The Ninth Visual Object Tracking VOT2021 Challenge Results

Matej Kristan[1], Jiří Matas[2], Aleš Leonardis[3], Michael Felsberg[4], Roman Pflugfelder[5,6], Joni-Kristian Kämäräinen[7], Hyung Jin Chang[3], Martin Danelljan[8], Luka Čehovin Zajc[1], Alan Lukežič[1], Ondrej Drbohlav[2], Jani Käpylä[7], Gustav Häger[4], Song Yan[7], Jinyu Yang[3], Zhongqun Zhang[3], Gustavo Fernández[5], Mohamed Abdelpakey[44], Goutam Bhat[8], Llukman Cerkezi[23], Hakan Cevikalp[15], Shengyong Chen[42], Xin Chen[14], Miao Cheng[52], Ziyi Cheng[27], Yu-Chen Chiu[41], Ozgun Cirakman[23], Yutao Cui[31], Kenan Dai[14], Mohana Murali Dasari[22], Qili Deng[12], Xingping Dong[21], Daniel K. Du[12], Matteo Dunnhofer[50], Zhen-Hua Feng[49], Zhiyong Feng[43], Zhihong Fu[10], Shiming Ge[45], Rama Krishna Gorthi[22], Yuzhang Gu[38], Bilge Gunsel[23], Qing Guo[32], Filiz Gurkan[23], Wencheng Han[11], Yanyan Huang[17], Felix Järemo Lawin[4], Shang-Jhih Jhang[41], Rongrong Ji[51], Cheng Jiang[31], Yingjie Jiang[25], Felix Juefei-Xu[9], Yin Jun[52], Xiao Ke[17], Fahad Shahbaz Khan[29], Byeong Hak Kim[26], Josef Kittler[49], Xiangyuan Lan[19], Jun Ha Lee[26], Bastian Leibe[36], Hui Li[25], Jianhua Li[14], Xianxian Li[18], Yuezhou Li[17], Bo Liu[24], Chang Liu[14], Jingen Liu[24], Li Liu[37], Qingjie Liu[10], Huchuan Lu[14,33], Wei Lu[52], Jonathon Luiten[36], Jie Ma[20], Ziang Ma[52], Niki Martinel[50], Christoph Mayer[8], Alireza Memarmoghadam[46], Christian Micheloni[50], Yuzhen Niu[17], Danda Paudel[8], Houwen Peng[28], Shoumeng Qiu[38], Aravindh Rajiv[22], Muhammad Rana[49], Andreas Robinson[4], Hasan Saribas[16], Ling Shao[21], Mohamed Shehata[44], Furao Shen[31], Jianbing Shen[21], Kristian Simonato[50], Xiaoning Song[25], Zhangyong Tang[25], Radu Timofte[8], Philip Torr[47], Chi-Yi Tsai[41], Bedirhan Uzun[15], Luc Van Gool[8], Paul Voigtlaender[36], Dong Wang[14], Guangting Wang[48], Liangliang Wang[12], Lijun Wang[14], Limin Wang[31], Linyuan Wang[52], Yong Wang[40], Yunhong Wang[10], Chenyan Wu[34], Gangshan Wu[31], Xiao-Jun Wu[25], Fei Xie[39], Tianyang Xu[25,49], Xiang Xu[31], Wanli Xue[42], Bin Yan[14], Wankou Yang[39], Xiaoyun Yang[35], Yu Ye[17], Jun Yin[52], Chengwei Zhang[13], Chunhui Zhang[45], Haitao Zhang[52], Kaihua Zhang[30], Kangkai Zhang[45], Xiaohan Zhang[14], Xiaolin Zhang[38], Xinyu Zhang[14], Zhibin Zhang[42], Shaochuan Zhao[25], Ming Zhen[12], Bineng Zhong[18], Jiawen Zhu[14], and Xue-Feng Zhu[25]

[1]University of Ljubljana, Slovenia
[2]Czech Technical University, Czech Republic
[3]University of Birmingham, United Kingdom
[4]Linköping University, Sweden
[5]Austrian Institute of Technology, Austria
[6]TU Wien, Austria
[7]Tampere University, Finland
[8]ETH Zurich, Switzerland
[9]Alibaba Group, USA
[10]Beihang University, China
[11]Beijing Institute of Technology, China
[12]ByteDance, China
[13]Dalian Maritime University, China
[14]Dalian University of Technology, China
[15]Eskisehir Osmangazi University, Turkey
[16]Eskisehir Technical University, Turkey

[17]Fuzhou University, China
[18]Guangxi Normal University, China
[19]Hong Kong Baptist University, China
[20]Huaqiao University, China
[21]Inception Institute of Artificial Intelligence, China
[22]Indian Institute of Technology Tirupati, India
[23]Istanbul Technical University, Turkey
[24]JD Finance America Corporation, USA
[25]Jiangnan University, China
[26]Korea Institute of Industrial Technology (KITECH), Korea
[27]Kyushu University, Japan
[28]Microsoft Research Asia, China
[29]Mohamed Bin Zayed University of Artificial Intelligence, UAE
[30]Nanjing University of Information Science and Technology, China
[31]Nanjing University, China
[32]Nanyang Technological University, Singapore
[33]Peng Cheng Laboratory, China
[34]Penn State University, USA
[35]Remark AI, United Kingdom
[36]RWTH Aachen University, Germany
[37]Shenzhen Research Institute of Big Data, China
[38]SIMIT, China
[39]Southeast University, China
[40]Sun Yat-sen University, China
[41]Tamkang University, Taiwan
[42]Tianjin University of Technology, China
[43]Tianjin University, China
[44]University of British Columbia, Canada
[45]University of Chinese Academy of Science, China
[46]University of Isfahan, Iran
[47]University of Oxford, United Kingdom
[48]University of Science and Technology of China, China
[49]University of Surrey, United Kingdom
[50]University of Udine, Italy
[51]Xiamen University, China
[52]Zhejiang Dahua Technology CO, China

Abstract

The Visual Object Tracking challenge VOT2021 is the ninth annual tracker benchmarking activity organized by the VOT initiative. Results of 71 trackers are presented; many are state-of-the-art trackers published at major computer vision conferences or in journals in recent years. The VOT2021 challenge was composed of four sub-challenges focusing on different tracking domains: (i) VOT-ST2021 challenge focused on short-term tracking in RGB, (ii) VOT-RT2021 challenge focused on "real-time" short-term tracking in RGB, (iii) VOT-LT2021 focused on long-term tracking, namely coping with target disappearance and reappearance and (iv) VOT-RGBD2021 challenge focused on long-term tracking in RGB and depth imagery. The VOT-ST2021 dataset was refreshed, while VOT-RGBD2021 introduces a training dataset and sequestered dataset for winner identification. The source code for most of the trackers, the datasets, the evaluation kit and the results along with the source code for most trackers are publicly available at the challenge website[1].

1. Introduction

The VOT[1] initiative was founded in 2013 as a response to the lack of performance evaluation consensus in visual object tracking. The goal was to establish evaluation standards (datasets, evaluation measures and toolkits) through interaction with the tracking community. In the effort towards building the tracking community, visual object tracking challenges were created as a community-oriented interaction platform to discuss evaluation-related issues and reaching a community-wide consensus. This lead to the organization of eight challenges, which have taken place in conjunction with ICCV2013 (VOT2013 [36]), ECCV2014 (VOT2014 [37]), ICCV2015 (VOT2015 [35]), ECCV2016 (VOT2016 [34]), ICCV2017 (VOT2017 [33]), ECCV2018 (VOT2018 [32]), ICCV2019 (VOT2019 [30]) and, most recently, with ECCV2020 (VOT2020 [31]).

To promote the development of general tracking methodologies, the VOT considers single-camera, single-target, model-free, causal trackers. The *model-free* property means that the only training information provided is the ground truth target location in the first frame. *Causality* requires that the tracker does not use any future frames, prior to re-initialization, to infer the object position in the current frame.

Initially, the VOT challenges considered only short-term trackers, which are assumed not to be capable of performing successful re-detection after the target is lost. This means that the test sequences and performance evaluation protocols assume the target is always within the camera fields of view and potentially occluded by a few frames on occasions. In 2018, another tracker category called *long-term trackers* was added to broaden the spectrum of tracking problems addressed by the challenges. *Long-term* tracking means that the trackers are *required* to perform re-detection after the target has been lost and are therefore *not* reset after such an event. Datasets and evaluation protocols considering these properties were developed as well.

The VOT has been gradually evolving the datasets, performance measures and challenges. A constant philosophy in VOT has been that large datasets might be useful for training, particularly in the deep learning era, but not necessarily for testing. Instead, dataset creation and maintenance protocol has been established to produce datasets which are sufficiently small for practical evaluation yet include a variety of challenging tracking situations for in-depth analysis. In VOT2017 [33], a sequestered dataset for identification of the short-term tracking challenge winner was introduced. This dataset has been refreshed along with the public versions over the years.

Various forms of ground truth annotation have been explored. Initially, in VOT2013 [36], targets were annotated by axis-aligned bounding boxes; in VOT2014 [37] rotated bounding boxes were introduced as more accurate target location approximations. In VOT2015 [35] it was shown that subjective bounding box placement leads to non-negligible uncertainty in ground truth annotation, in particular for articulated objects. In response, in VOT2016 [34] the bounding boxes were fitted to approximate segmentation masks by minimizing a well-defined loss. In VOT2020 [31], the bounding boxes were abandoned completely in the short-term tracking challenge, and transition to precise per-frame segmentation masks has been made.

The landscape of challenges has been gradually explored in VOT. The VOT2013 [36] started with a single short-term tracking challenge. In VOT2014 [37] tracking in thermal imagery was added as VOT-TIR challenge. A push towards the development of fast and robust trackers has been made in VOT2017 [33] by the introduction of the real-time tracking challenge VOT-RT. VOT2018 [32] extended the class of trackers to long-term tracking by introducing VOT-LT, and in VOT2019 [30], the landscape was further extended by considering multi-modal tracking. In particular, RGB+thermal and RGB+depth (VOT-RGBT and VOT-RGBD) challenges were introduced.

A significant effort has been placed on performance measures and evaluation protocols. The VOT2013 [36] introduced basic weakly correlated performance measures to evaluate the accuracy and robustness as primary tracker properties, and a ranking-based methodology to identify the top performer was proposed. In VOT2015 [35], the ranking methodology was abandoned, and a new expected average

[1] http://votchallenge.net

overlap score EAO was introduced as a principled and interpretable combination of the primary scores.

The VOT2013 [36] and later short-term tracking challenges applied a reset-based protocol in which a tracker is reset upon drifting off the target. This protocol allowed the exploitation of all frames in the sequences, which was in contrast to related no-reset protocols of the time. But as trackers substantially improved over the years, the protocol no longer offered robust analysis and was revisited. VOT2020 [31] thus introduced an anchor-based evaluation protocol that produces the most stable performance evaluation results compared to related protocols yet inherits the benefits from the reset-based protocol.

In addition to short-term tracking evaluation, long-term tracking performance evaluation measures and protocols have been introduced in VOT2018 [32], which drew on prior work of [45]. These measures have not changed over the VOT editions and have consistently shown good evaluation capabilities.

This paper presents the ninth edition of the VOT challenges, the VOT2021 challenge, organized in conjunction with the ICCV2021 Visual Object Tracking Workshop, and the results obtained. In the following, we overview the challenge and participation requirements.

1.1. The VOT2021 challenge

The evaluation toolkit and the datasets are provided by the VOT2021 organizers. The participants were required to use the new Python VOT toolkit that implements the most recent evaluation protocols introduced in VOT2020 and the new datasets. The challenge opened on April 20th and closed on May 23rd. The winners of the sub-challenges were identified in late June, but not publicly disclosed. The results were presented at ICCV2021 VOT2021 workshop on 16th October. The VOT2021 challenge thus contained four challenges:

1. **VOT-ST2021 challenge**: This challenge was addressing short-term tracking in RGB images and has been running since VOT2013 with annual updates and modifications. As in VOT-ST2020, which abandoned bounding boxes, the target position was encoded by a segmentation mask.

2. **VOT-RT2021 challenge**: This challenge addressed the same class of trackers as VOT-ST2021, except that the trackers had to process the sequences in real-time. The challenge was introduced in VOT2017. Following the VOT-RT2020 abandonment of bounding boxes, the target position was encoded by a segmentation mask.

3. **VOT-LT2021 challenge**: This challenge was addressing long-term tracking in RGB images. The challenge was introduced in VOT2018. The target positions were encoded by bounding boxes.

4. **VOT-RGBD2021 challenge**: This challenge was addressing long-term tracking in RGB+depth (RGBD) imagery. This challenge was introduced in VOT2019. The target positions were encoded by bounding boxes.

The authors participating in the challenge were required to integrate their tracker into the VOT2021 evaluation kit, which automatically performed a set of standardized experiments. The results were analyzed according to the VOT2021 evaluation methodology.

Participants were encouraged to submit their own new or previously published trackers as well as modified versions of third-party trackers. In the latter case, modifications had to be significant enough for acceptance. Participants were expected to submit a single set of results per tracker. Changes in the parameters did not constitute a different tracker. The tracker was required to run with fixed parameters in all experiments. The tracking method itself was allowed to internally change specific parameters, but these had to be set automatically by the tracker, e.g., from the image size and the initial size of the bounding box, and were not to be set by detecting a specific test sequence and then selecting the parameters that were hand-tuned for this sequence.

Each submission was accompanied by a short abstract describing the tracker, which was used for the short tracker descriptions in Appendix 5 – the authors were asked to provide a clear description useful to the readers of the VOT2021 results report. In addition, participants filled out a questionnaire on the VOT submission page to categorize their tracker along various design properties. Authors had to agree to help the VOT technical committee to reproduce their results in case their tracker was selected for further validation. Participants with sufficiently well-performing submissions who contributed with the text for this paper and agreed to make their tracker code publicly available from the VOT page were offered co-authorship of this results paper. The committee reserved the right to disqualify any tracker that, by their judgement, attempted to cheat the evaluation protocols.

Methods considered for prizes in the VOT2021 challenge were not allow to learn using certain datasets (OTB, VOT, ALOV, UAV123, NUSPRO, TempleColor and RGBT234). In the case of GOT10k, a list of 1k prohibited sequences was created in VOT2019, while the remaining 9k+ sequences were allowed for learning. The reason was that part of the GOT10k was used for VOT-ST2021 dataset update.

The use of class labels specific to VOT was not allowed (i.e., identifying a target class in each sequence and applying pre-trained class-specific trackers was not allowed). An agreement to publish the code online on VOT webpage was required. The organizers of VOT2021 were allowed to participate in the challenge, but where not eligible to win. Further details are available from the challenge homepage[2].

VOT2021 goes beyond previous challenges by updating the datasets in VOT-ST and VOT-RT, and introduces a training dataset as well as sequestered dataset in the VOT-RGBD challenge. The Python VOT evaluation toolkit was updated as well.

The remainder of this report is structured as follows. Section 2 describes the performance evaluation protocols, Section 3 describes the individual challenges, Section 4 overviews the results and conclusions are drawn in Section 5. Short descriptions of the tested trackers are available in Appendix 5.

2. Performance evaluation protocols

Since VOT2018, the VOT challenges adopt the following definitions from [45] to distinguish between short-term and long-term trackers:

- **Short-term tracker** (ST_0). The target position is reported at each frame. The tracker does not implement target re-detection and does not explicitly detect occlusion. Such trackers are likely to fail at the first occlusion as their representation is affected by any occluder.

- **Short-term tracker with conservative updating** (ST_1). The target position is reported at each frame. Target re-detection is not implemented, but tracking robustness is increased by selectively updating the visual model depending on a tracking confidence estimation mechanism.

- **Pseudo long-term tracker** (LT_0). The target position is not reported in frames when the target is not visible. The tracker does not implement explicit target re-detection but uses an internal mechanism to identify and report tracking failure.

- **Re-detecting long-term tracker** (LT_1). The target position is not reported in frames when the target is not visible. The tracker detects tracking failure and implements explicit target re-detection.

Since the two classes of trackers make distinct assumptions on target presence, separate performance measures and evaluation protocols were designed in VOT to probe the tracking properties.

2.1. The short-term evaluation protocol

The short-term performance evaluation protocol entails initializing the tracker at several frames in the sequence, called the anchor points, which are spaced approximately 50 frames apart. The tracker is run from each anchor - in the first half of the sequences in the forward direction, for anchors in the second half backwards, till the first frame. Performance is evaluated by two basic measures *accuracy* (A) and *robustness* (R). Accuracy is the average overlap on frames before tracking failure, averaged over all sub-sequences. Robustness is the percentage of successfully tracked sub-sequence frames, averaged over all sub-sequences. Tracking failure is defined as the frame at which the overlap between the ground truth and predicted target position dropped below 0.1 and did not increase above this value at least 10 frames later. This definition allows short-term failure recovery in short-term trackers. The primary performance measure is the expected average overlap EAO, which is a principled combination of the tracking accuracy and robustness. Please see [31] for further details on the VOT short-term tracking performance measures.

2.2. The long-term evaluation protocol

The long-term performance evaluation protocol follows the protocol proposed in [45] and entails initializing the tracker in the first frame of the sequence and running it until the end of the sequence. The tracker is required to report the target position in each frame along with a score that reflects the certainty that the target is present at that position. Performance is measured by two basic measures called the tracking precision (*Pr*) and the tracking recall (*Re*), while the overall performance is summarized by the tracking *F*-measure. The performance measures depend on the target presence certainty threshold, thus the performance can be visualized by the tracking precision-recall and tracking *F*-measure plots obtained by computing these scores for all thresholds. The final values of *Pr*, *Re* and *F*-measure are obtained by selecting the certainty threshold that maximizes tracker-specific *F*-measure. This avoids all manually-set thresholds in the primary performance measures.

3. Description of individual challenges

In the following we provide descriptions of all five challenges running in the VOT2021 challenge.

3.1. VOT-ST2021 challenge outline

This challenge addressed RGB tracking in a short-term tracking setup. The short-term tracking performance evaluation protocol and measures outlined in Section 2 were applied. In the following, the details of the dataset and the winner identification protocols are provided.

The dataset. Results of the VOT2020 showed that the dataset was not saturated [31], and the public dataset has been refreshed by replacing 10% of the sequences (see Figure 1). Similarly the sequestered dataset has been refreshed to match the attribute distribution of the public dataset. Following the protocols from VOT2019, the list of 1000 di-

[2] http://www.votchallenge.net/vot2021/participation.html

verse sequences[3] from the GOT-10k [26] training set was used. The sequence selection and replacement procedure followed that of VOT2019. In addition, object category and motion diversity was ensured by manual review.

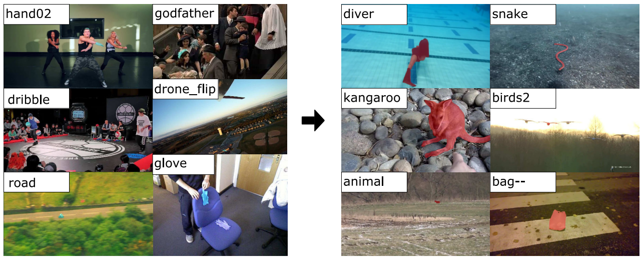

Figure 1. Six sequences in the VOT-ST2020 public dataset have been replaced for the VOT-ST2021 challenge.

Since VOT2020, the bounding boxes are no longer used in the VOT-ST/RT tracking sub-challenges. The target position is now encoded by the segmentation masks. In VOT2020, all sequences were manually segmented. The new sequences in the refreshed VOT-ST2021 dataset were segmented semi-automatically and frame-by-frame manually corrected.

Per-frame visual attributes were semi-automatically assigned to the new sequences following the VOT attribute annotation protocol. In particular, each frame was annotated by the following visual attributes: (i) occlusion, (ii) illumination change, (iii) motion change, (iv) size change, (v) camera motion.

Winner identification protocol. The VOT-ST2021 winner was identified as follows. Trackers were ranked according to the EAO measure on the public dataset. Top five ranked trackers were then re-run by the VOT2021 committee on the sequestered dataset. The top ranked tracker on the sequestered dataset not submitted by the VOT2021 committee members is the winner.

3.2. VOT-RT2021 challenge outline

This challenge addressed *real-time* RGB tracking in a short-term tracking setup. The dataset was the same as in the VOT-ST2021 challenge, but the evaluation protocol was modified to emphasize the real-time component in tracking performance. In particular, the VOT-RT2021 challenge requires predicting bounding boxes faster or equal to the video frame-rate. The toolkit sends images to the tracker via the Trax protocol [58] at 20fps. If the tracker does not respond in time, the last reported bounding box is assumed as the reported tracker output at the available frame (zero-order hold dynamic model). The same performance evaluation protocol as in VOT-ST2021 is then applied.

Winner identification protocol. All trackers are ranked on the public RGB short-term tracking dataset with respect to the EAO measure. The winner was identified as the top ranked tracker not submitted by the VOT2021 committee members.

3.3. VOT-LT2021 challenge outline

This challenge addressed RGB tracking in a long-term tracking setup and is a continuation of the VOT-LT2020 challenge. We adopt the definitions from [45], which are used to position the trackers on the short-term/long-term spectrum. A long-term performance evaluation protocol and measures from Section 2 were used to evaluate tracking performance on VOT-LT2021.

The dataset. Trackers were evaluated on the LTB50 [45], the same dataset as used in VOT-LT2020. The LTB50 dataset contains 50 challenging sequences of diverse objects (persons, cars, motorcycles, bicycles, boats, animals, etc.) with a total length of 215294 frames. Sequence resolutions range between 1280 × 720 and 290 × 217. Each sequence contains on average 10 long-term target disappearances, each lasting on average 52 frames. The targets are annotated by axis-aligned bounding boxes. Sequences are annotated by the following visual attributes: (i) Full occlusion, (ii) Out-of-view, (iii) Partial occlusion, (iv) Camera motion, (v) Fast motion, (vi) Scale change, (vii) Aspect ratio change, (viii) Viewpoint change, (ix) Similar objects. Note this is per-sequence, not per-frame annotation and a sequence can be annotated by several attributes. Please see [45] for more details.

Winner identification protocol. The VOT-LT2021 winner was identified as follows. Trackers were ranked according to the tracking F-score on the LTB50 dataset (no sequestered dataset available). The top ranked tracker on the dataset not submitted by the VOT2021 committee members was the winner of the VOT-LT2021 challenge.

3.4. VOT-RGBD2021 challenge outline

This challenge addressed long-term trackers using the RGB and depth channels (RGBD). The long-term performance evaluation protocol from Section 2 was used. Since its introduction in 2019, the VOT-RGBD challenge has used the 80 sequences of the CDTB [43] dataset to evaluate the submitted trackers. For the 2021 challenge a new dataset was collected.

The new dataset contains 70 sequences for training and 50 for testing. The test set was kept as sequestered data not revealed during the competition. The new sequences were captured with an Intel RealSense 415 RGBD camera that provides geometrically aligned RGB and Depth frames. The main difference between CDTB and the new data is

[3] http://www.votchallenge.net/vot2019/res/list0_prohibited_1000.txt

that the new sequences contain a more diverse set of targets and many sequences were selected to particularly challenge RGB.

Winner identification protocol. The VOT-RGBD2021 winner was identified as follows. Trackers were ranked according to the F-score on the public VOT-RGBD2021 dataset (CDTB). Top three ranked trackers were then re-run by the VOT2021 committee on the sequestered dataset. The top ranked tracker on the sequestered dataset not submitted by the VOT2021 committee members was the winner of the VOT-RGBD2021 challenge.

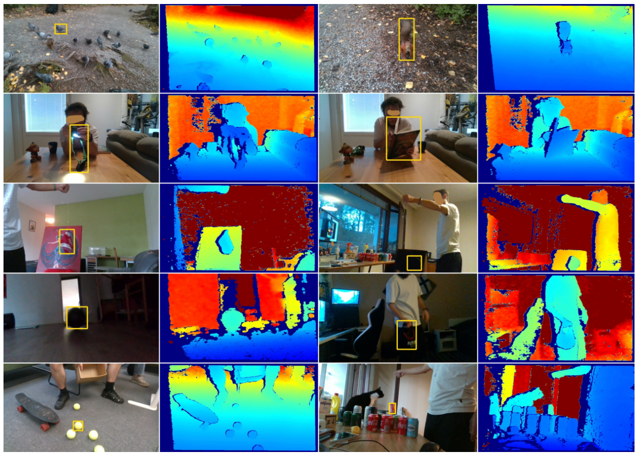

Figure 2. RGB and depth (D) frames from the VOT-RGBD sequestered dataset.

4. The VOT2021 challenge results

This section summarizes the trackers submitted, results analysis and winner identification for each of the five VOT2021 challenges.

4.1. The VOT-ST2020 challenge results

4.1.1 Trackers submitted

The VOT-ST2020 challenge tested 53 trackers, including the baselines contributed by the VOT committee. Each submission included the binaries or source code that allowed verification of the results if required. In the following we briefly overview the entries and provide the references to original papers in the Appendix A where available.

Of the participating trackers, 23 trackers (43%) were categorized as ST_0, 23 trackers (43%) as ST_1, 6 (11%) as LT_0 and 1 as LT_1. 91% applied discriminative and 9% applied generative models. Most trackers (79%) used holistic model, while 21% of the participating trackers used part-based models. Most trackers applied a equally probably displacement within a region centered at the current position[4] or a random walk dynamic model (97%) and only (3%) applied a nearly-constant-velocity or higher-order dynamic model. 40% of trackers localized the target in a single stage, while the rest applied several stages, typically involving approximate target localization and position refinement. Most of the trackers (90%) use deep features. 45% of these trackers re-trained their backbone on tracking or segmentation/detection datasets. This is the second year that the VOT short-term challenge considers target location encoded as a segmentation mask. We observe an increased trend towards developing trackers with segmentation outputs: 64% of trackers reported target position as a segmentation mask, while the rest (36%) reported a bounding box.

The trackers were based on various tracking principles. Four trackers were based on classical discriminative correlation filters (TCLCF A.9, FSC2F A.29, CSRDCF A.46, KCF A.47) and 21 trackers applied deep discriminative correlation filters (D3Sv2 A.31, SAMN A.5, D3S A.30, AR_ATOM A.39, TRATMask A.34, KYS A.48, ATOM A.45, CFRPT A.28, RPT A.1, fRPT A.19, RPT_AR A.10, LWL_B2S A.50, AR_SuperDiMP-50 A.43, LWL A.3, SAMN_DiMP A.6, keep_track A.35, AR_KYS A.41, AR_DiMP-50 A.40, AR_PrDiMP-50 A.42, PrDiMP-50 A.51, DiMP A.2). Ten trackers were based purely on Siamese correlation (SiamFc A.52, SiamUSCP A.20, SiamUSC A.12, SiamEM_R A.21, DCDAAR A.13, ACM A.11, NSpaceRDAR A.23, deepmix A.14, SION A.7, SiamFc A.52), while three combined Siamese correlation with the deep discriminative correlation filters (RPTMask A.36, TRASFUSTm A.4, AlphaRef A.15). Nine trackers were based on transformers (DualTFRon A.32, DualTFRst A.27, DualTFR A.26, RTT A.33, STARK_RT A.17, F_TregPlus A.24, TregPlus A.25, TransT_M A.22, TransT A.18). One tracker was based on generative adversarial networks VITAL++ A.8, one was a state-of-the-art video segmentation method STM A.53, one was meta-learning-based (ReptileFPN A.16) one entry was a sparse subspace tracker (L1APG A.49), one was a scale-adaptive mean-shift tracker (ASMS A.44) and two were part-based generative trackers (ANT A.38, LGT A.37).

In summary, we observe a continued popularity of the discriminative correlation filters (used in 56% of trackers) and Siamese correlation (used in 25% trackers), while we observe a rise of a new class of trackers based on transformers (17%), which are an emerging methodology in the wider filed of computer vision.

[4] The target was sought in a window centered at its estimated position in the previous frame. This is the simplest dynamic model that assumes all positions within a search region contain the target have equal prior probability.

4.1.2 Results

The results are summarized in the AR-raw plots and EAO plots in Figure 3, and in Table 6. The top ten trackers according to the primary EAO measure (Figure 3) are RPT-Mask A.36, CFRPT A.28, TransT_M A.22, TregPlus A.25, DualTFRon A.32, F_TregPlus A.24, DualTFRst A.27, STARK_RT A.17, DualTFR A.26 and RPT A.1. RPT-Mask and CFRPT are extensions of RPT – the winner of VOT-ST2020. The other trackers (TransT_M, DualTFRon, DualTFR, DualTFRst, TregPlus, F_TregPlus, STARK_RT) are based on transformers. DualTFRon, DualTFR and DualTFRst are variations of the same transformer-based tracker with bounding box prediction and Alpharef A.15 postprocessing step for target segmentation. TregPlus and F_TregPlus are variants of the same two-stage tracker that localizes the target by deep correlation filter and refines/segments it by a transformer. STARK_RT uses transformer for localization and Alpharef A.15 for segmentation, while TransT_M is a single-stage target segmentation transformer. Out of ten, five trackers thus apply Alpharef A.15 for target segmentation and one applies segmentation inspired by Alpharef.

The top performer on the public dataset is RPT-Mask A.36. This is a two-stage tracker based on the VOT-ST2020 winner RPT [47]. In the first stage, ATOM [16] coarsely localizes the target and estimates the bounding box by RPT [47]. The second stage involves improved video object segmentation method STM [51] to predict the target segmentation.

The second-best ranked tracker is CFRPT A.28, which is also based on the VOT-ST2020 winner RPT [47]. The novelty lies in a customized feature extractor, which is based on predicting the localization uncertainties of the extreme points and improving the feature extraction at these. Authors report significant performance gain in target regression. Another extensions is replacement of convolutional layers by coord-conv layers [41] and application of Alpharef A.15 for final target segmentation.

The third best-ranked tracker is TransT_M A.22, which is based on a transformer network. The tracker applies a ResNet50 with feature fusion based on ego-context self attention and classification/regression branches. In addition, a segmentation head combines the backbone and feature fusion intermediate outputs into a target segmentation mask. This transformer applies a static template and an adaptive one, with adaptation intensity controlled by an IoU-net like block.

The top four trackers in EAO (RPTMask, CFRPT, TransTM, TregPlus) stand out from the rest – the reason that they attain a high robustness as well as accuracy. RPTMask and CFRPT are two-stage trackers with separate segmentation post-processing steps, while TransTM and TregPlus integrate the segmentation as part of the single tracking architecture. All these trackers substantially outperform the VOT-ST2020 winner (RPT). The top ranked tracker RPT-Mask is negligibly less robust than RPT (1%), but outperforms it by 10% in accuracy, resulting in a healthy 8% boost in EAO.

Interestingly, a recent state-of-the-art video object segmentation (VOS) method STM A.53 [51] performs quite well compared to some of the recent state of the art trackers. This tracker outperforms 75% of submissions in accuracy, but falls behind in robustness (over 80% of the submissions outperform it). Overall, in the EAO score, this tracker is outperformed by 64% of submissions, which emphasizes the importance of robustness in tracking-centric video object segmentation task considered in VOT.

The trackers which have been considered as baselines or state-of-the-art in early, even recent, years of VOT challenges (e.g., SiamFc, KCF, L1APG, CSRDCF, ASMS) are positioned at the far end of the rank list. Even some more recent trackers like D3S [44] and a modified, segmentation-equipped KYS [3], are positioned at the middle of the rank list. This is a testament to the remarkable pace of development witnessed in visual object tracking over the last decade. Note that seven of the tested trackers have been published in major computer vision conferences and journals in the last two years (2020/2021). These trackers are indicated in Figure 3, along with their average performance (EAO= 0.424), which constitutes the VOT2021 state-of-the-art bound. Approximately 54% of submitted trackers exceed this bound, which further supports the remarkable pace of development.

	CM	**IC**	**OC**	**SC**	**MC**
Accuracy	0.59②	0.59③	0.50①	0.60	0.60
Robustness	0.75	0.71②	0.66①	0.74	0.72③

Table 1. VOT-ST2021 tracking difficulty with respect to the following visual attributes: camera motion (CM), illumination change (IC), motion change (MC), occlusion (OC) and size change (SC).

The per-attribute robustness analysis is shown in Figure 4 for individual trackers. The overall top performers remain at the top of per-attribute ranks as well. TransT_M appears be least affected by most attributes in terms of robustness, while several trackers outperform this tracker under normal circumstances (*no degradation*). According to the median failure over each attribute (Table 1) the most challenging attributes remains occlusion. The drop on this attribute is consistent for all trackers (Figure 4).

4.1.3 The VOT-ST2021 challenge winner

Top five trackers from the baseline experiment (Table 6) were re-run on the sequestered dataset. Their scores obtained on sequestered dataset are shown in Table 2. The top

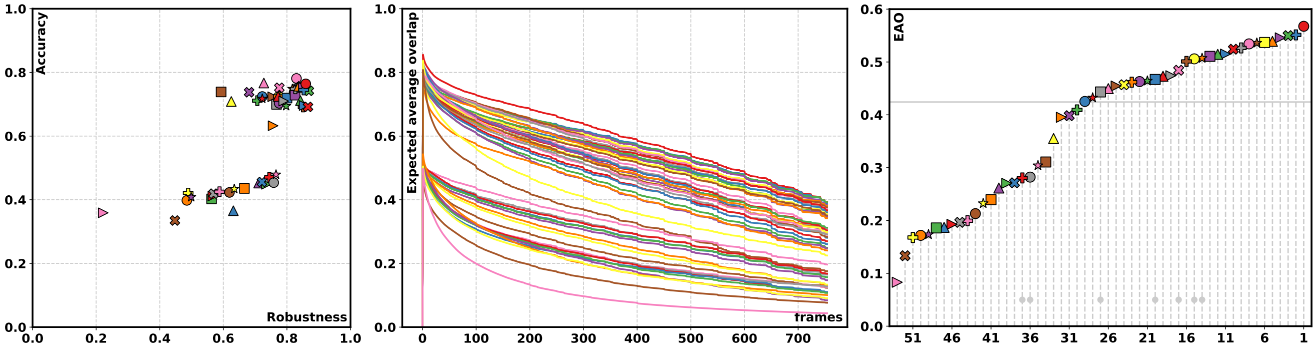

Figure 3. The VOT-ST2021 AR-raw plots generated by sequence pooling (left) and EAO curves (center) and the VOT-ST2021 expected average overlap graph with trackers ranked from right to left. The right-most tracker is the top-performing according to the VOT-ST2021 expected average overlap values. The dashed horizontal line denotes the average performance of ten state-of-the-art trackers published in 2020/2021 at major computer vision venues. These trackers are denoted by gray circle in the bottom part of the graph.

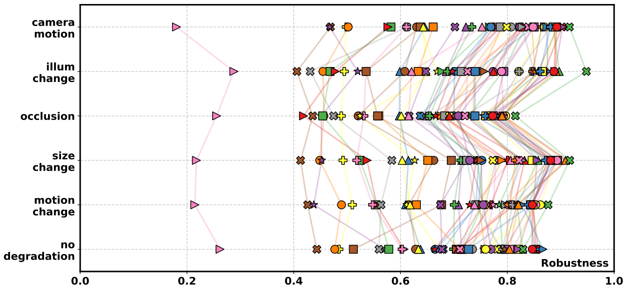

Figure 4. Robustness with respect to the visual attributes.

tracker according to the EAO is RPTMask A.36 and is thus the VOT-ST2021 challenge winner.

	Tracker	EAO	A	R
1.	RPTMask	0.505①	0.831	0.834
2.	TregPlus	0.496②	0.817	0.830
3.	CFRPT	0.486③	0.820	0.820
4.	DualTFRon	0.473	0.826	0.798
5.	TransT_M	0.470	0.812	0.808

Table 2. The top five trackers from Table 6 re-ranked on the VOT-ST2021 sequestered dataset.

4.2. The VOT-RT2021 challenge results

4.2.1 Trackers submitted

The trackers that entered the VOT-ST2021 challenge were also run on the VOT-RT2021 challenge. Thus the statistics of submitted trackers were the same as in VOT-ST2021. For details please see Section 4.1.1 and Appendix A.

4.2.2 Results

The EAO scores and AR-raw plots for the real-time experiments are shown in Figure 5 and Table 6. The top ten real-time trackers are TransT_M A.22, STARK_RT A.17, DualTFRst A.27, DualTFR A.26, TransT A.18, fRPT A.19, F_TregPlus A.24, AlphaRef A.15, LWL_B2S A.50 and SiamUSCP A.20.

The top two trackers, TrasT_M and STARK_RT are ranked 3rd and 8th in the VOT-ST2021 challenge, respectively. These two stand out from the rest in terms of EAO, which is primarily due to robustness. The trackers are not hampered by the processing time constraint of the VOT-RT2021 challenge – their realtime performance scores are nearly the same as their VOT-ST2021 scores. The top performer TrasT_M outperforms last year's winner of VOT-RT2020 Alpharef A.15 primarily in robustness (by 11%). Combined with better accuracy, this amounts to a marked 15% increase in EAO.

Five out of top seven real-time trackers are among the top six performers on VOT-ST2021 challenge. Top short-term trackers no longer sacrifice speed for performance. This is a new trend observed first in VOT2020 – before then, the top performers from VOT-ST challenge used to substantially drop in ranks under the realtime constraint. Notably, six out of top ten real-time trackers are based on transformers, which may indicate a new fast tracking framework with considerable tracking robustness and accuracy. We have to note that several top realtime trackers are variations of the same tracker. Nevertheless, while Siamese tracking architectures were dominating the VOT top ten realtime trackers lists, only a single Siamese tracker (SiamUSCP) remains among the top 10 this year.

Seven trackers (TransT_M, STARK_RT, DualTFRst, DualTFR, TransT, fRPT and F_TregPlus) outperform Alpharef, which is still ranked remarkably high (8th place). This shows that the real-time performance bar has been substantially pushed forward this year. Like in VOT-ST2021 challenge, seven of the tested trackers have been published in major computer vision conferences and journals in the last two years (2020/2021). These trackers are indicated in Figure 5, along with their average performance (EAO= 0.421), which constitutes the VOT2021 realtime state-of-the-art bound. Approximately 28% of submitted trackers exceed this bound, which is much lower than in the VOT-ST2021 challenge.

4.2.3 The VOT-RT2021 challenge winner

According to the EAO results in Table 6, the top performer and the winner of the real-time tracking challenge VOT-RT2021 is TransT_M (A.22).

4.3. The VOT-LT2020 challenge results

4.3.1 Trackers submitted

The VOT-LT2021 challenge received 10 valid entries. The VOT2021 committee contributed additional tracker SuperDiMP and the top-two performers from VOT-LT2020 as baselines; thus 13 trackers were considered in the challenge. In the following, we briefly overview the entries and provide the references to original papers in Appendix B where available.

All participating trackers were categorized as ST_1 according to the ST-LT taxonomy from Section 2 in that they implemented explicit target re-detection. All trackers were based on convolutional neural networks. Several trackers applied Transformer architecture akin to STARK [66] for target localization (mlpLT B.1, STARK_RGBD_LT B.5, STARK_LT B.7). Particularly, STARK_RGBD_LT B.5 is based purely on a Transformer-backbone [57] for feature extraction. Six trackers applied SuperDiMP structure [2] as their basic tracker (SLOT B.4, RincTrack B.8, keep_track_lt B.9, SuperDMU B.10, SuperDiMP B.11, LTMUB B.13). Three trackers are based on Region Proposal Network (RPN) for approximate target localization at detection stage (SION_LT B.2, TDIOT B.6, LTDSE B.12). One tracker is based on Siamese Network for tracking as well as for re-detection after long occlusions (SiamRCNN B.3). Five trackers combined different tracking methods and switched them based on their tracking scores (mlpLT B.1, STARK_RGBD_LT B.5, TDIOT B.6, STARK_LT B.7, RincTrack B.8). SLOT B.4 proposed a self-labeling method for tracking reliability classification and keep_track_lt B.9 proposed a self-supervised training strategy for modeling distractors objects.

4.3.2 Results

The overall performance is summarized in Figure 6 and Table 3. The top-three performers are mlpLT B.1, STARK_LT B.7 and STARK_RGBD_LT B.5. mlpLT obtains the highest F-score (0.735) in 2021, while last year winner (LT_DSE) obtains 0.695. It should be noted that the top 5 trackers in 2021 have surpassed last year winner (LT_DSE). All the results are based on the submitted numbers, but these were verified by running the codes multiple times.

The mlpLT is composed of a Transformer-based STARK tracker, meta-updater controlled SuperDiMP tracker and an online learned target verifier. STARK and SuperDiMP are run parallel for target localizations, and the decision strategy selects whose localization can be accepted based on the evaluations from the verifier. Additional strategies such as the computation of adaptive search areas, and the avoidance of wrong target size estimations, have also been implemented.

The STARK_LT architecture applies STARK tracker based on Transformer for target tracking in the local region and a global search algorithm based on GlobalTrack for the re-detection in the whole image. The global module is trained offline based on Siamese Network transferred from the detection model to find all the possible candidates of targets. The Kalman Filter and data association are also utilized to suppress the potential distractors.

The tracker STARK_RGBD_LT also applies Transformer-based STARK as their base tracker but change its backbone from ResNet50 to DeiT. The target position is then refined by the AlphaRefine method. When the score of STARK is low, the SuperDiMP tracker will take over for better tracking.

mlpLT achieves an overall best F-score and significantly surpasses STARK_LT (by 1.2%) and STARK_RGBD_LT (by 1.4%). All of these methods are based on Transformer. Figure 6 shows tracking performance with respect to nine visual attributes from Section 3.3. The most challenging attributes are fast motion, partial and full occlusion and target leaving the field of view (out-of-view attribute).

Tracker	Pr	Re	F-Score	Year
●mlpLT	0.741①	0.729①	0.735①	2021
✚STARK_LT	0.721	0.725②	0.723②	2021
✖STARK_RGBD_LT	0.719	0.724③	0.721③	2021
▶SLOT	0.727③	0.711	0.719	2021
▲keep_track_lt	0.725	0.700	0.712	2021
■SuperD_MU	0.738②	0.680	0.708	2021
★RincTrack	0.717	0.696	0.707	2021
●LT_DSE	0.715	0.677	0.695	2019
✚LTMU_B	0.698	0.680	0.689	2020
✖SuperDiMP	0.675	0.660	0.667	2021
▶SiamRCNN	0.654	0.673	0.664	2021
▲SION_LT	0.640	0.456	0.533	2021
■TDIOT	0.496	0.478	0.487	2021

Table 3. List of trackers that participated in the VOT-LT2021 challenge along with their performance scores (Pr, Re, F-score) and ST/LT categorization

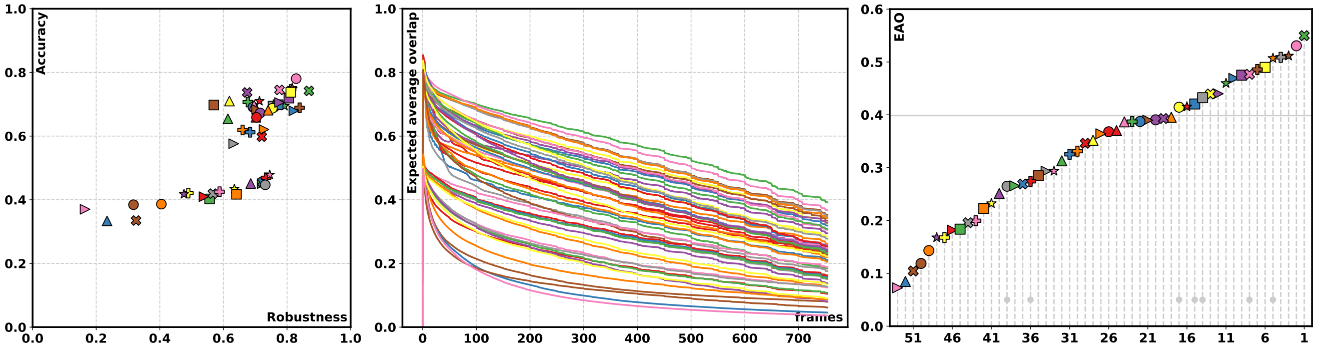

Figure 5. The VOT-RT2021 AR plot (left), the EAO curves (center) and the EAO plot (right). The dashed horizontal line denotes the average performance of ten state-of-the-art trackers published in 2020/2021 at major computer vision venues. These trackers are denoted by gray circle in the bottom part of the graph.

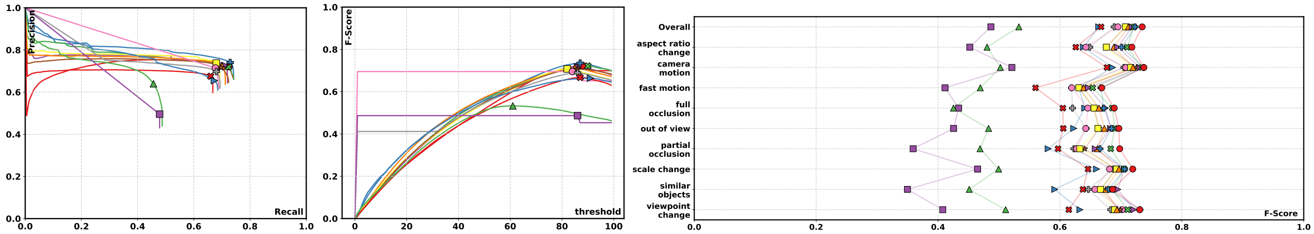

Figure 6. VOT-LT2021 challenge average tracking precision-recall curves (left), the corresponding F-score curves (middle) and per attribute F-scores (right). Tracker labels are sorted according to maximum of the F-score (see Table 3).

4.3.3 The VOT-LT2021 challenge winner

According to the F-score in Table 3, the top-performing tracker is mlpLT, closely followed by STARK_LT and STARK_RGBD_LT. Thus the winner of the VOT-LT 2021 challenge is mlpLT B.7.

4.4. The VOT-RGBD2021 challenge results

The public dataset of the RGBD 2021 challenge was the same as in the two previous years, but this time additional evaluation was conducted with a sequestered dataset of 50 new sequences.

4.4.1 Trackers submitted

The VOT-RGBD2021 challenge received five valid submissions: sttc_rgbd (see appendix C.1), STARK_RGBD (C.2), SLMD (C.3), TALGD (C.4) and DRefine (C.5). In addition to the submitted trackers the three best RGBD trackers from the 2020 competition were evaluated: ATCAIS (1st in 2020), DDiMP (2nd) and CLGS_D (3rd).

sttc_rgbd uses the transformer architecture (STARK) adopted from [66] with the addition that bounding box predictions are enhanced by correlation calculations. STARK_RGBD combines STARK and DiMP [2] and uses the backbone from [57] and target refinement module (AlphaRefine) from [67]. SLMD, TALGD and DRefine are DiMP [2] based trackers. SLMD adds spot-light masking, TALGD adds HTC [7] to detect background distractors and DRefine adds AlphaRefine to fine-tune the result.

It is noteworthy that similar to the previous year the submitted trackers are based on state-of-the-art deep RGB trackers, MDNet, ATOM, DiMP and STARK (new in 2021). It seems that the depth is mainly used for long-term tracking purposes such as detection of occlusion.

4.4.2 Results

The results for the 2021 RGBD submissions and the three best RGBD trackers from 2020 are shown in Figure 7 and Table 4 for the public VOT2021-RGBD dataset (CDTB). A clear winner is STARK_RGBD that obtains the best precision, recall and F-score. The second best tracker is TALGD that is the second best on all measures. DRefine is the third in recall and F-score, but is behind the 2020 tracker CLGS_D in precision. Two of the submitted trackers, SLMD and sttc_rgbd, perform worse than the three best methods from 2020. There is clear improvement from the last year winner (ATCAIS) to this year best (STARK_RGBD). However, the plots in Figure 7 show only small difference that indicates performance saturation with the CDTB dataset.

Additionally, we computed the results for the se-

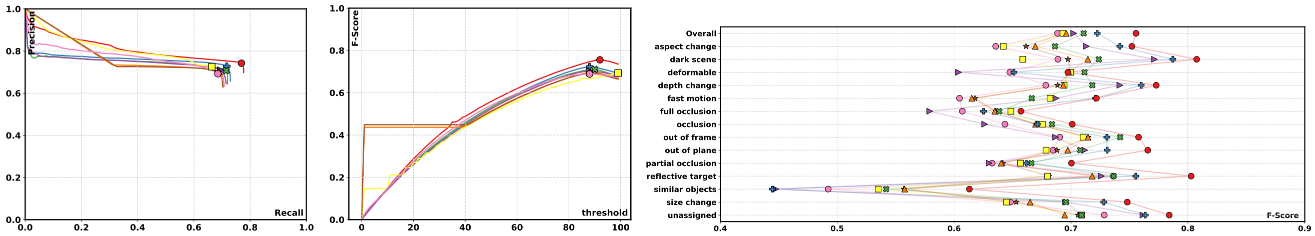

Figure 7. RGBD results for the public test set (CDTB): precision-recall (left), F-Score (middle) and per attribute F-scores (right). Note that the tracker marker depends on its ranking (see Table 4).

Tracker	Pr	Re	F-Score	Year
●STARK_RGBD	0.742①	0.769①	0.756①	2021
✚TALGD	0.728②	0.717②	0.722②	2021
✖DRefine	0.707	0.715③	0.711③	2021
▶ATCAIS	0.709	0.696	0.702	2020
△DDiMP	0.703	0.689	0.696	2020
□CLGS_D	0.725③	0.664	0.693	2020
★SLMD	0.701	0.685	0.693	2021
○sttc_rgbd	0.692	0.685	0.689	2021

Table 4. Results of the five submitted RGBD trackers for the public VOT 2021 RGBD test data (CDTB). The numbers were computed using the user provided data. The three additional trackers are the three best from the last year (2020).

Tracker	Pr	Re	F-Score	Year
●STARK_RGBD	0.558②	0.543①	0.550①	2021
✚TALGD	0.540③	0.482②	0.509②	2021
✖DDiMP	0.505	0.470③	0.487③	2020
▶ATCAIS	0.491	0.451	0.470	2020
△CLGS_D	0.585①	0.370	0.453	2020
□DRefine	0.468	0.432	0.449	2021

Table 5. RGBD results of the best three submitted trackers for 2021 sequestered RGBD sequences. The other tree trackers are from the previous year.

questered dataset (Figure 8 and Table 5). There is a notable drop in precision, recall and F-score numbers from the public CDTB to sequestered RGBD data. For example, the STARK_RGBD F-score drops from 0.756 to 0.550. There is also more performance variation between the trackers. The ranking of the two best methods remains the same. STARK_RGBD obtains the highest and TALGD the second highest F-score. The third and fourth places go to the last year trackers DDiMP and ATCAIS.

It is noteworthy that the best submitted tracker, STARK_RGBD, is the best on most of the attribute specific sequences whose F-scores are plotted in in Figures 7 and 8.

4.4.3 The VOT-RGBD2021 challenge winner

The winner of the VOT-RGBD2021 challenge is STARK_RGBD (C.2), which obtains the best F-score on the both public and sequestered datasets.

5. Conclusion

Results of the VOT2021 challenge were presented. The challenge is composed of the following four challenges focusing on various tracking aspects and domains: (i) the VOT2021 short-term RGB tracking challenge (VOT-ST2021), (ii) the VOT2021 short-term real-time RGB tracking challenge (VOT-RT2021), (iii) the VOT2021 long-term RGB tracking challenge (VOT-LT2021) and (iv) the VOT2021 long-term RGB and depth (D) tracking challenge (VOT-RGBD2021). In this edition, the VOT-ST2021 challenge datasets have been refreshed, while the VOT-RGBD2021 introduced training and sequestered datasets.

Similar to previous years, the most popular methodologies in VOT-ST2021 submissions are discriminative correlation filters (57% of submissions) and Siamese correlations (25% of submissions). We observe a rise of a new methodology among the top performers – transformers – which constitute approximately 17% of the tested trackers. The four top trackers in VOT-ST2021 stand out from the rest in performance and are either based on deep discriminative correlation filters or transformers. We observe that five of the top real-time trackers (VOT-RT2020 challenge) are among the six top-performers on the VOT-ST2021. This is a continuation of the trend observed in VOT2020, which is emergence of deep learning architectures that no longer sacrifice the speed for tracking accuracy (assuming a sufficiently powerful GPU is available). Six of the top seven VOT-RT2021 realtime trackers are based on transformers, which indicates the potential for this methodology in realtime tracking. As in VOT2020 short-term challenges, occlusion remains the most difficult attribute.

The VOT-LT2021 challenge's top-three performers all apply Transformer-based tracker structure for short-term localization and long-term re-detection. Among all submitted trackers, the dominant methodologies are SuperDiMP [2] and STARK [66], region proposals and self-supervised training strategy.

All trackers submitted to the VOT-RGBD2020 challenge are based on the SotA deep RGB trackers (DiMP and STARK) and add additional processing for the depth chan-

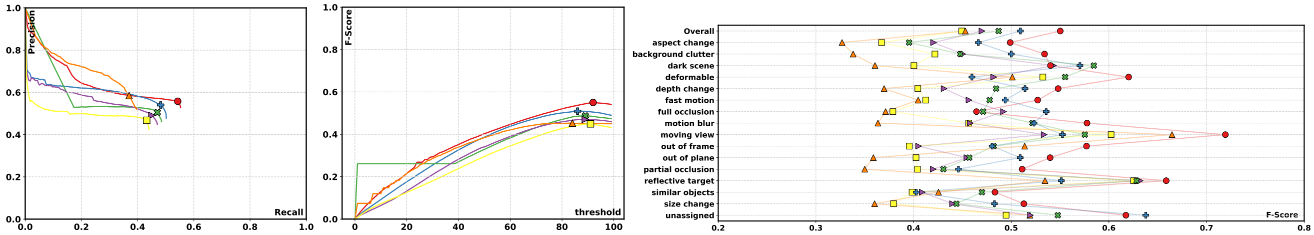

Figure 8. RGBD results for the sequestered test set: precision-recall (left), F-Score (middle) and per attribute F-scores (right). Note that the tracker marker depends on its ranking (see Table 5).

nel to support long-term tracking. In this sense, the 2021 submissions are similar to the previous year.

The top performer on the VOT-ST2021 *public dataset* is RPTMask A.36. This is a two-stage tracker, a variation of the VOT-ST2020 winner with improved localization and segmentation stage. On the public set, this tracker demonstrates a clear advantage over the second-best ranked tracker. RPTMask is also the top performer on the sequestered dataset and is thus a clear winner of the VOT-ST2021 challenge.

The top performer and the winner of the VOT-RT2021 challenge is TransT_M A.22. This is a transformer-based tracker with double visual model updated at different timescales (one fixed another updated conservatively), whose performance drop under realtime constraint is negligible compared to the VOT-ST2021 challenge. In fact, this tracker is ranked 3rd on the VOT-ST2021 challenge.

The top perfomer and the winner of the VOT-LT2021 is mlpLT B.1, which fuses the Transformer-based STARK with meta-updater controlled SuperDiMP due to their complementary features. This tracker obtains a significantly better performance than the second-best tracker.

The top performer and the winner of the VOT-RGBD2020 challenge is STARK_RGBD (C.2) that obtained the best F-score on the both public and sequestered datasets. Moreover, STARK_RGBD performed best on most of the attribute specific sequences that illustrates its strong overall performance for different types of scenes and objects.

The VOT primary objective is to establish a platform for discussion of tracking performance evaluation and contributing to the tracking community with verified annotated datasets, performance measures and evaluation toolkits. The VOT2021 was the ninth effort toward this, following the very successful VOT2013, VOT2014, VOT2015, VOT2016, VOT2017, VOT2018, VOT2019 and VOT2020.

This VOT edition continues a transition to a fully segmented ground truth. At this point, most of the top trackers are two-stage oriented with segmentation treated as a separate step. We hope to see in future a transition to streamlined direct segmentation trackers which will further narrow the gap between video object segmentation and tracking objects in challenging scenarios. In future editions we expect more sub-challenges to follow this direction, depending on manpower, as producing high-quality segmentation ground truth requires substantial efforts.

Acknowledgements

This work was supported in part by the following research programs and projects: Slovenian research agency research program P2-0214 and projects J2-2506, J2-9433, Z2-1866. The challenge was sponsored by the Faculty of Computer Science, University of Ljubljana, Slovenia. This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) and the Berzelius cluster at NSC, both funded by the Knut and Alice Wallenberg Foundation, as well as by ELLIIT, a strategic research environment funded by the Swedish government and Institute of Information and communications Technology Planning and evaluation (IITP) grant funded by the Korea government (MSIT) (2021-0-00537). Roman Pflugfelder and Gustavo Fernández were supported by the AIT Strategic Research Programme 2021 Visual Surveillance and Insight. Paul Voigtlaender, Jonathon Luiten and Bastian Leibe were supported by ERC Consolidator Grant DeeViSe (ERC-2017-COG-773161) and a Google Faculty Award.

A. VOT-ST2021 and VOT-RT2021 submissions

This appendix provides a short summary of trackers considered in the VOT-ST2021 and VOT-RT2021 challenges.

A.1. RPT: Learning Point Set Representation for Siamese Visual Tracking (RPT)

H. Zhang, L. Wang, Z. Ma, W. Lu, J. Yin, M. Cheng
1067166127@qq.com, {wanglinyuan, kobebean, lwhfh01}@zju.edu.cn, {yin_jun, cheng_miao}@dahuatech.com

RPT is formulated with a two-stage structure in series. The first stage is composed with two parallel subnets, one primarily accounting for target estimation with RepPoints [69] in an offline-trained embedding space, the other trained online to provide high robustness against distractors [16]. The target estimation head is constructed with Siamese-based feature extraction and matching. For the second stage, the set of RepPoints with highest confidence (i.e. online classification score) is fed into a modified D3S [44] to obtain the segmentation mask. The backbone is ResNet50 [24] pre-trained on ImageNet, while the target estimation head is trained using pairs of frames from YouTube-Bounding Box [53], COCO [40] and ImageNet VID [55] datasets.

A.2. Learning Discriminative Model Prediction for Tracking (DiMP)

G. Bhat, M. Danelljan, L. Van Gool, R. Timofte
{*goutam.bhat, martin.danelljan, vangool, timofter*}*@vision.ee.ethz.ch*

DiMP is an end-to-end tracking architecture, capable of fully exploiting both target and background appearance information for target model prediction. The target model here constitutes the weights of a convolution layer which performs the target-background classification. The weights of this convolution layer are predicted by the target model prediction network, which is derived from a discriminative learning loss by applying an iterative optimization procedure. The online learned target model is applied in each frame to perform target-background classification. The final bounding box is then estimated using the overlap maximization approach as in [16]. See [2] for more details about the tracker.

A.3. Learning What to Learn for Video Object Segmentation (LWL)

G. Bhat, F. Järemo Lawin, M. Danelljan, A. Robinson, M. Felsberg, L. Van Gool, R. Timofte
{*goutam.bhat, martin.danelljan, vangool, timofter*}*@vision.ee.ethz.ch*, {*felix.jaremo-lawin, andreas.robinson, michael.felsberg*}*@liu.se*

LWTL is an end-to-end trainable video object segmentation VOS architecture which captures the current target object information in a compact parametric model. It integrates a differentiable few-shot learner module, which predicts the target model parameters using the first frame annotation. The learner is designed to explicitly optimize an error between target model prediction and a ground truth label, which ensures a powerful model of the target object. In order to guide the learner to focus on the most crucial aspect of the target, a network module is trained to predict spatial importance weights for different elements in the few-shot learning loss. Since the optimization-based learner is differentiable, all modules in the architecture are trained end-to-end by maximizing segmentation accuracy on annotated VOS videos. See [4] for more details.

A.4. Tracking by Student FUSing Teachers (TRASFUSTm)

M. Dunnhofer, N. Martinel, C. Micheloni
{*matteo.dunnhofer, niki.martinel, christian.micheloni*}*@uniud.it*

The tracker TRASFUSTm is the combination of the TRASFUST bounding-box tracker [19] with the target-dependent segmentation generation method AlphaRefine [67]. The TRASFUST tracker consists of two components: (i) a fast processing CNN-based model called the Student; (ii) a pool of off-the-shelf trackers, i.e. the Teachers. The Student, which has to form of a deep regression tracker, is trained offline based on a learning scheme which combines knowledge distillation (KD) and reinforcement learning (RL). Relevant tracking knowledge is acquired through KD from multiple teacher trackers After learning the Student is capable of evaluating and selecting the target localization predicted by the best teacher in the pool at every frame of a video. In this submission, the SuperDiMP [2] and Stark [66] trackers compose the pool of Teachers.

A.5. Learning Spatio-Appearance Memory Network for High-Performance Visual Tracking (SAMN)

F. Xie, G. Wang, C. Wu
220191672@seu.edu.cn, flylight@mail.ustc.edu.cn, czw390@psu.edu

The tracker SAMN presents a novel segmentation-based tracking architecture, which is equipped with a spatio-appearance memory network to learn accurate spatio-temporal correspondence. Among it, an appearance memory network explores spatio-temporal non-local similarity to learn the dense correspondence between the segmentation mask and the current frame. Meanwhile, a spatial memory network is modelled as discriminative correlation filter to learn the mapping between feature map and spatial map. The appearance memory network helps to filter out the noisy samples in the spatial memory network while the latter provides the former with more accurate target geometrical centre.

A.6. SAMN_DiMP (SAMN_DiMP)

F. Xie, G. Wang, C. Wu
jaffe03@seu.edu.cn, flylight@mail.ustc.edu.cn, czw390@psu.edu

SAMN_DiMP uses two ResNet50 networks as backbone to extract features for the filter from DiMP and appearance memory network from SAMN. For higher speed and more robust performance, the spatial memory network is replaced by an independent model from DiMP. The appearance memory network is trained only using segmentation training datasets with position encoding. Thus, the spatial memory network can be replaced by other trackers such as the DiMP.

A.7. Siamse IOU and Occlusion aware Network for Tracking (SION)

M. Dasari, R. Gorthi
ee18d001@iittp.ac.in, rkg@iittp.ac.in

SION trackers extends SiamRPN++ [39] for identifying the occlusion in a given frame by formulating it as a supervised classification task. It also adds anchor overlap prediction branch for aiding occlusion identification and overall improvement of the tracker.

A.8. Visual Tracking via Adversarial Information Learning - VITAL++ (VITAL++)

A. Rajiv
ee17b032@iittp.ac.in

The tracker highlights the two-stage support that the Q-Network of the InfoGAN could provide in information extraction and feedback to improve the tracking framework. For the first time in visual object tracking, information-theoretic regularization is employed to couple the target and background classes with unique distributions. This coupling helps in the identification of the tracker loss through the Q-Network. The InfoGAN framework provides an efficient unsupervised feedback mechanism during tracker loss and the proposed re-sampling strategies enhance the performance of the Multi-Domain Network (MDNet) class of trackers. The unique features proposed in this tracking framework are (i) to improve the target selection strategy, (ii) unsupervised identification of tracker failure, and (iii) re-sampling strategies at identified tracker failures by leveraging the Q-Network parameter estimates.

A.9. Ensemble correlation filter tracking based on a temporal confidence learning (TCLCF)

C. Tsai
chiyi_tsai@gms.tku.edu.tw

TCLCF is a real-time ensemble correlation filter tracker based on a temporal confidence learning method. In the current implementation, we use four different correlation filters with HOG and Colornames features to collaboratively track the same target. The TCLCF tracker is a fast and robust generic object tracker without GPU acceleration. The tracker is implemented in C++ and is suitable for running on embedded platforms with limited computing resources.

A.10. RPT_AR: Accurate Point Set Representation Tracking by Alpha-Refine (RPT_AR)

C. Zhang, K. Zhang, Y. Wang, L. Liu, S. Ge
andyzhangchunhui@gmail.com, zhangkangkai@iie.ac.cn, wangyong5@mail.sysu.edu.cn, liuli@cuhk.edu.cn, geshiming@iie.ac.cn

This idea extends to the two-stage UAV tracking framework [70]. The tracker RPT_AR consists of two stages: tracking stage and refinement stage. At the tracking stage, we adopt the point set representation tracking network [47] to estimate the coarse target localization. This stage combines the offline-trained RepPoints [69] network to indicate the semantically and geometrically significant positions of target region for initial bounding box and an online classifier for accurate target localization. At the refinement stage, we conduct the Alpha-Refine module [67] to boost the segmentation mask accuracy of a modified D3S [44].

A.11. SiamBAN-ACM (ACM)

W. Han, X. Dong, F. Khan, L. Shao, J. Shen
wencheng@bit.edu.cn, xingping.dong@gmail.com, fahad.khan@liu.se, ling.shao@ieee.org, shenjianbingcg@gmail.com

We propose a learnable module, called the asymmetric convolution (ACM), which learns to better capture the semantic correlation information in off-line training on large-scale data. Different from DW-XCorr [39] and its predecessor (XCorr) [1], which regard a single feature map as the convolution kernel, our ACM decomposes the convolution operation on a concatenated feature map into two mathematically equivalent operations, thereby avoiding the need for the feature maps to be of the same size (width and height) during concatenation. The tracker ACM can incorporate useful prior information, such as bounding-box size, with standard visual features. Furthermore, ACM can easily be integrated into existing Siamese trackers based on DW-XCorr or XCorr. To demonstrate its generalization ability, we integrate ACM into three representative trackers: SiamFC, SiamRPN++ and SiamBAN.

A.12. SiamUSC:Uncertainty-aware Semantic Consistency Siamese Tracker (SiamUSC)

J. MA, B. Zhong, X. Lan, R. Ji, X. Li
majie@stu.hqu.edu.cn, bnzhong@gxnu.edu.cn, xiangyuanlan@life.hkbu.edu.hk, rrji@xmu.edu.cn, lixx@gxnu.edu.cn

We propose an uncertainty-aware semantic consistency tracker (named SiamUSC) based on an anchor-free Siamese network [9] to improve the unaligned problem between classification and regression. Firstly, SiamUSC use an uncertainty-aware semantic consistency module to characterize the trustworthiness of the features. Based on the enhanced features, the semantic branch adaptively validates the semantic consistency of a target instance estimated by the classification and regression branches. To effectively capture the semantic correlation information, a pyramid-wise cross correlation was designed. Finally, SiamUSC utilizes a segmentation head based on the modified D3S [44] to generate high-quality masks.

A.13. Deep Convolutional Descriptor Aggregation Tracker with Accurate Refine Module (DCDAAR)

Y. Li, X. Ke, Y. Huang, Y. Niu
liyuezhou.cm@gmail.com, kex@fzu.edu.cn, hyymay@foxmail.com, yuzhenniu@gmail.com

The deep convolutional descriptor aggregation (DCDA) tracker aims to mine the target representation capability of the pre-trained VGG-16 model. We propose an EAS and a CAS method to guide the aggregation of accuracy-aware and robustness-aware features. The tracker DCDA is derived from one-shot learning by designing a dedicated regression process to predict discriminative features in a few iterations. By exploiting robustness feature aggregation, the accuracy feature aggregation, and the discriminative regression, the DCDA with template enhancement [29] strategy enhances the target prediction capacity and it achieves a low-cost reuse of the pretrained model. In order to get a more precise representation, we combined the AlphaRefine [67] method to as a two-stage prediction process.

A.14. DeepMix: Online Auto Data Augmentation for Robust Visual Object Tracking (deepmix)

Z. Cheng, Q. Guo, F. Juefei-Xu
ziyicheng233@gmail.com, tsingqguo@ieee.org, juefei.xu@gmail.com

We propose the DeepMix [11] that takes historical samples' embeddings as input and generates augmented embeddings online, enhancing the state-of-the-art online learning methods for visual object tracking. Specifically, we design a novel network denoted as MixNet that is offline trained for performing online data augmentation for object and background regions via dramatically predicted convolution parameters within one-step. We construct our tracker by embedding the MixNet to SiamRPN++. During testing, we feed the latest ten historical frame features into MixNet and produce a new feature map as the updated template feature of the classification branch.

A.15. Alpha-Refine (AlphaRef)

B. Yan, D. Wang, H. Lu, X. Yang
yan_bin@mail.dlut.edu.cn, {wdice, lhchuan}@dlut.edu.cn, xiaoyun.yang@remarkai.co.uk

The tracker AlphaRef is the winner of the VOT2020-RT challenge. We propose a simple yet powerful two-stage tracker, which consists of a robust base tracker (super-dimp) and an accurate refinement module named Alpha-Refine. In the first stage, super-dimp robustly locates the target, generating an initial bounding box for the target. Then in the second stage, based on this result, Alpha-Refine crops a small search region to predict a high-quality mask for the tracked target. Besides, Alpha-Refine also deploys a delicate mask prediction head [54] to generate high-quality masks. The complete code and trained models of Alpha-Refine have been released at `https://github.com/MasterBin-IIAU/AlphaRefine`.

A.16. ReptileFPN (ReptileFPN)

C. Tsai, Y. Chiu, S. Jhang
{chiyi_tsai, kevin8401128, max_zhang5566}@gms.tku.edu.tw

ReptileFPN is a tracker based on FPN model and a meta-learning technique called Reptile. Inspired by Reptile Meta-Tracker, we trained a deep learning network offline by repeatedly sampling different tasks. The resulting network can quickly adapt to any domain without the need to train multi-domain branches like MDNet. The original architecture from Reptile Meta-Tracker used VGG like backbone, here we modified it using FPN to further improve the feature extraction ability. During online initialization, the ReptileFPN tracker only requires a few training examples from the first frame and a few steps of optimization to perform well in online tracking. See [28] for more details.

A.17. STARK for the Real-Time Challenge (transformer) (STARK_RT)

B. Yan, X. Zhang, H. Peng, D. Wang, H. Lu, X. Yang
{yan_bin, zxiaohan}@mail.dlut.edu.cn, Houwen.Peng@microsoft.com, {wdice, lhchuan}@dlut.edu.cn, xiaoyun.yang@remark.co.uk

The tracker STARK_RT consists of two stages. First, we use the Spatio-temporal transformer-based STARK to locate the target. We train the network for 2x longer training time. Then, we use the Alpha-Refine to predict one high-quality mask as the final result. The architecture of the Alpha-Refine is also the same as that in the VOT2020 Real-Time challenge, but we train it with larger datasets. We use the original AlphaRefine to generate pseudo mask labels for the LaSOT and GOT-10K. Then, we add these two large-scale datasets to our mask training set to train a more powerful AlphaRefine model. Finally, we use TensorRT to speed up our method for real-time speed.

A.18. Transformer Tracking (TransT)

X. Chen, J. Zhu, B. Yan, D. Wang, H. Lu, X. Yang
{chenxin3131, jiawen, yan_bin}@mail.dlut.edu.cn, {wdice, lhchuan}@dlut.edn.cn, xyang@remarkholdings.com

Transformer Tracking [8] presents a transformer-based feature fusion network, which effectively combines the template and the search region features using attention mechanism. TransT consists of three components: the siamese-like feature extraction backbone (ResNet50 [24]), the designed feature fusion network, and the prediction head. We extend our transformer tracking framework with a segmentation branch [5] to generate an accurate mask. The segmentation branch fuses the output features of the feature fusion network with the low-level features of the backbone in the FPN style. For more details about TransT, the reader is referred to [8].

A.19. fRPT: fast Representative Points based Tracker (fRPT)

L. Wang, H. Zhang, Z. Ma, Y. Jun
wanglinyuan@zju.edu.cn, 1067166127@qq.com,
kobebean@zju.edu.cn, yin_jun@dahuatech.com

We accelerate our representative points based tracker [47] with a lightweight online learning strategy. Specifically, we build a compact yet diverse representation of the training set that effectively reduces the number of samples in the learning [15]. We further reduce the channel dimension of the multi-level features from 256 to 64. The output of fRPT is fed into a modified D3S [44] to obtain the segmentation mask. The backbone of fRPT is ResNet50 pre-trained on ImageNet. The online classifier is trained with components obtained from the generative sample space, while the target estimation head is trained using pairs of frames from YouTube-Bounding Box [53], COCO [40] and ImageNet VID [55] datasets.

A.20. SiamUSCPlus: Online Uncertainty-aware Semantic Consistency Siamese Tracker (SiamUSCP)

J. Ma, B. Zhong, X. Lan, R. Ji, X. Li
majie@stu.hqu.edu.cn, bnzhong@gxnu.edu.cn,
xiangyuanlan@life.hkbu.edu.hk, rrji@xmu.edu.cn,
lixx@gxnu.edu.cn

This model is the extension of the SiamUSC tracker (A.12). Inspired by recent online models [72], we introduce an online branch to capture target object's appearance changes during tracking. We use the trustworthiness semantic consistency score to evaluate the most reliable tracking results.

A.21. Siamese Tracker with Template Enhancement and Mask Generation (SiamEM_R)

Y. Li, Y. Ye, X. Ke, Y. Niu, Y. Huang
liyuezhou.cm@gmail.com, yyfzu@foxmail.com,
kex@fzu.edu.cn, yuzhenniu@gmail.com,
hyymay@foxmail.com

Based on our SiamEM [29] method, we obtain the template enhancement method for SiamFC++ [65] with AlexNet. We replace the mask generation module with [67] to build SiamEM_R. Given that the essence of Siamese trackers is instance learning, the template enhancement construct an alternative template to address the under-fitting of the instance space. The method based on feature descriptor aggregation in baseline can predict the mask at low cost, but the prediction accuracy is also limited by the model capacity. The AlphaRefine method [68] has been fully retrained to predict a more accurate mask than baseline.

A.22. Multi-Template Transformer Tracking (TransT_M)

X. Chen, J. Zhu, B. Yan, D. Wang, H. Lu, X. Yang
{chenxin3131, jiawen, yan_bin}@mail.dlut.edu.cn,
{wdice, lhchuan}@dlut.edn.cn,
xyang@remarkholdings.com

TransT_M is a variant of TransT [8]. We add a segmentation branch, a Multi-Template design, and an IoU prediction head on TransT, forming an end-to-end framework. We concatenate two templates in the spatial dimension, and input them into the template branch of TransT. IoU prediction head is a three-layer perceptron to predict the bounding box's IoU and control the updating of the template. For details about TransT and the segmentation head, please refer to A.18.

A.23. NullSpaceRDAR (NSpaceRDAR)

M. Abdelpakey, M. Shehata
{mohamed.abdelpakey, mohamed.sami.shehata}@ubc.ca

NullSpaceRDAR is built upon DiMP tracker [2] and uses ResNet50 [24] as a backbone. However, NullSpaceRDAR learns a feature representation by projecting the traditional backbone feature space onto a novel discriminative null space that is used to regularize the backbone loss function. We refer to the discriminative null space herein as joint null space. The same target features (i.e., target-specific) in the proposed joint-null space are collapsed into a single point, and different target-specific features are collapsed into different points. Consequently, the joint-null space forces the network to be sensitive to the object's variations from the same class (i.e., intra-class variations). Moreover, an adaptive loss function is utilized for bounding box estimation to select the most suitable loss function from a super-set family of loss functions based on the training data.

A.24. Fast TREG++: fast target transformed regression and segmentation for accurate tracking (F_TregPlus)

Y. Cui, C. Jiang, L. Wang, G. Wu
{cuiyutao, mg1933027}@smail.nju.edu.cn,
{lmwang, gswu}@nju.edu.cn

This tracker is based on tracker TregPlus (A.25). The difference lies on the classifier component: while TregPlus uses three-scale classifiers, F_TregPlus uses only one scale which is similar with DiMP.

A.25. TREG++: target transformed regression and segmentation for accurate tracking (TregPlus)

Y. Cui, C. Jiang, L. Wang, G. Wu
{cuiyutao, mg1933027}@smail.nju.edu.cn,
{lmwang, gswu}@nju.edu.cn

This work is an extension of TREG [13]. We propose a Transformer-based regression and segmentation branch for accurate tracking. Our tracker is composed of two components, an online classifier proposed in DiMP [2] and a novel target-aware transformer for generating accurate bounding box and mask. The core to the tracker TregPlus is to model pair-wise relation between elements in target template and search region, and to use the resulted target enhanced visual representation for accurate bounding box regression and mask. In the first stage, the multi-scale classifiers locate the target center. We use three classifiers based on resnet-50 features of layer-1, layer-2 and layer3, respectively. In the second stage, we perform an accurate regression and segmentation based on the target center and target scale estimated in the previous frame.

A.26. Dual-branch Transformer Tracker with No Convolutional Neural Networks (DualTFR)

F. Xie, G. Wang, C. Wu, W. Yang
220191672@seu.edu.cn, flylight@mail.ustc.edu.cn,
czw390@psu.edu, wkyang@seu.edu.cn

We use pure transformer components to formulate our tracker without convolutional networks. No CNN backbone is used to extract feature embeddings like other transformer tracker. We use the half stage of the block 3 from Swin transformer [42] as the feature extractor. The feature extractor is pretrained in ImageNet1K [38]. The template feature and search feature are further processed by CrossVit-like network [6] for feature fusion. The CrossVit-like network is not pretrained. We only use tokens generated from each image patch. Further, a multi-layer perception layer is used for regression and classification like TransT style and we use AlphaRefine [68] to generate the object's mask.

A.27. Dual-branch Transformer Tracker with No Convolutional Neural Networks (DualTFRst)

F. Xie
220191672@seu.edu.cn

The tracker DualTFRst is based on tracker DualTFR (A.26). We further add context information of searching area to the template transformer branch. In every frame, the feature of search area will go through global average pooling layer and be concatenated to the template features. The model is re-trained with dynamic context information settings.

A.28. CFRPT:Customized Feature Representation for Siamese Visual Tracking (CFRPT)

H. Zhang, L. Wang, Z. Ma, Y. Jun
1067166127@qq.com, {wanglinyuan,
kobebean}@zju.edu.cn, yin_jun@dahuatech.com

We extend our representative points based tracker [47] with customized feature representation. We propose a customized feature extractor to capture accurate and task-aware visual patterns. Extreme-enhanced features are extracted and finally combined with the original point feature to precisely estimate the target state. Inspired from [41], we replace the standard convolutional layers for target state estimation with the CoordConv layers. Extra channels filled with coordinate information are concatenated to the input representation, which allows the convolutional filters to know where they are in Cartesian space. Finally, AlphaRefine [67] is employed to produce a mask prediction as the output.

A.29. Visual tracking via Fast Saliency-guided Continuous Correlation Filters (FSC2F)

A. Memarmoghadam
a.memarmoghadam@yahoo.com

The tracker FSC2F is based on the ECOhc approach [15]. A fast spatio temporal saliency map is added using the PQFT approach [20]. The PQFT model utilizes intensity, colour, and motion features for quaternion representation of the search image context around the previously pose of the tracked object. Therefore, attentional regions in the coarse saliency map can constrain target confidence peaks. Moreover, a faster scale estimation algorithm is utilised by enhancing the fast fDSST method [18] via jointly learning of the sparsely-sampled scale spaces.

A.30. Discriminative Sing-Shot Segmentation Tracker (D3S)

A. Lukezic, J. Matas, M. Kristan
{alan.lukezic, matej.kristan}@fri.uni-lj.si,
matas@cmp.felk.cvut.cz

The tracker represents the target using two visual models: (i) geometrically constrained Euclidean model (GEM) used for discriminative target localization and (ii) geometrically invariant models (GIM) to address significant target deformations. The results of both models are combined into the high-resolution segmentation output using refinement pathway.

A.31. Discriminative Sing-Shot Segmentation Tracker v2 (D3Sv2)

A. Lukezic, J. Matas, M. Kristan
{alan.lukezic, matej.kristan}@fri.uni-lj.si,
matas@cmp.felk.cvut.cz

The tracker is an extended version of the D3S tracker1 [44], published at CVPR 2020. The original method is extended in the following aspects: (i) a better backbone, (ii) channel attention mechanism in the upscaling modules in GIM, (iii) trainable MLP-based similarity computation in GIM, which replaces the 'handcrafted' top-K average operation and (iv) the new scale estimation module used for robust target size estimation.

A.32. Dual-branch Transformer Tracker with No Convolutional Neural Networks-online version (DualTFRon)

F. Xie
220191672@seu.edu.cn

The tracker DualTFRon is based on tracker DualTFR (A.26). We further add an online branch and a template update mechanism. The online branch is from ATOM [16]. We use two templates in our tracker, one is from the first frame and another is the dynamic template which will be updated in every fixed number of frame.

A.33. Refined Transformer Tracker (RTT)

T. Xu, X. Zhu, S. Zhao, Z. Tang, H. Li, X. Wu, Z. Feng, M. Rana, J. Kittler
{tianyang_xu, xuefeng_zhu95}@163.com, 7201905026@stu.jiangnan.edu.cn, {zhangyong_tang_jnu, hui_li_jnu}@163.com, wu_xiaojun@jiangnan.edu.cn, {z.feng, m.a.rana, j.kittler}@surrey.ac.uk

Refined Transformer Tracker employs the spatio-temporal transformer structure with a coarse-to-fine strategy for accurate target localisation. ResNet-101 network is used as backbone features to extract representations from the template and instance images. Transformer encoder-decoder is utilised to perform similarity comparison to predict the final target bounding box. A coarse-to-fine strategy is designed to perform two-stage tracking with a precise search region to suppress the background. An additional Alpha-Refine module is used to predict the final mask.

A.34. TRATMask: Tracking by Attention Using Spatio-Temporal Features (TRATMask)

H. Saribas, H. Cevikalp, B. Uzun
{hasansaribas48, hakan.cevikalp, eee.bedirhan}@gmail.com

The tracker TRATMask uses a two-stream network which consists of a 2D-CNN and a 3D-CNN, to use both spatial and temporal information in video streams. To obtain temporal (motion) information, 3D-CNN is fed by stacking the previous 4 frames with one stride. To extract spatial information, the 2D-CNN is used. Then, we fuse the two-stream network outputs by using an attention module. Finally, we propose a new segmentation module to extract segmentation mask of target object.

A.35. KeepTrack (keep_track)

C. Mayer, M. Danelljan, D. Paudel, L. Van Gool
{chmayer, martin.danelljan, paudel, vangool}@vision.ee.ethz.ch

We propose KeepTrack [48] a novel tacking method that keeps track of distractor objects in order to continue tracking the target. To this end, we introduce a learned association network, allowing us to propagate the identities of all target candidates from frame to frame. To tackle the problem of lacking ground-truth correspondences between distractor objects in visual tracking, we propose a training strategy that combines partial annotations with self-supervision. We employ super DiMP as our base tracker in order to extract target candidates and propose a target candidate association network that we use to identify the target and distractors across frames. In addition to our method described in [48], we use AlphaRefine [67] to produce segmentation masks from the predicted bounding boxes of KeepTrack.

A.36. RPTMask (RPTMask)

Z. Fu, L. Wang, Q. Deng, DK. Du, M. Zheng, Q. Liu, Y. Wang
fuzhihong@buaa.edu.cn, {wangliangliang.makalo, dengqili, dukang.daniel, zhengmin.666}@bytedance.com, {qingjie.liu, yhwang}@buaa.edu.cn

We propose a two-stage tracker, called RPTMask. The first stage is a base tracker responsible for locating the target bounding boxes. Specifically, we use ATOM [16] to coarsely locate the target and update the tracking model and use RPT [47] to generate the target bounding boxes. In the second stage, following STMVOS [51], we design a mask generation network to generate the target masks. First, only the first frame is set to be the memory frame. Second, we improve the space-time memory reader in STMVOS with the kernel trick [56] and the top-k filtering [10] strategy. Third, following AlphaRefine [67], we add a refined box regression head paralleled to the mask decoder. The backbone in all models is ResNet50.

A.37. Local-Global Tracking tracker (LGT)

Submitted by VOT Committee

The core element of LGT is a coupled-layer visual model that combines the target global and local appearance by interlacing two layers. By this coupled constraint paradigm between the adaptation of the global and the local layer, a more robust tracking through significant appearance changes is achieved. The reader is referred to [59] for details.

A.38. ANT (ANT)

Submitted by VOT Committee

The ANT tracker is a conceptual increment to the idea of multi-layer appearance representation that is first described in [59]. The tracker addresses the problem of self-supervised estimation of a large number of parameters by introducing controlled graduation in estimation of the free parameters. The appearance of the object is decomposed into several sub-models, each describing the target at a different level of detail. The sub models interact during target localization and, depending on the visual uncertainty, serve for cross-sub-model supervised updating. The reader is referred to [60] for details.

A.39. ATOM tracker with Alpha refine post-processing step (AR_ATOM)

Submitted by VOT Committee

This tracker employs the standard ATOM [16] (A.45) for predicting bounding boxes. The AlphaRefine [67] network is then employed to predict the final mask as a post-processing step.

A.40. DiMP50 tracker with Alpha refine post-processing step (AR_DiMP-50)

Submitted by VOT Committee

This tracker employs the standard DiMP50 [2] (A.2) for predicting bounding boxes. The AlphaRefine [67] network is then employed to predict the final mask as a post-processing step.

A.41. Know your surrondings tracker with Alpha refine post-processing step (AR_KYS)

Submitted by VOT Committee

This tracker employs the standard KYS [3] (A.48) for predicting bounding boxes. The AlphaRefine [67] network is then employed to predict the final mask as a post-processing step.

A.42. PrDiMP50 tracker with Alpha refine post-processing step (AR_PrDiMP-50)

Submitted by VOT Committee

This tracker employs the standard PrDiMP50 [17] (A.51) for predicting bounding boxes. The AlphaRefine [67] network is then employed to predict the final mask as a post-processing step.

A.43. SuperDiMP50 tracker with Alpha refine post-processing step (AR_SuperDiMP-50)

Submitted by VOT Committee

This tracker employs the standard SuperDiMP50 [2, 17, 21, 22] for predicting bounding boxes. The AlphaRefine [67] network is then employed to predict the final mask as a post-processing step.

A.44. Scale adaptive mean shift (ASMS)

Submitted by VOT Committee

The mean-shift tracker optimizes the Hellinger distance between template histogram and target candidate in the image. This optimization is done by a gradient descend. ASMS [62] addresses the problem of scale adaptation and presents a novel theoretically justified scale estimation mechanism which relies solely on the mean-shift procedure for the Hellinger distance. ASMS also introduces two improvements of the mean-shift tracker that make the scale estimation more robust in the presence of background clutter – a novel histogram colour weighting and a forward-backward consistency check. Code available at `https://github.com/vojirt/asms`.

A.45. Accurate Tracking by Overlap Maximization (ATOM)

Submitted by VOT Committee

ATOM [16] separates the tracking problem into two subtasks: i) target classification, where the aim is to robustly distinguish the target from the background; and ii) target estimation, where an accurate bounding box for the target is determined. Target classification is performed by training a discriminative classifier online. Target estimation is performed by an overlap maximization approach where a network module is trained offline to predict the overlap between the target object and a bounding box estimate, conditioned on the target appearance in first frame. See [16] for more details.

A.46. Discriminative Correlation Filter with Channel and Spatial Reliability (CSRDCF)

Submitted by VOT Committee

The CSR-DCF [46] improves discriminative correlation filter trackers by introducing the two concepts: spatial reliability and channel reliability. It uses color segmentation as spatial reliability to adjust the filter support to the part of the object suitable for tracking. The channel reliability reflects the discriminative power of each filter channel. The tracker uses only HoG and colornames features. This is the C++ openCv implementation.

A.47. Kernelized Correlation Filter (KCF)

Submitted by VOT Committee

This tracker is a C++ implementation of Kernelized Correlation Filter [25] operating on simple HOG features and Colour Names. The KCF tracker is equivalent to a Kernel Ridge Regression trained with thousands of sample patches around the object at different translations. It implements multi-thread multi-scale support, sub-cell peak estimation and replacing the model update by linear interpolation with a more robust update scheme. Code available at
`https://github.com/vojirt/kcf`.

A.48. Know your surrondings tracker (KYS)

Submitted by VOT Committee

The KYS tracker [3] presents a novel tracking architecture which can utilize scene information for tracking. Scene information consists of knowledge about the presence and locations of other objects in the surrounding scene, which can be highly beneficial in challenging cases where distractors are present. The KYS tracker represents such information as dense localized state vectors, which can encode, for example, if the local region is target, background, or distractor. These state vectors are propagated through the sequence and combined with the appearance model output to localize the target. Our network is learned to effectively utilize the scene information by directly maximizing tracking performance on video segments.

A.49. (L1APG)

Submitted by VOT Committee

L1APG considers tracking as a sparse approximation problem in a particle filter framework. To find the target in anew frame, each target candidate is sparsely represented in the space spanned by target templates and trivial templates.The candidate with the smallest projection error after solving anl1regularized least squares problem. The Bayesian state inference framework is used to propagate sample distributions over time.

A.50. Learning what to learn tracker with Box2Seg head (LWL_B2S)

Submitted by VOT Committee

This is the standard Learning What to Learn (LWL) [4] (A.3) video object segmentation and tracking approach, trained with the annotations generated by the approach [71]. That is, in addition to the YouTubeVOS and DAVIS training datasets, [71] is used to generate masks from bounding box annotated sequences in LaSOT and GOT10k. We then finetune LWL on the combined data. The same inference settings is used as in the standard LWL [4]. See [71] for details.

A.51. (PrDiMP-50)

Submitted by VOT Committee

PrDiMP [17] provides an energy-based probabilistic regression formulation for the classification and regression branch of the DiMP [2] tracker. The energy-based regression formulation is based on [21], and extends it by modeling noise in the labels.

A.52. SiameseFC-AlexNet (SiamFc)

Submitted by VOT Committee

SiamFC [1] applies a fully-convolutional Siamese network [12] trained to locate an exemplar image within a larger search image. The architecture is fully convolutional with respect to the search image: dense and efficient sliding-window evaluation is achieved with a bilinear layer that computes the cross-correlation of two inputs. The deep convnet is first trained offline on the large ILSVRC15 [55] video dataset to address a general similarity learning problem, and then this function is evaluated during testing by a simplistic tracker. SiamFc incorporates elementary temporal constraints: the object search is done within a region of approximately four times its previous size, and a cosine window is added to the score map to penalize large displacements. SiamFc also processes several scaled versions of the search image, any change in scale is penalised and damping is applied to the scale factor.

A.53. VOS SOTA method (STM)

Submitted by VOT Committee

STM [51] is a VOS method employing a space-time memory module combined with a dot-product attention layer. Please see the original paper for details [51].

B. VOT-LT2021 submissions

This appendix provides a short summary of trackers considered in the VOT-LT2021 challenge.

B.1. Fusing Complementary Trackers for Long-term Visual Tracking (mlpLT)

M. Dunnhofer, K. Simonato, C. Micheloni
$\{$*matteo.dunnhofer, christian.micheloni*$\}$*@uniud.it,*
simonato.kristian@spes.uniud.it

The idea behind the mlpLT tracker is to fuse the capabilities of different trackers. In particular, mlpLT implements a strategy that fuses the Stark [66] and SuperDiMP [2] trackers, which have been selected due to their complementary features. Indeed, the tracker Stark has been selected because of its ability to provide spatially accurate bounding-boxes, and to re-detect the target after disappearances. The meta-updater [14] controlled SuperDiMP was chosen due its robustness. The combination of such trackers is managed by a decision strategy based on an online learned target verifier [50]. At every frame, the trackers are run in parallel to predict their target localizations. Such outputs are checked by the verifier which quantifies how good the trackers are following the target. Based on such evaluations, the decision strategy selects which localization to give as output for the current frame. Such an outcome is also employed to correct the tracker that achieved the lowest performance according to the verifier. Additionally strategies such as the computation of adaptive search areas, and the avoidance of wrong target size estimations, have been implemented to the baseline trackers in order to make their localizations more consistent.

B.2. Siamse IOU and Occlusion aware Network for Tracking (SION_LT)

M. Dasari, R. Gorthi
ee18d001@iittp.ac.in, rkg@iittp.ac.in

SION tracker is extension of SiamRPN++ with added occlusion classification and anchor overlap prediction.

B.3. Siam R-CNN (SiamRCNN)

P. Voigtlaender, J. Luiten, P. Torr, B. Leibe
{voigtlaender, luiten}@vision.rwth-aachen.de, phst@robots.ox.ac.uk, leibe@vision.rwth-aachen.de

Siam R-CNN [61] is a Siamese re-detection architecture which unleashes the full power of two-stage object detection approaches for visual object tracking. Siam R-CNN is based on Faster R-CNN with a ResNet-101 backbone. Siam R-CNN uses a tracklet-based dynamic programming algorithm, which takes advantage of re-detections of both the first-frame template and previous-frame predictions, to model the full history of both the object to be tracked and potential distractor objects. This enables Siam R-CNN to make better tracking decisions, as well as to re-detect tracked objects after long occlusion. Finally, Siam R-CNN uses a novel hard example mining strategy to improve its robustness to similar looking objects.

B.4. SLOT (SLOT)

W. Xue, Z. Zhang, K. Zhang, B. Liu, C. Zhang, J. Liu, Z. Feng, S. Chen
xuewanli@email.tjut.edu.cn, tjut-zzb@hotmail.com, {zhkhua, kfliubo}@gmail.com, chenvy@dlmu.edu.cn, jingenliu@gmail.com, zyfeng@tju.edu.cn, sy@ieee.org

We propose a self-corrective network framework (termed as SLOT) including a self-modulated tracking reliability evaluation (STRE) and a self-adjusting proposal post-processor (SPPP) for long-term visual object tracking (LVOT). SLOT tracker adopts a tracking quality evaluator to reduce the cumulative error. Our key insight is that a long-term tracker should have the ability to recapture the target when encountering serious challenges (e.g., full occlusion and out of view). To achieve this objective, first, we build an effective tracking reliability classification on a modulation sub-network, whose training data is obtained from the unlabeled video by the adaptive self-labeling method. In particular, our self-labeling method can automatically label accurate and comprehensive samples according to the statistical characteristics of IoU and center distance without any fixed thresholds. Meanwhile, we propose a self-adjusting proposal post-processor module including a dynamic NMS, which is activated by STRE, to recapture the target in time and accurately. As the SLOT manuscript is under review, once the paper is accepted, we will promptly provide relevant information on `https://github.com/TJUT-CV/SLOT`.

B.5. DiMP Strengthened STARK for Longterm Tracking (STARK_RGBD_LT)

C. Liu, B. Yan, X. Zhang, L. Wang, H. Peng, D. Wang, H. Lu, X. Yang
{njx2019, yan_bin, zhangxy71102}@mail.dlut.edu.cn, ljwang@dlut.edu.cn, Houwen.Peng@microsoft.com, {wdice, lhchuan}@dlut.edu.cn, xiaoyun.yang@remarkai.co.uk

We take the powerful transformer-based STARK as our base method. We change the backbone of STARK from ResNet50 to DeiT, which boosts the feature extraction ability. Then we apply a refinement module similar to AlphaRefine to it. The refinement module is modified from a STARK tracker, with a smaller search region than its original design. Besides, we combine DiMPsuper with it for better dealing with the change of appearance. Specifically, if the tracking score of STARK is low (e.g. lower than 0.2), the DiMPsuper will take over. Finally, we evaluate the track of the object to judge whether the object is out of scope. If an object heads into the boundary of the scope, and the tracking score reduce heavily, we think that this object has moved out of the scope. We also apply the reward to the tracking score if the tracking boxes of STARK and DiMPsuper are identical enough.

B.6. Target-driven Inference for Deep Video Object Tracking (TDIOT)

F. Gurkan, O. Cirakman, B. Gunsel, L. Cerkezi
{gurkanf, cirakmano, gunselb}@itu.edu.tr, llukmancerkezi@gmail.com

We introduce TDIOT, a novel inference architecture placed on top of FPN-ResNet101 backbone to jointly perform detection and tracking, without requiring additional training for tracking purpose. TDIOT employs the pretrained Mask R-CNN model [23] and adopts it to a tracker at the inference stage. In particular, with the guidance of a target driven proposal sampler, TDIOT enables focus on the target object by filtering proposals generated by Mask R-CNN Region Proposal Network (RPN). On the inference head TDIOT applies an appearance similarity-based temporal matching for data association. In order to tackle tracking discontinuities, a local search and matching module is incorporated into the inference head layer based on Kernelized Correlation Filter (KCF) [25]. Also a low cost verification layer is incorporated into the inference architecture to monitor presence of the target based on LBP histogram model [52]. The code will be released at `https://github.com/msprITU/TDIOT`

B.7. STARK for the Long-Term Challenge (STARK_LT)

C. Liu, B. Yan, H. Peng, D. Wang, H. Lu, X. Yang
{njx2019, yan_bin}@mail.dlut.edu.cn,

Houwen.Peng@microsoft.com, {wdice, lhchuan}@dlut.edu.cn, xiaoyun.yang@remark.co.uk

The tracker STARK_LT is composed of a complementary local-global search framework to track on long-term sequences. STARK_LT is based on STARK tracker based on transformer and a modified global search algorithm based on GlobalTrack with ATSS detection model. The global module is trained offline based on siamese network transferred from detection model to find all the possible candidates of targets. The resnet50 backbone is utilized for feature extraction and the inputs are both the whole image. STARK is responsible for target tracking in local region, while the confidence score is lower than the threshold, global search model is performed to re-detect the target in the whole image. As the meanwhile, the Kalman Filter and data association are utilized to suppress the potential distractors as false target candidates. Then STARK is required to verify and find the best target candidate again in local region.

B.8. Switch-Refine Tracking Framework (RincTrack)

X. Xu, F. Shen
xux@smail.nju.edu.cn, frshen@nju.edu.cn

RincTrack is a long-term tracking framework designed with the ability to switch between local and global trackers and produce refined bounding boxes as well as segmentation masks as results. When having enough confidence, TrDiMP is used as the local tracker. Global tracker is called when the results from the local tracker are not reliable and will go back to the local tracker if the results from the global tracker are quite confident for continuous two frames. We utilized a simplified SiamR-CNN as the global tracker. The local tracker is online updated by the results of not only itself, but also the results from the global tracker. AlphaRefine module is cascaded to further refining the bounding box after both local and global trackers. The refine module can also produce segmentation masks if minor changes are made to the codes.

B.9. KeepTrack (keep_track_lt)

C. Mayer, M. Danelljan, D. Paudel, L. Van Gool
{chmayer, martin.danelljan, paudel, vangool}@vision.ee.ethz.ch

We propose KeepTrack a novel tacking method that keeps track of distractor objects in order to continue tracking the target. To this end, we introduce a learned association network, allowing us to propagate the identities of all target candidates from frame to frame. To tackle the problem of lacking ground-truth correspondences between distractor objects in visual tracking, we propose a training strategy that combines partial annotations with self-supervision. In particular, we employ super DiMP as our base tracker in order to extract target candidates and propose a target candidate association network that we use to identify the target and distractors across frames. Additionally, we use AlphaRefine to produce segmentation masks from the predicted bounding boxes of KeepTrack. More details can be found here `https://arxiv.org/pdf/2103.16556.pdf`.

B.10. A More Concise Long-Term Tracker with Meta-updater (SuperD_MU)

K. Dai, D. Wang, J. Li, H. Lu
dkn2014@mail.dlut.edu.cn, {wdice, jianhual, lhchuan}@dlut.edu.cn

The Super_DiMP [2] structure parameters are identical to the official version, and the model uses the official model entirely. The only modification is that the update mechanism is entirely decided by Meta_updater [14]. The training data of Meta-Updater is based on the specified tracker. For the current tracker, we first run over the LaSOT dataset with the original Super_DiMP, and then record the tracking information(bounding boxes, response map, result images, confidence scores) of each frame. Then we train our Meta-Updater with this information as the training data. Meta-Updater is then embedded in the SuperDimp and the training process is repeated again until Meta-Updater fits the tracker completely, refer to [14] for more details.

B.11. (SuperDiMP)

Submitted by VOT Committee

Please see the original paper for details [2].

B.12. (LT_DSE)

Submitted by VOT Committee

This algorithm divides each long-term sequence into several short episodes and tracks the target in each episode using short-term tracking techniques. Whether the target is visible or not is judged by the outputs from the short-term local tracker and the classification-based verifier updated online. If the target disappears, the image-wide re-detection will be conducted and output the possible location and size of the target. Based on these, the tracker crops the local search region that may include the target and sends it to the RPN based regression network. Then, the candidate proposals from the regression network will be scored by the online learned verier. If the candidate with the maximum score is above the pre-defined threshold, the tracker will regard it as the target and re-initialize the short-term components. Finally, the tracker conducts short-term tracking until the target disappears again.

B.13. A Baseline Long-Term Tracker with Meta-Updater (LTMU_B)

Submitted by VOT Committee

The tracker LTMU_B is a simplified version of LTMU [14] and LTDSE with comparable performance adding a RPN-based regression network, a sliding-window based re-detection module and a complex mechanism for updating models and target re-localization. The short-term tracker LTMU_B contains two components. One is for target localization and based on DiMP algorithm [2] using ResNet50 as the backbone network. The update of DiMP is controlled by meta-updater which is proposed by LTMU[5]. The second component is the SiamMask network [63] used for refining the bounding box after locating the centre of the target. It also takes the local search region as the input and outputs the tight bounding boxes of candidate proposals. For the verifier, we adopts MDNet network [50] which uses VGGM as the backbone and is pre-trained on ILSVRC VID dataset. The classification score is finally obtained by sending the tracking result's feature to three fully connected layers. GlobalTrack [27] is utilised as the global detector.

C. VOT-RGBD2021 submissions

This appendix provides a short summary of trackers considered in the VOT-RGBD2021 challenge.

C.1. Spatio-Temporal Transformer with Correlation for RGBD Visual Tracking (sttc_rgbd)

Y. Jiang, Z. Feng, T. Xu, X. Song
1161099088@qq.com, {z.feng, tianyang.xu}@surrey.ac.uk, x.song@jiangnan.edu.cn

The tracker STTC is a modified version of STARK [66] to track objects in RGB and Depth images. The box prediction head is enhanced by correlation calculations. We use HTC and FlowNetv2 to get region proposals when the tracking results are unreliable.

C.2. STARK for the RGBD challenge (STARK_RGBD)

X. Zhang, B. Yan, L. Wang, H. Peng, D. Wang, H. Lu, X. Yang
{zhangxy71102, yan_bin}@mail.dlut.edu.cn, ljwang@dlut.edu.cn, Houwen.Peng@microsoft.com, {wdice, lhchuan}@dlut.edu.cn, xiaoyun.yun@remarkai.co.uk

This is a method combining STARK [66] and DiMPsuper [2]. STARK is a powerful transformer-based tracker with a siamese structure. We first change the backbone of STARK into DeiT [57]. The transformer-based backbone DeiT strengthens the feature of STARK. We notice the STARK method is not good at handling the appearance change of the target. To this end, we combine the DeiT-strengthened STARK with the DiMPsuper tracker, a powerful online updating tracker. Specifically, when the STARK tracker's confidence is low or the prediction of STARK suddenly strays away, the DiMPsuper takes over the tracking process, providing a steady, appearance adaptive result. When the STARK's confidence resumes, we switch back to STARK. We also design a refinement module similar to AlphaRefine [67] by modifying the search region of STARK. The refinement module is applied to the final output of the whole tracking system for further boosting the quality of box estimation.

C.3. Spot-Light Masking feature enhanced DiMP for RGBD Tracking (SLMD)

J. Lee, B. Kim
{wnsgk986, bhkim81}@kitech.re.kr

The proposed tracker is based on the probabilistic DiMP (prDiMP) tracker [17], and uses the Spot-Light Masking feature enhanced method for the search area of the target tracking. SLMD is improving the input feature for the target probabilistic distribution network of prDiMP, and the adaptive gamma correction method [49] is applied to reinforce the feature of the input. The improved feature is applied to the spotlight masking process and is fused with the original input to be used as the input for the localization network. This method can be used to improve performances for not only prDiMP but also various types of trackers which are used to infer the center of the target based on the visual object trackers.

C.4. Towards Accurate RGB-D Tracking by Local and Global Detection (TALGD)

X. Zhu, Z. Tang, T. Xu, H. Li, S. Zhao, X. Wu, J. Kittler
{xuefeng_zhu95, zhangyong_tang_jnu}@163.com, tianyang.xu@surrey.ac.uk, hui_li_jnu@163.com, 7201905026@stu.jiangnan.edu.cn, wu_xiaojun@jiangnan.edu.cn, j.kittler@surrey.ac.uk

The TALGD method is based on the SuperDiMP [2] method and the HTC method [7]. The SuperDiMP is employed to detect local search region and global image to localise the tracked target. Then the target state predicted by local and global detection is refined by the HTC method through detecting background distractors. Besides, the depth image is adopted for occlusion or disappearance reasoning and target retrieval.

C.5. Object tracking based on deep information fusion and consistency constraint update (DRefine)

S. Qiu, Y. Gu, X. Zhang
skyshoumeng@163.com, {gyz, xlzhang}@mail.sim.ac.cn

We first use SuperDiMP [2] to make a preliminary estimation of the target's state and then send it to the AlphaRefine network [67] to fine-tune the tracking results. We fuse the depth information and make consistency judgements

[5] https://github.com/Daikenan/LTMU

based on the results obtained under different inputs (RGB and RGBD). We update the model when the results are consistent.

References

[1] Luca Bertinetto, Jack Valmadre, João Henriques, Philip H. S. Torr, and Andrea Vedaldi. Fully-convolutional siamese networks for object tracking. In *ECCV Workshops*, pages 850–865, 2016.

[2] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6182–6191, 2019.

[3] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Know your surroundings: Exploiting scene information for object tracking. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXIII*, volume 12368 of *Lecture Notes in Computer Science*, pages 205–221. Springer, 2020.

[4] Goutam Bhat, Felix Jaremo Lawin, Martin Danelljan, Andreas Robinson, Michael Felsberg, Luc Van Gool, and Radu Timofte. Learning what to learn for video object segmentation. In *European Conference on Computer Vision ECCV*, 2020.

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[6] Chun-Fu Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. *arXiv preprint arXiv:2103.14899*, 2021.

[7] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. Hybrid task cascade for instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[8] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *CVPR*, 2021.

[9] Zedu Chen, Bineng Zhong, Guorong Li, Shengping Zhang, and Rongrong Ji. Siamese box adaptive network for visual tracking. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 6667–6676. IEEE, 2020.

[10] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Modular interactive video object segmentation: Interaction-to-mask, propagation and difference-aware fusion. *arXiv preprint arXiv:2103.07941*, 2021.

[11] Ziyi Cheng, Xuhong Ren, Felix Juefei-Xu, Wanli Xue, Qing Guo, Lei Ma, and Jianjun Zhao. Deepmix: Online auto data augmentation for robust visual object tracking. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2021.

[12] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, 2005.

[13] Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu. Target transformed regression for accurate tracking. *arXiv preprint arXiv:2104.00403*, 2021.

[14] Kenan Dai, Yunhua Zhang, Dong Wang, Jianhua Li, Huchuan Lu, and Xiaoyun Yang. High-performance long-term tracking with meta-updater. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6298–6307, 2020.

[15] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ECO: efficient convolution operators for tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6638–6646. IEEE Computer Society, 2017.

[16] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ATOM: Accurate tracking by overlap maximization. In *CVPR*, pages 4660–4669, 2019.

[17] Martin Danelljan, Luc Van Gool, and Radu Timofte. Probabilistic regression for visual tracking. In *CVPR*, 2020.

[18] Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, and Michael Felsberg. Discriminative scale space tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(8):1561–1575, 2016.

[19] Matteo Dunnhofer, Niki Martinel, and Christian Micheloni. Tracking-by-trackers with a distilled and reinforced model. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, November 2020.

[20] Chenlei Guo and Liming Zhang. A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression. *IEEE Transactions on Image Processing*, 19(1):185–198, 2009.

[21] Fredrik K. Gustafsson, Martin Danelljan, Goutam Bhat, and Thomas B. Schön. Energy-based models for deep probabilistic regression. In *European Conference on Computer Vision ECCV*, 2020.

[22] Fredrik K. Gustafsson, Martin Danelljan, Radu Timofte, and Thomas B. Schön. How to train your energy-based model for regression. *CoRR*, abs/2005.01698, 2020.

[23] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask R-CNN. In *Proc. IEEE ICCV*, pages 2980–2988, 2017.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[25] J. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on PAMI*, 37(3):583–596, 2015.

[26] L. Huang, X. Zhao, and K. Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *arXiv:1810.11981*, 2018.

[27] Lianghua Huang, Xin Zhao, and Kaiqi Huang. GlobalTrack: A simple and strong baseline for long-term tracking. In *AAAI*, 2020.

[28] Shang-Jhih Jhang and Chi-Yi Tsai. Reptile meta-tracking. In *16th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 1–5, 2019.

[29] Xiao Ke, Yuezhou Li, Yu Ye, and Wenzhong Guo. Template enhancement and mask generation for siamese tracking. *IEEE Signal Processing Letters*, 2020.

[30] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, J.-K. Kamarainen, L. Čehovin, O. Drbohlav, A. Lukezic, A. Berg, A. Eldesokey, J. Kapyla, G. Fernández, and et al. The seventh visual object tracking vot2019 challenge results. In *ICCV2019 Workshops, Workshop on visual object tracking challenge*, 2019.

[31] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, J.-K. Kamarainen, L. Čehovin, D. Martin, A. Lukezic, O. Drbohlav, L. He, Y. Zhang, S. Yan, J. Yang, G. Fernández, and et al. The eighth visual object tracking vot2020 challenge results. In *ECCV2020 Workshops, Workshop on visual object tracking challenge*, 2020.

[32] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Čehovin, T. Vojíř, G. Bhat, A. Lukežič, A. Eldesokey, G. Fernández, and et al. The visual object tracking vot2018 challenge results. In *ECCV2018 Workshops, Workshop on visual object tracking challenge*, 2018.

[33] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Čehovin, T. Vojíř, G. Häger, A. Lukežič, A. Eldesokey, G. Fernández, and et al. The visual object tracking vot2017 challenge results. In *ICCV2017 Workshops, Workshop on visual object tracking challenge*, 2017.

[34] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Čehovin, T. Vojíř, G. Häger, A. Lukežič, G. Fernández, and et al. The visual object tracking vot2016 challenge results. In *ECCV2016 Workshops, Workshop on visual object tracking challenge*, 2016.

[35] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Čehovin, G. Fernández, T. Vojíř, G. Häger, G. Nebehay, R. Pflugfelder, and et al. The visual object tracking vot2015 challenge results. In *ICCV2015 Workshops, Workshop on visual object tracking challenge*, 2015.

[36] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, F. Porikli, L. Čehovin, Georg Nebehay, G. Fernández, T. Vojíř, and et al. The visual object tracking vot2013 challenge results. In *ICCV2013 Workshops, Workshop on visual object tracking challenge*, pages 98 –111, 2013.

[37] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, L. Čehovin, Georg Nebehay, T. Vojíř, G. Fernández, and et al. The visual object tracking vot2014 challenge results. In *ECCV2014 Workshops, Workshop on visual object tracking challenge*, 2014.

[38] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, pages 1097–1105, 2012.

[39] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4282–4291, 2019.

[40] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[41] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 9628–9639, 2018.

[42] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

[43] A. Lukežič, U. Kart, J. Kämäräinen, J. Matas, and M. Kristan. CDTB: A Color and Depth Visual Object Tracking Dataset and Benchmark. In *ICCV*, 2019.

[44] A. Lukežič, J. Matas, and M. Kristan. D3S - a discriminative single shot segmentation tracker. In *Proceedings of the IEEE/CVF CVPR*, pages 7131–7140. IEEE, 2020.

[45] A. Lukežič, L. Čehovin Zajc, T. Vojíř, J. Matas, and M. Kristan. Sperformance evaluation methodology for long-term single object tracking. *IEEE Transactions on Cybernetics*, 2020.

[46] A. Lukežič, T. Vojíř, L. Čehovin Zajc, J. Matas, and M. Kristan. Discriminative correlation filter with channel and spatial reliability. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6309–6318, July 2017.

[47] Ziang Ma, Linyuan Wang, Haitao Zhang, Wei Lu, and Jun Yin. RPT: learning point set representation for siamese visual tracking. In Adrien Bartoli and Andrea Fusiello, editors, *Computer Vision - ECCV 2020 Workshops - Glasgow, UK, August 23-28, 2020, Proceedings, Part V*, volume 12539 of *Lecture Notes in Computer Science*, pages 653–665. Springer, 2020.

[48] Christoph Mayer, Martin Danelljan, Danda Pani Paudel, and Luc Van Gool. Learning target candidate association to keep track of what not to track. *CoRR*, abs/2103.16556, 2021.

[49] Manca Žerovnik Mekuč, Ciril Bohak, Samo Hudoklin, Byeong Hak Kim, Min Young Kim, Matija Marolt, et al. Automatic segmentation of mitochondria and endolysosomes in volumetric electron microscopy data. *Computers in biology and medicine*, 119:103693, 2020.

[50] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, pages 4293–4302, 2016.

[51] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9226–9235, 2019.

[52] T. Ojala, M. Pietikänen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on PAMI*, 24(7):971 – 987, 2002.

[53] Esteban Real, Jonathon Shlens, Stefano Mazzocchi, Xin Pan, and Vincent Vanhoucke. Youtube-boundingboxes: A large

high-precision human-annotated data set for object detection in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5296–5305, 2017.

[54] Andreas Robinson, Felix Järemo Lawin, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Learning fast and robust target models for video object segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Computer Vision Foundation, June 2020.

[55] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[56] Hongje Seong, Junhyuk Hyun, and Euntai Kim. Kernelized memory network for video object segmentation. In *European Conference on Computer Vision*, pages 629–645. Springer, 2020.

[57] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

[58] Luka Čehovin. TraX: The visual Tracking eXchange Protocol and Library. *Neurocomputing*, 2017.

[59] L. Čehovin, M. Kristan, and A. Leonardis. Robust visual tracking using an adaptive coupled-layer visual model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(4):941–953, 2013.

[60] Luka Čehovin, Aleš Leonardis, and Matej Kristan. Robust visual tracking using template anchors. In *WACV*. IEEE, Mar 2016.

[61] Paul Voigtlaender, Jonathon Luiten, Philip H. S. Torr, and Bastian Leibe. Siam R-CNN: visual tracking by re-detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'20)*, pages 6577–6587, 2020.

[62] Tomas Vojíř, Jana Noskova, and Jiri Matas. Robust scale-adaptive mean-shift for tracking. *Pattern Recognition Letters*, 49:250–258, 2014.

[63] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *CVPR*, pages 1328–1338, 2019.

[64] Y. Wu, J. Lim, and M. H. Yang. Online object tracking: A benchmark. In *Comp. Vis. Patt. Recognition*, 2013.

[65] Yinda Xu, Zeyu Wang, Zuoxin Li, Ye Yuan, and Gang Yu. Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34-07, pages 12549–12556, 2020.

[66] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. *CoRR*, abs/2103.17154, 2021.

[67] Bin Yan, Dong Wang, Huchuan Lu, and Xiaoyun Yang. Alpha-refine: Boosting tracking performance by precise bounding box estimation. *arXiv preprint arXiv:2007.02024*, 2020.

[68] Bin Yan, Xinyu Zhang, Dong Wang, Huchuan Lu, and Xiaoyun Yang. Alpha-refine: Boosting tracking performance by precise bounding box estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5289–5298, 2021.

[69] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9657–9666, Oct 2019.

[70] Chunhui Zhang, and Kangkai Zhang Shiming Ge, and Dan Zeng. Accurate uav tracking with distance-injected overlap maximization. In *ACM MM*, pages 565–573, 2020.

[71] Bin Zhao, Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Generating masks from boxes by mining spatio-temporal consistencies in videos. *CoRR*, abs/2101.02196, 2021.

[72] Jinghao Zhou, Peng Wang, and Haoyang Sun. Discriminative and robust online learning for siamese visual tracking. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 13017–13024. AAAI Press, 2020.

	baseline			realtime			unsupervised
Tracker	EAO	A	R	EAO	A	R	AUC
●RPTMask	0.568①	0.764③	0.859③	0.368	0.659	0.704	0.683①
✚CFRPT	0.551②	0.745	0.853	0.325	0.612	0.684	0.655③
✖TransT_M	0.550③	0.742	0.869①	0.550①	0.742	0.869①	0.670②
▶TregPlus	0.546	0.753	0.852	0.440	0.706	0.777	0.615
▲DualTFRon	0.539	0.757	0.837	0.395	0.681	0.741	0.629
■F_TregPlus	0.537	0.753	0.848	0.490	0.738	0.812	0.626
★DualTFRst	0.536	0.755	0.836	0.512③	0.751②	0.816	0.623
●STARK_RT	0.534	0.781①	0.830	0.531②	0.780①	0.829③	0.631
✚DualTFR	0.527	0.748	0.826	0.509	0.746	0.813	0.619
✖RPT	0.524	0.692	0.866②	0.346	0.598	0.721	0.620
▶SiamUSCP	0.515	0.696	0.854	0.469	0.679	0.821	0.601
▲D3Sv2	0.514	0.712	0.843	0.313	0.654	0.614	0.607
■LWL_B2S	0.511	0.729	0.826	0.475	0.719	0.806	0.602
★TransT	0.507	0.748	0.817	0.507	0.748③	0.817	0.612
●TRASFUSTm	0.506	0.738	0.823	0.414	0.687	0.754	0.621
✚fRPT	0.501	0.691	0.851	0.486	0.689	0.840②	0.596
✖AlphaRef	0.484	0.752	0.776	0.477	0.745	0.776	0.608
▶RPT_AR	0.474	0.710	0.790	0.293	0.576	0.632	0.565
▲SuperDiMP_AR	0.473	0.725	0.772	0.370	0.659	0.702	0.580
■LWL	0.467	0.719	0.800	0.421	0.699	0.772	0.583
★SiamUSC	0.464	0.694	0.798	0.460	0.694	0.792	0.566
●SAMN_DiMP	0.463	0.703	0.776	0.391	0.673	0.716	0.510
✚keep_track	0.462	0.725	0.773	0.331	0.619	0.660	0.573
✖SAMN	0.457	0.723	0.774	0.439	0.698	0.770	0.537
▶KYS_AR	0.455	0.724	0.755	0.390	0.684	0.704	0.527
▲RTT	0.450	0.767②	0.727	0.387	0.697	0.696	0.610
■D3S	0.443	0.700	0.767	0.432	0.694	0.756	0.505
★DiMP_AR	0.432	0.717	0.722	0.415	0.710	0.713	0.558
●PrDiMP_AR	0.425	0.724	0.722	0.387	0.691	0.693	0.552
✚ATOM_AR	0.409	0.711	0.707	0.387	0.707	0.677	0.537
✖SiamEM_R	0.398	0.738	0.681	0.393	0.737	0.675	0.526
▶TRATMask	0.395	0.632	0.757	0.364	0.621	0.728	0.341
▲DCDAAR	0.355	0.709	0.625	0.352	0.710	0.619	0.419
■STM	0.311	0.739	0.593	0.285	0.698	0.570	0.457
★ACM	0.304	0.479	0.766	0.294	0.478	0.746	0.392
●KYS	0.282	0.454	0.758	0.265	0.447	0.732	0.370
✚PrDiMP	0.281	0.470	0.745	0.274	0.469	0.734	0.401
✖NSpaceRDAR	0.271	0.456	0.721	0.269	0.455	0.723	0.351
▶DiMP	0.270	0.449	0.736	0.266	0.451	0.722	0.374
▲ATOM	0.261	0.452	0.711	0.251	0.451	0.686	0.376
■deepmix	0.239	0.436	0.666	0.223	0.417	0.641	0.324
★SION	0.232	0.434	0.634	0.232	0.434	0.634	0.297
●ReptileFPN	0.213	0.423	0.619	0.119	0.384	0.317	0.275
✚TCLCF	0.200	0.425	0.588	0.200	0.425	0.588	0.247
✖ASMS	0.196	0.419	0.569	0.196	0.419	0.567	0.257
▶FSC2F	0.193	0.412	0.567	0.182	0.410	0.539	0.259
▲VITAL++	0.187	0.366	0.632	0.085	0.334	0.234	0.232
■CSRDCF	0.186	0.403	0.563	0.184	0.402	0.557	0.229
★SiamFC	0.173	0.408	0.499	0.168	0.417	0.476	0.220
●ANT	0.172	0.398	0.485	0.143	0.386	0.405	0.228
✚KCF	0.168	0.421	0.489	0.168	0.421	0.490	0.165
✖LGT	0.133	0.335	0.448	0.104	0.335	0.325	0.173
▶L1APG	0.083	0.359	0.222	0.073	0.370	0.165	0.102

Table 6. Results for VOT-ST2021 and VOT-RT2021 challenges. Expected average overlap (EAO), accuracy and robustness are shown. For reference, a no-reset average overlap AO [64] is shown under *Unsupervised*.