The Visual Object Tracking VOT2016 challenge results

Matej Kristan[1], Aleš Leonardis[2], Jiři Matas[3], Michael Felsberg[4], Roman Pflugfelder[5], Luka Čehovin[1], Tomáš Vojíř[3], Gustav Häger[4], Alan Lukežič[1], Gustavo Fernández[5], Abhinav Gupta[10], Alfredo Petrosino[30], Alireza Memarmoghadam[36], Alvaro Garcia-Martin[32], Andrés Solís Montero[39], Andrea Vedaldi[40], Andreas Robinson[4], Andy J. Ma[18], Anton Varfolomieiev[23], Aydin Alatan[26], Aykut Erdem[16], Bernard Ghanem[22], Bin Liu[45], Bohyung Han[31], Brais Martinez[38], Chang-Ming Chang[34], Changsheng Xu[11], Chong Sun[12], Daijin Kim[31], Dapeng Chen[43], Dawei Du[35], Deepak Mishra[21], Dit-Yan Yeung[19], Erhan Gundogdu[7], Erkut Erdem[16], Fahad Khan[4], Fatih Porikli[6,9,29], Fei Zhao[11], Filiz Bunyak[37], Francesco Battistone[30], Gao Zhu[9], Giorgio Roffo[42], Gorthi R K Sai Subrahmanyam[21], Guilherme Bastos[33], Guna Seetharaman[27], Henry Medeiros[25], Hongdong Li[6,9], Honggang Qi[35], Horst Bischof[15], Horst Possegger[15], Huchuan Lu[12], Hyemin Lee[31], Hyeonseob Nam[28], Hyung Jin Chang[20], Isabela Drummond[33], Jack Valmadre[40], Jae-chan Jeong[13], Jae-il Cho[13], Jae-Yeong Lee[13], Jianke Zhu[44], Jiayi Feng[11], Jin Gao[11], Jin Young Choi[8], Jingjing Xiao[2], Ji-Wan Kim[13], Jiyeoup Jeong[8], João F. Henriques[40], Jochen Lang[39], Jongwon Choi[8], Jose M. Martinez[32], Junliang Xing[11], Junyu Gao[11], Kannappan Palaniappan[37], Karel Lebeda[41], Ke Gao[37], Krystian Mikolajczyk[20], Lei Qin[11], Lijun Wang[12], Longyin Wen[34], Luca Bertinetto[40], Madan kumar Rapuru[21], Mahdieh Poostchi[37], Mario Maresca[30], Martin Danelljan[4], Matthias Mueller[22], Mengdan Zhang[11], Michael Arens[14], Michel Valstar[38], Ming Tang[11], Mooyeol Baek[31], Muhammad Haris Khan[38], Naiyan Wang[19], Nana Fan[17], Noor Al-Shakarji[37], Ondrej Miksik[40], Osman Akin[16], Payman Moallem[36], Pedro Senna[33], Philip H. S. Torr[40], Pong C. Yuen[18], Qingming Huang[17,35], Rafael Martin-Nieto[32], Rengarajan Pelapur[37], Richard Bowden[41], Robert Laganière[39], Rustam Stolkin[2], Ryan Walsh[25], Sebastian B. Krah[14], Shengkun Li[34], Shengping Zhang[17], Shizeng Yao[37], Simon Hadfield[41], Simone Melzi[42], Siwei Lyu[34], Siyi Li[19], Stefan Becker[14], Stuart Golodetz[40], Sumithra Kakanuru[21], Sunglok Choi[13], Tao Hu[35], Thomas Mauthner[15], Tianzhu Zhang[11], Tony Pridmore[38], Vincenzo Santopietro[30], Weiming Hu[11], Wenbo Li[24], Wolfgang Hübner[14], Xiangyuan Lan[18], Xiaomeng Wang[38], Xin Li[17], Yang Li[44], Yiannis Demiris[20], Yifan Wang[12], Yuankai Qi[17], Zejian Yuan[43], Zexiong Cai[18], Zhan Xu[44], Zhenyu He[17], and Zhizhen Chi[12]

[1] University of Ljubljana, Slovenia
[2] University of Birmingham, England
[3] Czech Technical University, Czech Republic
[4] Linköping University, Sweden
[5] Austrian Institute of Technology, Austria
[6] ARC Centre of Excellence for Robotic Vision, Australia
[7] Aselsan Research Center, Turkey
[8] ASRI, South Korea

[9] Australian National University, Australia
[10] Carnegie Mellon University, USA
[11] Chinese Academy of Sciences, China
[12] Dalian University of Technology, China
[13] Electronics and Telecommunications Research Institute, South Korea
[14] Fraunhofer IOSB, Germany
[15] Graz University of Technology, Austria
[16] Hacettepe University, Turkey
[17] Harbin Institute of Technology, China
[18] Hong Kong Baptist University, China
[19] Hong Kong University of Science and Technology, China
[20] Imperial College London, England
[21] Indian Institute of Space Science and Technology, India
[22] KAUST, Saudi Arabia
[23] Kyiv Polytechnic Institute, Ukraine
[24] Lehigh University, USA
[25] Marquette University, USA
[26] Middle East Technical University, Turkey
[27] Naval Research Lab, USA
[28] NAVER Corp., South Korea
[29] Data61/CSIRO, Australia
[30] Parthenope University of Naples, Italy
[31] POSTECH, South Korea
[32] Universidad Autónoma de Madrid, Spain
[33] Universidade Federal de Itajubá, Brazil
[34] University at Albany, USA
[35] University of Chinese Academy of Sciences, China
[36] University of Isfahan, Iran
[37] University of Missouri, USA
[38] University of Nottingham, England
[39] University of Ottawa, Canada
[40] University of Oxford, England
[41] University of Surrey, England
[42] University of Verona, Italy
[43] Xi'an Jiaotong University, China
[44] Zhejiang University, China
[45] Moshanghua Tech Co., China

Abstract. The Visual Object Tracking challenge VOT2016 aims at comparing short-term single-object visual trackers that do not apply pre-learned models of object appearance. Results of 70 trackers are presented, with a large number of trackers being published at major computer vision conferences and journals in the recent years. The number of tested state-of-the-art trackers makes the VOT 2016 the largest and most challenging benchmark on short-term tracking to date. For each participating tracker, a short description is provided in the Appendix. The VOT2016 goes beyond its predecessors by (i) introducing a new semi-automatic ground truth bounding box annotation methodology and (ii) extending the evaluation system with the no-reset experiment. The dataset, the evaluation kit as well as the results are publicly available at the challenge website[46] [47].

Keywords: Performance evaluation, short-term single-object trackers, VOT

1 Introduction

Visual tracking remains a highly popular research area of computer vision, with the number of motion and tracking papers published at high profile conferences exceeding 40 papers annually. The significant activity in the field over last two decades is reflected in the abundance of review papers [1–9]. In response to the high number of publications, several initiatives emerged to establish a common ground for tracking performance evaluation. The earliest and most influential is the PETS [10], which is the longest lasting initiative that proposed frameworks for performance evaluation in relation to surveillance systems applications. Other frameworks have been presented since with focus on surveillance systems and event detection, (e.g., CAVIAR[48], i-LIDS [49], ETISEO[50]), change detection [11], sports analytics (e.g., CVBASE[51]), faces (e.g. FERET [12] and [13]), long-term tracking [52] and the multiple target tracking [14, 15][53].

In 2013 the Visual object tracking, VOT, initiative was established to address performance evaluation for short-term visual object trackers. The initiative aims at establishing datasets, performance evaluation measures and toolkits as well as creating a platform for discussing evaluation-related issues. Since its emergence in 2013, three workshops and challenges have been carried out in conjunction with the ICCV2013 (VOT2013 [16]), ECCV2014 (VOT2014 [17]) and ICCV2015 (VOT2015 [18]). This paper discusses the VOT2016 challenge, organized in conjunction with the ECCV2016 Visual object tracking workshop, and the results obtained. Like VOT2013, VOT2014 and VOT2015, the VOT2016 challenge considers single-camera, single-target, model-free, causal trackers, applied to short-term tracking. The *model-free* property means that the only training example is provided by the bounding box in the first frame. The *short-term* tracking means that trackers are assumed not to be capable of performing successful re-detection after the target is lost and they are therefore reset after such event. 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,

[46] http://votchallenge.net

[47] This version of the results paper includes several corrections of errors discovered after the submission to VOT workshop and additional comments.

[48] http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1

[49] http://www.homeoffice.gov.uk/science-research/hosdb/i-lids

[50] http://www-sop.inria.fr/orion/ETISEO

[51] http://vision.fe.uni-lj.si/cvbase06/

[52] http://www.micc.unifi.it/LTDT2014/

[53] https://motchallenge.net

we overview the most closely related work and point out the contributions of VOT2016.

1.1 Related work

Several works that focus on performance evaluation in short-term visual object tracking [16, 17, 19–24] have been published in 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) [21], the 'Amsterdam Library of Ordinary Videos' (ALOV) [22] and the 'Visual object tracking challenge' (VOT) [16–18].

Performance measures The OTB- and ALOV-related methodologies, like [21, 22, 24, 25], evaluate a tracker by initializing it on the first frame and letting it run until the end of the sequence, while the VOT-related methodologies [16–18, 20, 19] reset the tracker once it drifts off the target. Performance is evaluated in all of these approaches by overlaps between the bounding boxes predicted from the tracker with the ground truth bounding boxes. The OTB and ALOV initially considered performance evaluation based on object center estimation as well, but as shown in [26], the center-based measures are highly brittle and overlap-based measures should be preferred. The ALOV measures the tracking performance as the F-measure at 0.5 overlap threshold and a similar measure was proposed by OTB. Recently, it was demonstrated in [19] that such threshold is over-restrictive, since an overlap below 0.5 does not clearly indicate a tracking failure in practice. 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 developed an ad-hoc performance measure computed as the area under the curve in this plot. This measure remains one of the most widely used measures in tracking papers. It was later analytically proven by [26, 20] that the ad-hoc measure is equivalent to the average overlap (AO), which can be computed directly without intermediate success plots, giving the measure a clear interpretation. An analytical model was recently proposed [19] to study the average overlap measures with and without resets in terms of tracking accuracy estimator. The analysis showed that the no-reset AO measures are biased estimators with large variance while the VOT reset-based average overlap drastically reduces the bias and variance and is not hampered by the varying sequence lengths in the dataset.

Čehovin et al. [26, 20] provided a highly detailed theoretical and experimental analysis of a number of the popular performance measures. Based on that analysis, the VOT2013 [16] selected the average overlap with resets and number of tracking failures as their main performance criteria, measuring geometric accuracy and robustness respectively. The VOT2013 introduced a ranking-based methodology that accounted for statistical significance of the results, which was extended with the tests of practical differences in the VOT2014 [17]. The notion of practical differences is unique to the VOT challenges and relates to the uncertainty of the ground truth annotation. The VOT ranking methodology treats each sequence as a competition among the trackers. Trackers are ranked on each sequence and ranks are averaged over all sequences. This is called the sequence-normalized ranking. An alternative is sequence-pooled ranking [19], which ranks the average performance on all sequences. Accuracy-robustness ranking plots were proposed [16] to visualize the results. A drawback of the AR-rank plots is that they do not show the absolute performance. In VOT2015 [18], the AR-raw plots from [20, 19] were adopted to show the absolute average performance. The VOT2013 [16] and VOT2014 [17] selected the winner of the challenge by averaging the accuracy and robustness ranks, meaning that the accuracy and robustness were treated as equivalent "competitions". A high average rank means that a tracker was well-performing in accuracy as well as robustness relative to the other trackers. While ranking converts 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 [18] introduced a new measure called the expected average overlap (EAO) that combines the raw values of per-frame accuracies and failures in a principled manner and has a clear practical interpretation. The EAO measures the expected no-reset overlap of a tracker run on a short-term sequence. In principle, this measure reflects the same property as the AO [21] measure, but, since it is computed from the VOT reset-based experiment, it does not suffer from the large variance and has a clear definition of what the short-term sequence means. VOT2014 [17] pointed out that speed is an important factor in many applications and introduced a speed measure called the equivalent filter operations (EFO) that partially accounts for the speed of computer used for tracker analysis.

The VOT2015 [18] noted that state-of-the-art performance is often misinterpreted as requiring a tracker to *score as number one* on a benchmark, often leading authors to creatively select sequences and experiments and omit related trackers in scientific papers to reach the apparent *top performance*. To expose this misconception, the VOT2015 computed the average performance of the participating trackers that were published at top recent conferences. This value is called the VOT2015 state-of-the-art bound and any tracker exceeding this performance on the VOT2015 benchmark should be considered state-of-the-art according to the VOT standards.

Datasets. The current trend in computer vision datasets construction appears to be focused on increasing the number of sequences in the datasets [27, 23, 24, 22, 25], but often much less attention is being paid to the quality of its content 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 even though they may occur only at a small number of frames in a video. The dataset size is commonly assumed to imply quality. In contrast, the VOT2013 [16] argued that large datasets do not necessarily imply diversity or richness in attributes. Over the last three years, the VOT has developed a methodology that automatically constructs a moderately sized dataset from a large pool of sequences. The uniqueness of this methodology is that it explicitly optimizes diversity in visual attributes while focusing on sequences which are difficult to track. In addition, the sequences in the VOT datasets are per-frame annotated by visual attributes, which is in stark contrast to the related datasets that apply global annotation. It was recently shown [19] 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 misannotations.

Most closely related works to the work described in this paper are the recent VOT2013 [16], VOT2014 [17] and VOT2015 [18] 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, per-frame annotated datasets and state-of-the-art performance evaluation methodology for in-depth tracker analysis from several performance aspects. The results were published in joint papers ([16], [17], [18]) of which the VOT2015 [18] paper alone exceeded 120 coauthors. 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[54]. The advances proposed by VOT have also influenced the development of related methodologies and benchmark papers like [23–25].

1.2 The VOT2016 challenge

VOT2016 follows VOT2015 challenge and considers the same class of trackers. The dataset and evaluation toolkit are provided by the VOT2016 organizers. The evaluation kit records the output bounding boxes from the tracker, and if it detects tracking failure, re-initializes the tracker. The authors participating in the challenge were required to integrate their tracker into the VOT2016 evaluation kit, which automatically performed a standardized experiment. The results were analyzed by the VOT2016 evaluation methodology. In addition to the VOT reset-based experiment, the toolkit conducted the main OTB [21] experiment in which a tracker is initialized in the first frame and left to track until the end of the sequence without resetting. The performance on this experiment is evaluated by the average overlap measure [21].

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. The organizers of VOT2016 were allowed to participate in the challenge, but did not compete for the winner of VOT2016 challenge title. Further

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

details are available from the challenge homepage[55].

The advances of VOT2016 over VOT2013, VOT2014 and VOT2015 are the following: (i) The ground truth bounding boxes in the VOT2015 dataset have been re-annotated. Each frame in the VOT2015 dataset has been manually per-pixel segmented and bounding boxes have been automatically generated from the segmentation masks. (ii) A new methodology was developed for automatic placement of a bounding box by optimizing a well defined cost function on manually per-pixel segmented images. (iii) The evaluation system from VOT2015 [18] is extended and the bounding box overlap estimation is constrained to image region. The toolkit now supports the OTB [21] no-reset experiment and their main performance measures. (iv) The VOT2015 introduced a second sub-challenge VOT-TIR2015 held under the VOT umbrella which deals with tracking in infrared and thermal imagery [28]. Similarly, the VOT2016 is accompanied with VOT-TIR2016, and the challenge and its results are discussed in a separate paper submitted to the VOT2016 workshop [29].

The remainder of this paper is structured as follows. In Section 2, the new dataset is introduced. The methodology is outlined in Section 3, the main results are discussed in Section 4 and conclusions are drawn in Section 5.

2 The VOT2016 dataset

VOT2013 [16] and VOT2014 [17] 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 [18], the methodology was extended into a fully automated sequence selection with the selection process focusing on challenging sequences. The methodology was applied in VOT2015 [18] to produce a highly challenging VOT2015 dataset.

Results of VOT2015 showed that the dataset was not saturated and the same sequences were used for VOT2016. The VOT2016 dataset thus contains all 60 sequences from VOT2015, where each sequence is per-frame annotated by the following 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.

In VOT2015, the rotated bounding boxes have been manually placed in each frame of the sequence by experts and cross checked by several groups for quality control. To enforce a consistency, the annotation rules have been specified. Nevertheless, we have noticed that human annotators have difficulty following the annotation rules, which makes it impossible to guarantee annotation consistency. For this reason, we have developed a novel approach for dataset annotation. The new approach takes a pixel-wise segmentation of the tracked object and places a bounding box by optimizing a well-defined cost function. In the following, Section 2.1 discusses per-frame segmentation mask construction and the new bounding box generation approach is presented in Section 2.2.

[55] http://www.votchallenge.net/vot2016/participation.html

2.1 Producing per-frame segmentation masks

The per-frame segmentations were provided for VOT by a research group that applied an interactive annotation tool designed by VOT[56] for manual segmentation mask construction. The tool applies Grabcut [30] object segmentation on each frame. The color model is initialized from the VOT2015 ground truth bounding box (first frame) or propagated from the final segmentation in the previous frame. The user can interactively add foreground or background examples to improve the segmentation. Examples of the object segmentations are illustrated in Fig. 1.

2.2 Automatic bounding box computation

The final ground truth bounding box for VOT2016 was automatically computed on each frame from the corresponding segmentation mask. We have designed the following cost function and constraints to reflect the requirement that the bounding box should capture object pixels with minimal amount of background pixels:

$$\begin{aligned} &\arg\min_{\mathbf{b}} \{ C(\mathbf{b}) = \alpha \sum_{\mathbf{x} \notin A(\mathbf{b})} [\mathrm{M}(\mathbf{x}) > 0] + \sum_{\mathbf{x} \in A(\mathbf{b})} [\mathrm{M}(\mathbf{x}) == 0] \}, \\ \text{subject to} \quad &\frac{1}{\mathrm{M}_f} \sum_{\mathbf{x} \notin A(\mathbf{b})} [\mathrm{M}(\mathbf{x}) > 0] < \Theta_f, \frac{1}{|A(\mathbf{b})|} \sum_{\mathbf{x} \in A(\mathbf{b})} [\mathrm{M}(\mathbf{x}) == 0] < \Theta_b, \end{aligned} \tag{1}$$

where $\mathbf{b}$ is the vector of bounding box parameters (center, width, height, rotation), $A(\mathbf{b})$ is the corresponding bounding box, M is the segmentation mask which is non-zero for object pixels, $[\cdot]$ is an operator which returns 1 iff the statement in the operator is true and 0 otherwise, M_f is number of object pixels and $|\cdot|$ denotes the cardinality. An intuitive interpretation of the cost function is that we want to find a bounding box which minimizes a weighted sum of the number of object pixels outside of the bounding box and the number of background pixels inside the bounding box, with percentage of excluded object pixels and included background pixels constrained by Θ_f and Θ_b, respectively. The cost (1) was optimized by Interior Point [31] optimization, with three starting points: (i) the VOT2015 ground truth bounding box, (ii) a minimal axis-align bounding box containing all object pixels and (iii) a minimal rotated bounding box containing all object pixels. In case a solution satisfying the constraints was not found, a relaxed unconstrained BFGS Quasi-Newton method [32] was applied. Such cases occurred at highly articulated objects. The bounding box tightness is controlled by parameter α. Several values, i.e., $\alpha = \{1, 4, 7, 10\}$, were tested on randomly chosen sequences and the final value $\alpha = 4$ was selected since its bounding boxes were visually assessed to be the best-fitting. The constraints

[56] https://github.com/vojirt/grabcut_annotation_tool

$\Theta_f = 0.1$ and $\Theta_b = 0.4$ were set to the values defined in previous VOT challenges. Examples of the automatically estimated ground truth bounding boxes are shown in Figure 1.

All bounding boxes were visually verified to avoid poor fits due to potential segmentation errors. We identified 12% of such cases and reverted to the VOT2015 ground truth for those. During the challenge, the community identified four frames where the new ground truth is incorrect and those errors were not caught by the verification. In these cases, the bounding box within the image bounds was properly estimated, but extended out of image bounds disproportionally. These errors will be corrected in the next version of the dataset and we checked, during result processing, that it did not significantly influence the challenge results. Table 1 summarizes the comparison of the VOT2016 automatic ground truth with the VOT2015 in terms of portions of object and background pixels inside the bounding boxes. The statistics were computed over the whole dataset excluding the 12% of frames where the segmentation was marked as incorrect. The VOT2016 ground truth improves in all aspects over the VOT2015. It is interesting to note that the average overlap between VOT2015 and VOT2016 ground truth is 0.74.

	%frames	#frames	fg-out	bg-in	Avg. overlap	#opt. failures
automatic GT	88%	18875	0.04	0.27	0.74	2597
VOT2015 GT	100%	21455	0.06	0.37	—	—

Table 1. The first two columns shows the percentage and number of frames annotated by the VOT2016 and VOT2015 methodology, respectively. The *fg-out* and *bg-in* denote the average percentage of object pixels outside and percentage of background pixels inside the GT, respectively. The average overlap with the VOT2015 annotations is denoted by *Avg. overlap*, while the *#opt. failures* denotes the number of frames in which the algorithm switched from constrained to unconstrained optimization.

2.3 Uncertainty of optimal bounding box fits

The cost function described in Section 2.2 avoids subjectivity of manual bounding box fitting, but does not specify how well constrained the solution is. The level of constraint strength can be expressed in terms of the average overlap of bounding boxes in the vicinity of the cost function (1) optimum, where we define the vicinity as a variation of bounding boxes within a maximum increase of the cost function around the optimum. The relative maximum increase of the cost function, i.e., the increase divided by the optimal value, is related to the annotation uncertainty in the per-pixels segmentation masks and can be estimated by the following rule-of thumb.

Let S_f denote the number of object pixels outside and let S_b denote the number of background pixels inside the bounding box. According to the central limit theorem, we can assume that S_f and S_b are normally distributed, i.e.,

$\mathcal{N}(\mu_f, \sigma_f^2)$ and $\mathcal{N}(\mu_b, \sigma_b^2)$, since they are sums of many random variables (per-pixel labels). In this respect, the value of the cost function C in (1) can be treated as a random variable as well and it is easy to show the following relation $\mathrm{var}(C) = \sigma_c^2 = \alpha^2\sigma_f^2 + \sigma_b^2$. The variance of the cost function is implicitly affected by the per-pixel annotation uncertainty through the variances σ_f^2 and σ_b^2. Assume that at most $x\mu_f$ and $x\mu_b$ pixels are incorrectly labeled on average. Since nearly all variation in a Gaussian is captured by three standard deviations, the variances are $\sigma_f^2 = (x\mu_f/3)^2$ and $\sigma_b^2 = (x\mu_b/3)^2$. Applying the three-sigma rule to the variance of the cost C, and using the definition of the foreground and background variances, gives an estimator of the maximal cost function change $\Delta_c = 3\sigma_c = x\sqrt{\alpha^2\mu_f^2 + \mu_b^2}$. Our goal is to estimate the maximal relative cost function change in the vicinity of its optimum C_{opt}, i.e., $r_{\max} = \frac{\Delta_c}{C_{\text{opt}}}$. Using the definition of the maximal change Δ_c, the rule of thumb for the maximal relative change is

$$r_{\max} = \frac{x\sqrt{\alpha^2\mu_f^2 + \mu_b^2}}{\mu_f + \mu_b}. \tag{2}$$

3 Performance evaluation methodology

Since VOT2015 [18], three primary measures are used to analyze tracking performance: accuracy (A), robustness (R) and expected average overlap (AEO). In the following these are briefly overviewed and we refer to [18–20] for further details. The VOT challenges apply a reset-based methodology. Whenever a tracker predicts a bounding box with zero overlap with the ground truth, a failure is detected and the tracker is re-initialized five frames after the failure. Čehovin et al. [20] identified two highly interpretable weakly correlated performance measures to analyze tracking behavior in reset-based experiments: (i) accuracy and (ii) robustness. The accuracy is the average overlap between the predicted and ground truth bounding boxes during successful tracking periods. On the other hand, the robustness measures how many times the tracker loses the target (fails) during tracking. The potential bias due to resets is reduced by ignoring ten frames after re-initialization in the accuracy measure, which is quite a conservative margin [19]. Stochastic trackers are run 15 times on each sequence to obtain reduce the variance of their results. 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. The third primary measure, called the expected average overlap (EAO), is an estimator of the average overlap a tracker is expected to attain on a large collection of short-term sequences with the same visual properties as the given dataset. This measure addresses the problem of increased variance and bias of AO [21] measure due to variable sequence lengths on practical datasets. Please see [18] for further details on the average expected overlap measure.

We adopt the VOT2015 ranking methodology that accounts for statistical significance and practical differences to rank trackers separately with respect to the accuracy and robustness ([18, 19]). Apart from accuracy, robustness and expected overlaps, 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 [17] 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 VOT2016.

In addition to the standard reset-based VOT experiment, the VOT2016 toolkit carried out the OTB [21] no-reset experiment. The tracking performance on this experiment was evaluated by the primary OTB measure, average overlap (AO).

4 Analysis and results

4.1 Practical difference estimation

As noted in Section 2.3, the variation in the per-pixel segmentation masks introduces the uncertainty of the optimally fitted ground truth bounding boxes. We expressed this uncertainty as the average overlap of the optimal bounding box with the bounding boxes sampled in vicinity of the optimum, which is implicitly defined as the maximal allowed cost increase. Assuming that on average, at most 10% of pixels might be incorrectly assigned in the object mask, the rule of thumb (2) estimates an increase of cost function by at most 7%. The average overlap specified in this way was used in the VOT2016 as an estimate of the per-sequence practical differences.

The following approach was thus applied to estimate the practical difference thresholds. Thirty uniformly dispersed frames were selected per sequence. For each frame a set of 3125 ground truth bounding box perturbations were generated by varying the ground truth regions by $\mathbf{\Delta_b} = [\Delta_x, \Delta_y, \Delta_w, \Delta_h, \Delta_\Theta]$, where all Δ are sampled uniformly (5 samples) from ranges ±5% of ground truth width (height) for $\Delta_x(\Delta_y)$, ±10% of ground truth width (height) for $\Delta_w(\Delta_h)$ and $\pm 4^\circ$ for Δ_Θ. These ranges were chosen such that the cost function is well explored near the optimal solution and the amount of bounding box perturbations can be computed reasonably fast. The examples of bounding boxes generated in this way are shown in Figure 1. An average overlap was computed between the ground truth bounding box and the bounding boxes that did not exceed the optimal cost value by more than 7%. The average of the average overlaps computed in thirty frames was taken as the estimate of the practical difference threshold for a given sequence. The boxplots in Figure 1 visualize the distributions of average overlaps with respect to the sequences.

4.2 Trackers submitted

Together 48 valid entries have been submitted to the VOT2016 challenge. Each submission included the binaries/source code that was used by the VOT2016

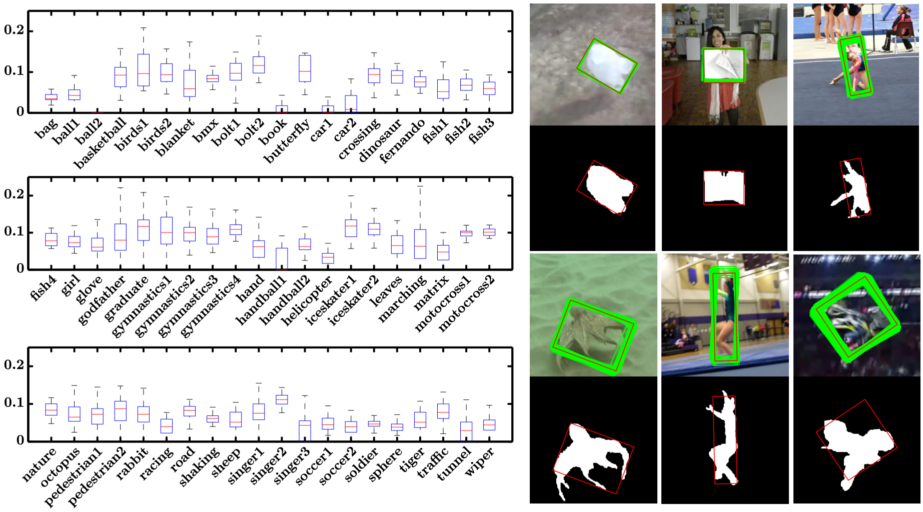

Fig. 1. Box plots of per-sequence overlap dispersion at 7% cost change (left), and examples of such bounding boxes (right). The optimal bounding box is depicted in red, while the 7% cost change bounding boxes are shown in green.

committee for results verification. The VOT2016 committee and associates additionally contributed 22 baseline trackers. For these, the default parameters were selected, or, when not available, were set to reasonable values. Thus in total 70 trackers were tested in the VOT2016 challenge. In the following we briefly overview the entries and provide the references to original papers in the Appendix A where available.

Eight trackers were based on convolutional neural networks architecture for target localization, MLDF (A.19), SiamFC-R (A.23), SiamFC-A (A.25), TCNN (A.44), DNT (A.41), SO-DLT (A.8), MDNet-N (A.46) and SSAT (A.12), where MDNet-N (A.46) and SSAT (A.12) were extensions of the VOT2015 winner MDNet [33]. Thirteen trackers were variations of correlation filters, SRDCF (A.58), SWCF (A.3), FCF (A.7), GCF (A.36), ART-DSST (A.45), DSST2014 (A.50), SMACF (A.14), STC (A.66), DFST (A.39), KCF2014 (A.53), SAMF2014 (A.54), OEST (A.31) and sKCF (A.40). Seven trackers combined correlation filter outputs with color, Staple (A.28), Staple+ (A.22), MvCFT (A.15), NSAMF (A.21), SSKCF (A.27), ACT (A.56) and ColorKCF (A.29), and six trackers applied CNN features in the correlation filters, deepMKCF (A.16), HCF (A.60), DDC (A.17), DeepSRDCF (A.57), C-COT (A.26), RFD-CF2 (A.47). Two trackers were based on structured SVM, Struck2011 (A.55) and EBT (A.2) which applied region proposals as well. Three trackers were based on purely on color, DAT (A.5), SRBT (A.34) and ASMS (A.49) and one tracker was based on fusion of basic features LoFT-Lite (A.38). One tracker was based on subspace learning, IVT (A.64), one tracker was based on boosting, MIL (A.68), one tracker was based on complex cells approach, CCCT (A.20), one on distributed fields, DFT (A.59), one tracker was based on Gaussian process regressors, TGPR (A.67), and one tracker was the basic normalized cross correlation tracker NCC (A.61). Nineteen submissions can be categorized as part-based trackers, DPCF (A.1), LT-FLO (A.43), SHCT (A.24), GGTv2 (A.18), MatFlow (A.10), Matrioska (A.11), CDTT (A.13), BST (A.30), TRIC-track (A.32), DPT (A.35), SMPR (A.48), CMT (A.70), HT (A.65), LGT (A.62), ANT (A.63), FoT (A.51), FCT (A.37), FT (A.69), and BDF (A.9). Several submissions were based on combination of base trackers, PKLTF (A.4), MAD (A.6), CTF (A.33), SCT (A.42) and HMMTxD (A.52).

4.3 Results

The results are summarized in sequence-pooled and attribute-normalized AR-raw plots in Figure 2. 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. The AR-raw plots were constructed in similar fashion. The expected average overlap curves and expected average overlap scores are shown in Figure 3. The raw values for the sequence-pooled results and the average overlap scores are also given in Table 2.

The top ten trackers come from various classes. The TCNN (A.44), SSAT (A.12), MLDF (A.19) and DNT (A.41) are derived from CNNs, the C-COT (A.26), DDC (A.17), Staple (A.28) and Staple+ (A.22) are variations of correlation filters with more or less complex features, the EBT (A.2) is structured SVM edge-feature tracker, while the SRBT (A.34) is a color-based saliency detection tracker. The following five trackers appear either very robust or very accurate: C-COT (A.26), TCNN (A.44), SSAT (A.12), MLDF (A.19) and EBT (A.2). The C-COT (A.26) is a new correlation filter which uses a large variety of state-of-the-art features, i.e., HOG [34], color-names [35] and the vgg-m-2048 CNN features pretrained on Imagenet [57]. The TCNN (A.44) samples target locations and scores them by several CNNs, which are organized into a tree structure for efficiency and are evolved/pruned during tracking. SSAT (A.12) is based on MDNet [33], applies segmentation and scale regression, followed by occlusion detection to prevent training from corrupt samples. The MLDF (A.19) applies a pre-trained VGG network [36] which is followed by another, adaptive, network with Euclidean loss to regress to target position. According to the EAO measure, the top performing tracker was C-COT (A.26) [37], closely followed by the TCNN (A.44). Detailed analysis of the AR-raw plots shows that the TCNN (A.44) produced slightly greater average overlap (0.55) than C-COT (A.26) (0.54), but failed slightly more often (by six failures). The best overlap was achieved by SSAT (A.12) (0.58), which might be attributed to the combination of segmentation and scale regression this tracker applies. The smallest number of failures achieved the MLDF (A.19), which outperformed C-COT (A.26) by a single failure, but obtained a much smaller overlap (0.49).

[57] http://www.vlfeat.org/matconvnet/

Under the VOT strict ranking protocol, the SSAT (A.12) is ranked number one in accuracy, meaning the overlap was clearly higher than for any other tracker. The second-best ranked tracker in accuracy is Staple+ (A.22) and several trackers share third rank SHCT (A.24), deepMKCF (A.16), FCF (A.7), meaning that the null hypothesis of difference between these trackers in accuracy could not be rejected. In terms of robustness, trackers MDNet-N (A.46), C-COT (A.26), MLDF (A.19) and EBT (A.2) share the first place, which means that the null hypothesis of difference in their robustness could not be rejected. The second and third ranks in robustness are occupied by TCNN (A.44) and SSAT (A.12), respectively.

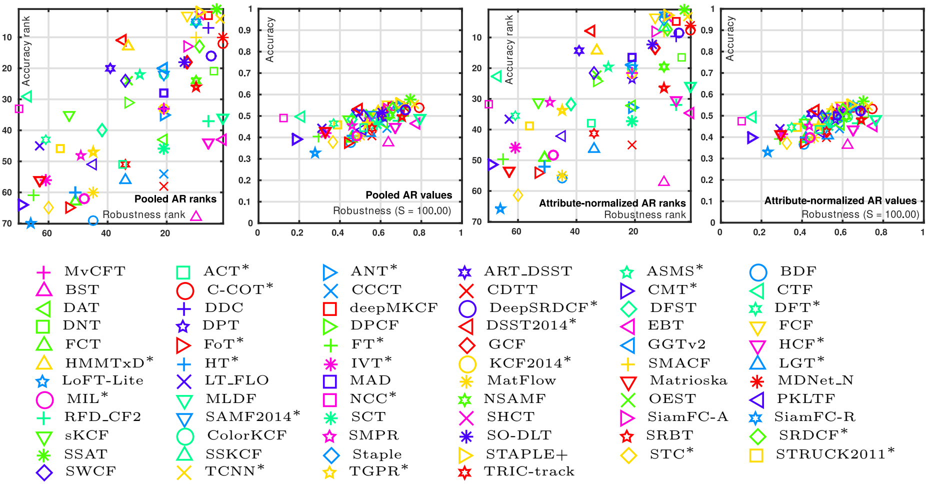

Fig. 2. The AR-rank plots and AR-raw plots generated by sequence pooling (left) and attribute normalization (right).

It is worth pointing out some EAO results appear to contradict AR-raw measures at a first glance. For example, the Staple obtains a higher EAO measure than Staple+, even though the Staple achieves a slightly better average accuracy and in fact improves on Staple by two failures, indicating a greater robustness. The reason is that the failures early on in the sequences globally contribute more to penalty than the failures that occur at the end of the sequence (see [18] for definition of EAO). For example, if a tracker fails once and is re-initialized in the sequence, it generates two sub-sequences for computing the overlap measure at sequence length N. The first sub-sequence ends with the failure and will contribute to any sequence length N since zero overlaps are added after the failure. But the second sub-sequence ends with the sequence end and zeros cannot be added after that point. Thus the second sub-sequence only contributes to the overlap computations for sequence lengths N smaller than its length. This means

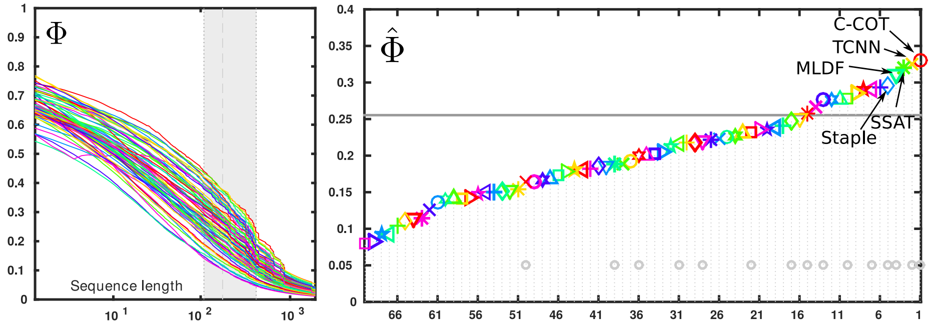

Fig. 3. Expected average overlap curve (left) and expected average overlap graph (right) with trackers ranked from right to left. The right-most tracker is the top-performing according to the VOT2016 expected average overlap values. See Figure 2 for legend. The dashed horizontal line denotes the average performance of fifteen state-of-the-art trackers published in 2015 and 2016 at major computer vision venues. These trackers are denoted by gray circle in the bottom part of the graph.

that re-inits very close to the sequence end (tens of frames) do not affect the EAO.

Note that the trackers that are usually used as baselines, i.e., MIL (A.68), and IVT (A.64) are positioned at the lower part of the AR-plots and the EAO ranks, which indicates that majority of submitted trackers are considered state-of-the-art. In fact, fifteen tested trackers have been recently (in 2015 and 2016) published at major computer vision conferences and journals. These trackers are indicated in Figure 3, along with the average state-of-the-art performance computed from the average performance of these trackers, which constitutes a very strict VOT2016 state-of-the-art bound. Approximately 22% of submitted trackers exceed this bound.

	Tracker	EAO	A	R	Ar	Rr	AO	EFO	Impl.
1.	○ C-COT	**0.331**	0.539	*0.238*	11.000	*2.000*	0.469	0.507	D M
2.	× TCNN	*0.325*	0.554	0.268	**1.000**	4.000	*0.485*	1.049	S M
3.	✳ SSAT	0.321	**0.577**	0.291	**1.000**	5.000	**0.515**	0.475	S M
4.	▽ MLDF	0.311	0.490	**0.233**	37.000	**1.000**	0.428	1.483	D M
5.	◇ Staple	0.295	0.544	0.378	8.000	14.000	0.388	11.114	D M
6.	+ DDC	0.293	0.541	0.345	8.000	7.000	0.391	0.198	D M
7.	◁ EBT	0.291	0.465	0.252	44.000	3.000	0.370	3.011	D C
8.	☆ SRBT	0.290	0.496	0.350	32.000	7.000	0.333	3.688	D M
9.	▷ STAPLE+	0.286	*0.557*	0.368	**1.000**	11.000	0.392	44.765	D M
10.	□ DNT	0.278	0.515	0.329	21.000	7.000	0.427	1.127	S M
11.	△ SSKCF	0.277	0.547	0.373	7.000	12.000	0.391	29.153	D C
12.	✡ SiamFC-R	0.277	0.549	0.382	**1.000**	15.000	0.421	5.444	D M
13.	○ DeepSRDCF*	0.276	0.528	0.326	17.000	6.000	0.427	0.380	S C
14.	× SHCT	0.266	0.547	0.396	*6.000*	16.000	0.392	0.711	D M
15.	✳ MDNet_N	0.257	0.541	0.337	11.000	7.000	0.457	0.534	S M

16.	▽	FCF	0.251	0.554	0.457	**1.000**	23.000	0.419	1.929	D	M
17.	◇	SRDCF*	0.247	0.535	0.419	11.000	18.000	0.397	1.990	S	C
18.	+	RFD_CF2	0.241	0.477	0.373	41.000	12.000	0.352	0.896	D	M
19.	◁	GGTv2	0.238	0.515	0.471	21.000	26.000	0.433	0.357	S	M
20.	☆	DPT	0.236	0.492	0.489	34.000	28.000	0.334	4.111	D	M
21.	▷	SiamFC-A	0.235	0.532	0.461	16.000	25.000	0.399	9.213	D	M
22.	□	deepMKCF	0.232	0.543	0.422	8.000	19.000	0.409	1.237	S	M
23.	△	HMMTxD	0.231	0.519	0.531	17.000	35.000	0.369	3.619	D	C
24.	✡	NSAMF	0.227	0.502	0.438	21.000	19.000	0.354	9.677	D	C
25.	○	ColorKCF	0.226	0.503	0.443	21.000	19.000	0.347	91.460	D	C
26.	×	CCCT	0.223	0.442	0.461	53.000	24.000	0.308	9.828	D	M
27.	✱	SO-DLT	0.221	0.516	0.499	17.000	31.000	0.372	0.576	S	M
28.	▽	HCF*	0.220	0.450	0.396	49.000	17.000	0.374	1.057	D	C
29.	◇	GCF	0.218	0.520	0.485	17.000	28.000	0.348	5.904	D	M
30.	+	SMACF	0.218	0.535	0.499	14.000	28.000	0.367	5.786	D	M
31.	◁	DAT	0.217	0.468	0.480	41.000	27.000	0.309	18.983	D	M
32.	☆	ASMS	0.212	0.503	0.522	21.000	34.000	0.330	82.577	D	C
33.	▷	ANT*	0.204	0.483	0.513	37.000	33.000	0.303	7.171	D	M
34.	□	MAD	0.202	0.497	0.503	29.000	32.000	0.328	8.954	D	C
35.	△	BST	0.200	0.376	0.447	66.000	19.000	0.235	13.608	S	C
36.	✡	TRIC-track	0.200	0.443	0.583	53.000	38.000	0.269	0.335	S	M
37.	○	KCF2014	0.192	0.489	0.569	37.000	37.000	0.301	21.788	D	M
38.	×	OEST	0.188	0.510	0.601	21.000	38.000	0.370	0.170	D	M
39.	✱	SCT	0.188	0.462	0.545	44.000	36.000	0.283	11.131	D	M
40.	▽	SAMF2014	0.186	0.507	0.587	21.000	38.000	0.350	4.099	D	M
41.	◇	SWCF	0.185	0.500	0.662	29.000	46.000	0.293	7.722	D	M
42.	+	MvCFT	0.182	0.491	0.606	34.000	42.000	0.308	5.194	D	M
43.	◁	DSST2014	0.181	0.533	0.704	15.000	50.000	0.325	12.747	D	M
44.	☆	TGPR*	0.181	0.460	0.629	44.000	44.000	0.270	0.318	D	M
45.	▷	DPCF	0.179	0.492	0.615	33.000	44.000	0.306	2.669	D	M
46.	□	ACT	0.173	0.446	0.662	49.000	47.000	0.281	9.840	S	C
47.	△	LGT*	0.168	0.420	0.605	57.000	42.000	0.271	3.775	S	M
48.	✡	ART_DSST	0.167	0.515	0.732	21.000	50.000	0.306	8.451	D	M
49.	○	MIL*	0.165	0.407	0.727	61.000	50.000	0.201	7.678	S	C
50.	×	CDTT	0.164	0.409	0.583	59.000	38.000	0.263	13.398	D	M
51.	✱	MatFlow	0.155	0.408	0.694	61.000	49.000	0.231	59.640	D	C
52.	▽	sKCF	0.153	0.485	0.816	37.000	57.000	0.301	91.061	D	C
53.	◇	DFST	0.151	0.483	0.778	41.000	50.000	0.315	3.374	D	M
54.	+	HT*	0.150	0.409	0.771	59.000	50.000	0.198	1.181	S	C
55.	◁	PKLTF	0.150	0.437	0.671	55.000	48.000	0.278	33.048	D	C
56.	☆	SMPR	0.147	0.455	0.778	49.000	55.000	0.266	8.282	D	M
57.	▷	FoT	0.142	0.377	0.820	66.000	59.000	0.165	105.714	D	C
58.	□	STRUCK2011	0.142	0.458	0.942	44.000	60.000	0.242	14.584	D	C
59.	△	FCT	0.141	0.395	0.788	64.000	56.000	0.199	-	D	M
60.	✡	DFT	0.139	0.464	1.002	44.000	60.000	0.209	3.330	D	C
61.	○	BDF	0.136	0.375	0.792	66.000	57.000	0.180	*138.124*	D	C
62.	×	LT_FLO	0.126	0.444	1.164	52.000	65.000	0.207	1.830	S	M

63.	✱	IVT*	0.115	0.419	1.109	57.000	63.000	0.181	14.880	D M
64.	▽	Matrioska	0.115	0.430	1.114	55.000	64.000	0.238	25.766	D C
65.	◇	STC	0.110	0.380	1.007	66.000	62.000	0.152	22.744	D M
66.	+	FT*	0.104	0.405	1.216	63.000	66.000	0.179	3.867	D C
67.	◁	CTF	0.092	0.497	1.561	29.000	68.000	0.187	3.777	D M
68.	☆	LoFT-Lite	0.092	0.329	1.282	66.000	67.000	0.118	2.174	D M
69.	▷	CMT*	0.083	0.393	1.701	65.000	68.000	0.150	16.196	S P
70.	□	NCC*	0.080	0.490	2.102	36.000	68.000	0.174	**226.891**	D C

Table 2. The table shows expected average overlap (EAO), accuracy and robustness raw values (A,R) and ranks (A_{rank},A_{rank}), the no-reset average overlap AO [21], the speed (in EFO units) and implementation details (M is Matlab, C is C or C++, P is Python). Trackers marked with * have been verified by the VOT2015 committee. A dash "-" indicates the EFO measurements were invalid.

The number of failures with respect to the visual attributes are shown in Figure 4. On camera motion attribute, the tracker that fails least often is the EBT A.2, on illumination change the top position is shared by RFD_CF2 A.47 and SRBT A.34, on motion change the top position is shared by EBT A.2 and MLDF A.19, on occlusion the top position is shared by MDNet_N A.46 and C-COT A.26, on the size change attribute, the tracker MLDF A.19 produces the least failures, while on the unassigned attribute, the TCNN A.44 fails the least often. The overall accuracy and robustness averaged over the attributes is shown in Figure 2. The attribute-normalized AR plots are similar to the pooled plots, but the top trackers (TCNN A.44, SSAT A.12, MDNet_N A.46 and C-COT A.26) are pulled close together, which is evident from the ranking plots.

We have evaluated the difficulty level of each attribute by computing the median of robustness and accuracy over each attribute. According to the results in Table 3, the most challenging attributes in terms of failures are occlusion, motion change and illumination change, followed by scale change and camera motion.

	cam. mot.	**ill. ch.**	**mot. ch.**	**occl.**	**scal. ch.**
Accuracy	0.49	0.53	0.44	**0.41**	*0.42*
Robustness	0.71	0.81	*1.02*	**1.11**	0.61

Table 3. Tracking difficulty with respect to the following visual attributes: camera motion (cam. mot.), illumination change (ill. ch.), motion change (mot. ch.), occlusion (occl.) and size change (scal. ch.) .

In addition to the baseline reset-based VOT experiment, the VOT2016 toolkit also performed the OTB [21] no-reset (OPE) experiment. Figure 5 shows the OPE plots, while the AO overall measure is given in Table 2. According to the AO measure, the three top performing trackers are SSAT (A.12), TCNN (A.44) and C-COT (A.26), which is similar to the EAO ranking, with the main difference that SSAT and C-COT exchange places. The reason for this switch can be deduced from the AR plots (Figure 2) which show that the C-COT is more robust than the other two trackers, while the SSAT is more accurate. Since the AO measure does not apply resets, it does not enhance the differences among the trackers on difficult sequences, where one tracker might fail more

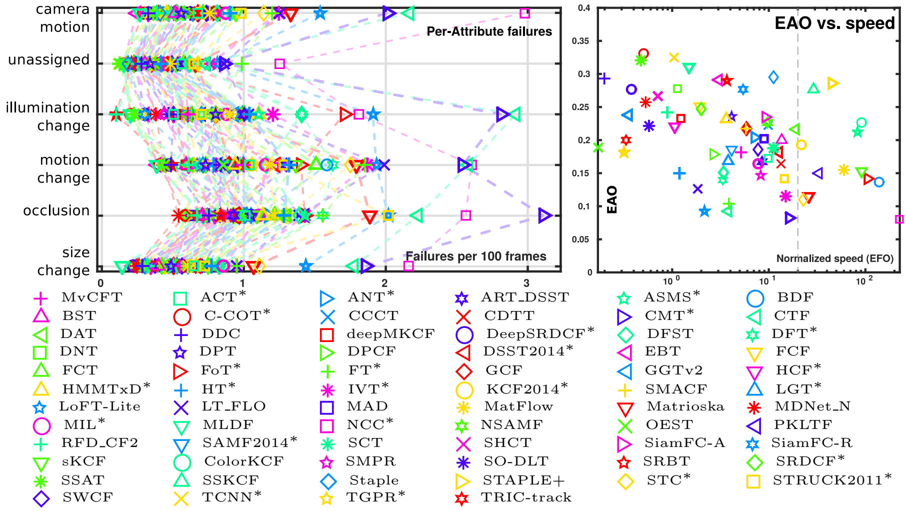

Fig. 4. The expected average overlap with respect to the visual attributes (left). Expected average overlap scores w.r.t. the tracking speed in EFO units (right). The dashed vertical line denotes the estimated real-time performance threshold of 20 EFO units. See Figure 2 for legend.

often than the other, whereas the EAO is affected by these. Thus among the trackers with similar accuracy and robustness, the EAO prefers trackers with higher robustness, while the AO prefers more accurate trackers. To establish a visual relation among the EAO and AO rankings, each tracker is shown in a 2D plot in terms of the EAO and AO measures in Figure 5. Broadly speaking, the measures are correlated and EAO is usually lower than EO, but the local ordering with these measures is different, which is due to the different treatment of failures.

Apart from tracking accuracy, robustness and EAO measure, the tracking speed is also crucial in many realistic tracking applications. We therefore visualize the EAO score with respect to the tracking speed measured in EFO units in Figure 4. 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 200 EFO units. All trackers that scored top EAO performed below realtime, while the top EFO was achieved by NCC (A.61), BDF (A.9) and FoT (A.51). Among the trackers within the VOT2016 realtime bound, the top two trackers in terms of EAO score were Staple+ (A.22) and SSKCF (A.27). The former is modification of the Staple (A.28), while the latter is modification of the Sumshift [38] tracker. Both approaches combine a correlation filter output with color histogram backprojection. According to the AR-raw plot in Figure 2, the SSKCF (A.27) tracks with a decent average overlap during successful tracking periods (~ 0.55) and produces decently long tracks. For example, the probability of SSKCF still tracking the target after $S = 100$ frames is approximately 0.69. The Staple+ (A.22) tracks with a similar overlap (~ 0.56) and tracks the target after 100 frames with probability 0.70. In the detailed analysis of the results we have found some discrepancies between the reported EFO units and the trackers speed in seconds for the Matlab trackers.

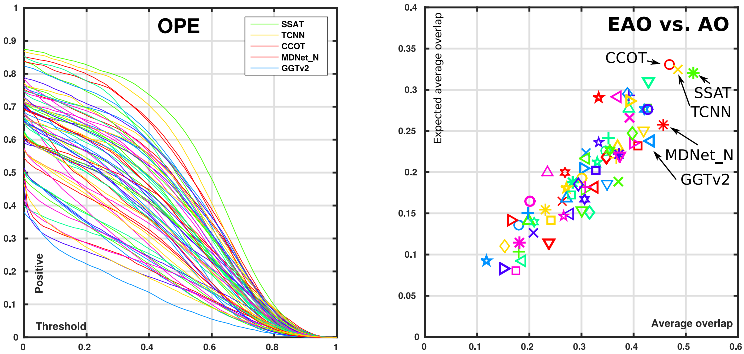

Fig. 5. The OPE no-reset plots (left) and the EAO-AO scatter plot (right).

The toolkit was not ignoring the Matlab start time, which can significantly vary across different trackers. This was particularly obvious in case of SiamFC trackers, which runs orders higher than realtime (albeit on GPU), and Staple, which is realtime, but are incorrectly among the non-realtime trackers in Figure 4.

5 Conclusion

This paper reviewed the VOT2016 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 the same as in the VOT2015 challenge and the per-frame visual attributes are the same as well. A new methodology was developed to automatically place the bounding boxes in each frame by optimizing a well-defined cost function. In addition, a rule-of-thumb approach was developed to estimate the uniqueness of the automatically placed bounding boxes under the expected bound on the per-pixel annotation error. A set of 70 trackers have been evaluated. A large percentage of trackers submitted have been published at recent conferences and top journals, including ICCV, CVPR, TIP and TPAMI, and some trackers have not yet been published (available at arXiv). For example, fifteen trackers alone have been published at major computer vision venues in 2015 and 2016 so far.

The results of VOT2016 indicate that the top performing tracker of the challenge according to the EAO score is the C-COT (A.26) tracker [37]. This is a correlation-filter-based tracker that applies a number of state-of-the-art features. The tracker performed very well in accuracy as well as robustness and trade-off between the two is reflected in the EAO. The C-COT (A.26) tracker is closely followed by TCNN (A.44) and SSAT (A.12) which are close in terms of accuracy, robustness and the EAO. These trackers come from a different class, they are pure CNN trackers based on the winning tracker of VOT2015, the MDNet [33]. It is impossible to conclusively decide whether the improvements of C-COT (A.26) over other top-performing trackers come from the features or the approach. Nevertheless, results of top trackers conclusively show that features play a significant role in the final performance. All trackers that scored the top EAO perform below real-time. Among the realtime trackers, the top performing trackers were Staple+ (A.22) and SSKCF (A.27) that implement a simple combination of the correlation filter output and histogram backprojection.

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 VOT2016 was a fourth attempt toward this, following the very successful VOT2013, VOT2014 and VOT2015. The VOT2016 also introduced a second sub-challenge VOT-TIR2016 that concerns tracking in thermal and infrared imagery. The results of that sub-challenge are described in a separate paper [29] that was presented at the VOT2016 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. Jiři Matas and Tomáš Vojíř 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 Wallenberg Autonomous Systems Program WASP, the Swedish Foundation for Strategic Research through the project CUAS, and the Swedish Research Council trough the project EMC^2. Gustavo Fernández and Roman Pflugfelder were supported by the research program Mobile Vision with funding from the Austrian Institute of Technology. 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 VOT2016 challenge.

A.1 Deformable Part-based Tracking by Coupled Global and Local Correlation Filters (DPCF)

O. Akin, E. Erdem, A. Erdem, K. Mikolajczyk
oakin25@gmail.com, {erkut, aykut}@cs.hacettepe.edu.tr,
k.mikolajczyk@imperial.ac.uk

DPCF is a deformable part-based correlation filter tracking approach which depends on coupled interactions between a global filter and several part filters. Specifically, local filters provide an initial estimate, which is then used by the global filter as a reference to determine the final result. Then, the global filter provides a feedback to the part filters regarding their updates and the related deformation parameters. In this way, DPCF handles not only partial occlusion but also scale changes. The reader is referred to [39] for details.

A.2 Edge Box Tracker (EBT)

G. Zhu, F. Porikli, H. Li
{*gao.zhu, fatih.porikli, hongdong.li*}*@anu.edu.au*

EBT tracker is not limited to a local search window and has ability to probe efficiently the entire frame. It generates a small number of 'high-quality' proposals by a novel instance-specific objectness measure and evaluates them against the object model that can be adopted from an existing tracking-by-detection approach as a core tracker. During the tracking process, it updates the object model concentrating on hard false-positives supplied by the proposals, which help suppressing distractors caused by difficult background clutters, and learns how to re-rank proposals according to the object model. Since the number of hypotheses the core tracker evaluates is reduced significantly, richer object descriptors and stronger detectors can be used. More details can be found in [40].

A.3 Spatial Windowing for Correlation Filter Based Visual Tracking (SWCF)

E. Gundogdu, A. Alatan
egundogdu@aselsan.com.tr, alatan@eee.metu.edu.tr

SWCF tracker estimates a spatial window for the object observation such that the correlation output of the correlation filter and the windowed observation (i.e. element-wise multiplication of the window and the observation) is improved. Concretely, the window is estimated by reducing a cost function, which penalizes the dissimilarity of the correlation of the recent observation and the filter to the desired peaky shaped signal, with an efficient gradient descent optimization. Then, the estimated window is shifted by pre-calculating the translational motion and circularly shifting the window. Finally, the current observation is multiplied element-wise with the aligned window, and utilized in the localization. The reader is referred to [41] for details.

A.4 Point-based Kanade Lukas Tomasi colour-Filter (PKLTF)

R. Martin-Nieto, A. Garcia-Martin, J. M. Martinez
{*rafael.martinn, alvaro.garcia, josem.martinez*}*@uam.es*

PKLTF [42] 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) [43] to choose the object features (using colour and motion coherence), while the second phase is based on mean shift gradient descent [44] to place the bounding box into the position of the object. The object model is based on the RGB colour and the luminance gradient and it consists of a histogram including the quantized values of the colour components, and an edge binary flag. The interested reader is referred to [42] for details.

A.5 Distractor Aware Tracker (DAT)

H. Possegger, T. Mauthner, H. Bischof
{*possegger, mauthner, bischof*}*@icg.tugraz.at*

The Distractor Aware Tracker is an appearance-based tracking-by-detection approach. A discriminative model using colour 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 [45] for details.

A.6 Median Absolute Deviation Tracker (MAD)

S. Becker, S. Krah, W. Hübner, M. Arens
{*stefan.becker, sebastian.krah, wolfgang.huebner,*
michael.arens}*@iosb.fraunhofer.de*

The key idea of the MAD tracker [46] is to combine several independent and heterogeneous tracking approaches and to robustly identify an outlier subset based on the Median Absolute Deviation (MAD) measure. The MAD fusion strategy is very generic and it only requires frame-based target bounding boxes as input and thus can work with arbitrary tracking algorithms. The overall median bounding box is calculated from all trackers and the deviation or distance of a sub-tracker to the median bounding box is calculated using the Jaccard-Index. Further, the MAD fusion strategy can also be applied for combining several instances of the same tracker to form a more robust swarm for tracking a single target. For this experiments the MAD tracker is set-up with a swarm of KCF [47] trackers in combination with the DSST [48] scale estimation scheme. The reader is referred to [46] for details.

A.7 Fully-functional correlation filtering-based tracker (FCF)

M. Zhang, J. Xing, J. Gao, W. Hu
{*mengdan.zhang, jlxing, jin.gao, wmhu*}*@nlpr.ia.ac.cn*

FCF is a fully functional correlation filtering-based tracking algorithm which is able to simultaneously model correlations from a joint scale-displacement space, an orientation space, and the time domain. FCF tracker firstly performs scale-displacement correlation using a novel block-circulant structure to estimate objects position and size in one go. Then, by transferring the target representation from the Cartesian coordinate system to the Log-Polar coordinate system, the circulant structure is well preserved and the object rotation can be evaluated in the same correlation filtering based framework. In the update phase, temporal correlation analysis is introduced together with inference mechanisms which are based on an extended high-order Markov chain.

A.8 Structure Output Deep Learning Tracker (SO-DLT)

N. Wang, S. Li, A. Gupta, D. Yeung
winsty@gmail.com, sliay@cse.ust.hk, abhinavg@cs.cmu.edu,
dyyeung@cse.ust.hk

SO-LDT proposes a 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 [49] for more details.

A.9 Best Displacement Flow (BDF)

M. Maresca, A. Petrosino
mariomaresca@hotmail.it, petrosino@uniparthenope.it

Best Displacement Flow (BDF) is a short-term tracking algorithm based on the same idea of Flock of Trackers [50] 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 [51] for details.

A.10 Matrioska Best Displacement Flow (MatFlow)

M. Maresca, A. Petrosino
mariomaresca@hotmail.it, petrosino@uniparthenope.it

MatFlow enhances the performance of the first version of Matrioska [52] with response given by the short-term tracker BDF (see A.9). 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.11 Matrioska

M. Maresca, A. Petrosino
mariomaresca@hotmail.it, petrosino@uniparthenope.it

Matrioska [52] decomposes tracking into two separate modules: detection and learning. The detection module can use multiple key point-based methods (ORB, FREAK, BRISK, SURF, etc.) inside a fall-back model, to correctly localize the object frame by frame exploiting the strengths of each method. The learning module updates the object model, with a growing and pruning approach, to account for changes in its appearance and extracts negative samples to further improve the detector performance.

A.12 Scale-and-State Aware Tracker (SSAT)

Y. Qi, L. Qin, S. Zhang, Q. Huang
qykshr@gmail.com, qinlei@ict.ac.cn, s.zhang@hit.edu.cn, qmhuang@ucas.ac.cn

SSAT is an extended version of the MDNet tracker [33]. First, a segmentation technique into MDNet is introduced. It works with the scale regression model of MDNet to more accurately estimate the tightest bounding box of the target. Second, a state model is used to infer whether the target is occluded. When the target is occluded, training examples from that frame are not extracted which are used to update the tracker.

A.13 Clustered decision tree based tracker (CDTT)

J. Xiao, R. Stolkin, A. Leonardis
Shine636363@sina.com, {R.Stolkin, a.leonardis}@cs.bham.ac.uk

CDTT tracker is a modified version of the tracker presented in [53]. The tracker first propagates a set of samples, using the top layer features, to find candidate target regions with different feature modalities. The candidate regions generated by each feature modality are adaptively fused to give an overall target estimation in the global layer. When an 'ambiguous' situation is detected (i.e. inconsistent locations of predicted bounding boxes from different feature modalities), the algorithm will progress to the local part layer for more accurate tracking. Clustered decision trees are used to match target parts to local image regions, which initially attempts to match a part using a single feature (first level on the tree), and then progresses to additional features (deeper levels of the tree). The reader is referred to [53] for details.

A.14 Scale and Motion Adaptive Correlation Filter Tracker (SMACF)

M. Mueller, B. Ghanem
{matthias.mueller.2, Bernard.Ghanem}@kaust.edu.sa

The tracker is based on [47]. Colourname features are added for better representation of the target. Depending on the target size, the cell size for extracting features is changed adaptively to provide sufficient resolution of the object being tracked. A first order motion model is used to improve robustness to camera motion. Searching over a number of different scales allows for more accurate bounding boxes and better localization in consecutive frames. For robustness, scales are weighted using a zero-mean Gaussian distribution centred around the current scale. This ensures that the scale is only changed if it results in a significantly better response.

A.15 A multi-view model for visual tracking via correlation Filters (MvCFT)

Z. He, X. Li, N. Fan
zyhe@hitsz.edu.cn, hitlixin@126.com, nanafanhit@gmail.com

The multi-view correlation filter tracker (MvCF tracker) fuses several features and selects the more discriminative features to enhance the robustness. Besides, the correlation filter framework provides fast training and efficient target locating. The combination of the multiple views is conducted by the Kullback-Leibler (KL) divergences. In addition, a simple but effective scale-variation detection mechanism is provided, which strengthens the stability of scale variation tracking.

A.16 Deep multi-kernelized correlation filter (deepMKCF)

J. Feng, F. Zhao, M. Tang
{jiayi.feng, fei.zhao, tangm}@nlpr.ia.ac.cn

deepMKCF tracker is the MKCF [54] with deep features extracted by using VGG-Net [36]. deepMKCF tracker combines the multiple kernel learning and correlation filter techniques and it explores diverse features simultaneously to improve tracking performance. In addition, an optimal search technique is also applied to estimate object scales. The multi-kernel training process of deepMKCF is tailored accordingly to ensure tracking efficiency with deep features. In addition, the net is fine-tuned with a batch of image patches extracted from the initial frame to make VGG-NET-19 more suitable for tracking tasks.

A.17 Discriminative Deep Correlation Tracking (DDC)

J. Gao, T. Zhang, C. Xu, B. Liu
gaojunyu2015@ia.ac.cn, tzzhang10@gmail.com, csxu@nlpr.ia.ac.cn, liubin@dress-plus.com

The Discriminative Deep Correlation (DDC) tracker is based on the correlation filter framework. The tracker uses foreground and background image patches and it has the following advantages: (i) It effectively exploit image patches from foreground and background to make full use of their discriminative context information, (ii) deep features are used to gain more robust target object representations, and (iii) an effective scale adaptive scheme and a long-short term model update scheme are utilised.

A.18 Geometric Structure Hyper-Graph based Tracker Version 2 (GGTv2)

T. Hu, D. Du, L. Wen, W. Li, H. Qi, S. Lyu
{*yihouxiang, cvdaviddo, lywen.cv.workbox, wbli.app, honggangqi.cas, heizi.lyu*}*@gmail.com*

GGTv2 is an improvement of GGT [55] by combining the scale adaptive kernel correlation filter [56] and the geometric structure hyper-graph searching framework to complete the object tracking task. The target object is represented by a geometric structure hyper-graph that encodes the local appearance of the target with higher-order geometric structure correlations among target parts and a bounding box template that represents the global appearance of the target. The tracker use HSV colour histogram and LBP texture to calculate the appearance similarity between associations in the hyper-graph. The templates of correlation filter is calculated by HOG and colour name according to [56].

A.19 Multi-Level Deep Feature Tracker (MLDF)

L. Wang, H. Lu, Yi. Wang, C. Sun
{*wlj,wyfan523,waynecool*}*@mail.dlut.edu.cn, lhchuan@dlut.edu.cn*

MLDF tracker is based on deep convolutional neural networks (CNNs). The proposed MLDF tracker draws inspiration from [57] by combining low, mid and high-level features from the pre trained VGG networks [36]. A Multi-Level Network (MLN) is designed to take these features as input and online trained to predict the centre location of the target. By jointly considering multi-level deep features, the MLN is capable to distinguish the target from background objects of different categories. While the MLN is used for location prediction, a Scale Prediction Network (SPN) [58] is applied to handle scale variations.

A.20 Colour-aware Complex Cell Tracker (CCCT)

D. Chen, Z. Yuan
dapengchenxjtu@foxmail.com, yuan.ze.jian@xjtu.edu.cn

The proposed tracker is a variant of CCT proposed in [59]. CCT tracker applies intensity histogram, oriented gradient histogram and colour name features to construct four types of complex cell descriptors. A score normalization strategy is adopted to weight different visual cues as well as different types of complex cell. Besides, occlusion inference and stability analysis are performed over each cell to increase the robustness of tracking. For more details, the reader is referred to [59].

A.21 A New Scale Adaptive and Multiple Feature based on kernel correlation filter tracker (NSAMF)

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

NSAMF is an improved version of the previous method SAMF [56]. To further exploit color information, NSAMF employs color probability map, instead of color name, as color based feature to achieve more robust tracking results. In addition, multi-models based on different features are integrated to vote the final position of the tracked target.

A.22 An improved STAPLE tracker with multiple feature integration (Staple+)

Z. Xu, Y. Li, J. Zhu
xuzhan2012@whu.edu.cn, {*liyang89, jkzhu*}*@zju.edu.cn*

An improved version of STAPLE tracker [60] by integrating multiple features is presented. Besides extracting HOG feature from merely gray-scale image as they do in [60], we also extract HOG feature from color probability map, which can exploit color information better. The final response map is thus a fusion of different features.

A.23 SiameseFC-ResNet (SiamFC-R)

L. Bertinetto, J. F. Henriques, J. Valmadre, P. H. S. Torr, A. Vedaldi
{*luca, joao, jvlmdr*}*@robots.ox.ac.uk,*
philip.torr@eng.ox.ac.uk, vedaldi@robots.ox.ac.uk

SiamFC-R is similar to SiamFC-A A.25, except that it uses a ResNet architecture instead of AlexNet for the embedding function. The parameters for this network were initialised by pre-training for the ILSVRC image classification problem, and then fine-tuned for the similarity learning problem in a second offline phase.

A.24 Structure Hyper-graph based Correlation Filter Tracker (SHCT)

L. Wen, D. Du, S. Li, C.-M. Chang, S. Lyu, Q. Huang
{*lywen.cv.workbox, cvdaviddo, shengkunliluo, mingching, heizi.lyu*}*@gmail.com, qmhuang@jdl.ac*

SHCT tracker constructs a structure hyper-graph model [61] to extract the motion coherence of target parts. The tracker also computes a part confidence map based on the extracted dense subgraphs on the constructed structure hyper-graph, which indicates the confidence score of the part belonging to the target. SHCT uses HSV colour histogram and LBP feature to calculate the appearance similarity between associations in the hyper-graph. Finally, the tracker combines the response maps of correlation filter and structure hyper-graph in a linear way to find the optimal target state (i.e., target scale and location). The templates of correlation filter are calculated by HOG and colour name according to [56]. The appearance models of correlation filter and structure hyper-graph are updated to ensure the tracking performance.

A.25 SiameseFC-AlexNet (SiamFC-A)

L. Bertinetto, J. F. Henriques, J. Valmadre, P. H. S. Torr, A. Vedaldi
{luca, joao, jvlmdr}@robots.ox.ac.uk,
philip.torr@eng.ox.ac.uk, vedaldi@robots.ox.ac.uk

SiamFC-A [62] applies a fully-convolutional Siamese network [63] trained to locate an exemplar image within a larger search image. The architecture is fully convolutional with respect to the search image: dense and efficient sliding-window evaluation is achieved with a bilinear layer that computes the cross-correlation of two inputs. The deep convnet (namely, a AlexNet [64]) is first trained offline on the large ILSVRC15 [65] video dataset to address a general similarity learning problem, and then this function is evaluated during testing by a simplistic tracker. SiamAN incorporates elementary temporal constraints: the object search is done within a region of approximately four times its previous size, and a cosine window is added to the score map to penalize large displacements. SiamAN also processes several scaled versions of the search image, any change in scale is penalised and damping is applied to the scale factor.

A.26 Continuous Convolution Operator Tracker (C-COT)

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

C-COT learns a discriminative continuous convolution operator as its tracking model. C-COT poses the learning problem in the continuous spatial domain. This enables a natural and efficient fusion of multi-resolution feature maps, e.g. when using several convolutional layers from a pre-trained CNN. The continuous formulation also enables highly accurate localization by sub-pixel refinement. The reader is referred to [37] for details.

A.27 SumShift Tracker with Kernelized Correlation Filter (SSKCF)

J.-Y. Lee, S. Choi, J.-C. Jeong, J.-W. Kim, J.-I. Cho
{jylee, sunglok, channij80, giraffe, jicho}@etri.re.kr

SumShiftKCF tracker is an extension of the SumShift tracker [38] by the kernelized correlation filter tracker (KCF) [47]. The SumShiftKCF tracker computes the object likelihood with the weighted sum of the histogram back-projection weights and the correlation response of KCF. Target is then located by the Sum-Shift iteration [38].

A.28 Sum of Template And Pixel-wise LEarners (Staple)

L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, P. H. S. Torr
{*luca, jvlmdr*}*@robots.ox.ac.uk, stuart.golodetz@ndcn.ox.ac.uk,*
{*ondrej.miksik, philip.torr*}*@eng.ox.ac.uk*

Staple is a tracker that combines two image patch representations that are sensitive to complementary factors to learn a model that is inherently robust to both colour changes and deformations. To maintain real-time speed, two independent ridge-regression problems are solved, exploiting the inherent structure of each representation. Staple combines the scores of two models in a dense translation search, enabling greater accuracy. A critical property of the two models is that their scores are similar in magnitude and indicative of their reliability, so that the prediction is dominated by the more confident. For more details, we refer the reader to [60].

A.29 Kalman filter ensemble-based tracker (ColorKCF)

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

The colourKCF method fuses the result of two out-of-the box trackers, a mean-shift tracker that uses colour histogram (ASMS) [66] and the kernelized correlation filter (KCF) [47] by using a Kalman filter. The tracker works in prediction and correction cycles. First, a simple motion model predicts the target next position, then, the trackers results are fused with the predicted position and the motion model is updated in the correction process. The fused result is the SMACF output which is used as last position of the tracker in the next frame. The Kalman filter needs a measure to define how reliable each result is during the fusion process. For this, the tracker uses the result confidence and the motion penalization which is proportional to the distance between the tracker result and the predict result. As confidence measure, the Bhattacharyya coefficient between the model and the target histogram is used in case of ASMS tracker, while the correlation result is applied in case of KCF tracker. The initial name of this tracker when submitted to the challenge was ColorKCF.

A.30 Best Structured Tracker (BST)

F. Battistone, A. Petrosino, V. Santopietro
{*battistone.francesco, vinsantopietro*}*@gmail.com, petrosino@uniparthenope.it*

BST is based on the idea of Flock of Trackers [67]: a set of local trackers tracks a little patch of the original target and then the tracker combines their information in order to estimate the resulting bounding box. Each local tracker separately analyzes the features extracted from a set of samples and then classifies them using a structured Support Vector Machine as Struck [67]. Once having predicted local target candidates, an outlier detection process is computed by analyzing the displacements of local trackers. Trackers that have been labeled as outliers are reinitialized. At the end of this process, the new bounding box is calculated using the Convex Hull technique.

A.31 Online Evaluation-based Self-Correction Tracker (OEST)

Z. Cai, P. C. Yuen, A. J. Ma, X. Lan
{*cszxcai, pcyuen, andyjhma, xylan*}*@comp.hkbu.edu.hk*

Online Evaluation-based Self-Correction Tracker aims at improving the tracking performance based on any existing tracker. OEST consists of three steps. Firstly, the long-term correlation tracker (LCT) [68] is employed to determine the bounding box of the target at the current frame. Secondly, an online tracking performance estimator is deployed to evaluate whether the output bounding box provided by the base tracker can correctly locate the target by analyzing the previous tracking results. Comparing existing performance estimators, the time-reverse method [69] achieves the best evaluation performance. Thirdly, if the online tracking performance estimator determines that the base tracker fails to track the target, a re-detection algorithm is performed to correct the output of the tracker. An online SVM detector as in [70] is employed in this re-detection step. Tracker outputs with high confidence determined by the performance estimator are used to update the detector.

A.32 Tracking by Regression with Incrementally Learned Cascades (TRIC-track)

X. Wang, M. Valstar, B. Martinez, M. H. Khan, T. 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) [71] 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 [72] applying a sequential incremental update. Shape constraints are, however, implicitly encoded by allowing patches sampled around neighbouring parts to vote for a given parts location. TRIC-track also possesses a multiple temporal scale motion model [73] 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.33 Correlation-based Tracker Level Fusion (CTF)

M. k. Rapuru, S. Kakanuru, D. Mishra, G. R K S. Subrahmanyam
madankumar.r@gmail.com, kakanurusumithra05@gmail.com,
{deepak.mishra, gorthisubrahmanyam}@iist.ac.in

The Correlation based Tracker level Fusion (CTF) method combines two state-of-the-art trackers, which have complementary nature in handling tracking challenges and also in the methodology of tracking. CTF considers the outputs of both trackers Tracking Learning Detection (TLD) [75] tracker and Kernelized Correlation Filters (KCF) tracker [47], and selects the best patch by measuring the correlation correspondence with the stored object model sample patches. An integration of frame level detection strategy of TLD with systematic model update strategy of KCF are used to increase the robustness. Since KCF tracker exploits the circulant structure in the training and testing data, a high frame rate with less overhead is achieved. CTF method can handle scale changes, occlusions and tracking resumption with the virtue of TLD, whereas KCF fails in handling these challenges. The proposed methodology is not limited to integrating just TLD and KCF, it is a generic model where any best tracker can be combined with TLD to leverage the best performance.

A.34 Salient Region Based Tracker (SRBT)

H. Lee, D. Kim
{lhmin, dkim}@postech.ac.kr

Salient Region Based Tracker separates the exact object region contained in the bounding box - called the salient region - from the background region. It uses the colour model and appearance model to estimate the location and size of the target. During an initialization step, the salient region is set to the ground truth region and is updated for each frame. While estimating the target location and updating the model, only the pixels inside the salient region can participate as contributors. An additional image template as appearance model is used to catch like edges and shape. The colour histogram model is adopted from DAT [45] excluding the distractor-awareness concept.

A.35 Deformable part correlation filter tracker (DPT)

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

DPT is a part-based correlation filter composed of a coarse and mid-level target representations. Coarse representation is responsible for approximate target localization and uses HOG as well as colour features. The mid-level representation is a deformable parts correlation filter with fully-connected parts topology and applies a novel formulation that threats geometric and visual properties within a single convex optimization function. The mid level as well as coarse level representations are based on the kernelized correlation filter from [47]. The reader is referred to [76] for details.

A.36 Guided correlation filter (GCF)

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

GCF (guided correlation filter) is a correlation filter based tracker that uses colour segmentation [77] (implementation from [78]) to improve the robustness of the correlation filter learning process. The segmentation mask is combined with the correlation filter to reduce the impact of the background and the circular correlations effects, which are the most problematic when tracking rotated or non-axis aligned objects. The tracker uses HOG [79] features for target localization and the DSST [48] approach for scale estimation.

A.37 Optical flow clustering tracker (FCT)

A. Varfolomieiev
a.varfolomieiev@kpi.ua

FCT is based on the same idea as the best displacement tracker (BDF) [51]. It uses pyramidal Lucas-Kanade optical flow algorithm to track individual points of an object at several pyramid levels. The results of the point tracking are clustered in the same way as in the BDF [51] to estimate the best object displacement. The initial point locations are generated by the FAST detector [80]. The tracker estimates a scale and an in-plane rotation of the object. These procedures are similar to the scale calculation of the median flow tracker [81], except that the clustering is used instead of median. In case of rotation calculation angles between the respective point pairs are clustered. In contrast to BDF, the FCT does not use consensus-based reinitialization. The current implementation of FCT calculates the optical flow only in the objects region, which is four times larger than the initial bounding box of the object, and thus speeds up the tracker with respect to its previous version [18].

A.38 Likelihood of Features Tracking-Lite (LoFT-Lite)

M. Poostchi, K. Palaniappan, F. Bunyak, G. Seetharaman, R. Pelapur, K. Gao, S. Yao, N. Al-Shakarji
mpoostchi@mail.missouri.edu, {pal, bunyak}@missouri.edu, guna@ieee.org
{rvpnc4, kg954, syyh4, nmahyd}@missouri.edu,

LoFT (Likelihood of Features Tracking)-Lite [82] is an appearance based single object tracker optimized for aerial video. Target objects are characterized using low level image feature descriptors including intensity, color, shape and edge attributes based on histograms of intensity, color-name space, gradient magnitude and gradient orientation. The feature likelihood maps are computed using fast integral histograms [83] within a sliding window framework that compares histogram descriptors. Intensity and gradient magnitude normalized cross-correlations likelihood maps are also used to incorporate spatial structure information. An informative subset of six features from the collection of eleven features is used that are the most discriminative based on an offline feature subset selection method [84]. LoFT performs feature fusion using a foreground-background model by comparing the current target appearance with the model inside the search region [85]. LOFT-Lite also incorporates an adaptive orientation-based Kalman prediction update to restrict the search region which reduces sensitivity to abrupt motion changes and decreases computational cost [86].

A.39 Dynamic Feature Selection Tracker (DFST)

G. Roffo, S. Melzi
{giorgio.roffo, simone.melzi}@univr.it

DFST proposes an optimized visual tracking algorithm based on the real-time selection of locally and temporally discriminative features. A feature selection mechanism is embedded in the Adaptive colour Names [87] (CN) tracking system that adaptively selects the top-ranked discriminative features for tracking. DFST provides a significant gain in accuracy and precision allowing the use of a dynamic set of features that results in an increased system flexibility. DFST is based on the unsupervised method Inf-FS [88, 89], which ranks features according with their 'redundancy' without using class labels. By using a fast online algorithm for learning dictionaries [90] the size of the box is adapted during the processing. At each update, multiple examples at different positions and scales around the target are used. A further improvement of the CN system is given by making micro-shifts at the predicted position according to the best template matching. The interested reader is referred to [89] for details.

A.40 Scalable Kernel Correlation Filter with Sparse Feature Integration (sKCF)

A. Solís Montero, J. Lang, R. Laganière
asolismo@uottawa.ca, {jlang, laganier}@eecs.uottawa.ca

sKCF [91] 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 along with some performace enhancements. In the submission, we introduce a model learning strategy to the original sKCF [91] which updates the model only for highly similar KCF responses of the tracked region as to the model. This potentially limits model drift due to temporary disturbances or occlusions. The original sKCF always updates the model in each frame.

A.41 Dual Deep Network Tracker (DNT)

Z. Chi, H. Lu, L. Wang, C. Sun
{*zhizhenchi, wlj, waynecool*}*@mail.dlut.edu.cn, lhchuan@dlut.edu.cn*

DNT proposes a dual network for visual tracking. First, the hierarchical features in two different layers of a deep model pre-trained are exploited for object recognition. Features in higher layers encode more semantic contexts while those in lower layers are more effective to discriminative appearance. To highlight geometric contours of the target, the hierarchical feature maps are integrated with an edge detector as the coarse prior maps. To measure the similarities between the network activation and target appearance, a dual network with a supervised loss function is trained. This dual network is updated online in a unique manner based on the observation that the tracking target in consecutive frames should share more similar feature representations than those in the surrounding background. Using prior maps as guidance, the independent component analysis with reference algorithm is used to extract the exact boundary of a target object, and online tracking is conducted by maximizing the posterior estimate on the feature maps with stochastic and periodic update.

A.42 Structuralist Cognitive model for visual Tracking (SCT)

J. Choi, H. J. Chang, J. Jeong, Y. Demiris, J. Y. Choi
jwchoi.pil@gmail.com, hj.chang@imperial.ac.uk, jy.jeong@snu.ac.kr,
y.demiris@imperial.ac.uk, jychoi@snu.ac.kr

SCT [92] is composed of two separate stages: disintegration and integration. In the disintegration stage, the target is divided into a number of small cognitive structural units, which are memorized separately. Each unit includes a specific colour or a distinguishable target shape, and is trained by elementary trackers with different types of kernel. In the integration stage, an adequate combination of the structural units is created and memorized to express the targets appearance. When encountering a target with changing appearance in diverse environments, SCT tracker utilizes all the responses from the cognitive units memorized in the disintegration stage and then recognizes the target through the best combination of cognitive units, referring to the memorized combinations. With respect to the elementary trackers, an attentional feature-based correlation filter (AtCF) is used. The AtCF focuses on the attentional features discriminated from the background. Each AtCF consists of an attentional weight estimator and a kernelized correlation filter (KCF) [47]. In the disintegration stage, multiple AtCFs are updated using various features and kernel types. The integration stage combines the responses of AtCFs by ordering the AtCFs following their performance.

A.43 Long Term Featureless Object Tracker (LT-FLO)

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

The tracker is based on and extends previous work of the authors on tracking of texture-less objects [93]. 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 [94, 95] for details.

A.44 Tree-structured Convolutional Neural Network Tracker (TCNN)

H. Nam, M. Baek, B. Han
{namhs09, mooyeol, bhhan}@postech.ac.kr

TCNN [96] maintains multiple target appearance models based on CNNs in a tree structure to preserve model consistency and handle appearance multi-modality effectively. TCNN tracker consists of two main components, state estimation and model update. When a new frame is given, candidate samples around the target state estimated in the previous frame are drawn, and the likelihood of each sample based on the weighted average of the scores from multiple CNNs is computed. The weight of each CNN is determined by the reliability of the path along which the CNN has been updated in the tree structure. The target state in the current frame is estimated by finding the candidate with the maximum likelihood. After tracking a predefined number of frames, a new CNN is derived from an existing one, which has the highest weight among the contributing CNNs to target state estimation.

A.45 Adaptive Regression Target Discriminative Scale Space Tracking (ART-DSST)

L. Zhang, J. Van de Weijer, M. Mozerov, F. Khan
{lichao, joost, mikhail}@cvc.uab.es, fahad.khan@liu.se

Correlation based tracking optimizes the filter coefficients such that the resulting filter response is an isotropic Gaussian. However, for rectangular shapes the overlap error diminishes anisotropically: faster along the short axes than the long axes of the rectangle. To exploit this observation, ART-DSST proposes the usage of an anisotropic Gaussian regression target which adapts to the shape of the bounding box. The method is general because it can be applied to all regression based trackers.

A.46 Multi-Domain Convolutional Neural Network Tracker (MDNet-N)

H. Nam, M. Baek, B. Han
{namhs09, mooyeol, bhhan}@postech.ac.kr

This algorithm is a variation of MDNet [33], which does not pre-train CNNs with other tracking datasets. The network is initialised using the ImageNet [97]. The new classification layer and the fully connected layers within the shared layers are then fine-tuned online during tracking to adapt to the new domain. The online update is conducted to model long-term and short-term appearance variations of a target for robustness and adaptiveness, respectively, and an effective and efficient hard negative mining technique is incorporated in the learning procedure. This experiment result shows that the online tracking framework scheme of MDNet is still effective without multi-domain training.

A.47 CF2 with Response Information Failure Detection (RFD-CF2)

R. Walsh, H. Medeiros
{ryan.w.walsh, henry.medeiros}@marquette.edu,

RFD-CF2 is a modified version of the Correlation Filters with Convolutional Features tracker (CF2) extended with a failure detection module [98]. Hard occlusions and blurring of the target are detected by extracting features out of the response map. The tracker uses this information to scale the trackers search space and minimize bad updates from occurring.

A.48 Scalable Multiple Part Regressors tracker (SMPR)

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

SMPR framework applies both global and local correlation filter-based part regressors in object modeling. To follow target appearance changes, importance weights are dynamically assigned to each model part via solving a multi linear ridge regression optimization problem. During model update, a helpful scale estimation technique based on weighted relative movement of pair-wise inlier parts is applied. Without loss of generality, conventional CN tracker [87] is utilized as a sample CFT baseline to expeditiously track each target object part by feeding color-induced attributes into fast CSK tracker [99]. Similar to CN approach [87], low dimensional colour names together with greyscale features are employed to represent each part of the object model.

A.49 Scale Adaptive Mean Shift (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 [100] 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 histogram colour weighting and a forward-backward consistency check.

A.50 Discriminative Scale Space Tracker (DSST2014)

Authors implementation. Submitted by VOT Committee

The Discriminative Scale Space Tracker (DSST) [48] extends the Minimum Output Sum of Squared Errors (MOSSE) tracker [101] 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.51 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.52 HMMTxD

Submitted by VOT Committee

The HMMTxD [102] 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.53 Kernelized Correlation Filter tracker (KCF2014)

Modified version of the authors implementation. Submitted by VOT Committee

This tracker is basically a Kernelized Correlation Filter [47] operating on simple HOG features. The KCF tracker 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.54 A kernel correlation filter tracker with Scale Adaptive and Feature Integration (SAMF2014)

Authors implementation. Submitted by VOT Committee

SAMF tracker is based on the idea of correlation filter-based trackers 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 STRUCK (Struck2011)

Submitted by VOT Committee

Struck [103] 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.

A.56 Adaptive Color Tracker (ACT)

Authors implementation. Submitted by VOT Committee

The Adaptive Color Tracker (ACT) [104] extends the CSK tracker [99] with colour information. ACT tracker contains three improvements to CSK tracker: (i) A temporally 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.57 Spatially Regularized Discriminative Correlation Filter with Deep Features (DeepSRDCF)

Authors implementation. Submitted by VOT Committee

The DeepSRDCF incorporates deep convolutional features in the SRDCF framework proposed in [105]. 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 [105] for details.

A.58 Spatially Regularized Discriminative Correlation Filter Tracker (SRDCF)

Authors implementation. Submitted by VOT Committee

Standard Discriminative Correlation Filter (DCF) based trackers such as [48, 87, 47] 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. [48, 87, 47]), 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 [105], 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 [105].

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 Hierarchical Convolutional Features for Visual Tracking (HCF)

Submitted by VOT Committee

HCF tracker [106] is a kernelized correlation filter applied to VGG convnet features. The tracker exploits boths spatial details and semantics. While the last convolutional layers encode the semantic information of targets, earlier convolutional layers retain more fine-grained spatial details providing more precise localization. The reader is referred to [106] for details.

A.61 Normalized Cross-Correlation (NCC)

Submitted by VOT Committee

The NCC tracker is a VOT2016 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.62 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 [107] for details.

A.63 Anchor Template Tracker (ANT)

Submitted by VOT Committee

The ANT tracker is a conceptual increment to the idea of multi-layer appearance representation that is first described in [107]. The tracker addresses the problem of self-supervised estimation of a large number of parameters by introducing controlled graduation in estimation of the free parameters. The appearance of the object is decomposed into several sub-models, each describing the target at a different level of detail. The sub models interact during target localization and, depending on the visual uncertainty, serve for cross-sub-model supervised updating. The reader is referred to [108] for details.

A.64 Incremental Learning for Robust Visual Tracking (IVT)

Submitted by VOT Committee

The idea of the IVT tracker [109] is to incrementally learn a low-dimensional subspace representation, adapting on-line 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.65 HoughTrack (HT)

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 [110] for details and to http://lrs.icg.tugraz.at/research/houghtrack/for code.

A.66 Spatio-temporal context tracker (STC)

Submitted by VOT Committee

The STC [111] 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.67 Transfer Learning Based Visual Tracking with Gaussian Processes Regression (TGPR)

Submitted by VOT Committee

The TGPR tracker [112] 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.68 Multiple Instance Learning tracker (MIL)

Submitted by VOT Committee

MIL tracker [113] 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.69 Robust Fragments based Tracking using the Integral Histogram - FragTrack (FT)

Submitted by VOT Committee

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

A.70 Consensus Based Matching and Tracking (CMT)

Submitted by VOT Committee

The CMT tracker is a keypoint-based method in a combined matching-and-tracking framework. To localise the object in every frame, each key point casts votes for the object center. A consensus-based scheme is applied for outlier detection in the voting behaviour. By transforming votes based on the current key point constellation, changes of the object in scale and rotation are considered. The use of fast keypoint detectors and binary descriptors allows the current implementation to run in real-time. The reader is referred to [114] for details.

References

1. Gavrila, D.M.: The visual analysis of human movement: A survey. Comp. Vis. Image Understanding **73**(1) (1999) 82–98
2. Moeslund, T.B., Granum, E.: A survey of computer vision-based human motion capture. Comp. Vis. Image Understanding **81**(3) (March 2001) 231–268
3. Gabriel, P., Verly, J., Piater, J., Genon, A.: The state of the art in multiple object tracking under occlusion in video sequences. In: Proc. Advanced Concepts for Intelligent Vision Systems. (2003) 166–173
4. Hu, W., Tan, T., Wang, L., Maybank, S.: A survey on visual surveillance of object motion and behaviors. IEEE Trans. Systems, Man and Cybernetics, C **34**(30) (2004) 334–352
5. Moeslund, T.B., Hilton, A., Kruger, V.: A survey of advances in vision-based human motion capture and analysis. Comp. Vis. Image Understanding **103**(2-3) (November 2006) 90–126
6. Yilmaz, A., Shah, M.: Object tracking: A survey. Journal ACM Computing Surveys **38**(4) (2006)
7. Yang, H., Shao, L., Zheng, F., Wang, L., Song, Z.: Recent advances and trends in visual tracking: a review. Neurocomputing **74**(18) (2011) 3823–3831
8. Zhang, S., Yao, H., Sun, X., Lu, X.: Sparse coding based visual tracking: Review and experimental comparison. Pattern Recognition **46**(7) (2013) 1772 – 1788
9. Li, X., Hu, W., Shen, C., Zhang, Z., Dick, A.R., Van den Hengel, A.: A survey of appearance models in visual object tracking. arXiv:1303.4803 [cs.CV] (2013)
10. Young, D.P., Ferryman, J.M.: Pets metrics: On-line performance evaluation service. In: ICCCN '05 Proceedings of the 14th International Conference on Computer Communications and Networks. (2005) 317–324

11. Goyette, N., Jodoin, P.M., Porikli, F., Konrad, J., Ishwar, P.: Changedetection.net: A new change detection benchmark dataset. In: CVPR Workshops, IEEE (2012) 1–8
12. Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J.: The feret evaluation methodology for face-recognition algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **22**(10) (2000) 1090–1104
13. Kasturi, R., Goldgof, D.B., Soundararajan, P., Manohar, V., Garofolo, J.S., Bowers, R., Boonstra, M., Korzhova, V.N., Zhang, J.: 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) (2009) 319–336
14. Leal-Taixé, L., Milan, A., Reid, I.D., Roth, S., Schindler, K.: Motchallenge 2015: Towards a benchmark for multi-target tracking. CoRR **abs/1504.01942** (2015)
15. Solera, F., Calderara, S., Cucchiara, R.: Towards the evaluation of reproducible robustness in tracking-by-detection. In: Advanced Video and Signal Based Surveillance. (2015) 1 – 6
16. Kristan, M., Pflugfelder, R., Leonardis, A., Matas, J., Porikli, F., Cehovin, L., Nebehay, G., G., F., Vojir, T., et al.: The visual object tracking vot2013 challenge results. In: ICCV2013 Workshops, Workshop on visual object tracking challenge. (2013) 98 –111
17. Kristan, M., Pflugfelder, R., Leonardis, A., Matas, J., Cehovin, L., Nebehay, G., Vojir, T., G., F., et al.: The visual object tracking vot2014 challenge results. In: ECCV2014 Workshops, Workshop on visual object tracking challenge. (2014)
18. Kristan, M., Matas, J., Leonardis, A., Felsberg, M., et al.: The visual object tracking vot2015 challenge results. In: ICCV2015 Workshops, Workshop on visual object tracking challenge. (2015)
19. Kristan, M., Matas, J., Leonardis, A., Vojir, T., Pflugfelder, R., Fernandez, G., Nebehay, G., Porikli, F., Čehovin, L.: A novel performance evaluation methodology for single-target trackers. IEEE Transactions on Pattern Analysis and Machine Intelligence (To appear 2016)
20. Čehovin, L., Leonardis, A., Kristan, M.: Visual object tracking performance measures revisited. IEEE Transactions on Image Processing **25**(3) (2015)
21. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: Computer Vision and Pattern Recognition. (2013)
22. Smeulders, A.W.M., Chu, D.M., Cucchiara, R., Calderara, S., Dehghan, A., Shah, M.: Visual Tracking: an Experimental Survey. TPAMI (2013)
23. Wu, Y., Lim, J., Yang, M.H.: Object tracking benchmark. IEEE-PAMI (2015)
24. Li, A., Li, M., Wu, Y., Yang, M.H., Yan, S.: Nus-pro: A new visual tracking challenge. IEEE-PAMI (2015)
25. Liang, P., Blasch, E., Ling, H.: Encoding color information for visual tracking: Algorithms and benchmark. IEEE Transactions on Image Processing **24**(12) (2015) 5630–5644
26. Čehovin, L., Kristan, M., Leonardis, A.: Is my new tracker really better than yours? WACV 2014: IEEE Winter Conference on Applications of Computer Vision (2014)
27. Wu, Y., Lim, J., Yang, M.: Object tracking benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(9) (2014) 1834–1848
28. Felsberg, M., Berg, A., Häger, G., Ahlberg, J., et al.: The thermal infrared visual object tracking VOT-TIR2015 challenge results. In: ICCV2015 workshop proceedings, VOT2015 Workshop. (2015)

29. Felsberg, M., Kristan, M., Leonardis, A., Matas, J., Pflugfelder, R., et al.: The thermal infrared visual object tracking VOT-TIR2016 challenge results. In: ECCV2016 Workshop Proceedings, VOT2016 Workshop. (2016)
30. Rother, C., Kolmogorov, V., Blake, A.: "grabcut": Interactive foreground extraction using iterated graph cuts. In: ACM SIGGRAPH 2004 Papers. SIGGRAPH '04, New York, NY, USA, ACM (2004) 309–314
31. Byrd, H.R., Gilbert, C.J., Nocedal, J.: A trust region method based on interior point techniques for nonlinear programming. Mathematical Programming **89**(1) (2000) 149–185
32. Shanno, D.F.: Conditioning of quasi-newton methods for function minimization. Mathematics of computation **24**(111) (1970) 647–656
33. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. In: CoRR. (2015)
34. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(9) (2010) 1627–1645
35. Van de Weijer, J., Schmid, C., Verbeek, J., Larlus, D.: Learning color names for real-world applications. IEEE Transactions on Image Processing **18**(7) (2009) 1512–1524
36. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR. (2015)
37. Danelljan, M., Robinson, A., Shahbaz Khan, F., Felsberg, M.: Beyond correlation filters: Learning continuous convolution operators for visual tracking. In: ECCV. (2016)
38. Lee, J.Y., Yu, W.: Visual tracking by partition-based histogram backprojection and maximum support criteria. In: Proceedings of the IEEE International Conference on Robotics and Biomimetic (ROBIO). (2011)
39. Akin, O., Erdem, E., Erdem, A., Mikolajczyk, K.: Deformable part-based tracking by coupled global and local correlation filters. Journal of Visual Communication and Image Representation **38** (2016) 763–774
40. Zhu, G., Porikli, F., Li, H.: Beyond local search: Tracking objects everywhere with instance-specific proposals. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016)
41. Gundogdu, E., Alatan, A.A.: Spatial windowing for correlation filter based visual tracking. In: ICIP. (2016)
42. González, A., Martín-Nieto, R., Bescós, J., Martínez, J.M.: Single object long-term tracker for smart control of a PTZ camera. In: International Conference on Distributed Smart Cameras. (2014) 121–126
43. Shi, J., Tomasi, C.: Good features to track. In: Computer Vision and Pattern Recognition. (June 1994) 593 – 600
44. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: Computer Vision and Pattern Recognition. Volume 2. (2000) 142–149
45. Possegger, H., Mauthner, T., Bischof, H.: In defense of color-based model-free tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015)
46. Becker, S., Krah, S.B., Hübner, W., Arens, M.: Mad for visual tracker fusion. SPIE Proceedings Optics and Photonics for Counterterrorism, Crime Fighting, and Defence **9995** (2016, to appear)

47. Henriques, J., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(3) (2015) 583–596
48. Danelljan, M., Häger, G., Khan, F.S., Felsberg, M.: Accurate scale estimation for robust visual tracking. In: Proceedings of the British Machine Vision Conference BMVC. (2014)
49. Wang, N., Li, S., Gupta, A., Yeung, D.Y.: Transferring rich feature hierarchies for robust visual tracking (2015)
50. Vojir, T., Matas, J.: Robustifying the flock of trackers. In: Computer Vision Winter Workshop, IEEE (2011) 91–97
51. Maresca, M., Petrosino, A.: 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)
52. Maresca, M.E., Petrosino, A.: Matrioska: A multi-level approach to fast tracking by learning. In: Proc. Int. Conf. Image Analysis and Processing. (2013) 419–428
53. Jingjing, X., Stolkin, R., Leonardis, A.: Single target tracking using adaptive clustered decision trees and dynamic multi-level appearance models. In: CVPR. (2015)
54. Tang, M., Feng, J.: Multi-kernel correlation filter for visual tracking. In: ICCV. (2015)
55. Du, D., Qi, H., Wen, L., Tian, Q., Huang, Q., Lyu, S.: Geometric hypergraph learning for visual tracking. In: CoRR. (2016)
56. Li, Y., Zhu, J.: A scale adaptive kernel correlation filter tracker with feature integration. In: Proceedings of the ECCV Workshop. (2014) 254–265
57. Wang, L., Ouyang, W., Wang, X., Lu, H.: Visual tracking with fully convolutional networks. In: ICCV. (2015)
58. Wang, L., Ouyang, W., Wang, X., Lu, H.: Stct: Sequentially training convolutional networks for visual tracking. In: CVPR. (2016)
59. Chen, D., Yuan, Z., Wu, Y., Zhang, G., Zheng, N.: Constructing adaptive complex cells for robust visual tracking. In: ICCV. (2013)
60. Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., Torr, P.H.S.: Staple: Complementary learners for real-time tracking. In: CVPR. (2016)
61. Du, D., Qi, H., Li, W., Wen, L., Huang, Q., Lyu, S.: Online deformable object tracking based on structure-aware hyper-graph. IEEE Transactions on Image Processing **25**(8) (2016) 3572–3584
62. Bertinetto, L., Valmadre, J., Henriques, J., Torr, P.H.S., Vedaldi, A.: Fully convolutional siamese networks for object tracking. In: ECCV Workshops. (2016)
63. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: CVPR. (2005)
64. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv:1512.03385 [cs.CV] (2015)
65. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge. IJCV (2015)
66. Vojir, T., Noskova, J., Matas, J.: Robust scale-adaptive mean-shift for tracking. Image Analysis (2013) 652–663
67. Vojir, T., Matas, J.: The enhanced flock of trackers. In Cipolla, R., Battiato, S., Farinella, G.M., eds.: Registration and Recognition in Images and Videos. Volume 532 of Studies in Computational Intelligence. Springer Berlin Heidelberg, Springer Berlin Heidelberg (January 2014) 113–136

68. Ma, C., Yang, X., Zhang, C., Yang, M.H.: Long-term correlation tracking. In: CVPR. (2015)
69. Wu, H., Sankaranarayanan, A.C., Chellappa, R.: Online empirical evaluation of tracking algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **32**(8) (2010) 1443–1458
70. Zhang, J., Ma, S., Sclaroff, S.: Meem: Robust tracking via multiple experts using entropy minimization. In: Computer Vision and Pattern Recognition. (2014)
71. Xuehan-Xiong, la Torre, F.D.: Supervised descent method and its application to face alignment. In: Computer Vision and Pattern Recognition. (2013)
72. Asthana, A., Zafeiriou, S., Cheng, S., Pantic, M.: Incremental face alignment in the wild. In: Computer Vision and Pattern Recognition. (2014)
73. Khan, M.H., Valstar, M.F., Pridmore, T.P.: Mts: A multiple temporal scale tracker handling occlusion and abrupt motion variation. In: Proceedings of the Asian Conference on Computer Vision. (2012) 86–97
74. Wang, X., Valstar, M., Martinez, B., Khan, H., Pridmore, T.: Tracking by regression with incrementally learned cascades. In: International Conference on Computer Vision. (2015)
75. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on **34**(7) (2012) 1409–1422
76. Lukezic, A., Cehovin, L., Kristan, M.: Deformable parts correlation filters for robust visual tracking. CoRR **abs/1605.03720** (2016)
77. Kristan, M., Perš, J., Sulič, V., Kovačič, S.: A graphical model for rapid obstacle image-map estimation from unmanned surface vehicles (2014)
78. Vojir, T.: Fast segmentation of object from background in given bounding box (2015)
79. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition. Volume 1. (June 2005) 886–893
80. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: Computer Vision ECCV 2014 Workshops. (2006) 244–253
81. Kalal, Z., Mikolajczyk, K., Matas, J.: Forward-backward error: Automatic detection of tracking failures. In: Computer Vision and Pattern Recognition. (2010)
82. Poostchi, M., Aliakbarpour, H., Viguier, R., Bunyak, F., Palaniappan, K., Seetharaman, G.: Semantic depth map fusion for moving vehicle detection in aerial video. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. (2016) 32–40
83. Poostchi, M., Palaniappan, K., Bunyak, F., Becchi, M., Seetharaman, G.: Efficient gpu implementation of the integral histogram. In: Asian Conference on Computer Vision, Springer (2012) 266–278
84. Poostchi, M., Bunyak, F., Palaniappan, K., Seetharaman, G.: Feature selection for appearance-based vehicle tracking in geospatial video. In: SPIE Defense, Security, and Sensing, International Society for Optics and Photonics (2013)
85. Palaniappan, K., Bunyak, F., Kumar, P., Ersoy, I., Jaeger, S., Ganguli, K., Haridas, A., Fraser, J., Rao, R., Seetharaman, G.: Efficient feature extraction and likelihood fusion for vehicle tracking in low frame rate airborne video. In: IEEE Conference on Information Fusion (FUSION). (2010) 1–8
86. Pelapur, R., Palaniappan, K., Seetharaman, G.: 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. (2012) 337–342

87. Danelljan, M., Khan, F.S., Felsberg, M., Van de Weijer, J.: Adaptive color attributes for real-time visual tracking. In: Computer Vision and Pattern Recognition. (2014)
88. Roffo, G., Melzi, S., Cristani, M.: Infinite feature selection. In: ICCV. (2015)
89. Roffo, G., Melzi, S.: Online feature selection for visual tracking. In: BMVC. (2016)
90. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online dictionary learning for sparse coding. In: Proceedings of the 26th Annual International Conference on Machine Learning, ICML. (2009) 689–696
91. Montero, A.S., Lang, J., Laganiere, R.: Scalable kernel correlation filter with sparse feature integration. In: The IEEE International Conference on Computer Vision (ICCV) Workshops. (December 2015) 24–31
92. Choi, J., Chang, H.J., Jeong, J., Demiris, Y., Choi, J.Y.: Visual tracking using attention-modulated disintegration and integration. In: CVPR. (2016)
93. Lebeda, K., Matas, J., Bowden, R.: Tracking the untrackable: How to track when your object is featureless. In: Proc. of ACCV DTCE. (2012)
94. Lebeda, K., Hadfield, S., Matas, J., Bowden, R.: Long-term tracking through failure cases. In: Proc. of ICCV VOT. (2013)
95. Lebeda, K., Hadfield, S., Matas, J., Bowden, R.: Texture-independent long-term tracking using virtual corners. IEEE Transactions on Image Processing (2016)
96. Nam, H., Baek, M., Han, B.: Modeling and propagating cnns in a tree structure for visual tracking. CoRR **abs/1608.07242** (2016)
97. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database a large-scale hierarchical image database. In: CVPR. (2009)
98. Ma, C., Huang, J.B., Yang, X., Yang, M.H.: Hierarchical convolutional features for visual tracking. In: ICCV. (2016)
99. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In: ECCV. (2012) 702–715
100. Vojir, T., Noskova, J., Matas, J.: Robust scale-adaptive mean-shift for tracking. Pattern Recognition Letters **49**(0) (2014) 250 – 258
101. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2010)
102. Vojir, T., Matas, J., Noskova, J.: Online adaptive hidden markov model for multitracker fusion. CoRR **abs/1504.06103** (2015)
103. Hare, S., Saffari, A., Torr, P.H.S.: Struck: Structured output tracking with kernels. In Metaxas, D.N., Quan, L., Sanfeliu, A., Gool, L.J.V., eds.: International Conference on Computer Vision, IEEE (2011) 263–270
104. Felsberg, M.: Enhanced distribution field tracking using channel representations. In: Vis. Obj. Track. Challenge VOT2013, In conjunction with ICCV2013. (2013)
105. Danelljan, M., Häger, G., Khan, F.S., Felsberg, M.: Learning spatially regularized correlation filters for visual tracking. In: International Conference on Computer Vision. (2015)
106. Ma, C., Huang, J.B., Yang, X., Yang, M.H.: Hierarchical convolutional features for visual tracking. In: International Conference on Computer Vision. (2015)
107. Čehovin, L., Kristan, M., Leonardis, A.: Robust visual tracking using an adaptive coupled-layer visual model. IEEE Trans. Pattern Anal. Mach. Intell. **35**(4) (2013) 941–953
108. Čehovin, L., Leonardis, A., Kristan, M.: Robust visual tracking using template anchors. In: WACV, IEEE (Mar 2016)

109. Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. International Journal of Computer Vision **77**(1-3) (2008) 125–141
110. Godec, M., Roth, P.M., Bischof, H.: Hough-based tracking of non-rigid objects. Comp. Vis. Image Understanding **117**(10) (2013) 1245–1256
111. Zhang, K., Zhang, L., Liu, Q., Zhang, D., Yang, M.H.: Fast visual tracking via dense spatio-temporal context learning. In: European Conference on Computer Vision. (2014) 127–141
112. Gao, J., Ling, H., Hu, W., Xing, J.: Transfer learning based visual tracking with gaussian processes regression. In: European Conference on Computer Vision. (2014) 188–203
113. Babenko, B., Yang, M.H., Belongie, S.: Robust object tracking with online multiple instance learning. IEEE Trans. Pattern Anal. Mach. Intell. **33**(8) (2011) 1619–1632
114. Nebehay, G., Pflugfelder, R.: Clustering of static-adaptive correspondences for deformable object tracking. In: Computer Vision and Pattern Recognition. (2015)