A Novel Performance Evaluation Methodology for Single-Target Trackers

Journal:	*Transactions on Pattern Analysis and Machine Intelligence*
Manuscript ID:	TPAMI-2014-03-0189
Manuscript Type:	Regular
Keywords:	I.4.8.d Motion < I.4.8 Scene Analysis < I.4 Image Processing and Computer Vision < I Computing Methodologies, I.4.8.m Time-varying imagery < I.4.8 Scene Analysis < I.4 Image Processing and Computer Vision < I Computing Methodologies, I.4.8.n Tracking < I.4.8 Scene Analysis < I.4 Image Processing and Computer Vision < I Computing Methodologies

A Novel Performance Evaluation Methodology for Single-Target Trackers

Matej Kristan, *Member, IEEE,* Roman Pflugfelder, Jiri Matas, Aleš Leonardis, Fatih Porikli, Georg Nebehay, Gustavo Fernandez, Tomáš Vojíř and Luka Čehovin

Abstract—This paper addresses the problem of single target tracker performance evaluation. The performance measures, the dataset and the evaluation system are identified as the most important components of tracker evaluation and critical requirements for each of the components are proposed. In line with these requirements, a new evaluation methodology is proposed that aims at simple, easily interpretable, tracker comparison and includes objective comparison in terms of statistical significance. To visually compare performance of trackers, we present a new performance visualization. A fully-annotated dataset with per-frame annotations with several visual attributes is also presented. To maximize the diversity of the dataset in terms of visual properties, the dataset is constructed in a novel way by clustering a large number of videos according to their visual attributes. A multi-platform evaluation system, which allows easy integration of third-party trackers is presented as well. The proposed evaluation methodology was evaluated on the VOT2013 challenge using the new dataset and 27 trackers. Using the results of this analysis, the performance of state-of-the-art trackers is discussed. Furthermore, an exhaustive analysis of the dataset from the perspective of tracking difficulty is carried out.

Index Terms—Performance analysis, single-target tracking, model-free tracking, tracker evaluation methodology, tracker evaluation datasets, tracker evaluation system

1 Introduction

Visual tracking is a rapidly evolving field that has been increasingly attracting attention of the vision community. One reason is that it offers many scientific challenges. Second, it emerges in other computer vision tasks, such as motion analysis, event detection and activity recognition. The steady increase of hardware performance and reduction in price, opened a vast application potential for tracking algorithms. These applications include surveillance systems, transport, sports analytics, medical imaging, mobile robotics, film post-production and human-computer interfaces.

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., [1], [2], [3], [4], [5], [6], [7]. However, despite the efforts invested in proposing new trackers, the field suffers from a lack of established methodology for objective comparison.

One of the most influential performance analysis efforts for object tracking is PETS (Performance Evaluation of Tracking and Surveillance) [8]. The first PETS workshop that took place in 2000, aimed at evaluation of visual tracking algorithms for surveillance applications. 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[1], i-LIDS [2], ETISEO[3], change detection [9], sports analytics (e.g., CVBASE[4]), or specialized on tracking specific objects like faces, e.g. FERET [10] and [11]. Recently there have been several publications that have focused on different aspects of model-free visual object tracking evaluation, eg., [12], [13], [14], [15], [16] to list just a few.

There are several important subfields in visual tracking, ranging from multi-camera, multi-target, to single-target trackers. These subfields are quite diverse, which prohibits the use of a single evaluation methodology for all. Rather, specific methodologies have been tailored to each subfield.

In this paper, single-camera, single-target, model-free, causal trackers, applied to short-term tracking are considered. 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 to infer the object position in the current frame. The tracker output is specified by an axis-aligned bounding box.

- *M. Kristan and L. Čehovin are with the Faculty of Computer and Information Science, University of Ljubljana, Slovenia.*
  E-mail: see http://www.vicos.si/People/Matejk
- *R. Pflugfelder, G. Nebehay and G. Fernandez are with Austrian Institute of Technology, Austria*
- *J. Matas and T. Vojíř are with Czech Technical University in Prague, Czech Republic*
- *A. Leonardis is with CNCR University of Birmingham, United Kingdom*
- *F. Porikli is with NICTA and Australian National University, Australia*

[1] http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1
[2] http://www.homeoffice.gov.uk/science-research/hosdb/i-lids
[3] http://www-sop.inria.fr/orion/ETISEO
[4] http://vision.fe.uni-lj.si/cvbase06/

1.1 Requirements for tracker evaluation

In general, the evaluation of new tracking algorithms depends on three essential components: (1) performance evaluation measures, (2) a dataset and (3) an evaluation system. In the following, requirements for these components are stated.

Performance measures. A wealth of performance measures have been proposed for single-object tracker evaluation, however, there is no consensus on which should be preferred. Ideally, measures should clearly express certain aspects of tracking and should enable ranking of trackers. Apart from merely ranking, we also need to determine cases when two or more trackers are performing equally well. This motivates the following requirements: (R_{M1}) The measures should allow easy interpretation and clear tracker comparison. (R_{M2}) The measures should support means of establishing a well defined tracker equivalence.

Datasets. The dataset should allow evaluation of trackers under various conditions like occlusion, clutter and illumination change. One approach is to construct a very large dataset, however, that does not guarantee diversity in visual attributes. A better approach is to annotate each sequence with the visual attributes occuring in that sequence. For example, a sequence is annotated as "occlusion" if the target is occluded anywhere in the sequence, etc. The trackers can then be compared only on the sequences 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 a tracker might fail due to some other effects occurring at the beginning of the sequence. In this case, the failure would be falsely attributed to occlusion. Thus a per-frame dataset labeling is required to facilitate a more precise analysis. This motivates the following requirements: (R_{D1}) The dataset should be diverse in visual attributes. (R_{D2}) Per-frame annotation of visual attributes is required.

Evaluation systems. For a rigorous evaluation, an evaluation system that performs the same experiment on different trackers using the same dataset is required. The wide-spread practice is to initialize the tracker in the first frame and let it run until the end of a sequence. However, the tracker might fail right at the beginning of the sequence due to some visual degradation, effectively meaning that the system utilized only the first few frames for evaluation of this tracker. Thus the first requirement for the system is that it fully uses the data. This means that once the tracker fails, the system has to detect the failure and reinitialize the tracker. Therefore, a certain level of interaction, that goes beyond simple running until the end of the sequence, is required. Furthermore, the evaluation system has to also account for the fact that the trackers are typically coded in various programming languages and often platform-dependent. This motivates the following set of requirements the evaluation system should meet: (R_{S1}) Full use of the dataset. (R_{S2}) Allow interaction with the tracker. (R_{S3}) Support for multiple platforms. (R_{S4}) Easy integration with trackers.

1.2 Our contributions

We claim the following four contributions. Our first contribution is the a new tracker evaluation methodology based on two simple, easily interpretable, performance measures. For this purpose, a ranking-based tracker comparison approach that accounts for both performance measures is presented. The new evaluation methodology explicitly addresses the statistical significance of the results and it also addresses the concept of tracker equivalence. A new visualization to aid comparative analysis of trackers is proposed as well. The second contribution is the new dataset and the evaluation system. A novel video clustering approach based on visual properties was applied to obtain a well balanced dataset in terms of objects and scenes. The dataset is fully annotated, all the sequences are labeled per-frame with visual attributes to facilitate in depth analysis. The proposed evaluation system enjoys multi-platform compatibility and offers easy integration with trackers. The system has been tested in a large-scale distributed experiment on the VOT2013 challenge. The third contribution is a detailed comparative analysis of 27 trackers using the proposed methodology. The forth contribution is a novel analysis of the sequences in the dataset from the perspective of tracking success. Preliminary versions of some parts of this paper have been previously published in two workshop papers [17], [18].

The remainder of the paper is structured as follows. In Section 2 the most related work is reviewed and discussed, the new tracker evaluation methodology is presented in Section 3, while the new dataset selection approach, the evaluation system and the results of the experimental analysis are presented in Section 4. Conclusions are drawn in Section 5.

2 Related work

2.1 Performance measures

A wealth of performance measures have been proposed for single-object tracker evaluation. These range from basic measures like center error [19], region overlap [20], tracking length [21], failure rate [22], [23], F-score [12], [15], pixel-based precision [12], to more sophisticated measures, such as CoTPS [24] and [25], 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 studies [26], [15] have shown that many measures are correlated and do not consider diverse 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. Smeulders et al., [15] propose using two measures: an F-score calculated at Pascal region overlap criterion (threshold 0.5) and a center error. 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 [14]. Since the center error becomes arbitrary high once the tracker fails, Wu et al. [14] propose to measure the percentage of frames in which the center distance is within some prescribed threshold. However, this threshold significantly depends on 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 [15] show that the trackers do not differ significantly under this measure which makes it less appropriate for tracker comparison. Čehovin et al. [26] propose the region overlap as a favorable choice over the center error. While it is important to study and evaluate the tracker performance separately in terms of several less correlated performance measures, it is sometimes required to rank trackers in a single rank list. In this case a good strategy is to combine these measures via partial ranking lists, similarly to what was done in the change detection challenge [9].

2.1.1 Visual performance evaluation

Several authors propose to visually compare tracking performance via performance summarization plots. These plots show the percentage of frames for which the estimated object location is within some threshold distance of the ground truth. Most notable are precision plots [6], [27], [14], which measure the object location accuracy in terms of center error. Alternatively, success plots [13], [14] use the region overlap instead. For easier comparison, [14] calculate the area under the curve in the success plots, which is shown in [26] to be equal to the average region overlap. Salti et al., [13] implicitly account for variable threshold dependency by plotting the percentage of correctly tracked frames with respect to the mean region overlap within these frames. Čehovin et al. [26] propose a similar visualization, but they apply a single, zero, threshold on the overlap. A tracker is thus represented as a single point in this 2D space, rather than a curve, which allows easier comparison. A drawback of performance plots is that they typically become cluttered when comparing several trackers on several sequences in the same plot. To address this, Smeulders et al. [15] calculate a performance measure per sequence for a tracker and order these values from highest to lowest, thus obtaining a so-called survival curve. Performance of several trackers is then compared on the entire dataset by visualizing their survival curves.

2.2 Datasets

It is a common practice to compare 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. [28] have proposed a protocol for stochastic tracker evaluation on a selected dataset that does not require ground truth labels. A similar approach was adapted by [29] to evaluate tracking algorithms on long sequences. Datasets with various visual phenomena equally represented are not usually used. In fact, many popular sequences are conceptually similar, which makes the results biased toward some particular types of the phenomena. To address this issue, Wu et al. [14] annotated each sequence with several visual attributes and report tracker performance with respect to each attribute separately. However, a per-frame annotation is not provided. Recently, Smeulders et al. [15], have presented a very large dataset called 'Amsterdam Library of Ordinary Videos' (ALOV). The dataset is composed of over three hundred sequences collected from the standard datasets and additional YouTube videos. Each sequence is annotated by one of thirteen classes of difficulty [30] and, with exception of ten long sequences, most sequences are kept short to increase diversity. Nevertheless, the sequences are not annotated per-frame with visual attributes. Moreover, some sequences contain cuts, which means that this dataset may favor detection-based trackers, since the object may change position and appearance in consecutive frames not by virtue of motion and ambient variation but rather video editing.

2.3 Evaluation systems

Currently, the most notable and general systems are ODViS [31], VIVID [32], ViPER [33]. 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. The recently proposed ViCamPEv [12] toolkit is dedicated to testing a pre-determined set of OpenCV-based basic tracker. Although it may be potentially extended by integrating new OpenCv-based trackers, it is constrained to Windows OS, thus making it unsuitable for integration of third-party trackers implemented in other OS and non C/C++ languages. Moreover, none of these systems support interaction with the tracker, which limits their applicability. Recently, Wu et al. [14] have performed a large-scale benchmark study of several trackers and developed an evaluation kit that allows integration of third-party trackers as well. However, the integration seems to be 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 attention 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. [16] have proposed a page-rank-like approach to data-mine the published results of second-best trackers 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 enough publications that would compare these trackers. Furthermore, the quality of ranking strongly relies on the quality of the evaluation methodology used in the existing publications. The latter may suffer from poor dataset selection and problems with performance measures, variables that cannot be easily controlled for by such meta-analysis.

3 Visual object tracker evaluation

For now it is assumed that the evaluation system and the dataset fulfill the requirements stated in Section 1.1. Briefly, the dataset is per-frame annotated by visual attributes and the object positions are denoted by axis-aligned bounding boxes. The evaluation system runs each tracker on each sequence of the dataset. Once the tracker drifts off the target, the system detects a tracking failure and reinitializes the tracker. All trackers are run multiple times to account for the possible stochastic nature of trackers. In this section the focus is put on the new methodology which was developed for tracker comparison, and come back to the evaluation system and the dataset in the experiments section (Section 4).

3.1 Evaluation methodology

Based on the recent analysis of widely-used performance measures [26] two weakly-correlated and easily interpretable measures were chosen: (i) accuracy and (ii) robustness.

The accuracy measures how well the bounding box predicted by the tracker overlaps with the ground truth bounding box. 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

$$\phi_t = \frac{A_t^G \cap A_t^T}{A_t^G \cup A_t^T}. \quad (1)$$

On the other hand, the robustness is 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 is indicated as soon the overlap measure (Eq. 1) drops to zero.

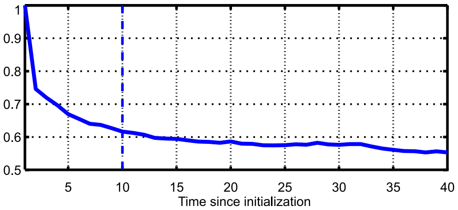

Fig. 1. Overlaps after reinitialization averaged over a large number of trackers and many reinitializations.

The reinitialization of trackers might introduce a bias into the performance measures. If a tracker fails at a particular frame it will likely fail again immediately after re-initialization. To reduce this bias, the tracker is re-initialized five frames after the failure. This number was determined experimentally on a separate dataset. A similar bias occurs in the accuracy measure. The overlaps in the frames right after the initialization are biased towards higher values over several frames and it takes a few frames of the burn-in period to reduce the bias. In a preliminary study we have determined by a large-scale experiment that the burn-in period is approximately ten frames (Figure 1). This means that we label the frames in the burn-in period as invalid and do not use them in computation of the accuracy.

A tracker is run on each sequence N_{rep} times to obtain a better statistic on its performance. In particular, let $\Phi_t(i,k)$ denote the accuracy of i-th tracker at frame t at experiment repetition k. The per-frame accuracy is obtained by taking the average over these, i.e., $\Phi_t(i) = \frac{1}{N_{\mathrm{rep}}}\sum_{k=1}^{N_{\mathrm{rep}}} \Phi_t(i,k)$. The average accuracy of the i-th tracker, $\rho_{\mathrm{A}}(i)$, over some set of N_{valid} valid frames is then calculated as the average of per-frame accuracies

$$\rho_{\mathrm{A}}(i) = \frac{1}{N_{\mathrm{valid}}} \sum_{j=1}^{N_{\mathrm{valid}}} \Phi_j(i). \quad (2)$$

In contrast to accuracy measurements, a single measure of robustness per experiment repetition is obtained. Let $F(i,k)$ be the number of times the i-th tracker failed in the experiment repetition k over a set of frames. The average robustness of the i-th tracker is then

$$\rho_R(i) = \frac{1}{N_{\mathrm{rep}}} \sum_{k=1}^{N_{\mathrm{rep}}} F(i,k). \quad (3)$$

In a typical dataset some visual attributes tend to be more frequently presented than the others, which would introduce a bias into the results. To address this, we calculate the accuracy (2) and robustness (3) separately for each attribute. For a particular attribute we calculate the two measures only on the subset of frames in the dataset that contain that attribute (attribute subset). To compare different trackers one might average the accuracy and robustness over all the attribute subset frames. However, these will likely be at a different scale across the attribute sequences in which case direct averaging of performance measures is not appropriate. Instead, we have developed a ranking-based methodology akin to [34], [35], [9]. We start by ranking all the trackers with respect to each measure on each attribute subset separately. Let $r(i,a,m)$ be the rank of the i-th tracker on the attribute subset a using the performance measure m. Now we can calculate the average rank for the i-th tracker by averaging over the attributes $r(i,m) = \frac{1}{N_{\mathrm{att}}} \sum_{a=1}^{N_{\mathrm{att}}} r(i,a,m)$. Giving an equal weight to each performance measure, we average the two corresponding rankings as

$$r(i) = \frac{1}{2} \sum_{m \in \{A,R\}} r(i,m), \tag{4}$$

where A stands for accuracy and R for robustness. The averaging over attribute subsets assures that every attribute contributes equally to the final ranking. Since the frequency of the attributes is uneven and some frames contain several attributes, it means that some frames contribute more than the other to the final rank. This is a subtlety that might not be immediately apparent, but has to be kept in mind when interpreting the results.

A group of trackers may perform equally well on a given attribute subset, in which case they should be assigned an equal rank. In particular, after ranking trackers on an attribute subset, the corrected ranks are calculated as follows. 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, is determined. The corrected rank of the i-th tracker is then calculated as the average of the ranks in the group of equivalent trackers. Note that this equality is not transitive, and should not be mistaken for a classical equivalence relation. For example, consider trackers T_1, T_2 and T_3. It may happen that a tracker T_2 performs equally well as T_1 and T_3, but this does not necessarily mean that T_1 performs equally well as both, T_2 and T_3. The equality relation between trackers should therefore be established for each tracker separately.

To determine for each tracker the group of equivalent trackers, a measure of equivalence on a given sequence is required. 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. When the differences are distributed normally, the Student's t-test, which is often used in the aeronautic tracking research [36], is the appropriate choice. However, in a preliminary study we have applied Anderson-Darling tests of normality [37] and have observed that the accuracies in frames are not always distributed normally, which might render the t-test inappropriate. Figure 2 further confirms this visually on an example of a typical histogram of differences from our evaluation. As an alternative, the Wilcoxon Signed-Rank test as in [34] is applied. In case of robustness, several measurements of the number of times the tracker failed over the entire sequence in different runs is obtained. However, these cannot be paired, and the Wilcoxon Rank-Sum (also known as Mann-Whitney U-test) [34] is used instead to test the difference in the average number of failures.

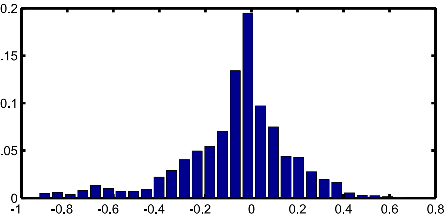

Fig. 2. An example of accuracy differences histogram.

When establishing an equivalence between two trackers, we have to keep in mind that statistical significance of performance differences does not directly imply a practical difference [38]. It is necessary to define a maximal difference in performance of two trackers at which both trackers are said to perform practically equally well. By varying the practical difference from 0.0 to 0.1 no significant changes in ranking were observed. However, since we could not find clear means to objectively define this difference, we reserve our methodology only to testing the statistical significance of differences.

3.1.1 Visualization of results

Results can be visualized per visual attribute using the accuracy-robustness plots proposed by [26]. Since we have extended the methodology of [26] to rankings, we also extend the visualization. In particular, we display the rank results either over the particular experiment or entire set of experiments using the accuracy-robustness (A-R) rank plots. Since each tracker is presented in terms of its rank with respect to robustness and accuracy, we can plot it as a single point on the corresponding 2D A-R rank plot as shown in Figure 3. Trackers that perform well relative to the others are positioned in the top-right part of the plot, while the, relatively speaking, poorly-performing trackers occupy the bottom-left part.

4 Experimental evaluation

4.1 VOT2013 challenge

The tracker comparison methodology from Section 3 was applied to a large-scale experiment. Instead of implementing various existing trackers ourselves, an evaluation kit was developed and a Visual object tracking challenge[5] (VOT2013) was organised. The evaluation kit includes the evaluation system and the dataset. Researchers were invited to participate by downloading the evaluation kit, integrate their tracker with the evaluation system and run it locally on their machines. The raw results from the evaluation system were then submitted

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

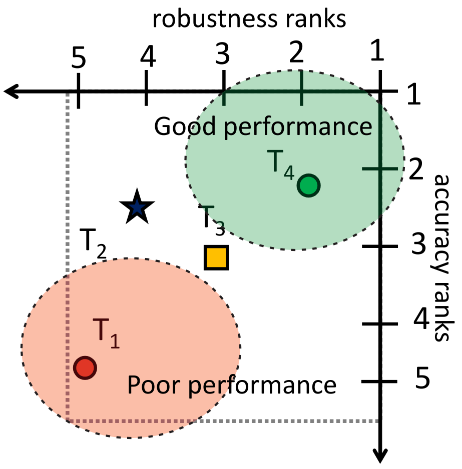

Fig. 3. Example of an A-R rank plot.

to the VOT2013 homepage, along with a short description of the trackers and optionally with the binaries or source code to allow the VOT2013 committee further verification of their results.

4.1.1 Evaluation system

The VOT2013 evaluation system was implemented in Matlab/Octave to fulfill the multi-platform, multi-programming language compatibility requirement from Section 1.1. A minimal API is defined on how to integrate a tracker with the system regardless of the programming language used to implement the tracker. A preset experiment is executed as follows. The evaluation system compiles a sequence of frames and provides the tracker with the path to the sequence as well as the initial bounding box. The tracker is run to track until the end of the sequence. The system then verifies the per-frame outputs to detect a tracking failure. Upon failure, the system recompiles the sequence with the first frame starting after the point of the detected failure and runs the tracker again. This keeps the API simple, while fulfilling the requirement of interaction with the tracker. The reader is referred to the evaluation kit document [39] for further details.

4.1.2 Dataset

To fulfil the attribute diversity requirement, one could in principle collect all the sequences from existing datasets into a new dataset. However, note that a big dataset does not necessarily mean rich in visual properties. In fact, many sequences may be visually similar and would not contribute to diversification of the dataset, while they would significantly prolong the execution of the experiments. We have therefore applied an approach that would lead to a dataset that includes various visual phenomena, while containing a small number of sequences to keep the time for performing the experiments reasonably low.

A large pool of sequences that have been used by various authors in the tracking community[6] was collected and six global attributes on each sequence were computed:

- *illumination change* defined as the maximal difference in object intensity;
- *object size change* calculated as the average of sequential differences in the ground-truth bounding box size;
- *object motion* calculated as the average of changes in bounding box center over the frames;
- *clutter* defined as the histogram difference within and outside the ground truth bounding box;
- *camera motion* calculated as the per-pixel average difference outside the bounding box;
- *blur* which was measured by a camera focus measure [40].

Table 1 lists these attributes. Thus, each sequence was encoded as a 6-dimensional feature vector. The sequences were clustered using the affinity propagation [41] into 16 clusters. From each cluster a single sequence was manually selected to keep rare short-term events like occlusion in the dataset. The resulting dataset is called the VOT 2013 dataset and consists of 16 sequences (please see the supplemental material for examples of the sequences).

TABLE 1
Performance, implementation and evaluation environment characteristics.

Attribute	Description
Illumination change	Maximal difference in object intensity
Object size change	Average of sequential differences in the ground-truth bounding box size
Object motion	Average of changes in bounding box center over the frames
Clutter	Histogram difference within and outside the ground truth bounding box
Camera motion	Per-pixel average difference outside the bounding box
Blur	Measured by a camera focus measure

Each object was annotated by an axis-aligned bounding box. Since these annotations were performed by various authors, there was no common guideline for the manual annotation. It appears that most authors followed the strategy of maintaining a high foreground/background ratio within the bounding box (at least $> 60\%$). In most cases, this ratio 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 large portion of the background at

[6] Note that at the time of the dataset preparation the ALOV dataset was not yet published.

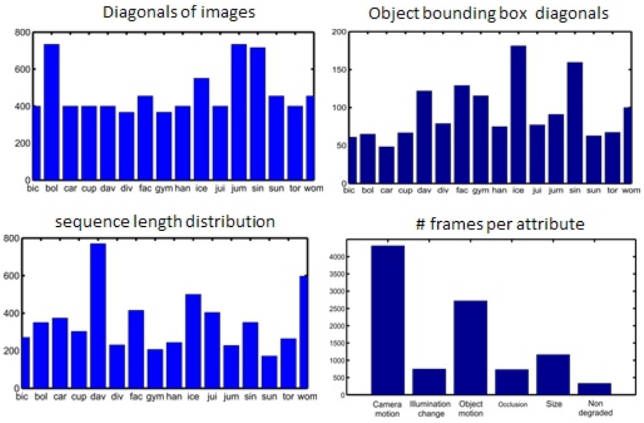

Fig. 4. Some general statistics of the VOT2013 dataset.

some frames as well. After inspecting all the bounding box annotations, those sequences in which the original annotations were out of place were re-annotated.

Additionally, we labeled each frame in each selected sequence with five visual attributes that reflect a particular challenge in appearance degradation: occlusion, illumination change, motion change, size change and camera motion. In case a particular frame did not correspond to any of the five degradations, we denoted it as non-degraded. Some general statistics of the dataset are shown in Figure 4.

4.1.3 The experiments

The VOT2013 challenge included the following three experiments:

- Experiment 1 (Baseline): This experiment tested a tracker on all sequences in the VOT2013 dataset by initializing it on the ground truth bounding boxes.
- Experiment 2 (Noise): This experiment performed Experiment 1, but initialized with noisy bounding box. The bounding boxes were randomly perturbed in position and size by drawing perturbations uniformly from ±10% interval of the ground truth bounding box size.
- Experiment 3 (Grayscale): This experiment performed the experiment 1 on all sequences with the color images changed to grayscale.

Trackers that did not use the color information only run Experiment 3 and the same results were assumed also for the Experiment 1. All the experiments were automatically performed by the VOT2013 evaluation kit[7]. A tracker was run on each sequence 15 times to obtain a better statistic on its performance.

4.1.4 Submitted trackers

In total 27 trackers were considered in the challenge. This included 19 original submissions and 8 baseline highly-cited trackers that were contributed by the VOT committee. The set was very diverse, ranging from trackers that use background-subtraction (MORP [17], STMT [17]), based on optical-flow or motion cues (FoT [42], TLD [43], SwATrack [44]), key-points (SCTT [17], Matrioska [45]), complex generative visual models (IVT [19], Meanshift [46], CCMS [17], DFT [47], ORIA [48], EDFT [49], AIF [50], CACTuS-FL [51], PJS-S [52], SwATrack [44]) or discriminative visual models (MIL [27], Struck [53], PLT [17], CT [54], RDET [55], ASAM [17], GSDT [56]), to trackers that use geometrical constellation of parts (HT [57], LGT [58], LGT++ [59], LT-FLO [60]) (see Table 2). For a detailed description of all trackers, the reader is referred to the VOT2013 challenge report [17] and the to supplemental material.

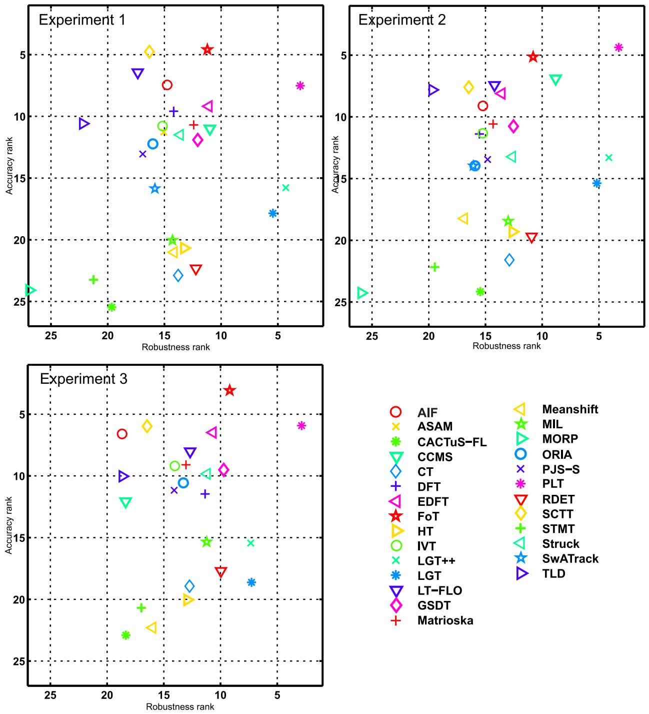

Fig. 5. The accuracy-robustness ranking plots with respect to the three experiments. The best tracker would reside in the top-right corner of the plot.

4.2 Results of VOT2013 experiments

Ranking results are summarized in Table 2 (for further details on rankings and plots please see the VOT2013 results homepage). For some of the trackers, the working binaries were obtained as well and we have further verified the submitted results for these trackers by re-running parts of the experiments. The verified trackers are denoted by $(\cdot)^*$.

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,

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

TABLE 2

Ranking results. The highest ranking trackers are marked by red, blue and green, respectively. The average ranking over all three experiments is shown in the last column. The trackers which were verified by the VOT committee are denoted by $(\cdot)^*$. The tracker implementation speed is denoted by sub-realtime ($-$), realtime (○), above-realtime ($+$), while N, M and S stand for native (C/C++), Matlab and Matlab/C++ implementation.

		Experiment 1			Experiment 2			Experiment 3			
	Speed	R_A	R_R	R	R_A	R_R	R	R_A	R_R	R	R_Σ
PLT* [17]	$+$ N	7.51	**3.00**	**5.26**	**4.38**	**3.25**	**3.81**	*5.90*	**2.83**	**4.37**	**4.48**
FoT* [42]	$+$ N	**4.56**	11.15	*7.85*	*5.14*	10.84	7.99	**3.08**	9.19	*6.13*	*7.33*
EDFT* [49]	$-$ S	9.14	11.04	10.09	8.14	13.61	10.88	6.52	10.66	8.59	9.85
LGT++* [59]	$-$ M	15.73	*4.25*	9.99	13.36	*4.14*	8.75	15.46	7.34	11.40	10.05
LT-FLO [60]	$-$ M	6.40	17.40	11.90	7.43	14.27	10.85	8.00	12.63	10.31	11.02
GSDT [56]	$-$ S	11.87	11.99	11.93	10.78	12.56	11.67	9.49	9.72	9.60	11.07
SCTT [17]	$-$ S	*4.75*	16.38	10.56	7.65	16.49	12.07	6.00	16.49	11.24	11.29
CCMS* [17]	$+$ S	10.97	10.95	10.96	6.94	8.87	*7.91*	12.10	18.35	15.23	11.36
LGT* [58]	$-$ M	17.83	5.42	11.62	15.38	5.20	10.29	18.63	*7.21*	12.92	11.61
Matrioska [45]	$-$ N	10.62	12.40	11.51	10.59	14.38	12.48	9.07	13.03	11.05	11.68
AIF [50]	○ N	7.44	14.77	11.11	9.17	15.25	12.21	6.60	18.64	12.62	11.98
Struck* [53]	$-$ N	11.49	13.66	12.58	13.24	12.64	12.94	9.82	11.20	10.51	12.01
DFT [47]	$-$ S	9.53	14.24	11.89	11.42	15.58	13.50	11.44	11.32	11.38	12.25
IVT* [19]	$-$ S	10.72	15.20	12.96	11.36	15.24	13.30	9.17	14.01	11.59	12.62
ORIA* [48]	$-$ S	12.19	16.05	14.12	14.00	15.92	14.96	10.56	13.26	11.91	13.66
PJS-S [52]	$-$ M	12.98	16.93	14.96	13.50	14.84	14.17	11.19	14.05	12.62	13.92
TLD* [43]	$-$ S	10.55	22.21	16.38	7.83	19.75	13.79	10.03	18.60	14.31	14.83
MIL* [27]	$-$ N	19.97	14.35	17.16	18.46	13.01	15.74	15.32	11.17	13.24	15.38
RDET [55]	○ S	22.25	12.22	17.23	19.75	10.97	15.36	17.69	9.97	13.83	15.48
HT* [57]	$-$ N	20.62	13.27	16.95	19.29	12.61	15.95	20.04	12.90	16.47	16.46
CT* [54]	$-$ M	22.83	13.86	18.35	21.58	12.93	17.26	18.92	12.68	15.80	17.13
Meanshift* [46]	$-$ S	20.95	14.23	17.59	18.29	16.94	17.62	22.33	15.97	19.15	18.12
SwATrack [44]	$-$ N	15.81	15.88	15.84	13.97	16.06	15.02	27.00	27.00	27.00	19.29
STMT [17]	$-$ N	23.17	21.31	22.24	22.17	19.50	20.84	20.67	16.96	18.81	20.63
CACTuS-FL [51]	$-$ S	25.39	19.67	22.53	24.17	15.46	19.82	22.92	18.33	20.62	20.99
ASAM [17]	$-$ S	11.23	15.09	13.16	27.00	27.00	27.00	27.00	27.00	27.00	22.39
MORP [17]	$-$ S	24.03	27.00	25.51	24.31	26.00	25.15	27.00	27.00	27.00	25.89

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 online sparse structural SVM to adaptively learn a discriminative model from color, grayscale and grayscale derivatives. The tracker does not apply an advanced dynamic model (uniform zero-order hold model) and does not adapt the size of the target. On the other hand, FoT, LGT++ and SCTT can be thought of as part-based models, while EDFT extends the DFT tracker [47] by estimating densities of visual features in a non-parametric way. In particular, 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, SCTT uses a sparse regression for target localization and EDFT derives an enhanced computational scheme by employing the theoretic connection between averaged histograms and channel representations. 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 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 rank plots in Figure 5 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. This can be also observed in the 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, 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 6 shows the A-R rank 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++

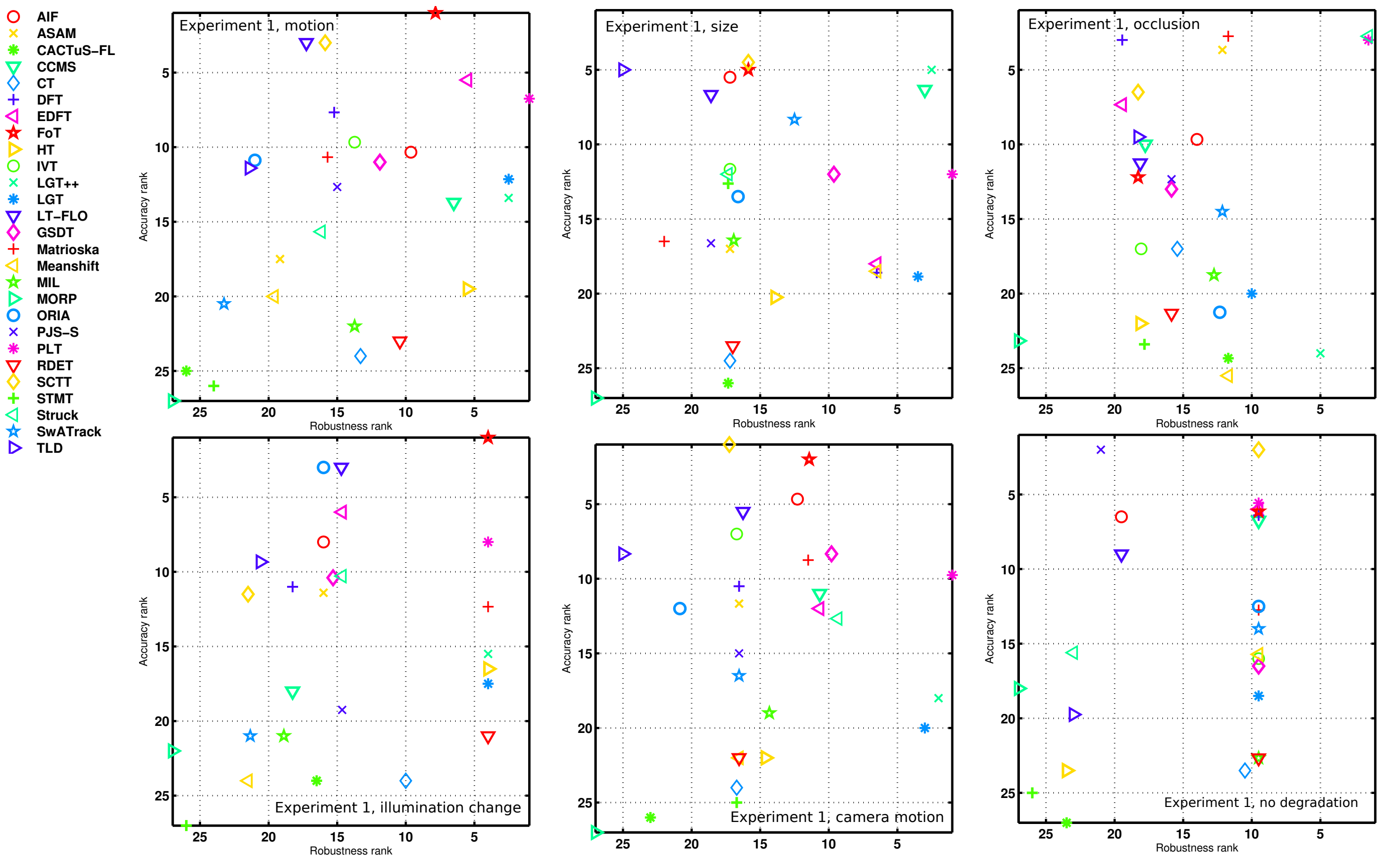

Fig. 6. The accuracy-robustness ranking plots of Experiment 1 with respect to the six sequence attributes. The tracker is better if it resides closer to the top-right corner of the plot.

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 not adapting 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 an experiment. From these measurements, the average tracking speed of each tracker[8] was estimated. 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. Trackers were classified as sub-realtime, realtime, above-realtime if their implementation frame rate was in the intervals $0-20$ fps, $20-30$ fps, >30 fps, respectively (see Table 2). The trackers PLT and FoT stand out as the fastest (above-realtime) trackers, achieving speeds in range of 150 frames per second (C++ implementations).

TABLE 3
Impact of visual attributes measured as average 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. The difficulty (of tracking) given an attributes was defined as the median of the average accuracies and failure rates. The ranking results computed from Experiment 1 are presented in Table 3. 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 in these sequences 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

[8] Please see the supplemental material for exact measurements

target and do not explicitly account 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 object size change. 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.

4.3 Results of additional VOT2013 experiments

In addition to the official challenge experiments, four additional experiments with the top five trackers where authors submitted a working executable were performed. The aim of the first experiment was to evaluate the effect of the overlap threshold, that defines tracker failure, on the ranking outcome. The remaining four experiments were designed to offer further insights into the tracker performance.

4.3.1 Effects of the failure threshold

Recall that the evaluation kit proclaimed a failure if the overlap between the predicted and ground-truth bounding box became zero. To study how increasing the threshold affects the ranking of the trackers, the baseline experiment with thresholds 0.1 to 0.5 was repeated. The results are shown in Figure 7. Note that the failure rate increased with the threshold, however, the increase was approximately the same for all five trackers up to threshold 0.3, and the ranks remain unchanged. After this, the combined ranks change slightly, while the ranks on robustness undergo a larger change for some trackers, e.g, the LGT++. For better perspective on the meaning of the overlap values, consider an overlap between two equally sized square bounding boxes. If one box is horizontally displaced w.r.t. the other by 33% of the width, the overlap becomes 0.5, and becomes 0.2 at 67% displacement. By rasing the overlap threshold, the ranking on robustness is expected to change at the point of the average overlap of a tracker – once the threshold approaches this value, the system with start detecting more failures for that tracker. This effect explains the changes in the robustness ranking at threshold 0.3. The combined ranks, however, remain approximately stable even after this threshold.

4.3.2 Sequence degradation

Four diverse challenging scenarios of sequence degradation were considered:

- **Empty frames**: Every fifth frame in the sequence is replaced by a black image to test the adaptability of the employed visual models.

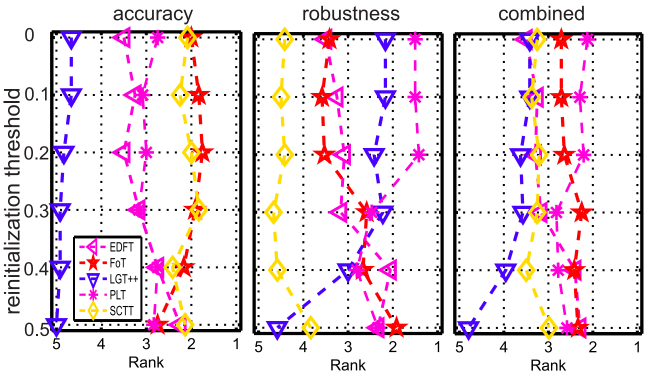

Fig. 7. Effects of failure threshold on ranking.

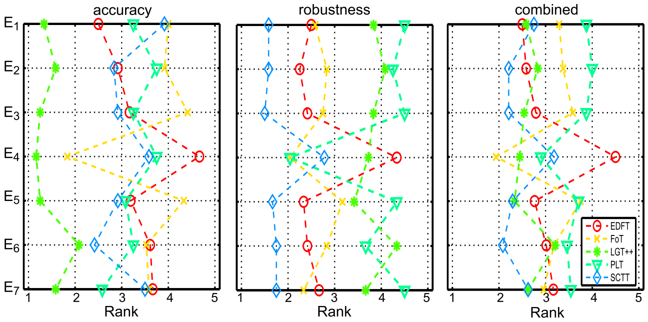

Fig. 8. Results of best performing trackers on: Baseline (E_1), Noise (E_2), Grayscale (E_3), Empty frames (E_4), Skipping frames (E_5), Frame Resize (E_6) and Reversed sequence (E_7).

- **Skipping frames**: Every third frame is removed from the sequence to simulate frame-drops that can occur in video transmission.
- **Frame resize**: The size of the images is reduced by 60% to study how the size of the target affects the tracking.
- **Reversed sequence**: The order of frames is reversed to test the importance of temporal relation in visual properties.

The overall results for the four additional experiments above are shown in Figure 8. In all but one experiment the ranking results do not change a lot, meaning that the trackers cope similarly with these degradations. Interestingly, in the Empty frames experiment, the performance of FoT and PLT relatively decreased while the performance of EDFT relatively increased. Note that the absolute performance decreased for all trackers, but this reduction was greater for FoT and PLT than it was for EDFT. The significant jump in ranking for the FoT can be explained by the way this tracker adapts its visual model. In particular, the FoT performs full adaptation in each frame. Once a black frame occurs the visual model becomes completely corrupted, which leads to failure. In case of PLT the decrease is most likely a result of fixed color model that is initialized at the first frame and is used to determine regions that most likely belong to the object. Once a black frame arrives, the discriminative

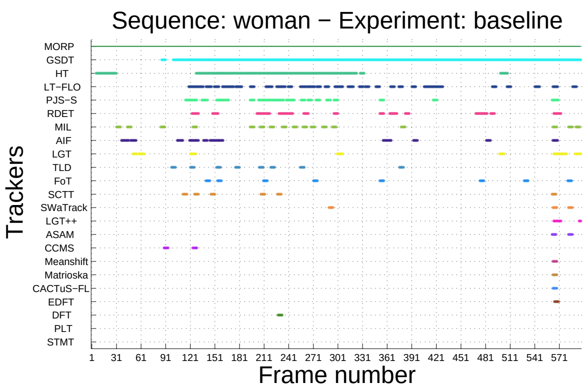

Fig. 9. Scatter plot for the *woman* sequence shows the failures for each tracker w.r.t. frame number.

power of model is rendered useless, which, may lead to unrepairable false matches of the visual model. EDFT on the other hand is better suited for this kind of changes, likely because of lazy adaptation of the visual model and a well designed motion model, which help it to survive short-term image degradations.

4.4 Results of Sequence analysis

A further analysis to gain an insight into the dataset from a tracker perspective was conducted. For each sequence we have recorded if a particular tracker failed at least once at a particular frame (Figure 9). By counting how many trackers failed at each frame, the level of difficulty can be visualized by the failure curve for each sequence (Figure 10). From these curves two measures of sequence difficulty are derived: *area* and *max*. The *area* is a sum of frame-wise values from the failure curve normalized by the number of frames, while the *max* is the maximum on this curve. While the *area* indicates the average level of difficulty of a sequence, the *max* is localized and focuses on a particular frame that presented the most difficult part of the sequence. For example, the *area* for the *david* sequence is smaller than the *area* for the *woman* sequence, which suggests that *david* sequence is less challenging that the *woman* sequence. Furthermore, a significant peak in the *woman* sequence (frame 565) suggests that this sequence contains a subsequence which is challenging to most of the trackers. Table 4 summarizes the *area* and *max* values for all sequences.

Using the *area* measure the sequences were labelled by the following three levels of difficulty: Hard (area greater than 4.00), intermediate (area between 4.00 and 2.00) and easy (area less than 1.50) (see Table 4). These levels were defined by simple observation of the areas grouping them into three different clusters. Note that the interval (2.00, 1.50) was not defined because the relative difference between areas of both sequences, singer and david, is not small ($\sim 20\%$). The *david*, *face*, *bicycle*, *juice*, *jump*, *car* and *cup* sequences do not present a significant challenge to most of the trackers. On average, less than a single tracker fails. Surprisingly, the *david* sequence (Figure 10) shows a small *area* in this study, although the sequence is usually considered in literature to be challenging. One explanation might be that the trackers are overfitted to this sequence since it is so often used in evaluation and development. The alternative explanation might be that the sequence is actually not very challenging for tracking, but appears to be to a human observer. The popularity would then be explained by the fact that it is appealing to demonstrate good tracking performance on a sequence that appears difficult, even though it might not be. The analysis also shows that the *bolt*, *diving* and *hand* sequences are the most challenging ones, followed by sequences of intermediate difficulty, the *gymnastics*, *woman*, *sunshade*, *torus*, and *iceskater* sequences and the *singer* sequence which seems to be easy-to-intermediate.

sequence	area	max	frame	difficulty
bolt	4.28	13	242	hard
diving	4.23	9	105	hard
hand	4.22	14	51	hard
gymnastics	3.13	12	98	interm.
woman	2.86	15	565	interm.
sunshade	2.79	11	85	interm.
torus	2.67	8	189	interm.
iceskater	2.38	6	227	interm.
singer	1.68	4	268	interm./easy
david	1.36	4	337	easy
face	1.22	3	140	easy
bicycle	1.22	11	178	easy
juice	1.12	4	242	easy
jump	0.93	4	203	easy
car	0.92	5	253	easy
cup	0.22	2	232	easy

TABLE 4
Analysis of sequence difficulty. The area under the failure curve (area), the maximal number of simultaneously failed trackers (max), the frame number with maximum number of failures (frame), and the difficulty category (difficulty).

Most of the difficulties in hard and intermediate sequences arise from changes in camera and object motion as well as from rapid changes in object size. For example, *bolt* is hard, as all three aforementioned nuisances occur simultaneously in the sequence. The *diving* sequence shows significant changes in object size while the *hand* sequence shows challenging pose variations of the person's hand.

Easy to intermediate sequences might remain valuable for tracker comparison as these sequences still conceal challenges in particular frames. These sequences are identified by considering *max* in Table 4. For example, the *woman* sequence at frame 565 (Figure 10) contains camera zooming which makes 15 out of 23 trackers fail. Similarly, the *bicycle* sequence at frame 178 (Figure 10) shows a peak in the failure curve. In this part of the sequence, an object is occluded, which is immediately followed by a shadow cast over the target. A significant peak is also present in the *bolt* sequence (Figure 10) at frame 242. Almost half of the trackers fail here. A closer

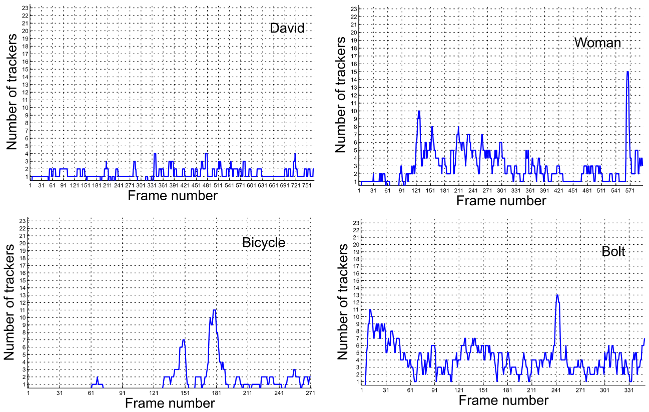

Fig. 10. Failure curves for *david*, *woman*, *bicycle* and *bolt* sequences.

look at the frame and its neighboring frames shows significant object motion between frames as a cause of failures.

5 Conclusion

In this paper a novel tracker performance evaluation methodology was presented. The performance measures, the dataset and the evaluation system were identified as the most important components of tracker evaluation and critical requirements for each of these components were proposed. In line with the requirements, a new evaluation methodology is proposed which aims at simple, easily interpretable, tracker comparison and includes objective comparison in terms of statistical significance. Besides, a new dataset and a cross-platform-compatible evaluation system were presented. The dataset consists of 16 sequences, which are per-frame annotated by visual attributes. Furthermore, a new dataset analysis from the tracking perspective was proposed. The power of the proposed tracker comparison approach and analysis was evaluated on a VOT2013 challenge with 27 trackers using the 16 sequences.

The results of an exhaustive analysis 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. However, there is some evidence showing that accuracy tends to be better for the trackers that do not apply global models, but rather split their visual models into parts. On the other hand, robustness is achieved by discriminative learning where variants of structured SVM, e.g. PLT, seem promising. Analysis of the new dataset showed that some sequences are challenging on average, other sequences are very challenging at particular frames, and some of them are well tackled by all the trackers. An interesting find is that one particular sequence (David), which is usually assumed challenging in the tracking community, seems not to be according to the presented analysis, as trackers rarely fail on this sequence.

While establishing standard datasets and evaluation methodology tends to result in significant short-term advances in the field, it can also have negative effects, leading to empoversihed specter of approaches that get put forward in the long run [61]. Evaluation is often reduced to a single performance score, which might lead to degradation in research. The primary goal of authors, i.e. comming up with new tracking concepts, shifts to increasing a single performance score, and this is further enforced by pre-occupied reviewers that may find appealing to base their decision on this single score as well. We would like to explicitly warn against this. In practical experiments we are in fact comparing performance of various implementations rather than concepts. Implementations sometimes contain tweaks that improve performance, while often being left out from the original papers in interest of purity of the theory. While we believe that it is difficult to overfit a tracker to a visually diverse dataset, tuning parameters may very likely contribute to higher ranks. Because of this unavoidable dependence on implementation and effort spent in adjusting the parameters, care has to be taken when deciding for or against a new tracker based on performance scores. One approach might be to apply comparative evaluation to position a new tracking approach against some baseline implementations using a single ranking experiment, use detailed analysis with respect to different visual properties and put further focus on the theory.

Our future work will focus on revising and carefully enriching the dataset with new sequences, e.g. including sequences from related datasets like the recent ALOV [15], with the aim to significantly increase the diversity of objects and visual attributes while keeping the number of sequences at a useful level. We also intend to improve the evaluation system, allowing faster execution of more complex experiments. Our work will focus on continual improvement of tracker evaluation methodology and, through further organization of the VOT challenges, pushing towards a standardised tracker comparison.

Acknowledgments

This work was supported in part by the following research programs and projects: Slovenian research agency research programs and projects P2-0095, P2-0214, J2-4284, J2-3607, the EU project EPiCS (grant agreement no 257906), the CTU Project SGS13/142/OHK3/2T/13 and by The Czech Science Foundation Project GACR P103/12/G084.

References

[1] D. M. Gavrila, "The visual analysis of human movement: A survey," *Comp. Vis. Image Understanding*, vol. 73, no. 1, pp. 82–98, 1999.

[2] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Comp. Vis. Image Understanding*, vol. 81, no. 3, pp. 231–268, March 2001.

[3] 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*, 2003, pp. 166–173.

[4] 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*, vol. 34, no. 30, pp. 334–352, 2004.

[5] T. B. Moeslund, A. Hilton, and V. Kruger, “A survey of advances in vision-based human motion capture and analysis,” *Comp. Vis. Image Understanding*, vol. 103, no. 2-3, pp. 90–126, November 2006.

[6] A. Yilmaz and M. Shah, “Object tracking: A survey,” *Journal ACM Computing Surveys*, vol. 38, no. 4, 2006.

[7] 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.

[8] 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*, 2005, pp. 317–324.

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

[10] 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.*, vol. 22, no. 10, pp. 1090–1104, 2000.

[11] 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.*, vol. 31, no. 2, pp. 319–336, 2009.

[12] B. Karasulu and S. Korukoglu, “A software for performance evaluation and comparison of people detection and tracking methods in video processing,” *Multimedia Tools and Applications*, vol. 55, no. 3, pp. 677–723, 2011.

[13] S. Salti, A. Cavallaro, and L. Di Stefano, “Adaptive appearance modeling for video tracking: Survey and evaluation,” *IEEE Trans. Image Proc.*, vol. 21, no. 10, pp. 4334 – 4348, 2012.

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

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

[16] Y. Pang and H. Ling, “Finding the best from the second bests – inhibiting subjective bias in evaluation of visual tracking algorithms,” in *Int. Conf. Computer Vision*, 2013.

[17] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, F. Porikli, L. Čehovin, G. Nebehay, G. Fernandez, T. Vojir, A. Gatt, A. Khajenezhad, A. Salahledin, A. Soltani-Farani, A. Zarezade, A. Petrosino, A. Milton, B. Bozorgtabar, B. Li, C. S. Chan, C. Heng, D. Ward, D. Kearney, D. Monekosso, H. C. Karaimer, H. R. Rabiee, J. Zhu, J. Gao, J. Xiao, J. Zhang, J. Xing, K. Huang, K. Lebeda, L. Cao, M. E. Maresca, M. K. Lim, M. E. Helw, M. Felsberg, P. Remagnino, R. Bowden, R. Goecke, R. Stolkin, S. Y. Lim, S. Maher, S. Poullot, S. Wong, S. Satoh, W. Chen, W. Hu, X. Zhang, Y. Li, and Z. Niu, “The Visual Object Tracking VOT2013 challenge results,” in *ICCV Workshops*, 2013, pp. 98–111.

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

[19] D. A. Ross, J. Lim, R. S. Lin, and M. H. Yang, “Incremental learning for robust visual tracking.” *Int. J. Comput. Vision*, vol. 77, no. 1-3, pp. 125–141, 2008.

[20] H. Li, C. Shen, and Q. Shi, “Real-time visual tracking using compressive sensing.” in *Comp. Vis. Patt. Recognition*. IEEE, 2011, pp. 1305–1312.

[21] 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*. IEEE, 2009, pp. 1208–1215.

[22] 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*, September 2005, pp. 322–327.

[23] M. Kristan, S. Kovacic, A. Leonardis, and J. Perš, “A two-stage dynamic model for visual tracking.” *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 40, no. 6, pp. 1505–1520, 2010.

[24] T. Nawaz and A. Cavallaro, “A protocol for evaluating video trackers under real-world conditions.” *IEEE Trans. Image Proc.*, vol. 22, no. 4, pp. 1354–1361, 2013.

[25] P. Carvalho, J. S. Cardoso, and L. Corte-Real, “Filling the gap in quality assessment of video object tracking,” *Image and Vision Computing*, vol. 30, no. 9, p. 630640, 2012.

[26] L. Čehovin, M. Kristan, and A. Leonardis, “Is my new tracker really better than yours?” ViCoS Lab, University of Ljubljana, Tech. Rep. 10, Oct 2013. [Online]. Available: http://prints.vicos.si/publications/302

[27] B. Babenko, M. H. Yang, and S. Belongie, “Robust object tracking with online multiple instance learning.” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, 2011.

[28] H. Wu, A. C. Sankaranarayanan, and R. Chellappa, “Online empirical evaluation of tracking algorithms.” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1443–1458, 2010.

[29] J. SanMiguel, A. Cavallaro, and J. Martnez, “Adaptive on-line performance evaluation of video trackers,” *IEEE Trans. Image Proc.*, vol. 21, no. 5, pp. 2812–2823, 2012.

[30] D. M. Chu and A. W. M. Smeulders, “Thirteen hard cases in visual tracking,” in *IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2010.

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

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

[33] D. Doermann and D. Mihalcik, “Tools and techniques for video performance evaluation,” in *Proc. Int. Conf. Pattern Recognition*, 2000, p. 167170.

[34] J. Demšar, “Statistical comparisons of classifiers over multiple datasets,” *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[35] 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*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

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

[37] T. W. Anderson and D. A. Darling, “Asymptotic theory of certain ‘goodness of fit’ criteria based on stochastic processes,” *The Annals of Mathematical Statistics*, vol. 23, no. 2, pp. 193–212, 1952.

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

[39] M. Kristan and L. Čehovin, *Visual Object Tracking Challenge (VOT2013) Evaluation Kit*, Visual Object Tracking Challenge, 2013.

[40] M. Kristan, J. Pers, M. Perse, and S. Kovacic, “Bayes spectral entropy-based measure of camera focus,” in *Computer Vision Winter Workshop*, February 2005, pp. 155–164.

[41] B. J. Frey and D. Dueck, “Clustering by passing messages between data points,” *Science*, vol. 315, pp. 972–976, 2007.

[42] T. Vojir and J. Matas, “Robustifying the flock of trackers,” in *Comp. Vis. Winter Workshop*. IEEE, 2011, pp. 91–97.

[43] Z. Kalal, K. Mikolajczyk, and J. Matas, “Tracking-learning-detection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 7, pp. 1409–1422, 2012.

[44] M. Lim, C. Chan, D. Monekosso, and P. Remagnino, “Swatrack: A swarm intelligence-based abrupt motion tracker,” in *Proceedings of IAPR MVA*, 2013, p. pages 3740.

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

[46] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-based object tracking,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 5, pp. 564–577, 2003.

[47] L. Sevilla-Lara and E. G. Learned-Miller, “Distribution fields for tracking,” in *Comp. Vis. Patt. Recognition*. IEEE, 2012, pp. 1910–1917.

[48] Y. Wu, B. Shen, and H. Ling, “Online robust image alignment via iterative convex optimization,” in *Comp. Vis. Patt. Recognition*. IEEE, 2012, pp. 1808–1814.

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

[50] 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.

[51] 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*. IEEE, 2010, pp. 1–8.

[52] 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.

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

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

[55] 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.

[56] 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.

[57] M. Godec, P. M. Roth, and H. Bischof, "Hough-based tracking of non-rigid objects," *Comp. Vis. Image Understanding*, vol. 117, no. 10, pp. 1245–1256, 2013.

[58] L. Čehovin, M. Kristan, and A. Leonardis, "Robust visual tracking using an adaptive coupled-layer visual model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 4, pp. 941–953, 2013.

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

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

[61] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *Comp. Vis. Patt. Recognition*. IEEE, 2011, pp. 1521–1528.

Matej Kristan received a Ph.D from the Faculty of Electrical Engineering, University of Ljubljana in 2008. He is an Assistant Professor at the ViCoS Laboratory at the Faculty of Computer and Information Science and at the Faculty of Electrical Engineering, University of Ljubljana. His research interests include probabilistic methods for computer vision with focus on visual tracking, dynamic models, online learning, object detection and vision for mobile robotics.

Roman Pflugfelder received his Ph.D. degree at the Institute of Computer Graphics at Graz University of Technology in 2008. Currently, Dr Pflugfelder is scientist and project leader at the Austrian Institute of Technology. His research interests lie in object tracking and camera autocalibration for video surveillance.

Jiri Matas is a Professor at the Center for Machine Perception, Faculty of Electrical Engineering at the Czech Technical University in Prague, Czech Republic. He is author or co-author of more than 250 papers in the area of Computer Vision and Machine Learning. His research interests include object recognition, image retrieval, tracking, sequential pattern recognition, invariant feature detection and Hough Transform and RANSAC-type optimization.

Aleš Leonardis is a Professor at the School of Computer Science, University of Birmingham and co-Director of the Centre for Computational Neuroscience and Cognitive Robotics. He is a professor at the FCIS, University of Ljubljana and adjunct professor at the FCS, TU-Graz. His research interests include robust and adaptive methods for computer vision, object and scene recognition and categorization, statistical visual learning, 3D object modeling, and biologically motivated vision.

Fatih Porikli is a Professor at the Australian National University in Canberra. Previously, Professor Porikli was a Distinguished Research Scientist at Mitsubishi Electric Research Labs (MERL). He received his PhD degree, with specialization in video object segmentation, from NYU Poly, USA. He has authored more than 100 publications and invented more than 60 patents. His work covers areas including computer vision, machine learning, video surveillance, multimedia processing among others.

Georg Nebehay received his Master Degree in Computer Science in 2011 from Vienna University of Technoogy, Austria. He is currently PhD student at the Austrian Institute of Technology. His research interests are tracking and video surveillance.

Gustavo Fernandez is scientist at the Video and Security Technology Group, Austrian Institute of Technology. He obtained his master degree from the University of Buenos Aires (Argentina) in 2000 and his PhD from Graz University of Technology (Austria) in 2004. Since 2005 he is at the Austrian Institute of Technology doing research in the area of Computer Vision. His research interests are both theory and applications of object detection, object tracking and cognitive systems.

Tomáš Vojíř received a Master degree at the Center for Machine Perception, Faculty of Electrical Engineering at the Czech Technical University in Prague in 2010 and is currently pursuing his PhD at the same faculty. His research interests are computer vision with focus in real-time visual object tracking.

Luka Čehovin received his Dipl.ing. and M.Sc. degree at the Faculty of Computer Science and Informatics, University of Ljubljana, Slovenia in 2007 and 2010, respectively. Currently he is working at the Visual Cognitive Systems Laboratory, Faculty of Computer Science and Informatics, University of Ljubljana, Slovenia as an assistant and a researcher. His research interests include computer vision, HCI, distributed intelligence and web-mobile technologies.