Finding the Best from the Second Bests – Inhibiting Subjective Bias in Evaluation of Visual Tracking Algorithms

Yu Pang Haibin Ling

Department of Computer and Information Science, Temple University, Philadelphia, USA

{yu.pang, hbling}@temple.edu

Abstract

Evaluating visual tracking algorithms, or "trackers" for short, is of great importance in computer vision. However, it is hard to "fairly" compare trackers due to many parameters need to be tuned in the experimental configurations. On the other hand, when introducing a new tracker, a recent trend is to validate it by comparing it with several existing ones. Such an evaluation may have subjective biases towards the new tracker which typically performs the best. This is mainly due to the difficulty to optimally tune all its competitors and sometimes the selected testing sequences. By contrast, little subjective bias exists towards the "second best" ones[1] *in the contest. This observation inspires us with a novel perspective towards inhibiting subjective bias in evaluating trackers by analyzing the results between the second bests. In particular, we first collect all tracking papers published in major computer vision venues in recent years. From these papers, after filtering out potential biases in various aspects, we create a dataset containing many records of comparison results between various visual trackers. Using these records, we derive performance rankings of the involved trackers by four different methods. The first two methods model the dataset as a graph and then derive the rankings over the graph, one by a rank aggregation algorithm and the other by a PageRank-like solution. The other two methods take the records as generated from sports contests and adopt widely used Elo's and Glicko's rating systems to derive the rankings. The experimental results are presented and may serve as a reference for related research.*

1. Introduction

As an important topic in computer vision, visual tracking has been a widely explored area attracting a great amount of research efforts. Over the decades, dozens of visual tracking algorithms, or *trackers* in short, have been developed and a great packs of public datasets are available alongside [73]. Evaluation of these algorithms, though of great interest, remains a challenge due to the hardly avoidable biases. These biases arise from many sources such as tracker parameters (e.g., number of particles), initialization, sequences used *etc*. It is therefore hard to tune many different trackers for a fair comparison.

Tracker	A	B	C ('our previous')	D	E ('ours')
Seq. 1	17.5	56.7	11.3	10.5	5.0
Seq. 2	7.0	39.2	8.5	39.2	6.1

Table 1. Tracking evaluation results (in terms of average center location errors) from a mock paper using two sequences. Tracker E is newly proposed in the mock paper; C is from the authors' previous work; other trackers (A, B and D) were proposed in papers sharing no co-author with the mock paper.

Several evaluation frameworks have been proposed and tested during the last two decades, such as the *International Workshop on Visual Surveillance* (VS), the *International Workshop on Performance Evaluation of Tracking and Surveillance* (PETS) and the VIVID Tracking Evaluation Website [14]. The basic idea is to test the trackers on a lot of public datasets and evaluate the results using uniformed metrics. The trackers are submitted by their authors and thus by assumption they are tuned optimally to win. These evaluations are considered fair in general. However, many state-of-the-art trackers have not been tested this way.

A recent trend when introducing a new tracker is to validate it by comparing with several existing state-of-the-art trackers. A byproduct of many such papers is the numerous evaluations for various tracking algorithms. However, such papers often have subjective biases towards their proposed new trackers which typically perform the *best* in the evaluation. This is understandable and reasonable. On one hand, new trackers usually have some advantages that the authors aim to highlight. On the other hand, it is non-trivial for the paper authors to optimize all other trackers involved

[1] We treat all trackers other than the "best" as second best ones.

in the contest. Nevertheless, we observe that there is little such bias towards the *second best* ones. For example, Table 1 simulates results in a typical tracking paper, where the newly proposed tracker E performs the best as expected. Though there may be bias in favor of E and possibly C as well, the comparisons between A, B and D are usually trustworthy.

This observation inspires us with a novel perspective towards unbiased evaluation of visual trackers – to explore the unbiased comparison information among the second best trackers reported by previous tracking papers. With this idea, we first collect all tracking papers published in major computer vision venues in recent years. From these papers, after filtering out potential biases in various aspects, we create a dataset containing many records of comparison results between various visual trackers. Using these records, we derive performance rankings of the involved trackers by four different methods. The first two methods model the dataset as a graph and then derive the rankings over the graph, one by a rank aggregation algorithm and the other by a PageRank-like solution. The other two methods take the records as generated from sports contests and adopt widely used Elo's and Glicko's rating systems to derive the rankings. The experimental results are presented and may serve as a reference for researchers interested in visual tracking.

Our contributions are twofold: First, we propose a subjective bias-resisting tracking evaluation method which has never been explored to the best of our knowledge. Second, the evaluation results provide a reference for related applications.

In the following section, we briefly summarize the related work. After that, we formulate our task and describe the data collection in Section 2. Then, the ranking methods are introduced in Sections 3, 4 and 5. Experimental results are reported in Section 6, followed by conclusion in Section 7.

1.1. Related work

The *VS Workshop* and the *PETS Workshop* are among the earliest ones to put efforts on comparing different trackers on public datasets. They provide datasets on different aspects of tracking scenarios and researchers apply their trackers on the same datasets, so that people can use certain metrics or evaluation methods to compare the results. At the early stage, the workshops focused more on bringing up new evaluation methods and tracking algorithms and there is no explicit comparison summary in each workshop. In recent years, *PETS* has covered many aspects of tracking scenarios and now its focus shifts towards multi-target tracking. But we still have no direct performance comparisons on single-target tracking algorithms. Further more, since the data is given at the first place, it is a temptation to tune tracking parameters to obtain the best performance on the specific data, thus the results may not generalize well.

There are also some literatures that evaluate performances of trackers, such as [35, 67]. Typically, these papers focus on introduction of new evaluation framework, including standard input datasets, initializations, and/or evaluation criteria. Despite the efforts, it is still hard to run various trackers without biases due to the reason mention in previous sections. Another type of solutions is to develop publicly available platforms, e.g., softwares and websites, for the evaluation. One such example is the VIVID [14]. Unfortunately, most of these attempts ended before long. Recently, Wu *et al*. [68] built a large benchmark on visual tracking and evaluated thoroughly the performances of over 29 trackers. By contrast, our approach performs the evaluation from a totally different perspective and can be treated as a complementary view for tracking evaluation.

2. Data Preparation

The key in data collection and processing is to inhibit potential biases as much as possible. Hereafter, we call a visual tracking algorithm being evaluated as a **tracker**, a paper containing comparison results as a **contest paper**, or **contest** for short and a pairwise comparison between two trackers as a **record**. In the following we describe each step in our data preparation.

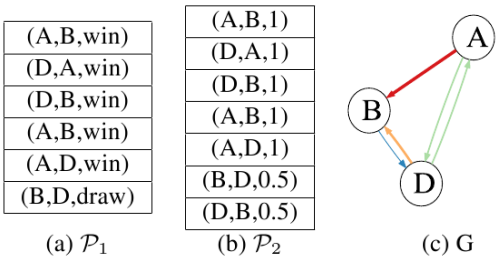
2.1. Collecting Contest Papers

First of all, we focus on single target tracking algorithms in this study. So we restrict contests to papers that have the same focus. We also exclude the papers designed to track specific models, such as those for eye tracking. Now the topic is determined, we collect contest papers from major computer vision journals including PAMI and IJCV, from 2000 to up-to-date issue. We also collect the data from major computer vision conferences including ICCV, CVPR and ECCV from 2005 to 2013. Interestingly, we have not found any contest satisfying all our criteria (see the rest of this section) before the year 2008 in journals or before 2009 in conferences.

After the initial collection, we need to filter out some contests to reduce potential biases. This is done according to the following criteria.

- **Conference to journal extension.** It is not uncommon to extend a conference paper to a journal one. Including both versions will apparently put more weight on the results in them. For this reason, we discard all conference papers that have corresponding journal extensions in our initial collection.

- **Duplicate experimental results.** There are a few contests having their experimental results partly imported



(a) $\mathcal{P}_1$ (b) $\mathcal{P}_2$ (c) G

Figure 1. The conversion example. (a) The record set $\mathcal{P}_1$ extracted using the example in Table 1. (b) The converted record set $\mathcal{P}_2$ as explained in Equation 1. (c) The resulting graph G constructed from the records.

from other literatures, which could cause unfairness if included. Such papers are excluded as well.

The final contest set has 45 contests, including [5–8, 10, 11, 16, 17, 23, 28, 29, 33, 34, 38–52, 54, 58, 59, 61, 64, 65, 69, 71, 72, 74–81].

2.2. Extracting Records

The key to extract records from contest papers is again to inhibit potential biases. For each contest, we remove the results which contain the trackers proposed by the authors of this very contest, including both the newly proposed one and possibly the trackers in the authors' previous study. This step eliminates potential biases that favor the authors' own work.

After the above filtering, what left in a contest paper is the evaluation of several trackers on several sequences. For example, only trackers A, B and D remains from Table 1. We will then construct two representations for the data. The ranking representation contains the rankings from each sequence. For example, from Table 1 we will extract $\pi_1 : (D < A < B)$ and $\pi_2 : (A < B = D)$. From now on π_i will be named partial rankings following the notations in [2] and this representation will be mainly used in the algorithm described in Section 3. The other representation is the pairwise representation. In particular, for every pair of trackers in the partial ranking, say A and B, we will generate a record as follows: If A performs better than or as good as B, we generate a record as $(\mathrm{A}, \mathrm{B}, \mathrm{label})$, such that $\mathrm{label} = 'win'$ if A is better or $'draw'$ otherwise. In this way, we convert results from a contest into a set of records. Figure 1(a) lists all records extracted from the mock paper in Table 1.

Following the above procedure, we obtain a set of records involving 48 trackers. Each tracker appears in 193.4 records on average. To further reduce the chance of biases, we remove any trackers who appear in less than 10 records or in only one contest. After all the cleaning, we have 15 trackers, 664 partial rankings and 6280 records among which there are 151 records of $'draw'$. We denote the tracker sets as $T = \{t_1, ..., t_n\}$, $n = 15$, and the record sets as

$$\mathcal{P}_1 = \Big\{ < t_l, t_r, l >^{(i)} : t_l, t_r \in T, \\ l \in \{'win', 'draw'\}, i = 1, \ldots, n_1 = 6280 \Big\}, \quad (1)$$

$$\mathcal{P}_2 = \Big\{ < t_l, t_r, a >^{(i)} : t_l, t_r \in T, \\ \alpha^i \in \{0.5, 1\}, i = 1, \ldots, n_2 = 6431 \Big\}, \quad (2)$$

where $\mathcal{P}_1$ is the raw record set and $\mathcal{P}_2$ is derived from $\mathcal{P}_1$, such that every raw $'win'$ record gains a value $\alpha = 1$ and every raw $'draw'$ record $(\mathrm{A}, \mathrm{B}, 'draw')$ is split into two records each with $\alpha = 0.5$, i.e. $(\mathrm{A}, \mathrm{B}, 0.5)$ and $(\mathrm{B}, \mathrm{A}, 0.5)$. An example is shown in Figure 1(b).

The 15 trackers are Meanshift [15], ColorPF [55], IVT [57], Ensemble [4], OFS [13], FragT [1], OBT [25], SemiBoost [26], MIL [5], L1T [51], BOBT [60], TLD [34], VTD [37], Struck [27] and MTT [76].

An illustrative figure is presented in Figure 2. For a directed edge from A to B, the color and thickness are proportional to the number of records that agree on "A is better than B".

It is worth noting there are some factors ignored in the above data preparation, such as the degree of challenges of different sequences or the extent to which one tracker outperforms another one, all of which could potentially affect the ranking results. It is unrealistic to model these factors given the huge amount of data needed. That said, given the diversity of the current dataset, the proposed approaches can produce significant results at least for the top ranked trackers, as shown in the consistency in the results of different rankings (Section 6). More discussions can be found in Section 6.3. It is also worth noting that there are many important tracking papers that do not follow the above evaluation paradigms for either trackers or contests. These papers, such as recent studies in [9, 12, 21, 30, 31, 56, 63, 66, 70] to name a few, are therefore not included.

3. Rank Aggregation Algorithm

Rank aggregation has been widely used in webpage ranking and other fields [2, 3, 18]. Given a universe set $T = \{t_1, t_2, ..., t_n\}$ and a set of partial rankings $\pi = \{\pi_1, \pi_2, ..., \pi_m\}$, where $\pi_i = [t_{i_1} \geq t_{i_2} \geq ... \geq t_{i_d}], i_d \leq n$ and $t_{i_k} \in T$ for some ordering relationship $\geq$. The task is to find a full ranking π', *i.e.*, a permutation of set T that satisfies some objective function. A normally chosen objective function is the generalized Kendall-τ distance which is considered to have many advantages [2, 18]. It is defined as $d(\pi_a, \pi_b) = |\{(i, j) | i < j, \pi_a(i) < \pi_a(j), \pi_b(i) > \pi_b(j)\}|$, where $\pi_a(i)$ is the element in π_a at position i *etc*. This measures the number of disagreements between two rankings π_a and π_b. The rank aggregation model has an equivalent

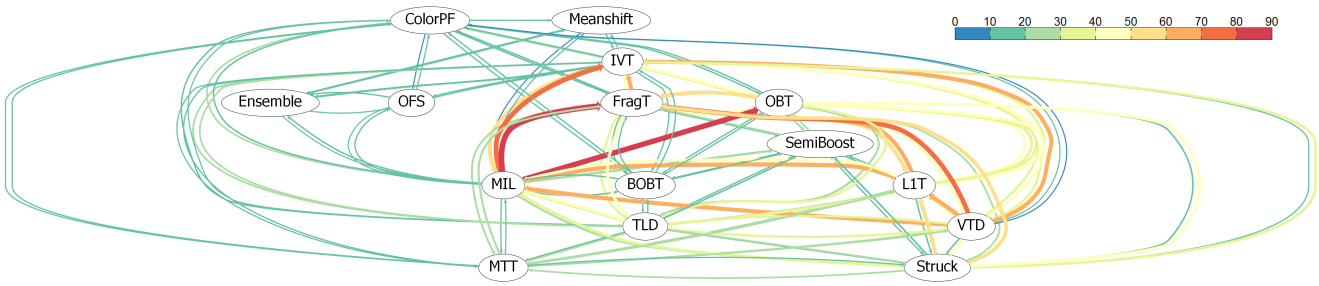


Figure 2. The weighted directed graph constructed from the collection of records. The color and thickness indicate the number of records agree on each edge. The name correspondences are listed at the end of Section 2.2.

weighted graph $G = (V, A)$, where $V = T$ is the same universe set and any $w_{ij} \in A$ is the fraction of inputs ranking i before j.

Our ranking representation can be naturally fit into the rank aggregation model. To construct the weighted graph G, it is more convenient to use the pairwise representation. We will first construct an unnormalized version $G' = (V, A')$, where $w'_{ij} \in A'$ is the number of $(i, j, 1)$ in the pair representation set $\mathcal{P}_2$. Then we have

$$w_{ij} = \frac{w'_{ij}}{w'_{ij} + w'_{ji}}. \tag{3}$$

An example graph corresponding to the record set in Figure 1(a) is shown in Figure 1(c). An unnormalized graph example of our total collected data is shown in Figure 2.

Kemeny optimization is a widely used method to solve this problem. However, the problem itself is proved to be NP-hard, thus many approximation algorithms have been proposed [2, 3, 18]. We will adopt the *LpKwikSort*$_h$ algorithm described in [2]. Given the weighted graph $G = (V, A)$ and a predefined piecewise-linear function h, the algorithm will output a full ranking that is proved to be within 3/2 of the true optimal value. Due to the page limitation, we encourage the reader to refer to [2] for details.

4. Ranking by a PageRank-like Algorithm

The graph model shares a lot of similarities with the widely known and used PageRank algorithm [53]. We consider the tracker nodes as website nodes and the edges as hyperlinks except they have weights. However, we also need to reverse the direction of the edges, which can be interpreted as the 'lose' tracker has a hyperlink to the 'win' tracker. Then we can ask the question who has the highest authority power in the graph.

It is well-known that Pagerank is closely related to the Markov chain, thus we will first construct the transition matrix M_t in the following steps:

1. For all $i \neq j$, if i and j are never compared, *i.e.* $w_{ij} = 0$ where $w_{ij} \in A$, then $t_{ij} = t_{ji} = 0.5$. Otherwise, $t_{ij} = w_{ij}$ and $t_{ji} = w_{ji}$.
2. Divide M_t by $|V|$ the number of nodes, then set $t_{ii} = 1 - \sum_{i \neq j} t_{ij}$.
3. Make M_t ergodic by multiplying M_t by $1 - \epsilon$ then add $\epsilon/|V|$, where ϵ is set empirically to 0.15.
4. Transpose M_t, so that the column sum is 1 for every column.

Then the next step is to find the eigenvector corresponding to the eigenvalue of 1. The ranking result will then be in the descending order of the eigenvector scores.

5. Sports Ranking Algorithms

Ranking is an essential problem in competitive sports which share many similarities with our problem. If we view the collected records $\mathcal{P}_1$ as competition results between several trackers, then our problem naturally simulates a sport game and each tracker naturally an athlete. There are in fact a bunch of sports that share the similar settings as in our problem, such as Chess [22], Go [19], Electronic sports and many others. We borrow ideas from the well-known ranking systems used for them.

5.1. Elo's Rating

One of the most successful ranking methods is the Elo's rating [20]. The core idea is that the ranking score is a scaling rating, so that the score difference between two nodes determines an estimation of expected outcomes. Given the ranking scores R_i and R_j for two trackers i and j, the expectation is estimated as:

$$E_{ij} = \frac{1}{1 + 10^{(R_j - R_i)/400}}. \tag{4}$$

An expectation $E_{ij} = 0.80$ means the chance for i to win over j is 80%. Notice that $E_{ij} + E_{ji} = 1$ is always held. Now we denote an actual outcome of the competition as S_{ij}

$$S_{ij} = \begin{cases} 1, & \text{if } i \text{ win over } j \\ 0.5, & \text{draw} \\ 0, & \text{if } i \text{ lose to } j \end{cases} \tag{5}$$

Then, the difference between the expected outcome and the actual outcome will be used to update the ranking score:

$$R_i^{New} = R_i^{Old} + K(S_{ij} - E_{ij}) \tag{6}$$

where K is the updating rate. When K is large, the score changes very fast or in other word is very sensitive to the results. Usually larger values are used for K at beginning and small values are used when sufficient information has been accumulated.

One of the interesting and desired properties of this model is that the ranking score does not necessarily increase when a winning is observed. If the expectation of winning is 1 and we observe a $'\text{win}'$ record, in Equation 6, we will have $S_{ij} - E_{ij} = 0$, thus the score will not be increased. In contrast, if we observe a $'\text{lose}'$ record, the score will have a relative big decrease, because $S_{ij} - E_{ij} = -1$ is the largest negative value it could be. Intuitively speaking, if the strong one wins, there is no surprise and we believe our ranking scores properly model the strengths between the two athletes. But if the strong one loses, we will reconsider our scores as inaccurately reflect the strengths, thus need large changes. Another good property is that if we have a constant number of athletes, the sum of all the scores in the system remains constant at every step. Because after every update, the winning one gains the amount of score exactly the same as the losing one loses.

The above method depends on the order of the input records since the ranking scores are updated in a sequential manner: at each step the result will be updated according to the actual outcome of S_{ij} and the expected outcome of E_{ij}. To address this issue, we use the average ranking position in many uniformly generated random runs.

5.2. Glicko's rating

Glicko's rating [24] is a generalized version of Elo's rating. It uses two parameters to model the rating: R_i is the expected rating score for node i and D_i measures its confidence. More precisely, we are 95% confident that the true rating of the i-th tracker ranges between $R_i - 2D_i$ to $R_i + 2D_i$. This method also introduces a time variable. The model will be updated after each time period. The more results we have for one tracker during one time period, the more we are confident about its estimated ranking, so D_i will decrease after the update. In contrast, if we have a small number of results or none for that tracker, we are less confident and thus the D_i will increase. Another difference is that the final rating score will be the lower 95% confidence score which is $R_i - 2D_i$.

The update formulas are:

$$D_i^{new} = \min(\sqrt{(D_i^{old})^2 + c^2 t}, 350)\ ,$$

$$R_i^{new} = R_i^{old} + \frac{q}{D_i^{-2} + d_i^{-2}} \sum_{j=1}^{m} g(D_j)(s_{ji} - E(s|R_i, R_j, D_j)),$$

where

$$q = \ln 10/400\ , g(D_j) = \frac{1}{\sqrt{1 + 3q^2 D_j^2/\pi^2}}\ ,$$

$$E(s|R_i, R_j, D_j) = \frac{1}{1 + 10^{-g(D_j)(R_i - R_j)/400}}\ ,$$

$$d_i^2 = \frac{1}{q^2 \sum_{j=1}^{m} g^2(D_j) E(s|R_i, R_j, D_j)(1 - E(s|R_i, R_j, D_j))},$$

and c is a decay coefficient which is set to $\sqrt{12000}$ in our study, meaning that after 10 rounds it will take $D = 50$ back to $D = 350$.

6. Experimental Results

6.1. Results

The ranking results using the above four algorithms are shown in Table 2. The *LpKwikSort*$_h$ algorithm is a randomized algorithm, so we run it over 10 million trials and took the one with the smallest overall score as our result shown in Table 2(a). We constructed the Pagerank-like transition matrix as described in Section 4 using the pairwise representation. Table 2(b) is the result. The "score" subcolumn is the eigenvector associated with the eigenvalue of 1. As described in Section 5.1, the order of the input will affect the output of Elo's rating. We used the total 6280 records as the original pool and uniformly generated a random sequence of 200,000 – a sufficiently large amount of records to run the algorithm. We used the traditional setting [20] of the parameters, each tracker was assigned an initial value of 1500 and update rate K was set to 30. We measured the means and standard deviations of the ranking positions for each tracker over 100 different runs as shown in Table 2(c). The result was ranked by their mean ranking positions. Similar to Elo's rating, we used a random sequence of 200,000 records for Glicko's rating. But this time we uniformly generated 10,000 records for one round, run the algorithm for 20 rounds and reported the results. According to [24], the initial score R_i was set to 1500 and D_i was set to 350 for each tracker. The result is shown in Table 2(d) and is ranked by the mean ranking positions over 100 runs.

	Rank aggregation	PageRank-like		Elo's rating		Glicko's rating	
rank	name	name	score	name	score	name	score
1	Struck [27]	Struck [27]	0.1069	Struck [27]	1.05 ± 0.22	Struck [27]	1.00 ± 0.00
2	MTT [77]	MIL [5]	0.0880	ColorPF [55]	3.81 ± 2.10	MIL [5]	2.00 ± 0.00
3	ColorPF [55]	ColorPF [55]	0.0822	MIL [5]	4.43 ± 2.02	VTD [37]	3.00 ± 0.00
4	TLD [34]	OFS [13]	0.0710	TLD [34]	5.32 ± 2.74	TLD [34]	4.00 ± 0.00
5	VTD [37]	TLD [34]	0.0656	VTD [37]	5.33 ± 2.50	OBT [25]	5.45 ± 0.59
6	MIL [5]	BOBT [60]	0.0636	MTT [77]	6.16 ± 2.72	FragT [1]	5.71 ± 0.62
7	OBT [25]	MTT [77]	0.0634	BOBT [60]	7.77 ± 2.95	L1T [51]	6.85 ± 0.44
8	SemiBoost [26]	SemiBoost [26]	0.0633	SemiBoost [26]	8.96 ± 2.61	ColorPF [55]	8.47 ± 0.64
9	L1T [51]	VTD [37]	0.0628	OBT [25]	9.06 ± 2.88	IVT [57]	8.93 ± 0.73
10	FragT [1]	OBT [25]	0.0589	OFS [13]	9.42 ± 3.01	MTT [77]	9.59 ± 0.68
11	OFS [13]	Ensemble [4]	0.0567	FragT [1]	9.56 ± 3.03	SemiBoost [26]	11.00 ± 0.00
12	IVT [57]	Meanshift [15]	0.0552	L1T [51]	9.77 ± 2.68	BOBT [60]	12.00 ± 0.00
13	BOBT [60]	FragT [1]	0.0542	IVT [57]	10.43 ± 2.41	OFS [13]	13.00 ± 0.00
14	Meanshift [15]	IVT [57]	0.0542	Ensemble [4]	14.21 ± 0.50	Ensemble [4]	14.00 ± 0.00
15	Ensemble [4]	L1T [51]	0.0540	Meanshift [15]	14.72 ± 0.53	Meanshift [15]	15.00 ± 0.00
	(a)	(b)		(c)		(d)	

Table 2. The ranking results generated by the four ranking algorithms. The trackers are ranked from top to bottom. Rank aggregation minimize an overall score, thus it has no individual score as shown in (a). The scores for Pagerank-like algorithm are shown in (b). For the latter two algorithms, we show both their average ranking positions and their standard deviations over 100 runs in (c) and (d) as described in Section 5.

6.2. Discussion

It can be seen that these four algorithms agree with each other in a broad sense. That is, the top few trackers are always top and the bottom trackers are often bottom. Also in sports ranking algorithms, we could see they form cliques in terms of their score distances. For example, TLD and VTD in Elo's, OBT and FragT in Glicko's. The average distance between different cliques is much larger than the distance within cliques. This suggests that in different runs, trackers will have different ranking positions within the clique, but the relative ranking positions of different cliques are mostly preserved.

Since we set transition probability 0.5 to those never compared pairs, Pagerank-like algorithm sometimes will overestimate or underestimate trackers. For example OFS, Ensemble and Meanshift have compared to only 4 other trackers within the set, compared to an average number of 8.07. In addition to these 3 trackers, ColorPF and BOBT also have quite low number of records compared with the average. So they tend to be estimated inaccurately. Among the four algorithms, only Glicko's rating considers such problem so that it introduces the confidence. For these less compared trackers, it will use a pessimistic score to rank it lower. Thus is the reason Glicko's rating has a much lower variance than Elo's rating, in other word, it is much stabler.

We have selected 15 out of 48 trackers to be compared according to our criteria. Below we list some explanations why many of them did not pass those criteria.

Some of the early trackers have been integrated into other trackers, such as the particle filter [32], [36]. The idea is widely used in the state-of-the-art trackers, but it is rarely compared as an individual tracker. Except for [55] which is used in several contests, thus is included in our tracker list. Some other early trackers have been considered as the baseline tracker, such as the Meanshift tracker [15]. Such trackers are mostly used to compare with the author's methods in their papers before 2009. However, since most of such papers compared only two trackers and one is their owns, we can not extract anything from these papers based on our criteria described in Section 2.2. Many other trackers including most recently published ones have no open-source codes. It is always hard to fully implement them, thus are rarely compared in the contests. Although some of the trackers are compared in their own authors' new work, according to our criteria, they are not included as well. There are also some newly developed trackers. Due to the limit in time, they are not yet widely tested by others.

6.3. Limitations

Nearly all the evaluation efforts in the existing literatures contain some kinds of biases. Our goal is to provide a novel perspective to look at this problem and be as unbiased as possible. However, we may not be able to avoid the systematic biases. Although we have several thousands of records, it is still quite insufficient to solve the problem. That is also why we have to bootstrap our data in the Elo's rating and Glicko's rating. By bootstrapping, we inevitably use a biased population based on our observations.

Another potential problem is some trackers are sensitive to initialization or parameter tuning, but we have not taken them into consideration. It may introduce biases if we do not have sufficient amount of data. But if we assume all the

experiments in contests are conducted independently, such problem can be neglected as we have more contests. Many of this kind of biases can be alleviated as we accumulate more and more data, so that we could give a better, *i.e.*, more unbiased evaluation.

We also make the assumption that all the sequences are independent, even when the same sequence appears in different contests. Because we have no idea about their implementation details, parameter tunings or initialization as mentioned above.

It is possible that different trackers have different specializations. For example, some trackers may be good at dealing with occlusion, illumination variation, re-identify the missing target and so on. In our paper, we only consider their abilities in overall scenarios so far. In the future if we can categorize the dataset into different scenarios when more records are available, we may be able to provide more specialized rankings.

We would also like to point out that the records we extracted from contests are possibly biased themselves, because some sequences are more popular than others and compared more often. Thus, the whole datasets do not necessarily reflect the real world. Another issue is that many sequences are shot intentionally to address the difficulties in tracking scenario, such as occlusion *etc*. Some of them are shot in constrained environment. So the argument is similar to [62], the whole datasets the tracking community shares may not be a good representation of the real world, thus the ranking results we have may only partially reflect their performances in the real world.

In summary, it is unrealistic to perform a rigorous unbiased evaluation for tracking algorithms. That said, the proposed approach provides a novel and effective way towards reducing the biases in the evaluation. In addition, most of the issues listed above will be mitigated when more and more data becomes available.

7. Conclusion

In this paper, we have proposed a novel method to compare trackers performances and rank them using four different algorithms. Following the trend in tracking papers, we are able to collect a dataset of comparisons of the "second best" ones. There is little subjective bias towards these comparison results, thus we may conduct an unbiased evaluation of the trackers. After filtering out potential biases in various aspects, we construct a dataset containing 15 trackers and 6280 records. We use four different methods to evaluate them. Rank aggregation is to use the partial rankings find a full ranking that optimize some objective function. Pagerank-like algorithm is an analogue to the webpage ranking. The latter two take the records as generated from sports contests and adopt widely used Elo's and Glicko's rating systems to derive the rankings. The results are presented and we have a few discussion on several issues.

Acknowledgment. We thank all anonymous reviewers for valuable suggestions, especially on the rank aggregation algorithm. This work is supported by US National Science Foundation (Grant IIS-1218156).

References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *CVPR*, 2006. 3, 6

[2] N. Ailon. Aggregation of partial rankings, p-ratings and top-m lists. *Algorithmica*, 2010. 3, 4

[3] N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: ranking and clustering. *JACM*, 2008. 3, 4

[4] S. Avidan. Ensemble tracking. *PAMI*, 2007. 3, 6

[5] B. Babenko, M. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *PAMI*, 2011. 3, 6

[6] Y. Bai and M. Tang. Robust tracking via weakly supervised ranking svm. In *CVPR*, 2012. 3

[7] C. Bao, Y. Wu, H. Ling, and H. Ji. Real time robust l1 tracker using accelerated proximal gradient approach. In *CVPR*, 2012. 3

[8] V. Belagiannis, F. Schubert, N. Navab, and S. Ilic. Segmentation based particle filtering for real-time 2d object tracking. In *ECCV*, 2012. 3

[9] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In *CVPR*, 2010. 3

[10] K. Cannons, J. Gryn, and R. Wildes. Visual tracking using a pixelwise spatiotemporal oriented energy representation. In *ECCV*, 2010. 3

[11] L. Cehovin, M. Kristan, and A. Leonardis. Robust visual tracking using an adaptive coupled-layer visual model. *PAMI*, 2012. 3

[12] D. Chen and J. Yang. Robust object tracking via online dynamic spatial bias appearance models. *PAMI*, 2007. 3

[13] R. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. *PAMI*, 2005. 3, 6

[14] R. Collins, X. Zhou, and S. K. Teh. An open source tracking testbed and evaluation web site. In *PETS*, 2005. 1, 2

[15] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 2002. 3, 6

[16] T. Dinh, N. Vo, and G. Medioni. Context tracker: Exploring supporters and distracters in unconstrained environments. In *CVPR*, 2011. 3

[17] G. Duan, H. Ai, S. Cao, and S. Lao. Group tracking: Exploring mutual relations for multiple object tracking. In *ECCV*, 2012. 3

[18] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *ACM-WWW*, 2001. 3, 4

[19] EGF. European go database. http://www.europeangodatabase.eu/EGD/EGF_rating_system.php. 4

[20] A. Elo. *The rating of chessplayers, past and present*. Batsford London, 1978. 4, 5

[21] J. Fan, X. Shen, and Y. Wu. Scribble tracker: a matting-based approach for robust tracking. *PAMI*, 2012. 3

[22] FIDE. World chess federation. http://www.fide.com. 4

[23] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky. Hough forests for object detection, tracking, and action recognition. *PAMI*, 2011. 3

[24] M. Glickman. Parameter estimation in large dynamic paired comparison experiments. *JRSSC*, 1999. 5

[25] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via online boosting. In *BMVC*, 2006. 3, 6

[26] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *ECCV*, 2008. 3, 6

[27] S. Hare, A. Saffari, and P. Torr. Struck: Structured output tracking with kernels. In *ICCV*, 2011. 3, 6

[28] S. He, Q. Yang, R. W. Lau, J. Wang, and M.-H. Yang. Visual tracking via locality sensitive histograms. In *CVPR*, 2013. 3

[29] J. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *ECCV*, 2012. 3

[30] S. Holzer, S. Ilic, and N. Navab. Multi-layer adaptive linear predictors for real-time tracking. *PAMI*, 2013. 3

[31] W. Hu, X. Li, W. Luo, X. Zhang, S. Maybank, and Z. Zhang. Single and multiple object tracking using log-euclidean riemannian subspace and block-division appearance model. *PAMI*, 2012. 3

[32] M. Isard and A. Blake. Condensationconditional density propagation for visual tracking. *IJCV*, 1998. 6

[33] X. Jia, H. Lu, and M. Yang. Visual tracking via adaptive structural local sparse appearance model. In *CVPR*, 2012. 3

[34] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *PAMI*, 2012. 3, 6

[35] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang. Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *PAMI*, 2009. 2

[36] Z. Khan, T. Balch, and F. Dellaert. Mcmc-based particle filtering for tracking a variable number of interacting targets. *PAMI*, 2005. 6

[37] J. Kwon and K. Lee. Visual tracking decomposition. In *CVPR*, 2010. 3, 6

[38] J. Kwon and K. Lee. Tracking by sampling trackers. In *ICCV*, 2011. 3

[39] J. Kwon and K. Lee. Wang-Landau Monte Carlo-based tracking methods for abrupt motions. *PAMI*, 2013. 3

[40] J. Kwon and K. M. Lee. Minimum uncertainty gap for robust visual tracking. In *CVPR*, 2013. 3

[41] G. Li, L. Qin, Q. Huang, J. Pang, and S. Jiang. Treat samples differently: Object tracking with semi-supervised online covboost. In *ICCV*, 2011. 3

[42] M. Li, J. Kwok, and B. Lu. Online multiple instance learning with no regret. In *CVPR*, 2010. 3

[43] X. Li, A. Dick, C. Shen, A. van den Hengel, and H. Wang. Incremental learning of 3d-dct compact representations for robust visual tracking. *PAMI*, 2013. 3

[44] X. Li, A. Dick, H. Wang, C. Shen, and A. van den Hengel. Graph mode-based contextual kernels for robust svm tracking. In *ICCV*, 2011. 3

[45] X. Li, C. Shen, Q. Shi, A. Dick, and A. van den Hengel. Non-sparse linear representations for visual tracking with online reservoir metric learning. In *CVPR*, 2012. 3

[46] Y. Li, H. Ai, T. Yamashita, S. Lao, and M. Kawade. Tracking in low frame rate video: A cascade particle filter with discriminative observers of different life spans. *PAMI*, 2008. 3

[47] D. Liang, Q. Huang, H. Yao, S. Jiang, R. Ji, and W. Gao. Novel observation model for probabilistic object tracking. In *CVPR*, 2010. 3

[48] B. Liu, J. Huang, L. Yang, and C. Kulikowsk. Robust tracking using local sparse appearance model and k-selection. In *CVPR*, 2011. 3

[49] B. Liu, L. Yang, J. Huang, P. Meer, L. Gong, and C. Kulikowski. Robust and fast collaborative tracking with two stage sparse optimization. In *ECCV*, 2010. 3

[50] V. Mahadevan and N. Vasconcelos. Biologically inspired object tracking using center-surround saliency mechanisms. *PAMI*, 2013. 3

[51] X. Mei and H. Ling. Robust visual tracking and vehicle classification via sparse representation. *PAMI*, 2011. 3, 6

[52] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan. Locally orderless tracking. In *CVPR*, 2012. 3

[53] L. Page, S. Brin, R. Motwani, and T. Winograd. *The PageRank citation ranking: Bringing order to the web.* Stanford InfoLab, 1999. 4

[54] D. Park, J. Kwon, and K. Lee. Robust visual tracking using autoregressive hidden markov model. In *CVPR*, 2012. 3

[55] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *ECCV*, 2002. 3, 6

[56] F. Porikli, O. Tuzel, and P. Meer. Covariance tracking using model update based on lie algebra. In *CVPR*, 2006. 3

[57] D. Ross, J. Lim, R. Lin, and M. Yang. Incremental learning for robust visual tracking. *IJCV*, 2008. 3, 6

[58] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof. Prost: Parallel robust online simple tracking. In *CVPR*, 2010. 3

[59] L. Sevilla-Lara and E. Learned-Miller. Distribution fields for tracking. In *CVPR*, 2012. 3

[60] S. Stalder, H. Grabner, and L. Van Gool. Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition. In *ICCV Workshops*, 2009. 3, 6

[61] J. S. Supancic III and D. Ramanan. Self-paced learning for long-term tracking. In *CVPR*, 2013. 3

[62] A. Torralba and A. Efros. Unbiased look at dataset bias. In *CVPR*, 2011. 7

[63] D. Tsai, M. Flagg, A. Nakazawa, and J. M. Rehg. Motion coherent tracking using multi-label mrf optimization. *IJCV*, 2012. 3

[64] D. Wang, H. Lu, and M.-H. Yang. Least soft-thresold squares tracking. In *CVPR*, 2013. 3

[65] S. Wang, H. Lu, F. Yang, and M. Yang. Superpixel tracking. In *ICCV*, 2011. 3

[66] X. Wang, G. Hua, and T. X. Han. Discriminative tracking by metric learning. In *ECCV*. Springer, 2010. 3

[67] H. Wu, A. Sankaranarayanan, and R. Chellappa. Online empirical evaluation of tracking algorithms. *PAMI*, 2010. 2

[68] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *CVPR*, 2013. 2

[69] Y. Wu, H. Ling, J. Yu, F. Li, X. Mei, and E. Cheng. Blurred target tracking by blur-driven tracker. In *ICCV*, 2011. 3

[70] M. Yang, Y. Wu, and G. Hua. Context-aware visual tracking. *PAMI*, 2009. 3

[71] R. Yao, Q. Shi, C. Shen, Y. Zhang, and A. van den Hengel. Robust tracking with weighted online structured learning. In *ECCV*, 2012. 3

[72] R. Yao, Q. Shi, C. Shen, Y. Zhang, and A. van den Hengel. Part-based visual tracking with online latent structural learning. In *CVPR*, 2013. 3

[73] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *CSUR*, 2006. 1

[74] J. Yoon, D. Kim, and K. Yoon. Visual tracking via adaptive tracker selection with multiple features. In *ECCV*, 2012. 3

[75] K. Zhang, L. Zhang, and M. Yang. Real-time compressive tracking. In *ECCV*, 2012. 3

[76] L. Zhang and L. van der Maaten. Structure preserving object tracking. In *CVPR*, 2013. 3

[77] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Robust visual tracking via structured multi-task sparse learning. *IJCV*, 2013. 3, 6

[78] X. Zhao, D. Gong, and G. Medioni. Tracking using motion patterns for very crowded scenes. In *ECCV*, 2012. 3

[79] B. Zhong, H. Yao, S. Chen, R. Ji, X. Yuan, S. Liu, and W. Gao. Visual tracking via weakly supervised learning from multiple imperfect oracles. In *CVPR*, 2010. 3

[80] W. Zhong, H. Lu, and M. Yang. Robust object tracking via sparsity-based collaborative model. In *CVPR*, 2012. 3

[81] K. Zimmermann, J. Matas, and T. Svoboda. Tracking by an optimal sequence of linear predictors. *PAMI*, 2009. 3