The Visual Object Tracking VOT2015 challenge results

Matej Kristan[1], Jiri Matas[2], Aleš Leonardis[3], Michael Felsberg[4], Luka Čehovin[1], Gustavo Fernández[5], Tomáš Vojíř[2], Gustav Häger[4], Georg Nebehay[5], Roman Pflugfelder[5], Abhinav Gupta[6], Adel Bibi[7], Alan Lukežič[1], Alvaro Garcia-Martin[8], Amir Saffari[10], Alfredo Petrosino[12], Andrés Solís Montero[13], Anton Varfolomieiev[14], Atilla Baskurt[15], Baojun Zhao[16], Bernard Ghanem[7], Brais Martinez[17], ByeongJu Lee[18], Bohyung Han[19], Chaohui Wang[20], Christophe Garcia[21], Chunyuan Zhang[22,23], Cordelia Schmid[24], Dacheng Tao[25], Daijin Kim[19], Dafei Huang[22,23], Danil Prokhorov[26], Dawei Du[27,28], Dit-Yan Yeung[29], Eraldo Ribeiro[30], Fahad Shahbaz Khan[4], Fatih Porikli[31,32], Filiz Bunyak[33], Gao Zhu[31], Guna Seetharaman[35], Hilke Kieritz[37], Hing Tuen Yau[38], Hongdong Li[31,39], Honggang Qi[27,28], Horst Bischof[40], Horst Possegger[40], Hyemin Lee[19], Hyeonseob Nam[19], Ivan Bogun[30], Jae-chan Jeong[41], Jae-il Cho[41], Jae-Yeong Lee[41], Jianke Zhu[42], Jianping Shi[43], Jiatong Li[25,16], Jiaya Jia[43], Jiayi Feng[44], Jin Gao[44], Jin Young Choi[18], Ji-Wan Kim[41], Jochen Lang[13], Jose M. Martinez[8], Jongwon Choi[18], Junliang Xing[44], Kai Xue[36], Kannappan Palaniappan[33], Karel Lebeda[45], Karteek Alahari[24], Ke Gao[33], Kimin Yun[18], Kin Hong Wong[38], Lei Luo[22], Liang Ma[36], Lipeng Ke[27,28], Longyin Wen[27], Luca Bertinetto[46], Mahdieh Pootschi[33], Mario Maresca[12], Martin Danelljan[4], Mei Wen[22,23], Mengdan Zhang[44], Michael Arens[37], Michel Valstar[17], Ming Tang[44], Ming-Ching Chang[27], Muhammad Haris Khan[17], Nana Fan[49], Naiyan Wang[29,11], Ondrej Miksik[46], Philip H S Torr[46], Qiang Wang[44], Rafael Martin-Nieto[8], Rengarajan Pelapur[33], Richard Bowden[45], Robert Laganière[13], Salma Moujtahid[15], Sam Hare[47], Simon Hadfield[45], Siwei Lyu[27], Siyi Li[29], Song-Chun Zhu[48], Stefan Becker[37], Stefan Duffner[15,21], Stephen L Hicks[46], Stuart Golodetz[46], Sunglok Choi[41], Tianfu Wu[48], Thomas Mauthner[40], Tony Pridmore[17], Weiming Hu[44], Wolfgang Hübner[37], Xiaomeng Wang[17], Xin Li[49], Xinchu Shi[44], Xu Zhao[44], Xue Mei[26], Yao Shizeng[33], Yang Hua[24], Yang Li[42], Yang Lu[48], Yuezun Li[27], Zhaoyun Chen[22,23], Zehua Huang[34], Zhe Chen[25], Zhe Zhang[9], Zhenyu He[49], and Zhibin Hong[25]

[1]University of Ljubljana, Slovenia
[2]Czech Technical University, Czech Republic
[3]University of Birmingham, United Kingdom
[4]Linköping University, Sweden
[5]Austrian Institute of Technology, Austria
[6]Carnegie Mellon University, USA
[7]King Abdullah University of Science and Technology, Saudi Arabia
[8]Universidad Autónoma de Madrid, Spain
[9]Baidu Corporation, China
[10]Affectv, United Kingdom
[11]TuSimple LLC
[12]Parthenope University of Naples, Italy
[13]University of Ottawa, Canada
[14]National Technical University of Ukraine, Ukraine
[15]Université de Lyon, France
[16]Beijing Institute of Technology, China
[17]University of Nottingham, United Kingdom

[18]Seoul National University, Korea
[19]POSTECH, Korea
[20]Université Paris-Est, France
[21]LIRIS, France
[22]National University of Defense Technology, China
[23]National Key Laboratory of Parallel and Distributed Processing Changsha, China
[24]INRIA Grenoble Rhône-Alpes, France
[25]University of Technology, Australia
[26]Toyota Research Institute, USA
[27]University at Albany, USA
[28]SCCE, Chinese Academy of Sciences, China
[29]Hong Kong University of Science and Technology, Hong Kong
[30]Florida Institute of Technology, USA
[31]Australian National University, Australia
[32]NICTA, Australia
[33]University of Missouri, USA
[34]Carnegie Mellon University, USA
[35]Naval Research Lab, USA
[36]Harbin Engineering University, China
[37]Fraunhofer IOSB, Germany
[38]Chinese University of Hong Kong, Hong Kong
[39]ARC Centre of Excellence for Robotic Vision, Australia
[40]Graz University of Technology, Austria
[41]Electronics and Telecommunications Research Institute, Korea
[42]Zhejiang University, China
[43]CUHK, Hong Kong
[44]Institute of Automation, Chinese Academy of Sciences, China
[45]University of Surrey, United Kingdom
[46]Oxford University, United Kingdom
[47]Obvious Engineering, United Kingdom
[48]University of California, USA
[49]Harbin Institute of Technology, China

Abstract

The Visual Object Tracking challenge 2015, VOT2015, aims at comparing short-term single-object visual trackers that do not apply pre-learned models of object appearance. Results of 62 trackers are presented. The number of tested trackers makes VOT 2015 the largest benchmark on short-term tracking to date. For each participating tracker, a short description is provided in the appendix. Features of the VOT2015 challenge that go beyond its VOT2014 predecessor are: (i) a new VOT2015 dataset twice as large as in VOT2014 with full annotation of targets by rotated bounding boxes and per-frame attribute, (ii) extensions of the VOT2014 evaluation methodology by introduction of a new performance measure. The dataset, the evaluation kit as well as the results are publicly available at the challenge website[1].

1. Introduction

Visual tracking is diverse research area that has attracted significant attention over the last fifteen years [20, 48, 18, 27, 49, 79, 43]. The number of accepted motion and tracking papers in high profile conferences, like ICCV, ECCV and CVPR, has been consistently high in recent years (∼40 papers annually). But the lack of established performance evaluation methodology combined with aforementioned high publication rate makes it difficult to follow the advancements made in the field.

Several initiatives have attempted to establish a common ground in tracking performance evaluation, starting with PETS [80] as one of most influential tracking performance analysis efforts. Other frameworks have been presented since with focus on surveillance systems and event detection, e.g., CAVIAR[2], i-LIDS [3], ETISEO[4], change detection [22], sports analytics (e.g., CVBASE[5]), faces, e.g. FERET [56] and [30], and the recent long-term tracking and detection of general targets[6] to list but a few.

This paper discusses the VOT2015 challenge organized in conjunction with the ICCV2015 Visual object tracking workshop and the results obtained. The challenge considers single-camera, single-target, model-free, causal trackers, applied to short-term tracking. The *model-free* property means that the only supervised training example is provided by the bounding box in the first frame. The *short-term* tracking means that the tracker does not perform re-detection after the target is lost. Drifting off the target is considered a failure. The *causality* means 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 VOT2015.

1.1. Related work

Several works that focus on performance evaluation in short-term visual object tracking [38, 36, 34, 64, 65, 76, 61, 77, 42] have been published over the last three years. The currently most widely used methodologies for performance evaluation originate from three benchmark papers, in particular the Online tracking benchmark (OTB) [76], the Amsterdam Library of Ordinary Videos (ALOV) [61] and the Visual object tracking challenge (VOT) [38, 36, 34]. The differences between these methodologies are outlined in the following paragraphs.

Performance measures. The OTB and the ALOV evaluate a tracker by initializing it on the first frame and letting it run until the end of the sequence, while the VOT resets the tracker once it drifts off the target. In all three methodologies the tracking performance is evaluated by overlaps between the bounding boxes predicted from the tracker with the ground truth bounding boxes. The ALOV measures the tracking performance as the F-measure at 0.5 overlap. The OTB introduced a success plot which represents the percentage of frames for which the overlap measure exceeds a threshold, with respect to different thresholds, and introduced an ad-hoc performance measure computed as the area under the curve in this plot. It was only later proven theoretically by other researchers [64] that the area under the curve equals the average overlap computed from all overlaps on the sequence. In fact, Čehovin et al. [64, 65] provided a highly detailed theoretical and experimental analysis of a number of the popular performance measures. Based on that analysis, the VOT2013 [38] selected the average overlap with resets and number of tracking failures as the main performance measures.

In the recent paper [34], the VOT committee analyzed the properties of average overlap with and without resets in terms of tracking accuracy estimator. The analysis showed that the OTB no-reset measure is a biased estimator while the VOT average overlap with resets drastically reduces the bias. A more significant finding was that the variance of the no-rest estimator [76] is orders of magnitude larger than for the reset-based estimator [34], meaning that the no-reset measure becomes reliable only on extremely large datasets. And since the datasets typically do not contain sequences of equal lengths, the variance is even increased. The VOT2013 [38] introduced a ranking-based methodology that accounted for statistical significance of the results and this was extended with the tests of practical differences in the VOT2014 [36].

[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/

It should be noted that the large variance of no-reset estimator combined with small number of sequences can distort the performance measurements. An overview of the papers published at top five conferences over the last three years shows that in several cases the no-reset evaluation combined with average overlap is carried out only with selected sequences, not the entire datasets. Therefore it is not clear whether the improvements over the state-of-the-art in those papers can be attributed to theoretical improvements of trackers or just to a careful selection of sequences. Note that this was hinted in the paper from Pang et al. [53] who performed meta-analysis of second-best trackers of published tracking papers and concluded that authors often report biased results in favor of their tracker.

Datasets. The recent trend in datasets construction appears to be focused on increasing the number of sequences in the datasets [75, 77, 42, 61], but often much less attention is being paid to the quality of its construction and annotation. For example, some datasets disproportionally mix grayscale and color sequences and in most datasets the attributes like occlusion and illumination change are annotated only globally enthough they may occupy only a short subsequence of frames in a video. The VOT2013 [38] argued that large datasets do not imply diversity nor richness in attributes and proposed a special methodology for dataset construction with per-frame visual attribute labelling. The per-frame labelling is crucial for proper attribute-wise performance analysis. A recent paper [34] showed that performance measures computed from global attribute annotations are significantly biased toward the dominant attributes in the sequences, while the bias is significantly reduced with per-frame annotation, even in presence of miss annotations.

Most closely related works to the work presented in this paper are the recent VOT2013 [38] and VOT2014 [36] 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, allowing easy integration with third-party trackers. The datasets are per-frame annotated with visual attributes and a state-of-the-art performance evaluation methodology was presented that accounts for statistical significance as well as practical difference of the results. A tracking speed measure that aims at reduction of hardware influence was proposed as well. The results were published in joint papers with over 50 co-authors [38], [36], while 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[7]. The advances proposed by VOT have also influenced the development of related methodologies. For example, the recent [77] now acknowledges that their area under the curve is an average overlap measure and have also adopted a variant of resets from VOT. The recent [42] benchmark adapted the approach of analyzing performance on subsequences instead of entire sequences to study the effects of occlusion.

1.2. The VOT2015 challenge

The VOT2015 follows the VOT2014 challenge and considers the same class of trackers. The dataset and evaluation toolkit are provided by the VOT2015 organizers. The evaluation kit records the output bounding boxes from the tracker, and if it detects tracking failure, re-initializes the tracker. The authors attending the challenge were required to integrate their tracker into the VOT2014 evaluation kit, which automatically performed a standardized experiment. The results were analyzed by the VOT2015 evaluation methodology.

Participants were expected to submit a single set of results per tracker. Participants who have investigated several trackers submitted a single result per tracker. Changes in the parameters did not constitute a different tracker. The tracker was required to run with fixed parameters on 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. Further details are available from the challenge homepage[8].

The VOT2015 improvements over VOT2013 and VOT2014 are the following:

(i) A new fully-annotated dataset is introduced which doubles the number of sequences compared to VOT2014. The dataset is per-frame annotated with visual properties and the objects are annotated with rotated bounding boxes. The annotation process was subject to quality control to increase annotation consistency.

(ii) A new dataset construction methodology is introduced that performs end-to-end automatic sequence selection and focuses on the sequences that are considered difficult to track.

(iii) The evaluation system from VOT2014 [36] is extended for easier tracker integration.

(iv) The evaluation methodology is extended by introducing a new performance measure which is easily interpretable. The trackers are ranked and the winner is selected using this measure.

(v) The VOT2015 introduces the first sub-challenge VOT-TIR2015 that is held under the VOT umbrella and deals with tracking in infrared and thermal imagery. The challenge and VOT-TIR2015 results are discussed in a separate paper submitted to the VOT2015 workshop [16].

[7] http://www.votchallenge.net

[8] http://www.votchallenge.net/vot2015/participation.html

2. The VOT2015 dataset

The VOT2013 [38] and VOT2014 [36] introduced a semi-automatic sequence selection methodology to construct a dataset rich in visual attributes but small enough to keep the time for performing the experiments reasonably low. In VOT2015, the methodology is extended such that the sequence selection is fully automated and that the selection process focuses on sequences that are likely challenging to track.

The dataset was prepared as follows. The initial pool of sequences was created by combining the sequences from two existing datasets OTB [76, 75] (51 sequences) and ALOV [61] (315 sequences), PTR [69] and obtained over 30 additional sequences from other sources summing to a set of 443 sequences. After removal of duplicate sequences, grayscale sequences and sequences that contained objects with area smaller than 400 pixels, we obtained 356 sequences. The new automatic sequence selection protocol required approximate annotation of targets in all sequences by bounding boxes. For most sequences the annotations already existed and we annotated the targets with axis-aligned bounding boxes for the sequences with missing annotations. Next, the sequences were automatically clustered according to their similarity in terms of the following globally calculated sequence visual attributes:

1. *Illumination change* is defined as the average of the absolute differences between the object intensity in the first and remaining frames.

2. *Object size change* is the sum of averaged local size changes, where the local size change at frame t is defined as the average of absolute differences between the bounding box area in frame t and past fifteen frame.

3. *Object motion* is the average of absolute differences between ground truth center positions in consecutive frames.

4. *Clutter* is the average of per-frame distances between two histograms: one extracted from within the ground truth bounding box and one from an enlarged area (by factor 1.5) outside of the bounding box.

5. *Camera motion* is defined as the average of translation vector lengths estimated by key-point-based RANSAC between consecutive frames.

6. *Blur* was measured by the Bayes-spectral-entropy camera focus measure [35].

7. *Aspect-ratio change* is defined as the average of per-frame aspect ratio changes. The aspect ratio change at frame t is calculated as the ratio of the bounding box width and height in frame t divided by the ratio of the bounding box width and height in the first frame.

8. *Object color change* defined as the change of the average hue value inside the bounding box.

9. *Deformation* is calculated by dividing the images into 8×8 grid of cells and computing the sum of squared differences of averaged pixel intensity over the cells in current and first frame.

10. *Scene complexity* represents the level of randomness (entropy) in the frames and it was calculated as $e = \sum_{i=0}^{255} b_i \log b_i$, where b_i is the number of pixels with value equal to i.

11. *Absolute motion* is the median of the absolute motion difference of the bounding box center points of the first frame and current one.

Note that the first ten attributes are taken from the VOT2014 [37, 34], with the attributes *object size* and *object motion* redefined to make their calculation more robust. The eleventh attribute (*absolute motion*) is newly introduced.

To reduce the influence of the varied scales among the attributes a binarization procedure was applied. A k-means clustering with $k = 2$ was applied to all values of a given attribute, thus each value was assigned a value, either zero or one. In this way each sequence was encoded as an 11D binary feature vector and the sequences were clustered by the Affinity propagation (AP) [17] using the Hamming distance. The only parameter in AP is the exemplar prior value p, which was set according to the rule-of-thumb proposed in [17]. In particular, we have set $p = 1.25\alpha_{\mathrm{sim}}$, where α_{sim} is the average of the similarity values among all pairs of sequences. This resulted in $K = 28$ sequence clusters, where each cluster k contained a different number of sequences N_k. The clustering stability was verified by varying the scaling value in range 1.2 to 1.3. The number of clusters varied in range of ± 3 clusters, indicating a stable clustering at the chosen parameter value.

The goal of sequence selection is to obtain a dataset of size M in which the following five visual attributes specified in VOT2014 are sufficiently well represented: (i) occlusion, (ii) illumination change, (iii) motion change, (iv) size change, (v) camera motion. The binary attributes were concatenated to form a feature vector $\mathbf{f}_i$ for each sequence i. The global presence of four of these attributes, except from occlusion, is indicated by the automatically calculated binarized values that were used for clustering. All sequences were manually inspected and occlusion was indicated if the target was at least partially occluded at any frame in the sequence. To estimate the sequence tracking difficulty, three well performing, but conceptually different, trackers (FoT [67], ASMS [69], KCF [25]) were evaluated using the VOT2014 methodology on the approximately annotated bounding boxes. In particular, the raw accuracy (average overlap) and raw robustness (number of failures per sequence) were computed for each tracker on each sequence and quantized into ten levels (i.e., into interval [0,9]). The quantized robustness was calculated by clipping the raw robustness at nine failures and the quantized accuracy was computed by $9 - \lfloor 10\Phi \rfloor$, where Φ is the VOT accuracy. The final tracking difficulty measure was obtained as the average of the quantized accuracy and robustness.

With the five global attributes and tracking difficulty estimated for each sequence, the automatic sequence selection algorithm proceeded as follows. First, the most difficult sequence from each cluster is selected as an initial pool of sequences and a maximum number of samples $\{S_k\}_{k=1}^{K}$ for each cluster k is calculated. From the selected pool of sequences the weighted balance vector $\mathbf{b}^0$ is computed and normalized afterwards. The balance vector controls the attribute representation inside the pool of selected sequences. We use weights to account for the unbalance distribution of the attributes in the dataset and compute them as follows $\mathbf{w} = N_s / \sum_i \mathbf{f}_i$, i.e., lowering weights to the attributes that are most common, therefore would always over-represented and the sequence without this attribute would be selected most of the time (e.g. object motion attribute). After initialization, the algorithm iterates until the number of selected sequences reaches the desired number M. In each iteration, the algorithm computes the attributes that are least represented, **aw**, using a small hysteresis so that multiple attributes can be chosen. Then, the Hamming distance between the desired attributes **aw** and all sequences is computed, excluding the sequences already selected and the sequences that belong to cluster which has already S_k sequences selected in the pool. From the set of most attribute-wise similar sequences the most difficult one is selected and added to the pool. At the end, the balance vector is recomputed and the algorithm iterates again. The sequence selection algorithm is summarized in Algorithm 1.

As in the VOT2014, we have manually or semi-automatically labeled each frame in each selected sequence with five visual attributes: (i) occlusion, (ii) illumination change, (iii) motion change, (iv) size change, (v) camera motion. In case a particular frame did not correspond to any of the five attributes, we denoted it as (vi) unassigned. To ensure quality control, the frames were annotated by an expert and then verified by another expert. Note that these labels are not mutually exclusive. For example, most frames in the dataset contain camera motion.

The relevant objects in all sequences were manually re-annotated by rotated bounding boxes. The annotation guidelines were predefined and distributed among the annotators. The bounding boxes were placed such that they approximated the target well, with a large percentage of pixels within the bounding box (at least $> 60\%$) belonging to the target. Each annotation was verified by two experts and corrected if necessary. The resulting annotations were then processed by approximating the rotated bounding boxes by axis-aligned bounding boxes if the ratio between the shortest and largest box edge was higher than 0.95 since the rotation is ambiguous for approximately round objects. The processed bounding boxes were again verified by an expert.

Algorithm 1: Sequence sampling algorithm

```
Input  : $N_s$, $M$, $K$, $\{N_k\}_{k=1}^{K}$, $\{\mathbf{f}_i\}_{i=1}^{N_s}$, $\mathbf{w}$
Output: ids
Initialize, $t = 0$
    $\{S_k\}_{k=1}^{K}$, $S_k = \lfloor \frac{N_k M}{N_s} \rfloor$
    select the most difficult sequence from each cluster
    $\mathbf{ids}^0 = \{\mathrm{id}_1, \ldots, \mathrm{id}_K\}$
    $\mathbf{b}^0 = \mathbf{w} \sum_{i \in \mathbf{ids}} \mathbf{f}_i$, $\mathbf{b}^0 = \mathbf{b}^0 / |\mathbf{b}^0|$
Iterate, $t = t + 1$
while $|ids| < M$ do
    $\mathbf{aw} = (\mathbf{h} < \min(\mathbf{h}) + \frac{0.1}{n})$, $\mathbf{h} = \frac{\mathbf{b}^{t-1}}{\max(\mathbf{b}^{t-1})}$
    $\{\mathrm{id}_1, \ldots\} = \mathrm{argmin}_i\ \mathrm{dist}(\mathbf{f}_i, \mathbf{aw})$
    s.t. if $i \in$ cluster $k$ then $|\text{cluster } k \cap \mathbf{ids}^{t-1}| < S_k$
    select the most difficult sequence $\mathrm{id}^* \in \{\mathrm{id}_1, \ldots\}$
    $\mathbf{ids}^t = \mathbf{ids}^{t-1} \cup \{\mathrm{id}^*\}$
    $\mathbf{b}^t = \mathbf{w} \sum_{i \in \mathbf{ids}} \mathbf{f}_i$, $\mathbf{b}^t = \mathbf{b}^t / |\mathbf{b}^t|$
end
```

3. Performance measures

As in VOT2014 [36], the following two weakly correlated performance measures are used due to their high level of interpretability [64, 65]: (i) accuracy and (ii) robustness. The accuracy measures how well the bounding box predicted by the tracker overlaps with the ground truth bounding box. On the other hand, the robustness measures how many times the tracker loses the target (fails) during tracking. A failure is indicated when the overlap measure becomes zero. To reduce the bias in robustness measure, the tracker is re-initialized five frames after the failure and ten frames after re-initialization are ignored in computation to further reduce the bias in accuracy measure [37]. Stochastic trackers are run 15 times on each sequence to obtain a better statistics on performance measures. The 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.

To analyze performance w.r.t. the visual attributes, the two measures can be calculated only on the subset of frames in the dataset that contain a specific attribute (attribute subset). The trackers are ranked with respect to each measure separately. The VOT2013 [38] recognized that subsets of trackers might be performing equally well and this should be reflected in the ranks. Therefore, for each i-th tracker a set of equivalent trackers is determined. In the VOT2013 and VOT2014 [38, 36], the corrected rank of the i-th tracker is obtained by averaging the ranks of these trackers including the considered tracker. The use of average operator on ranks may lead to unintuitive values of corrected ranks. Consider a set of trackers in which four top-performing trackers are estimated to perform equally well under the equivalence tests. The averaging will assign them a rank of 2.5, meaning that no tracker will be ranked as 1. Adding several equally performing tracker to the set will further increase the corrected rank value. For that reason we replace the averaging with the *min* operator in the VOT2014. In particular, the corrected rank is computed as the minimal rank of the equivalent trackers. As in VOT2014 [37] tests of statistical significance of the performance differences as well as tests of practical differences are used. The practical difference test was introduced in VOT2014 [36] and accounts for the fact that ground truth annotations may be noisy. As a result it is impossible to claim that one tracker is outperforming another if the difference between these two trackers is in the range of annotation noise on a given sequence. The level of the annotation ambiguity under which the trackers performance difference is considered negligible is called the practical difference threshold.

Apart from accuracy and robustness, the tracking speed is also an important property that indicates practical usefulness of trackers in particular applications. To reduce the influence of hardware, the VOT2014 [36] introduced a new unit for reporting the tracking speed called equivalent filter operations (EFO) that reports the tracker speed in terms of a predefined filtering operation that the tookit automatically carries out prior to running the experiments. The same tracking speed measure is used in VOT2015.

3.1. VOT2015 expected average overlap measure

The raw value of the accuracy and robustness measure offer a significant insight into tracker performance and further insight is gained by ranking trackers w.r.t. each measure since statistical and practical differences are accounted for. The average of these rank lists was used in the VOT2013 and VOT2014 [38, 36] challenges as the final measure for determining the winner of the challenge. A high average rank means that a tracker was well-performing in accuracy as well as robustness relative to the other trackers.

While ranking does convert the accuracy and robustness to equal scales, the averaged rank cannot be interpreted in terms of a concrete tracking application result. To address this, the VOT2015 introduces a new measure that combines the raw values of per-frame accuracies and failures in a principled manner and has a clear practical interpretation.

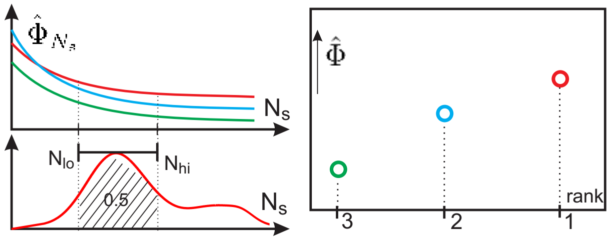

Figure 1. The expected average overlap curve (left, up), the sequence length pdf (left, bottom) and the expected average overlap plot (right).

Consider a short-term tracking example on a N_{s} frames long sequence. A tracker is initialized at the beginning of the sequence and left to track until the end. If a tracker drifts off the target it remains off until the end of the sequence. The tracker performance can be summarized in such a scenario by computing the average of per-frame overlaps, Φ_i, including the zero overlaps after the failure, i.e.,

$$\Phi_{N_{\mathrm{s}}} = \frac{1}{N_{\mathrm{s}}} \sum\nolimits_{i=1:N_{\mathrm{s}}} \Phi_i. \tag{1}$$

By averaging the average overlaps on a very large set of N_{s} frames long sequences, we obtain the expected average overlap $\hat{\Phi}_{N_{\mathrm{s}}} = \langle \Phi_{N_{\mathrm{s}}} \rangle$. Evaluating this measure for a range of sequence lengths, i.e., $N_{\mathrm{s}} = 1 : N_{\max}$ results in the *expected average overlap curve*. See for example Figure 1. The tracker performance is summarized as the VOT2015 *expected average overlap measure*, $\hat{\Phi}$, computed as the average of the *expected average overlap curve* values over an interval $[N_{\mathrm{lo}}, N_{\mathrm{hi}}]$ of typical short-term sequence lengths,

$$\hat{\Phi} = \frac{1}{N_{\mathrm{hi}} - N_{\mathrm{lo}}} \sum\nolimits_{N_{\mathrm{s}}=N_{\mathrm{lo}}:N_{\mathrm{hi}}} \hat{\Phi}_{N_{\mathrm{s}}}. \tag{2}$$

The tracker performance can be visualized by the VOT2015 *expected average overlap* plot shown in Figure 1. The performance measure in (2) requires computation of the expected average overlap $\hat{\Phi}_{N_{\mathrm{s}}}$ and specification of the range $[N_{\mathrm{lo}}, N_{\mathrm{hi}}]$. This is detailed in the following two subsections.

3.1.1 Estimation of expected average overlap

A brute force estimation of $\hat{\Phi}_{N_{\mathrm{s}}}$ (1) would in principle require running a tracker on an extremely large set of N_{s} frames long sequences and this process would have to be repeated for several values of N_{s} to compute the final performance measure $\hat{\Phi}$ (2). Note that this is in principle the OTB [76] measure computed on N_{s} frames-long sequences. But due to a large variance of such estimator [34], this would require a very large dataset and significant computation resources for the many tracker runs, since the experiments would have to be repeated for all values of N_{s}. Alternatively, the measure (2) can be estimated from the output of the VOT protocol.

Since the VOT protocol resets a tracker after each failure, several tracking segments are potentially produced per sequence and the segments from all sequences can be used to estimate the $\hat{\Phi}_{N_\mathrm{s}}$ as follows. All segments shorter than N_s frames that did not finish with a failure are removed and the remaining segments are converted into N_s frames long tracking outputs. The segments are either trimmed or padded with zero overlaps to the size N_s. An average overlap is computed on each segment and the average over all segments is the estimate of $\hat{\Phi}_{N_\mathrm{s}}$. Repeating this computation for different values of N_s produces an estimate of the expected average overlap curve.

3.1.2 Estimation of typical sequence lengths

The range of typical short-term sequence lengths $[N_\mathrm{lo}, N_\mathrm{hi}]$ in (2) is estimated as follows. A probability density function over the sequence lengths is computed by a kernel density estimate (KDE) [33, 32] from the given dataset sequence lengths and the most typical sequence length is estimated as the mode on the density. The range boundaries are defined as the closest points to the left and right of the mode for which $p(N_\mathrm{lo}) \approx p(N_\mathrm{hi})$ and the integral of the pdf within the range equals to 0.5. Thus the range captures the majority of typical sequence lengths (see Figure 1).

4. Analysis and results

4.1. Estimation of practical difference thresholds

The per sequence practical difference thresholds were estimated following the VOT2014 [36] protocol. Briefly, four frames with axis-aligned ground-truth bounding boxes were identified on each sequence and the annotators annotated those frames in several runs. By computing overlaps among all bounding boxes per frame, a set of 3300 samples of differences was obtained per sequence and used to compute the practical difference thresholds. Figure 2 shows boxplots of difference distributions w.r.t. sequences along side with examples of the annotations.

4.2. Estimation of sequence length range

The typical sequence range was estimated as discussed in Section 3.1.2. A batch KDE from [32] was applied to estimate the sequence length pdf from the lengths of sixty sequences of the VOT2015 dataset, resulting in the range values $[N_\mathrm{lo} = 108, N_\mathrm{hi} = 371]$. Figure 3 shows the estimated distribution along with the range values.

4.3. Trackers submitted

Together 41 entries have been submitted to the VOT2015 challenge. Each submission included the binaries/source code that was used by the VOT2015 committee for results verification. The VOT2015 committee additionally contributed 21 baseline trackers. For these, the default parameters were selected, or, when not available, were set to reasonable values. Thus in total 62 trackers were included in the VOT2015 challenge. In the following we briefly overview the entries and provide the references to original papers in the Appendix A where available.

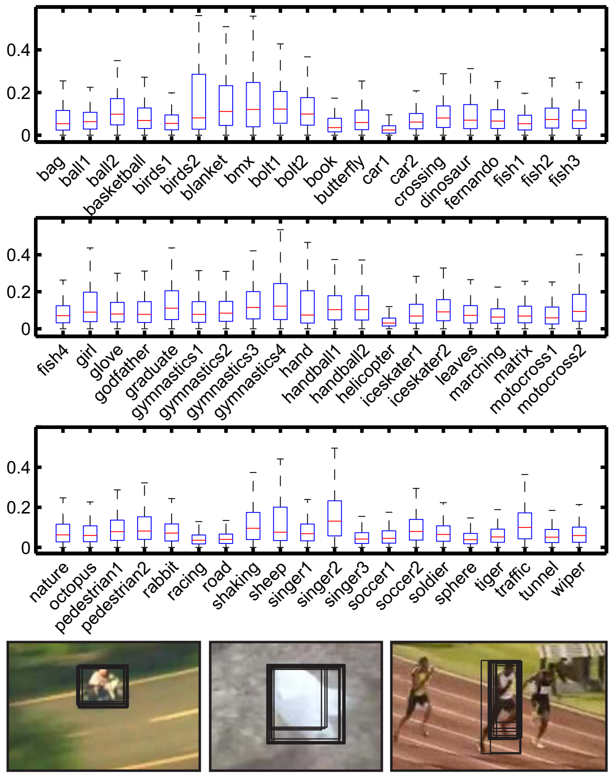

Figure 2. Box plots of differences per sequence along with examples of annotation variation.

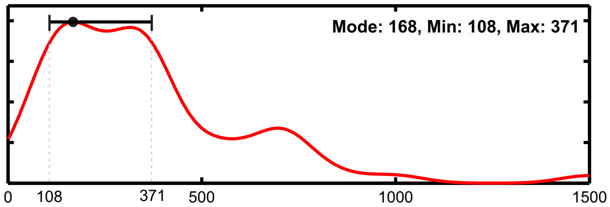

Figure 3. The estimated pdf of sequence lengths for the VOT2015 dataset (bottom).

Three trackers were based in convolutional neural networks, MDNet (A.29), DeepSRDCF (A.30) and SO-DLT (A.18), two trackers were using the object proposals [86] for object position generation or scoring, i.e., EBT (A.25) and KCFDP (A.21). Several trackers were based on Mean Shift tracker extensions [10], ASMS (A.48), SumShift (A.28), S3Tracker (A.32) and PKLTF (A.8), one tracker was based on distribution fields, DFT (A.59), several trackers were based on online boosting, OAB (A.44), MIL (A.47), MCT (A.20), CMIL (A.35), subspace learning IVT (A.46), CT (A.58), sparse learning L1APG (A.61), two trackers were based on tracking-by-detection learning MUSTer (A.1), sPST (A.41) and one tracker was based on pure color segmentation DAT (A.5). A number of trackers can be classified as part-based trackers. These were LDP (A.33), TRIC-track (A.22), G2T (A.17), AOG-Tracker (A.15), LGT (A.45), HoughTrack (A.53), MatFlow (A.7), CMT (A.42), LT-FLO (A.10), ZHANG (A.4), FoT (A.49), BDF (A.6), FCT (A.14), FragTrack (A.43). The CMT (A.42) and LT-FLO (A.10) can be considered long-term trackers meaning that they would liberally report a target loss. A number of trackers came from a class of holistic models that apply regression-based learning for target localization. Out for these, three were based on structured SVM learning, i.e., Struck (A.11), RobStruck (A.16), SRAT (A.38), one was based on Gaussian process regression, TGPR (A.51), one on logistic regression HRP (A.23) and one on kernelized-least-squares ACT (A.55). Several regression-based trackers used correlation filters [7, 25] as visual models. Some correlation filter based trackers maintained a single model for tracking, i.e., KCFv2 (A.2), DSST (A.56), SAMF (A.54), SRDCF (A.30), PTZ-MOSSE (A.12), NSAMF (A.24), RAJSSC (A.34), OACF (A.13), sKCF (A.3), LOFT-Lite (A.37), STC (A.50), MKCF+ (A.27), and several trackers applied multiple templates to model appearance variation, i.e., SME (A.19), MvCFT (A.9), KCFv2 (A.2) and MTSA-KCF (A.40). Some trackers combined several trackers or single-tracker instantiations HMMTxD (A.60), MEEM (A.62) and SC-EBT (A.26).

4.4. Results

The results are summarized in sequence pooled and attribute normalized AR rank and AR raw plots in Figure 4. The sequence pooled AR rank plot is obtained by concatenating the results from all sequences and creating a single rank list, while the attribute normalized AR rank plot is created by ranking the trackers over each attribute and averaging the rank lists. Similarly the AR raw plots were constructed. The raw values for the sequence pooled results are also given in Table 1.

The following trackers appear either very robust or very accurate among the top performing trackers on the sequence pooled AR-rank and AR-raw plots (closest to the upper right corner of rank plots): MDNet (A.29), DeepSRDCF (A.31), SRDCF (A.30), EBT (A.25), NSAMF (A.24), sPST (A.41), LDP (A.33), RAJSSC (A.34) and RobStruck (A.16). This set of trackers is followed by a large cluster of trackers that also perform nearly as well in accuracy, but with slightly reduced robustness. The situation is similar with per-attribute normalized plots, although several additional trackers like SODLT (A.18), OACF (A.13) and MvCFT (A.9) are pulled closer to the top-performing cluster. The two top-performing trackers, MDNet and DeepSRDCF, utilize convolutional neural network features. Note that these trackers are overlaid one over another in the AR-rank plots. MDNet is composed of two part-shared layers and doman-specific layers and has been trained on eighty sequences and ground truths that were not included in the VOT to obtain a generic representation of the sequence, while the DeepSRDCF is a correlation filter that used CNN kernels for feature extraction. The CNN features are also used in SODLT (A.18) which were trained to distinguish objects from non-objects. Several trackers are from a class of kernelized correlation filters [25] (KCF), i.e., SRDCF (A.30), DeepSRDCF (A.31), LDP (A.33), NSAMF (A.24), RAJSSC (A.34) and MvCFT (A.9). RAJSSC (A.34) is a KCF extended to address rotation in a correlation filter framework, NSAMF (A.24) is an extension of VOT2014 top-performing tracker that uses color in addition to edge features, SRDCF (A.30) is a regularized kernelized correlation filter that reduces the boundary effects in learning a filter and DeepSRDCF (A.31) is its extension that applies the convolution filters from a generically trained CNN [8] for feature extraction. MvCFT (A.9) applies a set of correlation filters for learning multiple object views and LDP (A.33) applies a deformable parts correlation filter to address non-rigid deformations. The tracker sPST (A.41) applies edge-box scores for hypothesis rescoring in combination with a linear SVM with HOG features for object detection and applies optical-flow-based Hough transform for estimation of object similarity transform. EBT (A.25) applies structured learning and object localization with edge-box region scores [86]. RobStruck (A.16) is an extension of the Struck [24] that uses richer features, adapts scale and applies a Kalman filter for motion estimation. Note that the submitted Struck (A.11) tracker is not the original [24], but its extension that applies multi-kernel learning and additional Haar and histogram features. According to the AR-rank plots (Figure 4), the top-two performing approaches are both based on CNNs, i.e., MDNet and DeepSRDCF. According to the AR-raw plots, the MDNet slightly outperforms the DeepSRDCF in accuracy as well as robustness. According to the ranking plots, the EBT perform on par with MDNet and DeepSRDCF in robustness.

The raw robustness with respect to the visual attributes are shown in Figure 5. The top three trackers with respect to the different visual attributes are mostly MDNet, DeepSRDCF and EBT with few exceptions. In the motion change attribute, the top-performing trackers are MKCF+ (A.27), MDNet and NSAMF (A.24). The most stable performance over the different attributes is observed for the MDNet and EBT tracker, with the attribute motion change being the most challenging. The motion change also

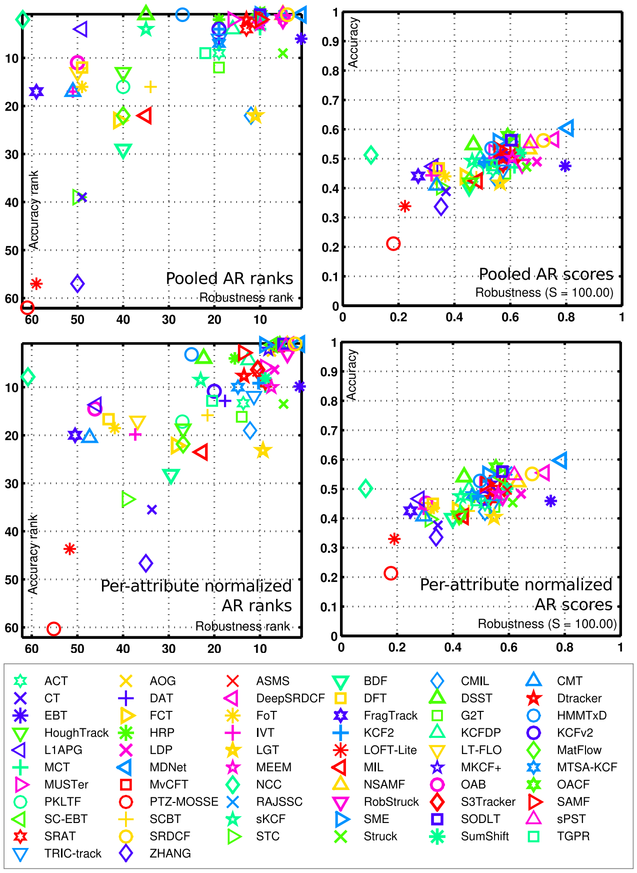

Figure 4. The AR rank plots and AR raw plots generated by sequence pooling (upper) and by attribute normalization (below).

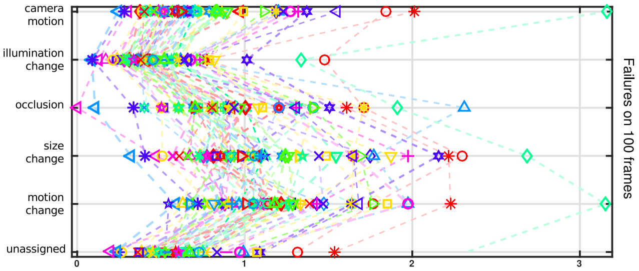

Figure 5. Robustness plots with respect to the visual attributes. See Figure 4 for legend.

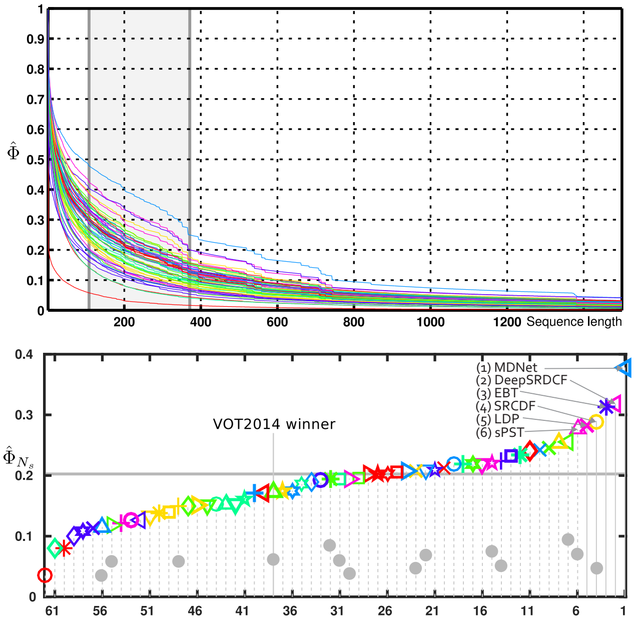

Figure 6. Expected average overlap curve (above) and expected average overlap graph (below) with trackers ranked from right to left. The right-most tracker is the top-performing according to the VOT2015 expected average overlap values. See Figure 4 for legend. The dashed horizontal line denotes the average performance of the state-of-the-art trackers published at ICCV, ECCV, CVPR, ICML or BMVC in 2014/2015 (nine papers from 2015 and six from 2014). These trackers are denoted by gray dots in the bottom part of the graph.

most significantly affects the DeepSRDCF relative to the performance of that tracker at other attributes.

The conclusions drawn from the analysis of the AR plots (Figure 4) are supported with the results from the expected average overlap scores in Figure 6. Since the MDNet scores highest in robustness and accuracy, it results in the highest expected average overlap, followed by the DeepSRDCF and closely behind is the EBT. The performance difference reflected by the expected average overlap score is also consistent with the expected average overlap curve in Figure 6. The MDNet consistently produces the highest overlap for all sequence lengths, followed by DeepSRDCF and EBT. The similarity in the expected average overlaps of EBT and DeepSRDCF comes from the fact that the DeepSRDCF is slightly more accurate during periods of successful tracking than EBT, but the EBT fails less often (see AR raw plots in Figure 4). As the result, the DeepSRDCF results in higher expected average overlap at short sequences, but slightly smaller on longer sequences. The fourth top-performing tracker is the SRDCF, followed closely by LDP and sPST. Table 1 shows all trackers ordered with respect to the expected average overlap scores. Note that the trackers that are usually used as baselines, i.e., OAB (A.44), MIL (A.47), IVT (A.46), CT (A.58) and L1APG (A.61) are positioned at the lower part of the list, which indicates that majority of submitted trackers are considered state-of-the-art. In fact, several tested trackers have been recently (in the last two years) published at major computer vision conferences. These trackers are pointed out in Figure 6, in which the average state-of-the-art performance computed from the average performance of these trackers is indicated. Observe that almost half of the submitted trackers are above this line. For completeness, we have also indicated the winner of VOT2014 in Figure 6. The advance of tested state-of-the-art since 2014 is clear.

Tracker	A	R	$\hat{\Phi}$	Speed	Impl.
MDNet*	**0.60**	**0.69**	**0.38**	0.87	M C G
DeepSRDCF*	0.56	1.05	*0.32*	0.38	M C
EBT	0.47	*1.02*	0.31	1.76	M C
SRDCF*	0.56	1.24	0.29	1.99	M C
LDP*	0.51	1.84	0.28	4.36	M C
sPST*	0.55	1.48	0.28	1.01	M C
SC-EBT	0.55	1.86	0.25	0.80	M C
NSAMF*	0.53	1.29	0.25	5.47	M
Struck*	0.47	1.61	0.25	2.44	C
RAJSSC	0.57	1.63	0.24	2.12	M
S3Tracker	0.52	1.77	0.24	14.27	C
SumShift	0.52	1.68	0.23	16.78	C
SODLT	0.56	1.78	0.23	0.83	M C G
DAT	0.49	2.26	0.22	9.61	M
MEEM*	0.50	1.85	0.22	2.70	M
RobStruck	0.48	1.47	0.22	1.89	C
OACF	*0.58*	1.81	0.22	2.00	M C
MCT	0.47	1.76	0.22	2.77	C
HMMTxD*	0.53	2.48	0.22	1.57	C
ASMS*	0.51	1.85	0.21	115.09	C
MKCF+	0.52	1.83	0.21	1.23	M C
TRIC-track	0.46	2.34	0.21	0.03	M C
AOG	0.51	1.67	0.21	0.97	*binary*
SME	0.55	1.98	0.21	4.09	M C
MvCFT	0.52	1.72	0.21	2.24	*binary*
SRAT	0.47	2.13	0.20	15.23	M C
Dtracker	0.50	2.08	0.20	10.43	C
SAMF*	0.53	1.94	0.20	2.25	M
G2T	0.45	2.13	0.20	0.43	M C
MUSTer	0.52	2.00	0.19	0.52	M C
TGPR*	0.48	2.31	0.19	0.35	M C
HRP	0.48	2.39	0.19	1.01	M C
KCFv2	0.48	1.95	0.19	10.90	M
CMIL	0.43	2.47	0.19	5.14	C
ACT*	0.46	2.05	0.19	9.84	M
MTSA-KCF	0.49	2.29	0.18	2.83	M
LGT*	0.42	2.21	0.17	4.12	M C
DSST*	0.54	2.56	0.17	3.29	M C
MIL*	0.42	3.11	0.17	5.99	C
KCF2*	0.48	2.17	0.17	4.60	M
sKCF	0.48	2.68	0.16	66.22	C
BDF	0.40	3.11	0.15	**200.24**	C
KCFDP	0.49	2.34	0.15	4.80	M
PKLTF	0.45	2.72	0.15	29.93	C
HoughTrack*	0.42	3.61	0.15	0.87	C
FCT	0.43	3.34	0.15	83.37	C
MatFlow	0.42	3.12	0.15	81.34	C
SCBT	0.43	2.56	0.15	2.68	C
DFT*	0.46	4.32	0.14	3.33	M
FoT*	0.43	4.36	0.14	143.62	C
LT-FLO	0.44	4.44	0.13	1.83	M C
L1APG*	0.47	4.65	0.13	1.51	M C
OAB*	0.45	4.19	0.13	8.00	C
IVT*	0.44	4.33	0.12	8.38	M
STC*	0.40	3.75	0.12	16.00	M
CMT*	0.40	4.09	0.12	6.72	C
CT*	0.39	4.09	0.11	12.90	M
FragTrack*	0.43	4.85	0.11	2.08	C
ZHANG	0.33	3.59	0.10	0.21	M
LOFT-Lite	0.34	6.35	0.08	0.75	M
NCC*	0.50	11.34	0.08	*154.98*	C
PTZ-MOSSE	0.20	7.27	0.03	18.73	C

Table 1. The table shows raw accuracy and the average number of failures, expected average overlap, tracking speed (in EFO) and implementation details (M is Matlab, C is C or C++, G is GPU). Trackers marked with * have been verified by the VOT2015 committee.

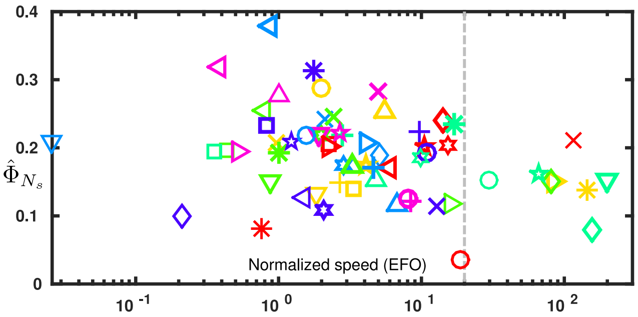

Figure 7. Expected average overlap scores w.r.t. the tracking speed in EFO units. The dashed vertical line denotes the estimated real-time performance threshold of 20 EFO units. See Figure 4 for legend.

Apart from tracking accuracy, robustness and expected average overlap at N_s frames, the tracking speed is also crucial in many realistic tracking applications. We therefore visualize the expected overlap score with respect to the tracking speed measured in EFO units in Figure 7. To put EFO units into perspective, a C++ implementation of a NCC tracker provided in the toolkit runs with average 140 frames per second on a laptop with an Intel Core i5-2557M processor, which equals to approximately 160 EFO units. Note that the two top-performing trackers according to the expected overlap graph, MDNet and DeepSRDCF, are among the slowest, which is likely due to the use of the CNN. For example, DeepSRDCF and SRDCF differ only in that DeepSRDCF applies CNN features which slows the tracker down by an order of magnitude. The vertical dashed line in Figure 7 indicates the real-time speed (equivalent to approximately 20fps). The top-performing tracker in terms of expected overlap among the trackers that exceed the real-time threshold is the scale-adaptive mean shift tracker, ASMS (A.48). From the AR rank plots we can see that this tracker achieves decent accuracy and robustness ranks, i.e., it achieves rank 10 to 20 in robustness and approximately rank 10 in accuracy. The raw values show that it tracks with a good accuracy of approximately 0.5 overlap during successful tracks, and the probability of still tracking after $S = 100$ frames is approximately 0.6. So this tracker tracks well in the short run. From the per-attribute failure plots (Figure 5) we can see that this tracker is most strongly affected by occlusion and motion change. The tracking speed methodology that we have employed has some limitations, e.g. note that SC-EBT was run distributed, so the measured time is much lower than the actual, since the toolkit considered only a single computer that performed the speed benchmarking.

5. Conclusions

This paper reviewed the VOT2015 challenge and its results. The challenge contains an annotated dataset of sixty sequences in which targets are denoted by rotated bounding boxes to aid a precise analysis of the tracking results. All the sequences are per-frame labeled with visual attributes and have been selected using a novel automatic sequence selection protocol that focuses on the sequences that are likely difficult to track, while ensuring balance in visual attributes. A new performance measure for determining the winner of the challenge was introduced, which estimates the expected average overlap of a tracker over a range of short-term tracking sequence lengths. Using this setup, a set of 62 trackers have been evaluated. A number of trackers submitted have been published at recent conferences, including BMVC2015, ICML2015, ECCV2014, CVPR2015 and ICCV2015, and some trackers have not yet been published (available at arXiv), which makes this the largest and most challenging benchmark to date.

The results of VOT2015 indicate that the best submitted tracker of the challenge according to the expected average overlap score is the MDNet (A.29) tracker. This tracker excelled in accuracy as well as robustness, which indicates that the tracker is tracking at a high accuracy during successful tracks and very rarely fails. As result, the expected average overlap over the VOT2015 defined interval of sequences lengths is greater by a decent margin than the second-best tracker. While the tracker performs very well under the overlap measures, it is computationally quite complex, resulting in a very slow tracking, which limits its practical applicability. It will be interesting to see in future whether certain steps could be simplified to achieve a faster tracking at comparable overlap performance.

The main goal of VOT is establishing a community-based common platform for discussion of tracking performance evaluation and contributing to the tracking community with verified annotated datasets, performance measures and evaluation toolkits. The VOT2015 was a third attempt toward this, following the very successful VOT2013 and VOT2014. The VOT2015 also introduced a new sub-challenge VOT-TIR that concerns tracking in thermal and infrared imagery. The results of that sub-challenge are described in a separate paper [16] that was presented at the VOT2015 workshop. Our future work will be focused on revising the evaluation kit, dataset, performance measures, and possibly launching other sub-challenges focused to narrow application domains, depending on the feedbacks and interest expressed from the community.

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 projects J2-4284, J2-3607, J2-2221 and European Union seventh framework programme under grant agreement no 257906. Jiri Matas and Tomas Vojir were supported by CTU Project SGS13/142/OHK3/2T/13 and by the Technology Agency of the Czech Republic project TE01020415 (V3C – Visual Computing Competence Center). Michael Felsberg and Gustav Häger were supported by the Swedish Foundation for Strategic Research through the project CUAS and the Swedish Research Council trough the project EMC^2. Some experiments where run on GPUs donated by NVIDIA.

A. Submitted trackers

In this appendix we provide a short summary of all trackers that were considered in the VOT2015 challenge.

A.1. Multi-Store Tracker (MUSTer)

Zhibin Hong, Zhe Chen, Chaohui Wang, Xue Mei, Danil Prokhorov, Dacheng Tao
{zhibin.hong, zhe.chen}@student.uts.edu.au,
chaohui.wang@u-pem.fr,
{xue.mei, danil.prokhorov}@tema.toyota.com,
dacheng.tao@uts.edu.au

MUlti-STore Tracker (MUSTer) [26] is a dual-component approach to object tracking, proposed with the inspiration from the Atkinson-Shiffrin Memory Model [2]. It consists of a short-term memory and a long-term memory. The short-term memory provides an instant response via two-stage filtering. When a failure or an occlusion is detected, the long-term memory estimates the state of the target and the short-term memory of the target appearance is refreshed accordingly. The reader is referred to [26] for details.

A.2. Restore Point guided Kernelized Correlation Filters (KCFv2)

Liang Ma, Kai Xue
mllx01161110@hotmail.com, xuekai@hrbeu.edu.cn

For target tracking, Kernelized Correlation Filters [25] use an online Support Vector Machine learning process in Fourier domain. Th KCFv2 tracker enhances its robustness by examining the similarity between each candidate patch generated by the KCF tracker and the Restore Point patch. This base patch characterizes target appearance in a short time period. The similarity likelihood of top k candidate positions produced by the KCF tracker at neighbouring scales are also measured and the likelihood function involves the histogram of colour and gradient.

A.3. Scalable Kernel Correlation Filter with Sparse Feature Integration (sKCF)

Andrés Solís Montero, Jochen Lang, Robert Laganière
asolismon@uottawa.ca,
{jlang,laganiereg}@eecs.uottawa.ca

sKCF extends Kernalized Correlation Filter (KCF) framework by introducing an adjustable Gaussian window function and keypoint-based model for scale estimation to deal with the fixed size limitation in the Kernelized Correlation Filter. Fast HoG descriptors and Intels Complex Conjugate Symmetric (CCS) are also integrated into sKCF to boost achievable frame rates.

A.4. ZHANG

Zhe Zhang, Hing Tuen Yau, Kin Hong Wong
zhangzhe9011@gmail.com,
{*htyau, khwong*}*@cse.cuhk.edu.hk*

ZHANG tracker is composed by two phases, learning and matching. In the learning phase, a dictionary is built using dense patch sampling and a target histogram of the desired object is generated. In the second phase, dense patches are sampled and candidate coefficients and candidate histograms are also generated which are compared with the coefficients and histogram generated in the first phase. A mean transform is run to yield tracking in all of orientation, rotation and scale, simultaneously.

A.5. Distractor Aware Tracker (DAT)

Horst Possegger, Thomas Mauthner, Horst Bischof
{*possegger, mauthner, bischof*}*@icg.tugraz.at*

The Distractor Aware Tracker is an appearance-based tracking-by-detection approach. A discriminative model using color histograms is implemented to distinguish the object from its surrounding region. Additionally, a distractor-aware model term suppresses visually distracting regions whenever they appear within the field-of-view, thus reducing tracker drift. The reader is referred to [57] for details.

A.6. Best Displacement Flow (BDF)

Mario Maresca, Alfredo Petrosino
mariomaresca@hotmail.it, petrosino@uniparthenope.it

Best Displacement Flow is a short-term tracking algorithm based on the same idea of Flock of Trackers [66] in which a set of local tracker responses are robustly combined to track the object. Firstly, BDF performs a clustering to identify the Best Displacement vector which is used to update the object's bounding box. Secondly, BDF performs a procedure named Consensus-Based Reinitialization used to reinitialize candidates which were previously classified as outliers. Interested readers are referred to [46] for details.

A.7. Matrioska Best Displacement Flow (MatFlow)

Mario Maresca, Alfredo Petrosino
mariomaresca@hotmail.it, petrosino@uniparthenope.it

MatFlow enhances the performance of the first version of Matrioska [47] with response given by the short-term tracker BDF (see A.6). By default, MatFlow uses the trajectory given by Matrioska. In the case of a low confidence score estimated by Matrioska, the algorithm corrects the trajectory with the response given by BDF. The Matrioska's confidence score is based on the number of keypoints found inside the object in the initialization. If the object has not a good amount of keypoints (i.e. Matrioska is likely to fail), the algorithm will use the trajectory given by BDF that is not sensitive to low textured objects.

A.8. Point-based Kanade Lukas Tomasi color-Filter (PKLTF)

Rafael Martin-Nieto, Alvaro Garcia-Martin, Jose M. Martinez
{*rafael.martinn, alvaro.garcia, josem.martinez*}*@uam.es*

PKLTF is a single-object long-term tracker that supports high appearance changes in the target, occlusions, and is also capable of recovering a target lost during the tracking process. PKLTF consists of two phases: The first one uses the Kanade Lukas Tomasi approach (KLT) [60] to choose the object features (using color and motion coherence), while the second phase is based on mean shift gradient descent [9] to place the bounding box into the position of the object. The object model is based on the RGB color and the luminance gradient and it consists of a histogram including the quantized values of the color components, and an edge binary flag. The interested reader is referred to [] for details.

A.9. Multi-view visual tracking via correlation filters (MvCFT)

He Zhenyu, Xin Li, Nana Fan
zyhe@hitsz.edu.cn

MvCFT tracker selects HoG features and intensity information to build up a model of the desired object. Correlation filters are used to generate different views of the model. An additional simple scale method is used to scale the size of the object.

A.10. Long Term Featureless Object Tracker (LT-FLO)

Karel Lebeda, Simon Hadfield, Jiri Matas, Richard Bowden
{*k.lebeda, s.hadfield, r.bowden*}*@surrey.ac.uk,*
matas@cmp.felk.cvut.cz

The tracker is based on and extends previous work of the authors on tracking of texture-less objects [40]. It significantly decreases reliance on texture by using edge-points instead of point features. LT-FLO uses correspondences of lines tangent to the edges and candidates for a correspondence are all local maxima of gradient magnitude. An estimate of the frame-to-frame transformation similarity is obtained via RANSAC. When the confidence is high, the current state is learnt for future corrections. On the other hand, when a low confidence is achieved, the tracker corrects its position estimate restarting the tracking from previously stored states. LT-FLO tracker also has a mechanism to detect disappearance of the object, based on the stability of the gradient in the area of projected edge-points. The interested reader is referred to [39] for details.

A.11. Struck

Stuart Golodetz, Sam Hare, Amir Saffari, Stephen L. Hicks, Philip H. S. Torr
sgolodetz@gxstudios.net, sam@samhare.net, amir@ymer.org, stephen.hicks@ndcn.ox.ac.uk, philip.torr@eng.ox.ac.uk

Struck 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. Current version of Struck uses multi-kernel learning (MKL) and larger feature vectors than were used in the past. The tracking performance is significantly improved by combining a Gaussian kernel on 192D Haar features with an intersection kernel on 480D histogram features, but at a cost in speed. Note that this version of the tracker is an improvement over the initial Struck from ICCV2011 [24] and was in the time of writing this paper under review as a journal submission.

A.12. PTZ-MOSSE

ByeongJu Lee, Kimin Yun, Jongwon Choi, Jin Young Choi
adolys@snu.ac.kr, ykmwww@snu.ac.kr, jwchoi.pil@gmail.com, jychoi@snu.ac.kr

PTZ-MOSSE tracker improves the robustness against occlusions and appearance changes by using motion likelihood map and scale change estimation as well as appearance correlation filter. A motion likelihood map is constructed from motion detection result in addition to the correlation filter. This map is generated by blurring the motion detection result, which shows high probability in the center of the target. The combination of the correlation filter and the motion likelihood map is formulated as an optimization problem.

A.13. Object-Aware Correlation Filter Tracker (OACF)

Luca Bertinetto, Ondrej Miksik, Stuart Golodetz, Philip H. S. Torr
{luca.bertinetto, ondrej.miksik}@eng.ox.ac.uk, stuart.golodetz@ndcn.ox.ac.uk, philip.torr@eng.ox.ac.uk

OACF tracker extends the scale adaptive DSST tracker [11] by using a per-pixel likelihood map of the target which is built using RGB histograms. Then, for each pixel x is estimated the probability that the pixel belongs to the object to track refining the estimation of a correlation filter. Details are available in [6].

A.14. Optical flow clustering tracker (FCT)

Anton Varfolomieiev
a.varfolomieiev@kpi.ua

FCT is based on the same idea as the best displacement tracker (BDF) [46]. It uses sparse pyramidal Lucas-Kanade optical flow to track individual points of the object at several pyramid levels. The results of point tracking are clustered in the same way as in BDF [46] to estimate the best object displacement. The initial point locations are generated by the FAST detector [59]. The tracker estimates the scale and an in-plane rotation of the object. These procedures are similar to the scale calculation of the median flow tracker [29], except that the clustering is used instead of median. In case of rotation calculation an angles between the respective point pairs are clustered. In contrast to BDF, the FCT does not use consensus-based reinitialization, but regenerate a regular grid of missed points, when the number of these points becomes less than certain predefined threshold.

A.15. AOGTracker

Tianfu Wu, Yang Lu, Song-Chun Zhu
{tfwu, yanglv}@ucla.edu, sczhu@stat.ucla.edu

AOGTracker tracker simultaneously tracks, learns and parses objects in video sequences with a hierarchical and compositional And-Or graph (AOG). The AOG explores latent discriminative part configurations to represent objects. AOGTracker takes into account the appearance of the object (e.g., lighting and partial occlusion) and structural variations of the object (e.g., different poses and viewpoints), as well as objects in the background which are similar to the desired object to track. The AOGTracker is formulated under the Bayesian framework and a spatial-temporal dynamic programming (DP) algorithm is derived to infer the state of the object. During an online learning phase, the AOG is updated iteratively with two steps in the latent structural SVM framework: (i) Identifying the false positives and false negatives of the current AOG in a new frame by exploiting the spatial and temporal constraints observed in the trajectory; (ii) updating the structure of the AOG based on the intractability of the current AOG and re-estimating the parameters based on the augmented training dataset.

A.16. Structure Tracker with the Robust Kalman filter (RobStruck)

Ivan Bogun, Eraldo Ribeiro
ibogun2010@my.fit.edu, eribeiro@cs.fit.edu

RobStruck is a modified version of the Struck tracker [24] extended to work on multiple scales. Feature representation of the bounding box is done by extracting histograms of oriented gradients and intensity histograms. Intersection kernel is used as a kernel function. To make the tracker more resilient to false positives, Robust Kalman filter is used. Each detection of the SVM is corrected with the filter to find out if incorrect detection occurred.

A.17. Geometric Structure Hyper-Graph based Tracker (G2T)

Yuezun Li, Dawei Du, Longyin Wen, Lipeng Ke, Ming-Ching Chang, Honggang Qi, Siwei Lyu
{liyuezun, cvdaviddo, wly880815, lipengke1, mingching, honggangqi.cas, heizi.lyu}@gmail.com

G2T tracker is especially designed for tracking deformable objects. G2T represents the target object by a geometric structure hyper-graph, which integrates the local appearance of the target with higher order geometric structure correlations among target parts. In each video frame, tracking is formulated as a hyper-graph matching between the target geometric structure hyper-graph and a candidate hyper-graph. Multiple candidate associations between the nodes of both hyper-graphs are built. The weight of the nodes indicate the reliability of the candidate associations based on the appearance similarity between the corresponding parts of each hyper-graph. A matching between the target and a candidate is solved by applying the extended pairwise updating algorithm of [45].

A.18. Structure Output Deep Learning Tracker (SO-DLT)

Naiyan Wang, Siyi Li, Abhinav Gupta, Dit-Yan Yeung
winsty@gmail.com, sliay@cse.ust.hk, abhinavg@cs.cmu.edu, dyyeung@cse.ust.hk

SO-LDT proposes a novel structured output CNN which transfers generic object features for online tracking. First, a CNN is trained to distinguish objects from non-objects. The output of the CNN is a pixel-wise map to indicate the probability that each pixel in the input image belongs to the bounding box of an object. Besides, SO-LDT uses two CNNs which use different model update strategies. By making a simple forward pass through the CNN, the probability map for each of the image patches is obtained. The final estimation is then determined by searching for a proper bounding box. If it is necessary, the CNNs are also updated. The reader is referred to [71] for more details.

A.19. Scale-adaptive Multi-Expert Tracker (SME)

Jiatong Li, Zhibin Hong, Baojun Zhao
{Jiatong.Li-3@student., Zhibin.Hong@student., yida.xu@}uts.edu.au, zbj@bit.edu.cn

SME is a multi-expert based scale adaptive tracker inspired by [81]. Unlike [81], SME proposes a trajectory consistency based score function as the expert selection criteria. Furthermore, an effective scale adaptive scheme is introduced to handle scale changes on-the-fly. Multi-channel based correlation filter tracker [25] is adopted as the base tracker, where HOG and colour features [13] are concatenated to enhance the performance.

A.20. Motion Context Tracker (MCT)

Stefan Duffner, Christophe Garcia
{stefan.duffner, christophe garcia}@liris.cnrs.fr

The Motion Context Tracker (MCT) is a discriminative on-line learning classifier based on Online Adaboost (OAB) which is integrated into the model collecting negative training examples for updating the classifier at each video frame. Instead of taking negative examples only from the surroundings of the object region or from specific distracting objects, MCT samples the negatives from a contextual motion density function in a stochastic manner.

A.21. Kernelized Correlation Filter with Detection Proposal (KCFDP)

Dafei Huang, Zhaoyun Chen, Lei Luo, Mei Wen, Chunyuan Zhang
chenzhaoyun@nudt.edu.cn

KCFDP couples the Kernelized Correlation Filter(KCF) tracker [25] with the class-agnostic detection proposal generator EdgeBoxes [86]. KCF is responsible for the preliminary estimation of target location. Then EdgeBoxes is employed to search for detection proposals nearby. While the unpromising proposals are rejected before evaluation, the most promising candidate is used to refine the target location and update the target scale and aspect ratio with a damping factor. The feature used in original KCF is extended to a combination of HOG, intensity, and colour naming similarly to [13, 44], and the robust model updating scheme in [13] is also adopted.

A.22. Tracking by Regression with Incrementally Learned Cascades (TRIC-track)

Xiaomeng Wang, Michel Valstar, Brais Martinez, Muhammad Haris Khan, Tony Pridmore
{psxxw, Michel.Valstar, brais.martinez, psxmhk, tony.pridmore}@nottingham.ac.uk

TRIC-track is a part-based tracker which directly predicts the displacements between the centres of sampled image patches and the target part location using regressors. TRIC-track adopts the Supervised Descent Method (SDM) [78] to perform the cascaded regression for displacement prediction, estimating the target location with increasingly accurate predictions. To adapt to variations in target appearance and shape over time, TRIC-track takes inspiration from the incremental learning of cascaded regression of [1] applying a sequential incremental update. TRIC-track also possesses a multiple temporal scale motion model [31] which enables it to fully exert the trackers advantage by providing accurate initial prediction of the target part location every frame. For more details, the interested reader is referred to [74].

A.23. Baseline Tracker (HRP)

Naiyan Wang, Jianping Shi, Dit-Yan Yeung, Jiaya Jia
{*winsty, shijianping5000*}*@gmail.com,*
dyyeung@cse.ust.hk, leojia@cse.cuhk.edu.hk

The HRP tracker is the best combination of tracking parts produced by the analysis in [72]. The tracker is composed of a HoG visual model with logistic regression and particle filter for localization. The authors of the original paper [72] have submitted this tracker to VOT2015 under the name ”Baseline tracker”, but to avoid confusion with the VOT baselines, we have abbreviated it into HRP (indicating HoG features, regression and particle filter).

A.24. NSAMF

Yang Li, Jianke Zhu
{*liyang89, jkzhu*}*@zju.edu.cn*

NSAM is based on the correlation filter framework [25, 7]. NSAM tracker is an improved version of the previous method SAMF [44]. While the latter uses colour name, the former employs colour probability. In addition, the final response map is a fusion of multi-models based on the different features.

A.25. Edge Box Tracker (EBT)

Gao Zhu, Fatih Porikli, Hongdong Li
{*gao.zhu, fatih.porikli, hongdong.li*}*@anu.edu.au*

EBT tracker uses sparse yet informative contours to score proposals based on the number of contours they wholly enclose into a detection-by-tracking process for visual tracking. EBT executes search in the entire image and focus only on those high-quality candidates to test and update the discriminative classifier. To reduce the spurious false positives and improve the tracking accuracy, high-quality candidates are used to choose better positive and negative samples. Since EBT employs only a few candidates to search the object, it has potential to use higher-dimensional features if needed. The reader is referred to [85] for details.

A.26. Self-Correction Ensemble Based Tracker (SC-EBT)

Naiyan Wang, Zehua Huang, Siyi Li, Dit-Yan Yeung
winsty@gmail.com, zehuah@cmu.edu,
{*sliay, dyyeung*}*@cse.ust.hk*

SC-EBT ensembles the output of several individual trackers in order to make the final prediction more accurate and robust. This problem can be cast into a challenging crowd sourcing problem on structured data with temporal dimension. To solve it, a factorial hidden Markov model (FHMM) is proposed for ensemble-based tracking by learning jointly the unknown trajectory of the target and the reliability of each tracker in the ensemble. A conditional particle filter algorithm by exploiting the structure of the joint posterior distribution of the hidden variables is applied for online inference of the FHMM. Four complementary trackers were chosen to be used in ensemble, namely, DAT [57], DSST [11], Baseline [72] and ASMS [69]. For more details, the interested reader is referred to [73].

A.27. Multi-kernelized Correlation Filter Plus (MKCF+)

Ming Tang, Jiayi Feng, and Xu Zhao
{*tangm, jiayi.feng, xu.zhao*}*@nlpr.ia.ac.cn*

MKCF+ tracker is based on the multi-kernelized correlation filter tracker (MKCF) [62] and background modelling algorithm ViBe [5]. The model drift problem suffered by MKCF is tackled by MKCF+ by adapting ViBe to alarm its locating failures. ViBe is launched only on frames with stable scenes. And in such case, it is probable for ViBe to find out the possible locations of the target in searching area. The candidate locations are then tested by MKCF to determine which one should be the target.

A.28. SumShift

Jae-Yeong Lee, Sunglok Choi, Jae-chan Jeong, Ji-Wan Kim, Jae-il Cho
{*jylee, sunglok, channij80, giraffe, jicho*}*@etri.re.kr*

SumShift tracker is an implementation of the histogram-based tracker suggested in [41]. SumShift improves conventional histogram-based trackers (e.g., mean-shift tracker) in two ways. Firstly, it uses a partition-based object model represented by multiple patch histograms to preserve geometric structure of the colour distribution of the object. Secondly, the object likelihood is computed by the sum of the patch probabilities which are computed from each corresponding patch histograms, enabling more robust and accurate tracking. The reader is referred to [41] for details.

A.29. Multi-Domain Convolutional Neural Network Tracker (MDNet)

Hyeonseob Nam and Bohyung Han
{*namhs09,bhhan*}*@postech.ac.kr*

MDNet tracker represents the target object using a Convolutional Neural Network (CNN). MDNet pre-trains the CNN using a set of videos with tracking ground-truth annotations to obtain a generic representation for an arbitrary new sequence. The network is composed of two partsshared layers and domain specific layers, where domains correspond to individual tracking sequences and each domain has a separate branch for binary classification. After training, a generic representation in the shared layers across all domains is obtained. The tracking is performed by sampling target candidates around the previous target state, evaluating them on the CNN, and identifying the sample with the maximum score. For more details, the interested reader is referred to [51].

A.30. Spatially Regularized Discriminative Correlation Filter Tracker (SRDCF)

Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, Michael Felsberg
{martin.danelljan, gustav.hager, fahad.khan, michael.felsberg}@liu.se

Standard Discriminative Correlation Filter (DCF) based trackers such as [11, 13, 25] suffer from the inherent periodic assumption when using circular correlation. The resulting periodic boundary effects leads to inaccurate training samples and a restricted search region.

The SRDCF mitigates the problems arising from assumptions of periodicity in learning correlation filters 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. Instead of solving for an approximate filter, as in previous DCF based trackers (e.g. [11, 13, 25]), the SRDCF employs an iterative optimization based on Gauss-Seidel that converges to the exact filter. The detection step employs a sub-grid maximization of the correlation scores to achieve more precise location estimates. In addition to the HOG features used in [12], the submitted variant of SRDCF also employs Colour Names and greyscale features. These features are averaged over the 4×4 HOG cells and then concatenated, giving a 42 dimensional feature vector at each cell. For more details, the reader is referred to [12].

A.31. Spatially Regularized Discriminative Correlation Filter with Deep Features (DeepSRDCF)

Martin Danelljan, Gustav Häger, Fahad Shahbaz Khan, Michael Felsberg
{martin.danelljan, gustav.hager, fahad.khan, michael.felsberg}@liu.se

The DeepSRDCF incorporates deep convolutional features in the SRDCF framework proposed in [12]. Instead of the commonly used hand-crafted features, the DeepSRDCF employs convolutional features from a pre-trained network. A Principal Component Analysis is used to reduce the feature dimensionality of the extracted activations. The reader is referred to [12] for details.

A.32. Scaled SumShift Tracker (S3Tracker)

Jae-Yeong Lee, Sunglok Choi, Jae-chan Jeong, Ji-Wan Kim, Jae-il Cho
{jylee, sunglok, channij80, giraffe, jicho}@etri.re.kr

S3Tracker is based on the authors previous work SumShift [41], with adaptive scale and aspect ratio selection. S3Tracker is also one of RGB histogram-based trackers. In addition to SumShift, S3Tracker chooses the scale and aspect ratio through maximizing likelihood density with consideration of size and area of object candidates. Such maximum likelihood density criterion enables robust and adaptive object tracking.

A.33. Layered Deformable Parts tracker (LDP)

A. Lukežič, L. Čehovin, Matej Kristan
alan.lukezic@gmail.com

LDP 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 color 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 [25].

A.34. Rotation adaptive joint scale-spatial correlation filter based tracker (RAJSSC)

Mengdan Zhang, Junliang Xing, Jin Gao, Xinchu Shi, Qiang Wang, Weiming Hu
{mengdan.zhang, jlxing, jgao, xcshi, qiang.wang, wmhu}@nlpr.ia.ac.cn

RAJSSC tracker is a correlation filter based tracking, which is able to simultaneously model target appearance changes from spatial displacements, scale variations, and rotation transformations. RAJSSC performs scale-spatial correlation jointly using a novel block-circulant structure for the object template with a joint space Gaussian response. By transferring the target template from the Cartesian coordinate system to the Log-Polar coordinate system, the circulant structure is preserved and the object rotation can be evaluated.

A.35. Multi-Channel Multiple-Instance-Learning Tracker (CMIL)

Hilke Kieritz, Stefan Becker, Wolfgang Hubner, Michael Arens
{hilke.kieritz, stefan.becker, wolfgang.huebner, michael.arens}@iosb.fraunhofer.de

CMIL is an extension of the multiple-instance-learning tracker MIL [3] with the use of integral channel features [14]. The CMIL uses multiple features channels and only the sum of one region per feature. The following features are used: LUV-color channels, six per gradient direction quantized gradient magnitude channels and the gradient magnitude channel. To track the object over scale changes the feature responses are scaled using a scaling factor depended on the feature channel as [14].

A.36. DTracker

Jae-Yeong Lee, Jae-chan Jeong, Sunglok Choi, Ji-Wan Kim, Jae-il Cho
{*jylee, channij80, sunglok, giraffe, jicho*}*@etri.re.kr*

DTracker extends the SumShift tracker [41] with an optical flow tracker and the NCC tracker. The colour distribution of an object is modelled by kernel density estimation (KDE) to provide continuous measure of colour similarity. Similarity evaluation of the KDE colour model and the NCC template matching acts as global localizer to bound possible drift of the tracker and the optical flow tracker has a role of adopting frame to frame variation.

A.37. Likelihood of Features Tracking-Lit (LOFT-Lite)

Rengarajan Pelapur, Kannappan Palaniappan, Filiz Bunyak, Guna Seetharaman, Mahdieh Pootschi, Ke Gao, Yao Shizeng
{*rvpnc4, pal, bunyak, guna, mpr69, kg954, syyh4*}*@missouri.edu*

LOFT (Likelihood of Features Tracking) [52, 54, 55] is an appearance based single object tracker that uses a set of image based features and correlation maps including histograms of gradient magnitude, gradient orientation, neighbourhood intensity, and shape based on the eigenvalues of the Hessian matrix. LOFT performs feature fusion by comparing a target appearance model within a search region using Bayesian maps which estimate the likelihood of each pixel within the search window belonging to part of the target [54]. Newly added per-color channel histograms are used to improve accuracy and robustness. The search region is updates by a Kalman filter [55].

A.38. Scale Ratio Adaptive Tracker (SRAT)

Hyemin Lee, Daejin Kim
{*lhmin, dkim*}*@postech.ac.kr*

The Scale Ratio Adaptive Tracker (SRAT) is an extended version of structured output tracker(Struck) [24]. The object model is learnt by structured output SVM using Gaussian kernelized raw feature. The tracking process consists of three steps: First, find the 2-D transition which maximizes the SVM response based on the trained model. Second, estimate the scale changes including width and height variance. Since the 2-D scale estimation is very costly, the subspace of scale estimation space is used. Among the all possible scale changes, the guide line for x-y scale ratio and allowed only small variation are set. Third, a translation tracking step within the range made by scale change is performed. The ambiguity when more confident targets are similar to the object model is solved by using a weight on current target location based on Gaussian distribution.

A.39. Scene Context-Based Tracker (SCBT)

Salma Moujtahid, Stefan Duffner, Atilla Baskurt
{*salma.moujtahid, stefan.duffner, atilla.baskurt*}*@liris.cnrs.fr*

The Scene Context-Based Tracker (SCBT) [50] combines several independent on-line trackers using visual scene context. The framework decides automatically at each point in time which specific tracking algorithm works best under the given scene or acquisition conditions. A set of generic global context features computed on different regions of each frame of a set of training videos is defined. It is also recorded the performance of each individual tracker on these videos in terms of object bounding box overlap with the ground truth. Using these information, the classifier is trained to estimate which tracker gives the best result given the global scene context in a particular frame. In this framework, 3 Online AdaBoost trackers [23] were used based on Haar, HoG and HoC features, respectively. The context classifier estimates a probability for each tracker to be the best for the current frame. Then, to avoid frequent and unnecessary switching between different trackers, the classifier response in time using a Hidden Markov Model is filtered.

A.40. Multi-Template Scale Adaptive Kernelized Correlation Filters (MTSA-KCF)

Adel Bibi, Bernard Ghanem
{*adel.bibi, bernard.ghanem*}*@kaust.edu.sa*

This tracker is an improvement over the popular kernelized correlation filter tracker best known as KCF [25]. MTSA-KCF addresses two main issues, model-filter update and the fixed scaling issue. As for scaling, a simple voting over-grid method similar to [11, 44] is proposed. But, instead of maximizing over the likelihood term of the scale grid by assuming the scales are equiprobable, the posterior distribution is maximized by assuming the scales follow a Gaussian prior centered around the scale in the previous frame. The other contribution consists of using multiple templates, with multi-dimensional features and non-linear kernel functions in the dual formulation. By relaxing the original problem and solving an alternating fixed point method optimization, a significant improvement in performance is achieved with real-time speeds.

A.41. simplified Proposal Selection Tracker (sPST)

Yang Hua, Karteek Alahari, Cordelia Schmid
firstname.lastname@inria.fr

The simplified Proposal Selection Tracker (sPST) is based on current work [28]. sPST operates in two phases. Firstly, a set of candidate object locations computed by common tracking-by-detection framework is proposed. The frame is used as is and rotate them according to the ground truth annotation in the initial frame if applicable. Secondly, the best candidate as the tracking result is determined by two cues: detection confidence score and an objectness measure computed with edges [86]. The reader is referred to [28] for details.

A.42. 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.

A.43. FragTrack

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. OAB

Submitted by VOT Committee

OAB employs feature selection by online boosting for object tracking. This allows to adapt a classifier while tracking the object. Therefore appearance changes of the object (e.g. out of plane rotations, illumination changes) are handled quite naturally. Moreover, depending on the background the algorithm selects the most discriminating features for tracking resulting in stable tracking results. By using fast computable features (e.g. Haar-like wavelets, orientation histograms, local binary patterns) the algorithm runs in real-time. OAB has been seminal in introducing the tracking-by-detection paradigm to model-free object tracking.

A.45. 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 [63] for details.

A.46. Incremental Learning for Robust Visual Tracking (IVT)

Submitted by VOT Committee

The idea of the IVT tracker [58] 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.47. Multiple Instance Learning tracker (MIL)

Submitted by VOT Committee

MIL tracker [3] 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.48. ASMS

Submitted by VOT Committee

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. The ASMS [70] method address the problem of scale adaptation and present a novel theoretically justified scale estimation mechanism which relies solely on the mean-shift procedure for the Hellinger distance. The 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 color weighting and a forward-backward consistency check.

A.49. Flock of Trackers (FoT)

Submitted by VOT Committee

The Flock of Trackers (FoT) [67] 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 Flock of Trackers object motion estimate is robust if it is from local tracker motions by a combination which is insensitive to failures.

A.50. Spatio-temporal context tracker (STC)

Submitted by VOT Committee

The STC [83] is a correlation filter based tracker, which uses image intensity features. It formulates the spatio temporal relationships between the object of interest and its locally dense contexts in a Bayesian framework, which models the statistical correlation between features from the target and its surrounding regions. For fast learning and detection the Fast Fourier Transform (FFT) is adopted.

A.51. Transfer Learning Based Visual Tracking with Gaussian Processes Regression (TGPR tracker)

Submitted by VOT Committee

The TGPR tracker [19] models the probability of target appearance using Gaussian Process Regression. The observation model is learned in a semi-supervised fashion using both labeled samples from previous frames and the unlabeled samples that are tracking candidates extracted from current frame.

A.52. Normalized Cross-Correlation (NCC)

Submitted by VOT Committee

The NCC tracker is a VOT2015 baseline tracker and follows the very basic idea of tracking by searching for the best match between a static grayscale template and the image using normalized cross-correlation.

A.53. HoughTrack

Submitted by VOT Committee

HoughTrack is a tracking-by-detection approach based on the Generalized Hough-Transform. The idea of Hough-Forests is extended to the online domain and the center vote based detection and back-projection is coupled with a rough segmentation based on graph-cuts. This is in contrast to standard online learning approaches, where typically bounding-box representations with fixed aspect ratios are employed. The original authors claim that HoughTrack provides a more accurate foreground/background separation and that it can handle highly non-rigid and articulated objects. The reader is referred to [21] for details and to http://lrs.icg.tugraz.at/research/houghtrack/for code.

A.54. A kernel correlation filter tracker with Scale Adaptive and Feature Integration (SAMF)

Authors implementation. Submitted by VOT Committee

SAMF tracker is based on the idea of correlation filter-based trackers [15,27,26,5] with aim to improve the overall tracking capability. To tackle the problem of the fixed template size in kernel correlation filter tracker, an effective scale adaptive scheme is proposed. Moreover, features like HoG and colour naming are integrated together to further boost the overall tracking performance.

A.55. Adaptive Color Tracker (ACT)

Authors implementation. Submitted by VOT Committee

The Adaptive Color Tracker (ACT) [15] extends the CSK tracker [] with colour information. ACT tracker contains three improvements to CSK tracker: (i) A tempo- rally consistent scheme for updating the tracking model is applied instead of training the classifier separately on single samples, (ii) colour attributes are applied for image representation, and (iii) ACT employs a dynamically adaptive scheme for selecting the most important combinations of colours for tracking.

A.56. Discriminative Scale Space Tracker (DSST)

Authors implementation. Submitted by VOT Committee

The Discriminative Scale Space Tracker (DSST) [11] 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.57. Kernelized Correlation Filter tracker (KCF2)

Modified version of the authors implementation. Submitted by VOT Committee

This tracker is basically a Kernelized Correlation Filter [25] operating on simple HOG features. The KCF is equivalent to a Kernel Ridge Regression trained with thousands of sample patches around the object at different translations. The improvements over the previous version are multi-scale support, sub-cell peak estimation and replacing the model update by linear interpolation with a more robust update scheme.

A.58. Compressive Tracking (CT)

Implementation from authors website. Submitted by VOT Committee

The CT tracker [84] uses an appearance model based on features extracted from the multi-scale image feature space with data-independent basis. It employs non-adaptive random projections that preserve the structure of the image feature space of objects. A very sparse measurement matrix is adopted to efficiently extract the features for the appearance model. Samples of foreground and background are compressed using the same sparse measurement matrix. The tracking task is formulated as a binary classification via a naive Bayes classifier with online update in the compressed domain.

A.59. Distribution fields Tracking (DFT)

Implementation from authors website. Submitted by VOT Committee

The tacker introduces a method for building an image descriptor using distribution fields (DFs), a representation that allows smoothing the objective function without destroying information about pixel values. DFs enjoy a large basin of attraction around the global optimum compared to related descriptors. DFs also allow the representation of uncertainty about the tracked object. This helps in disregarding outliers during tracking (like occlusions or small missalignments) without modeling them explicitly.

A.60. HMMTxD

Submitted by VOT Committee

The HMMTxD [68] 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.61. L1APG

Implementation from OTB. Submitted by VOT Committee

L1APG [4] 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.62. MEEM

Implementation from authors website. Submitted by VOT Committee

MEEM [82] 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.

References

[1] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Incremental face alignment in the wild. In *Computer Vision and Pattern Recognition*, 2014.

[2] R. C. Atkinson and R. M. Shiffrin. Human memory: A proposed system and its control processes. *The psychology of learning and motivation*, 2:89–195, 1968.

[3] 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.

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

[5] O. Barnich and M. V. Droogenbroeck. Vibe: A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image Processing*, 20(6):1709–1724, 2011.

[6] L. Bertinetto, M. O., J. Valmadre, G. S., and P. Torr. The importance of estimating object extent when tracking with correlation filters. *Preprint*, 2015.

[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. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *Proceedings of the British Machine Vision Conference BMVC*, 2014.

[9] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Computer Vision and Pattern Recognition*, volume 2, pages 142–149, 2000.

[10] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(5):564–577, 2003.

[11] 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.

[12] 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.

[13] 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.

[14] P. Dollar, S. Belongie, and P. Perona. The fastest pedestrian detector in the west. In *Proceedings of the British Machine Vision Conference BMVC*, volume 2, page 7, 2010.

[15] M. Felsberg. Enhanced distribution field tracking using channel representations. In *Vis. Obj. Track. Challenge VOT2013, In conjunction with ICCV2013*, 2013.

[16] M. Felsberg, A. Berg, G. Häger, and J. Ahlberg et al. The thermal infrared visual object tracking VOT-TIR2015 challenge results. *ICCV2015 workshop proceedings, VOT2015 Workshop*, 2015.

[17] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

[18] P. Gabriel, J. Verly, J. Piater, and A. Genon. The state of the art in multiple object tracking under occlusion in video sequences. In *Proc. Advanced Concepts for Intelligent Vision Systems*, pages 166–173, 2003.

[19] J. Gao, H. Ling, W. Hu, and J. Xing. Transfer learning based visual tracking with gaussian processes regression. In *European Conference on Computer Vision*, pages 188–203, 2014.

[20] D. M. Gavrila. The visual analysis of human movement: A survey. *Comp. Vis. Image Understanding*, 73(1):82–98, 1999.

[21] M. Godec, P. M. Roth, and H. Bischof. Hough-based tracking of non-rigid objects. *Comp. Vis. Image Understanding*, 117(10):1245–1256, 2013.

[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] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting. In *Proceedings of the British Machine Vision Conference BMVC*, pages 47–56, 2006.

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

[26] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao. Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 749–758, 2015.

[27] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Trans. Systems, Man and Cybernetics, C*, 34(30):334–352, 2004.

[28] Y. Hua, K. Alahari, and C. Schmid. Online object tracking with proposal selection. In *International Conference on Computer Vision*, 2015.

[29] Z. Kalal, K. Mikolajczyk, and J. Matas. Forward-backward error: Automatic detection of tracking failures. In *Computer Vision and Pattern Recognition*, 2010.

[30] 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.

[31] M. H. Khan, M. F. Valstar, and T. P. Pridmore. Mts: A multiple temporal scale tracker handling occlusion and abrupt motion variation. In *Proceedings of the Asian Conference on Computer Vision*, pages 86–97, 2012.

[32] M. Kristan. Fast kernel density estimator. Matlab Central, 2013.

[33] M. Kristan and A. Leonardis. Multivariate online kernel density estimation. In *Computer Vision Winter Workshop*, pages 77–84, 2010.

[34] M. Kristan, J. Matas, A. Leonardis, T. Vojir, R. P. Pflugfelder, G. Fernández, G. Nebehay, F. Porikli, and L. Cehovin. A novel performance evaluation methodology for single-target trackers. *CoRR*, abs/1503.01313, 2015.

[35] M. Kristan, J. Pers, M. Perse, and S. Kovacic. Bayes spectral entropy-based measure of camera focus. In *Computer Vision Winter Workshop*, pages 155–164, February 2005.

[36] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, L. Cehovin, G. Nebehay, T. Vojir, F. G., and et al. The visual object tracking vot2014 challenge results. In *ECCV2014 Workshops, Workshop on visual object tracking challenge*, 2014.

[37] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, F. Porikli, L. Cehovin, G. Nebehay, G. Fernandez, and T. Vojir. The vot2013 challenge: overview and additional results. In *Computer Vision Winter Workshop*, 2014.

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

[39] K. Lebeda, S. Hadfield, J. Matas, and R. Bowden. Long-term tracking through failure cases. In *Proc. of ICCV VOT*, 2013.

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

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

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

[43] X. Li, W. Hu, C. Shen, Z. Zhang, A. R. Dick, and A. Van den Hengel. A survey of appearance models in visual object tracking. *arXiv:1303.4803 [cs.CV]*, 2013.

[44] Y. Li and J. Zhu. A scale adaptive kernel correlation filter tracker with feature integration. In *Proceedings of the ECCV Workshop*, pages 254–265, 2014.

[45] H. Liu, X. Yang, L. J. Latecki, and S. Yan. Dense neighborhoods on affinity graph. *International Journal of Computer Vision*, 98(1):65–82, 2012.

[46] M. Maresca and A. Petrosino. Clustering local motion estimates for robust and efficient object tracking. In *Proceedings of the Workshop on Visual Object Tracking Challenge, European Conference on Computer Vision*, 2014.

[47] M. E. Maresca and A. Petrosino. Matrioska: A multi-level approach to fast tracking by learning. In *Proc. Int. Conf. Image Analysis and Processing*, pages 419–428, 2013.

[48] T. B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Comp. Vis. Image Understanding*, 81(3):231–268, March 2001.

[49] T. B. Moeslund, A. Hilton, and V. Kruger. A survey of advances in vision-based human motion capture and analysis. *Comp. Vis. Image Understanding*, 103(2-3):90–126, November 2006.

[50] S. Moujtahid, S. Duffner, and A. Baskurt. Classifying global scene context for on-line multiple tracker selection. In *Proceedings of the British Machine Vision Conference BMVC*, 2015.

[51] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *CoRR*, 2015.

[52] K. Palaniappan, F. Bunyak, P. Kumar, I. Ersoy, S. Jaeger, K. Ganguli, A. Haridas, J. Fraser, R. Rao, and G. Seetharaman. Efficient feature extraction and likelihood fusion for vehicle tracking in low frame rate airborne video. In *IEEE*

Conference on Information Fusion (FUSION), pages 1–8, 2010.

[53] Y. Pang and H. Ling. Finding the best from the second bests – inhibiting subjective bias in evaluation of visual tracking algorithms. In *International Conference on Computer Vision*, 2013.

[54] R. Pelapur, S. Candemir, F. Bunyak, M. Poostchi, G. Seetharaman, and K. Palaniappan. Persistent target tracking using likelihood fusion in wide-area and full motion video sequences. In *IEEE Conference on Information Fusion (FUSION)*, pages 2420–2427, 2012.

[55] R. Pelapur, K. Palaniappan, and G. Seetharaman. Robust orientation and appearance adaptation for wide-area large format video object tracking. In *Proceedings of the IEEE Conference on Advanced Video and Signal based Surveillance*, pages 337–342, 2012.

[56] 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.

[57] H. Possegger, T. Mauthner, and H. Bischof. In defense of color-based model-free tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[58] 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.

[59] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Computer Vision ECCV 2014 Workshops*, pages 244–253, 2006.

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

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

[62] M. Tang and J. Feng. Multi-kernel correlation filter for visual tracking. In *International Conference on Computer Vision*, 2015.

[63] 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.

[64] 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.

[65] L. Čehovin, A. Leonardis, and M. Kristan. Visual object tracking performance measures revisited. arXiv:1502.05803 [cs.CV], 2013.

[66] T. Vojir and J. Matas. Robustifying the flock of trackers. In *Computer Vision Winter Workshop*, pages 91–97. IEEE, 2011.

[67] T. Vojir 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.

[68] T. Vojir, J. Matas, and J. Noskova. Online adaptive hidden markov model for multi-tracker fusion. *CoRR*, abs/1504.06103, 2015.

[69] T. Vojir, J. Noskova, and J. Matas. Robust scale-adaptive mean-shift for tracking. *Image Analysis*, pages 652–663, 2013.

[70] T. Vojir, J. Noskova, and J. Matas. Robust scale-adaptive mean-shift for tracking. *Pattern Recognition Letters*, 49(0):250 – 258, 2014.

[71] N. Wang, S. Li, A. Gupta, and D. Y. Yeung. Transferring rich feature hierarchies for robust visual tracking, 2015.

[72] N. Wang, J. Shi, D.-Y. Yeung, , and J. Jia. Understanding and diagnosing visual tracking systems. In *International Conference on Computer Vision*, 2015.

[73] N. Wang and D.-Y. Yeung. Ensemble-based tracking: Aggregating crowdsourced structured time series data. In *ICML*, pages 1107–1115, 2015.

[74] X. Wang, M. Valstar, B. Martinez, H. Khan, and T. Pridmore. Tracking by regression with incrementally learned cascades. In *International Conference on Computer Vision*, 2015.

[75] Y. Wu, J. Lim, and M. Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1834–1848, 2014.

[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. *IEEE-PAMI*, 2015.

[78] Xuehan-Xiong and F. D. la Torre. Supervised descent method and its application to face alignment. In *Computer Vision and Pattern Recognition*, 2013.

[79] A. Yilmaz and M. Shah. Object tracking: A survey. *Journal ACM Computing Surveys*, 38(4), 2006.

[80] 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.

[81] J. Zhang, S. Ma, and S. Sclaroff. Meem: Robust tracking via multiple experts using entropy minimization. In *Computer Vision and Pattern Recognition*, 2014.

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

[83] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang. Fast visual tracking via dense spatio-temporal context learning. In *European Conference on Computer Vision*, pages 127–141, 2014.

[84] K. Zhang, L. Zhang, and M. H. Yang. Real-time compressive tracking. In *European Conference on Computer Vision*, Lecture Notes in Computer Science, pages 864–877. Springer, 2012.

[85] G. Zhu, F. Porikli, and H. Li. Tracking randomly moving objects on edge box proposals. In *CoRR*, 2015.

[86] C. L. Zitnick and P. Dollar. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, pages 391–405, 2014.