ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10446305

RANKING OF VISUAL TRACKERS USING ROBUST ERROR NORMS

Julien Valognes and Maria Amer

Department of Electrical and Computer Engineering, Concordia University, Montréal, Québec, Canada.

ABSTRACT

Object trackers are typically ranked by the average of averages, that is, a performance measure averaged over all frames of a video and then averaged over the entire dataset. The average is not a robust estimator. We propose to rank trackers based on robust error norms: we divide the performances of a set of trackers for a video, sorted from best to worst, into outliers (edge trackers) and inliers (trackers with similar performances); we propose an edge-stopping function that assigns the highest score to the highest-performance (top) tracker and scores other trackers accordingly. Our edge-stopping function stops at edge trackers (outliers) using a robust scale defined using the difference (error) between the performances of the top tracker and neighboring trackers. Our method is not a new performance measure but an approach to rank trackers robustly and systematically. We test our methods using five video datasets and 20 trackers. We show that the proposed score is more robust and representative of a tracker's performance than the widely-used average of averages.

Index Terms— Object tracking, performance evaluation, scoring, edge-stopping, robust error-norm, robust scale.

1. INTRODUCTION

Visual tracking is a highly-active research area constantly introducing numerous techniques (a recent survey is [1]). The challenge is not only robust modeling of appearance changes of the target but also robust evaluation (ranking) of the trackers to determine the best-performing ones systematically. A ranking method ranks a set of T trackers $\{t_i, i = 1, \cdots, T\}$ based on their performance data q_{il} (for example, average overlap ratio AOR) on a set of L video sequences $\{v_l, l = 1, \cdots, L\}$ of a dataset. Few tracker ranking approaches exist in the literature. Typically, trackers are ranked by computing the mean α_i of q_{il} over all frames of a sequence v_l and then over all sequences of the dataset; α_i is hence the average of averages. The mean, however, can be significantly affected by outliers or when the data distribution is not symmetric [4]. We examined the distribution of the performance data $\{q_{il}\}$ of trackers and noticed the distribution is not symmetric and heavily skewed to the right (see Figure 1).

With the average of averages, it is difficult to argue which is the best tracker: the one that performs well in certain sequences but poorly in others, the tracker that fluctuates least over sequences, or the tracker that performs consistently well but slightly less on average. Consider, for example, the following AOR values of some trackers: {0.80732 0.83085 0.83823 0.83908 0.85437 **0.85704** 0.87422 0.88037 0.8813 **0.88971** 0.8904 0.89125 0.89513 0.89612 0.89732 0.90102}; we can divide these values into performance regions (e.g., the blue values are one region). But how to robustly find the edges (i.e., outliers or regions' boundaries, e.g., those marked bold)? How, then, to rank the trackers: is 0.90102 better than 0.89732, or do they have the same rank?

Our contribution is a new approach using robust statistics concepts to systematically and robustly rank trackers' performance. Specifically, we define outliers (edge trackers) and a robust scale by observing that trackers' performances can be divided into uniform *regions of performance*. We then propose an *edge-stopping* function to score trackers based on the robust scale to distinguish outliers (i.e., edges or region boundaries) from inliers (i.e., similarly-performing trackers); the function assigns the highest score to the best-performing tracker and scores other trackers relative to that top.

Related Work: In visual tracking benchmarks [3, 5, 2, 6], the common tracker ranking method is the average of averages (mean). Based on this mean, the trackers under test are then numerically ranked from best to worst. Different than the benchmarks mentioned above [3, 5, 2, 6], our tracker ranking approach scores similarly-performing trackers based on a robust scale to distinguish outliers from inliers for each sequence. Few ranking methods exist: The tracker ranking method [7] applies four ranking methods and average them into a mean rank; the first two methods model datasets as graphs and assign ranks using both an aggregation algorithm and a PageRank-based solution [8], while the two last methods are derived from the Elo [9] and Glicko [10] sports rating systems. The authors identify trackers as best or second best, where any tracker not qualified as best is automatically assumed to be second best. The tracker ranking method [11] assigns a rank to a tracker by quantifying how much its performance deviates from the highest performance over all sequences using the median absolute deviation. The final score is obtained by weighting the best and second-best scores over all sequences. Unlike these related ranking methods [7, 11], our approach assigns scores systematically across all trackers (not only best and second best); these scores are well spread across the range $[0, 1]$, depending on tracker performances.

2. PROPOSED RANKING APPROACH

2.1. Edge-stopping for Scoring of Trackers

Two trackers, t_1, t_2, are *neighbors* if the difference (error) between their performances q_{1l}, q_{2l} for the sequence v_l is not an outlier, that is, the error remains within a robust scale σ_l. This can be interpreted as detecting the boundaries (edges) between the piece-wise uniform performance regions.

The input to our scoring method are the performance data $\{q_{il}\}$ for all tested trackers $\{t_i, i = 1, \cdots, T\}$ over all videos $\{v_l, l = 1, \cdots, L\}$ of a dataset. Let q_{bl} be the performance of the best-performing tracker t_b over a v_l among all $\{t_i\}$. We define e_{il} as the difference (error) in performances between the best-performing tracker t_b and a tracker t_i in sequence v_l

$$e_{il} = q_{bl} - q_{il}. \tag{1}$$

When the metric is the higher, the better, such as AOR, $q_{bl} = \max\{\text{AOR}_i\}$ and when it is the lower, the better, such as the failure rate FR, $q_{bl} = \min\{\text{FR}_i\}$. Obviously, for FR-like metrics, (1) becomes $e_{il} = q_{il} - q_{bl}$.

The set $\{e_{il}\}$ contains a population of samples, where the difference between the best tracker t_b and its similarly-performing (neighboring) trackers t_i is small. (Compared to image denoising, t_b acts as the central pixel and t_i as neighboring pixels.) The errors of neighboring trackers are from one (uniform) distribution, while far-neighbors' errors are from another distribution. Figure 1(a) shows that the histogram of the AOR errors $\{e_{il}\}$ can be approximated uniform. Figure 1(b) shows that the histogram of AOR $\{q_{il}\}$ is not symmetric and heavily skewed to the right.

Fig. 1: Histograms of AOR $\{e_{il}\}$ and $\{q_{il}\}$ under the combined dataset OTB+VOT-ST+NfS (see the Results Section).

We seek a scoring (edge-stopping) function $h(e_{il})$ that assigns a high score to a tracker t_i when the error e_{il} is small and outputs a lower score the larger the error becomes, depending on a robust scale σ_l per a sequence v_l. We aim to reject outliers, i.e., far-neighbors of t_b, depending on

$$\sigma_l = c \cdot \underset{1 \le i \le T}{\text{MAD}}\{e_{il}\}, \tag{2}$$

where σ_l is the robust scale of all errors e_{il} at sequence l, MAD is the median absolute deviation, and c is a scale factor (robust point estimate), which depends on the distribution family. Since we assume the $\{e_{il}\}$ are uniformly distributed, we select $c = \sqrt{4/3} = 1.1547$ (see [12] for derivation of c).

We examined the edge-stopping functions [13]: Lorentzian, Huber minmax, and Tukey biweight. Tukey biweight descends all the way to zero, ignoring the contributions of the worst-performing trackers. Huber minmax is constant for small errors and can, therefore, assign the same highest score to trackers similar to the best-performing tracker for a sequence. However, its scoring can get unstable and produce too many inliers (i.e., neighbors) when the scale σ_l is high. The Lorenztian edge-stopping delivers a good balance in-between Huber minmax and Tukey biweight. Thus, at a sequence l, we propose to assign a tracker t_i a score s_{il}

$$s_{il} = \frac{1}{1 + \frac{e_{il}^2}{2\sigma_l^2}}. \tag{3}$$

As a result, our method assigns higher scores to trackers comparable to the top tracker t_b and low scores to outliers (far neighbors) for a v_l. Note that since (3) is a monotonically decreasing function on $[0, 1]$, if $q_{1l} > q_{2l}$, i.e., a tracker t_1 has a better performance than a tracker t_2, then with (3) $s_{1l} > s_{2l}$.

When there are no outliers (for example, when all FR values are zero), the MAD in (2) may be zero, resulting in $\sigma_l = 0$. We thus modify (3) to

$$s_{il} = \begin{cases} \frac{1}{1+\frac{e_{il}^2}{2\sigma_l^2}} & : \sigma_l \neq 0 \\ q_{il} \cdot (1 - e_{il}) & : \sigma_l = 0 \end{cases} \tag{4}$$

Once the scores $\{s_{il}, i = 1 \cdots T, l = 1 \cdots L\}$ are computed, we combine all s_{il} of a dataset into a single score

$$s_i = \underset{1 \le l \le L}{\text{Mean}}\{s_{il}\}. \tag{5}$$

We select the mean since the $\{s_{il}\}$ distribution is symmetric.

2.2. Robust Grouping of Trackers

Some trackers may have "similar" performance scores, and clustering them into groups can make a fairer comparison. We propose a grouping algorithm to cluster a set of trackers $\{t_i\}, i = 1, \cdots, T$, into groups $\{g_j\}, j = 1, \cdots, G$, based on their scores $\{s_i\}$; $G \le T$. Each group $\{g_j\}$ consists of a set of trackers having similar scores s_i. For this, we define the robust scale σ_s of scores $\{s_i\}$ as

$$\sigma_s = c_s \cdot MAD\{\eta_i\}, \qquad \eta_i = s_b - s_i. \tag{6}$$

η_i is the error between the highest score s_b among the $\{s_i\}$ and a score s_i in $\{s_i\}$ of tracker i. The robust point estimate c_s depends on the family distribution of $\{\eta_i\}$. We then assign a tracker t_i to a group g_j if the error $\eta_i \le \sigma_s$. Once a tracker is given a group, we exclude it from the set $\{t_i\}$. We repeat the process for the set of not-yet-grouped trackers until all trackers are assigned a group: in each iteration, we update s_b, the errors $\{\eta_i\}$, the scale estimate σ_s, and compare η_i to

σ_s for each tracker. To determine the robust point estimate c_s in (6), we examined the distribution of $\{\eta_{il}\}$. Their histogram was not conclusive about the type of their probability distribution represents. Still, we notice it has heavy, slowly-decaying tails that do not fall to zero. The role of c_s in (6) is essential for grouping since it determines which trackers are clustered based on how similar their scores $\{s_i\}$ are. The smaller c_s is, the more restrictive (6) is in grouping similar trackers. In [12] (see Table 2.4), robust point estimates c_s are derived for different distributions. We select the smallest, i.e., $c_s = 0.9102$, which gives the most restrictive grouping, i.e., it is harder for a not-similar tracker to become an outlier in a group.

Algorithm 1 summarizes the proposed grouping: 1) Calculate the best score s_b among the set $\{s_i\}$. 2) Compute the errors $\{\eta_i\}$ and the scale estimate σ_s as in (6). 3) Assign tracker t_i a group g_j if η_i is within the range defined by σ_s. 4) Increment the counter j after each grouping round. The algorithm ends when each t_i is assigned a g_j.

Algorithm 1: Grouping of trackers $\{t_i\}$.

Data: Scores $\{s_i\}$ of the set of trackers $\{t_i\}$.
Result: Groups $\{g_j\}$.

$c_s = 0.9102$; $j = 1$;
while $\{t_i\} \neq \emptyset$ **do**
  $\{g_j\} = \emptyset$;
  $s_b = \max(\{s_i\})$;
  for *each* t_i *in* $\{t_i\}$ **do**
    $\eta_i = s_b - s_i$;
  end
  $\sigma_s = c_s \cdot MAD(\{\eta_i\})$;
  for *each* t_i *in* $\{t_i\}$ **do**
    if $\eta_i \leq \sigma_s$ **then**
      $\{g_j\} = \{g_j\} + t_i$;
      $\{t_i\} = \{t_i\} - t_i$;
    end
  end
  Output: $\{g_j\}$;
  $j++$;
end

3. SIMULATION RESULTS

To test our ranking approach, we use the datasets: OTB-100 [6], VOT2018-ST [2], NfS-30 [5], and VOT2018-LT [2]. To reduce possible bias towards certain datasets and test generalization ability of trackers, we merged all 260 videos of the short-term datasets OTB, VOT-ST, and NfS into a single short-term OTB+VOT-ST+NfS. We use the performance measures: AOR for tracker accuracy (the higher, the better) and FR for tracker robustness (the lower, the better), and 20 trackers of different methodologies, performances, and speeds: ATOM [14], CFWCR [15], CREST [16], CSRDCF [17], DASIAMRPN [18], DAT [19], DIMP [20], DLST [21], DSST [22], ECO [23], IBCCF [24], KCF [25], LADCF [26], MCCT [27], MDNET [28], SAMF [29], SIAMFC [30], SIAMRPN++ [31], STAPLE [32], and STRCF [33]. We used the code of the authors and ran it ten times.

3.1. Results on short and long-term datasets

For the short-term OTB+VOT-ST+NfS, Table 1 compares the widely-used mean, and our ranks for the AOR and FR metrics. It highlights two advantages of our ranking method. 1) It allows for fairer distinction between trackers than the mean method; for example, note in the AOR "Mean" column that the difference between the best (DIMP) and second best (ATOM) is only 0.0362; such a low difference does not justify ranking DIMP as "better". Instead, our scoring and grouping method classifies both trackers as group 1. Overall, our ranking method statistically highlights fairer similarly-performing trackers; for example, ATOM and DIMP are group 1 in both AOR and FR, and SIAMRPN++ is group 1 only in terms of FR; also, we see that a good number of trackers occupy group 2 in both AOR and FR. 2) We can combine multiple AOR and FR scores into one AOR/FR score, as seen under the column "Avg score"; this allows to rank trackers using that single score and at a higher level than the widely-used mean. (Recall that our scores are always in $[0, 1]$, and the higher, the better.)

Table 1: OTB+VOT-ST+NfS: comparing means and our scores and groups. Top trackers are in bold, blue, and green.

	AOR			FR			Avg score
Tracker	Mean	Score	Grp	Mean	Score	Grp	
ATOM	0.5906	0.7618	**1**	0.1306	0.8748	**1**	0.8183
CFWCR	0.5124	0.6496	2	0.1811	0.8318	2	0.7408
CSRDCF	0.4582	0.5638	4	0.2417	0.7757	3	0.6698
CREST	0.4396	0.5494	4	0.2784	0.7397	3	0.6446
DASIAMRPN	0.4696	0.5722	4	0.2050	0.7963	2	0.6843
DAT	0.2910	0.3357	7	0.4005	0.5694	7	0.4526
DIMP	**0.6269**	**0.8272**	**1**	**0.1047**	**0.9096**	**1**	**0.8685**
DLST	0.4414	0.5479	4	0.2667	0.7371	3	0.6426
DSST	0.3692	0.4584	5	0.3919	0.6091	5	0.5338
ECO	0.5290	0.6837	2	0.1913	0.8234	2	0.7536
IBCCF	0.4890	0.6246	3	0.2409	0.7718	3	0.6982
KCF	0.3243	0.3757	6	0.4022	0.5948	6	0.4853
LADCF	0.5155	0.6534	2	0.2193	0.7968	2	0.7251
MCCT	0.4693	0.5943	3	0.2690	0.7492	3	0.6718
MDNET	0.5222	0.6823	2	0.1996	0.8187	2	0.7505
SAMF	0.3995	0.4877	5	0.3367	0.6878	4	0.5878
SIAMFC	0.4165	0.5174	4	0.3177	0.6934	4	0.6054
SIAMRPN++	0.4826	0.5809	3	0.1510	0.8546	**1**	0.7178
STAPLE	0.4151	0.5154	4	0.3103	0.7120	4	0.6138
STRCF	0.5031	0.6328	2	0.2299	0.7802	2	0.7065

We applied our ranking method on the 35 long-term videos of VOT2018-LT for 18 trackers. (Note that the codes available from the authors of DLST and SIAMFC did not run for VOT2018-LT.) Table 2 compares the means and our ranks (scores and groups) of AOR and FR. As with the short-term dataset, we can easily merge multiple scores (here AOR and FR) into one to simplify ranking, as seen in the Table under the column "Avg score". Also, the difference between the best (bold, DIMP) and second-best (blue,DASIAMRPN) AOR means is minimal at 0.0649 while the difference between their respective scores is 0.2237, which allows statistically to justify better ranking them as best and second best; indeed, DIMP is the only tracker in group 1 for AOR, but that both DIMP and SIAMRPN++ are in group 1 for FR.

Table 2: VOT-LT dataset: Comparing means and our score and group. Top three trackers are in bold, blue, and green.

	AOR			FR			Avg score
Tracker	Mean	Score	Grp	Mean	Score	Grp	
ATOM	0.4303	0.6304	2	0.3573	0.6538	2	0.6422
CFWCR	0.3246	0.3934	4	0.3685	0.6197	2	0.5066
CSRDCF	0.2506	0.2634	6	0.5113	0.4647	4	0.3641
CREST	0.3029	0.3742	4	0.3982	0.6059	2	0.4901
DASIAMRPN	0.4468	0.6333	2	0.3404	0.6583	2	0.6459
DAT	0.2079	0.2065	7	0.5370	0.4122	5	0.3094
DIMP	**0.5117**	**0.8571**	**1**	**0.2638**	**0.8316**	**1**	**0.8444**
DSST	0.2289	0.2490	6	0.6127	0.3815	5	0.3153
ECO	0.3344	0.4342	3	0.3854	0.6521	2	0.5432
IBCCF	0.3126	0.3888	4	0.4987	0.5238	3	0.4563
KCF	0.1504	0.1581	8	0.6938	0.2689	7	0.2135
LADCF	0.3547	0.4872	3	0.4280	0.5841	2	0.5357
MCCT	0.2874	0.3088	5	0.5172	0.4651	4	0.3870
MDNET	0.3362	0.4372	3	0.4126	0.5860	2	0.5117
SAMF	0.2382	0.2550	6	0.5394	0.4427	4	0.3489
SIAMRPN++	0.4219	0.5783	2	0.3213	0.7456	**1**	0.6620
STAPLE	0.2264	0.2262	7	0.6244	0.3388	6	0.2826
STRCF	0.3238	0.4383	3	0.4388	0.5757	2	0.5071

3.2. How Stable Is Our Method?

To demonstrate the stability of our scoring method, we added impulse noise to the original performance data $\{q_{il}\}$. Impulse noise can represent faulty tracker performance due to changes such as occlusion or fast motion. Given the original $\{q_{il}\}$ and their noisy performance version $\{q_{il}^n\}$. Let the estimated scores (using our method) be $\{s_i\}$, and the estimated means (using the average of averages method) be $\{\alpha_i\}$. Let their noisy scores be $\{s_i^n\}$, and noisy means be $\{\alpha_i^n\}$. The scores are more robust than the means if

$$\forall i, \quad \Psi(s_i, \mu_{s_i^n}) > \Psi(\alpha_i, \mu_{\alpha_i^n}), \tag{7}$$

where $\Psi(\cdot) = \frac{min(\cdot)}{max(\cdot)} \in [0, 1]$ is a min-max ratio, i.e, the higher the ratio, the better. $\mu_{s_i^n} = \frac{1}{K}\sum_{k=1}^{K} s_i^k$ is the average of the trackers' scores over K noise levels, and $\mu_{\alpha_i^n} = \frac{1}{K}\sum_{k=1}^{K} \alpha_i^k$ is the average of the trackers' means over the same K noise levels. In other words, our score s_i is more robust than the mean α_i if, each t_i yields a score ratio $\Psi(s_i, \mu_{s_i^n})$ closer to 1 than the mean ratio $\Psi(\alpha_i, \mu_{\alpha_i^n})$. We added four levels of noise, i.e., $K = 4$ for (7) by selecting the impulse noise densities 0.05, 0.2, 0.35, and 0.5 . We ran the experiments 50 times. Table 3 displays the ratios $\Psi(\alpha_i, \mu_{\alpha_i^n})$ and $\Psi(s_i, \mu_{s_i^n})$. It shows that our scoring responds moderately to even strong variations in the data, not just on average, but for every tracker i. The score ratio stays above 0.995 for trackers, and the scores perform much better than the means.

Both the proposed score s_i and the widely-used mean α_i are generated from the same data $\{q_{il}\}$ and thus are correlated. However, when a tracker has inconsistent performance across a dataset, s_i better represents that performance and a tracker can be assigned a different rank using our scores compared to the mean, as shown in Table 4.

Table 3: OTB+VOT-ST+NfS: AOR mean ratios and score ratios under impulse noise averaged for densities 0.05, 0.2, 0.35, and 0.5 over 50 runs; the higher the ratio, the better.

Impulse noise	Mean ratio $\Psi(\cdot)$	Score ratio $\Psi(\cdot)$
ATOM	0.9579	**0.9997**
CFWCR	0.9936	**0.9999**
CSRDCF	0.9755	**0.9975**
CREST	0.9637	**0.9966**
DASIAMRPN	0.9824	**0.9980**
DAT	0.8350	**0.9987**
DIMP	0.9442	**0.9983**
DLST	0.9646	**0.9967**
DSST	0.9111	**0.9992**
ECO	0.9844	**0.9981**
IBCCF	0.9937	**0.9978**
KCF	0.8704	**0.9950**
LADCF	0.9917	**0.9979**
MCCT	0.9821	**0.9998**
MDNET	0.9884	**0.9993**
SAMF	0.9350	**0.9971**
SIAMFC	0.9480	**0.9979**
SIAMRPN++	0.9901	**0.9992**
STAPLE	0.9464	**0.9982**
STRCF	**0.9985**	0.9981
Average	0.9578	**0.9982**

Table 4: OTB-100: AOR and FR mean, score, and group. Top five trackers are in bold, blue, green, brown, and pink.

	AOR			FR		
Tracker	Mean	Score	Grp	Mean	Score	Grp
ATOM 0.6604	0.7459	**1**	0.0617	0.9165	**1**	
CFWCR	0.6547	0.7310	**1**	0.0670	0.9344	**1**
CSRDCF	0.5843	0.5880	2	0.1142	0.8763	2
CREST	0.5819	0.5949	2	0.1221	0.8666	2
DASIAMRPN	0.6003	0.5925	2	0.0734	0.9075	**1**
DAT	0.3344	0.2545	5	0.3421	0.5270	4
DIMP	0.6781	0.7827	**1**	**0.0395**	**0.9512**	**1**
DLST	0.5487	0.5347	3	0.1476	0.8088	2
DSST	0.5234	0.5268	3	0.2122	0.7264	3
ECO	**0.6783**	**0.7828**	**1**	0.0569	0.9450	**1**
IBCCF	0.6367	0.7016	**1**	0.0809	0.9098	**1**
KCF	0.4814	0.4105	4	0.1994	0.7351	3
LADCF	0.6751	0.7689	**1**	0.0703	0.9363	**1**
MCCT	0.6386	0.6927	**1**	0.0885	0.8925	**1**
MDNET	0.6614	0.7432	**1**	0.0617	0.9338	**1**
SAMF	0.5632	0.5599	3	0.1479	0.8223	2
SIAMFC	0.5816	0.5985	2	0.1450	0.8140	2
SIAMRPN++	0.5736	0.5314	3	0.0632	0.9202	**1**
STAPLE	0.5901	0.6281	2	0.1500	0.8184	2
STRCF	0.6679	0.7518	**1**	0.07495	0.9166	**1**

4. CONCLUSION

We proposed an alternative way to rank object trackers: not by the typical average of averages but by scoring similarly-performing trackers using an edge-stopping function that depends on a robust scale to robustly distinguish outliers (boundary trackers) from inliers (i.e., similar trackers). We extensively tested our method using five datasets, two performance metrics, and 20 trackers. We showed that the proposed score is more robust to variations than the widely-used average. We suggested combining tracker scores from multiple datasets and metrics into one score to facilitate ranking. The hyper-parameters of our method are determined using robust error norms, which is why it is robust. Our method easily allows the addition of new trackers to the pool $\{t_{il}\}$ for comparison: for a video l, all trackers (from 1 to T) are used to compute the final score.

References

[1] S. Javed et al., "Visual object tracking with discriminative filters and siamese networks: A survey and outlook," *IEEE Trans. Pattern Anal. Machine Intell.*, 2022.

[2] M. Kristan, A. Leonardis, J. Matas, et al., "Visual object tracking VOT2018 challenge results," in *European Conf. Computer Vision Workshops*, 2018.

[3] H. Fan, H. Bai, L. Lin, et al., "LaSOT: A high-quality large-scale single object tracking benchmark," *Int J Comput Vis*, vol. 129, pp. 439–461, 2021.

[4] P. T. Von Hippel, "Mean, median, and skew: Correcting a textbook rule," *Journal of Statistics Education*, vol. 13, no. 2, 2005.

[5] H. Kiani Galoogahi, A. Fagg, C. Huang, D. Ramanan, and S. Lucey, "Need for speed: A benchmark for higher frame rate object tracking," *arXiv preprint arXiv:1703.05884*, 2017.

[6] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 37, pp. 1442–1468, 2015.

[7] Y. Pang and H. Ling, "Finding the best from the second bests - inhibiting subjective bias in evaluation of visual tracking algorithms," in *IEEE Int. Conf. Computer Vision*, 2013.

[8] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," *Stanford InfoLab*, 1999.

[9] B. D. Cullity, *The rating of chessplayers, past and present*, Ishi Press, Batsford London, 1978.

[10] M. Glickoman, "Parameter estimation in large dynamic paired comparison experiments," in *Journal of the Royal Statistical Society Series C Applied Statistics*, 1999, pp. 377–394.

[11] T. Ghoniemy, J. Valognes, and M. Amer, "Robust scoring and ranking of object tracking techniques," in *IEEE Int. Conf. Image Processing (ICIP)*, 2018, pp. 236–240.

[12] D. J. Olive, *Applied Robust Statistics*, University of Minnesota, 1998.

[13] M. J. Black, G. Sapiro, D. Marimont, and D. Heeger, "Robust anisotropic diffusion," in *IEEE Trans. Image Process.*, 1998, vol. 7, pp. 421–432.

[14] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ATOM: Accurate tracking by overlap maximization," in *IEEE Conf. Computer Vision Pattern Recognition*, 2019.

[15] Z. He, Y. Fan, J. Zhuang, Y. Dong, and H. Bai, "Correlation filters with weighted convolution responses," in *IEEE Int. Conf. Computer Vision*, 2017.

[16] Y. Song, C. Ma, L. Gong, J. Zhang, R. Lau, and M.-H. Yang, "Crest: Convolutional residual learning for visual tracking," in *IEEE Int. Conf. Computer Vision*, 2017, pp. 2555–2564.

[17] A. Lukežič, T. Vojir, L. Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter tracker with channel and spatial reliability," *Int. J. Computer Vision*, 2018.

[18] Z. Zhu, Q. Wang, L. Bo, W. Wu, J. Yan, and W. Hu, "Distractor-aware siamese networks for visual object tracking," in *European Conf. Computer Vision*, 2018.

[19] H. Possegger, T. Mauthner, and H. Bischof, "In defense of color-based model-free tracking," in *IEEE Conf. Computer Vision Pattern Recognition*, 2015.

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

[21] L. Yang, R. Liu, D. Zhang, and L. Zhang, "Deep location-specific tracking," in *ACM Conf. on Multimedia*, 2017, pp. 1309–1317.

[22] M. Danelljan, G. Häger, F. Shahbaz Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *British Machine Vision Conf.*, 2014.

[23] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *IEEE Conf. Computer Vision Pattern Recognition*, 2017.

[24] F. Li, Y. Yao, P. Li, D. Zhang, W. Zuo, and M.-H. Yang, "Integrating boundary and center correlation filters for visual tracking with aspect ratio variation," in *IEEE Int. Conf. Computer Vision Workshops*, 2017.

[25] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Machine Intell.*, pp. 583–596, 2015.

[26] T. Xu, Z.-H. Feng, X.-J. Wu, and J. Kittler, "Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual tracking," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5596–5609, 2019.

[27] N. Wang, W. Zhou, Q. Tian, R. Hong, M. Wang, and H. Li, "Multi-cue correlation filters for robust visual tracking," in *IEEE Conf. Computer Vision Pattern Recognition*, 2018.

[28] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *IEEE Conf. Computer Vision Pattern Recognition*, June 2016.

[29] L. Yang and Z. Jianke, "A scale adaptive kernel correlation filter tracker with feature integration," in *European Conf. Computer Vision*, 2014, pp. 254–265.

[30] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *European Conf. Computer Vision Workshops*, 2016, pp. 850–865.

[31] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++: Evolution of siamese visual tracking with very deep networks," in *IEEE Conf. Computer Vision Pattern Recognition*, June 2019.

[32] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *IEEE Conf. Computer Vision Pattern Recognition*, June 2016.

[33] F. Li, C. Tian, W. Zuo, L. Zhang, and M.-H. Yang, "Learning spatial-temporal regularized correlation filters for visual tracking," in *IEEE Conf. Computer Vision Pattern Recognition*, 2018.