Machine Intelligence Research
www.mi-research.net

19(6), December 2022, 512-530
DOI: 10.1007/s11633-022-1344-1

Long-term Visual Tracking: Review and Experimental Comparison

Chang Liu[1] Xiao-Fan Chen[1] Chun-Juan Bo[1,2] Dong Wang[1]

[1]School of Information and Communication Engineering, Dalian University of Technology, Dalian 116024, China

[2]Dalian Minzu University, Dalian 116600, China

Abstract: As a fundamental task in computer vision, visual object tracking has received much attention in recent years. Most studies focus on short-term visual tracking which addresses shorter videos and always-visible targets. However, long-term visual tracking is much closer to practical applications with more complicated challenges. There exists a longer duration such as minute-level or even hour-level in the long-term tracking task, and the task also needs to handle more frequent target disappearance and reappearance. In this paper, we provide a thorough review of long-term tracking, summarizing long-term tracking algorithms from two perspectives: framework architectures and utilization of intermediate tracking results. Then we provide a detailed description of existing benchmarks and corresponding evaluation protocols. Furthermore, we conduct extensive experiments and analyse the performance of trackers on six benchmarks: VOTLT2018, VOTLT2019 (2020/2021), OxUvA, LaSOT, TLP and the long-term subset of VTUAV-V. Finally, we discuss the future prospects from multiple perspectives, including algorithm design and benchmark construction. To our knowledge, this is the first comprehensive survey for long-term visual object tracking. The relevant content is available at https://github.com/wangdong-dut/Long-term-Visual-Tracking.

Keywords: Visual object tracking, long-term tracking, short-term tracking, re-detection, online update.

Citation: C. Liu, X. F. Chen, C. J. Bo, D. Wang. Long-term visual tracking: Review and experimental comparison. *Machine Intelligence Research*, vol.19, no.6, pp.512–530, 2022. http://doi.org/10.1007/s11633-022-1344-1

1 Introduction

Visual object tracking is a fundamental and essential task in computer vision, and it has many practical applications, such as smart surveillance and autonomous driving and so on. Many attempts and efforts have been carried out in recent decades. Benefiting from the development of deep learning, the visual tracking field has developed quickly and achieved remarkable success. However, most existing tracking algorithms and benchmarks focus on short-term tracking, which effectively deals with the appearance and motion changes of an always visible target in a short period of time, typically 20–30 seconds. Relatively less attention has been paid to long-term tracking.

The long-term tracking task aims at tracking the specific target in videos with minute-level or even hour-level duration, which is closer to practical application. The target can suffer more sophisticated and severe challenges than in short-term tracking. Besides, the task needs to handle frequent target disappearance and reappearance in tracking scenes due to out of view or full occlusion. The re-detection ability is essential. Several recent studies have shown that short-term trackers perform poorly on very long sequences[1–3]. Short-term trackers are more likely to drift and fail in long-term scenes due to template contamination, localization error accumulation over a long time, and lack the re-detection ability to tackle the target disappearing issue. Fig. 1 visualizes some representative challenging scenes in long-term tracking. In the first row, the target disappears from the bottom of view and reappears from the top-left. In the second and third rows, the target is fully occluded by the background and reappears after occlusion from another region of view. In the fourth row, the target suffers huge appearance variations due to the changes in the angle and distances of observation.

The long-term tracking field is still on the initial step of study. Kalal et al.[4] proposed the earlier framework of "tracking-learning-detection" with 10 sequences for evaluation in 2011. Because of a dearth of mature datasets, there are few works on tracking focusing on the long-term property. Until 2016, a larger long-term dataset UAV20L[1] was proposed. Since then, another three benchmarks including OxUvA[5], LTB[3, 6] and TLP[2] have been presented. The visual object tracking (VOT) competition also introduced the long-term tracking challenge

Review
Manuscript received January 27, 2022; accepted June 6, 2022; published online November 7, 2022
Recommended by Associate Editor Hui-Yu Zhou
© Institute of Automation, Chinese Academy of Sciences and Springer-Verlag GmbH Germany, part of Springer Nature 2022

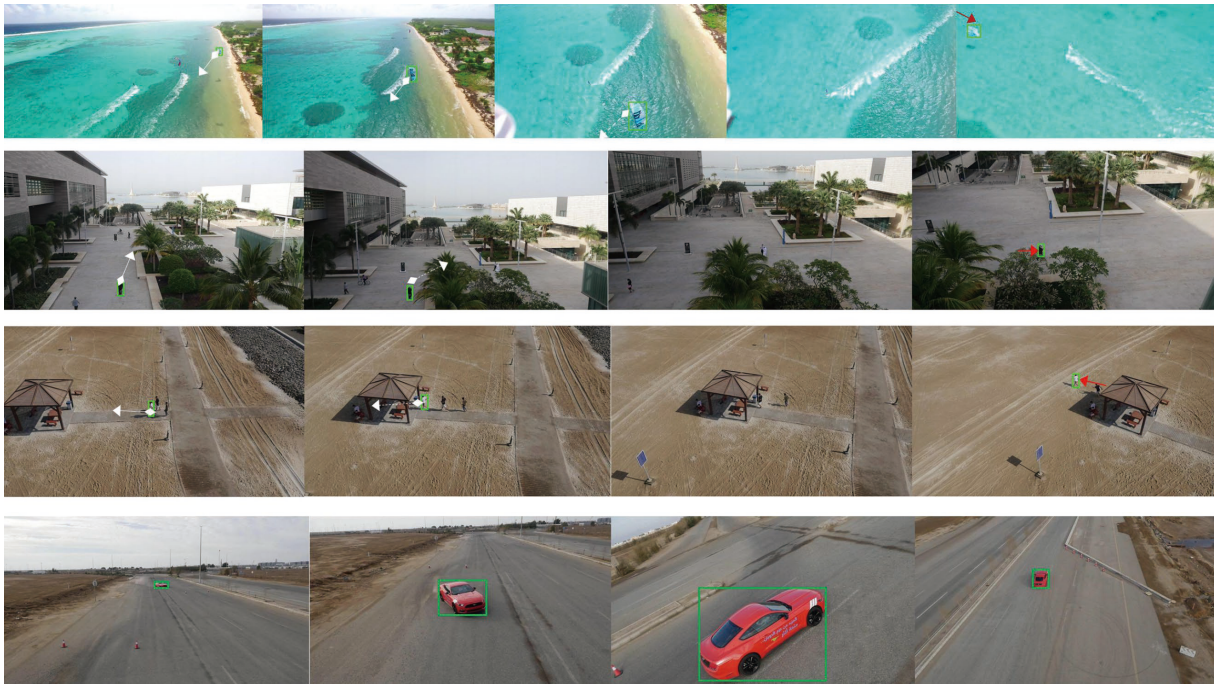

Fig. 1 Visualization of some representative challenging scenes in long-term tracking. The tracked target is in the green bounding box. The white arrow indicates the approximate direction of target's movement, while the dotted line denotes the direction of the target about to disappear. The red arrow indicates the direction of reappearance. Better viewed in color and in zoom.

from 2018, which inherits the dataset and evaluation protocol from [3]. With the increasing richness of the long-term benchmarks and the growing concern on the long-term tracking task, many excellent works have emerged[7–10] and achieved good performance on benchmarks in recent years (e.g., Fig. 2).

Many works have reviewed the short-term trackers[12–14]. However, although a variety of long-term tracking algorithms have been proposed, there has been no work to make a comprehensive and in-depth survey of the algorithms, evaluation benchmarks and detailed performance analysis. In this work, we revisit existing long-term tracking algorithms from unified views and compare them on popular benchmarks. Our main contributions are summarized as follows.

Comprehensive review of long-term tracking algorithms from various aspects in unified views. We collect existing long-term tracking algorithms and categorize them based on two views: framework architectures and utilization of intermediate tracking results. The long-term tracking benchmarks with corresponding evaluation protocols are also described in detail.

A comprehensive evaluation of popular long-term trackers on popular benchmarks. We collect representative long-term trackers and evaluate them on six benchmarks for comparison. We further analyse the advantages and drawbacks of different frameworks with the speed and accuracy results of experiments.

Prospective discussion for long-term tracking. We discuss the potential directions for long-term tracking from views of algorithm design and benchmark construction, which may provide possible guidance to researchers.

The rest of the paper is organized as follows. In Section 2, we introduce the development of short-term tracking and previous relevant summative works about long-term tracking. In Section 3, we describe our categories of existing long-term trackers with detailed analysis. The introduction of long-term tracking benchmarks with corresponding evaluation protocols is presented in Section 4. A comparison with short-term tracking benchmarks is also analysed in this section. In Section 5, we report the experimental results of representative long-term trackers on several benchmarks. Finally, we provide discussions about further directions of long-term tracking in Section 6 and conclude the paper in Section 7.

2 Background

2.1 Development of short-term tracking

Visual object tracking aims to predict the specific object in the following video frames, given the state of the tracked object in the first frame. The state is always the coordinates of axis-aligned boxes. Tracking methods can be divided into two types according to the average length of videos and sequence properties such as target visibility: short-term tracking and long-term tracking. The short-term tracking is the basic form of tracking and attracts the most attention during the development of visual tracking. Numerous methods have been proposed to im-

Fig. 2 Tracking results of Siam R-CNN[11] on sampled long-term videos. The score in the top right corner represents the prediction score of target presence. Better viewed in color and in zoom.

prove the performance.

The short-term tracking has experienced two mainstream frameworks, including the discriminative correlation filter (DCF) and the Siamese network. In 2010, Bolme et al.[15] introduced the DCF method with fast processing in the Fourier domain into the visual tracking. The algorithm achieves high-speed and good accuracy. Subsequently, more extended works based on DCF have made attempts to improve the performance. Henriques et al. propose CSK[16] exploiting the properties of the circulant matrix to obtain the approximate dense sampling of the cyclic shift instead of sparse sampling. KCF[17] realizes an efficient combination of multi-channel features and utilizes the kernel method, achieving a high speed and a significant performance promotion. In SAMF[18] and DSST[19], the multi-scale estimation mechanisms applied to the traditional DCF algorithm are introduced to deal with scale estimation. Danelljan et al.[20] attempt to add a spatial regularization to the learned filter to eliminate the boundary effect. Based on [20], C-COT[21] and ECO[22] convert discrete position estimation to continuous position estimation, and attempt multi-level and multi-resolution features fusion for better performance.

In the deep learning era, the short-term tracking has made greater progress mainly based on the Siamese network. The pioneering works based on the Siamese network are SINT[23] and SiamFC[24], which train a similarity metric between the target exemplar and candidate search regions offline. Li et al.[25] present a unified architecture of Siamese feature extraction and anchor-based region proposal subnetworks including the classification and regression branch with high speed. Based on [25], more variants with anchor-free architecture[26–29] emerge and make better performance. An extra discriminative training strategy is also designed in [30]. In SiamRPN++[31] and SiamDW[32], efforts are made to exploit the potential of deep backbone network for better tracking performance. Danelljan et al.[33, 34] propose a discriminative learning network with online update to combine the DCF idea with the Siamese network architecture, achieving excellent performance. More works[35, 36] follow the pipeline and make greater success. Recently, many works such as TransT[37] and STARK[38] explore the great power of transformer-based architectures and also achieve excellent performance.

2.2 Previous summative and related works about long-term tracking

Some studies conclude partial representative long-term trackers from the view of datasets or evaluation protocols. Such as in [3, 6], new challenging benchmarks with a new evaluation protocol for long-term tracking are proposed with a summary of evaluated trackers. Besides, Karthik et al.[39] propose novel evaluation strategies focusing on long-term aspects such as re-detection, recov-

ery and reliability and make an in-depth analysis of trackers from a long-term perspective.

Relatedly, Kuipers et al.[40] propose a small dataset containing different categories including the full-out-of-frame-occlusion challenge. In the multi-model tracking domain, a long-term visible-depth (RGB-D) dataset CDTB[41] is presented with periods of target absence. Qian et al.[42] propose a deep depth-aware long-term tracker with a depth-aware correlation filter utilized for re-detection. Kart et al.[43] utilize view-specific DCFs to localize the target after out-of-view rotation or heavy occlusion with target appearance changes from the 3D motion. However, there is no systematic and complete summary of long-term visual tracking. Therefore, we aim to conduct a comprehensive survey on this field.

3 Long-term visual tracking

This section provides an overview of long-term tracking from two perspectives: framework architectures and utilization of intermediate tracking results. The overall taxonomies are shown in Fig. 3.

3.1 Framework architectures

In this section, long-term trackers are categorized based on the framework architectures: local-global trackers and global trackers. Local-global trackers can be viewed as an extension of short-term trackers, which are usually composed of three parts: The local tracker (responsible for local mode), the detector (responsible for global mode) and the verifier module that interact between local trackers and detectors. The local tracker, a component played by the short-term tracker, focuses on the robust target model facing normal challenges and is responsible for searching in local region inherited from the last frame. The detector is responsible for searching target candidates from the whole image when the target is lost. For effective collaboration, the verifier module provides strategies such as scoring for switching or communicating between two modules. While for global trackers, they generally have only one target-specific detection module equipped with other spatial-temporal postprocessing methods, and then the most likely candidate of the target in every frame will be selected to constitute the tracking results of a sequence.

3.1.1 Local-global trackers

Local-global trackers can fully benefit from the progress of the local tracker, which is the essential module in all components. From the view of the local tracker component, local-global trackers can be further divided into three categories: H-tracker, D-ON-tracker and D-OFF-tracker. From the view of the detector component, local-global trackers can be further divided into two categories: CGA-detector and DSS-detector. From the view of the verifier component, local-global trackers can be further divided into two categories: local-verifier and other-verifier. The detailed explanations of abbreviations are presented in Table 1.

H-tracker. Various traditional hand-crafted features are presented before the deep learning era, such as HOG features and scale-invariant feature transform (SIFT) features. In earlier works, there exist keypoints methods using L-K optical flow in [44] and median flow in [4].

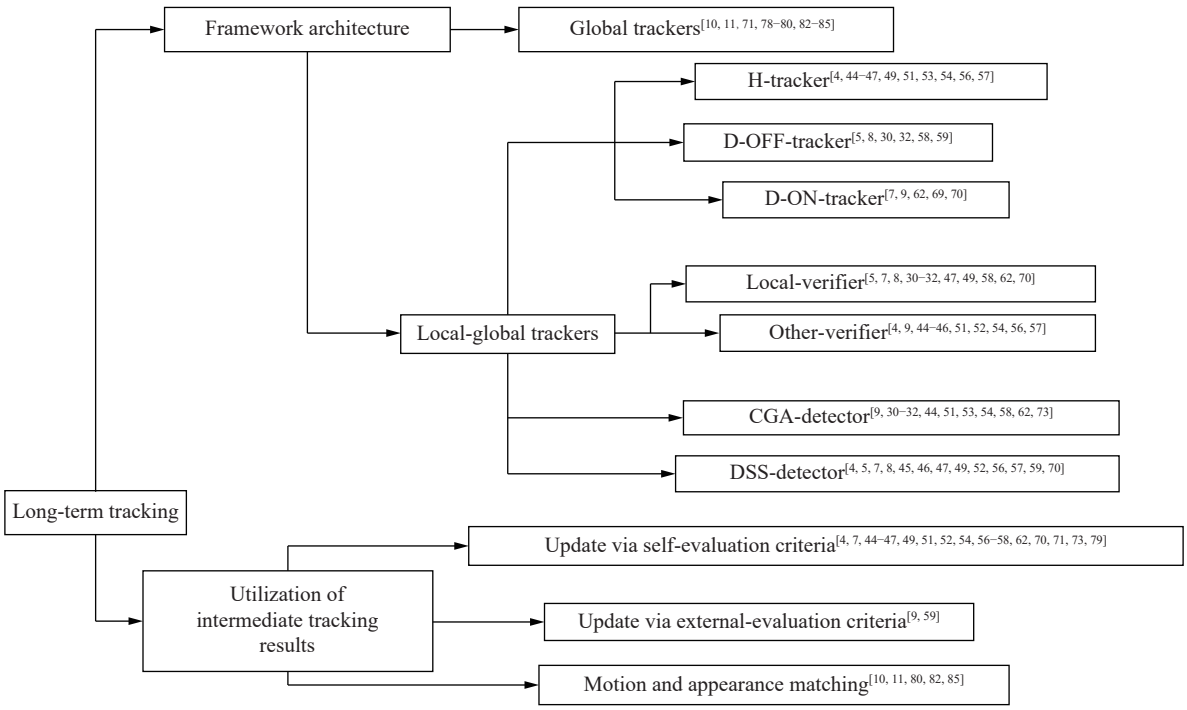

Fig. 3 Taxonomies and corresponding algorithms of long-term tracking

Table 1 Detailed explanations of abbreviations

Abbreviations	Explanations
H-tracker	Local-global trackers with a local tracker using hand-crafted features
D-OFF-tracker	Local-global trackers with an offline local tracker using deep features
D-ON-tracker	Local-global trackers with an online local tracker using deep features
CGA-detector	Local-global trackers with the candidate generating algorithm as the detector
DSS-detector	Local-global trackers with dense or sparse sampling in detector module
Local-verifier	Local-global trackers with the local tracker as the verifier
Other-verifier	Local-global trackers with other manners as the verifier

However, due to the weak representation ability, these trackers cannot deal with the complicated challenges. The short-term tracker with online update can capture the appearance variations of the target better when facing the long-term property than trackers without update. Except for some early works, such as [45], which utilizes online support vector machine (SVM) with HOG features to be responsible for searching in the local region, the discriminative correlation filter (DCF) is the most popular choice. Traditional DCFs learn a filter online as the discriminative template to model the target and perform correlation operations between it and the search region; then, the best response is regarded as the location of the target. Many works adopt variants of DCF trackers as the short-term tracker. In [46], two regression models based on DCF are utilized to estimate translation and scale separately with HOG features. Wang et al.[47] adopt Staple[48] as the short-term tracker with HOG and color features. Fan and Ling[49] utilize fDSST[50] to locate the center and scale of the target. Hong et al.[51] combine KCF[17] and DSST[19] to obtain more discriminative ability in the local search region, with HOG and SIFT features. A modified KCF is also utilized in [52] and similar idea is performed in [53]. In [54], a fully correlational long-term tracking framework is proposed, and the short-term tracker is designed based on CSR-DCF[55] with HOG and colornames features. Wang et al.[56] employ the support vector correlation filter with HOG features as the short-term tracker. Tang and Ling[57] select DSST[19] as the short-term tracker with contour features as a constraint.

D-OFF-tracker. The offline-trained deep Siamese network achieves excellent performance in the short-term tracking field without updating the object model. Therefore, they are widely adopted as the local tracker component in long-term tracking. In [5, 58], SiamFC[24] is adopted as the local tracker to perform tracking in the local search region. Gavves et al.[59] utilize SINT[23] as the local tracker for local tracking. Zhu et al.[30] extend DasiamRPN with an increasing search size when failure occurs, and in [30], the tracker outputs reliable scores benefiting from a reduction of the imbalance of training data distribution especially for semantic distractors. Zhang and Peng[32] also extend SiamDW with larger region for re-detection of the target after failure. Yan et al.[8] utilize SiamRPN[25] with lightweight MobileNets[60] and an offline-trained verification network as the robust local tracker, ensuring the balance of speed and accuracy.

D-ON-tracker. The MDNet[61], which is a deep online update tracker based on candidate region classification, is a good choice to be adopted as the local tracker. Zhang et al.[7] perform target-aware feature fusion to fuse the features of the template and search region with MobileNets[60] as the feature extractor, then a combination of a region proposal network and a MDNet-based[61] online verification network are utilized to track the target in a local region. Wu et al.[62] propose a combination of SiamRPN[25] and the online-updated MDNet[61] as the base local tracker. Besides, benefiting from the strong feature representation ability of deep learning[60, 63, 64], the combination of the DCF idea and deep networks is realized[33–35]. In ATOM[33], a two-conv-layer network with the similar thoughts of traditional DCF is designed to make an online-updated localization. DiMP[34] improves the target classification branch in a Siamese way with a discriminative loss function for distinguishing the target from background, and a powerful optimization strategy is also presented to ensure rapid convergence. This series of variants has been also the popular choice to play the role of local tracker[65–67]. Dai et al.[9] adopt a combination of the online tracker DiMP[34] and the refined module SiamMask[68] for bounding box regression as the local tracker. Zhang et al.[69] adopt ATOM[33] as the basic local tracker, and equip the tracker with squeeze-and-excitation networks to highlight more useful features and the relocation module to improve the scale adaptive ability. Choi et al.[70] improve SuperDimp[1] with additional background augmentation for more discriminative feature learning during online update as the local tracker.

CGA-detector. The detector is the core module to distinguish the long-term and short-term tracking in most cases. The module provides candidate search regions, bounding boxes, or keypoints in the whole image or a larger region from the global view when tracking fails in the local search region. Detectors based on the target candidate generating algorithms provide accurate candidate

[1] https://github.com/visionml/pytracking.

bounding boxes directly, similar to object detection. In [44, 51], keypoints detection is performed for later matching. Liao et al.[53] utilize EgdeBox[72] algorithm adjusted for the visual tracking task and optimize the generated candidates via a passively updated DCF tracker. Liu et al.[73] also propose an instance-specific proposal generator to exploit the prior distribution information of the target based on EgdeBox[72] algorithm. Wu et al.[62] apply the flow estimate module by PWC-Net[74] first and then perform Gaussian sampling or GA-RPN[75] to generate candidate boxes for classification, while the search region increases gradually until the entire image if the condition of refound is not met. Lee et al.[58] propose to correlate the features of the whole image and template extracted by VGGNet[63], and then the top N coarse positions of the candidates which have similar semantic meanings with the target are collected for further verification. Dai et al.[9] perform a cascade of faster R-CNN[76] and SiamRPN[7] to get the candidate boxes for the target. In some works, a gradually increasing or fixed larger search region is adopted instead of the entire image, and the short-term tracker with the region proposal subnetwork is directly utilized to provide one or more target candidates[30–32]. In particular, in [54], a modified DCF with alternating direction method of multipliers (ADMM) optimization is exploited which allows itself to search in an area with size unrelated to the object.

DSS-detector. Dense or sparse patch sampling is the more common method to provide initial candidate regions in the global search region. No extra target-generating networks are needed to provide potential target candidates. The scanning window, a dense sampling strategy is widely employed, followed by the random fern classifier, SVM, or online cascaded classifier[4, 45, 46, 52, 56]. In MBMD[7], a sliding window strategy is utilized to provide all possible search regions for the subsequent verifying network. In PTAV[49], sliding windows in an increasing region are utilized, and the Siamese network is followed to determine the detection result. An additional region pooling layer is employed to make the process simultaneous. However, heavy computational burden can be brought by dense sampling methods. More region filtering or sparse strategies are utilized. In [47], the sparse coding scheme based on reconstruction error is applied first to discard most of the false candidates, and then candidate selection through a particle filter is applied for accurate target localization. Yan et al.[8] realize fast and effective searching by applying an offline-trained skimming module. The skimming module is performed fast to select regions which are densely sampled with sliding windows, and then the top-K candidate regions are obtained for further verification in local regions. Two candidates are provided by motion estimation and pixel-wise color score map in a larger search region in [57]. For the latter, the search region is divided into six patches to find the best location of target. A random window strategy is utilized directly in [70], and multiple frames can approximately realize searching in the entire image. In SiamFC+R[5], a simple re-detection strategy which considers a search area at a random location in each frame is utilized until the maximum score is over the predefined threshold. In [59], a hierarchical global search strategy of three levels is adopted for better performance.

Local-verifier. The verifier is in charge of selecting the real target candidate and determines mode control such as the mode switching between the local and global, supplemented by strategies such as thresholds of external-module scores or existing scores from other two modules. In [7, 8, 30–32, 49, 58, 62], the local tracker is directly applied to every candidate search region, generating the box with the best confidence score. A threshold or reliability condition is designed based on the output similarity maps or scores to switch between local and global modes. Such as in [58], Lee et al. select the best candidate in the global mode depending on the local tracker and define multiple criteria to verify whether the final candidate is the target. The ratio of current maximum value and the average of recent responses′ maximum value provided by the local tracker is utilized to activate the detector. Similar ideas also exist in [5, 47, 70].

Other-verifier. Other manners of verifier include the following. In some works, the module works mainly by a classifier. Kalal et al.[4] utilize a cascaded classifier to verify the presence of target, and the P-N ferns are designed and updated online, which select training data provided by tracker for detector and initialize the tracker when it fails using the detector. In [46], an online random fern classifier is trained online with the local tracker for target verification. Similar situations exist in [45, 52, 56]. In [44], global search is employed to establish matches of keypoints based on appearance purely for a static model, and static-adaptive correspondences make complementary work. In [51], templates in short-term and long-term stores are utilized separately, then the algorithm selects the better of the two results from the tracker and detector based on indicators of occlusion and confidence. In [57], results from the short-term tracker and the detector are selected by elaborated decision-making. In [54], the tracking uncertainty estimation is designed to activate the detector working in parallel with the local tracker. LTMU[9] applies the external RT-MDNet[77] to every candidate box to score the probability of a real target of a real target, as shown in Fig. 4. When the target found by the short-term tracker is in low verification, the detector will launch. When the most possible candidate box reappears, it will be utilized to reset the local tracker.

3.1.2 Global trackers

Global trackers perform as a target-specific detector within the whole image in every video frame. They can treat every frame as an independent input to get one or more target proposals. For multiple target proposals, they are usually associated with spatial-temporal cues between

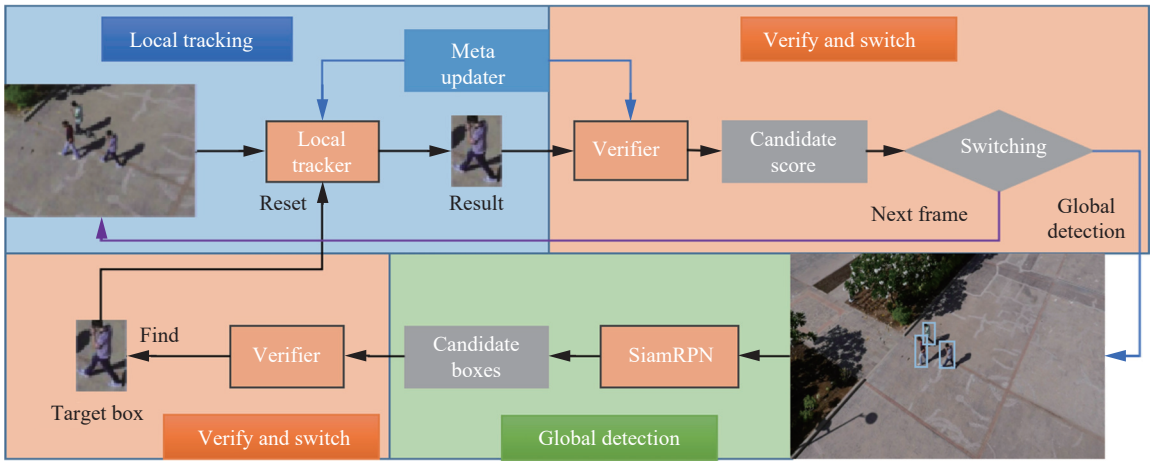

Fig. 4 Flowchart of LTMU[9] algorithm

frames to select the best one and then constitute continuous tracking outputs in a video sequence. In [71], edge-based detection is adopted to get proposals for the target, and an online SVM classifier is trained to score the proposals. Then all proposals are selected with classification scores and proper temporal constraints. In [78], multiple keypoint-based methods inside a fallback model are set as the detection model, and the algorithm learns to select good samples with a growing and pruning approach to update the object model for better detection. In [79], an online SVM with HOG features is used as a detector to provide proposals every frame, and then a dynamic programming is performed to solve the shortest path problem for tracking. The data selection strategy from self-paced learning is also utilized to re-learn the detector. Tracklet dynamic programming also exists in [11], and the tracker is designed as a Siamese two-stage detection network based on the faster R-CNN[76] architecture to provide target proposals. Dave et al.[80] convert a category-specific object detector into an object-specific detector based on the mask R-CNN[81] architecture, and propose a lightweight strategy for computing discriminative target templates end-to-end for handling distractors efficiently. Zhang et al.[82] also equip the target-specific Siamese detection network with re-identification association. A similar process exists in [83] and [84] but no temporal cues are utilized. The candidate with the top classification score will be selected directly as the target. Huang et al.[83] construct a target-specific object detector based on faster R-CNN[76] architecture, and design convolutional modules to learn how to modulate features of search regions with target template extracted by region of interest (ROI) align[81], as shown in Fig. 5. Choi et al.[84] adopt an anchor-free detection architecture instead, and a fine-matching stage with context embedding is added to improve the ability to distinguish distractors and the target. Li et al.[85] also detect target candidates via proposed two-stage tracking component based on faster R-CNN[76], and a CNN-based trajectory prediction module is proposed to exploit the target's temporal motion information for the suppression of distractors. KeepTrack[10] achieves excellent performance via elaborate target candidate association to suppress the distractors. The target candidates can be located via the similarity response map.

3.2 Utilization of intermediate tracking results

A great challenge in long-term property is that the tracked object may undergo more complicated target variations and interference during long-time duration. The only template of the first frame applied in offline-trained trackers makes long-term tracking particularly challenging. Except for some algorithms which adopt a

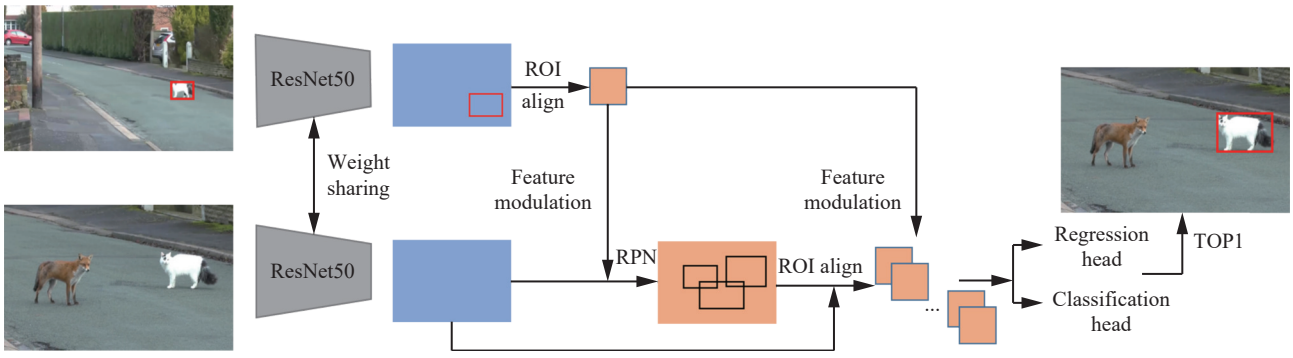

Fig. 5 Flowchart of GlobalTrack[83] algorithm

completely offline-trained network and do not exploit explicit historical information except for the local search region inheritance from last frame[5, 8, 30–32, 83, 84], in more cases, the intermediate tracking results are utilized to enhance trackers for better tracking performance in subsequent frames.

3.2.1 Update via self-evaluation criteria

For long-term tracking, trackers with update can capture the variations of target better. Tracking results of intermediate frames can be utilized to finetune the object model or enhance the template memories so that the tracker will capture richer visual information about the target in subsequent frames. Such as in early work[44], the adaptive model updates every frame with image patches with the position of the last frame. However, if the update is inappropriate, it may cause tracker′s performance degradation. So, most trackers exploit self-evaluation criteria with tracking outputs to assess the reliability of current training samples. These criteria include the confidence score provided by classifiers, the peak-to-sidelobe rate (PSR) or the maximum of similarity response map[4, 7, 46, 49, 52, 62, 70, 71, 79], the ratio of the current to the recent averaged response[54, 56], the ratio of numbers of keypoints in and out of box[51], and other combinations of different observations of current tracking results[47, 57, 58, 73]. Zhu et al.[71] online update the object model concentrating on hard false-positives supplied by proposals during tracking to suppress distractors. In [4], P-N learning acts as a filtering module for better samples to train the detector and tracker online. In [46], a k-nearest neighbour (KNN) classifier is used to filter samples for online training. Liu et al.[73] propose a novel adaptive updating strategy to prevent model degeneration caused by incorrect update. The state prediction strategy is designed based on motion cues to guide the update in [45]. Besides, in [51, 58], the memory model is separated into short-term and long-term stores inspired by the Atkinson-Shiffrin memory model (ASMM), and the update standard is set separately.

3.2.2 Update via external-evaluation criteria

Self-evaluation criteria have inevitable risks in bringing noise and template contamination. To better evaluate the trackers′ reliability, external networks are introduced in some works to exploit the potential valuable sequential information. Dai et al.[9] attempt to tackle the update problem with an additional meta-learning module using temporal information. The offline-trained meta-updater is constructed with the LSTM network and receives sequential historical tracking observations as inputs. Then the network learns to integrate the geometric cue, discriminative cue and appearance cue to output an update indicator, which is tracker-specific. It plays an essential role to determine when is the time suitable for tracker update. A similar idea exists in [59], a decay recognition network (DRN) based on LSTM is designed to estimate the bias for model update in the next frame from history similarity maps.

3.2.3 Motion and appearance matching

In some works, although the object model is not updated with the time, historical motion and appearance information is also utilized to make the tracking results more reliable. To suppress distractors better, motion information attracts attention. Dynamic attention guided multi-trajectory analysis is utilized in [86]. Tracklet dynamic programming or motion prediction is utilized in [11, 85] as well. Additionally, historical appearance features are also utilized in cascaded re-detection branches in [11]. In [80], temporal smoothness is incorporated to avoid latching onto distractors, which needs historical target proposals to build the motion connections. Multiple object tracking (MOT) philosophy is utilized in [82] to make target association and suppress distractors. KeepTrack[10] follows a similar idea, matching the distractors frame-by-frame and filtering them out.

4 Evaluation benchmarks

With the development of long-term tracking, many evaluation benchmarks have been proposed. The most prominent differences between long-term benchmarks and short-term benchmarks are the duration of videos and the proportion of situations where the target is visible. In this section, we summarize six popular long-term datasets and compare them with the short-term datasets OTB-100[87] and UAV123[1] in Table 2.

4.1 VOTLT

In 2018, the VOT challenge first introduces long-term tracking challenge. The dataset VOTLT2018[65], which inherits the LTB35 dataset[3], contains 35 challenging sequences of diverse objects (e.g., persons, cars, motorcycles, bicycles and animals) with a total length of 146 847 frames and 433 target disappearances. Twenty sequences among the dataset are obtained from the UAV20L[1] dataset tracked by low altitude drones. Each sequence contains about 12 target disappearances, with an average length of about 40 frames. The targets are all annotated by an axis-aligned bounding box and each sequence is annotated by nine visual attributes[65]: full occlusion, out-of-view, partial occlusion, camera motion, fast motion, scale change, aspect ratio change, viewpoint change and similar objects. For evaluation, three measures are proposed: Precision (Pr), Recall (Re) and tracking F-score. The Pr and Re with threshold τ_θ are defined as follows[3]:

$$Pr(\tau_\theta) = \frac{1}{N_p} \sum_{t \in \{t:A_t(\theta_t) \neq \emptyset\}} \Omega(A_t(\theta_t), G_t) \tag{1}$$

$$Re(\tau_\theta) = \frac{1}{N_g} \sum_{t \in \{t:G_t \neq \emptyset\}} \Omega(A_t(\theta_t), G_t) \tag{2}$$

Table 2 Statistics of popular long-term benchmarks and two short-term benchmarks: OTB100 and UAV123. The "Avg. absent" means the average number of absences per track (or sequence). The "dur." represents duration, which means the average duration per absence. For the VOTLT dataset, we assume that the average length is converted via 30 fps approximately.

Name	Track Num.	Avg. length	Avg. absent/dur.	Min/Max frames	Frame rate	Absent label
VOTLT2018	35	139.8 s	12.37/40.6 frames	1 389/29 700	–	Yes
VOTLT2019	50	143.5 s	10/52 frames	1 389/29 700	–	Yes
OxUvA	366	141.6 s	0.695/2.58 frames (dev)	900/37 440	30 fps	Yes
TLP	50	484.8 s	6.32/64 frames	4 320/28 590	24/30 fps	Yes
LaSOT (test)	280	81.6 s	3.27/20 frames	1 000/9 999	30 fps	Yes
Long-term subset of VTUAV-V (test)	74	179.4 s	1.74/132 frames	27 213/493	30 fps	Yes
UAV20L	20	97.8 s	–/–	1 717/5 527	30 fps	No
OTB100	100	19.6 s	–/–	71/3 872	30 fps	No
UAV123	123	30.5 s	–/–	109/3 085	30 fps	No

where G_t is the groundtruth of the target and $A_t(\tau_\theta)$ is the bounding box predicted by the tracker, θ_t is the prediction certainty score at time-step t, τ_θ is a classification (detection) threshold, $\Omega(A_t(\tau_\theta), G_t)$ is the intersection over union (IoU) between the tracker prediction and the groundtruth, N_g is the number of frames with $G_t \neq \emptyset$, N_p is the number of frames with existing prediction[65].

To combine precision and recall to a single metric, the F-measure is defined by (3):

$$F(\tau_\theta) = \frac{2Pr(\tau_\theta)Re(\tau_\theta)}{Pr(\tau_\theta) + Re(\tau_\theta)}. \tag{3}$$

Considering the influence of manual-set thresholds, the highest score on the F-measure plot, named the F-score, is defined as the rank metric of algorithms. Besides, VOTLT challenge also proposes the re-detection evaluation tested on artificial sequences generated from the initial frame, which aims to test the tracker's re-detection capability based on two criteria. The criteria include the average number of frames required for re-detection (Frames) and the percentage of sequences with successful re-detection (Success). To emphasize the re-detection capability, the target appearance was kept constant.

In 2019, the VOTLT dataset extends to 50 videos[6], with the total length of 215 294 frames and an average 10 of long-duration target disappearances each sequence, with each disappearance lasting for average 52 frames. The evaluation protocol and attribute categories for sequences keep the same. The VOTLT2020[67] and VOTLT2021[88] are the same as the VOTLT2019[66] in both dataset and evaluation protocol. The VOTLT benchmarks have the complete official evaluation toolkit in Python for testing and evaluation.

4.2 OxUvA

The OxUvA[5] dataset comprises 366 object tracks from 22 classes in 337 videos with two sets: development (dev) of 200 and test of 166 tracks. It is worth noting that the classes in the dev and test sets are disjoint, which are chosen randomly. The dataset contains sequences with an average duration of 2.3 minutes with a total frames of 1.55 million, labelled at a frequency of 1 Hz. An average of 2.2 absent labels per track and more than half of the videos with target disappearances exist. For evaluation, the groundtruth for the test set is only accessible via a rate-limited evaluation server[2], which helps avoid over-fitting hyper-parameters on the specific dataset. There exist three major criteria to evaluate the performance of different trackers: True positive rate (TPR), true negative rate (TNR) and maximum geometric mean (MaxGM). The TPR measures the proportion of present targets to targets reported present and located correctly, and the TNR measures the fraction of absence reported. To combine them into a single metric, the geometric mean GM is defined as follows:

$$GM = \sqrt{TPR \times TNR}. \tag{4}$$

To be compatible with trackers that do not have the ability to predict the absence of the target, the maximum geometric mean, MaxGM, is defined as follows for ranking algorithms[5]:

$$MaxGM = \max_{0 \leq p \leq 1} \sqrt{((1-p) \times TPR)((1-p) \times TNR + p)}. \tag{5}$$

For a given tracker, a larger MaxGM value means the better performance.

4.3 LaSOT

Large-scale single object tracking (LaSOT)[89] is a high-quality benchmark, and consists of 1 400 sequences with 70 categories and more than 3.5 M frames in total. The dataset consists of two sets: training and testing sub-

[2] https://oxuva.github.io/long-term-tracking-benchmark/

sets, with 1 120 videos and 280 videos, respectively. The sequences provide various challenges including target disappearing and reappearing in the view. Every frame is annotated with an axis-aligned bounding box with an absent label, either out-of-view or full occlusion. Each sequence is labelled with 14 attributes, including illumination variation (IV), full occlusion (FOC), partial occlusion (POC), deformation (DEF), motion blur (MB), fast motion (FM), scale variation (SV), camera motion (CM), rotation (ROT), background clutter (BC), low resolution (LR), viewpoint change (VC), out-of-view (OV) and aspect ratio change (ARC).

For evaluation, one-pass evaluation (OPE)[87] is performed. The precision, normalized precision and success are three criteria. The precision is computed by comparing the distance between tracking results and the groundtruth in pixels. Considering the sensitivity of the precision metric to target size and image resolution, the normalized precision is utilized as in [90]. The success metric is measured as the intersection over union (IoU) between tracking results and groundtruth. The area under curve (AUC) of the success plot is computed for the success metric. The tracking algorithms are usually ranked with the success metric.

4.4 TLP

The track long and prosper (TLP)[2] dataset consists of 50 videos from real world scenarios, with a duration of over 400 minutes, in total 676 K frames. The average length of sequence in the TLP[2] dataset is over 8 minutes. All videos are labelled with nine visual attributes: illumination variation, scale variation, deformation, motion blur, fast motion, out of view, background clutter, occlusion and multiple instances. Three criteria are adopted to evaluate the algorithms: precision and success as in [87] and longest subsequence measure (LSM). The distance threshold for precision metric is 20 pixels, while the AUC of the success plot represents for the success metric. The LSM metric computes the proportion of the length of the longest successfully tracked continuous subsequence to the total length of the sequence[2]. A subsequence is marked as successfully tracked, only if x% of frames within it have IoU > 0.5, where x is a hyper-parameter and usually fixed during evaluation. The tracking algorithms are usually ranked with the success metric as well.

4.5 Long-term subset of VTUAV-V

The VTUAV-V[91] is a large scale visible-thermal (RGB-T) dataset captured by a professional UAV. Among the multi-modal sequences, 74 long-term sequences with the visible modality RGB can also be evaluated for long-term tracking. The OPE protocol is adopted to compare the trackers.

4.6 Comparison and analysis

According to Table 2, sequences in common short-term datasets only have an average duration of 20–30 seconds or even shorter, while typical long-term datasets have a significantly longer average length of sequence, with absent labels for every frame. Besides, target disappearance and reappearance has a high frequency of occurrence. TLP[2] even has an average length with over 8 minutes. The average length of LaSOT[89] exceeds short-term datasets a lot but is not as long as TLP[2]. It is worth mentioning that, LaSOT[89] has abundant sequences with the OPE protocol, and it is also popular in the short-term tracking field.

From the view of evaluation protocol, OPE evaluation is more common[2, 89, 91], which is the same as the evaluation of short-term tracking. However, considerations about the unique characteristics such as re-detection ability are not taken seriously. The OxUvA[5] benchmark needs presence/absence predictions to determine whether the predicted confidence of target presence and rectangle will be used. VOTLT[65, 66] benchmarks also design a metric taking the confidence of target presence into consideration and a re-detection experiment. However, the re-detection evaluation of VOTLT with the artificial sequences is far away from real scenes which is not suitable enough.

5 Experiments

In this section, we conduct experiments on six public benchmarks and analyse the results of overall performance, attributed performance, and speed. Most of the selected algorithms are representative, and have publicly available implementations or tracking results. Some algorithms give the reported performances on datasets or these algorithms' performance reports are collected on corresponding benchmarks, we use them directly. For some experiments that need to be performed newly with publicly available implementations, we conduct experiments on the device with a RTX TITAN GPU with 24 GB memory and a Intel i9-9900K CPU(@3.60GHz × 16). One thing to mention is that, UAV20L[1] is a subset of VOTLT dataset, so we do not evaluate trackers on it additionally.

5.1 Experimental comparison on VOTLT benchmarks

We select 21 trackers to perform an overall comparison of the VOTLT2018[65] benchmark and 16 trackers for the VOTLT2019 (2020/2021) benchmark. The detailed results are shown in Table 3 and the left part of Table 4, which are ranked based on the F-score. According to the results, all of the top trackers are equipped with deep features. Since VOTLT2018 is a subset of VOTLT2019

Table 3 Experimental results on the VOTLT2018 benchmark

Tracker	Feature type	F-score	Precision	Recall
KeepTrack[10]	Deep feature	0.713	0.727	0.703
LTMU[9]	Deep feature	0.690	0.710	0.672
DMTrack[82]	Deep feature	0.683	0.687	0.655
RLTDiMP[70]	Deep feature	0.681	0.671	0.692
Siam R-CNN[11]	Deep feature	0.671	0.667	0.675
SiamRPN++ [31]	Deep features	0.626	0.644	0.608
SPLT[8]	Deep feature	0.616	0.633	0.600
LTA[80]	Deep feature	0.612	0.612	0.612
MBMD[7]	Deep feature	0.610	0.634	0.588
Dasiam-LT[30]	Deep feature	0.607	0.627	0.588
TACT[84]	Deep feature	0.560	0.575	0.546
MMLT[58]	Deep feature	0.546	0.574	0.521
flow-mdnet-rpn[62]	Deep feature	0.541	0.610	0.486
GlobalTrack[83]	Deep feature	0.523	0.560	0.491
FuCoLoT[54]	Hand-crafted feature	0.480	0.539	0.432
CALT[57]	Hand-crafted feature	0.41	–	–
PTAV[49]	Hand-crafted/Deep feature	0.31	–	–
MUSTer[51]	Hand-crafted feature	0.29	–	–
TLD[4]	Hand-crafted feature	0.27	–	–
LCT[46]	Hand-crafted feature	0.25	–	–
CMT[44]	Hand-crafted feature	0.22	–	–

(2020/2021), we focus on the performance on the latter. On the VOTLT2019 (2020/2021) dataset, mlpLT[88] which fuses STARK[38] and SuperDiMP gets the best rank. KeepTrack_LT[88] ranks second. STARK[38] and LTMU[9] follow them with a narrow margin. Among the trackers, mlpLT[88], LT_DSE[66], Megtrack[67] and CLGS[67] are briefly described in VOT challenge competitions. All the top trackers are equipped with deep features. Trackers with various manners of intermediate tracking results′ utilization[9, 10, 11, 38, 70, 82] exceed other trackers[7, 8, 32, 83, 84] with a relatively significant advantage, which demonstrates that these manners can tackle with the challenges in long-term tracking scenes better. Only target template of the first frame cannot work well in the long-term task.

5.2 Experimental comparison on OxUvA benchmark

We select 13 trackers to perform an overall comparison of the OxUvA[5] benchmark, as shown in the right part of Table 4, ranked by MaxGM. As the results are evaluated via submitting to a rate-limited evaluation server, we just record the available performance reports of the trackers from original papers. As shown in the right part of Table 4, KeepTrack[10] is significantly ahead of other trackers. LTMU[9] is in the second place with excellent performance. Similar to the performance on the VOTLT benchmarks, a large proportion of trackers with various manners of intermediate tracking results′ utilization perform better. However, the performance gap between different trackers is larger than on the VOTLT benchmarks.

Table 4 Experimental results on the VOTLT2019 (2020/2021) and OxUvA benchmarks

Dataset	VOTLT2019(2020/2021)			Dataset	OxUvA		
Tracker	F-score	Precision	Recall	Tracker	MaxGM	TPR	TNR
mlpLT[88]	0.735	0.741	0.729	KeepTrack[10]	0.812	0.796	0.828
KeepTrack_LT[88]	0.712	0.725	0.700	LTMU[9]	0.751	0.749	0.754
STARK-ST101[38]	0.701	0.702	0.701	Siam R-CNN[11]	0.723	0.701	0.745
LTMU[9]	0.697	0.721	0.674	LTA[80]	0.716	0.655	0.782
LT_DSE[66]	0.695	0.715	0.677	TACT[84]	0.709	0.809	0.622
Megtrack[67]	0.687	0.703	0.671	DMTrack[82]	0.688	0.686	0.694
DMTrack[82]	0.687	0.690	0.662	SPLT[8]	0.622	0.498	0.776
RLTDiMP[70]	0.681	0.667	0.695	GlobalTrack[83]	0.603	0.574	0.633
CLGS[67]	0.674	0.739	0.619	MBMD[7]	0.544	0.609	0.485
Siam R-CNN[11]	0.664	0.654	0.673	SiamFC+R[5]	0.454	0.427	0.481
SiamDW-LT[32]	0.656	0.678	0.635	TLD[4]	0.431	0.208	0.895
TACT[84]	0.569	0.578	0.561	LCT[46]	0.396	0.292	0.537
MBMD[7]	0.575	0.623	0.534	EBT[71]	0.283	0.321	0
SPLT[8]	0.565	0.587	0.544	–	–	–	–
GlobalTrack[83]	0.539	0.568	0.513	–	–	–	–
FuCoLoT[54]	0.411	0.507	0.346	–	–	–	–

5.3 Experimental comparison on LaSOT benchmark

There are results of 11 long-term trackers and 8 representative short-term trackers on the LaSOT[89] benchmark, as shown in Table 5. The results are ranked by success score. The "S-T" represents short-term trackers, and the "L-T" represents long-term trackers. As mentioned in Section 4.3, the LaSOT[89] is a popular benchmark in both short-term and long-term tracking due to its characteristics. KeepTrack[10] with a large search region and target candidate association to suppress distractors achieves the best performance. STARK[38] with a transformer-based architecture is only slightly behind, with temporal information utilized as well. Top trackers almost utilize the historical tracking results for object matching or updating. The purely offline trackers without historical results used like [8, 83, 84], have a relatively unsatisfactory performance. From the view of framework architecture, both the local-global trackers and global trackers can achieve competitive performance. Attribute-based performance is also shown in Fig. 6. The attributes of "Out-of-view" and "Full occlusion" are closely associated with characteristics unique to long-term tracking. The performance is similar to the overall performance above. KeepTrack[10] performs best on both attributes, and STARK[38] is a little behind on the "Full occlusion" attribute. KeepTrack[10], STARK[38] and Siam R-CNN[11] have comparable results on the "Out-of-view" attribute.

Table 5 Experimental results on the LaSOT benchmark

Tracker	S-T/L-T	Success	Precision	Norm. precision
KeepTrack[10]	L-T	0.673	0.704	0.774
STARK-ST101[38]	S-T	0.671	0.722	0.769
Siam R-CNN[11]	L-T	0.648	0.684	0.722
RLTDiMP[70]	L-T	0.644	0.660	0.735
SuperDiMP	S-T	0.640	0.659	0.730
TACT[84]	L-T	0.575	0.607	0.660
DMTrack[82]	L-T	0.574	0.580	–
LTMU[9]	L-T	0.572	0.572	0.665
DiMP[34]	S-T	0.568	0.564	0.648
ATOM[33]	S-T	0.518	0.506	0.576
GlobalTrack[83]	L-T	0.517	0.528	0.597
SiamRPN++ [31]	S-T	0.496	0.491	0.569
SPLT[8]	L-T	0.426	0.396	0.494
MDNet[61]	S-T	0.397	0.373	0.460
SiamFC[24]	S-T	0.358	0.341	0.449
PTAV[49]	L-T	0.250	0.254	0.274
LCT[46]	L-T	0.221	0.190	0.209
TLD[4]	L-T	0.210	0.174	0.193
fDSST[50]	S-T	0.203	0.184	0.208

5.4 Experimental comparison on TLP benchmark

The results on the TLP benchmark are shown in Table 6 ranked by success scores. KeepTrack[10] gets the top score leading with a significant advantage. Siam R-CNN[11] ranks second. Compared with global trackers without distractor-aware strategies[83, 84], Siam R-CNN[11] and KeepTrack[10] have an obvious advantages. This lead may benefit a lot from tracklet dynamic programming or target candidates association to suppress the distractors in complicated scenes. Local-global trackers with online update such as LTMU[9] and RLTDiMP[70] get obviously better performance than offline trackers SPLT[8] and MB-MD[7], indicating that an appropriate online update is es-

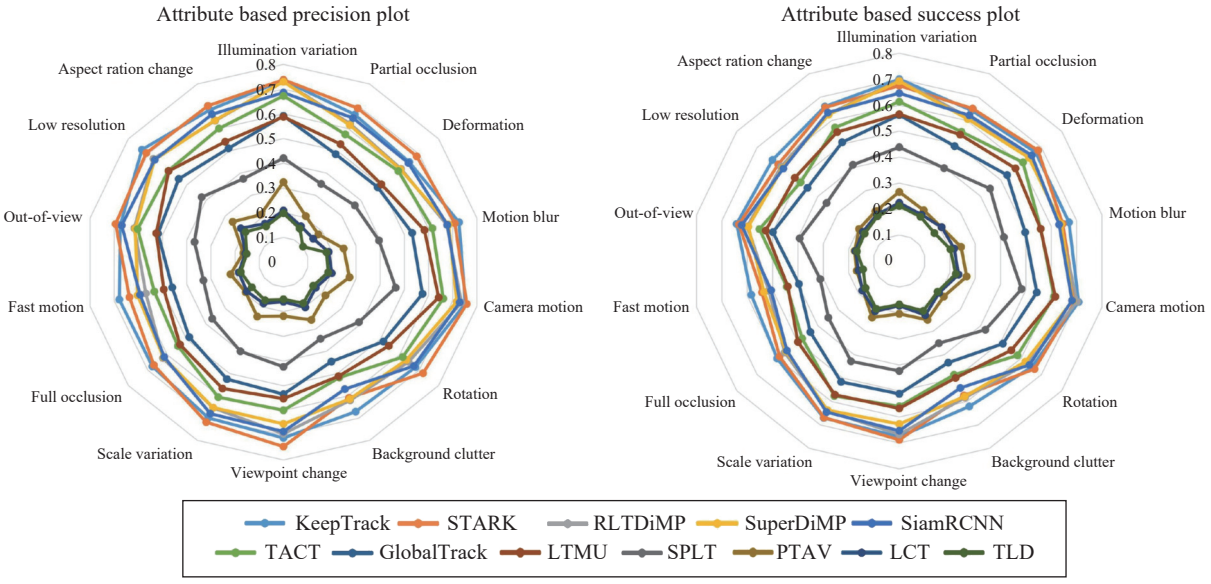

Fig. 6 Attribute results on the LaSOT benchmark

Table 6 Experimental results on the TLP benchmark

Tracker	S-T/L-T	Success	Precision
KeepTrack[10]	L-T	0.613	0.630
Siam R-CNN[11]	L-T	0.601	0.630
STARK-ST101[38]	S-T	0.577	0.594
LTMU[9]	L-T	0.571	0.608
RLTDiMP[70]	L-T	0.528	0.533
TACT[84]	L-T	0.523	0.545
GlobalTrack[83]	L-T	0.520	0.556
MBMD[7]	L-T	0.492	0.502
SPLT[8]	L-T	0.416	0.403
TLD[4]	L-T	0.122	0.116
LCT[46]	L-T	0.101	0.071

sential for longer sequences. In addition, SuperDiMP and STARK[38] with no explicit re-detection also perform well on LaSOT and TLP, which may indicate that they can be good choices for further extension design in long-term tracking.

5.5 Experimental comparison on the long-term subset of VTUAV-V benchmark

The results on the long-term subset of the VTUAV-V benchmark are shown in Table 7 ranked by success score. STARK[38] ranks first on the benchmark with a significant advantage. Some purely offline long-term trackers such as GlobalTrack[83] and SPLT[8] do not track well on this benchmark. As the benchmark is newly proposed, there is still much space for performance improvement.

Table 7 Experimental results on the long-term subset of the VTUAV-V benchmark

Tracker	S-T/L-T	Success	Precision
STARK-ST50[38]	S-T	0.504	0.565
LTMU[9]	L-T	0.487	0.569
DiMP[34]	S-T	0.387	0.445
SiamRPN++ [31]	S-T	0.360	0.415
SPLT[8]	L-T	0.360	0.418
GlobalTrack[83]	L-T	0.329	0.377
SiamFC[24]	S-T	0.238	0.288

5.6 Speed analysis

The speeds of long-term trackers are listed in Table 8. All the results and settings are collected from original papers, and the ranking is based on the F-score of VOTLT2018 except for STARK[38]. According to Table 8, most long-term tackers cannot achieve real-time speed on GPU or just run at a speed near real time. FuCoLoT[54] can run at 6 fps on CPU without need of GPU. However, its performance is worse than that of deep-feature trackers. Dasiam-LT[30] ranks second, but its performance still has a gap between the best trackers' results. DMTrack[82] balances speed and accuracy better. For some top-performance trackers such as KeepTrack[10], LTMU[9] and Siam R-CNN[11], there is an obvious gap between the speed and real-time speed on GPU. STARK[38] which is not designed specifically for long-term trackers, also shows great power in balancing speed and accuracy, and we conjecture that the transformer-architecture contributes a lot.

Table 8 Speed analysis of representative long-term trackers

Tracker	FPS	Device	Setting	Platform
KeepTrack[10]	12.7	GPU	RTX 2080Ti	Pytorch
STARK-ST101[38]	32	GPU	Tesla V100	Pytorch
LTMU[9]	13	GPU	RTX 2080Ti	TensorFlow, Pytorch
DMTrack[82]	31	GPU	Titan XP	Pytorch
RLTDiMP[70]	14.17	GPU	GTX 1080Ti	Pytorch
Siam R-CNN[11]	4.7	GPU	Tesla V100	TensorFlow
SiamRPN++ [31]	21	GPU	Titan Xp	Pytorch
SPLT[8]	26	GPU	GTX 1080Ti	TensorFlow
LTA[80]	7	GPU	–	–
MBMD[7]	4	GPU	GTX 1080Ti	TensorFlow
Dasiam-LT[30]	110	GPU	TITA X	Pytorch
TACT[84]	42	GPU	RTX 2080Ti	Pytorch
MMLT[58]	6.15	GPU	GTX 1080Ti	Matlab R2017a
GloalTrack[83]	6	GPU	GTX 1080Ti	Pytorch
FuCoLoT[54]	6	CPU	Intel Core i7	Matlab
PTAV[49]	27	GPU	GTX TITAN Z	C++, Caffe
SiamFC+R[5]	52	GPU	–	–

6 Future prospects

6.1 Algorithm design

Robust discriminative ability for long-term tracking. The discriminative ability of the appearance model is essential for visual tracking. Especially in long-term tracking, due to the long duration, the target may suffer from more severe variations or other challenges. Meanwhile, in long duration, error accumulation will be more distinct, so that failure may occur with a higher probability than in short-term tracking. Some works explore the update strategy to reduce template contamination and enhance adaptation to specific sequence and the discriminative ability. To reduce error accumulation, LTMU[9] attempts to train a meta-updater network using

historical tracking cues to predict whether the current frame is suitable for update. Memtrack[92] utilizes a dynamic memory network which stores the target information to maintain the variations of target appearance. MUSTer[51] and MMLT[58] set short-term and long-term stores of the appearance model separately. Ensuring the contiguity of correct tracking results while avoiding drift to semantic distractors or interfered background in long-term tracking is valuable. It is worth noting that the transformer-based tracker[38] also achieves good performance without complicated extra settings specified to the long-term property, which suggests that we can explore the potential of more new techniques. Besides, as inspired in [93], more works bridging long-term tracking and short-term tracking may also be worthy looking forward to.

Re-detection against distractors for long-term tracking. When failure occurs, the long-term trackers need to search the possible regions or bounding boxes and relocate the target successfully when the target reappears. Different solutions are proposed such as designed metrics or cascaded classifiers to verify the target presence, as described in Section 3.1. However, a larger search region can also introduce more distractors or background interference. Therefore, trackers also need to keep the tracklet avoiding drift when the target is under full occlusion or out-of-view. Recently, some works have achieved excellent performance in a sufficiently large search region with target association strategy to suppress distractors[10, 82], which is compatible with both short-term and long-term tasks. They adopt the similar idea of data association in the multiple object tracking task[94–96]. Siam R-CNN[11] adopts tracklet dynamic programming to build a distractor-aware model from a novel perspective for robust discriminative ability. Li et al.[85] exclude distractors with temporal information proposed by the motion model. The success rate of relocating the real target can be improved by filtering out distractors. More attention can be focused on giving a more elegant solution for this issue.

Speed for long-term tracking. According to Table 8, most state-of-the-art long-term trackers' speeds are slow or barely enough for real-time. Compared with short-term trackers, strategies for avoiding drift and searching in entire image may bring extra computational burdens. Moreover, since it is closer to the practical application, the speed of the algorithm should be higher. How to balance the cost of time and the level of performance is an essential issue. However, little attention is paid for it.

6.2 Benchmark construction

With the growing focus on long-term tracking, more long-term algorithms have been proposed. However, the large scale datasets designed specifically for long-term tracking are not sufficient. As mentioned in Section 4.6, some of the existing long-term evaluation protocols also lack the consideration about the long-term tracking characteristics. As the long-term tracking task is closer to the realistic needs, more related sequences should be collected to enlarge the evaluation datasets.

7 Conclusions

In this study, we provide a comprehensive survey of long-term visual tracking. First, we overview long-term tracking algorithms from two perspectives: framework architectures and utilization of intermediate tracking results. Then, we propose a detailed summary of existing benchmarks with evaluation protocols and compare their disadvantages and advantages. Subsequently, we compare the speed of algorithms and evaluate long-term trackers on six common long-term benchmarks, followed by a detailed analysis of the results. Finally, we propose a variety of perspectives for possible future directions, including aspects of algorithm design and benchmark construction.

Acknowledgements

This work was supported by National Natural Science Foundation of China (Nos. 62176041 and 62022021), Joint Fund of Ministry of Education for Equipment Preresearch, China (No. 8091B032155), the Science and Technology Innovation Foundation of Dalian, China (No. 2020 JJ26GX036), and the Fundamental Research Funds for the Central Universities, China (No. DUT21LAB127).

References

[1] M. Mueller, N. Smith, B. Ghanem. A benchmark and simulator for UAV tracking. In *Proceedings of the 14th European Conference on Computer Vision*, Springer, Amsterdam, The Netherlands, pp. 445–461, 2016. DOI: 10.1007/978-3-319-46448-0_27.

[2] A. Moudgil, V. Gandhi. Long-term visual object tracking benchmark. In *Proceedings of the 14th Asian Conference on Computer Vision*, Springer, Perth, Australia, pp. 629–645, 2019. DOI: 10.1007/978-3-030-20890-5_40.

[3] A. Lukežič, L. Č. Zajc, T. Vojíř, J. Matas, M. Kristan. Now you see me: Evaluating performance in long-term visual tracking. [Online], Available: https://arxiv.org/abs/1804.07056, 2018.

[4] Z. Kalal, K. Mikolajczyk, J. Matas. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012. DOI: 10.1109/TPAMI.2011.239.

[5] J. Valmadre, L. Bertinetto, J. F. Henriques, R. Tao, A. Vedaldi, A. W. M. Smeulders, P. H. S. Torr, E. Gavves. Long-term tracking in the wild: A benchmark. In *Proceedings of the 15th European Conference on Computer Vision*, Springer, Munich, Germany, pp. 692–707, 2018. DOI: 10.1007/978-3-030-01219-9_41.

[6] A. Lukežič, L. Č. Zajc, T. Vojíř, J. Matas, M. Kristan. Performance evaluation methodology for long-term visual object tracking. [Online], Available: https://arxiv.org/abs/

1906.08675, 2019.

[7] Y. H. Zhang, L. J. Wang, D. Wang, J. Q. Qi, H. C. Lu. Learning regression and verification networks for robust long-term tracking. *International Journal of Computer Vision*, vol. 129, no. 9, pp. 2536–2547, 2021. DOI: 10.1007/s11263-021-01487-3.

[8] B. Yan, H. J. Zhao, D. Wang, H. C. Lu, X. Y. Yang. 'Skimming-perusal' tracking: A framework for real-time and robust long-term tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, IEEE, Seoul, Korea, pp. 2385–2393, 2019. DOI: 10.1109/ICCV.2019.00247.

[9] K. N. Dai, Y. H. Zhang, D. Wang, J. H. Li, H. C. Lu, X. Y. Yang. High-performance long-term tracking with meta-updater. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 6297–6306, 2020. DOI: 10.1109/CVPR42600.2020.00633.

[10] C. Mayer, M. Danelljan, D. P. Paudel, L. Van Gool. Learning target candidate association to keep track of what not to track. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, IEEE, Montreal, Canada, pp. 13424–13434, 2021. DOI: 10.1109/ICCV48922.2021.01319.

[11] P. Voigtlaender, J. Luiten, P. H. S. Torr, B. Leibe. Siam R-CNN: Visual tracking by re-detection. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 6577–6587, 2020. DOI: 10.1109/CVPR42600.2020.00661.

[12] X. Q. Zhang, R. H. Jiang, C. X. Fan, T. Y. Tong, T. Wang, P. C. Huang. Advances in deep learning methods for visual tracking: Literature review and fundamentals. *International Journal of Automation and Computing*, vol. 18, no. 3, pp. 311–333, 2021. DOI: 10.1007/s11633-020-1274-8.

[13] P. X. Li, D. Wang, L. J. Wang, H. C. Lu. Deep visual tracking: Review and experimental comparison. *Pattern Recognition*, vol. 76, pp. 323–338, 2018. DOI: 10.1016/j.patcog.2017.11.007.

[14] S. M. Marvasti-Zadeh, L. Cheng, H. Ghanei-Yakhdan, S. Kasaei. Deep learning for visual tracking: A comprehensive survey. *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 5, pp. 3943–3968, 2022. DOI: 10.1109/TITS.2020.3046478.

[15] D. S. Bolme, J. R. Beveridge, B. A. Draper, Y. M. Lui. Visual object tracking using adaptive correlation filters. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, San Francisco, USA, pp. 2544–2550, 2010. DOI: 10.1109/CVPR.2010.5539960.

[16] J. F. Henriques, R. Caseiro, P. Martins, J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *Proceedings of the 12th European Conference on Computer Vision*, Springer, Florence, Italy, pp. 702–715, 2012. DOI: 10.1007/978-3-642-33765-9_50.

[17] J. F. Henriques, R. Caseiro, P. Martins, J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015. DOI: 10.1109/TPAMI.2014.2345390.

[18] Y. Li, J. K. Zhu. A scale adaptive kernel correlation filter tracker with feature integration. In *Proceedings of the European Conference on Computer Vision*, Springer, Zurich, Switzerland, pp. 254–265, 2015. DOI: 10.1007/978-3-319-16181-5_18.

[19] M. Danelljan, G. Häger, F. S. Khan, M. Felsberg. Accurate scale estimation for robust visual tracking. In *Proceedings of the British Machine Vision Conference*, BMVA Press, Nottingham, UK, pp. 1–11, 2014. DOI: 10.5244/C.28.65.

[20] M. Danelljan, G. Häger, F. S. Khan, M. Felsberg. Learning spatially regularized correlation filters for visual tracking. In *Proceedings of IEEE International Conference on Computer Vision*, Santiago, Chile, pp. 4310–4318, 2015. DOI: 10.1109/ICCV.2015.490.

[21] M. Danelljan, A. Robinson, F. S. Khan, M. Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *Proceedings of the 14th European Conference on Computer Vision*, Springer, Amsterdam, The Netherlands, pp. 472–488, 2016. DOI: 10.1007/978-3-319-46454-1_29.

[22] M. Danelljan, G. Bhat, F. S. Khan, M. Felsberg. ECO: Efficient convolution operators for tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, USA, pp. 6931–6939, 2017. DOI: 10.1109/CVPR.2017.733.

[23] R. Tao, E. Gavves, A. W. M. Smeulders. Siamese instance search for tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA, pp. 1420–1429, 2016. DOI: 10.1109/CVPR.2016.158.

[24] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, P. H. S. Torr. Fully-convolutional siamese networks for object tracking. In *Proceedings of the European Conference on Computer Vision*, Springer, Amsterdam, The Netherlands, pp. 850–865, 2016. DOI: 10.1007/978-3-319-48881-3_56.

[25] B. Li, J. J. Yan, W. Wu, Z. Zhu, X. L. Hu. High performance visual tracking with Siamese region proposal network. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, USA, pp. 8971–8980, 2018. DOI: 10.1109/CVPR.2018.00935.

[26] Y. D. Xu, Z. Y. Wang, Z. X. Li, Y. Yuan, G. Yu. SiamFC++: Towards robust and accurate visual tracking with target estimation guidelines. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, New York, USA, pp. 12549–12556, 2020. DOI: 10.1609/aaai.v34i07.6944.

[27] Z. P. Zhang, H. W. Peng, J. L. Fu, B. Li, W. M. Hu. Ocean: Object-aware anchor-free tracking. In *Proceedings of the 16th European Conference on Computer Vision*, Springer, Glasgow, UK, pp. 771–787, 2020. DOI: 10.1007/978-3-030-58589-1_46.

[28] Z. D. Chen, B. N. Zhong, G. R. Li, S. P. Zhang, R. R. Ji. Siamese box adaptive network for visual tracking. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 6667–6676, 2020. DOI: 10.1109/CVPR42600.2020.00670.

[29] D. Y. Guo, J. Wang, Y. Cui, Z. H. Wang, S. Y. Chen. SiamCAR: Siamese fully convolutional classification and regression for visual tracking. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 6268–6276, 2020. DOI: 10.1109/CVPR42600.2020.00630.

[30] Z. Zhu, Q. Wang, B. Li, W. Wu, J. J. Yan, W. M. Hu. Distractor-aware siamese networks for visual object tracking. In *Proceedings of the 15th European Conference on Computer Vision*, Springer, Munich, Germany, pp. 103–119,

2018. DOI: 10.1007/978-3-030-01240-3_7.

[31] B. Li, W. Wu, Q. Wang, F. Y. Zhang, J. L. Xing, J. J. Yan. SiamRPN++: Evolution of siamese visual tracking with very deep networks. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp. 4277–4286, 2019. DOI: 10.1109/CVPR.2019.00441.

[32] Z. P. Zhang, H. W. Peng. Deeper and wider siamese networks for real-time visual tracking. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp. 4586–4595, 2019. DOI: 10.1109/CVPR.2019.00472.

[33] M. Danelljan, G. Bhat, F. S. Khan, M. Felsberg. ATOM: Accurate tracking by overlap maximization. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp. 4655–4664, 2019. DOI: 10.1109/CVPR.2019.00479.

[34] G. Bhat, M. Danelljan, L. Van Gool, R. Timofte. Learning discriminative model prediction for tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, IEEE, Long Beach, USA, pp. 6181–6190, 2019. DOI: 10.1109/ICCV.2019.00628.

[35] M. Danelljan, L. Van Gool, R. Timofte. Probabilistic regression for visual tracking. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 7181–7190, 2020. DOI: 10.1109/CVPR42600.2020.00721.

[36] G. Bhat, M. Danelljan, L. Van Gool, R. Timofte. Know your surroundings: Exploiting scene information for object tracking. In *Proceedings of the 16th European Conference on Computer Vision*, Springer, Glasgow, UK, pp. 205–221, 2020. DOI: 10.1007/978-3-030-58592-1_13.

[37] X. Chen, B. Yan, J. W. Zhu, D. Wang, X. Y. Yang, H. C. Lu. Transformer tracking. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Nashville, USA, pp. 8122–8131, 2021. DOI: 10.1109/CVPR46437.2021.00803.

[38] B. Yan, H. W. Peng, J. L. Fu, D. Wang, H. C. Lu. Learning spatio-temporal transformer for visual tracking. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, IEEE, Montreal, Canada, pp. 10428–10437, 2021. DOI: 10.1109/ICCV48922.2021.01028.

[39] S. Karthik, A. Moudgil, V. Gandhi. Exploring 3 R′s of long-term tracking: Re-detection, recovery and reliability. In *Proceedings of IEEE Winter Conference on Applications of Computer Vision*, Snowmass, USA, pp. 1000–1009, 2020. DOI: 10.1109/WACV45572.2020.9093465.

[40] T. P. Kuipers, D. Arya, D. K. Gupta. Hard occlusions in visual object tracking. In *Proceedings of the European Conference on Computer Vision*, Springer, Glasgow, UK, pp. 299–314, 2020. DOI: 10.1007/978-3-030-68238-5_22.

[41] A. Lukezic, U. Kart, J. Käpylä, A. Durmush, J. K. Kamarainen, J. Matas, M. Kristan. CDTB: A color and depth visual object tracking dataset and benchmark. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, IEEE, Seoul, Korea, pp. 10012–10021, 2019. DOI: 10.1109/ICCV.2019.01011.

[42] Y. L. Qian, S. Yan, A. Lukežič, M. Kristan, J. K. Kämäräinen, J. Matas. DAL: A deep depth-aware long-term tracker. In *Proceedings of the 25th International Conference on Pattern Recognition*, IEEE, Milan, Italy, pp. 7825–7832, 2021.

[43] U. Kart, A. Lukežič, M. Kristan, J. K. Kämäräinen, J. Matas. Object tracking by reconstruction with view-specific discriminative correlation filters. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp. 1339–1348, 2019. DOI: 10.1109/CVPR.2019.00143.

[44] G. Nebehay, R. Pflugfelder. Clustering of static-adaptive correspondences for deformable object tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Boston, USA, pp. 2784–2791, 2015. DOI: 10.1109/CVPR.2015.7298895.

[45] Y. Hua, K. Alahari, C. Schmid. Occlusion and motion reasoning for long-term tracking. In *Proceedings of the 13th European Conference on Computer Vision*, Springer, Zurich, Switzerland, pp. 172–187, 2014. DOI: 10.1007/978-3-319-10599-4_12.

[46] C. Ma, X. K. Yang, C. Y. Zhang, M. H. Yang. Long-term correlation tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Boston, USA, pp. 5388–5396, 2015. DOI: 10.1109/CVPR.2015.7299177.

[47] N. Wang, W. G. Zhou, H. Q. Li. Reliable re-detection for long-term tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 3, pp. 730–743, 2019. DOI: 10.1109/TCSVT.2018.2816570.

[48] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, P. H. S. Torr. Staple: Complementary learners for real-time tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA, pp. 1401–1409, 2016. DOI: 10.1109/CVPR.2016.156.

[49] H. Fan, H. B. Ling. Parallel tracking and verifying. *IEEE Transactions on Image Processing*, vol. 28, no. 8, pp. 4130–4144, 2019. DOI: 10.1109/TIP.2019.2904789.

[50] M. Danelljan, G. Häger, F. S. Khan, M. Felsberg. Discriminative scale space tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1561–1575, 2017. DOI: 10.1109/TPAMI.2016.2609928.

[51] Z. B. Hong, Z. Chen, C. H. Wang, X. Mei, D. Prokhorov, D. C. Tao. Multi-store tracker (MUSTer): A cognitive psychology inspired approach to object tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Boston, USA, pp. 749–758, 2015. DOI: 10.1109/CVPR.2015.7298675.

[52] N. X. Liang, G. L. Wu, W. X. Kang, Z. Y. Wang, D. D. Feng. Real-time long-term tracking with prediction-detection-correction. *IEEE Transactions on Multimedia*, vol. 20, no. 9, pp. 2289–2302, 2018. DOI: 10.1109/TMM.2018.2803518.

[53] J. W. Liao, C. Qi, J. Z. Cao, L. Ren, G. P. Zhang. Real-time long-term tracker with tracking–verification–detection–refinement. *Journal of Visual Communication and Image Representation*, vol. 72, Article number 102896, 2020. DOI: 10.1016/j.jvcir.2020.102896.

[54] A. Lukežič, L. Č. Zajc, T. Vojíř, J. Matas, M. Kristan. FuCoLoT-a fully-correlational long-term tracker. In *Proceedings of the 14th Asian Conference on Computer Vision*, Springer, Perth, Australia, pp. 595–611, 2019. DOI: 10.1007/978-3-030-20890-5_38.

[55] A. Lukežic, T. Vojír, L. C. Zajc, J. Matas, M. Kristan. Discriminative correlation filter with channel and spatial reliability. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, USA, pp. 4847–4856, 2017. DOI: 10.1109/CVPR.2017.515.

[56] Z. P. Wang, H. Wang, B. F. Fang, C. J. Xie. Support vec-

tor correlation filter with long-term tracking. *Signal, Image and Video Processing*, vol. 12, no. 8, pp. 1541–1549, 2018. DOI: 10.1007/s11760-018-1310-0.

[57] F. Tang, Q. Ling. Contour-aware long-term tracking with reliable re-detection. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 12, pp. 4739–4754, 2020. DOI: 10.1109/TCSVT.2019.2957748.

[58] H. Lee, S. Choi, C. Kim. A memory model based on the siamese network for long-term tracking. In *Proceedings of the European Conference on Computer Vision Workshops*, Springer, Munich, Germany, pp. 100–115, 2019. DOI: 10.1007/978-3-030-11009-3_5.

[59] E. Gavves, R. Tao, D. K. Gupta, A. W. M. Smeulders. Model decay in long-term tracking. In *Proceedings of the 25th International Conference on Pattern Recognition*, IEEE, Milan, Italy, pp. 2685–2692, 2021. DOI: 10.1109/ICPR48806.2021.9412648.

[60] A. G. Howard, M. L. Zhu, B. Chen, D. Kalenichenko, W. J. Wang, T. Weyand, M. Andreetto, H. Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. [Online], Available: https://arxiv.org/abs/1704.04861, 2017.

[61] H. Nam, B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA, pp. 4293–4302, 2016. DOI: 10.1109/CVPR.2016.465.

[62] H. Wu, X. Y. Yang, Y. Yang, G. Z. Liu. Flow guided short-term trackers with cascade detection for long-term tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, IEEE, Seoul, Korea, pp. 170–178, 2019. DOI: 10.1109/ICCVW.2019.00026.

[63] K. Simonyan, A. Zisserman. Very deep convolutional networks for large-scale image recognition. [Online], Available: https://arxiv.org/abs/1409.1556, 2014.

[64] K. M. He, X. Y. Zhang, S. Q. Ren, J. Sun. Deep residual learning for image recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA, pp. 770–778, 2016. DOI: 10.1109/CVPR.2016.90.

[65] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Č. Zajc, T. Vojir, G. Bhat, A. Lukežič, A. Eldesokey, G. Fernández, Á. García-Martín, Á. Iglesias-Arias, A. A. Alatan, A. González-García, A. Petrosino, A. Memarmoghadam, A. Vedaldi, A. Muhič, A. F. He, A. Smeulders, A. G. Perera, B. Li, B. Y. Chen, C. Kim, C. S. Xu, C. Z. Xiong, C. Tian, C. Luo, C. Sun, C. Hao, D. Kim, D. Mishra, D. M. Chen, D. Wang, D. Wee, E. Gavves, E. Gundogdu, E. Velasco-Salido, F. S. Khan, F. Yang, F. Zhao, F. Li, F. Battistone, G. De Ath, G. R. K. S. Subrahmanyam, G. Bastos, H. B. Ling, H. K. Galoogahi, H. Lee, H. J. Li, H. J. Zhao, H. Fan, H. G. Zhang, H. Possegger, H. Q. Li, H. C. Lu, H. Zhi, H. Y. Li, H. Lee, H. J. Chang, I. Drummond, J. Valmadre, J. S. Martin, J. Chahl, J. Y. Choi, J. Li, J. Q. Wang, J. Q. Qi, J. Sung, J. Johnander, J. Henriques, J. Choi, J. Van De weijer, J. R. Herranz, J. M. Martínez, J. Kittler, J. F. Zhuang, J. Y. Gao, K. Grm, L. C. Zhang, L. J. Wang, L. X. Yang, L. Rout, L. Si, L. Bertinetto, L. T. Chu, M. Q. Che, M. E. Maresca, M. Danelljan, M. H. Yang, M. Abdelpakey, M. Shehata, M. Y. N. G. Kang, N. Lee, N. Wang, O. Miksik, P. Moallem, P. Vicente-Moñivar, P. Senna, P. X. Li, P. Torr, P. M. Raju, Q. Ruihe, Q. Wang, Q. Zhou, Q. Guo, R. Martín-Nieto, R. K. Gorthi, R. Tao, R. Bowden, R. Everson, R. L. Wang, S. Yun, S. Choi, S. Vivas, S. Bai, S. P. Huang, S. H. Wu, S. Hadfield, S. W. Wang, S. Golodetz, T. Ming, T. Y. Xu, T. Z. Zhang, T. Fischer, V. Santopietro, V. Štruc, W. Wei, W. M. Zuo, W. Feng, W. Wu, W. Zou, W. M. Hu, W. G. Zhou, W. J. Zeng, X. F. Zhang, X. H. Wu, X. J. Wu, X. M. Tian, Y. Li, Y. Lu, Y. W. Law, Y. Wu, Y. Demiris, Y. C. Yang, Y. F. Jiao, Y. H. Li, Y. H. Zhang, Y. X. Sun, Z. Zhang, Z. Zhu, Z. H. Feng, Z. H. Wang, Z. Q. He. The sixth visual object tracking VOT2018 challenge results. In *Proceedings of the European Conference on Computer Vision Workshops*, Springer, Munich, Germany, pp. 3–53, 2019. DOI: 10.1007/978-3-030-11009-3_1.

[66] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, R. Pflugfelder, J. K. Kämäräinen, L. C. Zajc, O. Drbohlav, A. Lukezic, A. Berg, A. Eldesokey, J. Käpylä, G. Fernández, A. Gonzalez-Garcia, A. Memarmoghadam, A. D. Lu, A. F. He, A. Varfolomieiev, A. Chan, A. S. Tripathi, A. Smeulders, B. S. Pedasingu, B. X. Chen, B. P. Zhang, B. Y. Wu, B. Li, B. He, B. Yan, B. Bai, B. Li, B. Li, B. H. Kim, C. Ma, C. Fang, C. Qian, C. Chen, C. L. Li, C. Q. Zhang, C. Y. Tsai, C. Luo, C. Micheloni, C. H. Zhang, D. C. Tao, D. Gupta, D. J. Song, D. Wang, E. Gavves, E. Yi, F. S. Khan, F. Y. Zhang, F. Wang, F. Zhao, G. De Ath, G. Bhat, G. Q. Chen, G. T. Li, H. Cevikalp, H. Du, H. J. Zhao, H. Saribas, H. M. Jung, H. L. Bai, H. Y. Yu, H. Y. Yu, H. W. Peng, H. C. Lu, H. Li, J. K. Li, J. H. Li, J. L. Fu, J. Chen, G. Gao, J. Zhao, J. Tang, J. Li, J. J. Wu, J. T. Liu, J. Q. Wang, J. Q. Qi, J. Y. Zhang, J. K. Tsotsos, J. H. Lee, J. van de Weijer, J. Kittler, J. H. Lee, J. F. Zhuang, K. K. Zhang, K. K. Wang, K. N. Dai, L. Chen, L. Liu, L. D. Guo, L. Zhang, L. Wang, L. L. Wang, L. C. Zhang, L. J. Wang, L. J. Zhou, L. Y. Zheng, L. T. Rout, L. Van Gool, L. Bertinetto, M. Danelljan, M. Dunnhofer, M. Ni, M. Y. Kim, M. Tang, M. H. Yang, N. Paluru, N. Martinel, P. F. Xu, P. F. Zhang, P. K. Zheng, P. Y. Zhang, P. H. S. Torr, Q. Z. Q. Wang, Q. Guo, R. Timofte, R. K. Gorthi, R. Everson, R. Z. Han, R. H. Zhang, S. You, S. C. Zhao, S. W. Zhao, S. H. Li, S. K. Li, S. M. Ge, S. Bai, S. S. Guan, T. F. Xing, T. Y. Xu, T. Y. Yang, T. Zhang, T. Vojir, W. Feng, W. M. Hu, W. Z. Wang, W. J. Tang, W. J. Zeng, W. Y. Liu, X. Chen, X. Qiu, X. Bai, X. J. Wu, X. Y. Yang, X. E. Chen, X. Li, X. Sun, X. Y. Chen, X. M. Tian, X. Tang, X. F. Zhu, Y. Huang, Y. N. Chen, Y. C. Lian, Y. Gu, Y. Liu, Y. J. Chen, Y. Zhang, Y. D. Xu, Y. M. Wang, Y. P. Li, Y. Zhou, Y. Dong, Y. F. Xu, Y. H. Zhang, Y. K. Li, Z. W. Z. Luo, Z. L. Zhang, Z. H. Feng, Z. Y. He, Z. C. Song, Z. H. Chen, Z. P. Zhang, Z. R. Wu, Z. W. Xiong, Z. J. Huang, Z. Teng, Z. H. Ni. The seventh visual object tracking VOT2019 challenge results. In *Proceedings of IEEE/CVF International Conference on Computer Vision Workshops*, IEEE, Seoul, Korea, pp. 2206–2241, 2019. DOI: 10.1109/ICCVW.2019.00276.

[67] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, J. K. Kämäräinen, M. Danelljan, L. Č. Zajc, A. Lukežič, O. Drbohlav, L. B. He, Y. S. Zhang, S. Yan, J. Y. Yang, G. Fernández, A. Hauptmann, A. Memarmoghadam, Á. García-Martín, A. Robinson, A. Varfolomieiev, A. H. Gebrehiwot, B. Uzun, B. Yan, B. Li, C. Qian, C. Y. Tsai, C. Micheloni, D. Wang, F. Wang, F. Xie, F. J. Lawin, F. Gustafsson, G. L. Foresti, G. Bhat, G. Q. Chen, H. B. Ling, H. T. Zhang, H. Cevikalp, H. J. Zhao, H. R. Bai, H. C. Kuchibhotla, H. Saribas, H. Fan, H. Ghanei-Yakhdan, H. Q. Li, H. W. Peng, H. C. Lu, H. Li, J. Khaghani, J. Bescos, J. H. Li, J. L. Fu, J. Q. Yu, J. T. Xu, J. Kittler, J. Yin, J. Lee, K. C. Yu, K. W. Liu, K. Yang, K. N. Dai, L. Cheng, L. Zhang, L. J. Wang, L. Y. Wang, L. Van Gool, L. Bertinetto, M. Dunnhofer, M. Cheng, M. M. Dasari, N. Wang, N. Wang, P. Y. Zhang, P. H. S. Torr, Q. Wang, R. Timofte, R. K. S. Gorthi, S. Choi, S. M. Marvasti-Zadeh,

S. C. Zhao, S. Kasaei, S. M. Qiu, S. H. Chen, T. B. Schön, T. Y. Xu, W. Lu, W. M. Hu, W. G. Zhou, X. Qiu, X. Ke, X. J. Wu, X. L. Zhang, X. Y. Yang, X. F. Zhu, Y. J. Jiang, Y. M. Wang, Y. W. Chen, Y. Ye, Y. Z. Li, Y. Yao, Y. Lee, Y. Z. Gu, Z. Z. Wang, Z. Y. Tang, Z. H. Feng, Z. J. Mai, Z. P. Zhang, Z. R. Wu, Z. A. Ma. The eighth visual object tracking VOT2020 challenge results. In *Proceedings of the European Conference on Computer Vision*, Springer, Glasgow, UK, pp. 547–601, 2020. DOI: 10.1007/978-3-030-68238-5_39.

[68] Q. Wang, L. Zhang, L. Bertinetto, W. M. Hu, P. H. S. Torr. Fast online object tracking and segmentation: A unifying approach. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp. 1328–1338, 2019. DOI: 10.1109/CVPR.2019.00142.

[69] W. H. Zhang, H. R. Wang, Z. J. Huang, Y. X. Li, J. L. Zhou, L. C. Jiao. Accuracy and long-term tracking via overlap maximization integrated with motion continuity. In *Proceedings of IEEE/CVF International Conference on Computer Vision Workshops*, IEEE, Seoul, Korea, pp. 109–117, 2019. DOI: 10.1109/ICCVW.2019.00019.

[70] S. Choi, J. Lee, Y. S. Lee, A. Hauptmann. Robust long-term object tracking via improved discriminative model prediction. In *Proceedings of the European Conference on Computer Vision*, Springer, Glasgow, UK, pp. 602–617, 2020. DOI: 10.1007/978-3-030-68238-5_40.

[71] G. Zhu, F. Porikli, H. D. Li. Beyond local search: Tracking objects everywhere with instance-specific proposals. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA, pp. 943–951, 2016. DOI: 10.1109/CVPR.2016.108.

[72] C. L. Zitnick, P. Dollár. Edge boxes: Locating object proposals from edges. In *Proceedings of the 13th European Conference on Computer Vision*, Springer, Zurich, Switzerland, pp. 391–405, 2014. DOI: 10.1007/978-3-319-10602-1_26.

[73] H. Liu, Q. Y. Hu, B. Li, Y. L. Guo. Robust long-term tracking via instance-specific proposals. *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 4, pp. 950–962, 2020. DOI: 10.1109/TIM.2019.2908715.

[74] D. Q. Sun, X. D. Yang, M. Y. Liu, J. Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, USA, pp. 8934–8943, 2018. DOI: 10.1109/CVPR.2018.00931.

[75] J. Q. Wang, K. Chen, S. Yang, C. C. Loy, D. H. Lin. Region proposal by guided anchoring. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp. 2960–2969, 2019. DOI: 10.1109/CVPR.2019.00308.

[76] S. Q. Ren, K. M. He, R. Girshick, J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017. DOI: 10.1109/TPAMI.2016.2577031.

[77] I. Jung, J. Son, M. Baek, B. Han. Real-time MDNet. In *Proceedings of the 15th European Conference on Computer Vision*, Springer, Munich, Germany, pp. 89–104, 2018. DOI: 10.1007/978-3-030-01225-0_6.

[78] M. E. Maresca, A. Petrosino. MATRIOSKA: A multi-level approach to fast tracking by learning. In *Proceedings of the International Conference on Image Analysis and Processing*, Springer, Naples, Italy, pp. 419–428, 2013. DOI: 10.1007/978-3-642-41184-7_43.

[79] J. S. Supancic III, D. Ramanan. Self-paced learning for long-term tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Portland, USA, pp. 2379–2386, 2013. DOI: 10.1109/CVPR.2013.308.

[80] A. Dave, P. Tokmakov, C. Schmid, D. Ramanan. Learning to track any object. [Online], Available: https://arxiv.org/abs/1910.11844, 2019.

[81] K. M. He, G. Gkioxari, P. Dollár, R. Girshick. Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 386–397, 2020. DOI: 10.1109/TPAMI.2018.2844175.

[82] Z. K. Zhang, B. N. Zhong, S. P. Zhang, Z. J. Tang, X. Liu, Z. X. Zhang. Distractor-aware fast tracking via dynamic convolutions and MOT philosophy. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Nashville, USA, pp. 1024–1033, 2021. DOI: 10.1109/CVPR46437.2021.00108.

[83] L. H. Huang, X. Zhao, K. Q. Huang. GlobalTrack: A simple and strong baseline for long-term tracking. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, New York, USA, pp. 11037–11044, 2020. DOI: 10.1609/aaai.v34i07.6758.

[84] J. Choi, J. Kwon, K. M. Lee. Visual tracking by TridentAlign and context embedding. In *Proceedings of the 15th Asian Conference on Computer Vision*, Springer, Kyoto, Japan, pp. 504–520, 2021. DOI: 10.1007/978-3-030-69532-3_31.

[85] Z. B. Li, Q. Wang, J. Gao, B. Li, W. M. Hu. Globally spatial-temporal perception: A long-term tracking system. In *Proceedings of IEEE International Conference on Image Processing*, Abu Dhabi, UAE, pp. 2066–2070, 2020. DOI: 10.1109/ICIP40778.2020.9191319.

[86] X. Wang, Z. Chen, J. Tang, B. Luo, Y. W. Wang, Y. H. Tian, F. Wu. Dynamic attention guided multi-trajectory analysis for single object tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 12, pp. 4895–4908, 2021. DOI: 10.1109/TCSVT.2021.3056684.

[87] Y. Wu, J. Lim, M. H. Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015. DOI: 10.1109/TPAMI.2014.2388226.

[88] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, R. Pflugfelder, J. K. Kämäräinen, H. J. Chang, M. Danelljan, L. Č. Zajc, A. Lukežič, O. Drbohlav, J. Käpylä, G. Häger, S. Yan, J. Y. Yang, Z. Q. Zhang, G. Fernández, M. Abdelpakey, G. Bhat, L. Cerkezi, H. Cevikalp, S. Y. Chen, X. Chen, M. Cheng, Z. Y. Cheng, Y. C. Chiu, O. Cirakman, Y. T. Cui, K. N. Dai, M. M. Dasari, Q. Deng, X. P. Dong, D. K. Du, M. Dunnhofer, Z. H. Feng, Z. Y. Feng, Z. H. Fu, S. M. Ge, R. K. Gorthi, Y. Z. Gu, B. Gunsel, Q. Guo, F. Gurkan, W. C. Han, Y. Y. Huang, F. J. Lawin, S. J. Jhang, R. G. Ji, C. Jiang, Y. J. Jiang, F. Juefei-Xu, Y. Jun, X. Ke, F. S. Khan, B. H. Kim, J. Kittler, X. Y. Lan, J. H. Lee, B. Leibe, H. Li, J. H. Li, X. X. Li, Y. Z. Li, B. Liu, C. Liu, J. G. Liu, L. Liu, Q. J. Liu, H. C. Lu, W. Lu, J. Luiten, J. Ma, Z. Ma, N. Martinel, C. Mayer, A. Memarmoghadam, C. Micheloni, Y. Z. Niu, D. Paudel, H. W. Peng, S. M. Qiu, A. Rajiv, M. Rana, A. Robinson, H. Saribas, L. Shao, M. Shehata, F. Shen, J. B. Shen, K. Simonato, X. N. Song, Z. Y. Tang, R. Timofte, P. Torr, C. Y. Tsai, B. Uzun, L. Van Gool, P. Voigtlaender, D. Wang, G. T. Wang, L. L. Wang, L. J. Wang, L. M. Wang, L. Y. Wang, Y. Wang, Y. H. Wang, C. Y. Wu, G. S. Wu, X. J. Wu, F. Xie, T. Y. Xu, X.

Xu, W. L. Xue, B. Yan, W. K. Yang, X. Y. Yang, Y. Ye, J. Yin, C. W. Zhang, C. H. Zhang, H. T. Zhang, K. H. Zhang, K. K. Zhang, X. H. Zhang, X. L. Zhang, X. Y. Zhang, Z. B. Zhang, S. C. Zhao, M. Zhen, B. N. Zhong, J. W. Zhu, X. F. Zhu. The ninth visual object tracking VOT2021 challenge results. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, IEEE, Montreal, Canada, pp. 2711–2738, 2021. DOI: 10.1109/ICCVW54120.2021.00305.

[89] H. Fan, L. T. Lin, F. Yang, P. Chu, G. Deng, S. J. Yu, H. X. Bai, Y. Xu, C. Y. Liao, H. B. Ling. LaSOT: A high-quality benchmark for large-scale single object tracking. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp. 5369–5378, 2019. DOI: 10.1109/CVPR.2019.00552.

[90] M. Müller, A. Bibi, S. Giancola, S. Alsubaihi, B. Ghanem. TrackingNet: A large-scale dataset and benchmark for object tracking in the wild. In *Proceedings of the 15th European Conference on Computer Vision*, Springer, Munich, Germany, pp. 310–327, 2018. DOI: 10.1007/978-3-030-01246-5_19.

[91] P. Y. Zhang, J. Zhao, D. Wang, H. C. Lu, X. Ruan. Visible-thermal UAV tracking: A large-scale benchmark and new baseline. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, New Orleans, USA, 2022.

[92] T. Y. Yang, A. B. Chan. Learning dynamic memory networks for object tracking. In *Proceedings of the 15th European Conference on Computer Vision*, Springer, Munich, Germany, pp. 153–169, 2018. DOI: 10.1007/978-3-030-01240-3_10.

[93] Z. D. Wang, H. S. Zhao, Y. L. Li, S. J. Wang, P. H. S. Torr, L. Bertinetto. Do different tracking tasks require different appearance models? In *Proceedings of the 35th Conference on Neural Information Processing Systems*, pp. 726–738, 2021.

[94] A. Bewley, Z. Y. Ge, L. Ott, F. Ramos, B. Upcroft. Simple online and realtime tracking. In *Proceedings of IEEE International Conference on Image Processing*, Phoenix, USA, pp. 3464–3468, 2016. DOI: 10.1109/ICIP.2016.7533003.

[95] N. Wojke, A. Bewley, D. Paulus. Simple online and realtime tracking with a deep association metric. In *Proceedings of IEEE International Conference on Image Processing*, Beijing, China, pp. 3645–3649, 2017. DOI: 10.1109/ICIP.2017.8296962.

[96] Y. F. Zhang, C. Y. Wang, X. G. Wang, W. J. Zeng, W. Y. Liu. FairMOT: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, vol. 129, no. 11, pp. 3069–3087, 2021. DOI: 10.1007/s11263-021-01513-4.

Chang Liu received the B. Eng. degree in communication engineering from Dalian University of Technology, China in 2019. She is currently a Ph. D. degree candidate in signal and information processing at School of Information and Communication Engineering, Dalian University of Technology, China.

Her research direction is visual object tracking.

E-mail: njx2019@mail.dlut.edu.cn

ORCID iD: 0000-0002-2018-7162

Xiao-Fan Chen received the B. Eng. degree in computer science from Dalian University of Technology, China in 2017. She is currently a master student in signal and information processing at School of Information and Communication Engineering, Dalian University of Technology, China.

Her research direction is visual object tracking.

E-mail: chenxf@mail.dlut.edu.cn

Chun-Juan Bo received the Ph. D. degree in signal and information processing from Dalian University of Technology, China in 2019. She is currently an associate professor with College of Information and Communication Engineering, Dalian Minzu University, China.

Her research interests include image classification and object tracking.

E-mail: bcj@dlnu.edu.cn

Dong Wang received the B. Eng. degree in electronic information engineering and the Ph. D. degree in signal and information processing from Dalian University of Technology (DUT), China in 2008 and 2013, respectively. He is currently a full professor with School of Information and Communication Engineering, DUT, China.

His research interests focuses on object detection and tracking.

E-mail: wdice@dlut.edu.cn (Corresponding author)

ORCID iD: 0000-0002-6976-4004

Machine Intelligence Research

Citation: C. Liu, X. F. Chen, C. J. Bo, D. Wang. Long–term visual tracking: review and experimental comparison. *Machine Inteligence Research*, vol.19, no.6, pp.512–530, 2022. https://doi.org/10.1007/s11633–022–1344–1

Articles may interest you

Advances in deep learning methods for visual tracking: literature review and fundamentals. *Machine Intelligence Research*, vol.18, no.3, pp.311-333, 2021.
DOI: 10.1007/s11633-020-1274-8

Fault information recognition for on-board equipment of high-speed railway based on multi-neural network collaboration. *Machine Intelligence Research*, vol.18, no.6, pp.935-946, 2021.
DOI: 10.1007/s11633-021-1298-8

A tracking registration method for augmented reality based on multi-modal template matching and point clouds. *Machine Intelligence Research*, vol.18, no.2, pp.288-299, 2021.
DOI: 10.1007/s11633-020-1265-9

Neural decoding of visual information across different neural recording modalities and approaches. *Machine Intelligence Research*, vol.19, no.5, pp.350-365, 2022.
DOI: 10.1007/s11633-022-1335-2

Correction to: a survey on 3d visual tracking of multicopters. *Machine Intelligence Research*, vol.18, no.5, pp.855-855, 2021.
DOI: 10.1007/s11633-019-1213-8

Fmri-based decoding of visual information from human brain activity: a brief review. *Machine Intelligence Research*, vol.18, no.2, pp.170-184, 2021.
DOI: 10.1007/s11633-020-1263-y

Dense face network: a dense face detector based on global context and visual attention mechanism. *Machine Intelligence Research*, vol.19, no.3, pp.247-256, 2022.
DOI: 10.1007/s11633-022-1327-2

WeChat: MIR

Twitter: MIR_Journal