The Visual Object Tracking VOT2017 challenge results

Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder et al

The Visual Object Tracking VOT2017 challenge results

Matej Kristan[1], Aleš Leonardis[2], Jiri Matas[3], Michael Felsberg[4], Roman Pflugfelder[5], Luka Čehovin Zajc[1], Tomáš Vojíř[3], Gustav Häger[4], Alan Lukežič[1], Abdelrahman Eldesokey[4], Gustavo Fernández[5], Álvaro García-Martín[24], A. Muhic[1], Alfredo Petrosino[34], Alireza Memarmoghadam[29], Andrea Vedaldi[31], Antoine Manzanera[11], Antoine Tran[11], Aydın Alatan[20], Bogdan Mocanu[18,35], Boyu Chen[10], Chang Huang[15], Changsheng Xu[9], Chong Sun[10], Dalong Du[15], David Zhang[14], Dawei Du[28], Deepak Mishra[17], Erhan Gundogdu[6,20], Erik Velasco-Salido[24], Fahad Shahbaz Khan[4], Francesco Battistone[34], Gorthi R K Sai Subrahmanyam[17], Goutam Bhat[4], Guan Huang[15], Guilherme Bastos[25], Guna Seetharaman[22], Hongliang Zhang[21], Houqiang Li[32], Huchuan Lu[10], Isabela Drummond[25], Jack Valmadre[31], Jae-chan Jeong[12], Jae-il Cho[12], Jae-Yeong Lee[12], Jana Noskova[3], Jianke Zhu[36], Jin Gao[9], Jingyu Liu[9], Ji-Wan Kim[12], João F. Henriques[31], José M. Martínez[24], Junfei Zhuang[7], Junliang Xing[9], Junyu Gao[9], Kai Chen[16], Kannappan Palaniappan[30], Karel Lebeda[23], Ke Gao[30], Kris M. Kitani[8], Lei Zhang[14], Lijun Wang[10], Lingxiao Yang[14], Longyin Wen[13], Luca Bertinetto[31], Mahdieh Poostchi[30], Martin Danelljan[4], Matthias Mueller[19], Mengdan Zhang[9], Ming-Hsuan Yang[27], Nianhao Xie[21], Ning Wang[32], Ondrej Miksik[31], P. Moallem[29], Pallavi Venugopal M[17], Pedro Senna[25], Philip H. S. Torr[31], Qiang Wang[9], Qifeng Yu[21], Qingming Huang[28], Rafael Martín-Nieto[24], Richard Bowden[33], Risheng Liu[10], Ruxandra Tapu[18,35], Simon Hadfield[33], Siwei Lyu[26], Stuart Golodetz[31], Sunglok Choi[12], Tianzhu Zhang[9], Titus Zaharia[18], Vincenzo Santopietro[34], Wei Zou[9], Weiming Hu[9], Wenbing Tao[16], Wenbo Li[26], Wengang Zhou[32], Xianguo Yu[21], Xiao Bian[13], Yang Li[36], Yifan Xing[8], Yingruo Fan[7], Zheng Zhu[9,28], Zhipeng Zhang[9], and Zhiqun He[7]

[1]University of Ljubljana, Slovenia
[2]University of Birmingham, United Kingdom
[3]Czech Technical University, Czech Republic
[4]Linköping University, Sweden
[5]Austrian Institute of Technology, Austria
[6]Aselsan Research Center, Turkey
[7]Beijing University of Posts and Telecommunications, China
[8]Carnegie Mellon University, USA
[9]Chinese Academy of Sciences, China
[10]Dalian University of Technology, China
[11]ENSTA ParisTech, Université de Paris-Saclay, France
[12]ETRI, Korea
[13]GE Global Research, USA
[14]Hong Kong Polytechnic University, Hong Kong
[15]Horizon Robotics, Inc, China
[16]Huazhong University of Science and Technology, China
[17]Indian Institute Space Science and Technology Trivandrum, India
[18]Institut Mines-Telecom/ TelecomSudParis, France
[19]KAUST, Saudi Arabia
[20]Middle East Technical University, Turkey

[21]National University of Defense Technology, China
[22]Naval Research Lab, USA
[23]The Foundry, United Kingdom
[24]Universidad Autónoma de Madrid, Spain
[25]Universidade Federal de Itajubá, Brazil
[26]University at Albany, USA
[27]University of California, Merced, USA
[28]University of Chinese Academy of Sciences, China
[29]University of Isfahan, Iran
[30]University of Missouri-Columbia, USA
[31]University of Oxford, United Kingdom
[32]University of Science and Technology of China, China
[33]University of Surrey, United Kingdom
[34]University Parthenope of Naples, Italy
[35]University Politehnica of Bucharest, Romania
[36]Zhejiang University, China

Abstract

The Visual Object Tracking challenge VOT2017 is the fifth annual tracker benchmarking activity organized by the VOT initiative. Results of 51 trackers are presented; many are state-of-the-art published at major computer vision conferences or journals in recent years. The evaluation included the standard VOT and other popular methodologies and a new "real-time" experiment simulating a situation where a tracker processes images as if provided by a continuously running sensor. Performance of the tested trackers typically by far exceeds standard baselines. The source code for most of the trackers is publicly available from the VOT page. The VOT2017 goes beyond its predecessors by (i) improving the VOT public dataset and introducing a separate VOT2017 sequestered dataset, (ii) introducing a real-time tracking experiment and (iii) releasing a redesigned toolkit that supports complex experiments. The dataset, the evaluation kit and the results are publicly available at the challenge website[1].

1. Introduction

Visual tracking is a popular research area with over forty papers published annually at major conferences. Over the years, several initiatives have been established to consolidate performance measures and evaluation protocols in different tracking subfields. The longest lasting PETS [78] proposed evaluation frameworks motivated mainly by surveillance applications. Other evaluation methodologies focus on event detection, (e.g., CAVIAR[2], i-LIDS [3], ETISEO[4]), change detection [22], sports analytics (e.g., CVBASE[5]), faces (e.g. FERET [50] and [28]), long-term tracking [6] and multiple target tracking [35, 61][7]. Recently, workshops focusing on performance evaluation issues in computer vision [8] have been organized and an initiative covering several video challenges has emerged [9].

In 2013, VOT — the Visual Object Tracking initiative — was started to address performance evaluation of short-term visual object trackers. The primary goal of VOT is establishing datasets, evaluation measures and toolkits as well as creating a platform for discussing evaluation-related issues. Since 2013, four challenges have taken place in conjunction with ICCV2013 (VOT2013 [32]), ECCV2014 (VOT2014 [33]), ICCV2015 (VOT2015 [30]) and ECCV2016 (VOT2016 [29]) respectively.

Due to the growing interest in (thermal) infrared (TIR) imaging, a new sub-challenge on tracking in TIR sequences was launched and run in 2015 (VOT-TIR2015 [19]) and 2016 (VOT-TIR2016 [20]). In 2017, the TIR challenge results are reported alongside the RGB results.

This paper presents the VOT2017 challenge, organized in conjunction with the ICCV2017 Visual Object Tracking workshop, and the results obtained. Like VOT2013, VOT2014, VOT2015 and VOT2016, the VOT2017 challenge considers single-camera, single-target, model-free, causal trackers, applied to short-term tracking. The *model-free* property means that the only training information provided is the bounding box in the first frame. The *short-term* tracking means that trackers are assumed not to be capable of performing successful re-detection after the target is lost and they are therefore reset after such event. *Causality* requires that the tracker does not use any future frames, or frames prior to re-initialization, to infer the object position in the current frame. In the following, we overview the most closely related work and point out the contributions of VOT2017.

1.1. Related work

Performance evaluation of short-term visual object trackers has received significant attention in the last five years [32, 33, 30, 31, 29, 68, 60, 77, 39, 40, 45, 41]. The currently most widely used methodologies developed from three benchmark papers: the "Visual Object Tracking challenge" (VOT) [32], the "Online Tracking Benchmark" (OTB) [77] and the "Amsterdam Library of Ordinary Videos" (ALOV) [60]. The benchmarks differ in the adopted performance measures, evaluation protocols and datasets. In the following we briefly overview these differences.

1.1.1 Performance measures

The OTB- and ALOV-related methodologies, like [77, 60, 39, 40], evaluate a tracker by initializing it on the first frame and letting it run until the end of the sequence, while the VOT-related methodologies [32, 33, 30, 68, 31] reset the tracker once it drifts of the target. ALOV [60] defines tracking performance as the F-measure at 0.5 overlap threshold between the ground truth and the bounding boxes predicted by the tracker. OTB [77] generates a plot showing the percentage of frames where the overlap exceeds a threshold, for different threshold values. The primary measure is the area under the curve, which was recently shown [68] to be equivalent to the average overlap (AO) between the ground truth and predicted bounding boxes over all test sequences. The strength of AO is in its simplicity and ease of interpretation. A downside is that, due to lack of resets, this is a

[1] http://votchallenge.net
[2] http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1
[3] http://www.homeoffice.gov.uk/science-research/hosdb/i-lids
[4] http://www-sop.inria.fr/orion/ETISEO
[5] http://vision.fe.uni-lj.si/cvbase06/
[6] http://www.micc.unifi.it/LTDT2014/
[7] https://motchallenge.net
[8] https://hci.iwr.uni-heidelberg.de/eccv16ws-datasets
[9] http://videonet.team

biased estimation of average overlap with a potentially large variance. In contrast, the bias and variance are reduced in reset-based estimators [31].

Čehovin et al. [67, 68] analyzed the correlation between popular performance measures and identified accuracy and robustness as two weakly-correlated measures with high interpretability. The accuracy is the average overlap during successful tracking periods and the robustness measures how many times the tracker drifted from the target and had to be reset. The VOT2013 [32] adopted these as the core performance measures. To promote the notion that some trackers might perform equally well, a ranking methodology was introduced, in which trackers are merged into the same rank based on statistical tests on performance difference. In VOT2014 [33], the notion of practical difference was introduced into rank merging as well to address the noise in the ground truth annotation. For the different rank generation strategies please see [31]. Accuracy-robustness ranking plots were proposed to visualize the results [32]. A drawback of the AR-rank plots is that they do not show the absolute performance. To address this, VOT2015 [30] adopted AR-raw plots from [68] to show the absolute average performance.

The VOT2013 [32] and VOT2014 [33] selected the winner of the challenge by averaging the accuracy and robustness ranks, meaning that the accuracy and robustness were treated as equally important "competitions". But since ranks lose the absolute performance difference between trackers, and are meaningful only in the context of a *fixed set* of evaluated trackers, the rank averaging was abandoned in later challenges.

Since VOT2015 [30], the primary measure is the expected average overlap (EAO) that combines the raw values of per-frame accuracies and failures in a principled manner and has a clear practical interpretation. The EAO measures the expected no-reset overlap of a tracker run on a short-term sequence. The EAO reflects the same property as the AO [77] measure, but, since it is computed from the VOT reset-based experiment, it does not suffer from the large variance and has a clear relation to the definition of short-term tracking.

In VOT2016 [29] the experiments indicated that EAO is stricter than AO in penalizing a tracker for poor performance on a subset of sequences. The reason is that a tracker is more often reset on sequences that are most challenging to track, which reduces the EAO. On the other hand the AO does not use resets which makes poor performance on a part of a dataset difficult to detect. Nevertheless, since the AO measure is still widely used in the tracking community, this measure and the corresponding no-reinitialization experiment was included in the VOT challenges since 2016 [29].

VOT2014 [33] recognized speed as an important factor in many applications and introduced a measure called the *equivalent filter operations* (EFO) that partially accounts for the speed of a computer used for tracker analysis. While this measure at least partially normalizes speed measurements obtained over different machines, it cannot completely address hardware issues. In VOT2016 [29] it was reported that significant EFO errors could be expected for very fast MatLab trackers due to the MatLab start-up overhead.

The VOT2015 committee pointed out that published papers more often than not reported presented trackers as scoring top performance on a standard benchmark. However, a detailed inspection of the papers showed that sometimes the results were reported only on a part of the benchmarks or that the top performing method on the benchmark were excluded from the comparison. This significantly skews the perspective on the current state-of-the-art and tends to force researchers into maximizing a single performance score, albeit only virtually by manipulating the presentation of the experiments. In response, the VOT has started to promote the approach that it should be sufficient to show a good-enough performance on benchmarks and that the authors (as well as reviewers) should focus on the novelty and the quality of the theory underpinning the tracker. VOT2015 [30] thus introduced a notion of state-of-the-art bound. This value is computed as the average performance of the trackers participating in the challenge that were published at top recent conferences. Any tracker exceeding this performance on the VOT benchmark is considered state-of-the-art according to the VOT standards.

For TIR sequences, two main challenges have been organized in the past. Within the series of workshops on Performance Evaluation of Tracking and Surveillance (PETS) [78], thermal infrared challenges have taken place twice, in 2005 and 2015. PETS challenges addressed multiple research areas such as detection, multi-camera/long-term tracking, and behavior (threat) analysis.

In contrast, the VOT-TIR2015 and 2016 challenges have focused on the problem of short-term tracking [19, 20]. The 2015 challenge has been based on a specifically compiled LTIR dataset [3], as available datasets for evaluation of tracking in thermal infrared had become outdated. The lack of an accepted evaluation dataset often leads to comparisons on proprietary datasets. Together with inconsistent performance measures it made it difficult to systematically assess the progress of the field. VOT-TIR2015 and 2016 adopted the well-established VOT methodology.

In 2016, the dataset for the VOT-TIR challenge was updated with more difficult sequences, since the 2015 challenge was close to saturated, i.e., near perfect performance was reported for top trackers [20]. Since the best performing method from 2015, based on the SRDCF [15], was not significantly outperformed in the 2016 challenge, VOT-TIR2016 has been re-opened in conjunction with VOT2017, and since no methodological changes have been made, the

results are reported as part of this paper instead of a separate one. For all technical details of the TIR challenge, the reader is referred to [20].

1.1.2 Datasets

Most tracking datasets [77, 39, 60, 40, 45] have partially followed the trend in computer vision of increasing the number of sequences. This resulted in impressive collections of annotated datasets, which have played an important role in tracker development and consistent evaluation over the last five years. Much less attention has being paid to the diversity of the data nd the quality of the content and annotation. For example, some datasets disproportionally represent grayscale or color sequences and in most datasets an attribute (e.g., occlusion) is assigned to the entire sequence even if it occupies only a fragment of the sequence. We have noticed several issues with bounding box annotation in commonly used datasets. Many datasets, however, assume the errors will average out on a large set of sequences and adopt the assumption that the dataset quality is correlated with its size.

In contrast, the VOT [31] has argued that large datasets do not necessarily imply diversity or richness in attributes. Over the last four years, VOT [32, 33, 30, 31, 29] has focused on developing a methodology for automatic construction and annotation of moderately large datasets from a large pool of sequences. This methodology is unique in that it optimizes diversity in visual attributes while focusing on sequences which are difficult to track. In addition, the VOT [32] introduced per-frame annotation with attributes, since global attribute annotation amplifies attribute crosstalk in performance evaluation [41] and biases performance toward the dominant attribute [31]. To account for ground truth annotation errors, VOT2014 [33] introduced the notion of practical difference, which is a performance difference under which two trackers cannot be considered as performing differently. VOT2016 [29] proposed an automatic ground truth bounding box annotation from per-frame segmentation masks, which requires semi-supervised segmentation of all frames. Their approach automatically estimates the practical difference values for each sequence.

Most closely related to the work described in this paper are the recent VOT2013 [32], VOT2014 [33], VOT2015 [30] and VOT2016 [29] challenges. Several novelties in benchmarking short-term trackers were introduced through these challenges. They provide a cross-platform evaluation kit with tracker-toolkit communication protocol [9], allowing easy integration with third-party trackers, per-frame annotated datasets and state-of-the-art performance evaluation methodology for in-depth tracker analysis from several performance aspects. The results were published in joint papers [32], [33], [30] and [29] with more than 140 coauthors.

The most recent challenge contains 70 trackers evaluated on primary VOT measures as well as the widely used OTB [77] measure. To promote reproducibility of results and foster advances in tracker development, the VOT2016 invited participants to make their trackers publicly available. Currently 38 state-of-the-art trackers along with their source code are available at the VOT site[10]. These contributions by and for the community make the VOT2016 the largest and most advanced benchmark. The evaluation kit, the dataset, the tracking outputs and the code to reproduce all the results are made freely-available from the VOT initiative homepage[11]. The advances proposed by VOT have arguably influenced the development of related methodologies and benchmark papers and have facilitated development of modern trackers by helping tease out promising tracking methodologies.

1.2. The VOT2017 challenge

VOT2017 follows the VOT2016 challenge and considers the same class of trackers. The dataset and evaluation toolkit are provided by the VOT2017 organizers. The evaluation kit records the output bounding boxes from the tracker, and if it detects tracking failure, re-initializes the tracker. The authors participating in the challenge were required to integrate their tracker into the VOT2017 evaluation kit, which automatically performed a standardized experiment. The results were analyzed according to the VOT2017 evaluation methodology. The toolkit conducted the main OTB [77] experiment in which a tracker is initialized in the first frame and left to track until the end of the sequence without resetting.

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 to this sequence. The organizers of VOT2017 were allowed to participate in the challenge, but did not compete for the winner of the VOT2017 challenge title. Further details are available from the challenge homepage[12].

The novelties of VOT2017 with respect to VOT2013, VOT2014, VOT2015 and VOT2016 are the following: (i) The dataset from VOT2016 has been updated. As in previous years, sequences that were least challenging were replaced by new sequences while maintaining the attribute

[10] http://www.votchallenge.net/vot2016/trackers.html
[11] http://www.votchallenge.net
[12] http://www.votchallenge.net/vot2017/participation.html

distribution. The ground truth annotation has been re-examined and corrected in the entire dataset. We call the set of sequences "the VOT2017 public dataset". (ii) A separate sequestered dataset was constructed with similar statistics to the public VOT2017 dataset. This dataset was not disclosed and was used to identify the winners of the VOT2017 challenge. (iii) A new experiment dedicated to evaluating real-time performance has been introduced. (iv) The VOT toolkit has been re-designed to allow the real-time experiment. Transition to the latest toolkit was a precondition for participation. (iv) The VOT-TIR2016 subchallenge, which deals with tracking in infrared and thermal imagery [19] has been reopened as VOT-TIR2017.

2. The VOT2017 datasets

Results of VOT2016 showed that the dataset was not saturated, but that some sequences have been successfully tracked by most trackers. In the VOT2017 public dataset the least challenging sequences in VOT2016 were replaced. The VOT committee acquired 10 pairs of new challenging sequences (i.e. 20 new sequences), which had not been part of existing tracking benchmarks. Each pair consists of two roughly equally challenging sequences similar in content. Ten sequences, one of each pair, were used to replace the ten least challenging sequences in VOT2016 (see Figure 2). The level of difficulty was estimated using the VOT2016 results [29].

In response to yearly panel discussions at VOT workshops, it was decided to construct another dataset, which will not be disclosed to the community, but will be used to identify the VOT2017 winners. This is called the VOT2017 *sequestered* dataset and was constructed to be close in attribute distribution to the VOT2017 public dataset with the same number of sequences (sixty).

Ten remaining sequences of the pairs added to the VOT2017 public dataset were included to the sequestered dataset. The remaining fifty sequences in the sequestered dataset were sampled from a large pool of sequences collected over the years by VOT (approximately 390 sequences) as follows. Distances between sequences in VOT2017 public dataset and sequences in the pool were computed. The distance was defined as Euclidean distance in the 11-dimensional global attribute space typically used in the VOT sequence clustering protocol [29]. For each sequence in the VOT2017 public dataset, all sequences in the pool with distance smaller than three times the minimal distance were identified. Among these, a sequence with the highest difficulty level estimated by the VOT2016 methodology [29] was selected for the VOT2017 sequestered dataset. The selected sequence was removed from the pool and the process was repeated for the remaining forty-nine sequences.

A semi-automatic segmentation approach by Vojíř and Matas [72] was applied to segment the target in all frames and bounding boxes were fitted to the segmentation masks according to the VOT2016 methodology [29]. All bounding boxes were manually inspected. The boxes that were incorrectly placed by the automatic algorithm were manually repositioned. Figure 1 shows the practical difference thresholds on the VOT2017 dataset estimated by the bounding box fitting methodology [29].

Following the protocol introduced in VOT2013 [32], all sequences in the VOT2017 public dataset are per-frame annotated by the following visual attributes: (i) occlusion, (ii) illumination change, (iii) motion change, (iv) size change, (v) camera motion. Frames that did not correspond to any of the five attributes were denoted as (vi) unassigned.

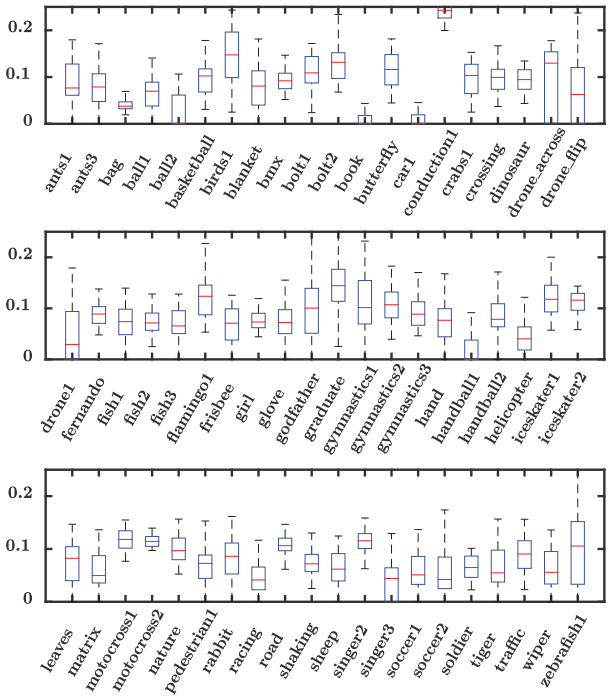

Figure 1. Practical difference plots for all sequences in the VOT2017 public dataset. For each sequence a distribution of overlap values between bounding boxes, which equally well fit the potentially noisy object segmentations are shown. The practical difference thresholds are denoted in red.

3. Performance evaluation methodology

Since VOT2015 [30], three primary measures are used to analyze tracking performance: accuracy (A), robustness (R) and expected average overlap (AEO). In the following, these are briefly overviewed and we refer to [30, 31, 68] for further details.

The VOT challenges apply a reset-based methodology. Whenever a tracker predicts a bounding box with zero overlap with the ground truth, a failure is detected and the

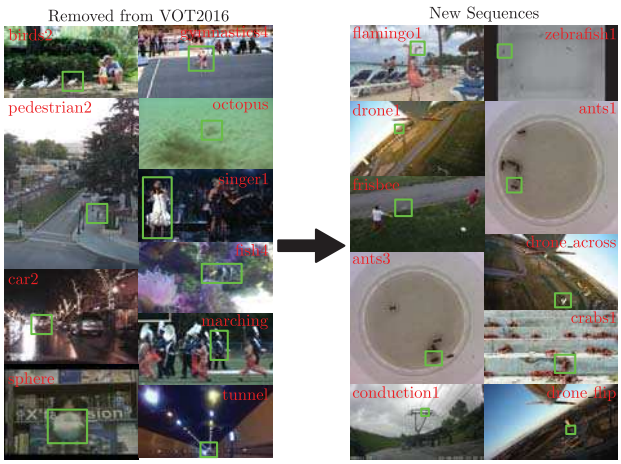

Figure 2. Images from the VOT2016 sequences (left column) that were replaced by new sequences in VOT2017 (right column).

tracker is re-initialized five frames after the failure. Accuracy and robustness [68] are the primary measures used to probe tracker performance in the reset-based experiments. The accuracy is the average overlap between the predicted and ground truth bounding boxes during successful tracking periods. The robustness measures how many times the tracker loses the target (fails) during tracking. The potential bias due to resets is reduced by ignoring ten frames after re-initialization in the accuracy measure, which is quite a conservative margin [31].

Stochastic trackers are run 15 times on each sequence to reduce the variance of their results. Per-frame accuracy is obtained as an average over these runs. Averaging per-frame accuracies gives per-sequence accuracy, while per-sequence robustness is computed by averaging failure rates over different runs.

The third primary measure, called the expected average overlap (EAO), is an estimator of the average overlap a tracker is expected to attain on a large collection of short-term sequences with the same visual properties as the given dataset. This measure addresses the problem of increased variance and bias of AO [77] measure due to variable sequence lengths. Please see [30] for further details on the average expected overlap measure.

VOT2016 argued that raw accuracy and robustness values should be preferred to their ranked counterparts. The ranking is appropriate to test whether performance difference is consistently in favor of one tracker over the others, but has been abandoned for ranking large numbers of trackers since averaging ranks ignores the absolute differences.

In addition to the standard reset-based VOT experiment, the VOT2017 toolkit carried out the OTB [77] no-reset experiment. The tracking performance on this experiment was evaluated by the primary OTB measure, the average overlap (AO).

3.1. The VOT2017 real-time experiment

The VOT has been promoting the importance of speed in tracking since the introduction of the EFO speed measurement unit in VOT2014. But these results do not reflect a realistic performance in real-time applications. In these applications, the tracker is required to report the bounding box for each frame at frequency higher than or equal to the video frame rate. The existing toolkits and evaluation systems do not support such advanced experiments, therefore the VOT toolkit has been re-designed.

The basic real-time experiment has been included in the VOT2017 challenge and was conducted as follows. The toolkit initializes the tracker in the first frame and waits for the bounding box response from the tracker (responding to each frame individually is possible due to the interactive communication between the tracker and the toolkit [9]). If a new frame becomes available before the tracker responds, a zero-order hold model is used, i.e., the last reported bounding box is assumed as the reported tracker output at the available frame.

The toolkit applies the reset-based VOT evaluation protocol by resetting the tracker whenever the tracker bounding box does not overlap with the ground truth. The VOT frame skipping is applied as well to reduce the correlation between resets.

The predictive power of his experiment is limited by fact that the tracking speed depends on the type of hardware used and the programming effort and skill, which is expected to vary significantly among the submissions. Nevertheless, this is the first published attempt to evaluate trackers in a simulated real-time setup.

3.2. VOT2017 winner identification protocol

The VOT2017 challenge winner was identified as follows. Trackers were ranked with respect to the EAO measure on the VOT2017 public dataset. The top 10 trackers were then run on a high performance cluster using the VOT2017 sequestered dataset and again ranked with respect to the EAO measure. The top-performing tracker that was not submitted by organizers was identified as the VOT2017 challenge winner. An additional requirement was that the authors have to make the tracker source code available to the tracking community.

Due to limited resources, the VOT2017 real-time winner was not identified on the sequestered dataset, but based on the results obtained on the VO2017 public dataset. The EAO measure was used to rank the tracker results from the real-time experiment. The same authorship and open source requirements as in the VOT2017 challenge winner protocol were applied.

4. VOT2017 analysis and results

4.1. Trackers submitted

In all, 38 valid entries were submitted to the VOT2017 challenge. Each submission included the binaries or source code that allowed verification of the results if required. The VOT2017 committee and associates additionally contributed 13 baseline trackers. For these, the default parameters were selected, or, when not available, were set to reasonable values. Thus in total 51 trackers were tested on the VOT2017 challenge. In the following we briefly overview the entries and provide the references to original papers in the Appendix A where available.

Of all participating trackers, 67% applied generative and 33% applied discriminative models. Most trackers – 73% – used holistic model, while 27% of the participating trackers used part-based models. Most trackers applied either a locally uniform dynamic model [13] (53%), a nearly-constant-velocity (20%), or a random walk dynamic model (22%), while a few trackers applied a higher order dynamic model (6%).

The trackers were based on various tracking principles: 17 trackers (31%) were based on CNN matching (ATLAS (A.26), CFWCR (A.14), CRT (A.2), DLST (A.15), ECO (A.30), CCOT (A.36), FSTC (A.33), GMD (A.29), GMDNetN (A.9), gnet (A.16), LSART (A.24), MCCT (A.4), MCPF (A.18), RCPF (A.34), SiamDCF (A.23), SiamFC (A.21) and UCT (A.19)), 25 trackers (49 %) applied discriminative correlation filters (ANT (A.1), CFCF (A.10), CFWCR (A.14), DPRF (A.27), ECO (A.30), ECOhc (A.31), gnet (A.16), KCF (A.8), KFebT (A.12), LDES (A.32), MCCT (A.4), MCPF (A.18), MOSSE_CA (A.35), RCPF (A.34), SiamDCF (A.23), SSKCF (A.25), Staple (A.20), UCT (A.19), CSRDCF (A.38), CSRDCFf (A.39), CSRDCF++ (A.40), dpt (A.41), SRDCF (A.50), DSST (A.42) and CCOT (A.36)), two (4%) trackers (BST (A.17) and Struck2011 (A.51)) were based on structured SVM, 5 trackers (10%) were based on Mean Shift (ASMS (A.6), KFebT (A.12), SAPKLTF (A.13), SSKCF (A.25) and MSSA (A.49)), 5 trackers (10%) applied optical flow (ANT (A.1), FoT (A.7), HMMTxD (A.11), FragTrack (A.43) and CMT (A.37)), one tracker was based on line segments matching (LTFLO (A.5)), one on a generalized Hough transform (CHT (A.28)) and three trackers (HMMTxD (A.11), KFEbT (A.12) and SPCT (A.22)) were based on tracker combination.

[13] 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.

4.2. The baseline experiment

The results are summarized in the AR-raw plots and EAO curves in Figure 3 and the expected average overlap plots in Figure 4. The values are also reported in Table 1.

The top ten trackers according to the primary EAO measure (Figure 4) are LSART (A.24), CFWCR (A.14), CFCF (A.10), ECO (A.30), gnet (A.16), MCCT (A.4), CCOT (A.36), CSRDCF (A.38), SiamDCF (A.23), MCPF (A.18). All these trackers apply a discriminatively trained correlation filter on top of multidimensional features. In most trackers, the correlation filter is trained in a standard form via circular shifts, except in LSART (A.24) and CRT (A.2) that treat the filter as a fully-connected layer and train it by a gradient descent.

The top ten trackers vary significantly in features. Apart from CSRDCF (A.38) that applies only HOG [47] and color-names [65], the trackers apply CNN features, which are in some cases combined with hand-crafted features. In almost all cases the CNN is a standard pre-trained CNN for object class detection except in the case of CFCF (A.10) and SiamDCF (A.23) which use feature training. Both of these trackers train their CNN representations on a tracking task from many videos to learn features that maximize discriminative correlation filter response using the approaches from [23], [75] and [5]. The CFCF (A.10) uses the first, fifth and sixth convolutional layers of VGG-M-2048 fine-tuned on the tracking task in combination with HOG [47] and Colour Names (CN) [65].

The top performer on public dataset LSART (A.24) decomposes the target into patches and applies a weighted combination of patch-wise similarities into a kernelized ridge regression formulated as a convolutional network. Spatial constraints are used to force channels in specializing to different parts of the target. A distance transform pooling is used to merge the channels. The network uses pre-learned VGG16 [59] layers 4-3, HoG [47] and colour names as low-level filters.

The top trackers in EAO are also among the most robust trackers, which means that they are able to track longer without failing. The top trackers in robustness (Figure 3) are LSART (A.24), CFWCR (A.14), ECO (A.30) and gnet (A.16). On the other hand, the top performers in accuracy are SSKCF (A.25), Staple (A.20) and MCCT (A.4). The SSKCF and Staple are quite similar in design and apply a discriminative correlation filter on hand-crafted features combined with color histogram back-projection.

The trackers which have been considered as baselines even five years ago, i.e., MIL (A.48), and IVT (A.44) are positioned at the lower part of the AR-plots and at the tail of the EAO rank list. It is striking that even trackers which are often considered as baselines in recent papers, e.g., Struck [24] and KCF [26] are positioned in the lower quarter of the EAO ranks. This speaks of the significant

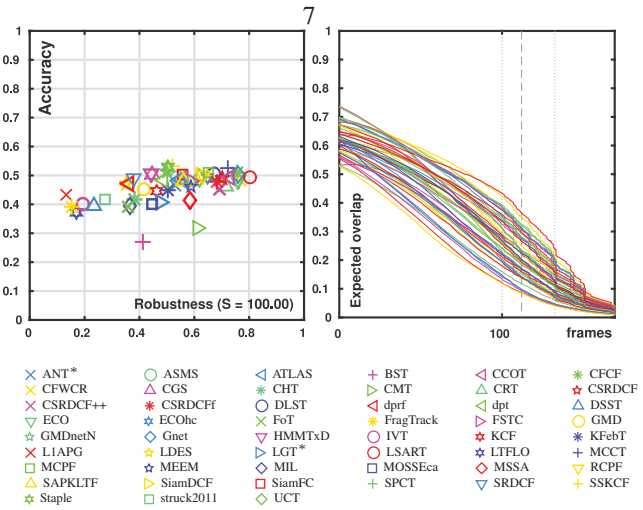

Figure 3. The AR-raw plots generated by sequence pooling (left) and EAO curves (right).

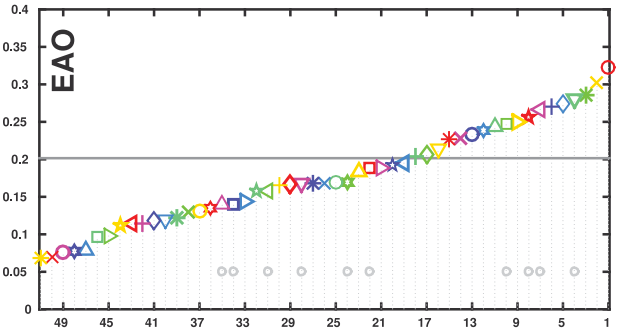

Figure 4. Expected average overlap curve (left) and expected average overlap graph (right) with trackers ranked from right to left. The right-most tracker is the top-performing according to the VOT2017 expected average overlap values. The dashed horizontal line denotes the average performance of ten state-of-the-art trackers published in 2016 and 2017 at major computer vision venues. These trackers are denoted by gray circle in the bottom part of the graph.

quality of the trackers submitted to VOT2017. In fact, ten tested trackers have been recently (2016 or later) published at major computer vision conferences and journals. These trackers are indicated in Figure 4, along with their average performance, which constitutes a very strict VOT2017 state-of-the-art bound. Over 35% of submitted trackers exceed this bound.

The number of failures with respect to the visual attributes is shown in Figure 5. LSART (A.24) fails least often among all trackers on camera motion, motion change, unassigned and scores second-best on illumination change. The top performer on illumination change is CFCF (A.10) and scores second best on size change attribute. The top performer on size change and occlusion is MCCT (A.4),

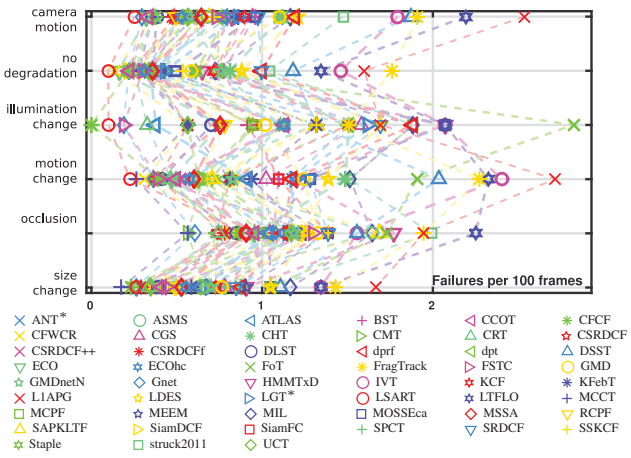

Figure 5. Failure rate with respect to the visual attributes.

which also scores as second-best on motion change.

We have evaluated the difficulty level of each attribute by computing the median of robustness and accuracy over each attribute. According to the results in Table 2, the most challenging attributes in terms of failures are occlusion, illumination change and motion change, followed by camera motion and scale change. The occlusion and motion change are the most difficult attributes for tracking accuracy as well.

In addition to the baseline reset-based VOT experiment, the VOT2016 toolkit also performed the OTB [77] no-reset (OPE) experiment. Figure 6 shows the OPE plots, while the AO overall measure is given in Table 1. According to the AO measure, the three top performing trackers are MCPF (A.18), LSART (A.24) and RCPF (A.34). Two of these trackers are among top 10 in EAO as well, i.e, LSART (ranked first) and MCPF (ranked tenth). The RCPF is a particle filter that applies a discriminative correlation filter for visual model, uses hand-crafted and deep features and applies a long-short-term adaptation. The adaptation strategy most likely aids in target re-localization after failure, which explains the high AO score.

4.2.1 The VOT2017 winner identification

The baseline experiment with the top 10 trackers from Table 1 was repeated on a sequestered dataset. The scores are shown in Table 3. The top tracker according to the EAO is CCOT (A.36), but this tracker is co-authored by the VOT organizers. According to the VOT winner rules, the VOT2017 challenge winner is therefore the CFCF tracker (A.10).

4.3. The realtime experiment

The EAO scores and AR-raw plots for the real-time experiment are shown in Figure 7 and Figure 8.

The top eleven real-time trackers are

		baseline			realtime			unsupervised	
	Tracker	**EAO**	**A**	**R**	**EAO**	**A**	**R**	**AO**	**Implementation**
1.	LSART	0.323 ①	0.493	0.218 ①	0.055	0.386	1.971	0.437 ②	S M G
2.	CFWCR	0.303 ②	0.484	0.267 ②	0.062	0.393	1.864	0.370	D M C
3.	CFCF	0.286 ③	0.509	0.281	0.059	0.339	1.723	0.380	D M G
4.	ECO	0.280	0.483	0.276 ③	0.078	0.449	1.466	0.402	D M G
5.	Gnet	0.274	0.502	0.276 ③	0.060	0.353	1.836	0.419	D M C
6.	MCCT	0.270	0.525 ③	0.323	0.060	0.353	1.775	0.428	D M C
7.	CCOT	0.267	0.494	0.318	0.058	0.326	1.461	0.390	D M G
8.	CSRDCF	0.256	0.491	0.356	0.099	0.477	1.054	0.342	D M G
9.	SiamDCF	0.249	0.500	0.473	0.135	0.503 ③	0.988	0.340	D M G
10.	MCPF	0.248	0.510	0.427	0.060	0.325	1.489	0.443 ①	S M G
11.	CRT	0.244	0.463	0.337	0.068	0.400	1.569	0.370	S P G
12.	ECOhc	0.238	0.494	0.435	0.177 ③	0.494	0.571 ②	0.335	D M C
13.	DLST	0.233	0.506	0.396	0.057	0.381	2.018	0.406	S M G
14.	CSRDCF++	0.229	0.453	0.370	0.212 ①	0.459	0.398 ①	0.298	D C G
15.	CSRDCFf	0.227	0.479	0.384	0.158	0.475	0.646	0.327	D C G
16.	RCPF	0.215	0.501	0.458	0.078	0.334	1.002	0.435 ③	S M G
17.	UCT	0.206	0.490	0.482	0.145	0.490	0.777	0.375	D M G
18.	SPCT	0.204	0.472	0.548	0.069	0.374	1.831	0.333	D M C
19.	ATLAS	0.195	0.488	0.595	0.117	0.455	1.035	0.341	D C G
20.	MEEM	0.192	0.463	0.534	0.072	0.407	1.592	0.328	S M C
21.	FSTC	0.188	0.480	0.534	0.051	0.389	2.365	0.334	D M G
22.	SiamFC	0.188	0.502	0.585	0.182 ②	0.502	0.604 ③	0.345	D M G
23.	SAPKLTF	0.184	0.482	0.581	0.126	0.470	0.922	0.334	D C C
24.	Staple	0.169	0.530 ②	0.688	0.170	0.530 ②	0.688	0.335	S M C
25.	ASMS	0.169	0.494	0.623	0.168	0.489	0.627	0.337	S C C
26.	ANT	0.168	0.464	0.632	0.059	0.403	1.737	0.279	D M C
27.	KFebT	0.168	0.450	0.688	0.169	0.451	0.684	0.296	D C C
28.	HMMTxD	0.168	0.506	0.815	0.074	0.404	1.653	0.330	D C C
29.	MSSA	0.167	0.413	0.538	0.124	0.422	0.749	0.327	D C C
30.	SSKCF	0.166	0.533 ①	0.651	0.164	0.530 ①	0.656	0.383	D C C
31.	DPT	0.158	0.486	0.721	0.126	0.483	0.899	0.315	D C C
32.	GMDnetN	0.157	0.513	0.696	0.079	0.312	0.946	0.402	S M C
33.	LGT	0.144	0.409	0.742	0.059	0.349	1.714	0.225	S M C
34.	MOSSEca	0.141	0.400	0.805	0.139	0.400	0.810	0.240	D M C
35.	CGS	0.140	0.504	0.806	0.075	0.290	0.988	0.338	S M C
36.	KCF	0.135	0.447	0.773	0.134	0.445	0.782	0.267	S C C
37.	GMD	0.130	0.453	0.878	0.076	0.416	1.672	0.252	S C G
38.	FoT	0.130	0.393	1.030	0.130	0.393	1.030	0.143	S C C
39.	CHT	0.122	0.418	0.960	0.123	0.417	0.937	0.246	D C C
40.	SRDCF	0.119	0.490	0.974	0.058	0.377	1.999	0.246	S M C
41.	MIL	0.118	0.393	1.011	0.069	0.376	1.775	0.180	S C C
42.	BST	0.115	0.269	0.883	0.052	0.267	1.662	0.146	S C C
43.	DPRF	0.114	0.470	1.021	-	-	-	0.258	D M C
44.	LDES	0.111	0.471	1.044	0.113	0.471	1.030	0.225	D M C
45.	CMT	0.098	0.318	0.492	0.079	0.327	0.642	0.125	S P C
46.	Struck2011	0.097	0.418	1.297	0.093	0.419	1.367	0.197	D C C
47.	DSST	0.079	0.395	1.452	0.077	0.396	1.480	0.172	S C C
48.	LTFLO	0.078	0.372	1.770	0.054	0.303	1.995	0.118	D C C
49.	IVT	0.076	0.400	1.639	0.065	0.386	1.854	0.130	S M C
50.	L1APG	0.069	0.432	2.013	0.062	0.351	1.831	0.159	S M C
51.	FragTrack	0.068	0.390	1.868	0.068	0.316	1.480	0.180	S C C

Table 1. The table shows expected average overlap (EAO), as well as accuracy and robustness raw values (A,R) for the baseline and the realtime experiments. For the unsupervised experiment the no-reset average overlap AO [76] is used. The last column contains implementation details (first letter: (D)eterministic or (S)tohastic, second letter: tracker implemented in (M)atlab, (C)++, or (P)ython, third letter: tracker is using (G)PU or only (C)PU). A dash "-" indicates that the realtime experiment was performed using an outdated version of the toolkit and that the results are invalid.

CSRDCF++ (A.40), SiamFC (A.21), ECOhc (A.31), Staple (A.20), KFebT (A.12), ASMS (A.6), SSKCF (A.25), CSRDCFf (A.39), UCT (A.19), MOSSE_CA (A.35) and SiamDCF (A.23). All trackers except ASMS (A.6), which is scale adaptive mean shift tracker, apply discriminative correlation filters in a wide sense of the term. Among these,

	cam. mot.	ill. ch.	mot. ch.	occl.	scal. ch.
Accuracy	0.48	0.46	0.45 ③	0.39 ①	0.41 ②
Robustness	0.84	1.16 ②	0.97 ③	1.19 ①	0.69

Table 2. Tracking difficulty with respect to the following visual attributes: camera motion (cam. mot.), illumination change (ill. ch.), motion change (mot. ch.), occlusion (occl.) and size change (scal. ch.) .

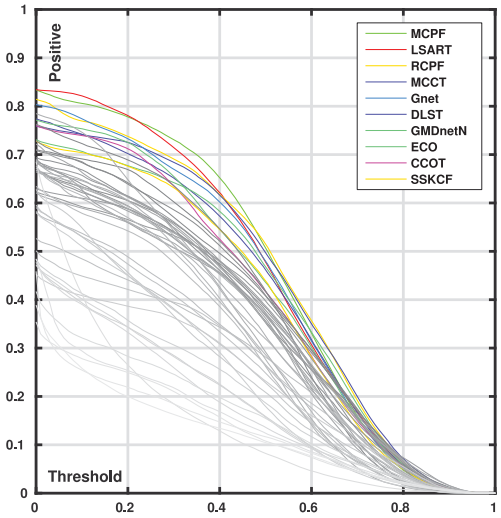

Figure 6. The OPE no-reset plots.

	Tracker	EAO	A	R
1.	CCOT	0.203 ①	0.575	0.444 ①
2.	CFCF	0.202 ②	0.587 ②	0.458 ②
3.	ECO	0.196 ③	0.586 ③	0.473
4.	Gnet	0.196	0.549	0.444 ①
5.	CFWCR	0.187	0.575	0.478
6.	LSART	0.185	0.535	0.460 ③
7.	MCCT	0.179	0.597 ①	0.532
8.	MCPF	0.165	0.543	0.596
9.	SiamDCF	0.160	0.555	0.685
10.	CSRDCF	0.150	0.522	0.631

Table 3. The top 10 trackers from Table 1 re-ranked on the VOT2017 sequestered dataset.

all but UCT (A.19) and SiamFC (A.21) apply the standard circular shift filter learning with FFT.

Most of the top trackers apply hand-crafted features except the SiamFC (A.21), SiamDCF (A.23) and UCT (A.19). The only tracker that applies a motion model is KFebT (A.12) which also combines ASMS [71], KCF [26] and NCC trackers.

The top-performer, CSRDCF++ (A.40), is a C++ implementation of the CSRDCF (A.38) tracker which runs on a single CPU. This is a correlation filter that learns a spatially constrained filter. The learning process implicitly addresses the problem of boundary effects in correlation circular shifts, learns from a wide neighborhood and makes the

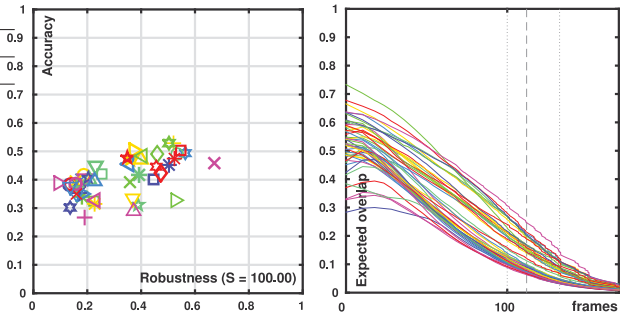

Figure 7. The AR plot (left) and the EAO curves (right) for the VOT2017 realtime experiment.

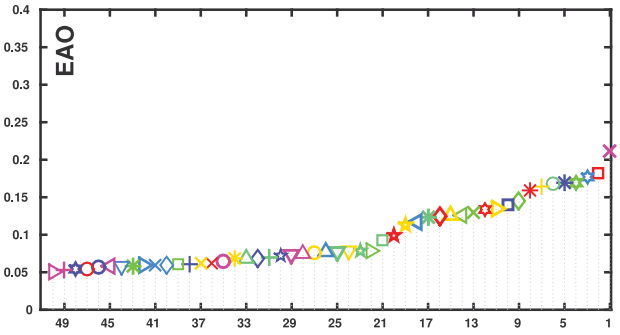

Figure 8. The EAO plot (right) for the realtime experiment.

filter robust to visual distractors. The second-best tracker, SiamFC (A.21), is conceptually similar in that it performs correlation for object localization over a number of feature channels. In contrast to CSRDCF++ (A.40), it does not apply any learning during tracking. Target is localized by directly correlating a multi-channel template extracted in the first frame with a search region. The features are trained on a large number of videos to maximize target discrimination even in presence of visual distractors. This tracker is cast as a system of convolutions (CNN) and leverages the GPU for intensive computations.

The best performing real-time trackers is CSRDCF++ (A.40), but this tracker is co-authored by the VOT organizers. According to the VOT winner rules, the winning real-time tracker of the VOT2017 is SiamFC (A.21).

5. VOT-TIR2017 analysis and results

5.1. Trackers submitted

The re-opening of the VOT-TIR2016 challenge [20] attracted 7 new submissions with binaries/source code included that allowed results verification: LTFLO (B.1), KFebT (B.2), DSLT (B.3), BST (B.4), UCT (B.5), SPCT (B.6), and MOSSE_CA (B.7, where only DSLT has not been submitted to the VOT2017 challenge. The VOT2017 committee and associates additionally con-

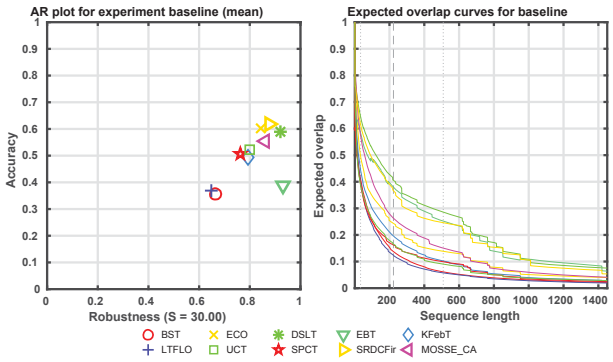

Figure 9. The AR-raw plots generated by sequence pooling (left) and EAO curves (right).

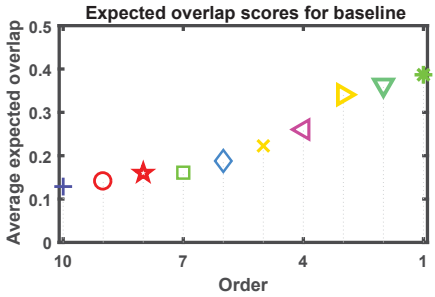

Figure 10. Expected average overlap graph with trackers ranked from right to left. The right-most tracker is the top-performing according to the VOT-TIR2017 expected average overlap values. See Figure 9 for legend.

tributed 3 baseline trackers: ECO (B.8), EBT (B.9) (winner VOT-TIR2016), and SRDCFir (B.10) (top-performer VOT-TIR2015).

Thus in total 10 trackers were compared on the VOT-TIR2017 challenge. In the following we briefly overview the entries and provide the references to original papers in the Appendix B where available.

The trackers were based on various tracking principles: two trackers (ECO (B.8) and UCT (B.5)) were based on CNN matching, 5 trackers applied discriminative correlation filters (ECO (B.8), KFebT (B.2), MOSSE_CA (B.7), UCT (B.5), and SRDCFir (B.10)), one tracker (BST (B.4)) was based on structured SVM, one tracker was based on Mean Shift (KFebT (B.2)), one tracker (DSLT (B.3)) applied optical flow, one tracker was based on line segments matching (LTFLO (B.1)), two trackers (KFebT (B.2) and SPCT (B.6)) were based on tracker combinations, and one tracker (EBT (B.9)) was based on object proposals.

5.2. Results

The results are summarized in the AR-raw plots and EAO curves in Figure 9 and the expected average overlap plots in Figure 10. The values are also reported in Table 4.

The top three trackers according to the primary EAO measure (Figure 10) are DSLT (B.3), EBT (B.9), and SRDCFir (B.10). These trackers are very diverse in the tracking approach and in contrast to the RGB-case no dominating methodology can be identified.

The top trackers in EAO are also among the most robust trackers, which means that they are able to track longer without failing. The top trackers in robustness (Figure 9) are EBT (B.9), DSLT (B.3) and SRDCFir (B.10). On the other hand, the top performers in accuracy are SRDCFir (B.10), ECO (B.8), and DSLT (B.3).

According to the EAO measure, the overall winner of the VOT-TIR2017 challenge is DSLT (B.3).

	Tracker	EAO	A	R
1.	**DSLT**	0.3990 ①	0.59 ③	0.92 ②
2.	**EBT**	0.3678 ②	0.39	0.93 ①
3.	**SRDCFir**	0.3566 ③	0.62 ①	0.88 ③
4.	**MOSSE_CA**	0.2713	0.56	0.86
5.	**ECO**	0.2363	0.60 ②	0.84
6.	**KFebT**	0.1964	0.49	0.79
7.	**UCT**	0.1694	0.52	0.80
8.	**SPCT**	0.1680	0.51	0.76
9.	**BST**	0.1482	0.36	0.66
10.	**LTFLO**	0.1356	0.37	0.65

Table 4. Numerical results of VOT-TIR2017 challenge.

6. Online resources

To facilitate advances in the field of tracking, the VOT initiative offers the code developed by the VOT committee for the challenge as well as the results from the VOT2017 page [14]. The page will be updated after publication of this paper with the following content:

1. The raw results of the baseline experiment.
2. The raw results of the real-time experiment.
3. The OTB [77] main OPE experiment.
4. Links to the source code of many trackers submitted to the challenge, already integrated with the new toolkit.
5. All results generated in the VOT-TIR2017 challenge.
6. The links to the new toolkit and toolboxes developed by the VOT committee.

7. Conclusion

Results of both the VOT2017 and VOT-TIR2017 challenges were presented. As already indicated by the last two

[14] http://www.votchallenge.net/vot2017

challenges, the popularity of discriminative correlation filters as means of target localization and CNNs as feature extractors is increasing. A large subset of trackers submitted to VOT2017 exploit these.

The top performer of the VOT2017 sequestered dataset is the CCOT (A.36), which is a continuous correlation filter utilizing standard pre-trained CNN features. The winner of the VOT2017 challenge, however, is the CFCF (A.10), which is a correlation filter that uses a standard CNN fine-tuned for correlation-based target localization.

The top performer of the VOT017 real-time challenge is CSRDCF++ (A.40), which uses a robust learning of discriminative correlation filter, applies hand-crafted features, running in real-time on a CPU. Since CSRDCF++ is co-authored by VOT organizers, the winner of the VOT2017 realtime challenge is the SiamFC (A.21), which is a fully convolutional tracker that does not apply learning during tracking and runs on GPU.

The top performer and the winner of the VOT-TIR2017 challenge is DSLT (B.3), combining very good accuracy and robustness using high-dimensional features. The approach is simple, i.e., does not adapt to scale, nor explicitly addresses object occlusion, but applies complex features: non-normalized HOG features and motion features.

The VOT aims to be platform for discussion of tracking performance evaluation and it contributes to the tracking community with verified annotated datasets, performance measures and evaluation toolkits. The VOT2017 was a fifth effort toward this, following the very successful VOT2013, VOT2014, VOT2015 and VOT2016.

Acknowledgements

This work was supported in part by the following research programs and projects: Slovenian research agency research programs P2-0214, P2-0094, Slovenian research agency project J2-8175. Jiři Matas and Tomáš Vojíř were supported by the Czech Science Foundation Project GACR P103/12/G084. Michael Felsberg and Gustav Häger were supported by WASP, VR (EMC2), SSF (SymbiCloud), and SNIC. Gustavo Fernández and Roman Pflugfelder were supported by the AIT Strategic Research Programme 2017 Visual Surveillance and Insight. The challenge was sponsored by Faculty of Computer Science, University of Ljubljana, Slovenia.

A. Submitted trackers VOT2017 challenge

In this appendix we provide a short summary of all trackers that were considered in the VOT2017 challenge.

A.1. ANT (ANT)

L. Čehovin Zajc
luka.cehovin@fri.uni-lj.si

The ANT tracker is a conceptual increment to the idea of multi-layer appearance representation that is first described in [66]. 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 [69] for details.

A.2. Convolutional regression for visual tracking (CRT)

K. Chen, W. Tao
chkap@hust.edu.cn, wenbingtao@hust.edu.cn

CRT learns a linear regression model by training a single convolution layer via gradient descent. The samples for training and predicting are densely clipped by setting the kernel size of the convolution layer to the size of the object patch. A novel objective function is also proposed to improve the running speed and accuracy. For more detailed information on this tracker, please see [11].

A.3. Constrained Graph Seeking based Tracker (CGS)

D. Du, Q. Huang, S. Lyu, W. Li, L. Wen, X. Bian
dawei.du@vipl.ict.ac.cn, {slyu, wli20}@albany.edu,
qmhuang@ict.ac.cn, {longyin.wen, xiao.bian}@ge.com

CGS is a new object tracking method based on constrained graph seeking, which integrates target part selection, part matching, and state estimation using a unified energy minimization framework to address two major drawbacks: (1) inaccurate part selection which leads to performance deterioration of part matching and state estimation and; (2) insufficient effective global constraints for local part selection and matching. CGS tracker also incorporates structural information in local part variations using the global constraint. To minimize the energy function, an alternative iteration scheme is used.

A.4. Multi-Cue Correlation Tracker (MCCT)

N. Wang, W. Zhou, H. Li
wn6149@mail.ustc.edu.cn, {zhwg, lihq}@ustc.edu.cn

The multi-cue correlation tracker (MCCT) is based on the discriminative correlation filter framework. By combining different types of features, our approach constructs multiple experts and each of them tracks the target independently. With the proposed robustness evaluation strategy, the suitable expert is selected for tracking in each frame. Furthermore, the divergence of multiple experts reveals the reliability of the current tracking, which helps update the experts adaptively to keep them from corruption.

A.5. Long Term FeatureLess Object tracker (LTFLO)

K. Lebeda, S. Hadfield, J. Matas, R. Bowden
karel@lebeda.sk, matas@cmp.felk.cvut.cz,
{s.hadfield,r.bowden}@surrey.ac.uk

LTFLO is based on and extends our previous work on tracking of texture-less objects [37, 36]. It decreases reliance on texture by using edge-points instead of point features. The use of edges is burdened by the *aperture problem*, where the movement of the edge-point is measurable only in the direction perpendicular to the edge. We overcome this by using correspondences of lines tangent to the edges, instead of using the point-to-point correspondences. Assuming the edge is locally linear, a slightly shifted edge-point generates the same tangent line as the true correspondence. RANSAC, then provides an estimate of the frame-to-frame transformation (similarity is employed in the experiments, but higher order transformations could be employed as well).

A.6. Scale Adaptive Mean-Shift Tracker (ASMS)

T. Vojíř, J. Noskova and J. Matas
vojirtom@cmp.felk.cvut.cz, noskova@mat.fsv.cvut.cz,
matas@cmp.felk.cvut.cz

The mean-shift tracker optimize the Hellinger distance between template histogram and target candidate in the image. This optimization is done by a gradient descend. ASMS [73] 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.7. Flock of Trackers (FoT)

T. Vojíř, J. Matas
{vojirtom, matas}@cmp.felk.cvut.cz

The Flock of Trackers (FoT) is a tracking framework where the object motion is estimated from the displacements or, more generally, transformation estimates of a number of local trackers covering the object. Each local tracker is attached to a certain area specified in the object coordinate frame. The local trackers are not robust and assume that the tracked area is visible in all images and that it undergoes a simple motion, e.g. translation. The FoT object motion estimate is robust if it is from local tracker motions by a combination which is insensitive to failures.

A.8. Kernelized Correlation Filter (KCF)

T. Vojíř
vojirtom@cmp.felk.cvut.cz

This tracker is a C++ implementation of Kernelized Correlation Filter [26] 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.9. Guided MDNet-N (GMDNetN)

P. Venugopal, D. Mishra, G. R K S. Subrahmanyam
pallavivm91@gmail.com,
{deepak.mishra, gorthisubrahmanyam}@iist.ac.in

The tracker Guided MDNet-N improves the existing tracker MDNetN [48] in terms of its computational efficiency and time without much compromise on the trackers performance. MDNet-N is a convolutional neural network tracker which initializes its network using the ImageNet [18]. This network is now directly taken for tracking where it takes 256 random samples around the previous target and selects the best possible sample out of it as the target. Guided MDNet-N chooses lesser number of guided samples by two of the efficient methods called as frame level detection of TLD [27] and non-linear regression model of KCF [26]. The speed of the Guided MDNet-N improves due to the lesser number of efficient guided samples chosen. All implementations and comparisons were done on the CPU.

A.10. Convolutional Features for Correlation Filters (CFCF)

E. Gundogdu, A. A. Alatan
egundogdu87@gmail.com, alatan@metu.edu.tr

The tracker 'CFCF' is based on the feature learning study in [23] and the correlation filter based tracker in [17]. The proposed tracker employs a fully convolutional neural network (CNN) model trained on ILSVRC15 [56] video dataset by the introduced learning framework in [23]. This framework is designed for correlation filter formulation in [13]. To learn features, convolutional layers of VGG-M-2048 network [10], which is trained on [18], with an extra convolutional layer is fine-tuned on ILSVRC15 dataset. The first, fifth and sixth convolutional layers of the learned network, HOG [47] and Colour Names (CN) [65] are integrated to the tracker of [17]. The reader is referred to [23] for details.

A.11. Online Adaptive Hidden Markov Model for Multi-Tracker Fusion (HMMTxD)

T. Vojíř, J. Noskova and J. Matas
vojirtom@cmp.felk.cvut.cz, noskova@mat.fsv.cvut.cz, matas@cmp.felk.cvut.cz

The HMMTxD method fuses observations from complementary out-of-the box trackers and a detector by utilizing a hidden Markov model whose latent states correspond to a binary vector expressing the failure of individual trackers. The Markov model is trained in an unsupervised way, relying on an online learned detector to provide a source of tracker-independent information for a modified Baum-Welch algorithm that updates the model w.r.t. the partially annotated data.

A.12. KFebT

P. Senna, I. Drummond, G. Bastos
{pedrosennapsc, isadrummond, sousa}@unifei.edu.br

The tracker KFebT [57] fuses the result of three out-of-the box trackers: a mean-shift tracker that uses colour histogram (ASMS) [73], a kernelized correlation filter (KCF) [26] and the Normalized Cross-Correlation (NCC) [8] by using a Kalman filter. The tracker works in prediction and correction cycles. First, a simple motion model predicts the target next position, then, the trackers results are fused with the predicted position and the motion model is updated in the correction process. The fused result is the KFebT output which is used as last position of the tracker in the next frame. To measure the reliability of the Kalman filter, the tracker uses the result confidence and the motion penalization which is proportional to the distance between the tracker result and the predicted result. As confidence measure, the Bhattacharyya coefficient between the model and the target histogram is used in case of ASMS tracker, while the correlation result is applied in case of KCF tracker and NCC tracker. The source code is public available in https://github.com/psenna/KF-EBT.

A.13. Scale Adaptive Point-based Kanade Lukas Tomasi colour-Filter (SAPKLTF)

E. Velasco-Salido, J. M. Martínez, R. Martín-Nieto, Á. García-Martín
{erik.velasco, josem.martinez, rafael.martinn, alvaro.garcia}@uam.es

The SAPKLTF [70] tracker is based on an extension of PKLTF tracker [21] with ASMS [73]. SAPKLTF is a single-object long-term tracker which consists of two phases: The first stage is based on the Kanade Lukas Tomasi approach (KLT) [58] choosing the object features (colour and motion coherence) to track relatively large object displacements. The second stage is based on scale adaptive mean shift gradient descent [73] to place the bounding box into the exact position of the object. The object model consists of a histogram including the quantized values of the RGB colour components, and an edge binary flag.

A.14. CFWCR

Z. He, Y. Fan, J. Zhuang
{he010103, evelyn}@bupt.edu.cn, junfei.zhuang@faceall.cn

CFWCR adopts Efficient Convolution Operators [12] tracker as the baseline approach. A continuous convolution operator based tracker is derived which fully exploits the discriminative power in the CNN feature representations. First, each individual feature extracted from different layers of the deep pre-trained CNN is normalised, and after that, the weighted convolution responses from each feature block are summed to produce the final confidence score. It is also found that the 10-layers design is optimal for continuous scale estimation. The empirical evaluations demonstrate clear improvements by the proposed tracker based on the Efficient Convolution Operators Tracker (ECO) [12].

A.15. Deep Location-Specific Tracking (DLST)

L. Yang, R. Liu, D. Zhang, L. Zhang

A Deep Location-Specific Tracking (DLST) framework is proposed based on deep Convolutional Neural Neworks (CNNs). The DLST decomposes the tracking into localization and classification, and trains an individual network for each task online. The localization network exploits the information in the current frame and provides another specific location to improve the probability of successful tracking. The classification network finds the target among many examples drawn around the target location in the previous frame and the location estimated in the current frame. The bounding box regression and online hard negative mining [48] technologies are also adopted in the proposed DLST framework.

A.16. gNetTracker (gnet)

Siddharta Singh, D. Mishra
siddharthaiist@gmail.com, deepak.mishra@iist.ac.in

The tracker gnet integrates GoogLeNet features with the spatially regularized model (SRDCF) and ECO model. In both cases, it was observed that tracking accuracy increased. The spatially regularized model on different combination of layers is evaluated. The results of these evaluations on VOT 2016 dataset indicated that features extracted from inception module 4d and 4e are most suitable for the purpose of object tracking. This finding is in direct contrast to the finding of previous studies done on VGGNet [14, 44] which recommended the use of shallower layers for tracking based on the argument that shallower layers have more resolution and hence can be used for object localization. It was found that a combination of shallow layers (like inception modules

4c and 4b) with deeper layers result in slight improvement in the performance of tracker but also leads to significant increase in computational cost.

A.17. Best Structured Tracker (BST)

F. Battistone, A. Petrosino, V. Santopietro
francesco.battistone, petrosino,
vincenzo.santopietro@uniparthenope.it

BST is based on the idea of Flock of Trackers [71]: a set of local trackers tracks a little patch of the original target and then the tracker combines their information in order to estimate the resulting bounding box. Each local tracker separately analyzes the Haar features extracted from a set of samples and then classifies them using a structured Support Vector Machine as Struck [24]. Once having predicted local target candidates, an outlier detection process is computed by analyzing the displacements of local trackers. Trackers that have been labeled as outliers are reinitialized. At the end of this process, the new bounding box is calculated using the Convex Hull technique.

A.18. Multi-task Correlation Particle Filter (MCPF)

T. Zhang, J. Gao, C. Xu
{tzzhang, csxu}@nlpr.ia.ac.cn, gaojunyu2015@ia.ac.cn

MCPF learns a multi-task correlation particle filter for robust visual tracking. The proposed MCPF is designed to exploit and complement the strength of a multi-task correlation filter (MCF) and a particle filter. First, it can shepherd the sampled particles toward the modes of the target state distribution via the MCF, thereby resulting in robust tracking performance. Second, it can effectively handle large-scale variation via a particle sampling strategy. Third, it can effectively maintain multiple modes in the posterior density using fewer particles than conventional particle filters, thereby lowering the computational cost. The reader is referred to [83] for details.

A.19. UCT

Z. Zhu, G. Huang, W. Zou, D. Du, C. Huang
{zhuzheng2014, wei.zou}@ia.ac.cn,
{guan.huang, dalong.du, chang.huang}@hobot.cc

UCT uses a fully convolutional network to learn the convolutional features and to perform the tracking process simultaneously, namely, a unified convolutional tracker (UCT). UCT treats both processes feature extraction and tracking as a convolution operation and trains them jointly, enabling learned CNN features are tightly coupled to tracking process. In online tracking, an efficient updating method is proposed by introducing peak-versus-noise ratio (PNR) criterion, and scale changes are handled by incorporating a scale branch into network.

A.20. Staple

Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, Philip Torr
{name.surname}@eng.ox.ac.uk

Staple is a tracker that combines two image patch representations that are sensitive to complementary factors to learn a model that is inherently robust to both colour changes and deformations. To maintain real-time speed, two independent ridge regression problems are solved, exploiting the inherent structure of each representation. Staple combines the scores of two models in a dense translation search, enabling greater accuracy. A critical property of the two models is that their scores are similar in magnitude and indicative of their reliability, so that the prediction is dominated by the more confident. For more details, we refer the reader to [4].

A.21. Fully-Convolutional Siamese Network (SiamFC)

Luca Bertinetto, João Henriques, Jack Valmadre, Andrea Vedaldi, Philip Torr
{name.surname}@eng.ox.ac.uk

SiamFC [5] applies a fully-convolutional Siamese network trained to locate an exemplar image within a larger search image. The network 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 conv-net is trained offline on the ILSVRC VID dataset [56] to address a general similarity learning problem. This similarity function is then used within a simplistic tracking algorithm. The architecture of the conv-net resembles 'AlexNet' [34]. This version of SiamFC incorporates some minor improvements and is available as the baseline model of the CFNet paper [64].

A.22. Spatial Pyramid Context-Aware Tracker (SPCT)

M. Poostchi, K. Palaniappan, G. Seetharaman, K. Gao
mpoostchi@mail.missouri.edu, pal@missouri.edu,
guna@ieee.org, kg954@missouri.edu

SPCT is a collaborative tracker that combines complementary cues in an intelligent fusion framework to address the challenges of persistent tracking in full motion video. SPCT relies on object visual features and temporal motion information [53]. The visual feature-based tracker usually takes the lead as long as object is visible and presents discriminative visual features, otherwise the tracker is assisted by motion information. A set of pre-selected complementary features is chosen including RGB color, intensity and spatial pyramid of HoG to encode object color, shape and spatial layout information [51]. SPCT utilizes image spatial

context at different level to make the video tracking system resistant to occlusion, background noise and improve target localization accuracy and exploits integral histogram as building block to meet the demands of real-time processing [52]. The estimated motion detection mask is fused with feature-based likelihood maps to filter out false background motion detections. Kalman motion prediction is used to detect the candidate region in the next frame and improve target localization when being partially or fully occluded.

A.23. SiamDCF (SiamDCF)

Q. Wang, J. Gao, J. Xing, M. Zhang, Z. Zhang, W. Hu
{*qiang.wang, jin.gao, jlxing, mengdan.zhang, zhpzhang, wmhu*}*@nlpr.ia.ac.cn*

SiamDCF is an end-to-end multitask learning based tracker, which learns two tasks simultaneously with the aim of mutual benefit. The low-level features are exploited using the task which bears a resemblance to the DCFNet [75] for precisely tracking; the high-level features are captured using the task inspired by SiameseFC [5] for robust tracking.

A.24. Learning Spatial-Aware Regressions for Visual Tracking (LSART)

C. Sun, J. Liu, H. Lu, M. Yang
waynecool@mail.dlut.edu.cn, jyliu0329@gmail.com, lhchuan@dlut.edu.cn, mhyang@ucmerced.edu

The LSART tracker exploits the complementary kernelized ridge regression (KRR) and convolution neural network (CNN) for tracking. A weighted cross-patch similarity kernel for the KRR model is defined and the spatially regularized filter kernels for the CNN model is used. While the former focuses on the holistic target, the latter focuses on the small local regions. The distance transform is exploited to pool layers for the CNN model, which determines the reliability of each output channel. Three kinds of features are used in the proposed method, which are respectively Conv4-3 of VGG-16, Hog, and Colour naming. The LSART tracker is based on [62] with some minor revisions (e.g., more features are used).

A.25. SumShift Tracker with Kernelized Correlation Filter (SSKCF)

J. Lee, S. Choi, J. Jeong, J. Kim, J. Cho
{*jylee, sunglok, channij80, giraffe, jicho*}*@etri.re.kr*

SumShiftKCF tracker is an extension of the SumShift tracker [38] by the kernelized correlation filter tracker (KCF) [26]. The SumShiftKCF tracker computes the object likelihood with the weighted sum of the histogram back-projection weights and the correlation response of KCF. The target is then located by the Sum-Shift iteration [38].

A.26. Adaptive single object Tracking using offline Learned motion And visual Similar patterns (ATLAS)

B. Mocanu, R. Tapu, T. Zaharia
{*bogdan.mocanu, ruxandra.tapu, titus.zaharia*}*@telecom-sudparis.eu*

ATLAS is a generic object tracker based on two convolutional neural networks trained offline. The key principle consists of alternating between tracking using motion information and predicting the object location in time based on visual similarity. As for GOTURN [25], ATLAS uses a regression-based approach to learn offline generic relationships between the object appearances and its associated motion patterns. Then, by using the DeepCompare [80] the system adaptively modifies the object bounding box position and shape. Starting from the initial candidate location, the object position is successively shifted within the context search area based on a patch similarity function which is learnt from annotated pairs of raw images. The final track is the one that correspond to the instance that provides the maximal value of similarity.

A.27. Correlation-based Visual Tracking via Dynamic Part Regressors Fusion (DPRF)

A. Memarmoghadam, P. Moallem
{*a.memarmoghadam, p_moallem*}*@eng.ui.ac.ir*

We propose an impressive part-wise tracking framework namely as DPRF for evolving single-patch correlation filter based trackers (CFTs) by simultaneously collaborating of both global and local CF-based part regressors in object modeling. Moreover, to intelligently follow target appearance changes, we dynamically assign importance weights to each parts model via solving a multi-linear ridge regression optimization problem towards achieving the most discriminative aggregated confidence map. In this respect, we rely on the most representative parts with higher weights and hence successfully handle heavy occlusions as well as other locally drastic appearance changes. Additionally, to further alleviating tracking drift during model update, we present a simple yet effective scale estimation technique based on relative importance movement of pair-wise inlier parts. We believe that our proposed DPRF tracker provides powerful framework for promoting single-patch CFTs known in the literature. Without loss of generality, here, we apply ordinary single-patch multi-channel KCF tracker [26] as the baseline approach for each part which expeditiously tracks the target object parts.

A.28. ColorHough Tracker (CHT)

A. Tran, A. Manzanera
{*antoine.tran, antoine.manzanera*}*@ensta-paristech.fr*

ColorHough Tracker [63] is a real-time object tracker relying on two complementary low-level features: colour

and gradient. While colour histogram is used as a global rotation-invariant model to separate object from the background, gradient orientation is used as an illumination-invariant index for a Generalised Hough Transform in order to provide a localisation of the target. These two parts are then merged, to estimate the object position. Model updating is done by computing independently two pixel-level confidence maps and by merging them. The original ColorHough Tracker [63] is improved by tuning the set of parameters.

A.29. GOTURN MDNet Tracker (GMD)

Y. Xing, K. M. Kitani
yxing1@andrew.cmu.edu, kkitani@cs.cmu.edu

GMD (GOTURN MDNet Tracker) is a Siamese Convolutional Neural Network based tracker. It combines the characteristics of high-speed tracker GOTURN [25] and mechanisms from MDNet [48] for a timely feedback on model update. GMD focuses on a classification based approach that achieves timely appearance model adaptation through online learning. By feeding the Siamese network the paired information of frame t-1 target and a set of candidates sampled from frame t, the network learns an observation model that assigns score to each of the candidates. The score measurement in turn provides signal for timely update of the network. ROI pooling over the candidates is used to speed up the feed-forward process. GMD is trained using ImageNet [56] video dataset that has high object appearance variations.

A.30. Efficient Convolution Operator Tracker (ECO)

M. Danelljan, G. Bhat, F. Shahbaz Khan, M. Felsberg
{martin.danelljan, goutam.bhat, fahad.khan, michael.felsberg}@liu.se

ECO addresses the problems of computational complexity and over-fitting in state of the art DCF trackers by introducing: (i) a factorized convolution operator, which drastically reduces the number of parameters in the model; (ii) a compact generative model of the training sample distribution, that significantly reduces memory and time complexity, while providing better diversity of samples; (iii) a conservative model update strategy with improved robustness and reduced complexity. The reader is referred to [12] for more details.

A.31. Efficient Convolution Operator Tracker - Hand Crafted (ECOhc)

M. Danelljan, G. Bhat, F. Shahbaz Khan, M. Felsberg
{martin.danelljan, goutam.bhat, fahad.khan, michael.felsberg}@liu.se

ECOhc is a faster version of the ECO tracker [12] using hand crafted features (Histogram of Oriented Gradients (HOG) [47] and Colour Names (CN) [65]).

A.32. Large Displacement Estimation of Similarity transformation on Visual Object Tracking (LDES)

Y. Li, J. Zhu
{liyang89, jkzhu}@zju.edu.cn

The tracker LDES estimates the object scale changes and it reduces the number of scale sampling to accelerate the speed of the current correlation filter. Search and training ranges are decomposed into two different areas to enable large displacement estimation. In addition, only HoG and colour information are employed to enhance the robustness of the tracker for the purpose of efficiency.

A.33. FSTC

B. Chen, L. Wang, H. Lu
{bychen, wlj}@mail.dlut.edu.cn, lhchuan@dlut.edu.cn

The proposed FSTC tracker draws inspiration from [74] by combining mid and high-level features from a deep network [54] pre-trained on the PASCAL-VOC 2007 data set with 20 categories to be detected. The algorithm leverages multi-level features from both lower and higher layers of a pre-trained Deep Neural Network (DNN). Considering that deep features of different levels may be suitable for different scenarios, a decision network is trained in the off-line stage to facilitate feature selection in on-line tracking. To better exploit the temporal consistency assumption of visual tracking, the decision network is implemented with long short term memory (LSTM) units, which are capable of capturing the historical context information to perform more reliable inference at the current time step. To further improve tracking accuracy, a promoting strategy for trackers with detection results of a generic object detector is proposed, reducing the risk of tracking drifts.

A.34. Robust Correlation Particle Filter (RCPF)

T. Zhang, J. Gao, C. Xu
{tzzhang, csxu}@nlpr.ia.ac.cn, gaojunyu2015@ia.ac.cn

The robust correlation particle filter (RCPF) is based on correlation filters and particle filters [82]. The tracker has the advantages of particle filters and correlation filters for scale variation and partial occlusion handling. The tracking robustness is improved with multiple different features (deep and HOG), an effective scale adaptive scheme and a long-short term model update scheme.

A.35. MOSSE_CA

M. Mueller
matthias.mueller.2@kaust.edu.sa

This tracker builds upon the very simple and fast correlation filter tracker MOSSE [6]. MOSSE_CA only uses gray-scale pixel values as features and does not adapt the scale of

the target. The key difference to the original tracker is that MOSSE_CA explicitly incorporates global context according a recent framework for CF trackers [46]. The tracking performance is certainly not state-of-the-art, but shows that even a very simple correlation filter tracker can achieve reasonable performance when incorporating context information.

A.36. Continuous Convolution Operator Tracker (C-COT)

M. Danelljan, A. Robinson, F. Shahbaz Khan, M. Felsberg
{*martin.danelljan, andreas.robinson, fahad.khan, michael.felsberg*}*@liu.se*

C-COT learns a discriminative continuous convolution operator as its tracking model. C-COT poses the learning problem in the continuous spatial domain. This enables a natural and efficient fusion of multi-resolution feature maps, e.g. when using several convolutional layers from a pre-trained CNN. The continuous formulation also enables highly accurate localization by sub-pixel refinement. The reader is referred to [17] for details.

A.37. Consensus Based Matching and Tracking (CMT)

Submitted by VOT Committee

The CMT tracker is a keypoint-based method in a combined matching-and-tracking framework. To localise the object in every frame, each key point casts votes for the object center. A consensus-based scheme is applied for outlier detection in the voting behaviour. By transforming votes based on the current key point constellation, changes of the object in scale and rotation are considered. The use of fast keypoint detectors and binary descriptors allows the current implementation to run in real-time. The reader is referred to [49] for details.

A.38. Discriminative Correlation Filter with Channel and Spatial Reliability (CSRDCF)

A. Lukežič, T. Vojíř, L. Čehovin, J. Matas, M. Kristan
{*alan.lukezic, luka.cehovin, matej.kristan*}*@fri.uni-lj.si,*
{*vojirtom, matas*}*@cmp.felk.cvut.cz*

The CSR-DCF [43] 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.

A.39. Discriminative Correlation Filter with Channel and Spatial Reliability - fast (CSRDCFf)

A. Lukežič, T. Vojíř, L. Čehovin, J. Matas, M. Kristan
{*alan.lukezic, luka.cehovin, matej.kristan*}*@fri.uni-lj.si,*
{*vojirtom, matas*}*@cmp.felk.cvut.cz*

The faster implementation of the Matlab tracker CSR-DCF [43]. The main performance improvements include optimized image resizing, scale interpolation, histogram extraction and backprojection.

A.40. Discriminative Correlation Filter with Channel and Spatial Reliability - C++ (CSRDCF++)

A. Muhic, A. Lukežič, T. Vojíř, L. Čehovin, J. Matas, M. Kristan
am4738@student.uni-lj.si,
{*alan.lukezic, luka.cehovin, matej.kristan*}*@fri.uni-lj.si,*
{*vojirtom, matas*}*@cmp.felk.cvut.cz*

The c++ implementation of the CSR-DCF [43] Matlab tracker. For the referenced description refer to A.38.

A.41. Deformable part correlation filter tracker (DPT)

A. Lukežič, L. Čehovin, M. Kristan
{*alan.lukezic, luka.cehovin, matej.kristan*}*@fri.uni-lj.si*

DPT is a part-based correlation filter composed of a coarse and mid-level target representations. Coarse representation is responsible for approximate target localization and uses HOG as well as colour features. The mid-level representation is a deformable parts correlation filter with fully-connected parts topology and applies a novel formulation that threats geometric and visual properties within a single convex optimization function. The mid level as well as coarse level representations are based on the kernelized correlation filter from [26]. The reader is referred to [42] for details.

A.42. Discriminative Scale Space Tracker (DSST)

Submitted by VOT Committee

The Discriminative Scale Space Tracker (DSST) [13] extends the Minimum Output Sum of Squared Errors (MOSSE) tracker [7] with robust scale estimation. The DSST additionally learns a one-dimensional discriminative scale filter, that is used to estimate the target size. For the translation filter, the intensity features employed in the MOSSE tracker is combined with a pixel-dense representation of HOG-features.

A.43. Robust Fragments based Tracking using the Integral Histogram - FragTrack (FT)

Submitted by VOT Committee

FragTrack represents the model of the object by multiple image fragments or patches. The patches are arbitrary

and are not based on an object model. Every patch votes on the possible positions and scales of the object in the current frame, by comparing its histogram with the corresponding image patch histogram. A robust statistic is minimized in order to combine the vote maps of the multiple patches. The algorithm overcomes several difficulties which cannot be handled by traditional histogram-based algorithms like partial occlusions or pose change.

A.44. Incremental Learning for Robust Visual Tracking (IVT)

Submitted by VOT Committee

The idea of the IVT tracker [55] is to incrementally learn a low-dimensional sub-space representation, adapting online to changes in the appearance of the target. The model update, based on incremental algorithms for principal component analysis, includes two features: a method for correctly updating the sample mean, and a forgetting factor to ensure less modelling power is expended fitting older observations.

A.45. L1APG

Submitted by VOT Committee

L1APG [2] considers tracking as a sparse approximation problem in a particle filter framework. To find the target in a new 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 an ℓ_1 regularized least squares problem. The Bayesian state inference framework is used to propagate sample distributions over time.

A.46. 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 [66] for details.

A.47. MEEM

Submitted by VOT Committee

MEEM [81] uses an online SVM with a redetection based on the entropy of the score function. The tracker creates an ensamble of experts by storing historical snapshots while tracking. When needed the tracker can be restored by the best of these experts, selected using an entropy minimization criterion.

A.48. Multiple Instance Learning tracker (MIL)

Submitted by VOT Committee

MIL tracker [1] uses a tracking-by-detection approach, more specifically Multiple Instance Learning instead of traditional supervised learning methods and shows improved robustness to inaccuracies of the tracker and to incorrectly labelled training samples.

A.49. MSSA

V.Zavrtanik, A. Lukežič
alan.lukezic@fri.uni-lj.si, vz1528@student.uni-lj.si

The MSSA tracker uses the mean-shift method for localization. It uses background weighted color histograms as a region similarity meassure in the mean-shift procedure. Additionally it utilizes the DSST [13] scale estimation technique.

A.50. Spatially Regularized Discriminative Correlation Filter Tracker (SRDCF)

Submitted by VOT Committee

Standard Discriminative Correlation Filter (DCF) based trackers such as [13, 16, 26] suffer from the inherent periodic assumption when using circular correlation. The Spatially Regularized DCF (SRDCF) alleviates this problem by introducing a spatial regularization function that penalizes filter coefficients residing outside the target region. This allows the size of the training and detection samples to be increased without affecting the effective filter size. By selecting the spatial regularization function to have a sparse Discrete Fourier Spectrum, the filter is efficiently optimized directly in the Fourier domain. For more details, the reader is referred to [15].

A.51. STRUCK (Struck2011)

Submitted by VOT Committee

Struck [24] is a framework for adaptive visual object tracking based on structured output prediction. The method uses a kernelized structured output support vector machine (SVM), which is learned online to provide adaptive tracking.

B. Submitted trackers VOT-TIR2017 challenge

In this appendix we provide a short summary of all trackers that were considered in the VOT-TIR2017 challenge.

B.1. Long Term FeatureLess Object tracker (LTFLO)

K. Lebeda, S. Hadfield, J. Matas, R. Bowden
karel@lebeda.sk, matas@cmp.felk.cvut.cz,
{s.hadfield,r.bowden}@surrey.ac.uk

For a tracker description, the reader is referred to A.5.

B.2. KFebT

P. Senna, I. Drummond, G. Bastos
{*pedrosennapsc, isadrummond, sousa*}*@unifei.edu.br*

For a tracker description, the reader is referred to A.12.

B.3. Dense Structural Learning based Tracker (DSLT)

X. Yu, Q. Yu, H. Zhang, N. Xie
yuxianguo_chn@163.com

DSLT extends Struck with the ability to learn from dense samples and high dimensional features. DSLT runs in a simple tracking by detection mode, with no scale adaptation and occlusion handling. Compared to our initial work [79], the feature representation in DSLT consists of 28D HOG features computed without block normalization as well as a 1D motion feature which is simply the absolute difference between consecutive frames. The search range is also set larger to account for faster motion and to get more training samples.

B.4. Best Structured Tracker (BST)

F. Battistone, A. Petrosino, V. Santopietro
francesco.battistone, petrosino,
vincenzo.santopietro@uniparthenope.it

For a tracker description, the reader is referred to A.17.

B.5. UCT

Z. Zhu, G. Huang, W. Zou, D. Du, C. Huang
{*zhuzheng2014, wei.zou*}*@ia.ac.cn,*
{*guan.huang, dalong.du, chang.huang*}*@hobot.cc*

For a tracker description, the reader is referred to A.19.

B.6. Spatial Pyramid Context-Aware Tracker (SPCT)

M. Poostchi, K. Palaniappan, G. Seetharaman, K. Gao
mpoostchi@mail.missouri.edu, pal@missouri.edu,
guna@ieee.org, kg954@missouri.edu

For a tracker description, the reader is referred to A.22.

B.7. MOSSE_CA

M. Mueller
matthias.mueller.2@kaust.edu.sa

For a tracker description, the reader is referred to A.35.

B.8. Efficient Convolution Operator Tracker (ECO)

Submitted by VOT Committee

For a tracker description, the reader is referred to A.30.

B.9. Edge Box Tracker (EBT)

Submitted by VOT Committee

For a tracker description, the reader is referred to the description by G. Zhu, F. Porikli, and H. Li in Section A.2 of the VOT-TIR2016 paper [20].

B.10. Spatially Regularized Discriminative Correlation Filter Tracker IR (SRDCFir)

Submitted by VOT Committee

For a tracker description, the reader is referred to A.50.

References

[1] B. Babenko, M. H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(8):1619–1632, 2011.

[2] C. Bao, Y. Wu, H. Ling, and H. Ji. Real time robust l1 tracker using accelerated proximal gradient approach. In *CVPR*, 2012.

[3] A. Berg, J. Ahlberg, and M. Felsberg. A Thermal Object Tracking Benchmark. In *12th IEEE International Conference on Advanced Video- and Signal-based Surveillance, Karlsruhe, Germany, August 25-28 2015*. IEEE, 2015.

[4] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr. Staple: Complementary learners for real-time tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1401–1409, 2016.

[5] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr. Fully-convolutional siamese networks for object tracking. In *ECCV 2016 Workshops*, pages 850–865, 2016.

[6] D. S. Bolme, J. R. Beveridge, B. Draper, Y. M. Lui, et al. Visual object tracking using adaptive correlation filters. In *CVPR*, 2010.

[7] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[8] K. Briechle and U. D. Hanebeck. Template matching using fast normalized cross correlation. In *Aerospace/Defense Sensing, Simulation, and Controls*, pages 95–102. International Society for Optics and Photonics, 2001.

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

[10] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014.

[11] K. Chen and W. Tao. Convolutional regression for visual tracking. *CoRR*, abs/1611.04215, 2016.

[12] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg. ECO: efficient convolution operators for tracking. In *CVPR*, 2017.

[13] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *Proceedings of the British Machine Vision Conference BMVC*, 2014.

[14] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg. Convolutional features for correlation filter based visual tracking. *ICCV Workshop*, 2015.

[15] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg. Learning spatially regularized correlation filters for visual tracking. In *International Conference on Computer Vision*, 2015.

[16] M. Danelljan, F. S. Khan, M. Felsberg, and J. Van de Weijer. Adaptive color attributes for real-time visual tracking. In *Computer Vision and Pattern Recognition*, 2014.

[17] M. Danelljan, A. Robinson, F. Shahbaz Khan, and M. Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *ECCV*, pages 472–488, 2016.

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: a large-scale hierarchical image database. In *CVPR*, 2009.

[19] M. Felsberg, A. Berg, G. Häger, J. Ahlberg, M. Kristan, A. Leonardis, J. Matas, G. Fernandez, L. Cehovin, and et al. The thermal infrared visual object tracking VOT-TIR2015 challenge results. In *ICCV workshop on VOT2015 Visual Object Tracking Challenge*, 2015.

[20] M. Felsberg, M. Kristan, J. Matas, A. Leonardis, R. Pflugfelder, G. Häger, A. Berg, A. Eldesokey, J. Ahlberg, L. Cehovin, T. Vojir, A. Lukezic, G. Fernandez, and et al. The Thermal Infrared Visual Object Tracking VOT-TIR2016 Challenge Results. In *COMPUTER VISION - ECCV 2016 WORKSHOPS, PT II*, volume 9914 of *Lecture Notes in Computer Science*, pages 824–849. SPRINGER INT PUBLISHING AG, 2016.

[21] A. González, R. Martín-Nieto, J. Bescós, and J. M. Martínez. Single object long-term tracker for smart control of a ptz camera. In *Proceedings of the International Conference on Distributed Smart Cameras*, page 39. ACM, 2014.

[22] N. Goyette, P. M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar. Changedetection.net: A new change detection benchmark dataset. In *CVPR Workshops*, pages 1–8. IEEE, 2012.

[23] E. Gundogdu and A. A. Alatan. Good features to correlate for visual tracking. *CoRR*, abs/1704.06326, 2017.

[24] S. Hare, A. Saffari, and P. H. S. Torr. Struck: Structured output tracking with kernels. In D. N. Metaxas, L. Quan, A. Sanfeliu, and L. J. V. Gool, editors, *International Conference on Computer Vision*, pages 263–270. IEEE, 2011.

[25] D. Held, S. Thrun, and S. Savarese. Learning to track at 100 fps with deep regression networks. In *European Conference Computer Vision (ECCV)*, pages 749–765, 2016.

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

[27] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7):1409–1422, 2012.

[28] R. Kasturi, D. B. Goldgof, P. Soundararajan, V. Manohar, J. S. Garofolo, R. Bowers, M. Boonstra, V. N. Korzhova, and J. Zhang. Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(2):319–336, 2009.

[29] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Čehovin, T. Vojir, 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.

[30] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Čehovin, G. Fernández, T. Vojir, 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.

[31] M. Kristan, J. Matas, A. Leonardis, T. Vojíř, R. Pflugfelder, G. Fernández, G. Nebehay, F. Porikli, and L. Čehovin. A novel performance evaluation methodology for single-target trackers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2137–2155, 2016.

[32] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, F. Porikli, L. Čehovin, G. Nebehay, F. G., 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.

[33] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, L. Čehovin, G. 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.

[34] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. *NIPS*, pages 1097–1105, 2012.

[35] L. Leal-Taixé, A. Milan, I. D. Reid, S. Roth, and K. Schindler. Motchallenge 2015: Towards a benchmark for multi-target tracking. *CoRR*, abs/1504.01942, 2015.

[36] K. Lebeda, S. Hadfield, J. Matas, and R. Bowden. Texture-independent long-term tracking using virtual corners. *IEEE Transactions on Image Processing*, 25(1):359–371, 2016.

[37] K. Lebeda, J. Matas, and R. Bowden. Tracking the untrackable: How to track when your object is featureless. In *Proc. of ACCV DTCE*, 2012.

[38] J.-Y. Lee and W. Yu. Visual tracking by partition-based histogram backprojection and maximum support criteria. In *Proc. IEEE International Conference on Robotics and Biomimetic (ROBIO)*, 2011.

[39] A. Li, M. Li, Y. Wu, M.-H. Yang, and S. Yan. Nus-pro: A new visual tracking challenge. *IEEE-PAMI*, 2015.

[40] P. Liang, E. Blasch, and H. Ling. Encoding color information for visual tracking: Algorithms and benchmark. *IEEE Transactions on Image Processing*, 24(12):5630–5644, 2015.

[41] A. L. Luka Čehovin Zajc, Alan Lukežič and M. Kristan. Beyond standard benchmarks: Parameterizing performance evaluation in visual object tracking. *ICCV*, abs/1612.00089, 2017.

[42] A. Lukežič, L. Č. Zajc, and M. Kristan. Deformable parts correlation filters for robust visual tracking. *IEEE Transactions on Cybernetics*, PP(99):1–13, 2017.

[43] A. Lukezic, T. Vojir, L. Cehovin 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.

[44] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang. Hierarchical convolutional features for visual tracking. *ICCV*, 2015.

[45] M. Mueller, N. Smith, and B. Ghanem. A benchmark and simulator for uav tracking. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2016.

[46] M. Mueller, N. Smith, and B. Ghanem. Context-aware correlation filter tracking. In *CVPR*, 2017.

[47] N. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893, June 2005.

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

[49] G. Nebehay and R. Pflugfelder. Clustering of static-adaptive correspondences for deformable object tracking. In *Computer Vision and Pattern Recognition*, 2015.

[50] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(10):1090–1104, 2000.

[51] M. Poostchi, F. Bunyak, K. Palaniappan, and G. Seetharaman. Feature selection for appearance-based vehicle tracking in geospatial video. In *SPIE Defense, Security, and Sensing*, 2013.

[52] M. Poostchi, K. Palaniappan, F. Bunyak, M. Becchi, and G. Seetharaman. Efficient gpu implementation of the integral histogram. In *Asian Conference on Computer Vision*, pages 266–278. Springer, 2012.

[53] M. Poostchi, K. Palaniappan, and G. Seetharaman. Spatial pyramid context-aware moving vehicle detection and tracking in urban aerial imagery. In *14th IEEE International Conference on Advanced Video and Signal Based Surveillance*. IEEE, 2017.

[54] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *PAMI*, pages 1–1, 2016.

[55] D. A. Ross, J. Lim, R. S. Lin, and M. H. Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3):125–141, 2008.

[56] 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.

[57] P. Senna, I. N. Drummond, and G. S. Bastos. Real-time ensemble-based tracker with kalman filter. In *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images*. IEEE, 2017.

[58] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition*, pages 593 – 600, June 1994.

[59] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[60] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual Tracking: an Experimental Survey. *TPAMI*, 2013.

[61] F. Solera, S. Calderara, and R. Cucchiara. Towards the evaluation of reproducible robustness in tracking-by-detection. In *Advanced Video and Signal Based Surveillance*, pages 1 – 6, 2015.

[62] C. Sun, H. Lu, and M.-H. Yang. Learning spatial-aware regressions for visual tracking. *arXiv preprint arXiv:1706.07457*, 2017.

[63] A. Tran and A. Manzanera. Mixing Hough and color histogram models for accurate real-time object tracking. In *Int. Conf. on Computer Analysis of Images and Patterns (CAIP'17)*, volume 10424 of *Lecture Notes in Computer Science*, Ystad, Sweden, August 2017. Springer.

[64] J. Valmadre, L. Bertinetto, J. F. Henriques, A. Vedaldi, and P. H. Torr. End-to-end representation learning for correlation filter based tracking. *arXiv preprint arXiv:1704.06036*, 2017.

[65] J. Van de Weijer, C. Schmid, J. Verbeek, and D. Larlus. Learning color names for real-world applications. *IEEE Transactions on Image Processing*, 18(7):1512–1524, 2009.

[66] 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.

[67] L. Čehovin, M. Kristan, and A. Leonardis. Is my new tracker really better than yours? *WACV 2014: IEEE Winter Conference on Applications of Computer Vision*, 2014.

[68] L. Čehovin, A. Leonardis, and M. Kristan. Visual object tracking performance measures revisited. *IEEE Transactions on Image Processing*, 25(3), 2015.

[69] L. Čehovin, A. Leonardis, and M. Kristan. Robust visual tracking using template anchors. In *WACV*. IEEE, Mar 2016.

[70] E. Velasco-Salido and J. M. Martínez. Scale adaptive point-based kanade lukas tomasi colour-filter tracker. *Under Review*, 2017.

[71] T. Vojíř and J. Matas. The enhanced flock of trackers. In R. Cipolla, S. Battiato, and G. M. Farinella, editors, *Registration and Recognition in Images and Videos*, volume 532 of *Studies in Computational Intelligence*, pages 113–136. Springer Berlin Heidelberg, Springer Berlin Heidelberg, January 2014.

[72] T. Vojíř and J. Matas. Pixel-wise object segmentations for the VOT 2016 dataset. Research Report CTU–CMP–2017–01, Center for Machine Perception, K13133 FEE Czech Technical University, Prague, Czech Republic, January 2017.

[73] T. Vojíř, J. Noskova, and J. Matas. Robust scale-adaptive mean-shift for tracking. *Pattern Recognition Letters*, 49:250–258, 2014.

[74] L. Wang, W. Ouyang, X. Wang, and H. Lu. Visual tracking with fully convolutional networks. In *ICCV*, 2015.

[75] Q. Wang, J. Gao, J. Xing, M. Zhang, and W. Hu. Dcfnet: Discriminant correlation filters network for visual tracking. *arXiv preprint arXiv:1704.04057*, 2017.

[76] Y. Wu, J. Lim, and M. H. Yang. Online object tracking: A benchmark. In *Computer Vision and Pattern Recognition*, 2013.

[77] Y. Wu, J. Lim, and M.-H. Yang. Object tracking benchmark. *PAMI*, 37(9):1834–1848, 2015.

[78] D. P. Young and J. M. Ferryman. PETS Metrics: On-line performance evaluation service. In *ICCCN '05 Proceedings of the 14th International Conference on Computer Communications and Networks*, pages 317–324, 2005.

[79] X. Yu and Q. Yu. Online structural learning with dense samples and a weighting kernel. *Pattern Recognition Letters*, 2017.

[80] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. *CoRR*, abs/1504.03641, 2015.

[81] J. Zhang, S. Ma, and S. Sclaroff. MEEM: robust tracking via multiple experts using entropy minimization. In *ECCV*, 2014.

[82] T. Zhang, S. Liu, C. Xu, and M.-H. Yang. Correlation particle filter for visual tracking. *IEEE Transactions on Image Processing*, 2017.

[83] T. Zhang, C. Xu, and M.-H. Yang. Multi-task correlation particle filter for robust object tracking. In *CVPR*, pages 1–9, 2017.