Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

A quantitative attribute-based benchmark methodology for single-target visual tracking*

Wen-jing KANG[1], Chang LIU[1,2], Gong-liang LIU‡[1]

[1]School of Information Science and Engineering, Harbin Institute of Technology, Weihai 264209, China

[2]School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore

E-mail: kwjqq@hit.edu.cn; liuc0051@e.ntu.edu.sg; liugl@hit.edu.cn

Received May 15, 2019; Revision accepted Oct. 9, 2019; Crosschecked Nov. 12, 2019

Abstract: In the past several years, various visual object tracking benchmarks have been proposed, and some of them have been used widely in numerous recently proposed trackers. However, most of the discussions focus on the overall performance, and cannot describe the strengths and weaknesses of the trackers in detail. Meanwhile, several benchmark measures that are often used in tests lack convincing interpretation. In this paper, 12 frame-wise visual attributes that reflect different aspects of the characteristics of image sequences are collated, and a normalized quantitative formulaic definition has been given to each of them for the first time. Based on these definitions, we propose two novel test methodologies, a correlation-based test and a weight-based test, which can provide a more intuitive and easier demonstration of the trackers' performance for each aspect. Then these methods have been applied to the raw results from one of the most famous tracking challenges, the Video Object Tracking (VOT) Challenge 2017. From the tests, most trackers did not perform well when the size of the target changed rapidly or intensely, and even the advanced deep learning based trackers did not perfectly solve the problem. The scale of the targets was not considered in the calculation of the center location error; however, in a practical test, the center location error is still sensitive to the targets' changes in size.

Key words: Visual tracking; Performance evaluation; Visual attributes; Computer vision
https://doi.org/10.1631/FITEE.1900245 **CLC number:** TP723

1 Introduction

Visual object tracking is the process of continuously locating a target in a video or a sequence of images over time. In other words, object tracking establishes the relationship between locations of the target in frames. As one of the most important and popular parts of computer vision, visual tracking has been used widely in a number of scenarios, such as surveillance and robot navigation (Mathew and Hiremath, 2018). It has also been of assistance in other fields in computer vision, such as video captioning (Zhang JC and Peng, 2019) and classification (Zuo et al., 2019). With the help of modern advanced deep learning techniques, a lot of work, including Li B et al. (2019), has achieved remarkable performance. At the same time, the trackers that use traditional machine learning approaches (e.g., Zuo et al. (2019)) are also developing rapidly along a different pathway. However, with so many trackers being proposed, one thing that should be noted is that no matter which technique is applied in the tracker, currently there does not exist any "standard" performance evaluation system for the trackers. Instead, a huge number of evaluation methodologies, criteria, and challenges exist in this field (Yilmaz et al., 2006; Wu et al., 2013; Kristan et al., 2016a), and some have been the current "de-facto" indicators that are used widely as

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (No. 61501139) and the Natural Scientific Research Innovation Foundation in Harbin Institute of Technology, Weihai (No. 2019KYCXJJYB06)

ORCID: Wen-jing KANG, https://orcid.org/0000-0002-7779-0106; Gong-liang LIU, https://orcid.org/0000-0001-7534-4201

performance proofs in Zhang RF et al. (2017) and Sun et al. (2018). However, most current tracker evaluation systems focus mainly on the overall performance or the average value of the whole test dataset, which cannot clearly reflect the trackers' performance in specific scenarios. As modern algorithms become increasingly powerful, a more detailed evaluation system that meets the need of choosing algorithms based on specific applications is needed.

In 2000, the first IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS) was held in Grenoble, France, where the first public dataset with the PETS metric (Young and Ferryman, 2005) was proposed. In the same year, the video verification of identity (VIVID) tracking evaluation dataset (Collins et al., 2005), together with a testbed, was developed. However, at that time visual tracking was applied in only a few fields, and most of the trackers were designed specifically for surveillance systems, leading to many issues in datasets, e.g., small data size and limited types of characteristics. Thus, it is a challenge to adopt large-scale or detailed benchmarks.

The problems had persisted until 2014 when the Amsterdam Library of Ordinary Videos (ALOV++) (Smeulders et al., 2014) was proposed. Smeulders et al. (2014) performed an in-depth survey of visual-tracking algorithms and furnished a huge dataset. Videos in the dataset were divided into 13 categories, which makes detailed and specialized tracker benchmarks and analysis possible. At about the same time, the online tracking benchmark (OTB) (Wu et al., 2013, 2015) was proposed. The OTB system integrates 27 publicly available or open-source trackers, and a dataset in which the videos were labeled and categorized carefully was published. One of the measures used in a benchmark called the "center location error (CLE)" has become a classic metric for evaluating tracker performance in many tracker papers (Danelljan et al., 2014, 2015a; Henriques et al., 2015). Another famous system proposed was the first Video Object Tracking (VOT) Challenge (Kristan et al., 2013). The challenge has continued on an annual basis (Kristan et al., 2015a, 2015b, 2016b, 2017) since 2013. Considering that CLE cannot reflect the trackers' performance on dealing with the violent change of the target size, an overlap-based metric, referred to as the excepted average overlap (EAO) (Kristan et al., 2016b), was invented in the challenge to evaluate trackers. From then on, CLE and overlap have been two of the most famous and classic measures in visual-tracking evaluations.

Though CLE and overlap have both been adopted in many studies, their exact meanings and characteristics still cannot be interpreted convincingly. It is known that different indicators reflect different aspects of the tracker's performance (Čehovin et al., 2016a), yet the relationship between the indicators and different tracker features cannot be distinguished clearly. We hope to explore the differences between the two indicators in greater depth, and to find out which characteristics in videos, or "visual attributes," have the strongest influence on each indicator in the tracker, to show more explicit significance of the two indicators.

At the same time, unlike other labeling systems which simply give each frame a set of binary attribution labels, we believe that "attributes" exist in every frame, and only the intensities are different. Therefore, in this study we do not intend to provide an updated or larger dataset; instead, the existing dataset is used to split the inter-frame changes in the video into several aspects, such as change in illumination, background clutter, and length of movement. All aspects are quantified, normalized, and calculated as quantitative attributes for each sequence. Next, novel correlation- and weight-based analysis methods are used to analyze the impact of each attribute on the performance of the tracker, to find the current bottlenecks in the field.

Since most evaluation systems provide their own datasets, in this study, we use the evaluation system names, OTB or VOT, to refer to both the evaluation system and its corresponding dataset.

2 Related work

2.1 Datasets

Numerous visual tracking datasets have been published in recent years. Most of them are general-purpose datasets that contain various kinds of videos, along with a few dedicated datasets which are constructed for certain specific purposes such as drone tracking or high-frame-rate tracking. Three famous and representative datasets are used in this study:

1. The Online Tracking Benchmark (OTB) (Wu et al., 2013). In the original version, also known as OTB-50, Wu et al. (2013) constructed a dataset that contained 50 sequences with fully labeled ground-truth and attribute annotations. Another 50 sequences were appended to the original OTB-50, constructing the final well-known OTB-100 (Wu et al., 2015).

2. The Video Object Tracking (VOT) Challenge (Kristan et al., 2013). Its first dataset edition is VOT 2013 and it has been updated annually since then. The aim is to build a "sufficiently small, well-annotated, and diverse" dataset using the VOT dataset construction methodology. Every frame in each sequence is annotated by a rotated bounding box and binary attributes.

3. The Amsterdam Library of Ordinary Videos (ALOV++) (Smeulders et al., 2014). It was constructed by the University of Amsterdam with 315 image sequences. Each video is aimed at one specific condition and the difficulty among the videos varies from very low to very high.

2.2 Attribute-based evaluations

Most modern datasets for visual tracking provide image attribute annotations for evaluation, which can reflect special characteristics for specific samples. Table 1 shows some representative types of attributes used in three famous datasets. In terms of the attribute content, it is obvious that the basic structures of the attributes are almost the same, and some significant attributes (e.g., illumination change) are shared in more than one benchmark system with different names. Meanwhile, for forms of attribution, the ALOV++ dataset assigns only one attribute to each sequence, while OTB and VOT both assign multiple attributes. In particular, VOT has used a frame-wise labeling technique for its sequences since 2015. It means that frames in the same image sequence may have different attributes. Additionally, in all published datasets, the attached attributes are binary tags and do not reflect the changes in the intensities of the attributes of the sequence.

Another noticeable point is the labeling approach. Currently, the study on attribute annotation in most datasets is carried out manually. Naturally, human unreliability will introduce an inevitable bias and unrepeatability to the annotation outcome. To address this problem, Kristan et al. (2015b) proposed a semiautomatic annotation methodology for image sequences. Nevertheless, there are a few doubtful cases in the actual test, where the annotations were inconsistent with an intuitive perception of the annotation. An instance is sequence "tiger," taken from the VOT Challenge 2017 dataset (Kristan et al., 2017) (Fig. 1). Fig. 1a shows the first frame of the sequence, which is the reference state of the illumination change attribute (illum_change). The 251st frame (Fig. 1b) gives a good example for the positive illumination change state, where the object is brighter than that in the first frame. However, in the 262nd frame (Fig. 1c), the luminance of the target area is more like the reference status, not obviously brighter or darker, but with a positive "illum_change" annotation. In the 286th frame (Fig. 1d), the luminance of the target area is obviously larger than that in the reference status, and even larger than that in Fig. 1b, but annotated as a negative "illum_change."

Table 1 Attribute groups in different benchmark systems

Benchmark system	Attribute group
Amsterdam Library of Ordinary Videos (ALOV++) (Smeulders et al., 2014)	Light
	Zooming camera
	Surface cover
	Shape
	Motion smoothness
	Moving camera
	Specularity
	Transparency
	Motion coherence
	Clutter
	Low contrast
	Confusion
	Long duration
Online Tracking Benchmark (OTB) (Wu et al., 2013)	Illumination variation
	Scale variation
	Occlusion
	Deformation
	Motion blur
	Fast motion
	In-plane rotation
	Out-plane rotation
	Out of view
	Background clutter
	Low resolution
Video Object Tracking (VOT) Challenge (Kristan et al., 2015a)	Illumination change
	Size change
	Occlusion
	Motion change
	Camera motion
	Unassigned

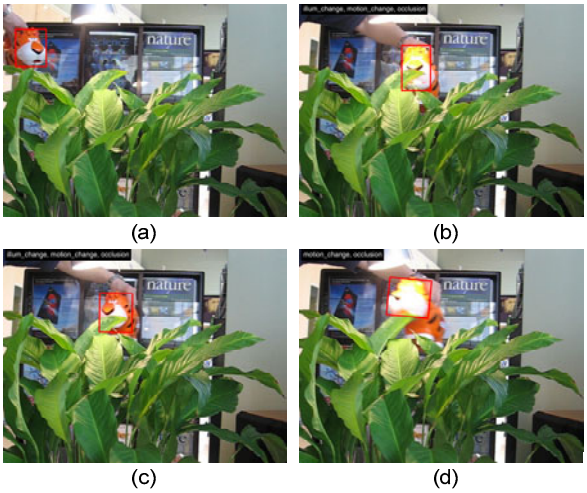

Fig. 1 An example of inaccurate attribute (illumination change) annotation from the Video Object Tracking (VOT) Challenge 2017 dataset: (a) the first frame (reference); (b) positive; (c) false positive; (d) false negative
References to color refer to the online version of this figure

3 Main work

In this study, 12 kinds of attributes were selected, quantified, and normalized. In addition, correlation- and weight-based analyses were performed to find how each attribute impacts the performance of the trackers. These two types of analyses were carried out on the raw output data taken from the VOT Challenge 2017 as an experimental test.

3.1 Definitions of attributes

Inspired by VOT 2015 (Kristan et al., 2015b), we collected the following 12 attributes for each frame:

1. Illumination change (IL);
2. Size change (SC);
3. Absolute size change (AS);
4. Target movement length (ML);
5. Background clutter (BC);
6. Camera movement (CM);
7. Blur (BL);
8. Aspect-ratio change (AR);
9. Color change (CC);
10. Deformation (DE);
11. Scene complexity (SE);
12. Absolute movement (AM).

These attributes exist in each frame in all sequences. Kristan et al. (2015a) gave some of them a brief definition. However, in Kristan et al. (2015a), the attributes were used mainly for sequence clustering; thus, no normalization was applied to the attributes, making them unsuitable for application on benchmarking across sequences directly. In this study, we improve the definitions to make them more precise by giving a formulaic description for each attribute. Meanwhile, the necessary normalization is applied to the attributes to eliminate the influence of those properties that are inherent to the sequence (e.g., resolution).

We first set up the model of a typical image sequence. As shown in Fig. 2, let F_t be the t^{th} frame of image sequence $S=\{F_t\}_{t=1,2,\ldots,N}$ with length N, F_t^{G} the target region (or ground-truth) at frame t, and $\boldsymbol{X}_t$ the center location of the target region. Meanwhile, let $\boldsymbol{q}_t=\{q_u\}_{u=1,2,\ldots,m}$ $\left(\sum_{u=1}^{m} q_u=1\right)$ be the gray-level histogram with m bins of the image at frame t, and $\boldsymbol{q}_t^{\mathrm{G}}$ is the histogram inside the target region of frame t.

IL is defined as the absolute difference between the averaged grayscale in the first frame $g(F_1^{\mathrm{G}})$ and subsequent frames $g(F_t^{\mathrm{G}})$ inside the target region. A larger value means the brightness change is more intense inside the target region. IL and $g(F_t^{\mathrm{G}})$ can be calculated as

$$\begin{cases} \mathrm{IL}(F_t)=\left|g(F_t^{\mathrm{G}})-g(F_1^{\mathrm{G}})\right|, \\ g(F_t^{\mathrm{G}})=\sum_{u=1}^{m} u\cdot q_{t,u}^{\mathrm{G}}. \end{cases} \tag{1}$$

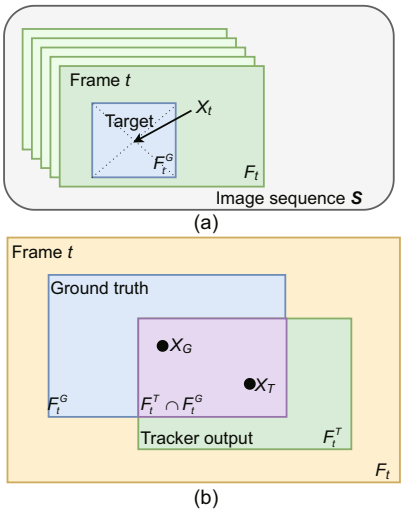

Fig. 2 Typical image sequences and frames: (a) an image sequence and its frames with the ground-truth; (b) bounding boxes of the ground-truth annotation and the prediction result of the tracker inside one frame

Sometimes, due to the rotation of the bounding boxes or inaccuracy in manual annotation, the values found directly may jitter severely and introduce considerable errors. The problem can be alleviated by first finding the illumination values of sequence $\boldsymbol{B}_t=\{g(F_t^{\mathrm{G}})\}_{t=1,2,\cdots,N}$, and then using a smoothed version $\boldsymbol{B}'_t$ in the calculation. Various smoothing methods, such as moving average and local regression, can be applied. In this study, we adopt a simple moving average filter:

$$\begin{cases}\mathrm{IL}(F_t)=\left\|\boldsymbol{B}'_t-\boldsymbol{B}'_1\right\|, \\ \boldsymbol{B}'_t=\begin{cases}\dfrac{1}{t}\displaystyle\sum_{k=1}^{t}\boldsymbol{B}_k, & 0<t\le L, \\ \dfrac{1}{2L+1}\displaystyle\sum_{k=t-L}^{t+L}\boldsymbol{B}_k, & t>L,\end{cases}\end{cases} \tag{2}$$

where $2L+1$ is the window length of the moving average filter.

One may argue that this attribute reflects only the intensity of the change in light, and cannot show the change in color of the ambient light, while some specific kinds of trackers (e.g., color-vision based trackers) may be sensitive to the color of the environmental light or the color tone of the scene. We first note that this characteristic can be reflected jointly by this attribute and another attribute CC. This will be described in detail later.

SC reflects the speed of changes in the scale of the target. In most cases, changes in this attribute are brought about by the target being near or far from the viewpoint. The SC value in the t^{th} frame is defined as the absolute difference between the size of the target region in the t^{th} frame and that in the $(t-n)^{\mathrm{th}}$ frame, where n is the interval (10 or 15 can be used). To remove the effect of different resolutions across image sequences, the absolute difference is divided by the size of the target region in the first frame for normalization:

$$\mathrm{SC}(F_t)=\begin{cases}\left|\mathrm{size}(F_t^{\mathrm{G}})-\mathrm{size}(F_1^{\mathrm{G}})\right|\big/\mathrm{size}(F_1^{\mathrm{G}}), & t\le n, \\ \left|\mathrm{size}(F_t^{\mathrm{G}})-\mathrm{size}(F_{t-n}^{\mathrm{G}})\right|\big/\mathrm{size}(F_1^{\mathrm{G}}), & t>n,\end{cases} \tag{3}$$

where larger values indicate more rapid scale changes.

Similarly, due to inaccuracy in annotation, the smoothed version of the target region size in each frame $\left\{\mathrm{size}(F_t^{G})\right\}_{t=1,2,\cdots,N}$ is adopted in the calculation.

AS reflects the intensity of scale changes of the target. It is defined as the absolute difference between the size of the target region in frame t and frame 1, divided by the size of the target area in frame 1. When $t\le n$, it has the same value as SC:

$$\mathrm{AS}(F_t)=\left|\mathrm{size}(F_t^{\mathrm{G}})-\mathrm{size}(F_1^{\mathrm{G}})\right|\big/\mathrm{size}(F_1^{\mathrm{G}}). \tag{4}$$

ML describes the moving speed of the target on the screen. It is defined as the norm of the absolute distance between the center positions of ground-truths $\boldsymbol{X}_t$ in two consecutive frames. The results are divided by the diagonal length of the image for normalization:

$$\mathrm{ML}(F_t)=\left\|\boldsymbol{X}_t-\boldsymbol{X}_{t-1}\right\|_2\big/\mathrm{diag}(F_t), \tag{5}$$

where the larger the value is, the faster the target moves.

BC is calculated from the difference between two gray-level histograms: one is taken from target region G; the other is taken from a 1.5x expanded area of target region G_l:

$$\mathrm{BC}(F_t)=\left\|\boldsymbol{q}_t^{G}-\boldsymbol{q}_t^{G_l}\right\|_2, \tag{6}$$

where a smaller value indicates that the surrounding background of the target is more like the target itself, and that the accurate edge of the target region is more difficult to determine. Fig. 3 shows a typical case where the exact border of the soldier is very difficult to tell even by a human.

CM describes the movement of the viewpoint or view angle. The scale-invariant feature transform (SIFT) (Lowe, 1999) feature point matching using random sample consensus (RANSAC) (Fischler and Bolles, 1981) is done first on the whole frame, and the attribute is defined as the average length of estimated vectors $\boldsymbol{l}$:

$$\mathrm{CM}(F_t)=\overline{\left\|\boldsymbol{l}\right\|}, \tag{7}$$

where the larger the value is, the faster the viewpoint moves.

Fig. 3 An example of the influence of background clutter on edge estimation, where the black box is the ground-truth of the target, whose border is difficult to specify exactly

Image source: https://commons.wikimedia.org/wiki/File:M249_FN_MINIMI_DM-SD-06-10452.jpg (References to color refer to the online version of this figure)

BL can be estimated as the result of the Bayes spectral entropy based measure of camera focus (Kristan et al., 2006) in the target region:

$$\mathrm{BL}(F_t)=\mathrm{BayesDCT}(F_t^{\mathrm{G}}), \tag{8}$$

where the larger the value is, the clearer and better focused the image is.

AR describes mainly the change in the shape of the target region, and reflects the rotation level of the object to some extent if the target object is rigid. The AR value in the t^{th} frame is calculated using the greater aspect ratio between the target region in frame t and the first frame, divided by the smaller one:

$$\begin{cases}\mathrm{AR}(F_t)=\dfrac{\max(r(F_1),r(F_t))}{\min(r(F_1),r(F_t))},\\ r(F_t)=\dfrac{\mathrm{height}(F_t^{\mathrm{G}})}{\mathrm{width}(F_t^{\mathrm{G}})}.\end{cases} \tag{9}$$

For the rotated bounding box, we use circumscribed rectangle aligining with x and y for ease of calculation. A larger value means that there are more intense changes in the shape of the target region.

CC is defined as the difference between the average hue values inside the target region in F_t and F_1:

$$\mathrm{CC}(F_t)=\overline{\mathrm{hue}(F_t^{\mathrm{G}})}-\overline{\mathrm{hue}(F_1^{\mathrm{G}})}, \tag{10}$$

where the larger the value is, the more intense the color changes inside the target region are.

As mentioned above, some specific kinds of trackers might be very sensitive to the color of the ambient light. Thus, it is worth considering whether any light-source color normalization or color constancy methods, such as those proposed by Funt et al. (1998) and Gao et al. (2015), should be applied to eliminate the influence. However, as far as we are concerned, in terms of competition, the top priority must be impartiality. Thus, in the test system, we tried not to make changes in particular for any kind of tracker. At the same time, another goal of this study is to provide an approach for users to select suitable trackers for specific scenarios. For environments where only the strength of the light changes, users could refer to the IL attribute; for environments where only the color of the light changes, users could refer to the CC attribute; for environments where both the strength and color of the light change, users could weigh these two attributes and make the decision accordingly.

DE reflects the absolute change in the pictures. It is defined by dividing F_t into cell units of 8×8 size and calculating the root-mean-square difference of the histogram between the corresponding cells in F_t and F_1 (Fig. 4). Let $\boldsymbol{q}_{t,i}$ be the histogram of the i^{th} 8×8 grid of F_t. We have

$$\mathrm{DE}(F_t)=\sqrt{\sum_{i=1}^{M}\left\|\boldsymbol{q}_{t,i}-\boldsymbol{q}_{1,i}\right\|_2^2}. \tag{11}$$

SE represents the absolute value of the level of randomness (entropy) in each frame:

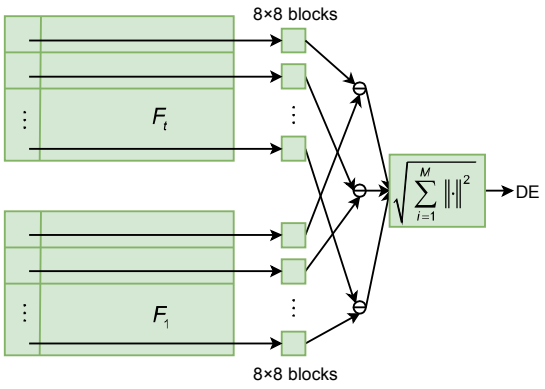

Fig. 4 The process of computing the deformation (DE) attribute for frame *t*

$$\mathrm{SE}(F_t)=\left\|\sum_{u=0}^{255} q_{t,u}\log q_{t,u}\right\|, \tag{12}$$

where a larger value indicates a higher level of randomness, or a more complicated scene.

AM reflects how far the object moves from its original location. It is computed in terms of the normalized distance between the center points of the target regions in frames 1 and t:

$$\mathrm{AM}(F_t)=\left\|\boldsymbol{X}_t-\boldsymbol{X}_1\right\|_2 \big/ \mathrm{diag}(F_t), \tag{13}$$

where the larger the value, the further the target moves from its initial position.

As mentioned before, all our attributes are frame-wise; i.e., each frame, even in the same image sequence, has its own attribute value. Of course, the overall attribute intensity of an image sequence can also be obtained by taking the average of the frame-wise attribute values. For example,

$$\overline{\mathrm{IL}(S)}=\frac{1}{N}\sum_{t=1}^{N}\mathrm{IL}(F_t). \tag{14}$$

3.2 Experimental benchmark

3.2.1 Test methodology

We propose two test methods: a correlation-based test and a weight-based test. As a holistic analysis, the correlation-based test shows how each attribute in the sequence affects the performance of the trackers by eliminating the effect of the trackers' performance baselines. It gives a clue to the problem "which aspect has the greatest impact on the participating tracking algorithms globally." The weight-based test takes both attributes and performance into consideration in calculation to make a horizontal comparison among all trackers. During this test, a ranking was obtained to show which tracker is the most robust in terms of each video attribute.

The correlation-based test can be executed in the unit of frame (Fig. 5). The final result of the correlation-based test is the average value of these coefficients on all sequences. For tracker T with performance index $\boldsymbol{I}_{T,S}=\left\{I_{T,F_n}\right\}_{n=1,2,\cdots,N}$ tested on each frame of sequence S, correlation coefficient

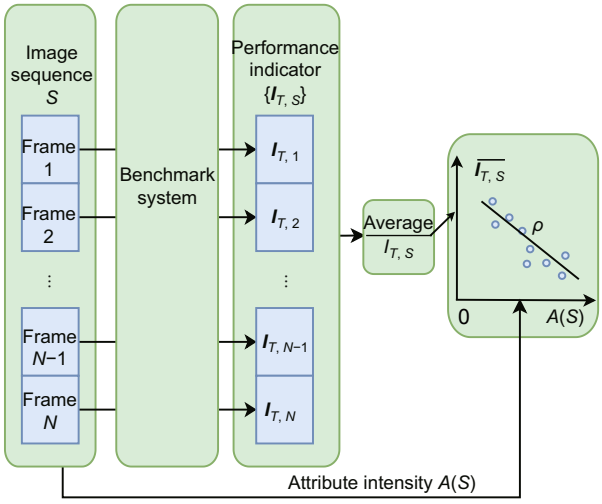

Fig. 5 Derivation of the correlation coefficient for a tracker on one image sequence

$\rho_{T,A,S}$ between $\boldsymbol{I}_{T,S}$ and the attribute value of sequence $A(S)$ is evaluated to show how much $\boldsymbol{I}_{T,S}$ is influenced by $A(S)$:

$$\rho_{T,A,S}=\mathrm{cov}(\boldsymbol{I}_{T,S},A(S))\big/(\sigma_{\boldsymbol{I}}\sigma_{A(S)}), \tag{15}$$

where $\sigma_{\boldsymbol{I}}$ and $\sigma_{A(S)}$ indicate the variance of $\boldsymbol{I}$ and $A(\cdot)$, respectively. By averaging $\rho_{T,A,S}$ for all sequences in the dataset, an overall influence score of attributes $\rho_{T,A}=\overline{\rho_{T,A,S}}$ is obtained for the tracker.

Alternatively, the score can be calculated in unit of sequence. For sequence S, we can calculate the average performance of tracker $I_T(S)=\sum_{n=1}^{N} I_{T,F_n}\big/N$ and the average for an attribute in sequence $\overline{A(S)}$, and then the correlation coefficient can be calculated as $\left\{\overline{\boldsymbol{I}_{T,S}},\overline{A(S)}\right\}$ on the whole dataset (image sequences). Calculation in unit of frame can be more accurate; however, it can also be subject to features of the performance index used in the test: the frame-wise calculation can be applied on only the frame-wise metric like overlap or CLE, but cannot be applied on sequence-wise indexes like the failure rate. Additionally, in the practical test, we found that a considerable random error can be generated if the ground-truth annotation was not smooth enough. A bias that cannot be eliminated will be introduced during the process. Therefore, we chose the sequence-wise method in this study.

The correlation-based test paves the way for longitudinal comparison. However, due to the principle of the correlation matrix, the test result is inappropriate as a baseline for comparison between different trackers. Thus, a parallel comparison method is needed. As mentioned above, the intensities of the attributes can be reflected by their corresponding quantized values. Thus, to emphasize the influence that one specific attribute has on all trackers, the basic idea of a weight-based test is to use this quantized value as a weight on the performance indicators of the trackers. If the performance indicators like CLE and overlap are comparable among trackers, the linearly weighted performance is also comparable. In this way, a clear overview of one specific attribute on all the trackers is generated.

The weight-based test is also in unit of sequence. For an attribute A, using its average value on sequence S to weigh the tracker's performance $\boldsymbol{I}_{T,S}$ and to calculate the weighted average value for all $\boldsymbol{I}_{T,S}$'s on the whole dataset, an attribute-based average performance score can be obtained. Note that, especially for BL and DE, their inverses are taken to ensure a positive correlation between the sequence difficulty and weight.

3.2.2 Test datasets

Because this study focuses mainly on tracking benchmark methodology, we do not propose any new dataset or image sequences. Instead, the "baseline" part of the VOT Challenge 2017 dataset, which can be accessed freely from https://www.votchallenge.net/vot2017/results.html, was used in the experiment. The number of frames in the dataset is over 4 000 000. It also provides an output result set ran by 51 advanced trackers on all image sequences in the dataset. The participated trackers are diverse enough to cover most categories of trackers, including depth trackers such as learning spatial-aware regressions for visual tracking (LSART) (Sun et al., 2018), efficient convolution operators (ECO) (Danelljan et al., 2017), correlation filters based trackers like good features to correlate for visual tracking (CFCF) (Gundogdu and Alatan, 2018), structured support vector machine (SVM) based trackers like Struck2011 (Hare et al., 2011), traditional mean-shift trackers like scale adaptive mean-shift (ASMS) (Vojíř et al., 2014), etc. This allows us to do a systematic analysis on different kinds of trackers easily.

3.2.3 Test measures

Several popular performance measures such as overlap, CLE, and F1 score exist in the field. Čehovin et al. (2016a) collated and compared a number of related measures. However, with more advanced trackers, especially deep learning based trackers introduced since then, the results in Čehovin et al. (2016a) may not represent the state of the art. We collated a few measures that appeared in recent publications; however, many of them are similar and some are doubtful. After deleting the measures that are dubious, and merging the rest into several categories, the seven most representative measures were selected as candidates for further consideration, as listed in the following.

1. Center location error (CLE)

CLE is one of the most easy, clearest, and most classic measures in visual tracking benchmark history (Čehovin et al., 2016a). The per-frame CLE is defined by the distance between the center locations of the tracker predicted bounding box and the ground-truth box on each frame. A sequence-wide CLE can be defined by averaging the per-frame CLE:

$$\begin{cases} \mathrm{CLE}(S^{\mathrm{G}}, S^{\mathrm{T}}) = \overline{\mathrm{CLE}_S} = \dfrac{1}{N}\sum\limits_{t=1}^{N}\mathrm{CLE}_t, \\ \mathrm{CLE}_t = \left\| \boldsymbol{X}_t^{\mathrm{G}} - \boldsymbol{X}_t^{\mathrm{T}} \right\|_2. \end{cases} \tag{16}$$

In most cases the "central locations" mentioned above can be denoted by the geometric centers of the ground-truth rectangles; however, it is better to annotate these locations according to the content of the picture. For example, in Fig. 6, as a central location, point 2 is not as relevantly appropriate as point 1.

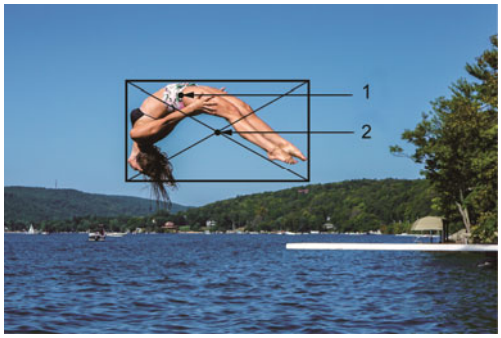

Fig. 6 The barycenter (point 1) of the target shape and the geometrical center of its bounding box (point 2) (image source: https://pxhere.com/en/photo/77043)

2. Root-mean-square error (RMSE)

RMSE is a sequence-wide measure, which means it is not available on one single frame. By taking the root-mean-square error for distances in all frames of the sequence, RMSE is defined by

$$\mathrm{RMSE}(S^{\mathrm{G}},S^{\mathrm{T}})=\overline{\mathrm{RMSE}_S}=\sqrt{\frac{1}{N}\sum_{t=1}^{N}\left\|\boldsymbol{X}_t^{\mathrm{G}}-\boldsymbol{X}_t^{\mathrm{T}}\right\|_2^2}. \quad (17)$$

3. Overlap (ϕ)

Overlap is also one of the most classic measures. It is still gaining popularity among researchers (Li AN et al., 2016; Galoogahi et al., 2017; Li SY and Yeung, 2017). The direct form of frame overlap ϕ_t of two regions (usually one is the tracker-predicted region and the other is the ground-truth) is the area of their intersection divided by the area of their union. The overlap value of an image sequence is the average value consisting of all its frame overlaps. The two types of calculations are expressed as

$$\begin{cases}\phi_t=\mathrm{size}(F_t^{\mathrm{G}}\cap F_t^{\mathrm{T}})\big/\mathrm{size}(F_t^{\mathrm{G}}\cup F_t^{\mathrm{T}}).\\ \phi(S^{\mathrm{G}},S^{\mathrm{T}})=\overline{\phi_S}=\dfrac{1}{N}\sum_{t=1}^{N}\phi_t.\end{cases} \quad (18)$$

Compared to CLE, the main difference of this measure is that it takes the scales of two regions into consideration.

4. Position-based measure (PBM) (Karasulu and Korukoglu, 2011)

This measure is adopted from its multiple-target tracking version. Let

$$T_h(t)=\frac{1}{2}\Big[\mathrm{width}(F_t^{\mathrm{T}})+\mathrm{height}(F_t^{\mathrm{T}})+\mathrm{width}(F_t^{\mathrm{G}})+\mathrm{height}(F_t^{\mathrm{G}})\Big].$$

Then, if $\phi_t>0$ (i.e., the tracker-predicted region has not drifted out completely from the target region), let $\mathrm{Dist}(t)$ be the L_1-distance between $\boldsymbol{X}_t^G$ and $\boldsymbol{X}_t^T$; otherwise, $\mathrm{Dist}(t)=T_h(t)$.

$$\begin{cases}\mathrm{PBM}(S^{\mathrm{G}},S^{\mathrm{T}})=\overline{\mathrm{PBM}_S}=\dfrac{1}{N}\sum_{t=1}^{N}\mathrm{PBM}_t,\\ \mathrm{PBM}_t=1-\dfrac{\mathrm{Dist}(t)}{T_h(t)}.\end{cases} \quad (19)$$

5. F1-score (Kwon and Lee, 2008)

Similarly, based on the intersections of two regions, F1-score is derived by

$$\begin{cases}\mathrm{Fl}(S^{\mathrm{G}},S^{\mathrm{T}})=\overline{\mathrm{Fl}_S}=\dfrac{1}{N}\sum_{t=1}^{N}\mathrm{Fl}_t,\\ \mathrm{Fl}_t=2\dfrac{p_t\cdot r_t}{p_t+r_t},\end{cases} \quad (20)$$

where

$$\begin{cases}p_t=\mathrm{size}(F_t^{\mathrm{T}}\cap F_t^{\mathrm{G}})\big/\mathrm{size}(F_t^{\mathrm{T}}),\\ r_t=\mathrm{size}(F_t^{\mathrm{T}}\cap F_t^{\mathrm{G}})\big/\mathrm{size}(F_t^{\mathrm{G}}).\end{cases}$$

6. Success percentage (P_τ)

In the test, a criterion could be set to determine whether a frame is tracked successfully or not. The ratio of the number of successfully tracked frames to the number of all frames in the sequence is defined as the success percentage (P_τ). Generally, the overlap can be used as a reference; i.e., the t^{th} frame is a successfully tracked frame if $\phi_t\geq\tau$. Then the number of successfully tracked frames is

$$\mathrm{NP}_\tau=\left|\left\{t\mid\phi_t\geq\tau,t=1,2,\cdots,N\right\}\right|,$$

where $|\cdot|$ denotes the cardinality of the set. The success percentage is

$$P_\tau(S^{\mathrm{G}},S^{\mathrm{T}})=\mathrm{NP}_\tau/N. \quad (21)$$

7. Failure rate (F_0)

In a reset-based test such as the "baseline" test in VOT, once the tracker "fails," the test tool reinitializes the tracker automatically in the next few frames and restarts the tracking process, until it fails again or reaches the end of the sequence. During this course, the number of reset times NF_0 is recorded. The ratio of NF_0 to the total number of frames in the sequence (N) is the failure rate of the tracker on the sequence:

$$F_0(S_{\mathrm{G}},S_{\mathrm{T}})=\mathrm{NF}_0/N. \quad (22)$$

In the VOT Challenge 2017, during a tracking procedure, once $\phi_t=0$ (i.e., the tracker loses its target completely), it is considered to be a "failure." Since our research is based on the results from VOT, the same failure criterion was used. In fact, this value can

also be obtained from the success percentage of zero thresholds:

$$F_0(S^{\mathrm{G}},S^{\mathrm{T}})=1-P_0(S^{\mathrm{G}},S^{\mathrm{T}}). \tag{23}$$

It seems that the failure rate and success percentage are quite similar. However, the largest difference is that the F_0 measure can be applied to only reset-based evaluation, while P_τ is available for both the reset-based and one-pass tests.

We calculated the correlation among the above seven measures on the reset-based dataset. Note that we chose τ=0.5 and τ=0.1 separately for the success percentage (P_τ) measure. The correlation matrix obtained is shown in Fig. 7. The results can obviously be clustered into three groups. The first group contains the CLE-based measures CLE and RMSE; the second group includes PBM, ϕ, $P_{0.5}$, and F_1, which are calculated based on the overlap rate; the last group includes $P_{0.1}$ and F_0. The first two groups describe the accuracy of the tracker, while the last group describes the robustness of the tracker (Čehovin et al., 2016a). We selected representative measures, CLE, overlap rate, and failure rate, respectively from the three groups for analysis.

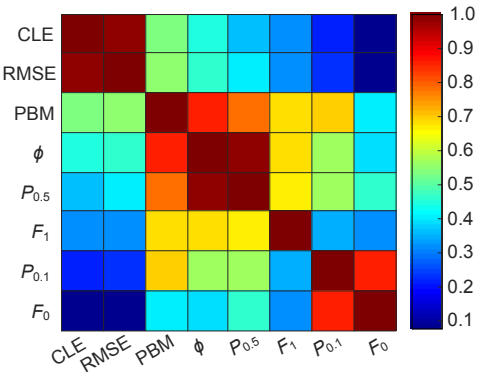

Fig. 7 Correlation matrix for seven measures

CLE: center location error; RMSE: root-mean-square error; PBM: position-based measure; ϕ: average overlap; $P_{0.5}$: success percentage when τ=0.5; F_1: F1-score; $P_{0.1}$: success percentage when τ=0.1; F_0: failure rate. References to color refer to the online version of this figure

4 Test results and analysis

4.1 Correlation-based test

We calculated the correlation among 51 trackers' overlap, CLE, F_0, and all the attributes mentioned above. For simplicity in analysis, the absolute values of results were used. See Tables 2–4 for an overview.

In Tables 2–4, it can be seen that several attributes, such as IL, CM, and CC, have little impact on the performance of the trackers. It means that a considerable number of trackers are quite insensitive to these attributes, and some trackers even handle the difficulty very well in practical tests. However, for several other trackers, e.g., RCPF (Zhang TZ et al., 2018), best structured tracker (BST) (Battistone et al., 2018), local-global tracking tracker (LGT) (Čehovin et al., 2011), and flock of trackers (FoT) (Vojíř and Matas, 2014), the impacts are still particularly prominent. For the structured SVM-based tracker, BST, in particular, its performance is highly related to quite a few attributes. These independent characteristics can be reflected clearly in the overall performance. In the VOT Challenge 2017 results, BST is ranked 42 out of the 51 trackers.

On the other hand, SC, ML, BC, BL, AR, SE, and AM are correlated highly with the trackers' performance by influencing at least one measure. Categorizing these attributes will help us understand this better.

Four attributes, SC, AS, BC, and AR, demonstrate the change mainly in the size of the target region with respect to speed, extent, boundary, and shape, respectively. Although all the participating trackers are scale-adaptive (i.e., the trackers can estimate the size of the target), all the trackers' final performance is affected tremendously by these factors. In particular, the correlation coefficient between the ECO tracker (Danelljan et al., 2017) and the SC attribute exceeds even 0.64. We can thus conclude that most trackers cannot handle the situation where the size of the object changes rapidly or complexly, and this may introduce huge performance loss. Because the wrong scale in the bounding box may take part of the background into account when using the feature extraction function for estimation in some trackers, leading to inaccuracy in template updating, the performance of the trackers may be greatly affected. Therefore, an accurate scale estimation method is what current tracking algorithms need to improve urgently. Fig. 8 is a representative example of the performance of the ECO tracker when handling a quick and intense scale change in the "Singer3" sequence. This sequence shows a clip during which the lens zooms in gradually from a long distance to a

Table 2 Impact of different attributes on average overlap

Attribute*	Average overlap**		
	Maximum	Mean	Minimum
IL	0.244 (RCPF)	0.093	0.001 (KFebT)
SC	0.383 (SPCT)	0.146	0.009 (KFebT)
AS	0.388 (SPCT)	0.134	0 (KFebT)
ML	0.269 (L1APG)	0.098	0.002 (CMT)
BC	0.425 (CHT)	0.297	0.074 (FoT)
CM	0.326 (BST)	0.066	0.001 (ATLAS)
BL	0.364 (MOSSE_CA)	0.219	0.012 (LTFLO)
AR	0.592 (UCT)	0.358	0.098 (LGT)
CC	0.391 (LGT)	0.073	0.003 (Gmdnetn)
DE	0.489 (FragTrack)	0.255	0.101 (SPCT)
SE	0.498 (FSTC)	0.279	0.128 (HMMTxD)
AM	0.470 (DACF)	0.297	0.014 (MSSA)

* See Section 3.1 for definitions of attributes
** See Appendix for references for participated trackers

Table 3 Impact of different attributes on the center location error

Attribute*	Center location error**		
	Maximum	Mean	Minimum
IL	0.235 (BST)	0.056	0.001 (DSST)
SC	0.641 (ECO)	0.342	0.013 (CMT)
AS	0.625 (ECO)	0.326	0.009 (CMT)
ML	0.410 (L1APG)	0.217	0.067 (ECO)
BC	0.277 (CMT)	0.066	0.003 (MOSSE_CA)
CM	0.385 (BST)	0.085	0.002 (LSART)
BL	0.443 (FragTrack)	0.315	0.101 (FoT)
AR	0.394 (BST)	0.096	0.003 (MEEM)
CC	0.312 (FoT)	0.120	0.014 (DSST)
DE	0.451 (BST)	0.152	0.007 (FoT)
SE	0.554 (BST)	0.174	0.051 (MSSA)
AM	0.273 (BST)	0.101	0.003 (MOSSE_CA)

* See Section 3.1 for definitions of attributes
** See Appendix for references for participated trackers

Table 4 Impact of different attributes on the failure rate

Attribute*	Failure rate**		
	Maximum	Mean	Minimum
IL	0.284 (SAPKLTF)	0.059	0 (FSTC)
SC	0.151 (ECOhc)	0.076	0.008 (SPCT)
AS	0.151 (DSST)	0.085	0 (SPCT)
ML	0.465 (DACF)	0.206	0.011 (CMT)
BC	0.416 (SRDCF)	0.155	0.005 (ECOhc)
CM	0.427 (CSR)	0.188	0 (GMD)
BL	0.365 (CFCF)	0.136	0.003 (HMMTxD)
AR	0.538 (Gmdnetn)	0.185	0.001 (gnet)
CC	0.328 (Struck2011)	0.165	0.005 (CFWCR)
DE	0.455 (FragTrack)	0.260	0.037 (ATLAS)
SE	0.498 (SRDCF)	0.267	0.027 (DACF)
AM	0.404 (CFWCR)	0.247	0.011 (Struck2011)

* See Section 3.1 for definitions of attributes
** See Appendix for references for participated trackers

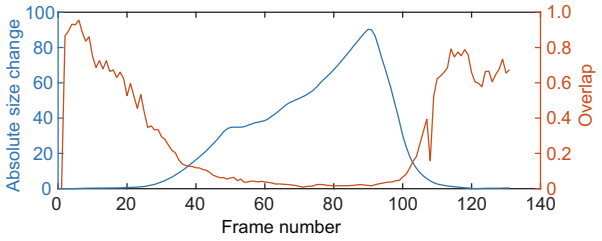

Fig. 8 Overlap and the absolute size change of the efficient convolution operators on the "Singer3" sequence
References to color refer to the online version of this figure

closeup of the singer's face and then zooms out and returns to the original place. It is easy to imagine that the size of the target, or the size of the singer in the video, changes greatly. Obviously, the tracker performance plummets so dramatically that it almost leads to a failure with the drastic change in the target size, while it increases rapidly and returns to a higher state when the target region is restored to the initial size.

Meanwhile, the ML and AM attributes indicate the tracker's performance in tracking fast-moving objects. Most trackers do not behave well in this section, or, in other words, the performance of most of the participating trackers will be affected greatly when the target moves too fast or too far from the original position. This can be confirmed in Fig. 9, which shows the average overlap performance run by the discriminative correlation filter (DACF) (Lukežič et al., 2017) against the AM attribute on all 60 sequences (which are represented by 60 points in Fig. 9). A significant negative correlation can be seen. However, the performance in this section is understandable. It is a consensus that one of the assumptions in visual tracking is that the location of the target does not change dramatically; thus, the search range of the tracker is limited under normal conditions. Meanwhile, increasing the search range will greatly reduce

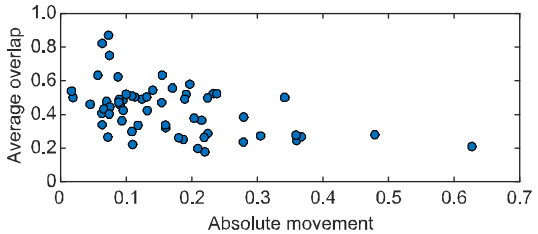

Fig. 9 The absolute movement of sequences from the Video Object Tracking (VOT) Challenge 2017 against the average overlap of the discriminative correlation filter

the speed of the tracker. This requires the tracking algorithm find a trade-off between the search range and speed.

There is another interesting thing that we can note. As mentioned above, the CM attribute has little impact on the trackers overall. Compared with ML's huge effects, we can interpret an interesting fact; i.e., in videos with a high frame rate, if we divide the intra-frame changes between two consecutive frames into two aspects, movements of the target and the background (which can be represented by the object's ML and CM, respectively), since these two kinds of changes always scale in an equal proportion when the frame rate changes, it is the decrease in movement of the target, but not the movement of the background, that has more influence on the performance of trackers.

By comparing the same attributes on different measures, more useful information can be obtained. It is believed generally that CLE cannot reflect the tracker's size-adaptive performance, because it uses only two points—one from the ground-truth and the other from the tracker-predicted region—in the calculation (Karasulu and Korukoglu, 2011; Wu et al., 2015; Čehovin et al., 2016a). This can be confirmed clearly from the BC and AR attributes, which indicate the boundary and shape of the target region, respectively, making a tremendous impact on the overlap while having little influence on CLE. Surprisingly, CLE is much more sensitive than overlap to the SC attribute. This shows that how well the tracker copes with the scale change of the target determines its ability to describe the whole target and to reflect on CLE. Therefore, the scale adaptability of the tracker is very important for improving the tracking performance. When the size of the target changes suddenly and the tracker does not have good scale adaptability, it is a common case that the tracker's predicted region R_t^{T} fully contains or is contained by target region R_t^{G} (Fig. 10). The overlap will be basically unchanged; however, the center of the tracker's predicted region is distributed randomly around the center of the ground-truth region. In this case, the measured overlap cannot reflect this kind of jitter.

In terms of robustness which can be reflected by the failure rate, it can be seen from Table 4 that no specific kind of attribute has a particular high impact on the failure rate. This shows that the failure of tracking is a complex consequence involving multiple factors and attributes, where the failure cannot be associated directly with any specific kind of intra-frame change. Because in this study we choose an overlap-based failure criterion, the correlation value for the failure rate tends to be similar to that of the overlap measure in some respects.

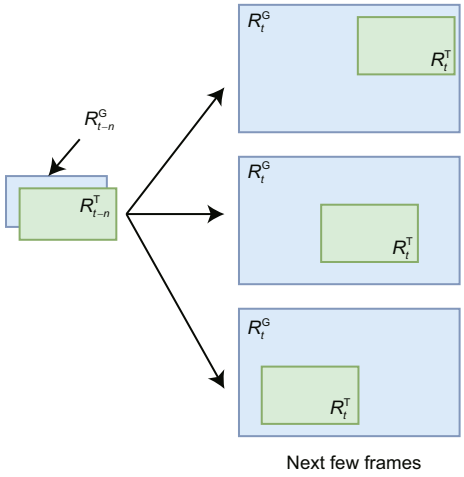

Fig. 10 An example showing the same overlap with different center location errors in three cases

4.2 Weight-based test

In Section 4.1, by averaging the correlation value of each measure, we can see that the overlap measure tends to be more affected by more attributes; thus, we chose the overlap measure for the weight-based test. In this test, we calculated the weighted performance of all trackers contained in the dataset on all sequences using the approach mentioned in Section 3. The results of the test are shown in Table 5.

The results are grouped into two categories: deep trackers and traditional trackers. Various tests (Galoogahi et al., 2017; Kristan et al., 2017) showed that deep trackers have better overall performance. In our test, deep tracker's performance was also significantly better. However, the performances of some traditional trackers were excellent on several attributes; e.g., the color mean shift-based ASMS tracker achieved the best performance in handling deformation situations in our test.

For the DE, BL, and SC attributes, deep trackers performed much better than traditional ones. Specifically, for DE, the overall performance of deep

Table 5 Performance of different trackers in Video Object Tracking (VOT) Challenge 2017 on all attributes using weight-based test methods

Tracker	Weighted performance index on the attributes											
	IL	AM	BC	CM	BL	DE	SE	CC	SC	AS	ML	AR
ATLAS*	**36.45**	42.42	**95.34**	16.12	**91.28**	6.66	75.76	8.92	10.70	39.72	70.06	24.49
CCOT*	16.77	21.59	46.51	2.65	25.47	27.55	**94.31**	**52.69**	16.35	36.51	24.46	20.47
CFWCR*	27.43	17.04	46.51	3.06	49.22	**50.61**	89.04	26.66	10.05	64.26	31.44	55.23
CRT*	12.98	15.53	41.86	14.14	27.18	6.67	68.37	10.71	1.65	53.63	32.86	41.88
DLST*	5.39	**67.42**	46.51	0.78	43.06	34.42	82.40	26.40	0.95	51.49	56.24	39.20
ECO*	9.89	3.40	51.16	3.18	35.38	13.59	90.81	30.79	3.75	35.66	9.04	19.38
FSTC*	5.71	26.51	**74.41**	16.02	34.66	14.94	**90.83**	31.44	**82.07**	39.37	30.78	24.01
GMD*	**100.00**	**100.00**	55.81	8.05	36.42	18.12	76.35	31.67	8.79	45.22	27.33	31.37
Gmdnetn*	25.45	13.25	41.86	**32.30**	**50.74**	11.98	**93.38**	**57.29**	3.37	39.51	**94.19**	24.16
gnet*	2.17	**67.80**	69.76	1.31	45.46	22.05	83.48	16.47	0.78	47.43	14.61	34.13
LSART*	12.03	50.75	51.16	3.55	41.78	**64.34**	**100.00**	49.87	2.59	57.06	58.60	46.17
MCCT*	12.00	**69.31**	44.18	6.98	**100.00**	19.33	90.56	30.35	0.22	55.99	32.50	44.89
MCPF*	1.50	29.16	72.09	21.81	41.46	7.36	78.82	5.83	9.01	66.04	**100.00**	57.42
RCPF*	21.43	13.25	53.48	4.33	33.26	3.47	64.70	40.63	1.97	**79.74**	38.11	**74.57**
SiamDCF*	13.72	37.12	32.55	3.87	26.54	6.11	66.22	18.87	0.39	44.36	24.68	30.30
SiamFC*	11.02	62.87	27.90	9.87	33.14	8.67	80.38	22.36	2.53	52.49	27.54	40.43
Mean*	19.62	39.77	53.48	9.25	44.70	19.74	82.84	28.80	9.70	50.49	42.03	38.01
Mean	12.62	24.24	46.51	10.92	32.82	13.00	66.62	23.30	9.20	47.43	33.66	34.11
ANT	1.03	35.60	69.76	1.30	30.50	**68.55**	87.61	5.50	0.65	**86.23**	38.14	**82.78**
ASMS	2.86	38.25	72.09	1.34	31.06	**100.00**	**91.63**	7.08	0.13	**76.17**	36.15	**70.11**
BST	3.43	6.43	18.60	0.13	20.31	**34.80**	48.35	28.21	0.46	20.18	23.18	0.00
CFCF	16.20	23.10	55.81	4.83	34.46	5.05	74.77	11.60	0.28	46.71	24.70	33.24
CGS	16.36	24.62	48.83	1.90	31.14	4.23	58.61	23.15	1.32	44.50	30.69	30.50
CHT	0.96	10.98	39.53	3.12	40.30	12.97	69.34	23.17	8.77	48.57	34.13	35.54
CMT	0.88	8.71	16.27	4.53	11.71	3.61	31.77	10.76	0.35	24.75	12.13	5.75
CSR	13.46	7.95	34.88	11.65	40.46	5.04	70.64	39.16	0.66	51.21	31.85	38.83
CSRF	1.39	48.10	**83.72**	**24.17**	**61.01**	24.10	87.31	29.44	11.71	57.84	59.25	47.22
DACF	29.59	30.30	**100.00**	10.65	42.22	6.01	66.24	**51.17**	1.80	52.06	69.05	39.94
DPRF	**46.00**	10.60	20.93	**100.00**	21.79	7.21	68.26	40.16	**100.00**	35.59	52.14	19.28
DPT	**30.10**	27.27	37.20	14.91	38.54	5.44	69.53	33.26	7.19	34.37	67.47	17.77
DSST	4.50	10.22	32.55	**22.88**	36.58	6.39	59.01	1.99	0.72	28.81	**79.29**	10.76
ECOhc	18.30	**88.63**	46.51	0.62	33.90	21.35	89.67	**52.78**	4.57	58.06	50.02	47.48
FoT	5.99	27.65	30.23	1.03	27.54	4.93	46.67	8.65	1.12	37.66	37.37	21.87
FragTrack	27.32	26.51	41.86	7.44	18.91	7.75	58.34	11.19	1.99	34.37	14.10	17.75
HMMTxD	21.76	29.16	53.48	9.76	27.58	12.70	85.27	**100.00**	8.67	61.76	19.32	52.13
IVT	16.24	8.33	34.88	1.90	30.90	4.51	56.04	4.63	2.17	49.64	27.46	36.92
KCF	13.43	34.47	46.51	3.00	28.50	4.66	61.06	3.81	8.42	54.77	19.61	43.31
KFebT	9.40	17.42	55.81	2.74	33.82	6.69	64.03	11.09	2.26	46.07	16.43	32.43
L1APG	4.45	17.42	39.53	1.94	36.34	5.41	52.63	10.83	1.38	40.15	9.52	25.04
LDES	4.56	32.19	34.88	14.51	21.79	3.74	54.90	37.15	14.72	41.08	**94.51**	26.15
LGT	0.34	9.09	58.14	1.36	**56.49**	9.98	70.81	27.77	0.17	39.08	5.42	23.66
LTFLO	5.17	13.25	18.60	2.55	23.43	4.37	50.96	13.43	0.19	29.45	9.68	11.62
MEEM	4.83	29.54	48.83	2.54	21.99	4.29	61.20	20.08	1.92	39.94	15.21	24.71
MIL	4.28	13.25	25.58	2.85	28.66	3.66	50.33	10.50	1.05	42.58	22.33	28.06
MOSSE_CA	**43.81**	34.84	37.20	11.40	24.27	5.38	60.32	13.31	0.82	32.38	15.95	15.23
MSSA	20.77	15.53	48.83	**54.16**	35.74	12.81	79.28	15.32	0.12	34.45	13.87	17.86
SAPKLTF	4.21	12.50	34.88	15.26	39.06	11.98	80.99	21.77	**75.07**	**100.00**	29.13	**100.00**
SPCT	20.74	51.51	27.90	18.68	35.70	9.71	79.22	42.59	**35.17**	**69.04**	46.85	**61.17**
SRDCF	3.66	22.34	55.81	1.46	42.62	8.58	59.06	13.87	0.05	40.30	**74.12**	25.23
SSKCF	11.21	15.90	53.48	3.44	37.70	7.67	73.40	5.30	0.91	38.44	30.64	22.83
Staple	25.29	17.80	**79.07**	8.84	32.50	10.57	87.65	32.47	2.33	53.92	22.15	42.30
struck2011	0.91	26.89	32.55	0.86	24.55	6.31	58.57	29.79	**23.76**	44.86	12.14	30.93
UCT	8.10	15.53	51.16	14.62	46.54	4.50	67.91	24.45	1.23	64.47	34.10	55.51

Values are normalized linearly between 0 and 100. See Table A1 for references for participated trackers. See Section 3.1 for definitions of attributes. Bold numbers show the top five trackers in the attribute. * indicates that the tracker uses deep features. The "Mean*" row gives the average performance of all deep trackers, while the "Mean" row shows the average performance of traditional trackers

trackers was 50% higher than that of the traditional trackers. Because these three attributes all demonstrate the trackers' ability in feature extraction, the greatest advantage of the deep tracker is their high accuracy and anti-jamming features of deep features, compared with the traditional feature extraction algorithms such as Haar feature (Hare et al., 2016) or histogram of gradients (HOG) (Dalal and Triggs, 2005). On the other hand, several drawbacks existed in deep trackers, compared with traditional trackers. For example, when dealing with camera movement, although the overall impact of this attribute on trackers is relatively small, deep trackers are not so good at handling complex camera movements and background shifts, compared with traditional trackers, and the average performance of deep trackers on this attribute is not as good as that of traditional trackers.

Nevertheless, for the scale-related attributes, the performance gap between deep and traditional trackers is not large. Despite the higher overall performance basis of the deep trackers, their overall performance is only slightly higher for the SC, BC, and AR attributes. This also shows that the problem of scale inaccuracy exists in all tracking algorithms.

5 Conclusions and discussion

This paper first clarified the confusing definitions in image sequence attributes by determining formulas for 12 well-known kinds of attributes. Two test methodologies, correlation- and weight-based test methods were proposed to analyze the detailed characteristics of the trackers. Several facts were discovered in our experimental test. First, some conditions in the visual tracking area such as changes in color and llumination have little influence on most of the current advanced trackers; in contrast, some other conditions, especially those with scale-related attributes, may impact the trackers' performance greatly, and have not been resolved sufficiently even in modern advanced deep trackers. Therefore, we suggest that the current bottleneck in visual tracking lies in the scale-adaptive part of the process. The impact of a target's moving speed cannot be ignored, because there do exist relationships between CLE and the target scale.

We compared the features of three popular test measures on different attributes. A notable point is that CLE can better indicate the influence of a rapid change in the size of the target on the tracker's performance. This also means that we cannot rely on simple theoretical conjectures or extreme cases to conclude the characteristics of a test measure or a methodology in visual tracking benchmarking. As a complex system, every measure may be influenced by numerous factors. We should carry out a performance test based on practical results.

This study also addressed the problem that with so many kinds of trackers being published, it is difficult to find the most suitable one for some specific applications. With the two proposed benchmark methods and the experimental test, we can give a clear suggestion on selecting appropriate trackers for different application scenarios.

Contributors

Gong-liang LIU designed the research. Chang LIU processed the data. Chang LIU and Wen-jing KANG drafted the manuscript. Wen-jing KANG and Gong-liang LIU revised and edited the final version.

Compliance with ethics guidelines

Wen-jing KANG, Chang LIU, and Gong-liang LIU declare that they have no conflict of interest.

References

Babenko B, Yang MH, Belongie S, 2011. Robust object tracking with online multiple instance learning. *IEEE Trans Patt Anal Mach Intell*, 33(8):1619-1632. https://doi.org/10.1109/TPAMI.2010.226

Bao CL, Wu Y, Ling HB, et al., 2012. Real time robust L1 tracker using accelerated proximal gradient approach. IEEE Conf on Computer Vision and Pattern Recognition, p.1830-1837. https://doi.org/10.1109/CVPR.2012.6247881

Battistone F, Petrosino A, Santopietro V, 2018. Watch out: embedded video tracking with BST for unmanned aerial vehicles. *J Signal Process Syst*, 90(6):891-900. https://doi.org/10.1007/s11265-017-1279-x

Bertinetto L, Valmadre J, Golodetz S, et al., 2016. Staple: complementary learners for real-time tracking. IEEE Conf on Computer Vision and Pattern Recognition, p.1401-1409. https://doi.org/10.1109/CVPR.2016.156

Čehovin L, Kristan M, Leonardis A, 2011. An adaptive coupled-layer visual model for robust visual tracking. IEEE Int Conf on Computer Vision, p.1363-1370. https://doi.org/10.1109/ICCV.2011.6126390

Čehovin L, Leonardis A, Kristan M, 2016a. Visual object tracking performance measures revisited. *IEEE Trans Image Process*, 25(3):1261-1274. https://doi.org/10.1109/TIP.2016.2520370

Čehovin L, Leonardis A, Kristan M, 2016b. Robust visual tracking using template anchors. IEEE Winter Conf on Applications of Computer Vision, p.1-8. https://doi.org/10.1109/WACV.2016.7477570

Chen K, Tao WB, 2018. Convolutional regression for visual tracking. *IEEE Trans Image Process*, 27(7):3611-3620. https://doi.org/10.1109/TIP.2018.2819362

Collins R, Zhou XH, Teh SK, 2005. An open source tracking testbed and evaluation web site. Proc IEEE Int Workshop on Performance Evaluation of Tracking and Surveillance, p.17-24.

Dalal N, Triggs B, 2005. Histograms of oriented gradients for human detection. IEEE Conf on Computer Vision and Pattern Recognition, p.886-893. https://doi.org/ 10.1109/CVPR.2005.177

Danelljan M, Häger G, Khan F, et al., 2014. Accurate scale estimation for robust visual tracking. Proc British Machine Vision Conf, p.1-11. https://doi.org/10.5244/C.28.65

Danelljan M, Häger G, Khan FS, et al., 2015a. Convolutional features for correlation filter based visual tracking. Proc IEEE Int Conf on Computer Vision Workshops, p.621-629. https://doi.org/10.1109/ICCVW.2015.84

Danelljan M, Häger G, Khan FS, et al., 2015b. Learning spatially regularized correlation filters for visual tracking. IEEE Int Conf on Computer Vision, p.4310-4318. https://doi.org/10.1109/ICCV.2015.490

Danelljan M, Robinson A, Khan FS, et al., 2016. Beyond correlation filters: learning continuous convolution operators for visual tracking. 14th European Conf on Computer Vision, p.472-488. https://doi.org/10.1007/978-3-319-46454-1_29

Danelljan M, Bhat G, Khan FS, et al., 2017. ECO: efficient convolution operators for tracking. IEEE Conf on Computer Vision and Pattern Recognition, p.6931-6939. https://doi.org/10.1109/CVPR.2017.733

Fischler MA, Bolles RC, 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun ACM*, 24(6):381-395. https://doi.org/10.1145/358669.358692

Funt B, Barnard K, Martin L, 1998. Is machine colour constancy good enough? 5th European Conf on Computer Vision, p.445-459. https://doi.org/10.1007/BFb0055683

Galoogahi HK, Fagg A, Huang C, et al., 2017. Need for speed: a benchmark for higher frame rate object tracking. IEEE Int Conf on Computer Vision, p.1134-1143. https://doi.org/10.1109/ICCV.2017.128

Gao SB, Yang KF, Li CY, et al., 2015. Color constancy using double-opponency. *IEEE Trans Patt Anal Mach Intell*, 37(10):1973-1985. https://doi.org/10.1109/TPAMI.2015.2396053

Gundogdu E, Alatan AA, 2018. Good features to correlate for visual tracking. *IEEE Trans Image Process*, 27(5):2526-2540. https://doi.org/10.1109/TIP.2018.2806280

Hare S, Saffari A, Torr PHS, 2011. Struck: structured output tracking with kernels. Int Conf on Computer Vision, p.263-270. https://doi.org/10.1109/ICCV.2011.6126251

Hare S, Golodetz S, Saffari A, et al., 2016. Struck: structured output tracking with kernels. *IEEE Trans Patt Anal Mach Intell*, 38(10):2096-2109. https://doi.org/10.1109/TPAMI.2015.2509974

He Z, Fan Y, Zhuang J, et al., 2017. Correlation filters with weighted convolution responses. IEEE Int Conf on Computer Vision Workshop, p.1992-2000.

Henriques JF, Caseiro R, Martins P, et al., 2015. High-speed tracking with kernelized correlation filters. *IEEE Trans Patt Anal Mach Intell*, 37(3):583-596. https://doi.org/10.1109/TPAMI.2014.2345390

Karasulu B, Korukoglu S, 2011. A software for performance evaluation and comparison of people detection and tracking methods in video processing. *Multim Tools Appl*, 55(3):677-723. https://doi.org/10.1007/s11042-010-0591-2

Kristan M, Perš J, Perše M, et al., 2006. A Bayes-spectral-entropy-based measure of camera focus using a discrete cosine transform. *Patt Recogn Lett*, 27(13):1431-1439. https://doi.org/10.1016/j.patrec.2006.01.016

Kristan M, Pflugfelder R, Leonardis A, et al., 2013. The Visual Object Tracking VOT2013 Challenge results. IEEE Int Conf on Computer Vision Workshops, p.98-111. https://doi.org/10.1109/ICCVW.2013.20

Kristan M, Pflugfelder R, Leonardis A, et al., 2015a. The Visual Object Tracking VOT2014 Challenge results. European Conf on Computer Vision, p.191-217. https://doi.org/10.1007/978-3-319-16181-5_14

Kristan M, Matas J, Leonardis A, et al., 2015b. The Visual Object Tracking VOT2015 Challenge results. IEEE Int Conf on Computer Vision Workshop, p.564-586. https://doi.org/10.1109/ICCVW.2015.79

Kristan M, Matas J, Leonardis A, et al., 2016a. A novel performance evaluation methodology for single-target trackers. *IEEE Trans Patt Anal Mach Intell*, 38(11):2137-2155. https://doi.org/10.1109/TPAMI.2016.2516982

Kristan M, Leonardis A, Matas J, et al., 2016b. The Visual Object Tracking VOT2016 Challenge results. European Conf on Computer Vision, p.777-823. https://doi.org/10.1007/978-3-319-48881-3_54

Kristan M, Leonardis A, Matas J, et al., 2017. The Visual Object Tracking VOT2017 Challenge results. IEEE Int Conf on Computer Vision Workshops, p.1949-1972. https://doi.org/10.1109/ICCVW.2017.230

Kwon J, Lee KM, 2008. Tracking of abrupt motion using Wang-Landau Monte Carlo estimation. 10th European Conf on Computer Vision, p.387-400. https://doi.org/10.1007/978-3-540-88682-2_30

Li AN, Lin M, Wu Y, et al., 2016. NUS-PRO: a new visual tracking challenge. *IEEE Trans Patt Anal Mach Intell*, 38(2):335-349. https://doi.org/10.1109/TPAMI.2015.2417577

Li B, Wu W, Wang Q, et al., 2019. SiamRPN++: evolution of Siamese visual tracking with very deep networks. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.4282-4291.

Li SY, Yeung DY, 2017. Visual object tracking for unmanned aerial vehicles: a benchmark and new motion models. Proc 31st AAAI Conf on Artificial Intelligence, p.4140-4146.

Lowe DG, 1999. Object recognition from local scale-invariant features. Proc 7th IEEE Int Conf on Computer Vision, p.150-1157. https://doi.org/10.1109/ICCV.1999.790410

Lukežič A, Vojír T, Zajc LC, et al., 2017. Discriminative correlation filter with channel and spatial reliability. IEEE Conf on Computer Vision and Pattern Recognition, p.4847-4856. https://doi.org/10.1109/CVPR.2017.515

Lukežič A, Zajc LČ, Kristan M, 2018. Deformable parts correlation filters for robust visual tracking. *IEEE Trans Cybern*, 48(6):1849-1861. https://doi.org/10.1109/TCYB.2017.2716101

Mathew R, Hiremath SS, 2018. Control of velocity-constrained stepper motor-driven Hilare robot for waypoint navigation. *Engineering*, 4(4):491-499. https://doi.org/10.1016/j.eng.2018.07.013

Mocanu B, Tapu R, Zaharia T, 2017. Single object tracking using offline trained deep regression networks. 7th Int Conf on Image Processing Theory, Tools and Applications, p.1-6. https://doi.org/10.1109/IPTA.2017.8310091

Nebehay G, Pflugfelder R, 2015. Clustering of static-adaptive correspondences for deformable object tracking. IEEE Conf on Computer Vision and Pattern Recognition, p.2784-2791. https://doi.org/10.1109/CVPR.2015.7298895

Ross DA, Lim J, Lin RS, et al., 2008. Incremental learning for robust visual tracking. *Int J Comput Vis*, 77(1-3):125-141. https://doi.org/10.1007/s11263-007-0075-7

Senna P, Drummond IN, Bastos GS, 2017. Real-time ensemble-based tracker with Kalman filter. 30th SIBGRAPI Conf on Graphics, Patterns and Images, p.338-344. https://doi.org/10.1109/SIBGRAPI.2017.51

Smeulders AWM, Chu DM, Cucchiara R, et al., 2014. Visual tracking: an experimental survey. *IEEE Trans Patt Anal Mach Intell*, 36(7):1442-1468. https://doi.org/10.1109/TPAMI.2013.230

Sun C, Wang D, Lu HC, et al., 2018. Learning spatial-aware regressions for visual tracking. IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.8962-8970. https://doi.org/10.1109/CVPR.2018.00934

Tran A, Manzanera A, 2017. Mixing Hough and color histogram models for accurate real-time object tracking. 17th Int Conf on Computer Analysis of Images and Patterns, p.43-54. https://doi.org/10.1007/978-3-319-64689-3_4

Valmadre J, Bertinetto L, Henriques J, et al., 2017. End-to-end representation learning for correlation filter based tracking. IEEE Conf on Computer Vision and Pattern Recognition, p.5000-5008. https://doi.org/10.1109/CVPR.2017.531

Vojíř T, Matas J, 2014. The enhanced flock of trackers. In: Cipolla R, Battiato S, Farinella GM (Eds.), Registration and Recognition in Images and Videos. Springer, Berlin, p.113-136. https://doi.org/10.1007/978-3-642-44907-9_6

Vojíř T, Noskova J, Matas J, 2014. Robust scale-adaptive mean-shift for tracking. *Patt Recogn Lett*, 49:250-258. https://doi.org/10.1016/j.patrec.2014.03.025

Wu Y, Lim J, Yang MH, 2013. Online object tracking: a benchmark. IEEE Conf on Computer Vision and Pattern Recognition, p.2411-2418. https://doi.org/10.1109/CVPR.2013.312

Wu Y, Lim J, Yang MH, 2015. Object tracking benchmark. *IEEE Trans Patt Anal Mach Intell*, 37(9):1834-1848. https://doi.org/10.1109/TPAMI.2014.2388226

Yang LX, Liu RS, Zhang D, et al., 2017. Deep location-specific tracking. Proc 25th ACM Int Conf on Multimedia, p.1309-1317. https://doi.org/10.1145/3123266.3123381

Yilmaz A, Javed O, Shah M, 2006. Object tracking: a survey. *ACM Comput Surv*, 38(4):13. https://doi.org/10.1145/1177352.1177355

Young DP, Ferryman JM, 2005. PETS metrics: on-line performance evaluation service. IEEE Int Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, p.317-324. https://doi.org/10.1109/VSPETS.2005.1570931

Zhang JC, Peng YX, 2019. Object-aware aggregation with bidirectional temporal graph for video captioning. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.8327-8336.

Zhang JM, Ma SG, Sclaroff S, 2014. MEEM: robust tracking via multiple experts using entropy minimization. 13th European Conf on Computer Vision, p.188-203. https://doi.org/10.1007/978-3-319-10599-4_13

Zhang RF, Deng T, Wang GH, et al., 2017. A robust object tracking framework based on a reliable point assignment algorithm. *Front Inform Technol Electron Eng*, 18(4):545-558. https://doi.org/10.1631/FITEE.1601464

Zhang TZ, Liu S, Xu CS, et al., 2018. Correlation particle filter for visual tracking. *IEEE Trans Image Process*, 27(6):2676-2687. https://doi.org/10.1109/TIP.2017.2781304

Zuo WM, Wu XH, Lin L, et al., 2019. Learning support correlation filters for visual tracking. *IEEE Trans Patt Anal Mach Intell*, 41(5):1158-1172. https://doi.org/10.1109/TPAMI.2018.2829180

Appendix: References for participated trackers

References for participated trackers are shown in Table A1.

Table A1 Participated tracker list

Tracker	Full name	Reference
ATLAS	Adaptive single object tracking using offline learned motion and visual similar patterns	Mocanu et al., 2017
CCOT	Continuous convolution operator tracker	Danelljan et al., 2016
CFWCR	Correlation filters with weighted convolution responses	He et al., 2017
CRT	Convolutional regression for visual tracking	Chen and Tao, 2018
DLST	Deep location-specific tracking	Yang et al., 2017
ECO	Efficient convolution operators	Danelljan et al., 2017
FSTC	–	–
GMD	GOTURN MDNet tracker	–
Gmdnetn	Guided MDNet-N	–
gnet	gNetTracker	–
LSART	Learning spatial-aware regressions for visual tracking	Sun et al., 2018
MCCT	Multi-cue correlation tracker	–
MCPF	Multi-task correlation particle filter	Zhang TZ et al., 2018
RCPF	Robust correlation particle filter	Zhang TZ et al., 2018
SiamDCF	–	–
SiamFC	Fully-convolutional Siamese network	Valmadre et al., 2017
ANT	Anchored tracking	Čehovin et al., 2016b
ASMS	Scale adaptive mean-shift	Vojíř et al., 2014
BST	Best structured tracker	Battistone et al., 2018
CFCF	Convolutional features for correlation filters	Gundogdu and Alatan, 2018
CGS	Constrained graph seeking based tracker	–
CHT	ColorHough tracker	Tran and Manzanera, 2017
CMT	Consensus based matching and tracking	Nebehay and Pflugfelder, 2015
CSR (CSRDCF)	–	Lukežič et al., 2017
CSRF (CSRDCFf)	–	Lukežič et al., 2017
DACF (CSRDCF++)	Discriminative correlation filter with channel and spatial reliability	Lukežič et al., 2017
DPRF	Correlation-based visual tracking via dynamic part regressors fusion	–
DPT	Deformable part correlation filter tracker	Lukežič et al., 2018
DSST	Discriminative scale space tracker	Danelljan et al., 2014
ECOhc	Efficient convolution operator tracker—hand crafted	–
FoT	Flock of trackers	Vojíř and Matas, 2014
FragTrack	Robust fragments based tracking using the integral histogram	–
HMMTxD	Online adaptive hidden Markov model for multi-tracker fusion	–
IVT	Incremental learning for robust visual tracking	Ross et al., 2008
KCF	Kernelized correlation filter	Henriques et al., 2015
KFebT	Kalman filter ensemble-based tracker	Senna et al., 2017
L1APG	–	Bao et al., 2012
LDES	Large displacement estimation of similarity transformation on visual object tracking	–
LGT	Local-global tracking tracker	Čehovin et al., 2011
LTFLO	Long-term featureless object tracker	–
MEEM	Multiple experts using entropy minimization	Zhang JM et al., 2014
MIL	Multiple instance learning tracker	Babenko et al., 2011
MOSSE_CA	–	–
MSSA	–	–
SAPKLTF	Scale adaptive point-based Kanade Lukas Tomasi color-filter	–
SPCT	Spatial pyramid context-aware tracker	–
SRDCF	Spatially regularized discriminative correlation filter tracker	Danelljan et al., 2015b
SSKCF	SumShift tracker with kernelized correlation filter	–
Staple	Sum of template and pixel-wise LEarners	Bertinetto et al., 2016
struck2011	Struck: structured output tracking with kernels	Hare et al., 2011
UCT	Unified convolutional tracker	–

Absent information can be found in the VOT website (https://votchallenge.net)