Performance Evaluation Methodology for Long-Term Single Object Tracking

Alan Lukežič*[1], Luka Čehovin Zajc*[1], Tomáš Vojíř[2], Jiří Matas[2] and Matej Kristan[1]
[1]Faculty of Computer and Information Science, University of Ljubljana, Slovenia
[2]Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic
* The authors contributed equally
{alan.lukezic,luka.cehovin}@fri.uni-lj.si

***Abstract*—A long-term visual object tracking performance evaluation methodology and a benchmark are proposed. Performance measures are designed by following a long-term tracking definition to maximize the analysis probing strength. The new measures outperform existing ones in interpretation potential and in better distinguishing between different tracking behaviors. We show that these measures generalize the short-term performance measures, thus linking the two tracking problems. Furthermore, the new measures are highly robust to temporal annotation sparsity and allow annotation of sequences hundreds of times longer than in the current datasets without increasing manual annotation labor. A new challenging dataset of carefully selected sequences with many target disappearances is proposed. A new tracking taxonomy is proposed to position trackers on the short-term/long-term spectrum. The benchmark contains an extensive evaluation of the largest number of long-term trackers and comparison to state-of-the-art short-term trackers. We analyze the influence of tracking architecture implementations to long-term performance and explore various re-detection strategies as well as influence of visual model update strategies to long-term tracking drift. The methodology is integrated in the VOT toolkit to automate experimental analysis and benchmarking and to facilitate future development of long-term trackers.**

***Index Terms*—Visual object tracking, long-term tracking, performance measures, tracking benchmark.**

I. Introduction

VISUAL object tracking has significantly advanced over the last decade with emergence of standard datasets, performance evaluation protocols [1], [2], [3], [4], [5] and tracking challenges [6], [5].

Popular single-target tracking benchmarks [7], [2], [3], [4] focus on short-term trackers. The introduction of gradually more demanding benchmarks lead to the development of short-term trackers that cope well with significant appearance and motion changes and are robust to short-term occlusions. Several recent publications [8], [9], [10] show that short-term trackers fare poorly on very long sequences since localization errors and updates gradually deteriorate their visual model, leading to drift and failure. Typically, short-term trackers assume that the target is always in the field of view (this is reflected in the standard dataset). When this is not the case, the short-term tracker fails, forever.

Long-term trackers are designed for scenarios where the target may disappear from the field of view, may be fully occluded for long periods of time and where cuts, i.e. unpredictable abrupt changes of target pose and appearance, may occur. A long-term tracker thus requires to have the ability to report that the target is not present, e.g. by providing a confidence score of the estimated pose, which may be binary or continuous, with low confidence suggesting the target is absent. A crucial difference to short-term tracking is thus the re-detection capability, i.e. the ability to localize the target when no information about current poses is available (Figure 1). This requires fundamentally different search strategies and visual model adaptation mechanisms. These long-term aspects have been explored far less than the short-term counterparts due to lack of benchmarks and performance measures probing long-term capabilities. This is the focus of our work.

Apart from coping with long sequences, long-term tracking primarily refers to the sequence properties (number of target disappearances, etc.) and the type of tracking output expected. We define the notion of the pure long-term tracker and contrast it with pure short-term tracking. We then argue there is a spectrum of tracker designs on the short-term/long-term axis and present a new tracking taxonomy for fine categorization of long-term trackers.

Based on the long-term definition we propose new performance measures, evaluation protocol and the dataset, all carefully designed to expose the long-term tracking properties. We experimentally show the proposed performance measures produce well-interpretable results. We also show significant robustness to the annotation sparsity.

Using the proposed evaluation tools, we provide an in-depth analysis of a number of long-term trackers. The analysis includes a new re-detection experiment that exposes crucial long-term tracking capabilities. The tracker performance is analyzed with respect to sequence attributes and the target disappearance rate. Analysis of long-term tracking architectures is provided as well. We test the overall performance of the architectures and analyze re-detection strategies and influence of the model update strategies on the long-term tracking drift. We make the following contributions:

- A new long-term tracking performance evaluation methodology which introduces novel performance measures. These are the first measures that principally reflect detection as well as localization accuracy in a long-term tracking domain. Comparison with exiting measures shows significant advantages in their expressive power.
- A new dataset is constructed of carefully selected sequences with a large number of target disappearances per sequence to emphasize long-term tracking properties.

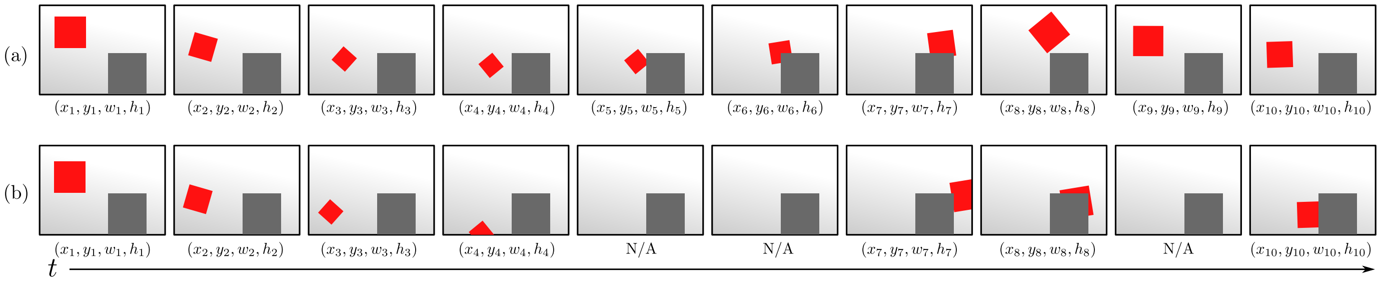

Fig. 1. Differences between short-term and long-term tracking. (a) In short-term tracking, the target, a red box, may move and change appearance, but it is always at least partially visible. (b) In long-term tracking, the box may disappear from the view or be fully occluded by other objects for long periods of time. Within these periods, the state of the object is not defined and should not be reported by the tracker.

Sequences are annotated with nine visual attributes which enable in-depth analysis of trackers.

- A detailed analysis of a number of long-term trackers covering several aspects of long-term properties, sequence attribute and target disappearance rate.
- A detailed analysis of long-term tracker architectures from perspective of re-detection and drift-prevention approaches.

All trackers, performance measures and evaluation protocol have been integrated into the VOT toolkit [4], to automate experimental analysis and benchmarking and facilitate development of long-term trackers. The dataset, all the trackers as well as the changes to the toolkit are publicly available[1].

II. Related work

Performance evaluation in single-object tracking has primarily focused on short-term trackers [7], [4], [3], [2]. The currently widely-used methodologies originate from three benchmarks, OTB [1], [7], VOT [11], [4] and ALOV [2] which primarily differ in the dataset construction, performance measures and evaluation protocols.

Benchmarks like [7], [2], [3] propose large datasets, reasoning that quantity reduces the variance in performance estimation. On the other hand, the longest-running benchmark [4] argues that quantity does not necessarily mean quality and promotes moderate-sized datasets with carefully chosen diverse sequences for fast and informative evaluation. Several works have focused on specific tracking setups. Mueller et al. [8] proposed the UAV123 dataset for tracking from drones. Galoogahi et al. [12] introduced a high-frame-rate dataset to analyze trade-offs between tracker speed and robustness. Čehovin et al. [13] proposed a dataset with an active camera view control using omni directional videos for accurate tracking analysis as a function camera motion attributes. The target never leaves the field of view in these datasets, making them unsuitable for long-term tracking properties evaluation.

Many performance measures have been explored to evaluate and rank single-target trackers [14]. All dominant short-term performance measures [7], [2], [4] are based on the overlap (intersection over union) between the ground truth bounding boxes and tracker predictions, but significantly differ in its use. ALOV [2] uses the F-measure computed at overlap threshold of 0.5. OTB [7] avoids the threshold by computing the average overlap over the sequences as the primary measure. The VOT [4] resets the tracker once the overlap drops to zero, and proposes to measure robustness by the number of times the tracker was reset, the accuracy by average overlap during successful tracking periods and an expected average overlap on a typical short-term sequence. These measures do not account for tracker ability to report target absence and are therefore not suitable for long-term tracking.

A large number of performance measures have been proposed for multi-object tracking [15], [16]. The two most widely used are MOTA and MOTP [17]. MOTA is based on counting wrong target predictions. MOTP [17] measures the average overlap on frames where target is correctly identified and the overlap is greater than 0.5. Both measures require setting a threshold that defines whether the target is successfully located. The measure are sensitive to the setting since a small change of the threshold may have a large impact on the results [4].

Another group of measures is based on target trajectories [18]. The trajectory of each annotated target in the video is classified into three classes: mostly tracked (MT), partially tracked (PT) and mostly lost (ML). These measures require (ad hoc) thresholds. Since the measures are defined on multiple trajectories, applying them to a single trajectory, which is the case in single-target tracking, translates them to success rate [1] calculated at specific thresholds.

A few papers have recently proposed datasets focusing on long-term performance evaluation. Tao et al. [10] created artificial long sequences by repeatedly playing shorter sequences forward and backward. Such a dataset exposes the problem of gradual drift in short-term trackers, but does not fully expose the long-term abilities since the target never leaves the field of view. Mueller et al. [8] proposed UAV20L dataset of twenty long sequences with target frequently exiting and re-entering the scene, but used it to evaluate mostly short-term trackers. A dataset with many cases of fully occluded and absent target has been recently proposed by Moudgil and Gandhi [9]. Unfortunately, the large number of target disappearances was obtained by significantly increasing the sequence length, which significantly increases the storage requirements. To cope with this, a very high video compression is applied, thus sacrificing the image quality.

In the absence of a clear long-term tracking definition,

[1] Dataset and evaluation available on: http://www.votchallenge.net/

much less attention has been paid to long-term performance measures. The UAV20L [8] and [9] apply the average overlap measure [7], a short-term criterion that does not account for situation when the tracker reports target absence and favors the trackers that report target positions for every frame. Tao et al. [10] adapted this measure by assigning overlap of 1 when the tracker correctly predicts the target absence. This value is not comparable with tracker accuracy when the target is visible which skews the overlap-based measure. Furthermore, reducing the actual tracking accuracy and failure detection to a single overlap score significantly limits the insight it brings.

Long-term tracker analysis requires including sequences, which are much longer than those encountered in short-term tracking evaluation. Target annotation in each frame thus significantly increases the amount of manual labor compared to short-term benchmarks. Recently, Mueller et al. [19] considered semi-automatic annotation of short-term sequences used for training localization CNNs. They annotate a single frame per-second and interpolate between them by a discriminative correlation filter. Given a typical sequence frame-rate, this means they manually annotate only every 25th frame. The amount of annotation is reduced and the quality is acceptable for training purposes, but it is not clear whether the interpolation adds bias if such an approach is used for performance evaluation.

Valmadre et al. [20] propose to completely avoid interpolation and consider only one frame per-second. They argue that sparse annotation is acceptable for long-term tracker evaluation on long sequences. Their experiment on a short-term dataset OTB100 [7] shows that evaluating at every $\sim$ 25th frame keeps the variance of their tracking performance measure within reasonable bounds. Increasing the annotation skipping length increases the variance, which could be addressed by increasing the number of sequences.

III. The Short-term/Long-term tracking spectrum

A long-term tracker is required to handle target disappearance and reappearance (Figure 1). Relatively few published trackers fully address the long-term requirements, and some short-term trackers address them partially. We argue that trackers should not be simply classified as short-term or long-term, but they rather cover an entire short-term–long-term *spectrum*. The following taxonomy is used in our experimental section for accurate performance analysis.

1) **Short-term tracker** (ST_0). The target position is reported for each frame. The tracker does not implement target re-detection and does not explicitly detect occlusion. Such trackers are likely to fail on the first occlusion as their representation is affected by any occluder.
2) **Short-term tracker with conservative updating** (ST_1). The target position is reported for each frame. Target re-detection is not implemented, but tracking robustness is increased by selectively updating the visual model depending on a tracking confidence estimation mechanism.
3) **Pseudo long-term tracker** (LT_0). The target position is reported only if the tracker believes the target is visible. The tracker does not implement explicit target re-detection but uses an internal mechanism to identify and report tracking failure.
4) **Re-detecting long-term tracker** (LT_1). The target position is reported only if the tracker believes the target is visible. The tracker detects tracking failure and implements explicit target re-detection.

The ST_0 and ST_1 trackers are what is commonly considered a short-term tracker. Typical representatives from ST_0 are KCF [21], DSST [22], SRDCF [23], CSRDCF [24], BACF [25] and CREST [26], which apply a constant visual model update. Typical examples of ST_1 are NCC [11], SiamFC [27] and the current state-of-the-art short-term trackers MDNet [28] and ECO [29]. All these trackers apply conservative updating mechanisms, which makes them ST_1 level. Many short-term trackers can be trivially converted into pseudo long-term trackers (LT_0) by using their visual model similarity scores at the reported target position. While straightforward, this offers means to evaluate short-term trackers in the long-term context.

The level LT_1 trackers are the most sophisticated long-term trackers, in that they cover all long-term requirements. These trackers typically combine two components, a short-term tracker and a detector, and implement an algorithm for their interaction. The LT_1 trackers originate from two main paradigms introduced by TLD [30] and Alien [31], with modern examples CMT [32], Matrioska [33], HMMTxD [34], MUSTER [35], LCT [36], PTAV [37], and FCLT [38].

IV. Long-term tracking performance measures

A long-term tracking performance measure should reflect the localization accuracy, but unlike short-term measures, it should also capture the accuracy of target detection capabilities (target absence prediction and target re-detection). The latter is not addressed by the standard short-term tracking measures. In detection literature [39], *precision* and *recall* measures evaluate the detector by considering the amount of predicted bounding boxes whose overlap with the ground truth bounding boxes exceeds a pre-defined threshold. However, threshold-dependent overlap measures do not fully reflect the tracking accuracy, and should be avoided [4], [14]. In the following we provide a new formulation of *tracking precision* and *tracking recall* measures which are tailored for tracking domain and avoid the deficiencies of their counterparts from the detection literature. The new measures are rigorously compared to the existing ones in Section VI-A.

Let G_t be the ground truth target pose, let $A_t(\tau_\theta)$ be the pose predicted by the tracker, θ_t the prediction certainty score at time-step t and τ_θ be a classification threshold. If the target is absent, the ground truth is an empty set, i.e., $G_t = \emptyset$. Similarly, if the tracker did not predict the target or the prediction certainty score is below a classification threshold i.e., $\theta_t < \tau_\theta$, the output is $A_t(\tau_\theta) = \emptyset$. The agreement between the ground truth and prediction is specified by their intersection over union $\Omega(A_t(\tau_\theta), G_t)$[2]. In the detection literature, the prediction matches the ground truth if the overlap $\Omega(A_t(\tau_\theta), G_t)$ exceeds a threshold τ_Ω. Given the

[2] The output of $\Omega(\cdot,\cdot)$ is 0 if either of the two regions is $\emptyset$.

two thresholds $(\tau_\theta, \tau_\Omega)$, the precision Pr and recall Re are defined as

$$\begin{aligned} &Pr(\tau_\theta, \tau_\Omega) = \\ &\qquad |\{t : \Omega(A_t(\tau_\theta), G_t) \geq \tau_\Omega \wedge A_t(\tau_\theta) \neq \emptyset\}|/N_p, \\ &Re(\tau_\theta, \tau_\Omega) = \\ &\qquad |\{t : \Omega(A_t(\tau_\theta), G_t) \geq \tau_\Omega \wedge G_t \neq \emptyset\}|/N_g, \end{aligned} \tag{1}$$

where $|\cdot|$ is the cardinality, N_g is the number of frames with $G_t \neq \emptyset$ and N_p is the number of frames with existing prediction, i.e. $A_t(\tau_\theta) \neq \emptyset$. Note that N_g is defined by ground truth and is constant for a selected sequence, while N_p is a function of the target prediction certainty threshold τ_θ.

In detection literature, the overlap threshold is set to 0.5 or higher, while recent work [4] has demonstrated that such threshold is over-restrictive and does not clearly indicate a tracking failure in practice. A popular short-term performance measure [1], for example, addresses this by averaging performance over various thresholds, which was shown in [14] to be equal to the average overlap. Using the same approach[3], we reduce the precision and recall to a single threshold by integrating over τ_Ω, i.e.,

$$\begin{aligned} Pr(\tau_\theta) &= \int_0^1 Pr(\tau_\theta, \tau_\Omega) d\tau_\Omega \\ &= \frac{1}{N_p} \sum_{t \in \{t : A_t(\tau_\theta) \neq \emptyset\}} \Omega(A_t(\tau_\theta), G_t), \end{aligned} \tag{2}$$

$$\begin{aligned} Re(\tau_\theta) &= \int_0^1 Re(\tau_\theta, \tau_\Omega) d\tau_\Omega \\ &= \frac{1}{N_g} \sum_{t \in \{t : G_t \neq \emptyset\}} \Omega(A_t(\tau_\theta), G_t). \end{aligned} \tag{3}$$

We call $Pr(\tau_\theta)$ *tracking* precision and $Re(\tau_\theta)$ *tracking* recall to distinguish them from their detection counterparts. Detection-like precision/recall plots can be drawn to analyze the tracking as well as detection capabilities of a long-term tracker (Figure 6). Similarly, a standard trade-off between the precision and recall can be computed in form of a *tracking* F-measure [39]

$$F(\tau_\theta) = 2Pr(\tau_\theta)Re(\tau_\theta)/(Pr(\tau_\theta) + Re(\tau_\theta)), \tag{4}$$

and visualized by the F-score plots (Figure 6). Our primary score for ranking long-term trackers is therefore defined as the highest F-score on the F-score plot, i.e., taken at the tracker-specific optimal threshold. This avoids manually-set thresholds in the primary performance measure. Furthermore, it avoids forcing a tracker to internally threshold its target presence uncertainty and more fairly evaluates different trackers at their optimal performance point.

Note that the proposed primary measure (4) for the long-term trackers is consistent with the established short-term tracking methodology. Consider an ST_0 short-term tracking scenario: the target is always (at least partially) visible and the target position is predicted at each frame with equal certainty. In this case our F-measure (4) reduces to the average overlap, which is a standard measure in short-term tracking [1], [4].

A. Performance evaluation protocol

A tracker is evaluated on a dataset of several sequences by initializing on the first frame of a sequence and run until the end of the sequence without re-sets. The precision-recall curve (2, 3) is calculated on each sequence and averaged into a single plot. This guarantees that the result is not dominated by extremely long sequences. The F-measure plot (4) is computed from the average precision-recall plot. The evaluation protocol along with plot generation was implemented in the VOT [4] toolkit to automate experiments and thus reduce potential human errors.

V. The long-term dataset (LTB50)

Table I quantifies the long-term statistics of the common short-term and existing long-term tracking datasets. Target disappearance is missing in the standard short-term datasets except for UAV123 which contains on average less than one full occlusion per sequence. This number increases four-fold in UAV20L [8] long-term dataset. The recent TLP [9] dataset increases the number of target disappearances by an order of magnitude, but at a cost of increasing the dataset size in terms of the number of frames by more than an order of magnitude, i.e. target disappearance events are less frequent in TLP [9] than in UAV20L [8], see Table I. Moreover, the videos are heavily compressed with many artifacts that affect tracking.

TABLE I

Datasets – comparison of long-term properties: the number of sequences, the total number of frames, the number of target disappearances (DSP), the average length of disappearance interval (ADL), the average number of disappearances in sequence (ADN). The first four datasets are short-term with virtually no target disappearances, the last column shows the properties of the proposed dataset.

Dataset	ALOV 300	OTB 100	VOT 2017	UAV 123	UAV 20L	TLP	LTB50 (ours)
# sequences	315	100	60	123	20	50	50
Frames	89364	58897	21356	112578	58670	676431	215294
DSP	0	0	0	63	40	316	525
ADL	0	0	0	42.6	60.2	64.1	52.0
ADN	0	0	0	0.5	2	6.3	10.5

In the light of the limitations of the existing datasets, we created a new long-term dataset. We followed the VOT [4] dataset construction paradigm (recently experimentally validated by other authors [40]) which states that the datasets should be kept moderately large and manageable, but rich in attributes relevant to the tested tracker class. We started by including all sequences from UAV20L since they contain a moderate occurrence of occlusions and potentially difficult to track small targets. Five long sequences with challenging targets were taken from [30]. We collected 19 additional sequences from Youtube. The sequences contain larger targets with numerous disappearances. To further increase the number of target disappearances per sequence, we have utilized the recently proposed camera view generator from omni-directional dataset AMP [13]. Six additional challenging sequences were generated from this dataset by controlling the camera such that the target was repeatedly entering and leaving the field-of-view.

[3] A detailed derivation is available in the supplementary material document.

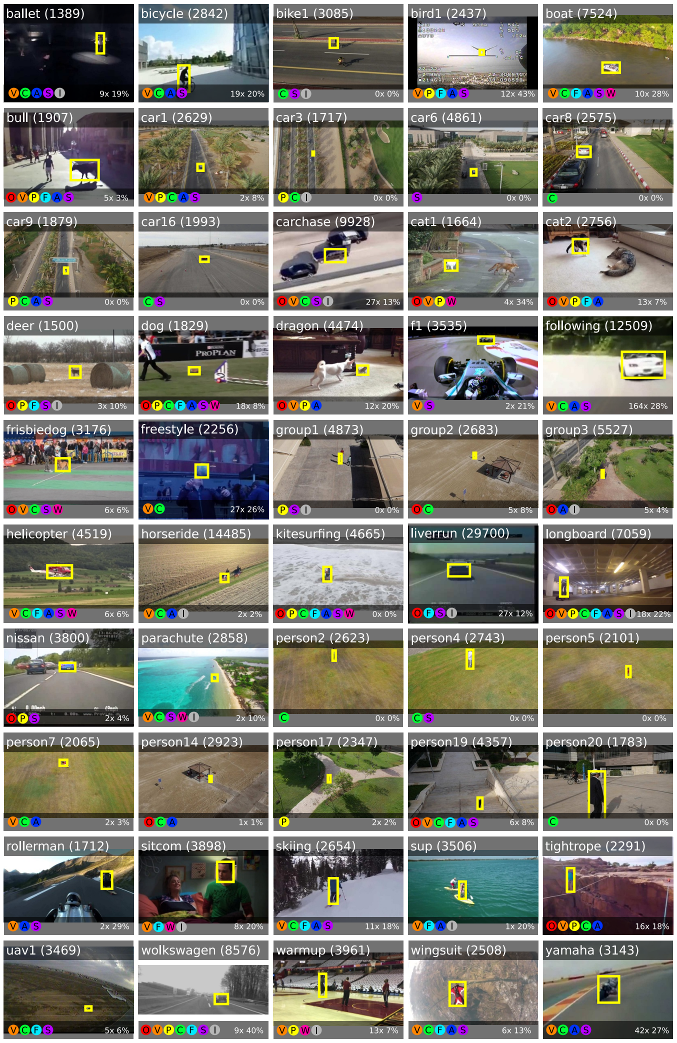

Fig. 2. The LTB50 dataset – a frame selected from each sequence. Name and length (top), number of disappearances and percentage of frames without target (bottom right). Visual attributes (bottom left): (O) Full occlusion, (V) Out-of-view, (P) Partial occlusion, (C) Camera motion, (F) Fast motion, (S) Scale change, (A) Aspect ratio change, (W) Viewpoint change, (I) Similar objects. The dataset is highly diverse in attributes, target types and contains many target disappearances.

The targets were annotated by axis-aligned bounding-boxes. (i.e., visible parts). Each sequence is annotated by nine visual attributes: full occlusion, out-of-view motion, partial occlusion, camera motion, fast motion, scale change, aspect ratio change, viewpoint change and similar objects. The LTB50 thus contains 50 challenging sequences of diverse objects (persons, car, motorcycles, bicycles, boat, animals, etc.) with the total length of $215,294$ frames. Sequence resolutions range between 1280×720 and 290×217. Each sequence contains on average 10 long-term target disappearances, each lasting on average 52 frames. An overview of the dataset is shown in Figure 2.

VI. Analysis of performance measures

A. Comparison with existing measures

Two threshold-free performance measures were recently used for long-term tracking performance evaluation [8], [10], [9]. The AUC measure is used in UAV20L [8]. As discussed in Section II, this is a primary short-term measure from [1] that computes average overlap between the tracker prediction and ground truth bounding boxes. In recent work [10], [9] AUC was adapted to account for target absence by assigning overlap of 1 to frames in which the tracker correctly predicts the target absence – which we denote by $\mathrm{AUC}_{\mathrm{mod}}$. We experimentally compare our long-term performance measures from Section IV with AUC and $\mathrm{AUC}_{\mathrm{mod}}$ using the approach with theoretical trackers introduced by Čehovin et al. [14].

The following four theoretical trackers were run on the LTB50 to expose the differences between the tested performance measures:

- $\mathrm{T}_{\mathrm{gt,gt}}$: Always reports the correct target position (the ground truth), and reports uncertainty 0 when target is visible and 1 when target is not visible.
- $\mathrm{T}_{\mathrm{gt,co}}$: Always reports the correct target position (the ground truth), and reports constant uncertainty in all frames.
- $\mathrm{T}_{\mathrm{im,co}}$: Reports a bounding box covering entire image in all frames with constant uncertainty, resulting in non-zero overlap in all frames with the target present. The optimal operation point for this tracker is thus to report target always present.
- $\mathrm{T}_{\mathrm{lost}}$: Reports a 1×1 bounding box in the top-left corner and constant uncertainty in all frames, which is interpreted as if reporting *target not visible* in all frames. In contrast to $\mathrm{T}_{\mathrm{im,co}}$, which always reports a non-zero overlap, the overlap is always zero for this tracker, thus the optimal operation point is obtained by reporting target always lost.

Results are summarized in Figure 3. The AUC [1] measure assigns equal scores to $\mathrm{T}_{\mathrm{gt,gt}}$ and $\mathrm{T}_{\mathrm{gt,co}}$. This means it does not distinguish between trackers that can detect target absence and those that cannot. Consequently this measure favors reporting the bounding box in every frame even if the target is not present.

In contrast, the modified AUC, $\mathrm{AUC}_{\mathrm{mod}}$ from [10] and [9], does distinguish between $\mathrm{T}_{\mathrm{gt,gt}}$ and $\mathrm{T}_{\mathrm{gt,co}}$. But this measure assigns a constant overlap 1 to all frames in which the target absence is correctly predicted. This is not calibrated by an average overlap when the target is present. Furthermore, since the target absence prediction and localization are mixed into a single score, it is unclear whether the high score values are mostly due to accurate prediction of the target position or the ability to correctly report target absence. For example, $\mathrm{AUC}_{\mathrm{mod}}$ assigns an average overlap of 11% to tracker $\mathrm{T}_{\mathrm{lost}}$ even though it does not make a single correct prediction of the target position. The basic AUC, on the other hand, correctly assigns a score 0 to this tracker.

Like the $\mathrm{AUC}_{\mathrm{mod}}$, the proposed tracking F-measure is capable of distinguishing between $\mathrm{T}_{\mathrm{gt,gt}}$ and $\mathrm{T}_{\mathrm{gt,co}}$. In contrast to $\mathrm{AUC}_{\mathrm{mod}}$, the basic primary measures, tracking Precision/Recall, offer a clear interpretation of the reason for the performance difference. The high tracking Precision of $\mathrm{T}_{\mathrm{gt,gt}}$ indicates a better target absence prediction compared to $\mathrm{T}_{\mathrm{gt,co}}$. But both trackers equally accurately predict target position when visible, which results in an equally high tracking Recall.

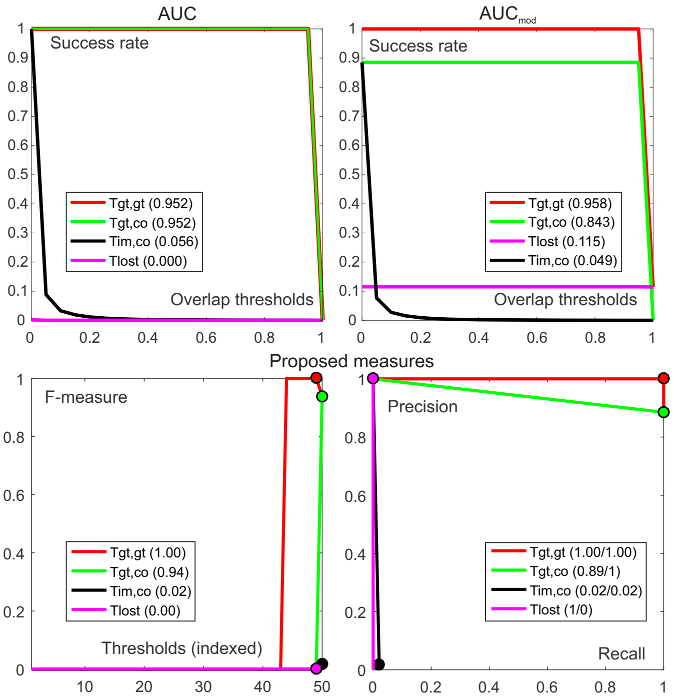

Fig. 3. Tracking performance of the four trackers described in Section VI-A evaluated by the proposed and existing performance measures. The success rate plots corresponding to AUC and $\mathrm{AUC}_{\mathrm{mod}}$ are shown in the first two plots, while the second two show the tracking F-measure and tracking Precision/Recall graphs. The overall tracker scores are shown in parentheses next to the tracker labels with the two values in the bottom-right plot showing the precision and recall values.

Another example is the tracker $\mathrm{T}_{\mathrm{lost}}$. The tracking F-score is zero, indicating a complete tracking failure and the Recall zero means that the reason is inability to localize the target.

B. Robustness to annotation sparsity

Manual annotation of every frame in long-term sequences requires a significant amount of manual labor, since these are often an order of magnitude longer than short-term sequences. An approach to reduce the labor is annotating every N-th frame [20]. The amount of skipped frames is typically constrained by the robustness of the performance measure. We utilize densely annotated long-term sequences in LTB50 to test the behavior of the performance measures introduced in Section IV with respect to the annotation sparsity.

A set of trackers described in Section VII-A was run on the LTB50 dataset. The trackers were evaluated by computing the tracking Precision, Recall and F-measure by considering every N-th frame with $N \in [1, 12, 25, 50, 100, 200]$. This is equal to annotating every 0.04s, 0.5s, 1s, 2s, 4s and 8s assuming a 25fps frame rate. Figure 4 shows the behavior of the performance measures with increasing annotation sparsity.

The tracking Precision/Recall deviate a bit at very high annotation sparsity levels, but largely maintain the order of trackers. A striking result is that the deviations in Precision/Recall appear to cancel out in tracking F-score, which maintains extremely stable results over *the whole range* of annotation sparsity levels.

Detailed performance analysis can be carried out with annotations every 25 or 50 frames since the measure values minimally differ from those obtained from dense annotations. The annotation interval length of 50 frames is theoretically supported by the average target disappearance period length (which is 52 frames in our dataset), and means that annotating the target position every 50 frames ensures that most of the target disappearance events will be covered by at least one annotation.

Note that situations in which a tracker fails for a very few frames and recovers may not be detected if they occur in between the consecutive annotations. If these are rare events, they would not have affected the overall performance measure even if they were detected due to averaging over a large number of frames. However, if these occur frequently, they will be detected due to non-regularity of failure intervals in real world setups, and the performance measures will reflect such behavior. This property is supported experimentally verified in Figure 4 – the measures are stable even at sparsity levels that significantly exceed the average disappearance period (52 frames).

Our results imply that sequences 50 times longer than typical short-term sequences can be annotated with the same amount of manual labor without losing analysis accuracy. If only an overall performance analysis is required (tracking F-measure) then the annotation may be even sparser, allowing *up to 200 times longer sequences* at a moderate annotation effort.

VII. Long-term tracking evaluation

A. Evaluated trackers

An extensive collection including top-performing trackers was complied to cover the short-term–long-term spectrum. In total, twenty trackers summarized in Table II and Figure 9 were evaluated. We included eight long-term state-of-the-art trackers with publicly available source code: (i) TLD [30], which uses optical flow for short-term component and normalized-cross-correlation for detector and a P-N learning framework for detector update. (ii) LCT [36] and (iii) MUSTER [35] that use a discriminative correlation filter for the short-term component and random ferns and keypoints, respectively, for the detector. (iv) PTAV [37], that uses a correlation filter for the short-term component and a CNN retrieval system [44] for the detector. (v) FCLT [38], that uses a correlation filter for both, the short-term component and the detector. (vi) CMT [32], that uses optical flow for the short-term component and key-points for the detector. (vii) HMMTxD [34], that applies an ensemble of short-term trackers and a keypoint-based detector. (viii) SiamRPN++ [41], that uses a siamese network and a region proposal module for target localization using dept-wise correlation. Target re-detection is performed similarly as localization within an enlarged search region. These trackers further vary in the frequency and approach for model updates (see Table II).

In addition to the selected long-term trackers, we have included a baseline NCC tracker [11] and recent state-of-the art short-term trackers: the standard discriminative correlation

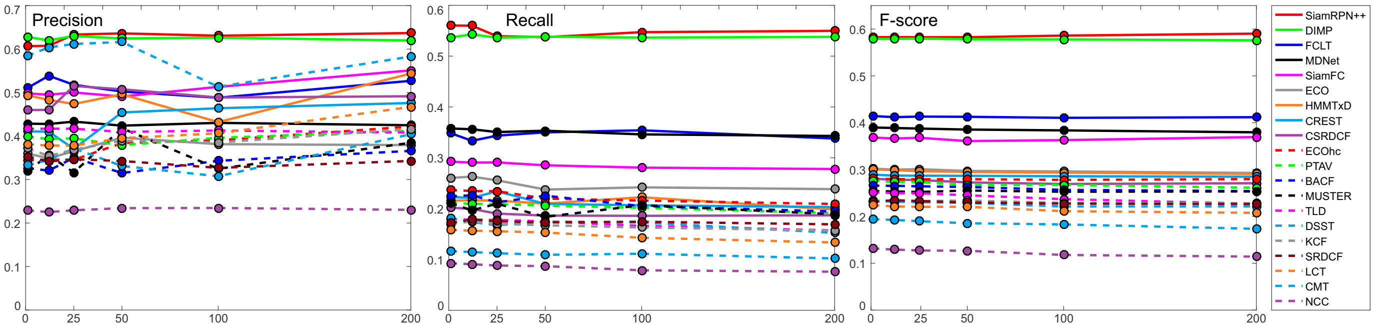

Fig. 4. A simulation with sparsely sampled ground truth – every N-th frame ($N = 1, 12, 25, 50, 100, 200$) is considered in the computation of tracking precision, recall and F-score. The primary measure, the F-score, stays extremely stable, even for a very sparse annotation.

TABLE II

Evaluated trackers are characterized by the short-term component and a confidence score. Long-term trackers are in addition characterized by the detector type and its interaction with the short-term component. Model update and search strategies are indicated. Trackers marked by * were published as LT_1, but did not pass the re-detection test. Results for the re-detection experiment are shown in columns denoted as *Redet. Success* , i.e., the number of sequences with successful re-detection (out of 50), and *Redet. Frames*, i.e., the average number of frames before re-detection. Columns denoted as F, AUC and AUC$_\mathrm{M}$ represent the tracing performance measured by tracking F-score, area under the success curve and modified area under the success curve, respectively. The last column (FPS) represents average tracking speed in frames per second.

Tracker	**S-L**	**Detector**	**Short-term component**	**Interaction Score**	**Update**	**Search**	**Redet. Success**	**Redet. Frames**	**F**	**AUC**	**AUC$_\mathrm{M}$**	**FPS**
SiamRPN++ [41]	LT_0	CNN (siam.)	CNN	Score: max corr.	Never	Search window (enlarged region)	1	42.0	0.58	0.59	0.56	12.6
DIMP [42]	ST_1	-	CF (deep.)	Score: max. corr.	When confident (hard negatives)	Search window (enlarged region)	0	-	0.57	0.55	0.57	29.1
FCLT [38]	LT_1	CF (reg.)	CF (reg.)	Resp. thresh., Score: resp. quality	ST: when confident LT: mix ST + LT	Entire image (correlation + motion)	50	76.8	0.41	0.40	0.40	6.3
MDNet [28]	ST_1	-	CNN	- Score: CNN score	When confident (hard negatives)	Random sampling	0	-	0.39	0.36	0.38	0.9
SiamFC [27]	ST_1	-	CNN	- Score: max. corr.	Never	Search window (enlarged region)	0	-	0.37	0.32	0.36	22.4
ECO [43]	ST_1	-	CF (deep f.)	- Score: max. corr.	Always (clustering)	Search window (enlarged region)	0	-	0.30	0.29	0.31	7.2
HMMTxD [34]	LT_1	Keypoints (static)	Flow + CF + ASMS	HMM Score: # keypoints	ST: when confident LT: when confident	Entire image (keypoint matching)	46	1.7	0.30	0.30	0.28	2.1
CREST [26]	ST_0	-	CNN	- Score: max. corr.	Always (backprop)	Search window (enlarged region)	0	-	0.29	0.26	0.29	0.3
CSRDCF [24]	ST_0	-	CF	- Score: max. corr.	Always (exp. forget.)	Search window (enlarged region)	0	-	0.28	0.23	0.28	7.9
ECOhc [43]	ST_1	-	CF	- Score: max. corr.	Always (clustering)	Search window (enlarged region)	0	-	0.28	0.26	0.29	52.5
PTAV* [37]	LT_0	Siamese network	CF (fDSST)	Conf. thresh, const. verif. interval Score: CNN score	ST: always, LT: never	Search window (enlarged region)	1	35.0	0.28	0.24	0.26	4.5
BACF [25]	ST_0	-	CF	- Score: max. corr.	Always (exp. forget.)	Search window (enlarged region)	1	7.0	0.27	0.26	0.29	15.9
MUSTER [35]	LT_1	Keypoints (SIFT)	CF	F-B, RANSAC Score: max. corr.	ST: every frame LT: when confident	Entire image (keypoint matching)	41	0.0	0.25	0.23	0.27	0.8
TLD [30]	LT_1	Random fern	Flow	P-N learning Score: conser. sim.	Positive, negative samp.	Entire image (cascade)	18	0.0	0.25	0.18	0.26	10.0
DSST [22]	ST_0	-	CF	- Score: max. corr.	Always (exp. forget.)	Search window (enlarged region)	0	-	0.23	0.20	0.24	8.1
KCF [21]	ST_0	-	CF	- Score: max. corr.	Always (exp. forget.)	Search window (enlarged region)	0	-	0.23	0.22	0.24	55.2
SRDCF [23]	ST_0	-	CF	- Score: max. corr.	Always (exp. forget.)	Search window (enlarged region)	0	-	0.23	0.21	0.24	3.1
LCT* [36]	LT_0	Random fern	CF	k-NN, resp. thresh. Score: max. corr.	When confident	Search window (enlarged region)	0	-	0.22	0.19	0.22	16.0
CMT [32]	LT_1	Keypoints (static)	Keypoints (flow)	F-B, clustering, correspondencies Score: # keypoints	ST: always LT: never	Entire image (keypoint matching)	42	0.0	0.20	0.14	0.22	22.3
NCC	ST_1	-	Correlation	- Score: max. corr.	Never	Search window (enlarged region)	0	-	0.13	0.11	0.14	41.3

filters KCF [21] and DSST [22], four recent advanced versions SRDCF [23], CSRDCF [24], BACF [25], ECOhc [43] and the top-performer on the OTB [1] benchmark ECO [43]. Two state-of-the-art CNN-based top-performers from the VOT [45] benchmark SiamFC [27] and MDNet [28] and a state-of-the-art CNN-based tracker CREST [26] were included as well. For a complete analysis we include the most recent short-term tracker DIMP [42], which is currently the top-performing tracker on the majority of the short-term tracking benchmarks. All these short-term trackers were modified to be LT$_0$ compliant, i.e., able to report the target absence. For each tracker, the score reflecting an internal belief of the target presence was used for the target prediction certainty score. For consistency, the same ”type” of score was reported for

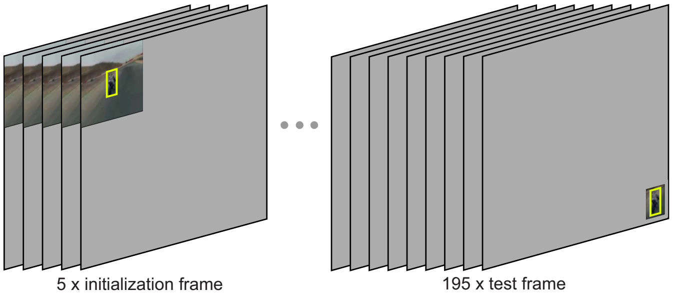

Fig. 5. Re-detection experiment – the artificially created sequence structure by repetition, padding and target displacement. Please see the text for experiment details.

all trackers from the same group of trackers e.g., maximum correlation response for all correlation filter based trackers. All trackers were integrated in the VOT [4] toolkit for automatic evaluation.

B. Re-detection experiment

An experiment was designed to position the tested trackers on the LT/ST spectrum, and in particular to verify their image-wide re-detection capability. Artificial sequences were generated from the initial frame of each sequence in our dataset. The initial frame was placed into the top-left corner of a zero-initialized image, which is three times wider and higher than the original image (Figure 5). The artificial sequences start with five copies of the enlarged frame. For the remainder of the sequence, the target region was cropped from the initial image and copied to the bottom right corner of a zero-initialized frame. A tracker was initialized in the first frame and we measured the number of frames required to re-detect the target after position change.

Results are summarized in Table II (columns *Redet. Success/Frames*). Trackers MDNet, ECO, ECOhc, SRDCF, SiamFC, CREST, CSRDCF, KCF, DSST and NCC never re-detected the target, which confirms their short-term design. The BACF tracker re-detects the target in one sequence by coincidence (random drift) and it is not the result of a re-detection mechanism. The only tracker that always re-detected the target was FCLT, while HMMTxD, MUSTER, CMT and TLD were successful in most sequences – this result classifies them as LT_1 trackers. The difference in detection success come from the different detector design. FCLT and TLD both train template-based detectors. The good performance of FCLT likely comes from the efficient discriminative filter training framework of the FCLT detector. The keypoint-based detectors in HMMTxD, MUSTER and CMT are similarly efficient, but require sufficiently well textured targets. Interestingly, the re-detection is immediate for MUSTER, CMT, HMMTxD and TLD, while FCLT requires on average 77 frames. This difference comes from the dynamic models. MUSTER, CMT, HMMTxD and TLD apply a uniform spatial prior in the dynamic model in the detector phase over the entire image, while the FCLT applies a random walk model that gradually increases the target search range with time.

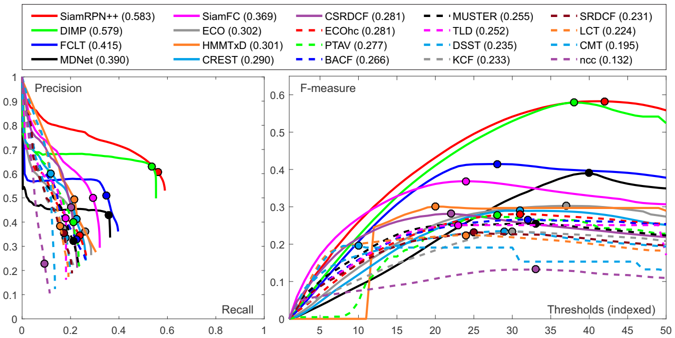

Fig. 6. Long-term tracking performance on the LTB50 dataset. The average tracking precision-recall curves (left), the corresponding F-measure curves (right) – F as a function of prediction certainty linearly rescaled for each tracker; 0 - the minimum over all sequences output by a given tracker, 100 - the maximum. Tracker labels are sorted according to F-scores, i.e., F-measure maxima.

Surprisingly, two recent long-term trackers, LCT and PTAV nearly never successfully detected the target. A detailed inspection of their source code revealed that these trackers do not apply their detector to the whole image, but rather a small neighborhood of the previous target position, which makes these two trackers a pseudo long-term, i.e., LT_0 level. A similar property can be observed for SiamRPN++ tracker, which fails in image-wide target re-detection since the search region is enlarged to a fixed size, smaller than the image, during re-detection.

C. Overall performance

The overall performance on the LTB50 dataset is summarized in Figure 6. Ranked highest are the most recent trackers SiamRPN++ and DIMP (ST_1). SiamRPN++ (LT_0) is a long-term tracker with a deep siamese network as a feature extractor and a region-proposal module. This tracker does not update the visual model to prevent model contamination and compensates by using deep rich feature representation. DIMP is an ST_1 tracker which formulates discriminative filter learning, similar as DCFs, within a framework of CNNs. The tracker applies a large search region, and an improved discriminative training by selective updates and hard negative mining – all of which contribute to stable long-term tracking. Third-ranked tracker is FCLT, an LT_1 class tracker, which uses discriminative correlation filters on hand-crafted features for short-term component as well as detector in the entire image. Three *short-term* ST_1 class CNN-based trackers are following: MDNet, SiamFC and ECO. These implement different mechanisms to deal with occlusion. MDNet applies very conservative updates (similar as DIMP), SiamFC does not update the model at all and ECO applies clustering-based update mechanism to prevent learning from outliers. SiamFC, ECO and MDNet search a fairly large region which is beneficial for target re-detection.

Another LT_1 long-term tracker, HMMTxD, achieves comparable performance to ECO. It uses an ensemble of short-term trackers with weak visual models, and performs image-wide target re-detection. Two long-term trackers CMT (LT_1) and LCT (LT_0) perform the worst among the tested trackers.

The CMT entirely relies on keypoints, which perform poorly on non-textured targets. The relatively poor performance of LCT is likely due to a small search window and poor detector learning. This is supported by the fact that LCT performance is comparable to KCF, a standard correlation filter, also used as the short-term component in LCT. The performance of short-term trackers ST_0 class trackers does not vary significantly.

D. Attribute evaluation

Figure 7 shows tracking performance with respect to nine visual attributes from Section V. Long-term tracking is mostly characterized by performance on *full occlusion* and *out-of-view* attributes, since these require re-detection. Top performance is achieved by SiamRPN++ (LT_0) and DIMP (ST_1) due to a stable discriminative target representation, which includes robust updating in DIMP and no updating in SiamRPN++, and relatively large search regions. The FCLT (LT_1 class) achieves good performance, which is likely due to the efficient learning of the detector component. Another LT_1 tracker, HMMTxD, performs comparably to the best short-term trackers (SiamFC and MDNet), while the CMT, TLD and MUSTER performance is lower due to a poor visual model.

A highly challenging attribute is *fast motion* which is related to long-term re-detection combined with blurring. Top performance is obtained by trackers with a relatively large search range and powerful deep features (SiamRPN++ and DIMP).

Another attribute specific for long-term tracking is *viewpoint change* which includes video cuts and camera hand-overs. Top-performing trackers at this attribute are SiamRPN++ and DIMP, demonstrating that their deep target representation is powerful enough to robustly localize the target even under a different viewing angle. Other well-performing trackers at this attribute are SiamFC and MDNet, which indicates that in most of these viewpoint changes the target did not move significantly in image coordinates, and moderate search range sufficed in target re-detection. The result also shows that the target appearance did change and was well addressed by the deep features.

The *similar objects* attribute exposes fine-grained discrimination capability between the tracked object and visually similar objects in the vicinity. Top performance is achieved by DIMP, which means that the model updating is robust enough to capture slight differences between targets. The second-best tracker, SiamRPN++, on the other hand does not update the model, but rather applies powerful discriminatively trained features. The excellent performance of the third-best tracker, MDNet, can be likely attributed to the use of hard-negative mining in the visual model update and a moderately sized search range. Another well performing tracker is FCLT. In contrast to MDNet, this tracker performs image-wide re-detection, which increases the probability of drifting to another object, even if the object is far away. These false detections are mitigated by the motion model, that gradually increases the effective detection range after localization becomes uncertain.

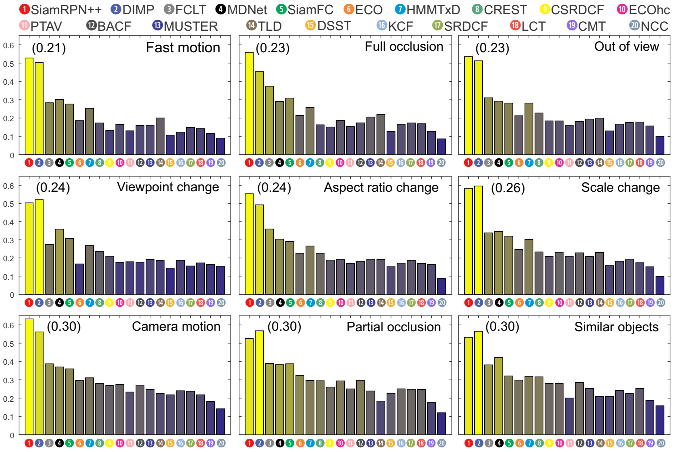

Fig. 7. Average per-tracker F-scores for each visual attribute. Average F-score of each attribute is shown at the top-left corner. The most challenging attributes are fast motion, full occlusion and out-of-view.

E. Influence of disappearance frequency

We divided the sequences of LTB50 into groups according to the number of target disappearances: (Group 1) over ten disappearances, (Group 2) between one and ten disappearances and (Group 3) no disappearances. Per-sequence F-scores are summarized in Figure 8.

Group 1 results: Most short-term trackers, except the recent DIMP, performed poorly due to lack of target re-detection. Long-term trackers generally perform well, but there are differences depending on their structure. For example, the “following” and “liverrun” sequences contain cars, which only moderately change the appearance. SiamFC does not adapt the visual model and is highly successful on these sequences. The LCT generally performs poorly, except for the “yamaha” sequence in which the target leaves and re-enters the view at the same location. Thus the poor performance of LCT is due to a fairly small re-detection range. MDNet, CREST and SiamFC perform moderately well, despite the fact that they are short-term trackers. A likely reason is their highly discriminative visual features (CNNs) and a relatively large target localization range.

Group 2 results: Performance variation comes from a mix of target disappearance and other visual attributes. However, in “person14” the poor performance is related to a long-lasting occlusion at the beginning, where most trackers fail. Only a few long-term trackers (SiamRPN++, FCLT, MUSTER, HMMTxD and TLD) overcome the occlusion and obtain excellent performance.

Group 3 results: The performance of long-term trackers does not significantly differ from short-term trackers since the target is always visible. The strength of the features and learning in visual models play a major role. These sequences are least challenging for all trackers in our benchmark.

VIII. Tracker architecture evaluation

A. Overall architecture analysis

We analyze contributions of architectural choices important for successful long-term tracking by categorizing the tested

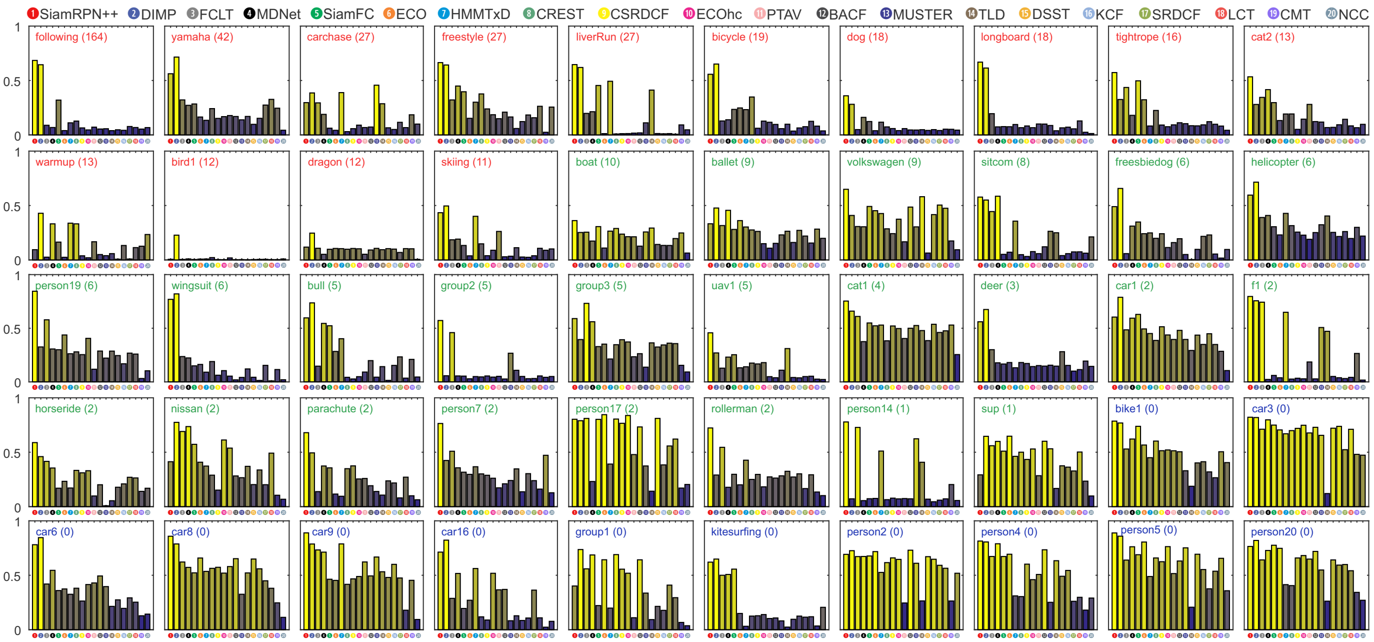

Fig. 8. Target disappearance analysis: The plots show per-sequence F-scores for all trackers. Sequences are sorted, left-to-right, top-to-bottom, by the number of target disappearances, i.e. the largest number at top-left. Red label: (>10) disappearances, green: $(1-10)$ disappearances, blue: (0) disappearances.

trackers along the following four aspects: (i) detector design, (ii) short-term component design, (iii) features used and (iv) visual model adaptation strategy. To aid interpretation, we generate a connection plot Figure 9 where each tracker is connected to the specific choice of the four design aspects thus visualizing a design trend by color-coding.

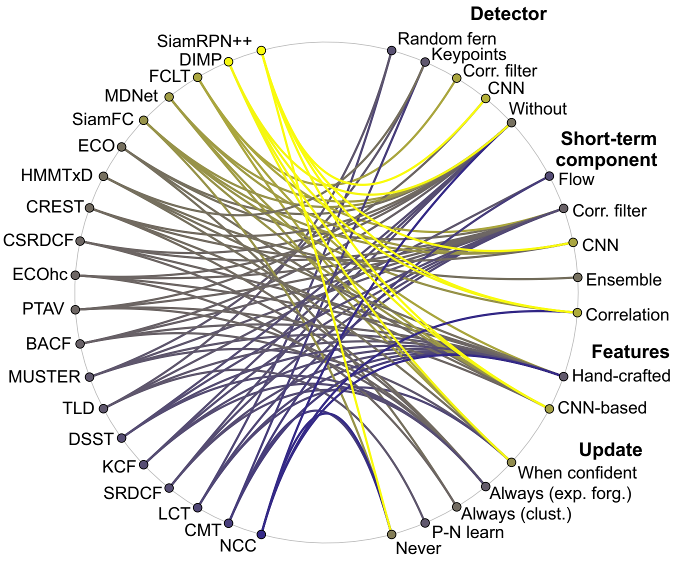

Fig. 9. The connection plot visualizing the tracker architectural choices. Each tracker is connected to at least one architectural choice in each of the four components of a long-term tracker (detector, short-term component, features and update strategy). The connections are color coded by the tracking F-score on LTB50 dataset with yellow indicating high values and blue indicating low values. The architecture of the trackers is analyzed in Section VIII-A.

Detector design: The results show that CNN-based detectors consistently deliver promising performance. Correlation filters are widely used in short-term trackers, but are generally not used for image-wide detection, except for in the FCLT. The plot indicates that deep CNN-based detectors might be a very promising research direction in long-term tracker design, but they require special hardware (GPU). Another promising direction in long-term tracker design are fast DCF-based detectors like the one used in FCLT which does not require a GPU. The quality of keypoint-based detectors varies significantly among the trackers. The benefit lies in potential to re-detect target even under a similarity or affine transform, but a common drawback is the inability to detect small or homogeneous targets.

Short-term component: The most promising design choices for the short-term component follow the trend in state-of-the-art short-term trackers. The connection plot indicates that CNN-based and DCF-based methods are most successful short-term design choices.

Visual features: Visual models with features based on CNN generally achieve improved performance over hand-crafted features. The reason is likely in discriminative capacity of the pre-trained networks. A drawback is that these features typically entail significant computational resources and dedicated hardware (i.e., GPU). On the other hand, robust long-term tracking is also feasible by hand-crafted features combined with a well-designed re-detection strategy or update mechanism (e.g., FCLT).

Adaptation strategy: In long-term tracking scenarios, the target may leave the field of view or become occluded for longer periods. Constant updating irreversibly corrupts the visual model leading to drift and failure and reduces the potential for target re-detection. Conservative updating such as implemented in MDNet, DIMP or FCLT appears to be the best strategy. An extreme conservative update strategy, i.e., no update at all, appears to work as well, but this requires highly expressive features such as localization-trained CNN in SiamFC and SiamRPN++.

B. Importance of re-detection strategy

We further explore the importance of re-detection strategy by the following experiment. All tracker outputs were modified to report a constant uncertainty, which was treated as if the tracker is always reporting the target as present. Tracking Recall was computed, which we denote by Re. Then for each tracker, all overlaps after the first failure (i.e., overlap drops to zero) were set to zero and the tracking Recall was re-computed (Re_0).

Figure 10 shows differences between the two recalls (i.e., $Re - Re_0$). Large values indicate greater failure recovery capabilities of the trackers. SiamRPN++, FCLT, HMMTxD and TLD most often re-detected the target. Surprisingly, the differences in tracking Recalls of three short-term trackers DIMP, MDNet and SiamFC are very large, which indicates that these two trackers indeed posses long-term properties. There are two probable explanations: (i) the trackers posses a large search region which enables target re-detection or (ii) the trackers posses efficient visual model update mechanism that prevents visual model corruption during target loss and they eventually drift back to the target.

Additional analysis of the tracking performance was carried out to determine the reason for the apparent long-term properties. Let o_i denote an overlap at i-th frame of a sequence. We identified the pair of frames where $o_{i-1} = 0$ and $o_i > 0$ which is a point at which the tracker re-detects the target. Euclidean distance was computed between the predicted bounding box centers in these frames. We expect that large distances indicate large target search ranges and compute the mean value of the ten percent of largest Euclidean distances as an indicator of the recorded search range size.

The results are shown in Figure 10. The search range size of MDNet and SiamFC are comparable to other short-term trackers. DIMP has a bit larger search range, but still significantly smaller than the search range of long-term trackers. This means that the key factor for their excellent long-term tracking performance compared to the other short-term trackers is the visual model. Both of these trackers use pre-trained CNN-based features. DIMP and MDNet updates the visual model only on frames where tracking is considered reliable while SiamFC does not update the visual model at all. Both mechanisms prevent training from incorrect examples, which enables eventual re-detection once the target gets close to the current tracker prediction, even though the search range is not the whole image. Long-term trackers FCLT, TLD, CMT and HMMTxD have the largest search range, which confirms the image-wide target re-detection ability tested in Section VII-B. SiamRPN++, MUSTER and PTAV have a moderately large search region, while LCT has the smallest search range among all long-term trackers.

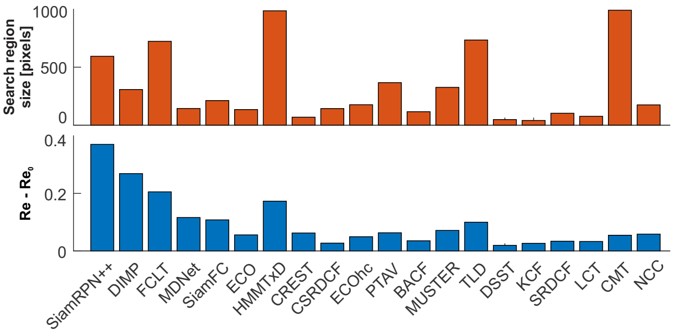

Fig. 10. The red graph shows results of the re-detection search range size experiment (above). Higher values indicate a larger search range. The blue graph (below) shows the difference in tracking Recall between the original tracking result and the one with all overlaps set to zero after the first failure. Large values indicate increased influence of the re-detection.

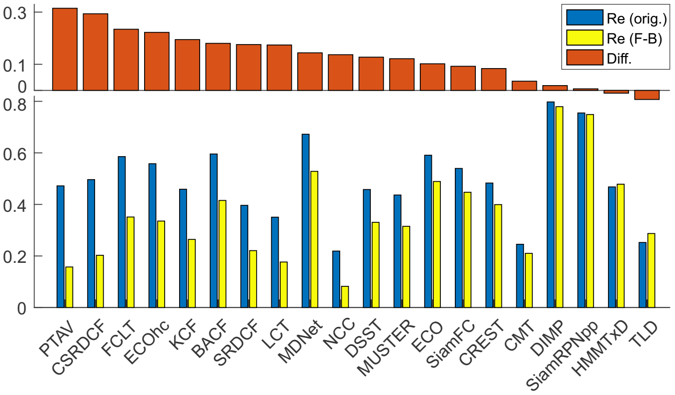

Fig. 11. The bottom plot shows tracking recall of trackers evaluated on sequences from LTB50 with target always visible (blue) and the same measure when running the trackers on the same sequences forward/backward five times (yellow). The differences between these recalls indicate tracker sensitivity to long sequences due to error accumulation in the visual model and are shown in the upper plot (red). Larger difference represents larger impact of the sequence length. Note that the error increases for all trackers except from TLD, that learns from experience and actually decreases the error.

C. Impact of visual model error accumulation

Visual model update strategy plays a central role in drift prevention, which is crucial in tracking over long periods. We designed the following experiment to evaluate drifting, which is not caused by target disappearance. Based on the disappearance frequency analysis from Section VII-E, we selected eleven long sequences from LTB50 in which the target never disappears and extended them by looping forward and backward five times. This set of extended sequences thus contains 302,330 frames with an average sequence length of 27,485 frames.

The trackers were re-run on this dataset and the tracking Recall was computed with the same tracker output modification to a constant value as in Section VIII-B. Since the target never leaves the field of view and the tracker always reports the target, the tracking Recall is equivalent in this case to the average overlap on all frames (see Section IV).

The results are shown in Figure 11. The highest tracking recall is achieved by DIMP (ST$_1$), SiamRPN++ (LT$_0$), two ST$_1$ short-term trackers MDNet, ECO, an ST$_0$ tracker BACF and an LT$_1$ long-term tracker FCLT. Top positions are not dominated by long-term trackers in part because the target is always present and false activations of the detector during uncertain target localization periods may lead to tracker jumping to a location away from the target, which reduces performance.

Performance drop is smallest for TLD, HMMTxD, CMT, CREST and SiamFC. The CMT fails early on in the original sequences, which explains the apparently small performance drop. CREST applies end-to-end updating of all parameters in a CNN with small learning rate, which leads to robust tracking in situations when the target is always visible. The small performance drop in SiamFC is likely due to the fact that this tracker does not update the visual model. Combined with the deep features, this proves as a robust strategy to reduce visual model contamination and leads to a successful tracking. Interestingly, the recall actually increases for TLD and HMMTxD as the sequences are looped, which is consistent with the observations in the original paper [30]. TLD applies P-N learning, a conservative form of learning that retrospectively expands the visual model with new training examples. The longer the target is observed, the stronger the visual model becomes. Similarly, the HMMTxD uses combination of feature-based detector, which is trained only in the initial frame and set to high precision mode, that guides the on-line learning of the hidden Markov model. The HMM encodes the relationship of the performance of individual trackers and their confidences using Baum-Welch algorithm. This combination enables choosing, on-the-fly, which tracker should be used in every frame and improves over time. The selective update strategies from MDNet and LCT, which mix short-term and long-term updates also appear beneficial – MDNet, for example, actually keeps track of appearance samples from a longer time-scale and uses these in combination with a local hard negative mining in the model update.

The largest performance drop is observed for the long-term trackers PTAV and FCLT and the short-term trackers CSRDCF and ECOhc. There are several reasons for these performance drops. All four trackers use a DSST [22] scale estimation method. We observed that the scale at these trackers gradually drifts in extremely long sequences. The reason might be that DSST scale estimation relies very much on the target localization accuracy. Inaccurate localization leads to incorrect scale estimation, and gradual error accumulation from constant scale updates further reduces the localization accuracy, leading to drift.

The long-term FCLT and PTAV are affected by false activations of their detector, which in some cases leads to tracker jumping off the target. FCLT updates the detectors over several scales only during certain tracking periods and eventually redetects the target in most cases. PTAV applies a CNN instance-based object detector without updates. The strength of this detector is that it generalizes well enough to detect the target even under deformation. On the other hand, this generalization leads to failure when similar objects are located in the target vicinity. This is an obvious reason in the *bike1* sequence where the detector jumps to another bicyclist.

There is a significant difference in performance drops of ECO and ECOhc, even though they both use the same visual model decontamination strategy during updates. Part of the difference can be attributed to different scale estimation strategies these two trackers use. ECO applies the tracker over several scales, while ECOhc applies the DSST on the estimated target position. Another significant difference is that ECO applies deep features, while ECOhc uses only HOG and Colornames. Therefore the longer tracking periods observed in ECO might be likely due to deep features and greedy scale search.

IX. Discussion and conclusion

A new long-term single-object tracking benchmark was presented. A new short-term/long-term tracking taxonomy that predicts performance on sequences with long-term properties was introduced. The taxonomy considers (i) the target absence prediction capability, (ii) the target re-detection strategy and (iii) the visual model update mechanism.

A new long-term tracking performance evaluation methodology which introduces new performance measures – *tracking* Precision, Recall and F-score – is proposed. These measures extend the detection analysis capabilities to tracking in a principled way and theoretically link the short-term and long-term tracking problem. Even though the proposed measures have been proposed for single-target tracking, multi-target tracking might benefit from their probing power as well. However, we leave such extensions for future work.

A new dataset (LTB50) of carefully selected sequences is constructed, with a significant number of target disappearances per sequence to emphasize long-term tracking properties. Our experiments in Section VII-E indicate that target disappearance is in fact the most challenging aspect of long-term tracking. The diversity of the dataset has been ensured by including a number of target examples typical for long-term tracking in a variety of environments. Sequences are annotated with nine visual attributes which enable in-depth analysis of trackers. Seven long-term trackers and eleven state-of-the-art short-term trackers were categorized using the new taxonomy and analyzed using the proposed methodology and the dataset.

Comparison with existing performance measures using theoretical trackers (Section VI-A) shows that the proposed tracking Precision, Recall and F-score outperform existing measures, distinguish well between different tracking behaviors and facilitate its interpretation. Furthermore, these measures are highly robust (Section VI-B) allowing detailed analysis with only every 50th frame annotated. The overall ranking based on the primary measure is even more robust allowing even sparser annotations (e.g., every 200th frame).

The evaluation and analysis covers a comprehensive collection of long-term trackers. According to the overall analysis (Section VII-C), the best performance is obtained by a LT_0 long-term tracker SiamRPN++ [41], which uses a siamese deep network with a region proposal module for target localization. Interestingly, SiamRPN++ does not update the visual model at all, which indicates that pre-trained features are powerful enough to cope to some extent with the long-term target appearance variation. On the other hand, the second-ranked tracker DIMP [42] (ST_1 level) does apply updating by optimizing the localization filter within a deep CNN framework, but applies robust update mechanism to prevent long-term contamination of the visual model. The third rank is obtained by a LT_1 long-term tracker FCLT [38]. This tracker applies discriminative correlation filter as the short-term component as well as for detector. It applies updating of the visual models at various temporal scales and uses the correlation output for predicting target absence. Similar as DIMP is MDNet [28], which is a CNN-based tracker trained for the tracking task. It applies hard-negative mining and conservative updating of a few top-layer CNN features. Attribute analysis (Section VII-D) indicates that full occlusions and out-of-view disappearances are among the most challenging attributes, followed by similar objects and viewpoint change. The analysis also shows that the LTB50 dataset is challenging, the best tracker achieves the average F-score of 0.58, leaving room for improvement.

Further insights are obtained by analyzing architecture designs of the long-term trackers (Section VIII). CNN-based detectors consistently deliver improved performance, which is likely due to their expressive power of robustly localizing the target even under moderate appearance changes. However, appearance generalization may come at a cost when visually similar objects are located in the same scene. In these cases the CNN features may not distinguish between the different objects, leading to tracking the wrong target (Section VIII-C). Even though discriminative correlation filters are not widely used for detectors, results show that careful learning e.g., [38], [42] makes them an excellent choice due to speed and robustness. We expect to see many long-term trackers adapt these in future. Keypoint-based detectors can potentially detect the target under similarity transform e.g., [35], [34], but require textured targets and sufficient resolution, which makes them brittle in practice.

The re-detection experiments from Section VII-B and Section VIII-B show that most successful re-detection strategies are those used in FCLT [38], HMMTxD [34] and TLD [30]. Results in Section VIII-B also show that re-detection quality largely depends on the visual model update strategy. Conservative updates [27] and hard-negative mining [28] show promise. These techniques are crucial for tracking on very long sequences even if the target is always visible (Section VIII-C), since they largely reduce the tracking drift. This finding opens an opportunity for improving long-term tracking by considering best practices in visual features and model updating from short-term trackers.

Scale estimation methods play an important role in tracking drift. A popular approach is to first localize the target and then estimate the scale e.g., by [22], considering only a single position. Trackers with this technique typically fare much worse than those that greedily localize the target on several scales. The reason is that inaccurate localization leads to poor scale estimation, which consequently leads to poorer localization. On long sequences, the errors accumulate in the visual model, resulting in drift.

All tested trackers and performance evaluation methodology have been integrated in the VOT toolkit [4] and will be made publicly available to the research community. We believe that this, along with the evaluation methodology and detailed analysis presented in this paper, will significantly impact the field of long-term tracking from the point of dataset construction with extremely long sequences, performance analysis protocols as well as long-term tracker designs.

Acknowledgment

This research was in part funded by Slovenian Research Agency (ARRS) projects J2-8175, J2-9433 and Z2-1866 and research program P2-0214. A. Lukežič was financed by the Young researcher program of the ARRS. J. Matas was supported by the Czech Science Foundation grant GA18-05360S.

Appendix

Performance measures derivation details

This section provides a detailed derivation of the primary performance measures, tracking Precision and tracking Recall, introduced in Section IV of the main paper. The derivation closely follows the proof from [14], which states that an area-under-the-curve measure, AUC [1], equals the average overlap between the per-frame bounding boxes reported by the tracker and those from the ground truth.

We start with deriving the tracking precision, which is defined as

$$Pr(\tau_\theta, \tau_\Omega) = |\{t : \Omega(A_t(\tau_\theta), G_t) \geq \tau_\Omega \wedge A_t(\tau_\theta) \neq \emptyset\}|/N_p,$$

where $|\cdot|$ is the set cardinality, and N_p is the number of frames with existing prediction, i.e. $A_t(\tau_\theta) \neq \emptyset$. To maintain the same notation as in [14], we define $\phi_t = \Omega(A_t(\tau_\theta), G_t)$ and as a τ_θ is fixed and will be omitted for the purpose of derivation. This means that we have a set of per-frame bounding box overlaps for the subset of frames where $A_t(\tau_\theta) \neq \emptyset$, in total N_p frames. As in [14], we assume that the per-frame overlaps are *ordered* by size in an ascending order and $\phi_0 = 0$, i.e.

$$0 = \phi_0 \leq \phi_1 \leq \cdots \leq \phi_{N_p}.$$

Let $P(\tau_\Omega) = |\{j : \phi_j \geq \tau_\Omega\}|$ be the number of overlaps greater than τ_Ω. A central part of the derivation is showing that averaging over a particular set of per-frame overlaps equals integration over $\frac{P(\tau_\Omega)}{N_p}$, i.e.,

$$\frac{1}{N_p}\sum_{i=1}^{N_p} \phi_i = \frac{1}{N}\int_0^1 P(\tau_\Omega) d\tau_\Omega.$$

Proof: Function P is a step function, i.e., is constant on the interval between ϕ_i and ϕ_{i+1}). Therefore its integral I is

$$I = \sum_{i=0}^{N_p-1} P(\phi_i)(\phi_{i+1} - \phi_i).$$

The sum can be reorganized in the following way:

$$\begin{aligned} I &= P(\phi_0)(\phi_1 - \phi_0) + P(\phi_1)(\phi_2 - \phi_1) + P(\phi_2)(\phi_3 - \phi_2) + \dots \\ &= \phi_1 P(\phi_0) - \phi_0 P(\phi_0) + \phi_2 P(\phi_2) - \phi_1 P(\phi_1) + \phi_3 P(\phi_3) - \dots \\ &= -\phi_0 P(\phi_0) + \phi_1 (P(\phi_0) - P(\phi_1)) + \phi_2 (P(\phi_1) - P(\phi_2)) \cdots \\ &= 0 \cdot P(\phi_0) + \phi_1 \cdot 1 + \phi_2 \cdot 1 + \cdots \qquad (5) \\ &= \phi_0 + \phi_1 + \phi_2 + \cdots \qquad (6) \\ &= \sum_{i=1}^{N} \phi_i. \end{aligned}$$

In (5) we have assumed that the shift between the two consequential values of $P(\tau_\Omega)$, i.e. $P(\phi_i) - P(\phi_{i+1})$ equals to 1, that is true if all ϕ_i are different. If k consequential ϕ_i are equal then the corresponding $k-1$ shifts are 0, while the last one is k. However, in (5) we add $(\phi_i \cdot 1)$ k times. This completes the proof. ■

Derivation of the tracking recall follows the same procedure, the only difference is that the subset of frames is defined by the groundtruth, i.e. frames where $G_t \neq \emptyset$.

Tracking Speed Analysis

Tracking speed is a decisive factor in many applications. We provide a detailed analysis by three measures[4]: (i) initialization time, (ii) maximum per-frame time and (iii) average per-frame time. The initialization time is computed as the initial frame processing time averaged over all sequences. The maximum per-frame time is computed as the median of the slowest 10% of the frames averaged over all sequences. We also measure average speed by averaging over all frames in the dataset. All measurements are in milliseconds per frame (mpf). The experiments were carried out at a standard desktop computer with 3.4GHz 6700-i7 CPU, 16GB of RAM and NVidia GTX 1060 GPU with 6GB of RAM.

The tracking speeds are reported in Figure 12 with trackers categorized into three groups according to the average speed: fast (> 15fps), moderately fast (1fps-15fps) and slow (< 1fps). The fastest tracker is KCF due to efficient model learning and localization by fast Fourier transform. The slowest methods are CNN-based MDNet and CREST due to the time-consuming model adaptation and MUSTER due to slow keypoint extraction in the detection phase. Several trackers exhibit a very high initialization time (in order of several thousand mpf). The delay comes from loading CNNs (SiamFC, ECO, PTAV, MDNet, CREST) or pre-calculating visual models (ECOhc, CMT, TLD, SRDCF, DSST).

Ideally, the tracking speed would be approximately constant over all frames guaranteeing completion within a fixed time delay. Small differences between the maximum per-frame and average time indicate stability. This difference is the largest for the following trackers: ECOhc and ECO (due to a time-consuming update every five frames), FCLT (due to re-detection on the entire image, which is moderately slow for large images), PTAV (due to the slow CNN-based detector) and MDNet (due to the slow update during reliable tracking period).

Comparison of performance measure rankings

To gain further insights into the proposed primary measure, i.e., tracking F-score (Section IV in the paper), we compare it with the existing measures – area under success rate curve (AUC [1]) and to the modified AUC (AUC$_{mod}$ [9], [10]).

Since area under the curve is based on the overlap measure, the AUC is defined only on frames where target is visible (overlap between a predicted region and an empty region is not defined). The modified AUC extends the basic measure so that it re-defines the definition of overlap in the following way: when target is not visible the overlap is 1 if a tracker predicts that target is not visible and the overlap is zero if tracker makes a prediction of the position.

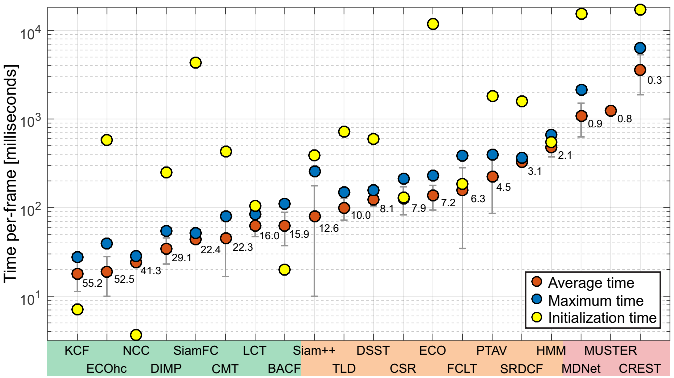

Fig. 12. Tracking speed. Trackers are grouped into three classes: fast (green), moderately fast (orange) and slow (red). All measurements are in milliseconds-per-frame (mpf) – initialization (yellow), average (red), maximum (blue). Average speed measured as frames-per-second is shown next to the position of average mpf. Note that logarithmic scale is used on the y-axis.

Figure 13 shows the values of the different performance measures computed for all trackers tested in the paper. The results show that the AUC measure punishes trackers like TLD and CSRDCF, which are ranked higher according to the F-score. The reason is that AUC does not measure ability to predict target absence. Observe that the performance of most of the short-term trackers increases under the AUC$_{mod}$, compared to AUC, while the long-term trackers with a strong re-detection capability (SiamRPN++, FCLT, HMMTxD) have lower AUC$_{mod}$ than AUC. This result supports our observation that AUC$_{mod}$ is not tailored well for exposing all long-term tracking capabilities, as it favours trackers which are more successful in predicting target absence than re-detecting the target and tracking longer.

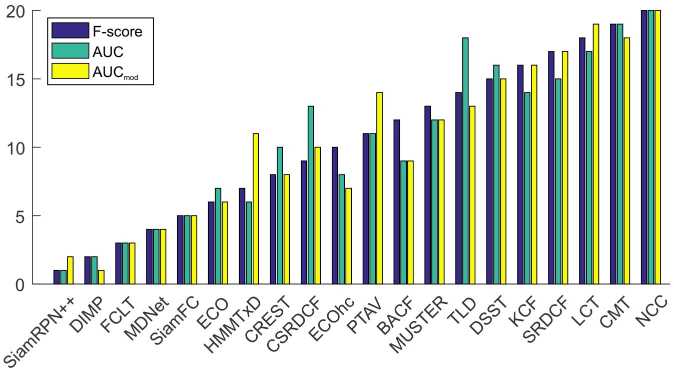

Fig. 13. Comparison of the proposed tracking F-measure with the existing area under the success curve (AUC) and modified AUC (AUC$_{mod}$).The y-axis shows trackers ranks according to the corresponding measures.

[4] Due to the limitations of the source code of MUSTER provided by the authors we were able to calculate the average speed, but not initialization and maximum per-frame times.

References

[1] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Comp. Vis. Patt. Recognition*, 2013, pp. 2411– 2418.

[2] A. Smeulders, D. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1442–1468, July 2014.

[3] P. Liang, E. Blasch, and H. Ling, "Encoding color information for visual tracking: Algorithms and benchmark," *IEEE Trans. Image Proc.*, vol. 24, no. 12, pp. 5630–5644, Dec 2015.

[4] M. Kristan, J. Matas, A. Leonardis, T. Vojir, R. Pflugfelder, G. Fernandez, G. Nebehay, F. Porikli, and L. Cehovin, "A novel performance evaluation methodology for single-target trackers," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016.

[5] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "MOTChallenge 2015: Towards a benchmark for multi-target tracking," *arXiv:1504.01942 [cs]*, Apr. 2015, arXiv: 1504.01942. [Online]. Available: http://arxiv.org/abs/1504.01942

[6] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Cehovin Zajc, T. Vojir, G. Hager, A. Lukezic, A. Eldesokey, and G. Fernandez, "The visual object tracking vot2017 challenge results," in *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[7] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sept 2015.

[8] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for uav tracking," in *Proc. European Conf. Computer Vision*, 2016, pp. 445–461.

[9] A. Moudgil and V. Gandhi, "Long-term visual object tracking benchmark," in *Proc. Asian Conf. Computer Vision*, 2018, pp. 629–645.

[10] R. Tao, E. Gavves, and A. W. Smeulders, "Tracking for half an hour," *arXiv preprint arXiv:1711.10217*, 2017.

[11] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, F. Porikli, L. Čehovin, G. Nebehay, G. Fernandez, and T. e. a. Vojir, "The visual object tracking vot2013 challenge results," in *Vis. Obj. Track. Challenge VOT2013, In conjunction with ICCV2013*, Dec 2013, pp. 98–111.

[12] H. Kiani Galoogahi, A. Fagg, C. Huang, D. Ramanan, and S. Lucey, "Need for speed: A benchmark for higher frame rate object tracking," in *Int. Conf. Computer Vision*, 2017.

[13] L. Cehovin Zajc, A. Lukezic, A. Leonardis, and M. Kristan, "Beyond standard benchmarks: Parameterizing performance evaluation in visual object tracking," in *Int. Conf. Computer Vision*, 2017.

[14] L. Čehovin, A. Leonardis, and M. Kristan, "Visual object tracking performance measures revisited," *IEEE Trans. Image Proc.*, vol. 25, no. 3, pp. 1261–1274, 2016.

[15] F. Yin, D. Makris, and S. Velastin, "Performance evaluation of object tracking algorithms," in *10th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS2007)*, 2007.

[16] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," *arXiv:1603.00831 [cs]*, 2016, arXiv: 1603.00831.

[17] R. Stiefelhagen, K. Bernardin, R. Bowers, J. Garofolo, D. Mostefa, and P. Soundararajan, "The clear 2006 evaluation," in *Multimodal Technologies for Perception of Humans*, 2007, pp. 1–44.

[18] B. Wu and R. Nevatia, "Tracking of multiple, partially occluded humans based on static body part detection," in *Comp. Vis. Patt. Recognition*, vol. 1, 2006, pp. 951–958.

[19] M. Müller, A. Bibi, S. Giancola, S. Al-Subaihi, and B. Ghanem, "Trackingnet: A large-scale dataset and benchmark for object tracking in the wild," *CoRR*, vol. abs/1803.10794, 2018.

[20] J. Valmadre, L. Bertinetto, J. F. Henriques, R. Tao, A. Vedaldi, A. W. Smeulders, P. H. Torr, and E. Gavves, "Long-term tracking in the wild: a benchmark," in *Proc. European Conf. Computer Vision*, September 2018.

[21] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, 2015.

[22] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Discriminative scale space tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1561–1575, 2017.

[23] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Int. Conf. Computer Vision*, 2015, pp. 4310–4318.

[24] A. Lukežič, T. Vojíř, L. Čehovin Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," in *Comp. Vis. Patt. Recognition*, 2017, pp. 6309–6318.

[25] H. Kiani Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Int. Conf. Computer Vision*, no. 1135–1143, 2017.

[26] Y. Song, C. Ma, L. Gong, J. Zhang, R. W. H. Lau, and M.-H. Yang, "Crest: Convolutional residual learning for visual tracking," in *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[27] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," 2016.

[28] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Comp. Vis. Patt. Recognition*, June 2016, pp. 4293–4302.

[29] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *IEEE International Conference on Computer Vision Workshop (ICCVW)*, Dec 2015, pp. 621–629.

[30] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, July 2012.

[31] F. Pernici and A. Del Bimbo, "Object tracking by oversampling local features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 12, pp. 2538–2551, 2013.

[32] G. Nebehay and R. Pflugfelder, "Clustering of static-adaptive correspondences for deformable object tracking," in *Comp. Vis. Patt. Recognition*, 2015, pp. 2784–2791.

[33] 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.

[34] T. Vojir, J. Matas, and J. Noskova, "Online adaptive hidden markov model for multi-tracker fusion," *Comp. Vis. Image Understanding*, vol. 153, pp. 109 – 119, 2016.

[35] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking," in *Comp. Vis. Patt. Recognition*, June 2015, pp. 749–758.

[36] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-term correlation tracking," in *Comp. Vis. Patt. Recognition*, 2015, pp. 5388–5396.

[37] H. Fan and H. Ling, "Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking," in *Int. Conf. Computer Vision*, 2017, pp. 5486–5494.

[38] A. Lukezic, L. C. Zajc, T. Vojír, J. Matas, and M. Kristan, "FCLT - A fully-correlational long-term tracker," *CoRR*, vol. abs/1711.09594, 2017.

[39] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[40] L. Huang, X. Zhao, and K. Huang, "Got-10k: A large high-diversity benchmark for generic object tracking in the wild," *arXiv preprint arXiv:1810.11981*, 2018.

[41] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," in *Comp. Vis. Patt. Recognition*, 2019.

[42] G. Bhat, M. Danelljan, L. Van Gool, and R. Timofte, "Learning discriminative model prediction for tracking," in *Int. Conf. Computer Vision*, 2019.

[43] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg, "Eco: Efficient convolution operators for tracking," in *Comp. Vis. Patt. Recognition*, 2017, pp. 6638–6646.

[44] R. Tao, E. Gavves, and A. W. M. Smeulders, "Siamese instance search for tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[45] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Čehovin, T. Vojir, G. Häger, A. Lukežič, and G. et al. Fernandez, "The visual object tracking vot2016 challenge results," in *Proc. European Conf. Computer Vision*, 2016.

Alan Lukežič received the Dipl.ing. and M.Sc. degrees at the Faculty of Computer and Information Science, University of Ljubljana, Slovenia in 2012 and 2015, respectively. He is currently a PhD student at Faculty of Computer and Information Science, University of Ljubljana, Slovenia. His research interests include computer vision, data mining and machine learning.

Luka Čehovin Zajc received his Ph.D from the Faculty of Computer and Information Science, University of Ljubljana, Slovenia in 2015. Currently he is working at the Visual Cognitive Systems Laboratory, Faculty of Computer and Information Science, University of Ljubljana, Slovenia as an assistant professor and a researcher. His research interests include computer vision, HCI, distributed intelligence and web-mobile technologies.

Tomáš Vojíř received his Ph.D from the Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic in 2018. Currently he is working at the Machine Intelligence Laboratory, Department of Engineering, University of Cambridge, United Kingdom as a research associate. His research interests include real-time visual object tracking, object detection, online learning and vision for autonomous driving.

Jiří Matas is a professor at the Center for Machine Perception, Czech Technical University in Prague. He holds a PhD degree from the University of Surrey, UK (1995). He has published more than 200 papers in refereed journals and conferences. His publications have approximately 34000 citations in Google Scholar and 13000 in the Web of Science. His h-index is 65 (Google scholar) and 43 (Clarivate Analytics Web of Science) respectively. He received the best paper prize at the British Machine Vision Conferences in 2002 and 2005, at the Asian Conference on Computer Vision in 2007 and at Int. Conf. on Document analysis and Recognition in 2015. He is on the editorial board of IJCV and was an Associate Editor-in-Chief of IEEE T. PAMI. His research interests include visual tracking, object recognition, image matching and retrieval, sequential pattern recognition, and RANSAC- type optimization methods.

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.