# The Visual Object Tracking VOT2014 challenge results

Matej Kristan[1], Roman Pflugfelder[2], Aleš Leonardis[3], Jiri Matas[4], Luka Čehovin[1], Georg Nebehay[2], Tomáš Vojíř[4], Gustavo Fernández[2], Alan Lukežič[1], Aleksandar Dimitriev[1], Alfredo Petrosino[5], Amir Saffari[6], Bo Li[7], Bohyung Han[8], CherKeng Heng[7], Christophe Garcia[9], Dominik Pangeršič[1], Gustav Häger[10], Fahad Shahbaz Khan[10], Franci Oven[1], Horst Possegger[11], Horst Bischof[11], Hyeonseob Nam[8], Jianke Zhu[12], JiJia Li[13], Jin Young Choi[14], Jin-Woo Choi[15], João F. Henriques[16], Joost van de Weijer[17], Jorge Batista[16], Karel Lebeda[18], Kristoffer Öfjäll[10], Kwang Moo Yi[19], Lei Qin[20], Longyin Wen[21], Mario Edoardo Maresca[5], Martin Danelljan[10], Michael Felsberg[10], Ming-Ming Cheng[22], Philip Torr[22], Qingming Huang[23], Richard Bowden[18], Sam Hare[24], Samantha YueYing Lim[7], Seunghoon Hong[8], Shengcai Liao[21], Simon Hadfield[18], Stan Z. Li[21], Stefan Duffner[9], Stuart Golodetz[22], Thomas Mauthner[11], Vibhav Vineet[22], Weiyao Lin[13], Yang Li[12], Yuankai Qi[23], Zhen Lei[21], and ZhiHeng Niu[7]

[1] University of Ljubljana, Slovenia
[2] Austrian Institute of Technology, AIT, Austria
[3] University of Birmingham, United Kingdom
[4] Czech Technical University, Czech Republic
[5] Parthenope University of Naples, Italy
[6] Affectv Limited, United Kingdom
[7] Panasonic R&D Center, Singapore
[8] POSTECH, Korea
[9] LIRIS, France
[10] Linköping University, Sweden
[11] Graz University of Technology, Austria
[12] Zhejiang University, China
[13] Shanghai Jiao Tong University, China
[14] Seoul National University, ASRI, Korea
[15] Electronics and Telecommunications Research Institute, Daejeon, Korea
[16] University of Coimbra, Portugal
[17] Universitat Autonoma de Barcelona, Spain
[18] University of Surrey, United Kingdom
[19] EPFL CVLab, Switzerland
[20] ICT CAS, China
[21] Chinese Academy of Sciences, China
[22] University of Oxford, United Kingdom
[23] Harbin Institute of Technology, China
[24] Obvious Engineering Limited, United Kingdom

**Abstract.** The Visual Object Tracking challenge 2014, VOT2014, aims at comparing short-term single-object visual trackers that do not apply pre-learned models of object appearance. Results of 38 trackers are

presented. The number of tested trackers makes VOT 2014 the largest benchmark on short-term tracking to date. For each participating tracker, a short description is provided in the appendix.
Features of the VOT2014 challenge that go beyond its VOT2013 predecessor are introduced: (i) a new VOT2014 dataset with full annotation of targets by rotated bounding boxes and per-frame attribute, (ii) extensions of the VOT2013 evaluation methodology, (iii) a new unit for tracking speed assessment less dependent on the hardware and (iv) the VOT2014 evaluation toolkit that significantly speeds up execution of experiments. The dataset, the evaluation kit as well as the results are publicly available at the challenge website[25].

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

## 1 Introduction

Visual tracking has received a significant attention over the last decade largely due to the diversity of potential applications which makes it a highly attractive research problem. The number of accepted motion and tracking papers in high profile conferences, like ICCV, ECCV and CVPR, has been consistently high in recent years (∼40 papers annually). For example, the primary subject area of twelve percent of papers accepted to ECCV2014 was motion and tracking. The significant activity in the field is also reflected in the abundance of review papers [23,43,22,29,44,65,40] summarizing the advances published in conferences and journals over the last fifteen years.

The use of different datasets and inconsistent performance measures across different papers, combined with the high annual publication rate, makes it difficult to follow the advances made in the field. Indeed, in computer vision fields like segmentation [19,18], optical-flow computation [3], change detection [24], the ubiquitous access to standard datasets and evaluation protocols has substantially contributed to cross-paper comparison [56]. Despite the efforts invested in proposing new trackers, the field suffers from a lack of established evaluation methodology.

Several initiatives have been put forward in an attempt to establish a common ground in tracking performance evaluation. Starting with PETS [66] as one of most influential performance analysis efforts, frameworks have been presented since with focus on surveillance systems and event detection, e.g., CAVIAR[26], i-LIDS [27], ETISEO[28], change detection [24], sports analytics (e.g., CVBASE[29]), faces, e.g. FERET [50] and [31], and the recent long-term tracking and detection of general targets[30] to list but a few.

[25] http://votchallenge.net

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

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

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

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

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

This paper discusses the VOT2014 challenge organized in conjunction with the ECCV2014 Visual object tracking workshop and the results obtained. The challenge considers single-camera, single-target, model-free, causal trackers, applied to short-term tracking. The *model-free* property means that the only supervised training example is provided by the bounding box in the first frame. The *short-term* tracking means that the tracker does not perform re-detection after the target is lost. Drifting off the target is considered a failure. The *causality* means that the tracker does not use any future frames, or frames prior to re-initialization, to infer the object position in the current frame. In the following we overview the most closely related work and then point out the contributions of VOT2014.

### 1.1 Related work

Recently, several attempts have been made towards benchmarking the class of trackers considered in this paper. Most notable are the online tracking benchmark (OTB) by Wu et al. [62] and the experimental survey based on Amsterdam Library of Ordinary Videos (ALOV) by Smeulders et al. [53]. Both benchmarks compare a number of recent trackers using the source code obtained from the original authors. All trackers were integrated into their experimental environment by the benchmark authors themselves and both report carefully setting the parameters. Nevertheless, it is difficult to guarantee equal quality of the parameter setting since, for some trackers, the operation requires thorough understanding.

The OTB [62] contains a dataset containing 50 sequences and annotates each sequence globally with eleven visual attributes. Sequences are not per-frame annotated. For example, a sequence has the “occlusion” attribute if the target is occluded anywhere in the sequence. The evaluation kit with pre-integrated trackers is publicly available. However, in our experience, the integration of third-party trackers into this kit is not straightforward due to a lack of standardization of the input/output communication between the tracker and the evaluation kit.

The ALOV [53] benchmark provides an impressive dataset with 315 sequences annotated with thirteen visual attributes. A drawback of this dataset is that some sequences contain cuts and ambiguously defined targets such as fireworks.

OTB [62] evaluates trackers using two measures: *precision* score and *success* score. Precision score represents the percentage of frames for which the center-distance error (e.g., [51,33]) is below 20 pixels. However, this threshold is strongly affected by the object size, which makes this particular measure quite brittle. A normalized center error measured during successful tracks may be used to alleviate the object size problem, however, the results in [53] show that the trackers do not differ significantly under this measure which makes it less appropriate for tracker comparison. The success plot represents the percentage of frames for which the overlap measure (e.g., [39,58]) exceeds a threshold, with respect to different thresholds. The area under the success plot is taken as an overall success measure. Čehovin et al. [58] have recently shown that this is simply an average overlap computed over the sequence. Alternatively, F-score based on Pascal overlap (threshold 0.5) is proposed in ALOV [53]. Note that the F-score based measure was originally designed for object detection. The threshold 0.5 is also rather high and there is no clear justification of why exactly this threshold should be used to compare trackers [62]. The ALOV [53] proposes an original approach to visualize tracking success. For each tracker, a performance measure is calculated per-sequence. These values are ordered from highest to lowest, thus obtaining a so-called survival curve and a test of statistical significance of differences is introduced to compare these curves across trackers. Special care has to be taken in interpreting the differences between these curves, as the orderings differ between trackers.

Both, the OTB and ALOV initialize the trackers at the beginning of the sequence and let them run until the end. While such a setup significantly simplifies the evaluation kit, it is not necessarily appropriate for short-term tracker evaluation, since short-term trackers are not required to perform re-detection. Therefore, the values of performance measures become irrelevant after the point of tracking failure, which significantly distorts the value of globally computed performance measure. The results are reported with respect to visual attributes in OTB and ALOV for in-depth analysis. However, most visual phenomena do not usually last throughout the entire sequence. For example, consider a tracker that performs poorly on a sequence with attribute occlusion according to a globally calculated performance measure. This might be interpreted as poor performance under occlusion, but actual occlusion might occur at the end of the sequence, while the poor performance is in fact due to some other effects occurring at the beginning of the sequence.

Collecting the results from the existing publications is an alternative for benchmarking trackers. Pang et al. [48] have proposed a page-rank-like approach to data-mine the published results and compile unbiased ranked performance lists. However, as the authors state in their paper, the proposed protocol is not appropriate for creating ranks of the recently published trackers due to the lack of sufficiently many publications that would compare these trackers.

The most closely related work is the recent visual object tracking challenge, VOT2013 [36]. The authors of that challenge provide the evaluation kit, a fully annotated dataset and an advanced performance evaluation methodology. In contrast to related benchmarks, the goal of VOT2013 was to have as many experiments as possible performed by the original authors of trackers while the results were analyzed by the VOT2013 committee. VOT2013 introduced several novelties in benchmarking short-term trackers: The evaluation kit is cross-platform, allowing easy integration with third-party trackers, the dataset is per-frame annotated with visual attributes and a state-of-the-art performance evaluation methodology was presented that accounts for statistical significance of the results on all measures. The results were published in a joint paper with over 50 co-authors [36], while the evaluation kit, the dataset, the tracking outputs and the code to reproduce all the results are made freely-available from the VOT2013 homepage[31].

[31] http://www.votchallenge.net/vot2013/

## 1.2 The VOT2014 challenge

The VOT2014 follows the VOT2013 challenge and considers the same class of trackers. The organisers of VOT2014 provided an evaluation kit and a dataset for automatic evaluation of the trackers. The evaluation kit records the output bounding boxes from the tracker, and if it detects tracking failure, re-initializes the tracker. The authors attending the challenge were required to integrate their tracker into the VOT2014 evaluation kit, which automatically performed a standardized experiment. The results were analyzed by the VOT2014 evaluation methodology.

Participants were expected to submit a single set of results per tracker. Participants who have investigated several trackers submitted a single result per tracker. Changes in the parameters did not constitute a different tracker. The tracker was required to run with fixed parameters on all experiments. The tracking method itself was allowed to internally change specific parameters, but these had to be set automatically by the tracker, e.g., from the image size and the initial size of the bounding box, and were not to be set by detecting a specific test sequence and then selecting the parameters that were hand-tuned to this sequence. Further details are available from the challenge sequence[32].

**The VOT2014 improves on VOT2013 in several aspects:**

- A new fully-annotated dataset is introduced. The dataset is per-frame annotated with visual properties, while the objects are annotated with rotated bounding boxes to more faithfully denote the target position.
- Unlike in VOT2013, trackers can predict the target position as a rotated bounding box as well.
- A new evaluation system is introduced that incorporates direct communication with the tracker [59] and offers faster execution of experiments and is backward compatible with VOT2013.
- The evaluation methodology from VOT2013 is extended to take into account that while the difference in accuracy of pair of trackers may be statistically significant, but negligibly small from perspective of ground truth ambiguity.
- A new unit for tracking speed is introduced that is less dependant on the hardware used to perform experiments.
- All accepted trackers are required to outperform the reference NCC tracker provided by the VOT2014 evaluation kit.
- A new web-based system for interactive exploration of the competition results has been implemented.

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

[32] http://www.votchallenge.net/vot2014/participation.html

## 2 The VOT2014 dataset

VOT2013 noted that a big dataset does not necessarily mean richness in visual properties and introduced a dataset selection methodology to compile a dataset that includes various real-life visual phenomena, while containing a small number of sequences to keep the time for performing the experiments reasonably low. We have followed the same methodology in compiling the VOT2014 dataset. Since the evaluation kit for VOT2014 is significantly more advanced than that of VOT2013, we were able to increase the number of sequences compared to VOT2013, while still keeping the time for experiments reasonably low.

The dataset was prepared as follows. The initial pool included 394 sequences, including sequences used by various authors in the tracking community, the VOT2013 benchmark [36], the recently published ALOV dataset [53], the Online Object Tracking Benchmark [62] and additional, so far unpublished, sequences. The set was manually filtered by removing sequences shorter than 200 frames, grayscale sequences, sequences containing poorly defined targets (e.g., fireworks) and sequences containing cuts. Ten global attributes were automatically computed for each of the 193 remaining sequences. In this way each sequence was represented as a 10-dimensional feature vector. Sequences were clustered in an unsupervised way using affinity propagation [21] into 12 clusters. From these, 25 sequences were manually selected such that the various visual phenomena like, occlusion, were still represented well within the selection.

The relevant objects in each sequence are manually annotated by bounding boxes. Most sequences came with axis-aligned bounding boxes placed over the target. For most frames, the axis-aligned bounding boxes approximated the target well with large percentage of pixels within the bounding box (at least $> 60\%$) belonging to the target. Some sequences contained elongated, rotating or deforming targets and these were re-annotated by rotated bounding boxes.

As in the VOT2013, we have manually or semi-manually labeled each frame in each selected sequence with five visual attributes that reflect a particular challenge in appearance attribute: (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 degradations, we denoted it as (vi) neutral. In the following we will use the term *attribute sequence* to refer to a set of frames with the same attribute pooled together from all sequences in the dataset.

## 3 Performance measures and evaluation methodology

As in VOT2013, the following two weakly correlated performance measures are used due to their high level of interpretability [58]: (i) accuracy and (ii) robustness. The accuracy measures how well the bounding box predicted by the tracker overlaps with the ground truth bounding box. On the other hand, the robustness measures how many times the tracker loses the target (fails) during tracking. A failure is indicated when the overlap measure becomes zero. To reduce the bias in robustness measure, the tracker is re-initialized five frames after the failure and ten frames after re-initialization are ignored in computation to further reduce the bias in accuracy measure [34]. Trackers are run 15 times on each sequence to obtain a better statistics on performance measures. The per-frame accuracy is obtained as an average over these runs. Averaging per-frame accuracies gives per-sequence accuracy, while per-sequence robustness is computed by averaging failure rates over different runs.

Apart from accuracy and robustness, the tracking speed is also an important property that indicates practical usefulness of trackers in particular applications. While accuracy and robustness results can be made comparable across different trackers by using the same experiments and dataset, the speed measurement depends on the programming language, implementation skills and most importantly, the hardware used to perform the experiments. To reduce the influence of hardware, the VOT2014 introduces a new unit for reporting the tracking speed. When an experiment is conducted with the VOT2014 evaluation kit, the kit benchmarks the machine by measuring the time required to perform a maximum pixel value filter on a grayscale image of size 600 × 600 with a 30 × 30 pixel window. The benchmark filter operation was coded in C by the VOT2014 committee. The VOT tracking speed is then reported by dividing the measured tracking time with the time required for the filtering operation. Thus the speed is reported in equivalent filter operations (EFO) which are defined by the VOT2014 evaluation kit.

### 3.1 Evaluation methodology

To address the unequal representation of the attributes in the sequences, the two measures are calculated only on the subset of frames in the dataset that contain that attribute (attribute subset). The trackers are ranked with respect to each measure separately on each attribute. The VOT2013 recognized that subsets of trackers might be performing equally well and this should be reflected in the ranks. Therefore, for each $i$-th tracker a set of equivalent trackers is determined. The corrected rank of the $i$-th tracker is obtained by averaging the ranks of these trackers including the considered tracker. The final ranking is obtained by averaging the ranks.

The equivalency of trackers is determined in VOT2013 by testing for the statistical significance of difference in performance of pairs of trackers. Separate statistical tests are applied for accuracy and robustness. The VOT2013 acknowledged that statistical significance of performance differences does not directly imply a practical difference [16], but did not address that. The practical difference is a level of difference that is considered negligibly small. This level can come from the noise in annotation, the fact that multiple ground truth annotations might be equally valid, or simply from the fact that very small differences in trackers are negligible from a practical point of view.

The VOT2014 extends the methodology by introducing tests of practical difference on tracking accuracy. In VOT2014, a pair of trackers is considered to perform equally well in accuracy if their difference in performance is not statistically significant or if it fails the practical difference test.

**Testing for practical difference:** Let $\phi_t(i)$ and $\phi_t(j)$ be the accuracies of the $i$-th and the $j$-th tracker at the $t$-th frame and let $\mu(i) = \frac{1}{T}\sum_{t=1}^{T}\phi_t(i)$ and $\mu(j) = \frac{1}{T}\sum_{t=1}^{T}\phi_t(j)$ be the average accuracies calculated over a sequence of $T$ frames. The trackers are said to perform differently if the difference of their averages is greater than a predefined threshold $\gamma$, i.e., $|\mu(i) - \mu(j)| > \gamma$, or, by defining $d_t(i,j) = \phi_t(i) - \phi_t(j)$, expanding the sums and pulling the threshold into the summation, $\frac{1}{T}|\sum_{t=1}^{T} d_t(i,j)/\gamma| > 1$. In VOT2014, the frames $t = 1 : T$ actually come from multiple sequences, and $\gamma$ values may vary over frames. Therefore, in VOT2014, a pair of trackers passes the test for practical difference if the following relation holds

$$\frac{1}{T}|\sum_{t=1}^{T} d_t(i,j)/\gamma_t| > 1, \tag{1}$$

where $\gamma_t$ is the practical difference threshold corresponding to $t$-th frame.

**Estimation of practical difference threshold:** The practical difference strongly depends on the target as well as the number of free parameters in the annotation model (i.e., in our case a rotated bounding box). Ideally a per-frame estimate of $\gamma$ would be required for each sequence, but that would present a significant undertaking. On the other hand, using a single threshold for entire dataset is too restrictive as the properties of targets vary across the sequences. A compromise can be taken in this case by computing one threshold per sequence. We propose selecting $M$ frames per sequence and have $J$ expert annotators place the bounding boxes carefully $K$ times on each frame. In this way $N = K \times J$ bounding boxes are obtained per frame. One of the bounding boxes can be taken as a possible ground truth and $N-1$ overlaps can be computed with the remaining ones. Since all annotations are considered "correct", any two overlaps should be considered equivalent, therefore the difference between these two overlaps is an example of negligibly small difference. By choosing each of the bounding boxes as ground truth, $M(N((N-1)^2 - N + 1))/2$ samples of differences are obtained per sequence. The practical difference threshold per sequence is estimated as the average of these values.

# 4 Analysis and results

## 4.1 Estimation of practical difference thresholds

The per sequence practical difference thresholds were estimated by the following experiment. For each sequence of the dataset, we identified four frames with axis-aligned ground-truth bounding boxes. The annotators were presented with two images side by side. The first image showed the first frame with overlaid ground-truth bounding box. This image served as a guidance on which part of the object should be annotated and was kept visible throughout the annotation of the four frames from the same sequence. These frames were displayed in the second image and the annotator was asked to place an axis-aligned bounding box on the target in each one. The process of annotation was repeated by each annotator three times. See Figure 1 In this setup a set of 15960 samples of differences was obtained per sequence and used to compute the practical difference threshold as discusses in Section 3.1.

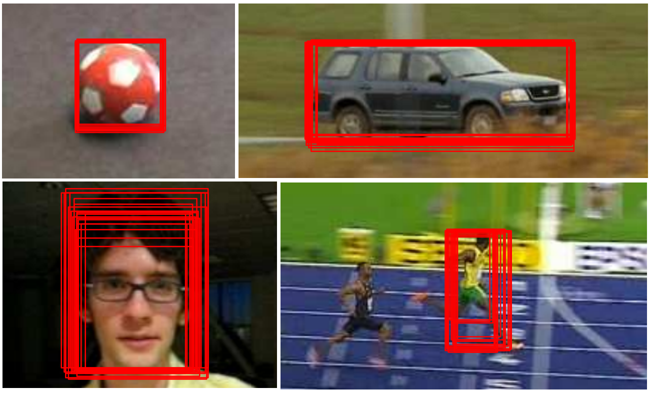

**Fig. 1.** Examples of diversity of bounding box annotations for different images.

Figure 2 shows boxplots of difference distributions w.r.t. sequences and a distribution over entire dataset. It is clear that the threshold on practical difference varies over the sequences. For the sequences containing rigid objects, the practical difference threshold is small (e.g., ball) and becomes large for sequences with deformable/articulated objects (e.g., bolt).

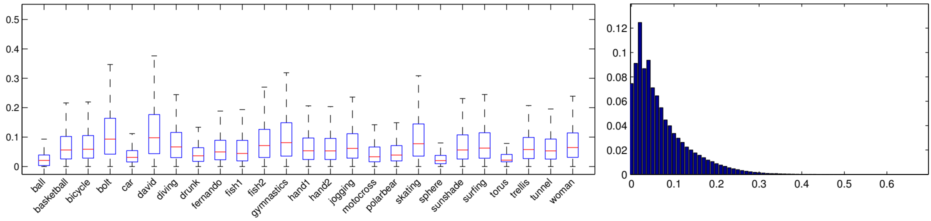

**Fig. 2.** Box plots of differences per sequence (left) and distribution of differences over entire dataset (right).

## 4.2 The VOT2014 experiments

The VOT2014 challenge includes the following two experiments:

- Experiment 1: This experiment runs a tracker on all sequences in the VOT2014 dataset by initializing it on the ground truth bounding boxes.
- Experiment 2: This experiment performs Experiment 1, but initializes with a noisy bounding box. By a noisy bounding box, we mean a randomly perturbed bounding box, where the perturbation is in the order of ten percent of the ground truth bounding box size.

In Experiment 2 there was a randomness in the initialization of the trackers. The bounding boxes were randomly perturbed in position and size by drawing perturbations uniformly from ±10% interval of the ground truth bounding box size, while the rotation was perturbed by drawing uniformly from ±0.1 radians. All the experiments were automatically performed by the evaluation kit[33]. A tracker was run on each sequence 15 times to obtain a better statistic on its performance. Note that it does not make sense to perform Experiment 1 multiple times for the deterministic trackers. In this case, the evaluation kit automatically detects whether the tracker is deterministic and reduces the number of repetitions accordingly.

### 4.3 Trackers submitted

Together 33 entries have been submitted to the VOT2014 challenge. Each submission included the binaries/source code that was used by the VOT2014 committee for results verification. The VOT2014 committee additionally contributed 5 baseline trackers. For these, the default parameters were selected, or, when not available, were set to reasonable values. Thus in total 38 trackers were included in the VOT2014 challenge. In the following we briefly overview the entries and provide the references to original papers. For the methods that are not officially published, we refer to the Appendix A instead.

Several tracker explicitly decomposed target into parts. These ranged from key-point-based trackers CMT (A.32), IIVTv2 (A.6), Matrioska (A.11) and its derivative MatFlow (A.13) to general part-based trackers LT-FLO (A.10), PT+ (A.27), LGT (A.33), OGT (A.30), DGT (A.31), ABS (A.2), while three trackers applied flock-of-trackers approaches FoT (A.22), BDF (A.12) and FRT (A.34). Several approaches were applying global generative visual models for target localization: a channel blurring approach EDFT (A.4) and its derivative qwsEDFT (A.3), GMM-based VTDMG (A.7), scale-adaptive mean shift eASMS (A.21), color and texture-based ACAT (A.20), HOG correlation-based SAMF (A.9), NCC based tracker with motion model IMPNCC (A.15), two color-based particle filters SIR-PF (A.1) and IPRT (A.18), a compressive tracker CT (A.35) and intensitiy-template-based pca tracker IVT (A.36). Two trackers applied fusion of flock-of-trackers and mean shift, HMM-TxD (A.23) and DynMS (A.26). Many trackers were based on discriminative models, i.e., boosting-based particle filter MCT (A.8), multiple-instance-learning-based tracker MIL (A.37), detection-based FSDT (A.29) while several applied regression-based techniques,

[33] https://github.com/vicoslab/vot-toolkit

i.e., variations of online structured SVM, Struck (A.16), aStruck (A.5), ThunderStruck (A.17), PLT_13 (A.14) and PLT_14 (A.19), kernelized-correlation-filter-based KCF (A.28), kernelized-least-squares-based ACT (A.24) and discriminative correlation-based DSST (A.25).

### 4.4 Results

The results are summarized in Table 1 and visualized by the AR rank plots [36,58], which show each tracker as a point in the joint accuracy-robustness rank space (Figure 3 and Figure 4). For more detailed rankings and plots please see the VOT2014 results homepage. At the time of writing this paper, the VOT committee was able to verify some of the submitted results by re-running parts of the experiments using the binaries of the submitted trackers. The verified trackers are denoted by * in Table 1. The AR rank plots for baseline experiment (Experiment 1) and noise experiment (Experiment 2) are shown in Figure 3, while per-visual-attribute ranking plots for the baseline experiment are shown in Figure 4.

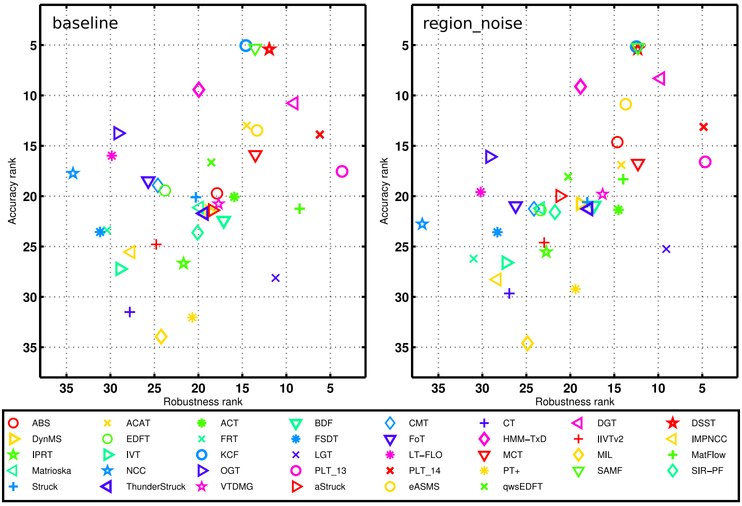

**Fig. 3.** The accuracy-robustness ranking plots with respect to the two experiments. Tracker is better if it resides closer to the top-right corner of the plot.

In terms of accuracy, the top performing trackers on both experiments, starting with best performing, are DSST, SAMF and KCF (Figure 3). Averaging together the accuracy and robustness, the improvement of DSST over the other two is most apparent at *size change* and *occlusion* attributes (Figure 4). For the noise experiment, these trackers remain the top performing, but the difference in accuracy is very small. In terms of robustness, the top performing trackers on the baseline experiment are PLT_13, PLT_14, MatFlow and DGT. These trackers come from two classes of trackers. The first two, PLT_13 and PLT_14 are extensions of the Struck [25] tracker that apply histogram backprojection as feature selection strategy in SVM training. The second two trackers are part-based trackers that apply different types of parts. MatFlow is extension of Matrioska [42] which applies a ORB/SURF keypoints and robust voting and matching techniques. On, the other hand, DGT decomposes target into parts by superpixels and applies graph-matching techniques to perform association of parts across the frames. The DGT is generally well ranked with respect to different visual properties, however, it significantly drops in performance during illumination changes (Figure 3). In the second experiment with initialization noise, MatFlow drops in ranks and the fourth-top tracker becomes the MCT which applies a holistic discriminative model and a motion model with particle filter. From Figure 4, we can see that a large majority of trackers, including NCC, performed equally well on frames denoted as *neutral* in terms of robustness, but differed quite significantly in accuracy.

The entries included several trackers from the same class. The top-performing trackers in accuracy, DSST, SAMF and KCF, formulate tracking as a ridge regression problem for correlation filter learning and apply HOG [13] in their visual model. The DSST is an extension of the MOSSE [5] that uses grayscale in addition to HOG, while SAMF and KCF seem to be extensions of [27] that address scale change. The similarity in design is reflected in the AR-rank plots as they form tight clusters in baseline as well as noise experiment. The PLT_13 and PLT_14 are also from the same class of trackers. The PLT_13 is the winner of the VOT2013 challenge [36] which does not adapt the target size, while the PLT_14 is an extension of PLT_13 that adapts the size as well. Interestingly, the PLT_14 does improve in accuracy compared to PLT_13, but sacrifices the robustness. In the noise experiment the PLT_14 is still outperforms the PLT_13 in accuracy, but the difference in robustness is reduced. MatFlow is an extension of Matrioska that applies a flock-of-trackers variant BDF. At a comparable accuracy ranks, the MatFlow by far outperforms the original Matrioska in robustness. The boost in robustness ranks might be attributed to addition of BDF, which is supported by the fact that BDF alone outperforms in robustness the FoT and trackers based on variations of FoT, i.e., aStruck, HMMTxD and dynMS. This speaks of resiliency to outliers in flock selection in BDF. Two trackers combine color-based mean shift with flow, i.e., dynMS and HMMTxD and obtain comparable ranks in robustness, however, the HMMTxD achieves a significantly higher accuracy rank, which might be due to considerably more sophisticated tracker merging scheme in HMMTxD. Both methods are outperformed in robustness by the scale-adaptive mean shift eASMS that applies motion prediction and colour space selection. While this version of mean shift performs quite well over a range of visual attributes, the performance drops in ranks drastically for *occlusion* and *illumination change*. The entries contained the original Struck and two variations, ThunderStruck and aStruck. ThunderStruck is a CUDA-speeded-up Struck and performs quite similarly to the original Struck in baseline and noise

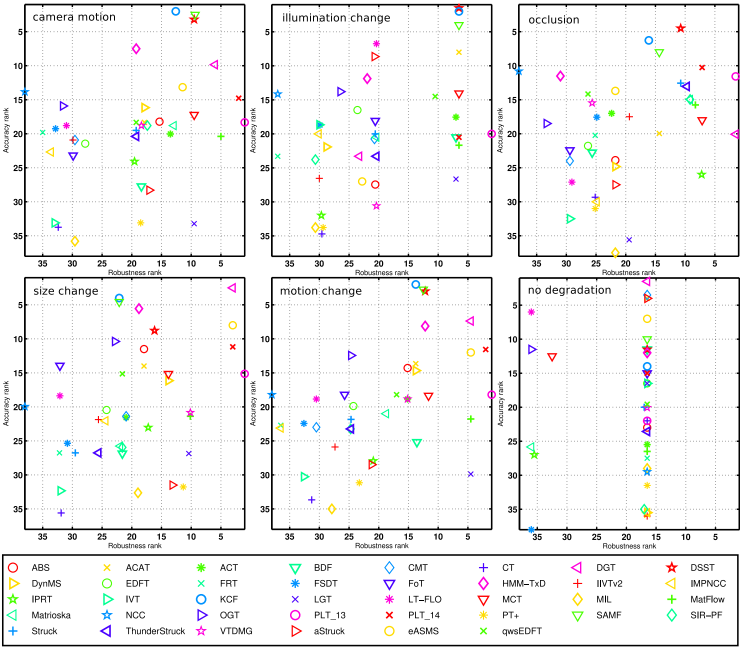

**Fig. 4.** The accuracy-robustness ranking plots of Experiment 1 with respect to the six sequence attributes. The tracker is better if it resides closer to the top-right corner of the plot.

experiment. The aStruck applies the flock-of-trackers for scale adaptation in Struck and improves in robustness on the baseline experiment, but is ranked lower in the noise experiment.

Note that majority of the trackers submitted to VOT2014 are fairly competitive trackers. This is supported by the fact that the trackers, that are often used as baseline trackers, NCC, MIL, CT, FRT and IVT, occupy the bottom-left part of the AR rank plots. Obviously these approaches vary in accuracy and robustness and are thus spread perpendicularly to the bottom-left-to-upper-right diagonal of AR-rank plots. In both experiments, the NCC is the least robust tracker. In summary, as in VOT2013 [36], the most robust tracker over individual visual properties remains the PLT_13 (A.14). This tracker is surpassed by far in combined accuracy-robustness rank by the trackers DSST (A.25), SAMF (A.9) and KCF (A.28), of which the DSST (A.25) outperforms the other two in robustness. According to the average ranks, the DSST (A.25) is thus the winner of VOT2014.

The VOT2014 evaluation kit also measured the times required to perform a repetition of each tracking run. For each tracker, the average tracking speed was estimated from these measurements. Table 1 shows the tracking speed per frame in the EFO units, introduced in Section 3. Note that the times for the Matlab trackers included an overhead required to load the Matlab environment, which

**Table 1.** Ranking results. The top, second and third lowest average ranks are shown in red, blue and green respectively. The $R_\Sigma$ column displays a joined ranking for both experiments, which were also used to order the trackers. The trackers that have been verified by the VOT committee are denoted by the asterisk *.

| | baseline | | | region_noise | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $R_A$ | $R_R$ | $R$ | $R_A$ | $R_R$ | $R$ | $R_\Sigma$ | **Speed** | **Impl.** |
| **DSST*** | 5.41 | 11.93 | **8.67** | 5.40 | 12.33 | 8.86 | **8.77** | 7.66 | Matlab & Mex |
| **SAMF*** | *5.30* | 13.55 | *9.43* | *5.24* | 12.30 | **8.77** | *9.10* | 1.69 | Matlab & Mex |
| **KCF*** | **5.05** | 14.60 | 9.82 | **5.17** | 12.49 | *8.83* | 9.33 | 24.23 | Matlab & Mex |
| **DGT** | 10.76 | 9.13 | 9.95 | 8.31 | 9.73 | 9.02 | 9.48 | 0.23 | C++ |
| **PLT_14*** | 13.88 | *6.19* | 10.03 | 13.12 | *4.85* | 8.99 | 9.51 | 62.68 | C++ |
| **PLT_13** | 17.54 | **3.67** | 10.60 | 16.60 | **4.67** | 10.63 | 10.62 | *75.92* | C++ |
| **eASMS*** | 13.48 | 13.33 | 13.40 | 10.88 | 13.70 | 12.29 | 12.85 | 13.08 | C++ |
| **HMM-TxD*** | 9.43 | 19.94 | 14.69 | 9.12 | 18.83 | 13.98 | 14.33 | 2.08 | C++ |
| **MCT** | 15.88 | 13.52 | 14.70 | 16.75 | 12.30 | 14.52 | 14.61 | 1.45 | C, C++ |
| **ACAT** | 12.99 | 14.49 | 13.74 | 16.90 | 14.20 | 15.55 | 14.65 | 3.24 | unknown |
| **MatFlow** | 21.25 | 8.49 | 14.87 | 18.33 | 13.99 | 16.16 | 15.51 | 19.08 | C++ |
| **ABS** | 19.72 | 17.88 | 18.80 | 14.63 | 14.65 | 14.64 | 16.72 | 0.62 | Matlab & Mex |
| **ACT** | 20.08 | 15.91 | 18.00 | 21.36 | 14.53 | 17.94 | 17.97 | 18.26 | Matlab |
| **qwsEDFT** | 16.65 | 18.53 | 17.59 | 18.07 | 20.24 | 19.15 | 18.37 | 3.88 | Matlab |
| **LGT*** | 28.12 | 11.22 | 19.67 | 25.25 | 9.08 | 17.17 | 18.42 | 1.23 | Matlab & Mex |
| **VTDMG** | 20.77 | 17.70 | 19.24 | 19.81 | 16.33 | 18.07 | 18.65 | 1.83 | C++ |
| **BDF** | 22.42 | 17.12 | 19.77 | 20.91 | 17.29 | 19.10 | 19.44 | 46.82 | C++ |
| **Struck** | 20.11 | 20.29 | 20.20 | 20.60 | 18.08 | 19.34 | 19.77 | 5.95 | C++ |
| **DynMS*** | 21.54 | 18.75 | 20.14 | 20.76 | 18.84 | 19.80 | 19.97 | 3.21 | Matlab & Mex |
| **ThunderStruck** | 21.71 | 19.35 | 20.53 | 21.26 | 17.92 | 19.59 | 20.06 | 19.05 | C++ |
| **aStruck*** | 21.41 | 18.40 | 19.90 | 19.98 | 21.19 | 20.59 | 20.24 | 3.58 | C++ |
| **Matrioska** | 21.15 | 19.86 | 20.50 | 21.19 | 23.39 | 22.29 | 21.40 | 10.20 | unknown |
| **SIR-PF** | 23.62 | 20.09 | 21.86 | 21.58 | 21.74 | 21.66 | 21.76 | 2.55 | Matlab & Mex |
| **EDFT** | 19.43 | 23.80 | 21.61 | 21.39 | 23.37 | 22.38 | 22.00 | 4.18 | Matlab |
| **OGT** | 13.76 | 29.15 | 21.45 | 16.09 | 29.16 | 22.63 | 22.04 | 0.39 | unknown |
| **CMT*** | 18.93 | 24.61 | 21.77 | 21.26 | 24.13 | 22.69 | 22.23 | 2.51 | Python, C++ |
| **FoT*** | 18.48 | 25.70 | 22.09 | 20.96 | 26.21 | 23.58 | 22.84 | **114.64** | C++ |
| **LT-FLO** | 15.98 | 29.84 | 22.91 | 19.59 | 30.20 | 24.90 | 23.90 | 1.10 | Matlab |
| **IPRT** | 26.68 | 21.68 | 24.18 | 25.54 | 22.73 | 24.14 | 24.16 | 14.69 | C, C++ |
| **IIVTv2** | 24.79 | 24.79 | 24.79 | 24.61 | 22.97 | 23.79 | 24.29 | 3.67 | C++ |
| **PT+** | 32.05 | 20.68 | 26.37 | 29.23 | 19.41 | 24.32 | 25.34 | 49.89 | C++ |
| **FSDT** | 23.55 | 31.17 | 27.36 | 23.58 | 28.29 | 25.93 | 26.65 | 1.47 | C++ |
| **IMPNCC** | 25.56 | 27.66 | 26.61 | 28.28 | 28.32 | 28.30 | 27.45 | 8.37 | Matlab |
| **IVT*** | 27.23 | 28.92 | 28.07 | 26.60 | 27.29 | 26.95 | 27.51 | 2.35 | Matlab & Mex |
| **FRT*** | 23.38 | 30.38 | 26.88 | 26.21 | 30.99 | 28.60 | 27.74 | 3.09 | C++ |
| **NCC*** | 17.74 | 34.25 | 26.00 | 22.78 | 36.83 | 29.80 | 27.90 | 6.88 | Matlab |
| **CT*** | 31.51 | 27.79 | 29.65 | 29.66 | 26.94 | 28.30 | 28.98 | 6.29 | C++ |
| **MIL*** | 33.95 | 24.22 | 29.09 | 34.61 | 24.87 | 29.74 | 29.41 | 1.94 | C++ |

depends mostly depends on hard drive reading speed which was measured during the evaluation. Table 1 shows adjusted times that accounted for this overhead. While one has to be careful with speed interpretation, we believe that these measurements still give a good comparative estimate of the trackers practical complexity. The trackers that stand out are the FoT and PLT_13, achieving speeds in range of around 100 EFO units (C++ implementations). To put this into perspective, a C++ implementation of a NCC tracker provided in the toolkit processes the VOT2014 dataset with an average of 220 frames per second on a laptop with an Intel Core i5 processor, which equals to approximately 80 EFO units.

## 5 Conclusions

This paper reviewed the VOT2014 challenge and its results. The challenge contains a annotated dataset of sequences in which targets are denoted by rotated bounding boxes to aid a precise analysis of the tracking results. All the sequences are labelled per-frame with attributes denoting various visual phenomena. The challenge also introduces a new *Matlab/Octave* evaluation kit for fast execution of experiments, proposes a new unit for measuring tracker speed, and extends the VOT2013 performance evaluation methodology to account for practical equivalence of tracker accuracy. The dataset, evaluation kit and VOT2014 results are publicly available from the challenge webpage.

The results of VOT2014 indicate that a winner of the challenge according to the average results is the DSST (A.25) tracker. The results also show that trackers tend to specialize either for robustness or accuracy. None of the trackers consistently outperformed the others by all measures at all sequence attributes. One class of trackers that consistently appears at the top of ranks are large margin regression-based trackers which apply global visual models[34], while the other class of trackers is the part-based trackers in which the target is considered as a set of parts or keypoints.

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. Following the very successful VOT2013, VOT2014 was the second attempt towards this. Our future work will be focused on revising the evaluation kit, dataset, performance measures, and possibly launching 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 projects J24284, J23607 and J2-2221 and European

[34] We consider the Structured SVM as regression from image intensities to image displacements.

Union seventh framework programme under grant agreement no 257906. Jiri Matas and Tomas Vojir were supported by CTU Project SGS13/142/OHK3/2T/13 and by the Technology Agency of the Czech Republic project TE01020415 (V3C – Visual Computing Competence Center).

# A Submitted trackers

In this appendix we provide a short summary of all trackers that were considered in the VOT2014 competition.

## A.1 Sequential Importance Re-sampling Particle Filter (SIR-PF)

*D. Pangeršič (dp3698@student.uni-lj.si)*

SIR-PF tracker makes Particle Filter approach more robust on sequences with fast motion and illumination changes. To do that, the tracker changes RGB data into YCbCr data and it generates a background model used by Comaniciu et al. [11]. The tracking task is done by using a window adaptation approach and a reference histogram adaptation to perform the matching between candidate objects.

## A.2 Appearance-Based Shape-Filter (ABS)

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

ABS tracker relies on appearance and shape cues for tracking. In particular, a histogram-based pixel-wise foreground is modelled to create a filter capturing discriminative object areas. This model combined with colour gradient templates to capture the object shape, allows to efficiently localize the object using mean shift tracking. ABS employs graph cut segmentation based on the pixel-wise foreground probabilities to adapt changes of object scales.

## A.3 Power Updated Weighted Comparison Enhanced Distribution Field Tracker (qwsEDFT)

*K. Öfjäll, M. Felsberg ({kristoffer.ofjall, michael.felsberg}@liu.se)*

A model matching approach where the tracked model is represented by a channel distribution field. Previous approaches such as DFT [52] and EDFT [20] do not exploit the possibilities of the model representation. The qwsEDFT tracker features a power update scheme and a standard deviation weighted comparison.

## A.4 Enhanced Distribution Fields for Tracking (EDFT)

*M. Felsberg (michael.felsberg@liu.se)*

The EDFT is a novel variant of the DFT tracker as proposed in [52]. EDFT derives an enhanced computational scheme by employing the theoretic connection between averaged histograms and channel representations. For further details, the interested reader is referred to [20].

### A.5 Scale adaptative Struck tracker (aStruck)

*A. Lukežič, L. Čehovin (alan.lukezic@gmail.com, luka.cehovin@fri.uni-lj.si)*

aStruck is a combination of optical-flow-based tracker and the discriminative tracker Struck [25]. aStruck uses low-level cues such as optical flow to handle significant scale changes. Besides, a framework akin to the FoT [60] tracker is utilized to robustly estimate the scale changes using the sparse Lucas-Kanade [41] pyramidal optical flow at points placed at a regular grid.

### A.6 Initialization Insensitive Visual Tracker Version 2 (IIVTv2)

*K. Moo Yi, J. Y. Choi (kwang.yi@epfl.ch, jychoi@snu.ac.kr)*

IIVTv2 is an implementation of the extended version of the initialization insensitive tracker [63]. The change from the original version include motion prior calculated from optical flow [54], normalization of the two proposed saliency weights in [63], inclusion of recent features in the feature database, and location based initialization of SURF [4] feature points.

### A.7 Visual Tracking with Dual Modeling through Gaussian Mixture Modeling (VTDMG)

*K. M. Yi, J. Y. Choi (kwang.yi@epfl.ch, jychoi@snu.ac.kr)*

VTDMG is an extended implementation of the method presented in [64]. Instead of using simple Gaussian modelling, VTDMG uses mixture of Gaussians. Besides, VTDMG models the target object and the background simultaneously and finds the target object through maximizing the likelihood defined using both models.

### A.8 Motion Context Tracker (MCT)

*S. Duffner, C. Garcia ({stefan.duffner, christophe garcia}@liris.cnrs.fr)*

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

### A.9 A kernel correlation filter tracker with Scale Adaptive and Feature Integration (SAMF)

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

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

### A.10 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*)

LT-FLO is designed to track texture-less objects. It significantly decreases reliance on texture by using edge-points instead of point features. The 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 reader is referred to [37] for details.

### A.11 Matrioska

*M. E. Maresca, A. Petrosino* (*{mariomaresca, petrosino}@uniparthenope.it*)

Matrioska [42] 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 fallback 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 Best Displacement Flow (BDF)

*M. E. Maresca, A. Petrosino* (*{mariomaresca, petrosino}@uniparthenope.it*)

Best Displacement Flow is a new short-term tracking algorithm based on the same idea of Flock of Trackers [60] in which a set of local tracker responses are robustly combined to track the object. BDF presents two main contributions: (i) BDF performs a clustering to identify the Best Displacement vector which is used to update the object's bounding box, and (ii) BDF performs a procedure named Consensus-Based Reinitialization used to reinitialize candidates which were previously classified as outliers.

### A.13 Matrioska Best Displacement Flow (MatFlow)

*M. E. Maresca, A. Petrosino* (*{mariomaresca, petrosino}@uniparthenope.it*)

MatFlow enhances the performance of the first version of Matrioska [42] with response given by aforementioned new short-term tracker BDF (see A.12). By default, MatFlow uses the trajectory given by Matrioska. In the case of a low confidence score estimated by Matrioska, MatFlow corrects the trajectory with the response given by BDF. Matrioska's confidence score is based on the number of key points found inside the object in the initialization.

### A.14 Single scale pixel based LUT tracker (2013) (PLT_13)

*C. Heng, S. YueYing Lim, Z. Niu, B. Li*
(*{hengcherkeng235, yueying53, niuzhiheng, libohit}@gmail.com*)

PLT runs a classifier at a fixed single scale for each test image, to determine the top scoring bounding box which is then the result of object detection. The classifier uses a binary feature vector constructed from colour, greyscale and gradient information. To select a small set of discriminative features, an online sparse structural SVM [25] is used. For more details, the interested reader is referred to [36].

### A.15 Improved Normalized Cross-Correlation Tracker (IMPNCC)

*A. Dimitriev (ad7414@student.uni-lj.si)*

This tracker improves the NCC tracker [7] in three ways: (i) by using a non-constant adaptation, the template is updated with new information; (ii) scale changes are handled by running an sliding window for the original image and two resized ones choosing the maxima of them; (iii) a Kalman Filter [30] is also used to smooth the trajectory and reduce drift. This improved tracker was based on the code of the original NCC tracker supplied with the VOT 2013 toolkit [35].

### A.16 Struck

*S. Hare, A. Saffari, P. H. S. Torr*
*(sam@samhare.net, amir@ymer.org, philip.torr@eng.ox.ac.uk)*

Struck [25] presents a framework for adaptive visual object tracking based on structured output prediction. By explicitly allowing the output space to express the needs of the tracker, need for an intermediate classification step is avoided. The method uses a kernelized structured output support vector machine (SVM), which is learned online to provide adaptive tracking.

### A.17 ThunderStruck

*S. Hare, A. Saffari, S. Golodetz, V. Vineet, M. Cheng, P. H. S. Torr*
*(sam@samhare.net, amir@ymer.org, sgolodetz@gxstudios.net, vibhav.vineet@gmail.com, cmm.thu@qq.com, philip.torr@eng.ox.ac.uk)*

ThunderStruck is a CUDA-based implementation of the Struck tracker presented by Hare et al. [25]. As with the original Struck, tracking is performed using a structured output SVM. On receiving a new frame, the tracker predicts a bounding box for the object in the new frame by sampling around the old object position and picking the location that maximises the response of the current SVM. The SVM is then updated using LaRank [6]. A support vector budget is used to prevent the unbounded growth in the number of support vectors that would otherwise occur during tracking.

### A.18 Iterative particle repropagation tracker (IPRT)

*J.-W. Choi (jwc@etri.re.kr)*

IPRT is a particle filter based tracking method inspired by colour-based particle filter [47,49] with the proposed iterative particle re-propagation. Multiple HSV colour histograms with $6 \times 6 \times 6$ bins are used as an observation model. In order to reduce the chance of tracker drift, the states of particles are saved before propagation. If tracker drift is detected, particles are restored and re-propagated. The tracker drift is detected by a colour histogram similarity measure derived from the Bhattacharyya coefficient.

### A.19 Size-adaptive Pixel based LUT tracker (2014) (PLT_14)

*C. Heng, S. YueYing Lim, Z. Niu, B. Li*
*({hengcherkeng235, yueying53, niuzhiheng, libohit}@gmail.com)*

PLT_14 tracker is an improved version of PLT tracker used in VOT 2013 [?], with size adaptation for the tracked object. PLT_14 uses discriminative pixel features to compute the scanning window score in a tracking-by-detection framework. The window score is 'back projected' to its contributing pixels. For each pixel, the pixel score is computed by summing the back projected scores of the windows that use this pixel. This score contributes to estimate which pixel belongs to the object during tracking and determine a best bounding box.

## A.20 Augment Color Attributes Tracker (ACAT)

*L. Qin, Y. Qi, Q.g Huang*
(*qinlei@ict.ac.cn, {yuankai.qi, qingming.huang}@vipl.ict.ac.cn*)

Augment Color Attributes Tracker is based on the method of Colour Attributes Tracker (CAT) [15]. Colour features used in CAT is just colour. CAT extends CSK tracker [26] to multi-channel colour features and it also augments CAT by including texture features and shape features.

## A.21 Enhanced Scale Adaptive MeanShift (eASMS)

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

eASMS tracker is a variation of the scale adaptive mean-shift [11,10,12]. It enhances its performance by utilizing background subtraction and motion prediction to allow the mean-shift procedure to converge in presence of high background clutter. The eASMS tracker also incorporates automatic per-frame selection of colour space (from pool of the available ones, e.g. HSV, Luv, RGB).

## A.22 Flock of Trackers (FoT)

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

The Flock of Trackers (FoT) [60] is a tracking framework where the object motion is estimated from the displacements or using a number of local trackers covering the object. Each local tracker is attached to a certain area specified in the object coordinate frame. The FoT object motion estimate is robust due to the combination of local tracker motions.

## A.23 Hidden Markov Model Fusion of Tracking and Detection (HMM-TxD)

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

The HMM-TxD tracker is a novel method for fusing diverse trackers by utilizing a hidden Markov model (HMM). The HMM estimates the changes in individual tracker performance, its state corresponds to a binary vector predicting failure of individual trackers. The proposed approach relies on a high-precision low-recall detector that provides a source of independent information for a modified Baum-Welch algorithm that updates the Markov model. Two trackers were used in the HMM-TxD: Flock of Trackers [60] estimating similarity and scale adaptive mean-shift tracker [11,10,12].

### A.24 Adaptive Color Tracker (ACT)

*M. Danelljan, F. S. Khan, M. Felsberg, J. van de Weijer*
(*{fmartin.danelljan, fahad.khan, michael.felsberg}@liu.se, joost@cvc.uab.es*)

The Adaptive Color Tracker (ACT) [15] extends the CSK tracker [26] 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 [61] are applied for image representation, and (iii) ACT employs a dynamically adaptive scheme for selecting the most important combinations of colours for tracking.

### A.25 Discriminative Scale Space Tracker (DSST)

*M. Danelljan, G. Häger, F. S. Khan, M. Felsberg*
(*fmartin.danelljan@liu.se, hager.gustav@gmail.com,*
*{fahad.khan, michael.felsberg}@liu.se*)

The Discriminative Scale Space Tracker (DSST) [14] extends the Minimum Output Sum of Squared Errors (MOSSE) tracker [5] with robust scale estimation. The MOSSE tracker works by training a discriminative correlation filter on a set of observed sample grey scale patches. This correlation filter is then applied to estimate the target translation in the next frame. 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.26 Dynamic Mean Shift (DynMS)

*Franci Oven, Matej Kristan* (*frenk.oven@gmail.com, matej.kristan@fri.uni-lj.si*)

DynMS is a Mean Shift tracker [9] with an isotropic kernel bootstrapped by a flock-of-features (FoF) tracker. The FoF tracker computes a sparse Lucas Kanade flow [41] and uses MLESAC [55] with similarity transform to predict the target position. The estimated states of the target are merged by first moving to estimated location of FoF and then using Mean Shift to find the object.

### A.27 Pixeltrack+ (PT+)

*S. Duffner, C. Garcia* (*{stefan.duffner, christophe garcia}@liris.cnrs.fr*)

Pixeltrack+ is based on the Pixeltrack tracking algorithm [17]. The algorithm uses two components: a detector that makes use of the generalised Hough transform with pixel-based descriptors, and a probabilistic segmentation method based on global models for foreground and background. The original Pixeltrack method [17] has been improved to cope with varying scale by estimating the objects size based on the current segmentation.

### A.28 Kernelized Correlation Filter (KCF) tracker (KCF)

*J. F. Henriques, J. Batista* (*{henriques, batista}@isr.uc.pt*)

This tracker is basically a Kernelized Correlation Filter [27] operating on simple HOG features. The KCF is equivalent to a Kernel Ridge Regression trained with thousands of sample patches around the object at different translations. The improvements over the previous version [27] are multi-scale support, sub-cell peak estimation and replacing the model update by linear interpolation with a more robust update scheme [15].

### A.29 Adaptive Feature Selection and Detection Based Tracker (FSDT)

*J. Li, W. Lin ({lijijia, wylin}@sjtu.edu.cn)*

FSDT is a tracking-by detection method that exploits the detection results to modify the tracker in the process of tracking. The detection part maintains a variable features pool where features are added or deleted as frames are processed. The tracking part implements a rough estimation of object tracked mainly by the velocity of objects. Afterwards, detection results are used to modify the rough tracked object position and to generate the final tracking result.

### A.30 Online Graph-based Tracking (OGT)

*H. Nam, S. Hong, B. Han ({namhs09, maga33, bhhan}@postech.ac.kr)*

OGT [45] is an online Orderless Model-Averaged tracking (OMA) [28]. OGT uses an unconventional graphical model beyond chain models, where each node has a single outgoing edge but may have multiple incoming edges. In this framework, the posterior is estimated by propagating multiple previous posteriors to the current frame along the identified graphical model, where the propagation is performed by a patch matching technique [32] as in [28]. The propagated densities are aggregated by weighted Bayesian model averaging, where the weights are determined by the tracking plausibility.

### A.31 Dynamic Graph based Tracker (DGT)

*L. Wen, Z. Lei, S. Liao, S. Z. Li (lywen, zlei, scliao, szli}@nlpr.ia.ac.cn)*

DGT is an improvement of the method proposed in [8]. The tracking problem is formulated as a matching problem between the target graph G(V;E) and the candidate graph G0(V0;E0). SLIC algorithm is used to oversegment the searching area into multiple parts (superpixels), and exploit the Graph Cut approach to separate the foreground superpixels from background superpixels. An affinity matrix based on motion, appearance and geometric constraints is built to describe the reliability of the matchings. The optimal matching from candidate superpixels is found from the affinity matrix applying the spectral technique [38]. The location of the target is voted by a series of the successfully matched parts according to their matching reliability.

### A.32 Consensus-based Matching and Tracking (CMT)

*G. Nebehay, R. Pflugfelder ({Georg.Nebehay.fl, Roman.Pflugfelder}@ait.ac.at)*

The CMT tracker [46] is a key point-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 key point detectors and binary descriptors allows the current implementation to run in real-time.

## A.33 Local-Global Tracking (LGT)

*L. Čehovin, M. Kristan, A. Leonardis*
({*luka.cehovin, matej.kristan, ales.leonardis*}*@fri.uni-lj.si*)

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 [57] for details.

## A.34 Fragment Tracking (FRT)

*VOT 2014 Technical Committee*

The FRT tracker [1] 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. We then minimize a robust statistic 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.35 Compressive Tracking (CT)

*VOT 2014 Technical Committee*

The CT tracker [67] uses an appearance model based on features extracted from the multi-scale image feature space with data-independent basis. It employs non-adaptive random projections that preserve the structure of the image feature space of objects. A very sparse measurement matrix is adopted to efficiently extract the features for the appearance model. Samples of foreground and background are compressed using the same sparse measurement matrix. The tracking task is formulated as a binary classification via a naive Bayes classifier with online update in the compressed domain.

## A.36 Incremental Learning for Robust Visual Tracking (IVT)

*VOT 2014 Technical Committee*

The idea of the IVT tracker [51] is to incrementally learn a low-dimensional subspace representation, adapting online to changes in the appearance of the target. The model update, based on incremental algorithms for principal component analysis, includes two features: a method for correctly updating the sample mean, and a forgetting factor to ensure less modelling power is expended fitting older observations.

## A.37 Multiple Instance Learning Tracking (MIL)

*VOT 2014 Technical Committee*

MIL [2] is a tracking-by-detection approach. MIL uses Multiple Instance Learning instead of traditional supervised learning methods and shows improved robustness to inaccuracies of the tracker and to incorrectly labeled training samples.

## References

1. Adam, A., Rivlin, E., Shimshoni, I.: Robust Fragments-based Tracking using the Integral Histogram. In: CVPR. vol. 1, pp. 798–805. IEEE Computer Society (Jun 2006)
2. Babenko, B., Yang, M.H., Belongie, S.: Robust object tracking with online multiple instance learning. IEEE Trans. Pattern Anal. Mach. Intell. 33(8), 1619–1632 (2011)
3. Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., R., S.: A database and evaluation methodology for optical flow. Int. J. Comput. Vision 92(1), 1–31 (2011)
4. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. Lecture notes in computer science 3951, 404 (2006)
5. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: Comp. Vis. Patt. Recognition (2010)
6. Bordes, A., Bottou, L., Gallinari, P., Weston, J.: Solving multiclass support vector machines with larank. In: Proceedings of the 24th International Conference on Machine Learning (ICML) (2007)
7. Briechle, K., Hanebeck, U.D.: Template matching using fast normalized cross correlation. In: Aerospace/Defense Sensing, Simulation, and Controls, International Society for Optics and Photonics. pp. 95–102 (2001)
8. Cai, Z., Wen, L., Yang, J., Lei, Z., Li, S.Z.: Structured visual tracking with dynamic graph. In: Proceedings Asian Conference on Computer Vision (ACCV). pp. 86–97 (2012)
9. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(5), 603–619 (2002)
10. Comaniciu, D., Ramesh, V., Meer, P.: The variable bandwidth mean shift and data-driven scale selection. In: Int. Conf. Computer Vision. vol. 1, pp. 438 – 445 (2001)
11. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. Pattern Analysis and Machine Intelligence, IEEE Transactions on 25(5), 564–577 (2003)
12. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: Comp. Vis. Patt. Recognition. vol. 2, pp. 142–149 (2000)
13. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Comp. Vis. Patt. Recognition. vol. 1, pp. 886–893 (June 2005)
14. 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)
15. Danelljan, M., Khan, F.S., Felsberg, M., Van de Weijer, J.: Adaptive color attributes for real-time visual tracking. In: 2014 Conference on Computer Vision and Pattern Recognition CVPR (2014)
16. Demšar, J.: On the appropriateness of statistical tests in machine learning. In: Workshop on Evaluation Methods for Machine Learning ICML (2008)
17. Duffner, S., Garcia, C.: Pixeltrack: a fast adaptive algorithm for tracking non-rigid objects. In: Proceedings of the International Conference on Computer Vision (ICCV). pp. 2480–2487 (2013)
18. Everingham, M., Eslami, S. M. A.and Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes challenge - a retrospective. Int. J. Comput. Vision (2014)
19. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. Int. J. Comput. Vision 88(2), 303–338 (Jun 2010)

20. Felsberg, M.: Enhanced distribution field tracking using channel representations. In: Vis. Obj. Track. Challenge VOT2013, In conjunction with ICCV2013 (2013)
21. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. Science 315, 972–976 (2007)
22. 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. pp. 166–173 (2003)
23. Gavrila, D.M.: The visual analysis of human movement: A survey. Comp. Vis. Image Understanding 73(1), 82–98 (1999)
24. Goyette, N., Jodoin, P.M., Porikli, F., Konrad, J., Ishwar, P.: Changedetection.net: A new change detection benchmark dataset. In: CVPR Workshops. pp. 1–8. IEEE (2012)
25. 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.) Int. Conf. Computer Vision. pp. 263–270. IEEE (2011)
26. Henriques, F., Caseiro, R., Martins, P., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In: 2012 European Conference on Computer Vision ECCV (2012)
27. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. 1(3), 125–141 (2014)
28. Hong, S., Kwak, S., Han, B.: Orderless tracking through model-averaged posterior estimation. In: Proceedings of the International Conference on Computer Vision (ICCV) (2013)
29. 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), 334–352 (2004)
30. Kalman, R.E.: A new approach to linear filtering and prediction problems. Trans. ASME, J. Basic Engineering 82, 34–45 (1960)
31. 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), 319–336 (2009)
32. Korman, S., Avidan, S.: Coherency sensitive hashing. In: Proceedings of the International Conference on Computer Vision (ICCV) (2011)
33. Kristan, M., Perš, J., Perše, M., Kovačič, S.: Closed-world tracking of multiple interacting targets for indoor-sports applications. Comput. Vision Image Understanding 113(5), 598–611 (May 2009)
34. Kristan, M., Pflugfelder, R., Leonardis, A., Matas, J., Porikli, F., Cehovin, L., Nebehay, G., Fernandez, G., Vojir, T.: The vot2013 challenge: overview and additional results. In: Computer Vision Winter Workshop (2014)
35. Kristan, M., Čehovin, L.: Visual Object Tracking Challenge (VOT2013) Evaluation Kit. Visual Object Tracking Challenge (2013)
36. Kristan, M., Pflugfelder, R., Leonardis, A., Matas, J., Porikli, F., Čehovin, L., Nebehay, G., Fernandez, G., Vojir, T., Gatt, A., Khajenezhad, A., Salahledin, A., Soltani-Farani, A., Zarezade, A., Petrosino, A., Milton, A., Bozorgtabar, B., Li, B., Chan, C.S., Heng, C., Ward, D., Kearney, D., Monekosso, D., Karaimer, H.C., Rabiee, H.R., Zhu, J., Gao, J., Xiao, J., Zhang, J., Xing, J., Huang, K., Lebeda, K., Cao, L., Maresca, M.E., Lim, M.K., Helw, M.E., Felsberg, M., Remagnino, P., Bowden, R., Goecke, R., Stolkin, R., Lim, S.Y., Maher, S., Poullot, S., Wong,

S., Satoh, S., Chen, W., Hu, W., Zhang, X., Li, Y., Niu, Z.: The Visual Object Tracking VOT2013 challenge results. In: ICCV Workshops. pp. 98–111 (2013)
37. Lebeda, K., Bowden, R., Matas, J.: Long-term tracking through failure cases. In: Vis. Obj. Track. Challenge VOT2013, In conjunction with ICCV2013 (2013)
38. Leordeanu, M., Hebert, M.: A spectral technique for correspondence problems using pairwise constraints. In: Proceedings of the International Conference on Computer Vision (ICCV). vol. 2, pp. 1482–1489 (2005)
39. Li, H., Shen, C., Shi, Q.: Real-time visual tracking using compressive sensing. In: Comp. Vis. Patt. Recognition. pp. 1305–1312. IEEE (2011)
40. 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)
41. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Imaging Understanding Workshop. pp. 121–130 (1981)
42. Maresca, M.E., Petrosino, A.: Matrioska: A multi-level approach to fast tracking by learning. In: Proc. Int. Conf. Image Analysis and Processing. pp. 419–428 (2013)
43. Moeslund, T.B., Granum, E.: A survey of computer vision-based human motion capture. Comp. Vis. Image Understanding 81(3), 231–268 (March 2001)
44. 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), 90–126 (November 2006)
45. Nam, H., Hong, S., Han, B.: Online graph-based tracking. In: To appear in 2014 European Conference on Computer Vision ECCV (2014)
46. Nebehay, G., Pflugfelder, R.: Consensus-based matching and tracking of keypoints for object tracking. In: Winter Conference on Applications of Computer Vision. IEEE (Mar 2014)
47. Nummiaro, K., Koller-Meier, E., Van Gool, L.: Color features for tracking non-rigid objects. Chinese J. Automation 29(3), 345–355 (May 2003)
48. Pang, Y., Ling, H.: Finding the best from the second bests – inhibiting subjective bias in evaluation of visual tracking algorithms. In: Int. Conf. Computer Vision (2013)
49. Pérez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-based probabilistic tracking. In: Proc. European Conf. Computer Vision. vol. 1, pp. 661–675 (2002)
50. 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), 1090–1104 (2000)
51. Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. Int. J. Comput. Vision 77(1-3), 125–141 (2008)
52. Sevilla-Lara, L., Learned-Miller, E.G.: Distribution fields for tracking. In: Comp. Vis. Patt. Recognition. pp. 1910–1917. IEEE (2012)
53. Smeulders, A.W.M., Chu, D.M., Cucchiara, R., Calderara, S., Dehghan, A., Shah, M.: Visual Tracking: an Experimental Survey. TPAMI (2013)
54. Tomasi, C., Kanade, L.: Detection and tracking of point features. Tech. rep., Carnegie Mellon University (1991)
55. Torr, P.H., Zisserman, A.: MLESAC: A new robust estimator with application to estimating image geometry. Computer Vision and Image Understanding 78(1), 138–156 (2000)
56. Torralba, A., Efros, A.A.: Unbiased look at dataset bias. In: Comp. Vis. Patt. Recognition. pp. 1521–1528. IEEE (2011)
57. Čehovin, L., Kristan, M., Leonardis, A.: Robust Visual Tracking using an Adaptive Coupled-layer Visual Model. TPAMI 35(4), 941–953 (Apr 2013)

58. Čehovin, L., Kristan, M., Leonardis, A.: Is my new tracker really better than yours? In: IEEE WACV2014 (2014)
59. Čehovin, L.: Trax: Visual tracking exchange protocol (Apr 2014)
60. Vojir, T., Matas, J.: Robustifying the flock of trackers. In: Comp. Vis. Winter Workshop. pp. 91–97. IEEE (2011)
61. Van de Weijer, J., Schmid, C., Verbeek, J.J., Larlus, D.: Learning color names for real-world applications. IEEE Transaction in Image Processing 18(7), 1512–1524 (2009)
62. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: Comp. Vis. Patt. Recognition (2013)
63. Yi, K.M., Jeong, H., Heo, B., Chang, H.J., Choi, J.Y.: Initialization-insensitive visual tracking through voting with salient local features. In: 2013 IEEE International Conference on Computer Vision ICCV. pp. 2912–2919 (2013)
64. Yi, K.M., Jeong, H., Kim, S.W., Choi, J.Y.: Visual tracking with dual modeling. In: Proceedings of the 27th Conference on Image and Vision Computing New Zealand, IVCNZ 12. pp. 25–30 (2012)
65. Yilmaz, A., Shah, M.: Object tracking: A survey. Journal ACM Computing Surveys 38(4) (2006)
66. 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. pp. 317–324 (2005)
67. Zhang, K., Zhang, L., Yang, M.H.: Real-time compressive tracking. In: Proc. European Conf. Computer Vision. pp. 864–877. Lecture Notes in Computer Science, Springer (2012)