter Sensitivity Investigation*

The number $K$ of RGB-color histogram bins, the number $M$ of samples in each search region, the update rate $\beta$ of histograms, and the update interval parameters $\Delta t_F/\Delta t_B$ for the foreground $F$ and background $B$ histograms are tunable parameters within MAT. Here, the performance sensitivity of MAT to $K$, $M$, $\beta$, and $\Delta t_F/\Delta t_B$ will be investigated. We set $K \in \{8, 16, 32, 64, 128\}$, $M \in \{5, 10, 15, 20\}$, $\beta \in \{0.02, 0.04, 0.06, 0.08\}$, and $\Delta t_Z \in \{1, 3\}, Z \in \{F, B\}$, respectively. The experimental results are shown in Table III. According to the results, $K, M$ and $\beta$ are set to 32, 10 and 0.04, respectively. Because there are many combinations between $\Delta t_F$ and $\Delta t_B$, we empirically choose $\Delta t_F = 3$ and $\Delta t_B = 1$. It can be observed as follows.

1) Both the small and the large value of $K$ will weaken the representation ability of histograms.
2) The small value of $M$ would impact the reliability of the entropy term. The large value of $M$ would weak the discrimination ability of the entropy term.
3) MAT with the small value of $\beta$ will ignore the appearance changes of the target. It is easy for MAT with the large value of $\beta$ to fall into the current background in challenge scenes, such as occlusion.
4) MAT_STAPLE ($\Delta t_F = 3$, $\Delta t_B = 1$) achieves the best performance. The main reason is that compared to the foreground, the color information of the background is prone to mutability within a short time.

### *F. Ablation Analysis*

*1) Ablation Analysis of Masks:* The pixel-level object adobe mask $\mathcal{M}(u)$ and the Gaussian mask $\mathcal{G}(u)$ are designed to improve the quality of proposed anchors. In this section, a comparison experiment of MAT using different masks is conducted. The results are shown in Table IV. MAT_STAPLE without the pixel-level object adobe mask $\mathcal{M}(u)$ and the Gaussian mask $\mathcal{G}(u)$ are denoted as MAT_STAPLE w/o M(u) and MAT_STAPLE w/o G(u), respectively. It can be observed as follows.

1) The performances of MAT_STAPLE w/o $\mathcal{M}(u)$ and MAT_STAPLE w/o $\mathcal{G}(u)$ are better than that of STAPLE. It means that both $\mathcal{M}(u)$ and $\mathcal{G}(u)$ are useful for proposing tracking anchors.
2) MAT_STAPLE achieves the best performance profiting from these two masks simultaneously.
3) Due to the dataset bias that the current position of the tracking target generally appears near the previous position in most cases of existing datasets, the influence of $\mathcal{G}(u)$ is greater than that of $\mathcal{M}(u)$.
4) When the anchors are proposed equally over the entire image [i.e., MAT_STAPLE w/o $\mathcal{G}(u)$], the proposed

anchors and the anchor selection mechanism improve the performance.

*2) Ablation Analysis of Loss Terms:* There are two terms in the loss function [i.e., (13)]. Here, a comparison experiment within MAT_STAPLE with different combinations of the loss terms is designed. The results are shown in Table IV. MAT_STAPLE without the likelihood term and MAT_STAPLE without the entropy term are denoted as MAT_STAPLE w/o L and MAT_STAPLE w/o E, respectively. We can observe that both MAT_STAPLE w/o L and MAT_STAPLE w/o E achieve the performances better than the base tracker STAPLE but worse than MAT_STAPLE. It means that under the support of proposed anchors, the likelihood term and the entropy term both benefit enhancing the tracking performance.

### G. Potential Analysis

In this section, we design an experiment to illustrate the potential of MAT. Because it is hard to do the quantitative simulation of final prediction results as different base trackers are implemented in our method, we focus on the stage of proposing tracking anchors.

Here, we set several hypotheses as follows.

1) The tracking capability of a base tracker is defined as a constant value $\Theta \in \{0.50, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95, 1.00\}$, which means that the overlap between the tracking anchor inherited from the previous prediction and the ground truth of the current frame is equal to $\Theta$.
2) There are no tracking drifts because it is too difficult to simulate this phenomenon. It infers that the base tracker can track targets according to the overlap score $\Theta$ in all frames.
3) In each frame, the scale rate is randomly selected from a set $\{1.0824, 1.0612, 1.0404, 1.0200, 1.000, 0.9804, 0.9612, 0.9423, 0.9238\}$. It is used to simulate the scale errors.

The actual situation of visual tracking is much more complicated than the experimental hypotheses. Thus, we mainly provide trend analyses. According to these hypotheses, we estimate the potential using the overlap between the best tracking anchor and the ground truth. The large overlap represents the small distance, which will benefit predicting the target location in the search region. The best overlap of the tracking anchor in the anchor ensemble, which includes $N + 1$ tracking anchors, is adopted as the final score in the current frame. $N$ is set as 3 and the experimental results on the OTB100 dataset are shown in Fig. 7. In Fig. 7(a), the proposed method with the $\Theta$-valued base tracker is denoted as MAT_potential($\Theta$), and the legend contains the AUC score. As shown in Fig. 7(b), the curves represent the relationship between the value of $\Theta$ and the score of AUC. The base_potential and MAT_potential denote the plots of the base tracker and its MAT version, respectively. It can be observed as follows.

1) The proposed anchors can significantly improve the potential of the base tracker (i.e., the AUC score is larger than the value of $\Theta$) while the base performance is not good. The reason is that compared with the base tracker, the color-based objectness method can provide higher quality anchors in most cases.

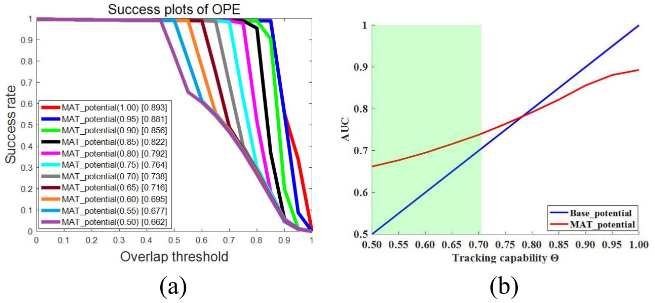


Fig. 7. Illustration of the potential of the proposed tracking anchors on the OTB100 dataset. (a) Success plots of OPE of MAT_potential. (b) Relationship between the tracking capability and AUC.

2) The effect of proposed anchors becomes weak while the base tracker is very strong (e.g., $\Theta > 0.8$), because the tracking anchor inherited from the previous prediction achieve the best quality.
3) To a certain extent, the crossover point in Fig. 7(b) represents the upper bound of the potential of the proposed anchors. Currently, on the OTB100 dataset, the tracking capability of existing visual trackers generally lie in the green region.
4) Theoretically, in Fig. 7(b), the curves at the right of the crossover point should overlap (i.e., the AUC score is equal to the value of $\Theta$). Nevertheless, they do not coincide with each other because of the scale errors that are set in the third hypothesis.

### H. Time Consumption Analysis

In this section, we will estimate the time consumption of MAT on the OTB100 dataset. STAPLE is chosen as the base tracker because of its real-time speed and high performance. Besides the time consumption of STAPLE, two main facts will impact the speed of MAT_STAPLE: 1) the number $N$ of proposed tracking anchors and 2) the size of the region used to propose the tracking anchors. In order to analyze the first factor, we run MAT_STAPLE with different values of $N$ in the whole image, and the corresponding trackers are called MAT_STAPLE($N$). For the second factor, we fix $N$ as 3 and narrow the region of proposing tracking anchors to a square with $(4 \times \max(w_b, h_b))^2$ area, where $(w_b, h_b)$ represents the width and height of the target. This is denoted as MAT_STAPLE$(3)^S$. The detailed results are provided in Tables V and VI. The time consumption is estimated on i7-6600K 4.0-GHz CPU. It can be seen as follows.

1) In Table V, MAT can improve the performance of STAPLE [4] with acceptable time consumption. In a 400×300 image, proposing tracking anchors and anchor selection cost about 10 ms. Moreover, the time comsuption is proportional to the number of anchors and decided by the base tracker (e.g., roughly 3 ms per anchor for STAPLE [4]). As shown in Table V, MAT_STAPLE(1) can be implemented at real-time speed. The DP and the overlap success (AUC) have a

TABLE V
DP(%)/AUC(%) COMPARISON BETWEEN STAPLE [4] AND THE CORRESPONDING COUNTERPARTS WITH DIFFERENT NUMBER $N$ OF ANCHORS ON THE OTB100 DATASET, WHICH IS DENOTED AS MAT_STAPLE($N$). FRAME-PER-SECOND (FPS)/TIME IS THE RUN-TIME PERFORMANCE. THE FIRST AND SECOND BEST VALUES ARE HIGHLIGHTED BY BOLD AND UNDERLINE, RESPECTIVELY

| Tracker | DP | AUC | FPS / Time(mS) |
|---|---|---|---|
| STAPLE [4] | 78.4 | 57.9 | **77.53** / 13.0 |
| MAT_STAPLE(1) | 82.6 | 60.8 | <u>33.74</u> / 29.6 |
| MAT_STAPLE(3) | 84.3 | 61.8 | 28.13 / 35.5 |
| MAT_STAPLE(5) | **85.8** | **63.3** | 24.08 / 41.5 |
| MAT_STAPLE(7) | <u>84.6</u> | <u>62.5</u> | 21.15 / 47.3 |
| MAT_STAPLE(9) | 84.1 | 62.5 | 18.90 / 52.9 |

TABLE VI
DP(%)/AUC(%) COMPARISON BETWEEN MAT_STAPLE(3) AND MAT_STAPLE(3)$^S$ WITH DIFFERENT SIZE OF THE REGION USED TO PROPOSE TRACKING ANCHORS. FPS IS THE RUNTIME PERFORMANCE. THE BEST VALUE IS LABELED BY BOLD

| Tracker | DP | AUC | FPS |
|---|---|---|---|
| MAT_STAPLE(3) | **84.3** | **61.8** | 28.13 |
| MAT_STAPLE(3)$^S$ | 82.6 | 61.2 | **35.71** |

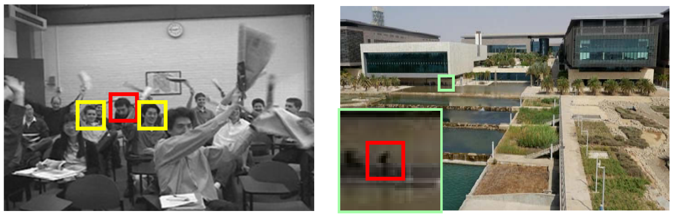


Fig. 8. Failure cases of proposing tracking anchors: gray sequences on the OTB100 dataset and targets with an approximate monochromatic color on the UAV123 dataset. The targets are marked with the red boxes. In the left image, the distractors are marked with the yellow boxes. In the right image, the enlarged area is shown in the green box.

slight decline while a large number of tracking anchors are used. The reason may be that the employment of more tracking anchors will lead to several search regions away from the real target, which would introduce a few distractors [24].

2) Table VI shows that MAT_STAPLE(3)$^S$ achieves higher speed than MAT_STAPLE(3) but with puny performance sacrifice. It means that a small range for anchor proposals will restrain the ability of selectively discovering tracking anchors. Hence, we can set different region sizes for proposing anchors to balance robustness and speed.

### I. Discussion About Failure Cases

We will discuss two scenarios in which the performance of proposed tracking anchors will be weakened: 1) gray sequences and 2) targets with an approximate monochromatic color in color sequences. In the left image of Fig. 8, the tracking anchors are easily disturbed by the distractors because the target only contains gray information. In the right image of Fig. 8, the color histogram of the target is useless because of its monochromatic color. In the procedure of proposing anchors, the color information is used to build the objectness map. Therefore, the proposed method is recommended for color images, and the anchor inherited from the previous prediction is reserved to alleviate the problem caused by the approximate monochromatic color.

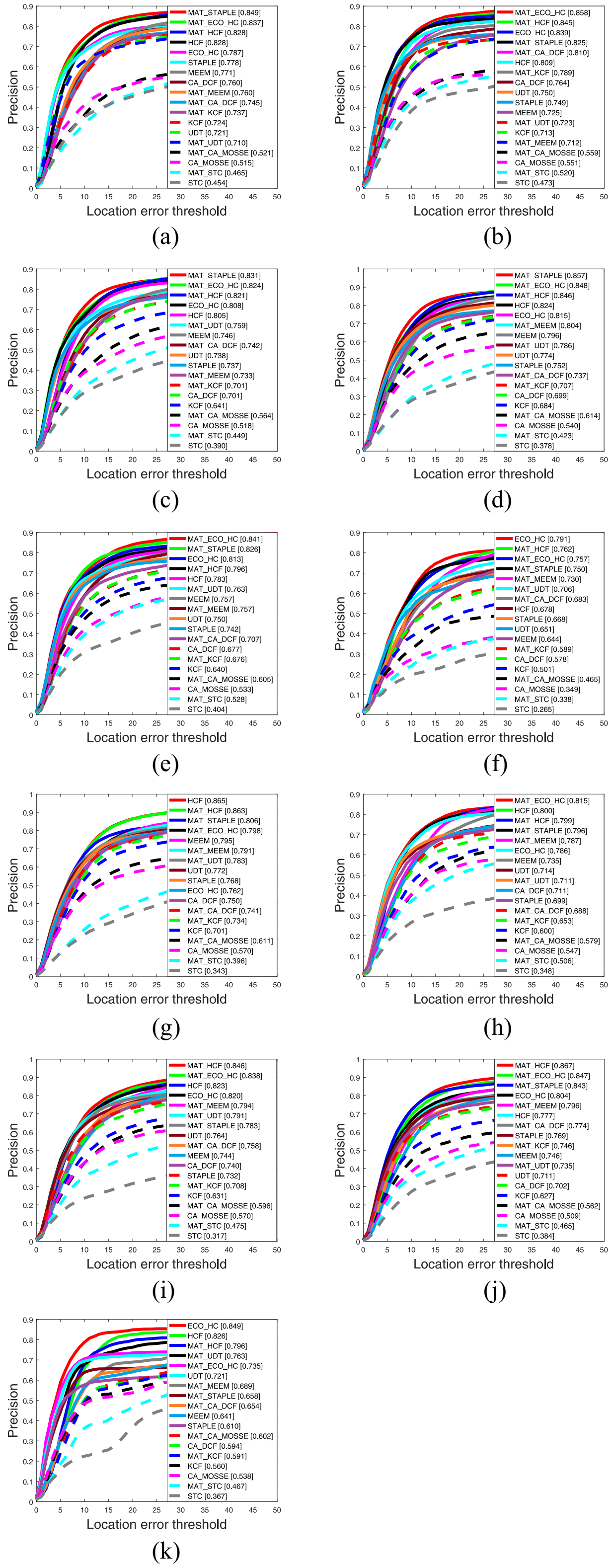


Fig. 9. DP plots over different attributes using one-pass evaluation (OPE) on the OTB100 dataset. (a) Illumination variation. (b) Background clutter. (c) Scale variation. (d) Out-of-plane rotation. (e) Occlusion. (f) Out of view. (g) In-plane rotation. (h) Motion blur. (i) Fast motion. (j) Deformation. (k) Low resolution.

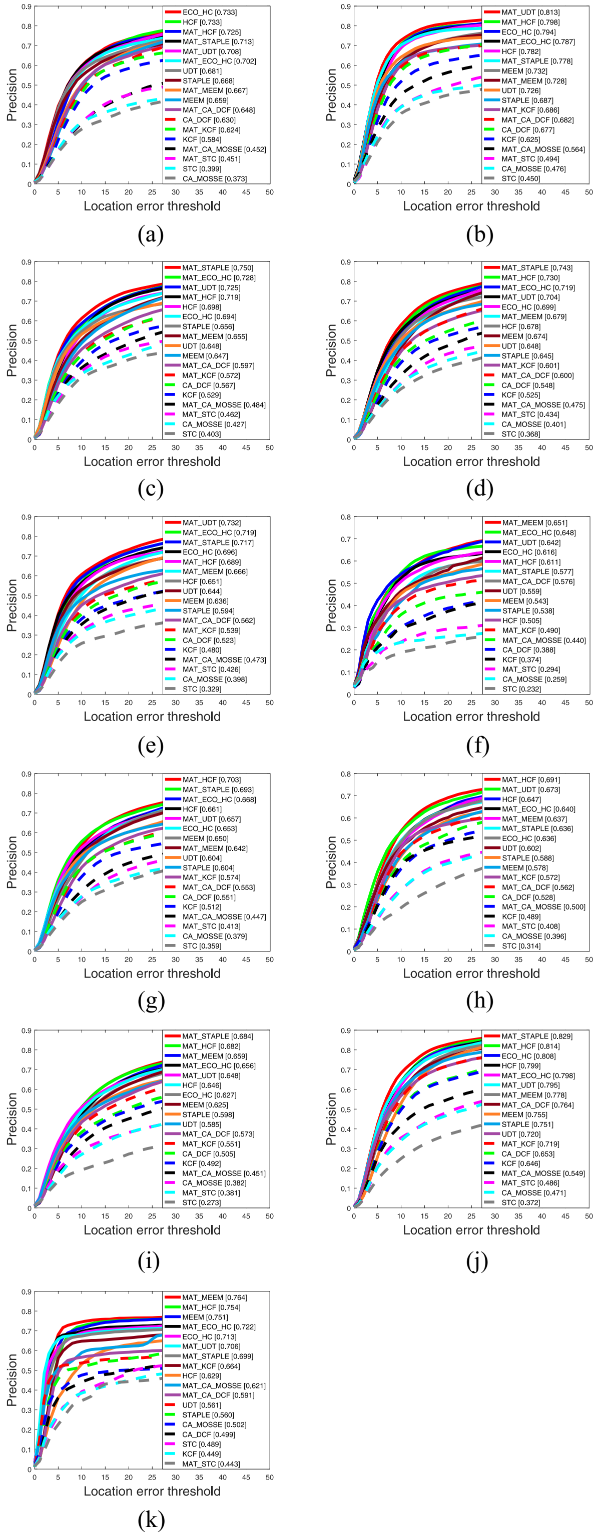


Fig. 10. DP plots over different attributes using OPE on the Temple Color dataset. (a) Illumination variation. (b) Background clutters. (c) Scale variation. (d) Out-of-plane rotation. (e) Occlusion. (f) Out of view. (g) In-plane rotation. (h) Motion blur. (i) Fast motion. (j) Deformation. (k) Low resolution.

### *J. Attribute-Based Evaluation*

The plots of different attributes on the OTB100, Temple Color, and UAV123 datasets are presented in Figs. 9–11. Under different attributes, MAT generally improves the performance of base trackers. We can have the following observations.

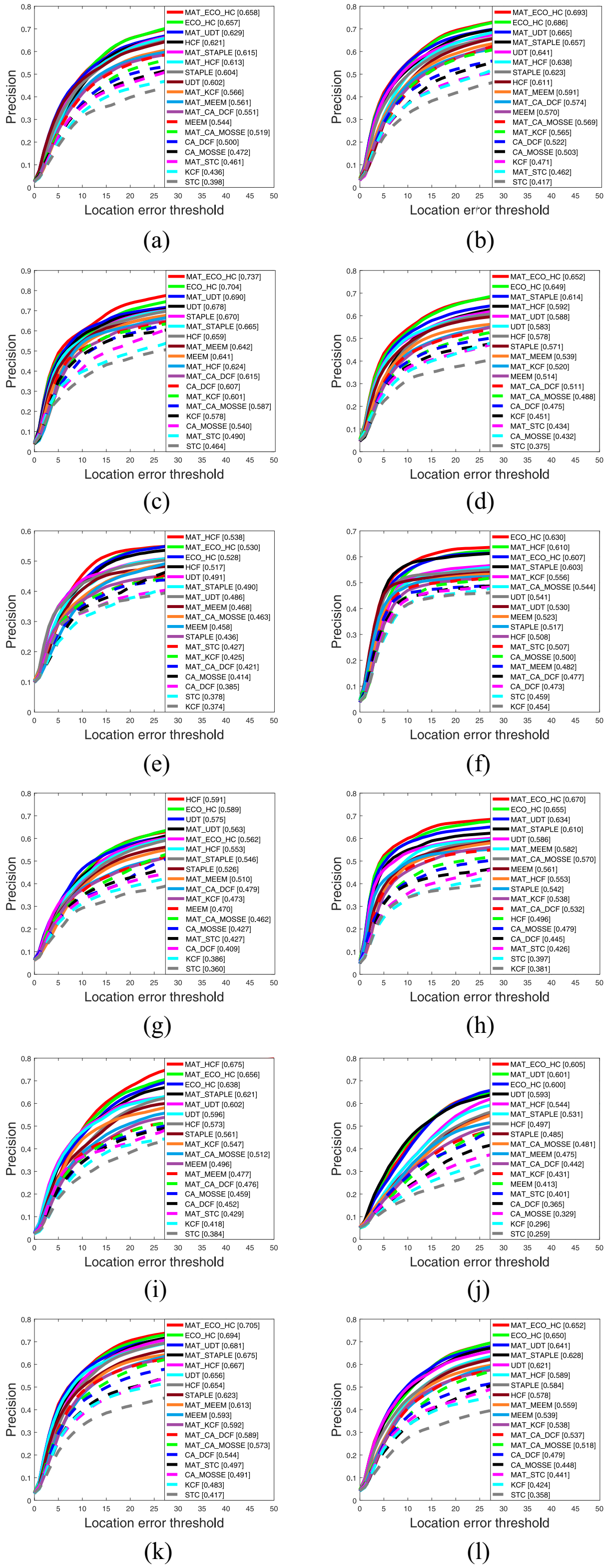


Fig. 11. DP plots over different attributes using OPE on the UAV123 dataset. (a) Viewpoint change. (b) Scale variation. (c) Similar object. (d) Partial occlusion. (e) Full occlusion. (f) Background clutter. (g) Out-of-view. (h) Low resolution. (i) Illumination variation. (j) Fast motion. (k) Camera motion. (l) Aspect ratio change.

1) Under the attributes of out-of-plane rotation and deformation, unstable shape information may lead to tracking drift. The proposed mechanism exploits the per-pixel

color histogram score to estimate region objectness score, and then proposes tracking anchors. Therefore, the proposed anchors are insensitive to shape variations.

2) Under the attribute of occlusion, MAT also leverages the tracker performance, because the multianchor visual tracking mechanism allows the tracker to adjust its tracking anchor so as to relocate the target after it reappears.
3) Under the attribute of fast motion, the previous prediction is often far from the real target in the current frame. Because of the objectness analysis in a large search range, the selectively proposed anchors can easily catch fast moving objects.

## VII. Conclusion

In this article, we proposed a real-time multianchor visual tracking mechanism, which can effectively improve the tracking performance based on selective search region discovery. Moreover, the anchor selection procedure, which provides a chance to select the best tracking anchor in each frame, breaks the limitation of solo depending on the previous prediction. The experimental results showed that our multianchor visual tracking mechanism can facilitate numerous kinds of tracking paradigms. Furthermore, it can discover the selective search regions at real-time speed. Thus, using fast base trackers, the multianchor visual tracking mechanism can meet the requirements of both real-time speed and high performance.

## References

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

[2] 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.

[3] H. Possegger, T. Mauthner, and H. Bischof, "In defense of color-based model-free tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, 2015, pp. 2113–2120.

[4] 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)*, Las Vegas, NV, USA, Jun. 2016, pp. 1401–1409.

[5] Y. Zhou, X. Bai, W. Liu, and L. J. Latecki, "Similarity fusion for visual tracking," *Int. J. Comput. Vis.*, vol. 118, pp. 337–363, Jan. 2016.

[6] C. Deng, Y. Han, and B. Zhao, "High-performance visual tracking with extreme learning machine framework," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2781–2792, Jun. 2020.

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

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

[9] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 234–247.

[10] A. Saffari, C. Leistner, J. Santner, and M. Godec, "On-line random forests," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Kyoto, Japan, 2009, pp. 1393–1400.

[11] C. Leistner, M. Godec, A. Saffari, and H. Bischof, "On-line multi-view forests for tracking," in *Pattern Recognition*. Heidelberg, Germany: 2010, pp. 493–502.

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

[13] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, 2017, pp. 6931–6939.

[14] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 4293–4302.

[15] 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.*, Venice, Italy, 2017, pp. 2555–2564.

[16] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 8971–8980.

[17] Y. Song *et al.*, "VITAL: Visual tracking via adversarial learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 8990–8999.

[18] S. Pu, Y. Song, C. Ma, H. Zhang, and M.-H. Yang, "Deep attentive tracking via reciprocative learning," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2018, pp. 1931–1941.

[19] W. Liu *et al.*, "Deformable object tracking with gated fusion," *IEEE Trans. Image Process.*, vol. 28, pp. 3766–3777, 2019.

[20] F. Tang, S. Brennan, Q. Zhao, and H. Tao, "Co-tracking using semi-supervised support vector machines," in *Proc. 11th IEEE Int. Conf. Comput. Vis.*, Rio de Janeiro, Brazil, 2007, pp. 1–8.

[21] N. Wang and D. Y. Yeung, "Ensemble-based tracking: Aggregating crowdsourced structured time series data," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 1107–1115.

[22] G. Zhu, F. Porikli, and H. Li, "Beyond local search: Tracking objects everywhere with instance-specific proposals," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 943–951.

[23] T. B. Dinh, N. Vo, and G. Medioni, "Context tracker: Exploring supporters and distracters in unconstrained environments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Providence, RI, USA, 2011, pp. 1177–1184.

[24] M. Mueller, N. Smith, and B. Ghanem, "Context-aware correlation filter tracking," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, 2017, pp. 1387–1395.

[25] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. Eur. Conf. Comput. Vis.*, vol. 8693, 2014, pp. 391–405.

[26] Z. Fang, Z. Cao, Y. Xiao, L. Zhu, and J. Yuan, "Adobe boxes: Locating object proposals using object adobes," *IEEE Trans. Image Process.*, vol. 25, no. 9, pp. 4116–4128, Sep. 2016.

[27] M. M. Cheng, Z. Zhang, W. Y. Lin, and P. Torr, "BING: Binarized normed gradients for objectness estimation at 300fps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, 2014, pp. 3286–3293.

[28] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2005, pp. 529–536.

[29] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang, "Fast visual tracking via dense spatio-temporal context learning," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 127–141.

[30] N. Wang, Y. Song, C. Ma, W. Zhou, W. Liu, and H. Li, "Unsupervised deep tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, 2019, pp. 1308–1317.

[31] 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.

[32] P. Liang, E. Blasch, and H. Ling, "Encoding color information for visual tracking: Algorithms and benchmark," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5630–5644, Dec. 2015.

[33] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for UAV tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 445–461.

[34] M. Kristan *et al.*, "The sixth visual object tracking VOT2018 challenge results," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–53.

[35] H. Fan *et al.*, "LaSOT: A high-quality benchmark for large-scale single object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, 2019, pp. 5374–5383.

[36] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "Atom: Accurate tracking by overlap maximization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, 2019, pp. 4660–4669.

[37] X. Li, C. Ma, B. Wu, Z. He, and M.-H. Yang, "Target-aware deep tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, 2019, pp. 1369–1378.

[38] L. Čehovin, A. Leonardis, and M. Kristan, "Robust visual tracking using template anchors," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Lake Placid, NY, USA, 2016, pp. 1–8.

[39] J. Zhang, Y. Wu, W. Feng, and J. Wang, "Spatially attentive visual tracking using multi-model adaptive response fusion," *IEEE Access*, vol. 7, pp. 83873–83887, 2019.

[40] N. Alt, S. Hinterstoisser, and N. Navab, "Rapid selection of reliable templates for visual tracking," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, 2010, pp. 1355–1362.

[41] G. D. Hager and P. N. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 10, pp. 1025–1039, Oct. 1998.

[42] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 810–815, Jun. 2004.

[43] G. Wang, C. Luo, Z. Xiong, and W. Zeng, "SPM-tracker: Series-parallel matching for real-time visual object tracking," 2019. [Online]. Available: arXiv:1904.04452.

[44] Z. Chen, X. You, B. Zhong, J. Li, and D. Tao, "Dynamically modulated mask sparse tracking," *IEEE Trans. Cybern.*, vol. 47, no. 11, pp. 3706–3718, Nov. 2017.

[45] Q. Liu, J. Yang, K. Zhang, and Y. Wu, "Adaptive compressive tracking via online vector boosting feature selection," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4289–4301, Dec. 2017.

[46] Z. Zhang and H. Peng, "Deeper and Wider Siamese networks for real-time visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, 2019, pp. 4591–4600.

[47] G. Zhu, J. Wang, P. Wang, Y. Wu, and H. Lu, "Feature distilled tracking," *IEEE Trans. Cybern.*, vol. 49, no. 2, pp. 440–452, Feb. 2019.

[48] K. Dai, D. Wang, H. Lu, C. Sun, and J. Li, "Visual tracking via adaptive spatially-regularized correlation filters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, 2019, pp. 4670–4679.

[49] Y. Sun, C. Sun, D. Wang, Y. He, and H. Lu, "ROI pooled correlation filters for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, 2019, pp. 5783–5791.

[50] H. Fan and H. Ling, "Siamese cascaded region proposal networks for real-time visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, 2019, pp. 7952–7961.

[51] A. Lukežič, L. Č. Zajc, and M. Kristan, "Deformable parts correlation filters for robust visual tracking," *IEEE Trans. Cybern.*, vol. 48, no. 6, pp. 1849–1861, Jun. 2018.

[52] G. S. Walia, H. Ahuja, A. Kumar, N. Bansal, and K. Sharma, "Unified graph-based multicue feature fusion for robust visual tracking," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2357–2368, Jun. 2020.

[53] L. Wang, H. Lu, and M.-H. Yang, "Constrained superpixel tracking," *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 1030–1041, Mar. 2018.

[54] H. Zhang, J. Chen, G. Nie, and S. Hu, "Uncertain motion tracking based on convolutional net with semantics estimation and region proposals," *Pattern Recognit.*, vol. 102, Jun. 2020, Art. no. 107232.

[55] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–575, May 2003.

[56] H. Kiani Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Venice, Italy, 2017, pp. 1135–1143.

[57] 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.

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

[59] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, no. 1, pp. 125–141, 2008.

[60] T. Zhang, C. Xu, and M.-H. Yang, "Multi-task correlation particle filter for robust object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 1, 2017, p. 3.

[61] T. Zhang, C. Xu, and M. Yang, "Learning multi-task correlation particle filters for visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 365–378, Feb. 2019.

[62] Y. Li, J. Zhu, and S. C. Hoi, "Reliable patch trackers: Robust visual tracking by exploiting reliable patches," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, 2015, pp. 353–361.

[63] J. H. Kotecha and P. M. Djuric, "Gaussian particle filtering," *IEEE Trans. Signal Process.*, vol. 51, no. 10, pp. 2592–2601, Oct. 2003.

[64] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Toward real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2015, pp. 91–99.

[65] P. Liang, Y. Pang, C. Liao, X. Mei, and H. Ling, "Adaptive objectness for object tracking," *IEEE Signal Process. Lett.*, vol. 23, no. 7, pp. 949–953, Jul. 2016.

[66] 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.*, 2015, pp. 1–12.

[67] S. Stalder, H. Grabner, and L. Van Gool, "Dynamic objectness for adaptive tracking," in *Proc. Asian Conf. Comput. Vis.*, 2012, pp. 43–56.

[68] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++: Evolution of siamese visual tracking with very deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4282–4291.

[69] Z. Fang, Z. Cao, Y. Xiao, and H. Lu, "Refine BING using effective cascade ranking," *Appl. Soft Comput.*, vol. 52, pp. 487–500, Mar. 2017.

[70] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, 2013.

**Zhiwen Fang** received the B.S. and M.S. degrees from the Automation School, Beihang University, Beijing, China, in 2004 and 2008, respectively, and the Ph.D. degree from the Huazhong University of Science and Technology, Wuhan, China, in 2017.

He was a Research Fellow with the Institute of Media Innovation, Nanyang Technological University, Singapore, and a Research Scientist with the Institute of High Performance Computing, Research Agency for Science, Technology, and Research, Singapore. His research interests include object detection, object tracking, anomaly detection, medical image analysis, and machine learning.

**Zhiguo Cao** (Member, IEEE) received the B.S. and M.S. degrees in communication and information systems from the University of Electronic Science and Technology of China, Chengdu, China, in 1985 and 1990, respectively, and the Ph.D. degree in pattern recognition and intelligent systems from the Huazhong University of Science and Technology, Wuhan, China, in 2001.

He is a Professor with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology. His research interests spread across image understanding and analysis, depth information extraction, 3-D video processing, motion detection, and human action analysis. His research results, which have been published in dozens of papers at international journals and prominent conferences, have been applied to automatic observation systems for crop growth in agriculture, for weather phenomenon in meteorology, and for object recognition in video surveillance systems based on computer vision.

**Yang Xiao** received the B.S., M.S., and Ph.D. degrees from the Huazhong University of Science and Technology, Wuhan, China, in 2004, 2007, and 2011, respectively.

He was a Research Fellow with the School of Computer Engineering and the Institute of Media Innovation, Nanyang Technological University, Singapore. He is currently an Associate Professor with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology. His research interests involve computer vision, image processing, and machine learning.

**Kaicheng Gong** received the B.S. degree from the School of Automation, Huazhong University of Science and Technology, Wuhan, China, in 2016, and the M.S. degree in pattern recognition and intelligent systems from the Artificial Intelligence and Automation School, Huazhong University of Science and Technology, in 2019.

He currently takes on the object tracking based on computer vision. His research interests involve computer vision, image processing, and machine learning.

**Junsong Yuan** (Senior Member, IEEE) received the M.Eng. degree from the National University of Singapore, Singapore, in 2005, and the Ph.D. degree from Northwestern University, Evanston, IL, USA, in 2009.

He was an Associate Professor with the School of Electrical and Electronics Engineering, Nanyang Technological University (NTU), Singapore. He is currently an Associate Professor and the Director of the Visual Computing Lab, Computer Science and Engineering Department, University at Buffalo, State University of New York, Buffalo, NY, USA.

Dr. Yuan received the Best Paper Award from IEEE TRANSACTIONS ON MULTIMEDIA, Nanyang Assistant Professorship from NTU, and the Outstanding EECS Ph.D. Thesis award from Northwestern University. He served as a Senior Area Editor for *Journal of Visual Communication and Image Representation*, and an Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING, and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. He was the Program Co-Chair of ICME'18 and VCIP'15, and the Area Chair of CVPR, ACM MM, ACCV, WACV, ICPR, and ICIP. He is a Fellow of the International Association of Pattern Recognition.