The Visual Object Tracking VOT2013 challenge results

Matej Kristan [a] Roman Pflugfelder [b] Aleš Leonardis [c] Jiri Matas [d] Fatih Porikli [e]
Luka Čehovin [a] Georg Nebehay [b] Gustavo Fernandez [b] Tomáš Vojíř [d] Adam Gatt [f]
Ahmad Khajenezhad [g] Ahmed Salahledin [h] Ali Soltani-Farani [g] Ali Zarezade [g]
Alfredo Petrosino [i] Anthony Milton [j] Behzad Bozorgtabar [k] Bo Li [l]
Chee Seng Chan [m] CherKeng Heng [l] Dale Ward [j] David Kearney [j]
Dorothy Monekosso [n] Hakki Can Karaimer [o] Hamid R. Rabiee [g] Jianke Zhu [p]
Jin Gao [q] Jingjing Xiao [c] Junge Zhang [r] Junliang Xing [q] Kaiqi Huang [r]
Karel Lebeda [s] Lijun Cao [r] Mario Edoardo Maresca [i] Mei Kuan Lim [m]
Mohamed ELHelw [h] Michael Felsberg [t] Paolo Remagnino [u] Richard Bowden [s]
Roland Goecke [v] Rustam Stolkin [c] Samantha YueYing Lim [l] Sara Maher [h]
Sebastien Poullot [w] Sebastien Wong [f] Shin'ichi Satoh [x] Weihua Chen [r] Weiming Hu [q]
Xiaoqin Zhang [q] Yang Li [p] ZhiHeng Niu [l]

Abstract

Visual tracking has attracted a significant attention in the last few decades. The recent surge in the number of publications on tracking-related problems have made it almost impossible to follow the developments in the field. One of the reasons is that there is a lack of commonly accepted annotated data-sets and standardized evaluation protocols that would allow objective comparison of different tracking methods. To address this issue, the Visual Object Tracking (VOT) workshop was organized in conjunction with ICCV2013. Researchers from academia as well as industry were invited to participate in the first VOT2013 challenge which aimed at single-object visual trackers that do not apply pre-learned models of object appearance (model-free). Presented here is the VOT2013 benchmark dataset for evaluation of single-object visual trackers as well as the results obtained by the trackers competing in the challenge. In contrast to related attempts in tracker benchmarking, the dataset is labeled per-frame by visual attributes that indicate occlusion, illumination change, motion change, size change and camera motion, offering a more systematic comparison of the trackers. Furthermore, we have designed an automated system for performing and evaluating the experiments. We present the evaluation protocol of the VOT2013 challenge and the results of a comparison of 27 trackers on the benchmark dataset. The dataset, the evaluation tools and the tracker rankings are publicly available from the challenge website[1].

[a] University of Ljubljana, Slovenia
[b] Austrian Institute of Technology, Austria
[c] University of Birmingham, United Kingdom
[d] Czech Technical University in Prague, Czech Republic
[e] Australian National University
[f] DSTO, Edinburgh, SA, Australia
[g] Sharif University of Technology, Tehran, Iran
[h] Center for Informatics Science, Nile University, Giza, Egypt
[i] Parthenope University of Naples, Italy
[j] University of South Australia, Mawson Lakes, SA, Australia
[k] Vision and Sensing, ESTeM, University of Canberra, Australia
[l] Panasonic R&D Center, Singapore
[m] CISP, University of Malaya, Malaysia
[n] Eng. Design and Math., University of West England, United Kingdom
[o] Izmir Institute of Technology, Turkey
[p] College of Computer Science, Zhejiang University, China
[q] NLPR, Institute of Automation, CAS, Beijing, China
[r] Chinese Academy of Sciences, China
[s] University of Surrey, United Kingdom
[t] Linköping University, Sweden
[u] Robotic Vision Team, Kingston University, United Kingdom
[v] IHCC, CECS, Australian National University, Australia
[w] NII, JFLI, Hitotsubashi, Japan
[x] NII, Hitotsubashi, Japan

[1] http://votchallenge.net

1. Introduction

Visual tracking is a rapidly evolving field of computer vision that has been increasingly attracting attention of the vision community. One reason is that it offers many challenges as a scientific problem. Second, it is a part of many higher-level problems of computer vision, such as motion analysis, event detection and activity understanding. Furthermore, the steady advance of HW/SW technology in terms of computational power, form factor and price, opens vast application potential for tracking algorithms. Applications include surveillance systems, transport, sports analytics, medical imaging, mobile robotics, film post-production and human-computer interfaces.

In this paper, we focus on single-object trackers that do not apply pre-learned models of the object appearance (model-free), since they offer a particularly large application domain. The activity in the field is reflected by the abundance of new tracking algorithms presented and evaluated in journals and at conferences, and summarized in the many survey papers, e.g., [17, 35, 14, 22, 36, 52, 32]. A review of recent high-profile conferences like ICCV, ECCV and CVPR shows that the number of accepted tracking papers has been consistently high (40-50 annually). At the ICCV2013 conference, for example, 38 papers with the topic motion and tracking were published. The topic was the third most popular if measured by the number of accepted papers.

Evaluation of new tracking algorithms, and their comparison to the state-of-the-art, depends on three essential components: (1) a dataset, (2) an evaluation protocol, and (3) performance evaluation measures. Indeed, much of the advances in several computer vision fields, like object detection, classification and segmentation [12], optical-flow computation [3], can be attributed to a ubiquitous access to standard datasets and evaluation protocols [43]. Despite the efforts invested in proposing new trackers, the field suffers from a lack of established methodology for objective comparison.

1.1. Related work

One of the most influential performance analysis efforts for object tracking is PETS (Performance Evaluation of Tracking and Surveillance) [53]. The first PETS workshop that took place in 2000, aimed at evaluation of visual tracking algorithms for surveillance applications. However, its focus gradually shifted to high-level event interpretation algorithms. Other frameworks and datasets have been presented since, but these focussed on evaluation of surveillance systems and event detection, e.g., CAVIAR[2], i-LIDS [3], ETISEO[4], change detection [19], sports analytics (e.g., CVBASE[5]), or specialized on tracking specific objects like faces, e.g. FERET [39] and [26].

Issues with datasets. A trend has emerged in the single-object model-free tracking community to test newly proposed trackers on larger datasets that include different real-life visual phenomena like occlusion, clutter and illumination change. As a consequence, various authors nowadays compare their trackers on many publicly-available sequences, of which some have became a de-facto standard in evaluation of new trackers. However, many of these sequences lack a standard ground truth labeling, which makes comparison of proposed algorithms difficult. To sidestep this issue, Wu et al. [48] have proposed a protocol for tracker evaluation on a selected dataset that does not require ground truth labels. However, this protocol is only appropriate for stochastic trackers. Furthermore, authors usually do not use datasets with various visual phenomena equally represented. In fact, many popular sequences exhibit the same visual phenomenon, which makes the results biased toward some particular types of the phenomena. In their paper, Wu et al. [49] address this issue by annotating each sequence with several visual attributes. For example, a sequence is annotated as “occlusion” if the target is occluded anywhere in the sequence, etc. The results are reported only on the subsets corresponding to a particular attribute. However, visual phenomena like occlusion do not usually last throughout the entire sequence. For example, an 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. Thus a per-frame dataset labeling is required to facilitate a more precise analysis.

Evaluation systems. For objective and rigorous evaluation, an evaluation system that performs on different trackers the same experiment using the same dataset is required. Most notable and general are the ODViS system [23], VIVID [6] and ViPER [11] toolkits. The former two focus on design of surveillance systems, while the latter is a set of utilities/scripts for annotation and computation of different types of performance measures. Recently, Wu et al. [49] have performed a large-scale benchmark of several trackers and developed an evaluation kit that allows integration of other trackers as well. However, in our experience, the integration is not straightforward due to a lack of standardization of the input/output communication between the tracker and the evaluation kit. Collecting the results from the existing publications is an alternative to using an evaluation system that locally runs the experiment. However, such evaluation is hindered by the biases the authors tend to insert in their results. In particular, when publishing a paper on a new tracker, a significant care is usually taken to adjust the parameters of the proposed method such that it delivers the best performance. On the other hand, much less atten-

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

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

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

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

tion is given to competing trackers, leading to a biased preference in the results. Under the assumption that authors introduce bias only for their proposed tracker, Pang et al. [38] 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.

Performance measures. A wealth of performance measures have been proposed for single-object tracker evaluation. These range from basic measures like center error [40], region overlap [31], tracking length [29] and failure rate [28, 27] to more sophisticated measures, such as CoTPS [37], which combine several measures into a single measure. A nice property of the combined measures is that they provide a single score to rank the trackers. A downside is that they offer little insight into the tracker performance. In this respect the basic measures, or their simple derivatives, are preferred as they usually offer a straight-forward interpretation. While some authors choose several basic measures to compare their trackers, the recent study [44] has shown that many measures are correlated and do not reflect different aspects of tracking performance. In this respect, choosing a large number of measures may in fact again bias results toward some particular aspects of tracking performance. Thus a better strategy is to apply few less correlated measures and combine them via ranking lists, similarly to what was done in the change detection challenge [19].

VOT2013. Recognizing the above issues, the Visual Object Tracking (VOT2013) challenge and workshop was organized. The goal was to provide an evaluation platform that goes beyond the current state-of-the-art. In particular, we have compiled a labeled dataset collected from widely used sequences showing a balanced set of various objects and scenes. All the sequences are labeled per-frame with different visual attributes to aid a less biased analysis of the tracking results. We have created an evaluation kit in *Matlab/Octave* that automatically performs three basic experiments on a tracker using the provided dataset. A new comparison protocol based on basic performance measures is also proposed. A significant novelty of the proposed evaluation protocol is that it explicitly addresses the statistical significance of the results and addresses the equivalence of trackers. A dedicated VOT2013 homepage http://votchallenge.net/ has been set up, from which the dataset, the evaluation kit and the results are publicly available. The authors of tracking algorithms have an opportunity to publish their source code at the VOT homepage as well, thus pushing the field of visual tracking towards a reproducible research.

In the following we first review the VOT2013 challenge (Section 2), the dataset (Section 2.1), the performance measures (Section 2.2), the VOT2013 experiments (Section 2.3) and the evaluation methodology (Section 2.4), respectively. The analysis of the VOT2013 results is provided in Section 3 and Section 4 concludes the paper.

2. The VOT2013 challenge

The VOT2013 challenge targets the case in which a user manually initializes a tracker in the first image of a sequence. In case the tracker fails (e.g., drifts away from the target), the user would reinitialize the tracker at the image of failure. The tracker is therefore required to predict a single bounding box of the target for each frame of the sequence. A failure is automatically detected by comparing the predicted bounding box with the ground truth annotation, in case of zero overlap, a failure is proclaimed.

The organisers of VOT2013 provide an evaluation kit and a dataset for performing objective evaluation of the trackers. The authors attending the challenge are required to integrate their tracker into the VOT2013 evaluation kit, which automatically performs a standardized experiment. The results are analyzed by the VOT2013 evaluation methodology. For more details on the participation, please refer to the challenge page[6].

For the sake of simplicity of the evaluation kit, the trackers participating in the VOT2013 challenge have to be causal and always provide a complete reinitialization when initialized by the evaluation kit. Causality requires the tracker to solely process the frames from the initialization up to the current frame without using any information from the future frames. If a tracker fails at some point during tracking, it is reinitialized by the evaluation kit. A complete reinitialization at time-step t requires that any learned information, like appearance and dynamics from the previous frames, should be discarded.

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

2.1. The VOT2013 dataset

The VOT2013 dataset includes various real-life visual phenomena, while containing a small number of sequences to keep the time for performing the experiments reasonably

[6] http://votchallenge.net/participation.html

low. We initially collected a large pool of sequences that have been used by various authors in the tracking community. Each frame of the sequence was labeled with several attributes and a subset of 16 sequences was selected from this pool such that the various visual phenomena like occlusion and illumination changes, were still represented well within the selection.

For most of the selected sequences, the per-frame bounding boxes placed over the object of interest were already available. Since the bounding boxes were annotated by various authors, it is difficult to specify a common rule that guided the annotators. It appears that the bounding boxes were placed such that large percentage of pixels within the bounding box (at least $> 60\%$) belonged to the target. In most cases, this percentage is quite high since the upright bounding box tightly fits the target. But in some cases, (e.g., the *gymnastics* sequence) where an elongated target is rotating significantly, the bounding box contains a larger portion of the background at some frames as well. After inspecting all the bounding box annotations, we have re-annotated those sequences in which the original annotations were poor.

To gain a better insight into the performance of trackers, we have manually or semi-manually labeled each frame in each selected sequence with five visual attributes that reflect a particular challenge in appearance degradation: (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) non-degraded. Such labeling allows us to compare the trackers only on the subsets of frames corresponding to the same attribute. 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.

2.2. The VOT2013 performance measures

There exists an abundance of performance measures in the field of visual tracking (e.g., [48, 38, 19, 26, 49]). The guideline for choosing the performance measures was the interpretability of the measures while selecting as few measures as possible to provide a clear comparison among trackers. Based on the recent analysis of widely-used performance measures [44] we have chosen two orthogonal measures: (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 during tracking. The tracking accuracy at time-step t is defined as the overlap between the tracker predicted bounding box A_t^T and the ground truth bounding box A_t^G

$$\phi_t = \frac{A_t^G \cap A_t^T}{A_t^G \cup A_t^T}. \tag{1}$$

As we will see later, we repeat the experiments multiple times, which results in multiple measurements of accuracy per frame. For further processing, the multiple measurements are averaged, yielding a single, average, accuracy per frame. We can summarize the accuracies in a set of frames by calculating the average of these over the *valid frames*. Note that all frames are not valid for computation of the accuracy measure. In fact, the overlaps in the frames right after the initialization are biased toward higher overlaps since the (noise-free) initialization starts at maximum overlap and it takes a few frames of the burn-in period for the performance to become unbiased by the initialization. In a preliminary study we have determined by a large-scale experiment that the burn-in period is approximately ten frames. This means that ten frames after initialization will be labeled as invalid for accuracy computation.

The robustness was measured by the failure rate measure, which counts the number of times the tracker drifted from the target and had to be reinitialized. A failure was detected once the overlap measure (1) dropped to zero. It is expected that if a tracker fails in a particular frame it will likely fail again if it is initialized in the next frame. To reduce this immediate correlation, the tracker was initialized five frames after the failure. Again, due to multiple repetitions of the experiment we will have multiple measurements of failure rate on a given sequence of frames. The average of these yields an average robustness on a given sequence.

2.3. The VOT2013 experiments

The challenge included the following three experiments:

- Experiment 1: This experiment tested a tracker on all sequences in the VOT2013 dataset by initializing it on the ground truth bounding boxes.
- Experiment 2: This experiment performed Experiment 1, but initialized with noisy bounding box. By noisy bounding box, we mean a randomly perturbed bounding box, where the perturbation is in order of ten percent of the ground truth bounding box size.
- Experiment 3: This experiment performed the Experiment 1 on all sequences with the color images changed to grayscale.

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. Trackers that did not use the color information were allowed to be run only on Experiment 3 and the same results were assumed also for the Experiment 1. All the experiments were automatically performed by the evaluation kit[7]. A tracker was run on each sequence 15 times to obtain a better statistic on its performance.

2.4. The VOT2013 evaluation methodology

Our goal was to compare the performance of trackers in each experiment of Section 2.3 on the six different attribute sequences from Section 2.1 with respect to the two performance measures from Section 2.2. Since we need to establish how well a tracker performs compared to the other trackers, we have developed a ranking-based methodology akin to [9, 12, 19]. In short, within a single experiment, we rank the trackers separately for each performance measure on each attribute sequence. By averaging the ranks of each tracker over the different attributes we obtain the ranking with respect to a performance measure. Giving equal weight to all performance measures, we obtain the final ranking on a selected experiment by averaging the corresponding two rankings.

Note that a group of trackers may perform equally well on a given attribute sequence, in which case they should be assigned an equal rank. In particular, after ranking trackers on an attribute sequence, we calculate for each i-th tracker its corrected rank as follows. We determine for each tracker, indexed by i, a group of equivalent trackers, which contains the i-th tracker as well as any tracker that performed equally well as the selected tracker. The corrected rank of the i-th tracker is then calculated as the average of the ranks in the group of equivalent trackers.

To determine the group of equivalent trackers, we require an objective measure of equivalence on a given sequence. In case of accuracy measure, a per-frame accuracy is available for each tracker. One way to gauge equivalence in this case is to apply a paired test to determine whether the difference in accuracies is statistically significant. In case the differences are Gaussian distributed, the Student's T-test, which is often used in the aeronautic tracking research [4], is the appropriate choice. However, in a preliminary study we have observed that the accuracies in frames are not always Gaussian distributed, which might render this test inappropriate. As alternative, we apply the Wilcoxon Signed-Rank test as in [9]. In case of robustness, we obtain several measurements of number of times the tracker failed over the entire sequence in different runs. However, these cannot be paired, and we use the Wilcoxon Rank-Sum (also known as Mann-Whitney U-test) instead to test the difference in the average number of failures.

When establishing equivalence, we have to keep in mind that statistical significance of performance differences does not directly imply a practical difference [10]. One would have to define a maximal difference in performance of two trackers at which both trackers are said to perform practically equally well. However, since we could not find clear means to objectively define this difference, we reserve our methodology only to testing the statistical significance of the differences. Note, however, that if such a difference was available, our Wilcoxon equivalence tests can readily apply it.

3. The VOT analysis

In this section we analyze the results of the challenge. We begin with a short overview of the trackers considered in the challenge and then present and interpret the overall results. More detailed description of the evaluated trackers as well as a detailed analysis can be found in the Appendix A and the VOT2013 homepage[8], respectively.

3.1. Description of trackers

We have received 19 entries from various authors in the VOT2013 challenge. All of these have performed the baseline Experiment 1, 17 have performed all three experiments, and one performed only Experiment 1 and 3. The VOT committee additionally performed all three experiments with eight baseline trackers. For these the default parameters were selected, or, when not available, were set to reasonable values. Thus a total of 27 trackers were included in the VOT2013 challenge. In the following we briefly overview the entries and provide the reference to an original published paper. In cases where the method was not officially published, we refer to the Appendix A instead.

We have received two entries that applied background adaptation and subtraction to localize the target, MORP (Appendix A.18) and STMT (Appendix A.24). Two trackers applied key-point features to localize the target, Matrioska [34] and SCTT (Appendix A.23). Several approaches were applying global generative visual model for target localization: the incremental subspace-based IVT [40], the histogram-based mean-shift tracker MS [7] and its improved version CCMS (Appendix A.4), two channel blurring approaches DFT [42] and the EDFT [13], two adaptive multiple-feature-combination-based AIF [5] and CactusFl (Appendix A.3), and a sparsity-based PJS-S (Appendix A.20). Many trackers were based on the discriminative global visual models. Among these were the multiple-instance-learning-based tracker MIL [2], the STRUCK [20] and its derivative PLT (Appendix A.21), the compressive tracking based CT [55] and its derivative RDET [41], the sparsity-based ORIA [50] and ASAM (Appendix A.2), and the graph-embedding-based GSDT [15]. The competition entries included several part-based trackers as well. Namely, the generalized Hough-transform-based HT [18], the LGT [45] and its extension LGT++ [51], and the edge-based LT-FLO [30]. Some trackers were utilizing optical flow, e.g., FoT [46], while the TLD [24] combined the local

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

[8] http://votchallenge.net/

Table 1. Ranking results. Highest ranking tracker is marked with red color, the second highest is marked with blue color, and the third highest is marked with green color. Last row displays a joined ranking for all three experiments, which were also used to order the trackers. The trackers that have been verified by the VOT committee are denoted by the asterisk $(\cdot)^*$.

	Experiment 1			Experiment 2			Experiment 3			
	R_A	R_R	R	R_A	R_R	R	R_A	R_R	R	R_Σ
PLT*	7.51	**3.00**	**5.26**	**4.38**	**3.25**	**3.81**	*5.90*	**2.83**	**4.37**	**4.48**
FoT*	**4.56**	11.15	*7.85*	*5.14*	10.84	7.99	**3.08**	9.19	*6.13*	*7.33*
EDFT*	9.14	11.04	10.09	8.14	13.61	10.88	6.52	10.66	8.59	9.85
LGT++*	15.73	*4.25*	9.99	13.36	*4.14*	8.75	15.46	7.34	11.40	10.05
LT-FLO	6.40	17.40	11.90	7.43	14.27	10.85	8.00	12.63	10.31	11.02
GSDT	11.87	11.99	11.93	10.78	12.56	11.67	9.49	9.72	9.60	11.07
SCTT	*4.75*	16.38	10.56	7.65	16.49	12.07	6.00	16.49	11.24	11.29
CCMS*	10.97	10.95	10.96	6.94	8.87	*7.91*	12.10	18.35	15.23	11.36
LGT*	17.83	5.42	11.62	15.38	5.20	10.29	18.63	*7.21*	12.92	11.61
Matrioska	10.62	12.40	11.51	10.59	14.38	12.48	9.07	13.03	11.05	11.68
AIF	7.44	14.77	11.11	9.17	15.25	12.21	6.60	18.64	12.62	11.98
Struck*	11.49	13.66	12.58	13.24	12.64	12.94	9.82	11.20	10.51	12.01
DFT	9.53	14.24	11.89	11.42	15.58	13.50	11.44	11.32	11.38	12.25
IVT*	10.72	15.20	12.96	11.36	15.24	13.30	9.17	14.01	11.59	12.62
ORIA*	12.19	16.05	14.12	14.00	15.92	14.96	10.56	13.26	11.91	13.66
PJS-S	12.98	16.93	14.96	13.50	14.84	14.17	11.19	14.05	12.62	13.92
TLD*	10.55	22.21	16.38	7.83	19.75	13.79	10.03	18.60	14.31	14.83
MIL*	19.97	14.35	17.16	18.46	13.01	15.74	15.32	11.17	13.24	15.38
RDET	22.25	12.22	17.23	19.75	10.97	15.36	17.69	9.97	13.83	15.48
HT*	20.62	13.27	16.95	19.29	12.61	15.95	20.04	12.90	16.47	16.46
CT*	22.83	13.86	18.35	21.58	12.93	17.26	18.92	12.68	15.80	17.13
Meanshift*	20.95	14.23	17.59	18.29	16.94	17.62	22.33	15.97	19.15	18.12
SwATrack	15.81	15.88	15.84	13.97	16.06	15.02	27.00	27.00	27.00	19.29
STMT	23.17	21.31	22.24	22.17	19.50	20.84	20.67	16.96	18.81	20.63
CACTuS-FL	25.39	19.67	22.53	24.17	15.46	19.82	22.92	18.33	20.62	20.99
ASAM	11.23	15.09	13.16	27.00	27.00	27.00	27.00	27.00	27.00	22.39
MORP	24.03	27.00	25.51	24.31	26.00	25.15	27.00	27.00	27.00	25.89

motion estimates with discriminative learning of patches for object re-detection.

3.2. Results

The results are summarized in Table 1 and visualized by the A-R rank plots inspired by the A-R score plots [44], which show each tracker as a point in the joint accuracy-robustness rank space. For more detailed rankings and plots please see the VOT2013 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 $(\cdot)^*$ in Table 1. Looking at the baseline results (Experiment 1), the trackers ranked lowest are MORP, CACTuS-FL and STMT. The low performance of MORP and STMS is not surprising, since they both apply adaptive/dynamic background subtraction, which tends to be less robust in situations with non-static camera and/or the background. The CaCtus-FL is a more sophisticated tracker, however, the tracker does not work well for the objects that significantly move with respect to the image frame. The top performing trackers on the baseline are PLT, FoT, LGT++, EDFT and SCTT. The PLT stands out as a single-scale detection-based tracker that applies on-line sparse structural SVM to adaptively learn a discriminative model from color, grayscale and grayscale derivatives. The tracker does not apply a motion model and does not adapt the size of the target. On the other hand, the FoT, LGT++ and EDFT do apply a motion model. All of these trackers, except for EDFT, can be thought of as part-based models. In particular, the PLT applies a sparse SVM, FoT is an array of Lucas-Kanade predictors that are robustly combined to estimate the object motion, the visual model in LGT++ is a weakly coupled constellation of parts and SCTT uses a sparse regression for target localization. Here, we can consider sparse methods as part-based methods with parts organized in a rigid grid. The target localization in PLT, FoT and EDFT is deterministic, while the

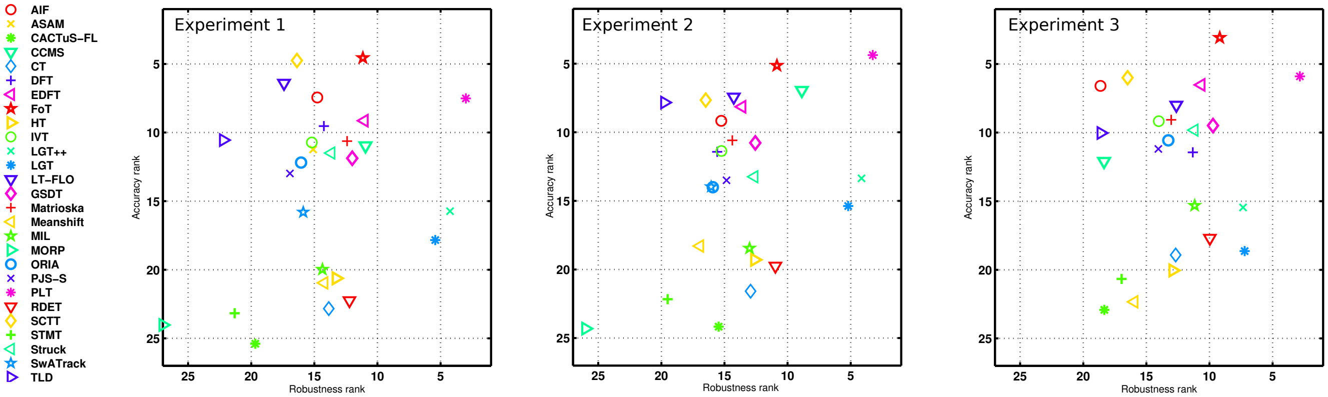

Figure 1. The accuracy-robustness ranking plots with respect to the three experiments. Tracker is better if it resides closer to the top-right corner of the plot.

LGT++ and SCTT are stochastic trackers.

When considering the results averaged over all three experiments, the top-ranked trackers are PLT and FoT, followed by EDFT and LGT++. The A-R ranking plots in Figure 1 offer further insights into the performance of trackers. We can see that, in all three experiments, the PLT yields by far the largest robustness. In the baseline experiment, the two trackers that fairly tightly follow the PLT are the LGT++ and the original LGT. We can see that we have the same situation in experiment with noise, which means that these three trackers perform quite well even in noisy initializations in terms of robustness. However, when considering the accuracy, we can see that the top performing tracker on the baseline is in fact FoT, tightly followed by SCTT and a RANSAC-based edge tracker LT-FLO. In the experiment with noise, the FoT tracker comes second best to PLT, suggesting a bit lower resilience to noisy initializations. This might speak of a reduced robustness of the local motion combination algorithm in FoT in case of noisy initializations. Considering the color-less sequences in Experiment 3, the PLT remains the most robust, however, the FoT comes on top when considering the accuracy.

Figure 2 shows the A-R ranking plots of the Experiment 1 separately for each attribute. The top ranked trackers in the averaged ranks remain at the top also with respect to each attribute, with two exceptions. When considering the size change, the best robustness is still achieved by PLT, however, the trackers that yield best trade-off between the robustness and accuracy are the LGT++ and the size-adaptive mean shift tracker CCMS. When considering occlusion, the PLT and STRUCK seem to share the first place in the best trade-off.

In summary, the sparse discriminative tracker PLT seems to address the robustness quite well, despite that it does not adapt the target size, which reduces its accuracy when the size of the tracked object is significantly changing. On the other hand, the part-based trackers with a rigid part constellation yield a better accuracy at reduced robustness. The robustness is increased with part-based models that relax the constellation, but this on average comes at a cost of significant drop in accuracy.

Apart from the accuracy and robustness, the VOT evaluation kit also measured the times required to perform a repetition of each tracking run. From these measurements, we have estimated the average tracking speed of each tracker (Table 3). Care has to be taken when interpreting these results. The trackers were implemented in different programming languages and run on different machines, with different levels of code optimization. However, we believe that these measurements still give a good estimate of the trackers practical complexity. The trackers that stand out are the PLT and FoT, achieving speeds in range of 150 frames per second (C++ implementations).

Table 2. Degradation difficulty for the six visual attributes: camera motion (camera), illumination change (illum.), object size change (size), object motion change (mot) and non-degraded (nondeg).

	camera	**illum.**	**occl.**	**size**	**mot.**	**nondeg**
Acc.	0.57	0.57	0.58	0.42	0.57	0.61
Fail.	1.58	0.56	0.66	0.93	0.85	0.00

Next we have ranked the individual types of visual degradation according to the tracking difficulty they present to the tested trackers. For each attribute sequence we have computed the median over the average accuracies and failure rates across all the trackers. This median scores were the basis for the attribute ranking. The ranking results computed from Experiment 1 are presented in Table 2. These results confirm that the subsequences that do not contain any specific degradation present little difficulty for the trackers in general. Most trackers do not fail on such intervals and achieve best average overlap. On the other hand, camera motion is the hardest degradation in this respect. One way to explain this is that most trackers focus primarily on appearance changes of the target and do not explicitly account

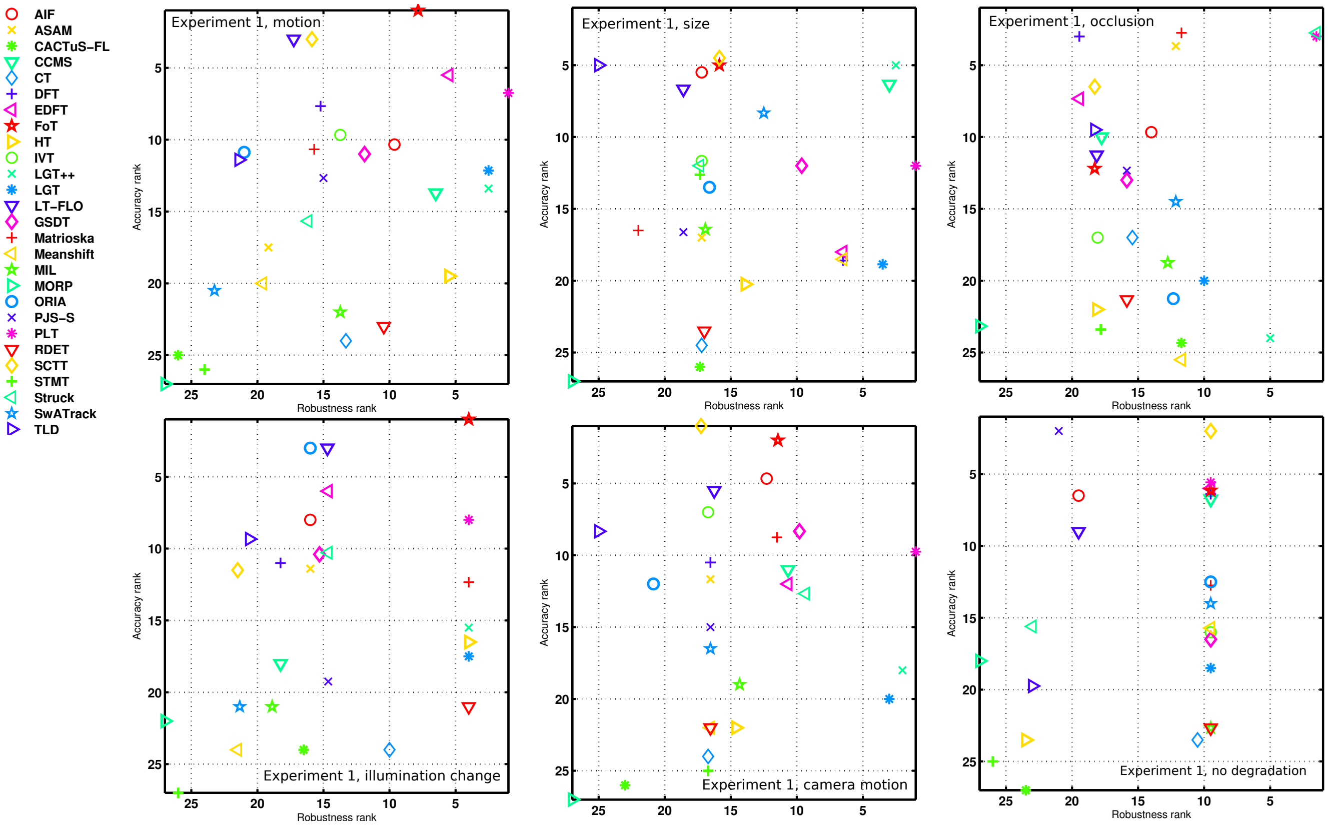

Figure 2. The accuracy-robustness ranking plots of Experiment 1 with respect to the six sequence attributes. The tracker is better if it resides closer to the top-right corner of the plot.

for changing background. Note that camera motion does not necessarily imply that the object is significantly changing position in the image frame. In terms of accuracy the hardest degradation is the changes of object size. This is reasonable as many trackers do not adapt in this respect and sacrifice their accuracy for a more stable visual model that is more accurate in situations where the size of the target does not change. Occlusions and illumination changes are apparently less difficult according to these results. Note, however, that occlusion does pose a significant difficulty to the trackers but the numbers do not indicate extreme difficulty. This might be because the occlusions in our dataset are short-term and partial at best.

4. Conclusion

In this paper we have reviewed the VOT2013 challenge and its results. The VOT2013 contains an annotated dataset comprising many of the widely used sequences. All the sequences have been labeled per-frame with attributes denoting various visual phenomena to aid a more precise analysis of the tracking results. We have implemented an evaluation kit in *Matlab/Octave* that automatically performs three basic experiments on the tracker using the new dataset. A new comparison protocol based on basic performance measures was also proposed. We have created a publicly-available repository and web page that will host the VOT2013 challenge (dataset, evaluation kit, tracking results, source code and/or binaries if the authors choose so). The results of VOT2013 indicate that a winner of the challenge according to the average results is the PLT tracker (Appendix A.21). However, 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. It is impossible to conclusively say what kind of tracking strategy works best in general, however, there is some evidence showing that robustness tends to be better for the trackers that do not apply global models, but rather split the visual models into parts.

The absence of homogenization of the single-tracking performance evaluation makes it difficult to rigorously compare trackers across publications and stands in the way of faster development of the field. We expect that the homogenization of performance evaluation will not happen without involving a critical part of the community and without providing a platform for discussion. The VOT2013 challenge and workshop was an attempt toward this goal. Our future work will be focused on revising the evaluation kit, dataset

Table 3. Performance, implementation and evaluation environment characteristics.

	FPS	Implem.	Hardware
PLT	**169.59**	C++	Intel Xeon E5-16200
FoT	*156.07*	C++	Intel i7-3770
EDFT	12.82	Matlab	Intel Xeon X5675
LGT++	5.51	Matlab / C++	Intel i7-960
LT-FLO	4.10	Matlab / C++	Intel i7-2600
GSDT	1.66	Matlab	Intel i7-2600
SCTT	1.40	Matlab	Intel i5-760
CCMS	57.29	Matlab	Intel i7-3770
LGT	2.25	Matlab / C++	AMD Opteron 6238
Matrioska	16.50	C++	Intel i7-920
AIF	30.64	C++	Intel i7-3770
Struck	3.46	C++	Intel Pentium 4
DFT	6.65	Matlab	Intel Xeon X5675
IVT	5.03	Matlab	AMD Opteron 6238
ORIA	1.94	Matlab	Intel Pentium 4
PJS-S	1.18	Matlab / C++	Intel i7-3770K
TLD	10.61	Matlab	Intel Xeon W3503
MIL	4.45	C++	AMD Opteron 6238
RDET	22.50	Matlab	Intel i7-920
HT	4.03	C++	Intel i7-970
CT	9.15	Matlab / C++	Intel Pentium 4
Meanshift	8.76	Matlab	Intel Xeon
SwATrack	2.31	C++	Intel i7
STMT	0.24	C++	Intel Xeon X7460
CACTuS-FL	0.72	Matlab	Intel Xeon X5677
ASAM	0.93	Matlab	Intel i5-2400
MORP	9.88	Matlab	Intel i7

as well as challenges through the feedbacks gained from the community.

Acknowledgements

This work was supported in part by the following research programs and projects: Slovenian research agency research programs P2-0214, P2-0094, Slovenian research agency projects J2-4284, J2-3607, J2-2221 and European Union seventh framework programme under grant agreement no 257906. Jiri Matas and Tomas Vojir were supported by CTU Project SGS13/142/OHK3/2T/13 and by the Technology Agency of the Czech Republic project TE01020415 (V3C – Visual Computing Competence Center).

A. Submitted trackers

In this appendix we provide a short summary of all trackers that were considered in the VOT2013 competition.

A.1. Tracking with an Adaptive Integrated Feature (AIF)

Submitted by:

Weihua Chen *weihua.chen@nlpr.ia.ac.cn*
Lijun Cao *ljcao@nlpr.ia.ac.cn*
Junge Zhang *jgzhang@nlpr.ia.ac.cn*
Kaiqi Huang *kqhuang@nlpr.ia.ac.cn*

AIF tackles the discriminative learning problem in low resolution, lack of illumination and clutter by presenting an adaptive multi-feature integration method in terms of feature invariance, which can evaluate the stability of features in sequential frames. The adaptive integrated feature (AIF) consists of several features with dynamic weights, which describe the degree of invariance of each single feature. The reader is referred to [5] for details.

A.2. Adaptive Sparse Appearance Model Tracker (ASAM)

Submitted by:

B. Bozorgtabar *behzad.bozorgtabar@canberra.edu.au*
Roland Goecke *roland.goecke@ieee.org*

ASAM accounts for drastic appearance changes by modelling the object as a set of appearance models. An online algorithm is used based on a discriminative and generative sparse representation. A two-stage algorithm is used to exploit both the information of the example in the first frame and successive observations obtained online.

A.3. Competitive Attentional Correlation Tracker using Shape and Feature Learning (CACTuS-FL)

Submitted by:

Sebastien Wong *sebastien.wong@dsto.defence.gov.au*
Adam Gatt *adam.gatt@dsto.defence.gov.au*
Anthony Milton *Anthony.Milton@IEEE.org*
Dale Ward *Dale.Ward@unisa.edu.au*
David Kearney *david.kearney@unisa.edu.au*

CACTuS-FL tackles model drift of the object by additionally tracking sources of clutter and then assigning observations to the tracks that best describe the observations. This tracker augments the work described in [16].

A.4. Color Correspondences Mean-Shift (CCMS)

Submitted by:

Tomas Vojir *vojirt1@fel.cvut.cz*
Jir Matas *matas@cmp.felk.cvut.cz*

The Color Correspondences Mean-Shift tracker maximizes a likelihood ratio of similarity between the target model and the target candidate and the similarity between the target candidate and the background model for each color (histogram bin) separately by a standard Mean-Shift iteration (as proposed by Comaniciu et al. [7]). The weighted mean of the correspondences is then used as a motion estimation. This process is iterative and runs for each frame until it converges or until the maximum number of iteration is reached.

A.5. Compressive Tracking (CT)

Submitted by:
VOT2013 Technical Committee

The CT tracker 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. The reader is referred to [55] for details and to http://www4.comp.polyu.edu.hk/~cslzhang/CT/CT.htm for code.

A.6. Distribution Fields for Tracking (DFT)

Submitted by:
Michael Felsberg *michael.felsberg@liu.se*

A common technique for gradient descent based trackers is to smooth the objective function by blurring the image. However, blurring destroys image information, which can cause the target to be lost. DFT introduces a method for building an image descriptor using distribution fields, a representation that allows smoothing the objective function without destroying information about pixel values. The reader is referred to [42] and to http://people.cs.umass.edu/~lsevilla/trackingDF.html for code.

A.7. Enhanced Distribution Fields for Tracking (EDFT)

Submitted by:
Michael Felsberg *michael.felsberg@liu.se*

The EDFT is a novel variant of the DFT [42]. EDFT derives an enhanced computational scheme by employing the theoretic connection between averaged histograms and channel representations. The reader is referred to [13] for details.

A.8. Flock of Trackers (FoT)

Submitted by:
Tomas Vojir *vojirt1@fel.cvut.cz*
Jiri Matas *matas@cmp.felk.cvut.cz*

The Flock of Trackers (FoT) estimates the object motion from the transformation estimates of a number of local trackers covering the object. The reader is referred to [46] for details.

A.9. HoughTrack (HT)

Submitted by:
VOT2013 Technical 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 [18] for details and to http://lrs.icg.tugraz.at/research/houghtrack/ for code.

A.10. Incremental Learning for Robust Visual Tracking (IVT)

Submitted by:
VOT2013 Technical Committee

The idea of the IVT tracker 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. The reader is referred to [40] for details and to http://www.cs.toronto.edu/~dross/ivt/ for code.

A.11. Local-Global Tracking (LGT)

Submitted by:
Luka Čehovin *luka.cehovin@fri.uni-lj.si*
Matej Kristan *matej.kristan@fri.uni-lj.si*
Aleš Leonardis *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 [45] for details.

A.12. An enhanced adaptive coupled-layer visual LGTracker++ (LGTracker++)

Submitted by:
Jingjing Xiao *shine636363@sina.com*
Rustam Stolkin *r.stolkin@cs.bham.ac.uk*
Aleš Leonardis *ales.leonardis@fri.uni-lj.si*

LGTracker++ improves the LGT tracker [45] in the cases of environment clutter, significant scale changes, failures due to occlusion and rapid disordered movement. Algorithmically, the scale of the patches is adapted in addition to adapting the bounding box. marginal patch distributions are used to solve patch drifting in environment clutter. a memory is added and used to assist recovery from occlusion. situations where the tracker may lose the target are automatically detected, and a particle filter is substituted for the Kalman filter to help recover the target. The reader is referred to [51] for details.

A.13. Long Term Featureless Object Tracker (LT-FLO)

Submitted by:
Karel Lebeda *k.lebeda@surrey.ac.uk*
Richard Bowden *r.bowden@surrey.ac.uk*
Ji Matas *matas@cmp.felk.cvut.cz*

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 [30] for details.

A.14. Graph Embedding Based Semi-Supervised Discriminative Tracker (GSDT)

Submitted by:
Jin Gao *jgao10@nlpr.ia.ac.cn*
Junliang Xing *jlxing@nlpr.ia.ac.cn*
Weiming Hu *wmhu@nlpr.ia.ac.cn*
Xiaoqin Zhang *xqzhang@nlpr.ia.ac.cn*

GSDT is based on discriminative learning, where positive and negative samples are collected for graph embedding. GSDT adopts graph construction based classifiers without the assistance of learning object subspace generatively as previous work did. The tracker also uses a new graph structure to characterize the inter-class separability and the intrinsic local geometrical structure of the samples. The reader is referred to [15] for details.

A.15. Matrioska

Submitted by:
Mario Edoardo Maresca *mariomaresca@hotmail.it*
Alfredo Petrosino *petrosino@uniparthenope.it*

Matrioska decomposes tracking into two separate modules: detection and learning. The detection module can use multiple keypoint-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. The reader is referred to [34] for details.

A.16. Meanshift

Submitted by:
VOT2013 Technical Committee

Meanshift uses a feature histogram-based target representation that is regularized by spatial masking with an isotropic kernel. The masking induces spatially-smooth similarity functions suitable for gradient-based optimization, hence, the target localization problem can be formulated using the basin of attraction of the local maxima. Meanshift employs a metric derived from the Bhattacharyya coefficient as similarity measure, and use the mean shift procedure to perform the optimization. The reader is referred to [7] for details.

A.17. MIL

Submitted by:
VOT2013 Technical Committee

MIL 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. The reader is referred to [2] for details and to http://vision.ucsd.edu/~bbabenko/project_miltrack.shtml for code.

A.18. Object Tracker using Adaptive Background Subtraction and Kalman Filter (MORP)

Submitted by:
Hakki Can Karaimer *cankaraimer@iyte.edu.tr*

MORP basically works in three major steps: (i) pixels are assigned to foreground by taking the difference between the next image frame and the current background. MORP uses an effective global thresholding technique in this step. The current background is computed by averaging image frames at the beginning of the tracking process and after the first ten frames (adaptive background subtraction) . (ii) foreground pixel (blobs) are processed by morphological opening with a disc whose diameter is six pixels, then a morphological closing with a disc whose diameter is ten pixels. Blobs whose area is less than eight by eight pixel are eliminated. After this step, the biggest remaining blob is selected as the blob to be tracked. (iii) according to the detected blob position and velocity, the next position of the object is calculated by using a Kalman filter.

A.19. Online Robust Image Alignment (ORIA)

Submitted by:
VOT2013 Technical Committee

The ORIA tracker treats the tracking problem as the problem of online aligning a newly arrived image to previously well-aligned images. The tracker treats the newly arrived image, after alignment, as being linearly and sparsely reconstructed by the well-aligned ones. The task is accomplished by a sequence of convex optimization that minimizes the L1 norm. After that, online basis updating is pursued in two different ways: (1) a two-stage incremental alignment for joint registration of a large image dataset which is known a prior, and (2) a greedy online alignment of dynamically increasing image sequences, such as in the tracking scenario. The reader is referred to [50] for details.

A.20. Patchwise Joint Sparse-SOMP (PJS-S)

Submitted by:
Ali Zarezade *zarezade@ce.sharif.edu*
Hamid R. Rabiee *rabiee@sharif.edu*
Ali Soltani-Farani *a_soltani@ce.sharif.edu*
Ahmad Khajenezhad *khajenezhad@ce.sharif.edu*

PJS-S models object appearance using a dictionary composed of target patches from previous frames. In each frame, the target is found from a set of candidates via a likelihood measure that is proportional to the sum of the reconstruction error of each candidate patch. The tracker assumes slow changes of object appearance, hence target and traget candidates are expected to to belong to the same subspace. PJS-S imposes this intuition by using joint sparsity inducing norms, to enforce the target and previous best candidates to have the same sparsity pattern. The reader is referred to [54] for details.

A.21. Single scale pixel based LUT tracker (PLT)

Submitted by:
Cher Keng Heng *Hengcherkeng235@gmail.com*
Samantha Yue Ying Lim *yueying53@gmail.com*
Zhi Heng Niu *niuzhiheng@gmail.com*
Bo Li *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 color, grayscale and gradient information. To select a small set of discriminative features, an online sparse structural SVM [20] is used. Since the object can be non-rigid and the bounding box may be noisy, not all pixels in the bounding box belong to the object. Hence, a probabilistic object-background segmentation mask from color histograms is created and used to weight the features during SVM training. The resulting weighted and convex problem can be solved in three steps: (i) compute the probability that a pixel belongs to the object by using its color. (ii) solve the original non-sparse structural SVM and (iii) shrink the solution [21], i.e. features with smallest values are discarded. Since the feature vector is binary, the linear classifier can be implemented as a lookup table for fast speed.

A.22. Random Diverse Ensemble Tracker (RDET)

Submitted by:
Ahmed Salahledin *ahmed.salaheldin.hussein@gmail.com*
Sara Maher *m.a.elhelw@googlemail.com*
Mohamed ELHelw *s.m.elkerdawy@gmail.com*

RDET proposes a novel real-time ensemble approach to tracking by detection. It creates a diverse ensemble using random projections to select strong and diverse sets of compressed features. The reader is referred to [41] for details.

A.23. Structural Convolutional Treelets Tracker (SCTT)

Submitted by:
Yang Li *fliyang89@zju.edu.cn*
Jianke Zhu *jkzhug@zju.edu.cn*

SCTT is a generative tracker, which is mainly inspired by convolutional treelets keypoint matching algorithm [47]. SCTT employs a two-layer treelets [1] to extract the image features from the input video frames. The proposed two-layer structural framework is able to improve the representation power of treelets by dividing image into smaller pieces while reducing the feature dimensionality. Once image features are extracted, LSST-distance [8] is calculated and the patch with the smallest distance as the tracked target is selected. Note that the reconstruction error is under the Laplace distribution in LSST-distance, which is more robust to partial occlusions. When SCTT finds the nearest patch with LSST-distance in image, a similarity update threshold is set. As treelets require fewer samples than PCA, only those patches with high confidence are added into the updating process. Thus, the proposed updating strategy is very robust to the noises.

A.24. Spatio-temporal motion triangulation based tracker (STMT)

Submitted by:
Sebastien Poullot *poullot.sebastien@free.fr*
Shin'ichi Satoh *satoh@nii.ac.jp*

STMT is based on a two layer process: camera motion estimation then object motion estimation. The process flow begins by registering two frames, yielding the camera motion. Successive image frames are aligned, candidate objects are obtained by frame differencing and association is established either by the intersection of bounding boxes or by employing a SIFT matching.

A.25. Struck

Submitted by:
VOT2013 Technical Committee

Struck 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. The reader is referred to [20] for details and to http://www.samhare.net/research/struck/code for code.

A.26. An Adaptive Swarm Intelligence-based Tracker (SwATrack)

Submitted by:

Mei Kuan Lim *imeikuan@siswa.um.edu.my*
Chee Seng Chan *cs.chan@um.edu.my*
Dorothy Monekosso *dorothy.monekosso@uwe.ac.uk*
Paolo Remagnino *p.remagnino@kingston.ac.uk*

SwATrack deems tracking as an optimisation problem and adapted the particle swarm optimisation (PSO) algorithm as the motion estimator for target tracking. PSO is a population based stochastic optimisation methodology, which was inspired by the behavioural models of bird flocking. The reader is referred to [33] for details.

A.27. TLD

Submitted by:
VOT2013 Technical Committee

TLD explicitly decomposes the long-term tracking task into tracking, learning, and detection. The detector localizes all appearances that have been observed so far and corrects the tracker if necessary. The learning estimates the detector errors and updates it to avoid these errors in the future. The reader is referred to [25] for details and to https://github.com/zk00006/OpenTLD for code.

References

[1] B. N. A. Lee and L. Wasserman. Treelets - an adaptive multi-scale basis for sparse unordered data. *Ann. Appl. Stat.*, 2(2):435–471, 2008.

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

[3] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and S. R. A database and evaluation methodology for optical flow. *Int. J. Comput. Vision*, 92(1):1–31, 2011.

[4] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*, chapter 11, pages 438–440. John Wiley & Sons, Inc., 2001.

[5] W. Chen, L. Cao, J. Zhang, and K. Huang. An adaptive combination of multiple features for robust tracking in real scene. In *Vis. Obj. Track. Challenge VOT2013, In conjunction with ICCV2013*, 2013.

[6] R. Collins, X. Zhou, and S. K. Teh. An open source tracking testbed and evaluation web site. In *Perf. Eval. Track. and Surveillance*, 2005.

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

[8] H. L. D. Wang and M.-H. Yang. Least soft-threshold squares tracking. In *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2013)*, Portland, June 2013.

[9] J. Demšar. Statistical comparisons of classifiers over multiple datasets. 7:1–30, 2006.

[10] J. Demšar. On the appropriateness of statistical tests in machine learning. In *Workshop on Evaluation Methods for Machine Learning in conjunction with ICML*, 2008.

[11] D. Doermann and D. Mihalcik. Tools and techniques for video performance evaluation. In *Proc. Int. Conf. Pattern Recognition*, page 167170, 2000.

[12] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision*, 88(2):303–338, June 2010.

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

[14] P. Gabriel, J. Verly, J. Piater, and A. Genon. The state of the art in multiple object tracking under occlusion in video sequences. In *Proc. Advanced Concepts for Intelligent Vision Systems*, page 166173, 2003.

[15] J. Gao, J. Xing, W. Hu, and X. Zhang. Graph embedding based semi-supervised discriminative tracker. In *Vis. Obj. Track. Challenge VOT2013, In conjunction with ICCV2013*, 2013.

[16] A. Gatt, S. Wong, and D. Kearney. Combining online feature selection with adaptive shape estimation. In *Image and Vision Computing New Zealand (IVCNZ), 2010 25th International Conference of*, pages 1–8. IEEE, 2010.

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

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

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

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

[21] C.-K. Heng, S. Yokomitsu, Y. Matsumoto, and H. Tamura. Shrink boost for selecting multi-lbp histogram features in object detection. In *Comp. Vis. Patt. Recognition*, pages 3250–3257, 2012.

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

[23] C. Jaynes, S. Webb, R. Steele, and Q. Xiong. An open development environment for evaluation of video surveillance systems. In *PETS*, 2002.

[24] Z. Kalal, J. Matas, and K. Mikolajczyk. P-n learning: Bootstrapping binary classifiers by structural constraints. In *Comp. Vis. Patt. Recognition*, pages 49–56. IEEE, 2010.

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

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

[27] M. Kristan, S. Kovacic, A. Leonardis, and J. Pers. A two-stage dynamic model for visual tracking. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 40(6):1505–1520, 2010.

[28] M. Kristan, J. Perš, M. Perše, M. Bon, and S. Kovačič. Multiple interacting targets tracking with application to team sports. In *International Symposium on Image and Signal Processing and Analysis*, pages 322–327, September 2005.

[29] J. Kwon and K. M. Lee. Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping monte carlo sampling. In *Comp. Vis. Patt. Recognition*, pages 1208–1215. IEEE, 2009.

[30] K. Lebeda, R. Bowden, and J. Matas. Long-term tracking through failure cases. In *Vis. Obj. Track. Challenge VOT2013, In conjunction with ICCV2013*, 2013.

[31] H. Li, C. Shen, and Q. Shi. Real-time visual tracking using compressive sensing. In *Comp. Vis. Patt. Recognition*, pages 1305–1312. IEEE, 2011.

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

[33] M. Lim, C. Chan, D. Monekosso, and P. Remagnino. Swatrack: A swarm intelligence-based abrupt motion tracker. In *In proceedings of IAPR MVA*, page pages 3740, 2013.

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

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

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

[37] T. Nawaz and A. Cavallaro. A protocol for evaluating video trackers under real-world conditions. *IEEE Trans. Image Proc.*, 22(4):1354–1361, 2013.

[38] Y. Pang and H. Ling. Finding the best from the second bests – inhibiting subjective bias in evaluation of visual tracking algorithms. In *Comp. Vis. Patt. Recognition*, 2013.

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

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

[41] A. Salaheldin, S. Maher, and M. E. Helw. Robust real-time tracking with diverse ensembles and random projections. In *Vis. Obj. Track. Challenge VOT2013, In conjunction with ICCV2013*, 2013.

[42] L. Sevilla-Lara and E. G. Learned-Miller. Distribution fields for tracking. In *Comp. Vis. Patt. Recognition*, pages 1910–1917. IEEE, 2012.

[43] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *Comp. Vis. Patt. Recognition*, pages 1521–1528. IEEE, 2011.

[44] L. Čehovin, M. Kristan, and A. Leonardis. Is my new tracker really better than yours? Technical Report 10, ViCoS Lab, University of Ljubljana, Oct 2013.

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

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

[47] C. Wu, J. Zhu, J. Zhang, C. Chen, and D. Cai. A convolutional treelets binary feature approach to fast keypoint recognition. In *ECCV*, pages 368–382, 2013.

[48] H. Wu, A. C. Sankaranarayanan, and R. Chellappa. Online empirical evaluation of tracking algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(8):1443–1458, 2010.

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

[50] Y. Wu, B. Shen, and H. Ling. Online robust image alignment via iterative convex optimization. In *Comp. Vis. Patt. Recognition*, pages 1808–1814. IEEE, 2012.

[51] J. Xiao, R. Stolkin, and A. Leonardis. An enhanced adaptive coupled-layer LGTracker++. In *Vis. Obj. Track. Challenge VOT2013, In conjunction with ICCV2013*, 2013.

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

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

[54] A. Zarezade, H. R. Rabiee, A. Soltani-Frani, and A. Khajenezhad. Patchwise joint sparse tracker with occlusion detection using adaptive markov model. *preprint in arXiv*, 2013.

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