A method for performance diagnosis and evaluation of video trackers

SIViP
DOI 10.1007/s11760-017-1086-7

ORIGINAL PAPER

A method for performance diagnosis and evaluation of video trackers

Tahir Nawaz[1] · **Anna Ellis**[1] · **James Ferryman**[1]

Received: 22 December 2016 / Revised: 19 February 2017 / Accepted: 20 March 2017

Abstract Several measures for evaluating multi-target video trackers exist that generally aim at providing 'end performance.' End performance is important particularly for ranking and comparing trackers. However, for a deeper insight into trackers' performance it would also be desirable to analyze key contributory factors (false positives, false negatives, ID changes) that (implicitly or explicitly) lead to the attainment of a certain end performance. Specifically, this paper proposes a new approach to enable a diagnosis of the performance of multi-target trackers as well as providing a means to determine the end performance to still enable their comparison in a video sequence. Diagnosis involves analyzing probability density functions of false positives, false negatives and ID changes of trackers in a sequence. End performance is obtained in terms of the extracted performance scores related to false positives, false negatives and ID changes. In the experiments, we used four state-of-the-art trackers on challenging real-world public datasets to show the effectiveness of the proposed approach.

Keywords Video tracking · Performance diagnosis · Performance evaluation

This work has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under Grant Agreement No. 312784.

✉ Tahir Nawaz
t.h.nawaz@reading.ac.uk

Anna Ellis
a.l.ellis@reading.ac.uk

James Ferryman
j.m.ferryman@reading.ac.uk

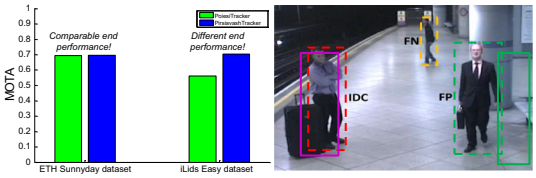
[1] Computational Vision Group, Department of Computer Science, University of Reading, Reading, UK

1 Introduction

Evaluation measures [4–6,9,15,19,22] are important techniques of providing a means to draw performance comparisons among different multi-target tracking algorithms [3,17,18,20,21]. These measures are generally aimed to determine end performance of trackers. *End performance* provides an overall quantification of goodness or badness of trackers' results in the form of a score at frame level [12,19], or sequence level [4,15], without separately analyzing in an explicit manner the key factors (i.e., false positives, false negatives, ID changes [4]) that contribute to the achievement of a certain performance score. Analysis of these contributory factors may indeed be needed in interpreting performance behavior of tracking algorithms against a variety of datasets. It would therefore be desirable from a researcher's perspective to obtain a deeper insight into these factors in addition to the end performance.

Existing measures are broadly made up of composite error counts [4,15], tracking success counts [5,12], tracking failure counts [11,14] and temporal averaging of scores [13,15]; providing tracking quality measurements without giving an explicit insight as why the performance of a tracker is less than perfect. Consider, for example, two cases where a well-known existing measure, Multiple Object Tracking Accuracy (MOTA) [4], provides a comparable performance for a pair of multi-target trackers on a dataset, and ranks one tracker to be better than the other on another dataset (Fig. 1a). Indeed, the measure provides an end performance comparison in the form of a score for each tracker but does not reveal as why those end performances are obtained by trackers. There appears to be a need to also perform a *diagnosis* that is aimed at revealing and dissecting different aspects of tracking performance that could help in understanding why a certain performance is achieved. Such an approach may be used

Published online: 30 March 2017



(a) (b)

Fig. 1 **a** In given examples, MOTA provides a comparable performance for a tracker by Poiesi et al. (PoiesiTracker) [18] and a tracker by Pirsiavash et al. (PirsiavashTracker) [17] on ETH Sunnyday dataset [8], and ranks latter better than former on iLids Easy dataset [7]. **b** Definition of a false positive (FP), a false negative (FN) and an ID change (IDC) in a *frame*. Ground truth: *dotted bounding box*; tracker's result: *solid bounding box*. *Bounding box color* represents a unique ID

in conjunction with end performance measures to provide a clearer and a more detailed picture of tracker's performance. Next we formally define the problem to further clarify what *diagnosis* means in this paper.

1.1 Problem definition

Let $\mathscr{X}$ be a set of tracks estimated by a multi-target tracker in a video sequence, V: $\mathscr{X} = \{\mathfrak{X}_j\}_{j=1}^{J}$, where J is the total number of estimated tracks. $\mathfrak{X}_j$ is the estimated track for target j: $\mathfrak{X}_j = (X_{k,j})_{k=k^j_{\text{start}}}^{k^j_{\text{end}}}$, where k^j_{start} and k^j_{end} are the first and final frame numbers of $\mathfrak{X}_j$, respectively. $X_{k,j}$ is the estimated state of target j at frame $k : k = 1, \ldots, K$ with K as the total number of frames in V. $X_{k,j} = (x_{k,j}, y_{k,j}, A_{k,j}, l_j)$, where $(x_{k,j}, y_{k,j})$ and $A_{k,j}$ denote at frame k the position and occupied area of target j on image plane, respectively, and l_j defines its ID. $A_{k,j}$ may use rectangular (bounding box) [18], elliptical [10], or contour [1] representations. The number of estimated targets at frame k is denoted as n_k, which are defined as $\{X_{k,1}, \ldots, X_{k,j}, \ldots, X_{k,n_k}\}$. Likewise, the notations for the ground-truth quantities corresponding to $\mathscr{X}$, $\mathfrak{X}_j$, J, $X_{k,j}$, k^j_{start}, k^j_{end}, $x_{k,j}$, $y_{k,j}$, $A_{k,j}$, l_j and n_k are $\bar{\mathscr{X}}$, $\bar{\mathfrak{X}}_i$, I, $\bar{X}_{k,i}$, $\bar{k}^i_{\text{start}}$, $\bar{k}^i_{\text{end}}$, $\bar{x}_{k,i}$, $\bar{y}_{k,i}$, $\bar{A}_{k,i}$, $\bar{l}_i$ and $\bar{n}_k$, respectively.

A typical diagnostic procedure for a system starts as a result of identification of *symptom(s)* that may allude to the deterioration in system's performance. For a multi-target tracking system, deterioration may refer to deviation of $\mathscr{X}$ from $\bar{\mathscr{X}}$ [16], which is computed as a discrepancy between $\mathscr{X}$ and $\bar{\mathscr{X}}$. The deterioration of performance in a system results from the occurence of *fault(s)* in it. In the case of a tracking system, the basic set of faults may include ID change, false positive, and false negative; referring to the error in maintaining a unique target ID, incorrect estimation, and missed estimation at frame k, respectively (see Fig. 1b). Here, we consider these three frame-level faults (ID change, false positive, false negative) as they often implicitly or explicitly form a basis for, or contribute to, estimating existing track-level assessment proposals [4–6,15,19,22] (see details in Sect. 2). A diagnostic procedure involves performing *fault diagnosis*. Therefore, for a tracking system, diagnosis may include analyzing across the frames of a sequence the occurrence of false positives, false negatives, and ID changes, which is expected to dissect and reveal more into the achievement of a certain end performance.

1.2 Contributions

In this paper, we present a new approach that, instead of presenting only the end performance of trackers, is also aimed at diagnosis in terms of providing a more revealing picture of the performance of multi-target trackers. It involves analyzing probability density functions (PDFs), in addition to extracting performance scores for each fault type (false positives, false negatives, ID changes) in a video sequence. Performance scores quantify the per frame concentration and robustness of a tracker to each fault type. We show the usefulness of the proposed method using state-of-the-art trackers on four challenging publicly available datasets.

2 Related work

Measures exist that *implicitly* account for faults (false positives, false negatives, ID changes) in their formulation to provide end tracking performance, with respect to the ground-truth information [15,19]. Optimal Sub-Pattern Assignment (OSPA) metric [19] provides a frame-level target positional evaluation by combining accuracy and cardinality errors. The cardinality error (difference between the number of estimated and ground-truth targets) is the number of unassociated targets at frame k; hence, it encapsulates the information about false positives and false negatives. Inspired from OSPA, Multiple Extended-target Tracking Error (METE) [15] also quantifies frame-level performance by combining accuracy and cardinality errors; taking also into account the information about the occupied target region. Multiple Extended-target Lost-Track ratio (MELT) [15] quantifies the performance at sequence level based on the use of lost-track ratio. Given an associated pair of estimated and ground-truth tracks, lost-track ratio is computed as a ratio of the number of frames with an overlap $(O(\bar{A}_{k,i}, A_{k,j}) = \frac{|\bar{A}_{k,i} \cap A_{k,j}|}{|\bar{A}_{k,i} \cup A_{k,j}|}$, such that $|.|$ is the cardinality of a set) between a pair of $X_{k,j}$ and $\bar{X}_{k,i}$ less than a predefined threshold value, and the number of frames in a ground-truth track i. When $O(\cdot)$ is less than the threshold, it may point toward the presence of a false positive, or a false negative in a frame.

Some measures *explicitly* use information about the fault and combine them to quantify the end performance [2,4,5].

False Alarm Rate (FAR) [2,5], Specificity, Positive Prediction, and False Positive Rate [2] use the number of false positives with other quantities in the evaluation procedure at frame level. Negative Prediction and False Negative Rate [2] use information about the number of false negatives with other quantities in evaluation at frame level. Multiple Object Tracking Accuracy (MOTA) [4] estimates performance by combining information about the number of false positives, false negatives, and ID changes at each frame and normalizing across the sequence.

Measures that quantify performance by *separately* using the information from a specific fault include Normalized ID changes (NIDC) [15], False Positive track matches and False Negative track matches [6], False Alarm Track (FAT) and Track Detection Failure (TDF) [22], Track Fragmentation (TF) [5], and ID Changes (IDC) [22]. NIDC normalizes the number of ID changes by length of the track in which they occur. False Positive track matches and FAT use information about false positives across frames. False Negative track matches and TDF use information about false negatives across frames. TF and IDC count the number of ID changes across frames of an individual track and all tracks, respectively.

As reviewed above, existing measures (OSPA, METE, MELT, MOTA, FAR, Specificity, Positive and Negative Predictions, False Positive and Negative Rates) focus on evaluating end performance of trackers without separately providing an explicit insight into each fault type that could be needed to understand the attainment of a certain end performance. Some measures (NIDC, FAT, TDF, TF, IDC, False Positive and False Negative track matches) do provide a separate evaluation for each fault type; however, counting (or combining) false positives, false negatives, or ID changes would still provide an end performance evaluation that may not enable understanding tracker's performance behavior in terms of its ability to deal with each fault type. In this paper, we address this limitation by proposing an approach that involves dissecting a tracker's performance by separately analyzing the behavior of each fault type, while still enabling the end performance evaluation.

3 Tracking performance diagnosis and evaluation

Without loss of generality, $A_{k,j}$ is considered in the form of a bounding box in which case $X_{k,j}$ can be re-written as: $X_{k,j} = (x_{k,j}, y_{k,j}, w_{k,j}, h_{k,j}, l_j)$, where $w_{k,j}$ and $h_{k,j}$ denote width and height of the bounding box for target j at frame k. The notations for ground-truth quantities corresponding to $w_{k,j}$ and $h_{k,j}$ are $\bar{w}_{k,i}$ and $\bar{h}_{k,i}$, respectively. Given a set of estimated states $\{X_{k,1}, \ldots, X_{k,j}, \ldots, X_{k,n_k}\}$, and a set of ground-truth states $\{\bar{X}_{k,1}, \ldots, \bar{X}_{k,i}, \ldots, \bar{X}_{k,\bar{n}_k}\}$ at frame k, the association between the elements of the two sets is established using Hungarian algorithm by minimizing the overlap cost $\big(1 - O(\cdot)\big)$, where $O(\cdot)$ defines the amount of overlap between a pair of $X_{k,j}$ and $\bar{X}_{k,i}$, as described in Sect. 2, i.e., $O(\bar{A}_{k,i}, A_{k,j}) = \frac{|\bar{A}_{k,i} \cap A_{k,j}|}{|\bar{A}_{k,i} \cup A_{k,j}|}$. FP_k, the false positives, are the number of associated pairs of estimated and ground-truth targets with $O(\cdot) < \tau$ (where τ is a threshold value) plus the number of unassociated estimated targets at frame k. FN_k, the false negatives, are the number of ground-truth targets that are missed by a tracker at frame k. IDC_k, the ID changes, are the number of changed associations corresponding to the ground-truth tracks at frame k. See also Fig. 1b. Next, we describe the proposed method including performance diagnosis (Sect. 3.1) and evaluation (Sect. 3.2), followed by highlighting the advantages of using the proposed method.

3.1 Performance diagnosis

Analyzing the occurence of FP_k, FN_k, and IDC_k at each frame can be cumbersome for longer sequences, and also make it difficult to analyze and compare trackers' performance across different sequences with different lengths. Additionally, as discussed in the Sect. 2, looking solely at the total numbers of false positives, false negatives, or ID changes across a sequence is still an end performance evaluation, and a deeper insight would be desirable for performance diagnosis. Instead, the analysis of the distributions of false positives, false negatives, and ID changes in a sequence is expected to provide a more revealing picture of tracker's performance behavior for a fault type, irrespective of sequence length. Moreover, the analysis of distributions (in normalized form) could enable inferring trends about performance of trackers across different datasets.

We therefore compute probability density functions (PDFs) for false positives, false negatives, and ID changes in a sequence. A PDF is computed as a normalized histogram for a particular fault type; hence, the area under each PDF equals 1, i.e., the sum of bin values on y-axis is equal to 1. We denote PDFs for false positives, false negatives, and ID changes for a sequence as $Pr[0 \leq \mathrm{FP}_k \leq N_{\mathrm{fp}}]$, $Pr[0 \leq \mathrm{FN}_k \leq N_{\mathrm{fn}}]$ and $Pr[0 \leq \mathrm{IDC}_k \leq N_{\mathrm{idc}}]$, respectively, where $N_{\mathrm{fp}} = \max\{\mathrm{FP}_k\}_{k=1}^{K}$, $N_{\mathrm{fn}} = \max\{\mathrm{FN}_k\}_{k=1}^{K}$, and $N_{\mathrm{idc}} = \max\{\mathrm{IDC}_k\}_{k=1}^{K}$. Figure 2 shows PDFs of false positives, false negatives, and ID changes for existing trackers with the datasets used in this study. For example: $Pr[\mathrm{FP}_k = 0]$ is read as a probability in terms of the percentage of frames in which the tracker produces zero false positive; similarly, $Pr[\mathrm{FN}_k = 2]$ refers to a probability in terms of the percentage of frames in which the tracker produces two false negatives; likewise, $Pr[\mathrm{IDC}_k > 2]$ means a probability in terms of the percentage of frames in which the tracker produces more than two ID changes.

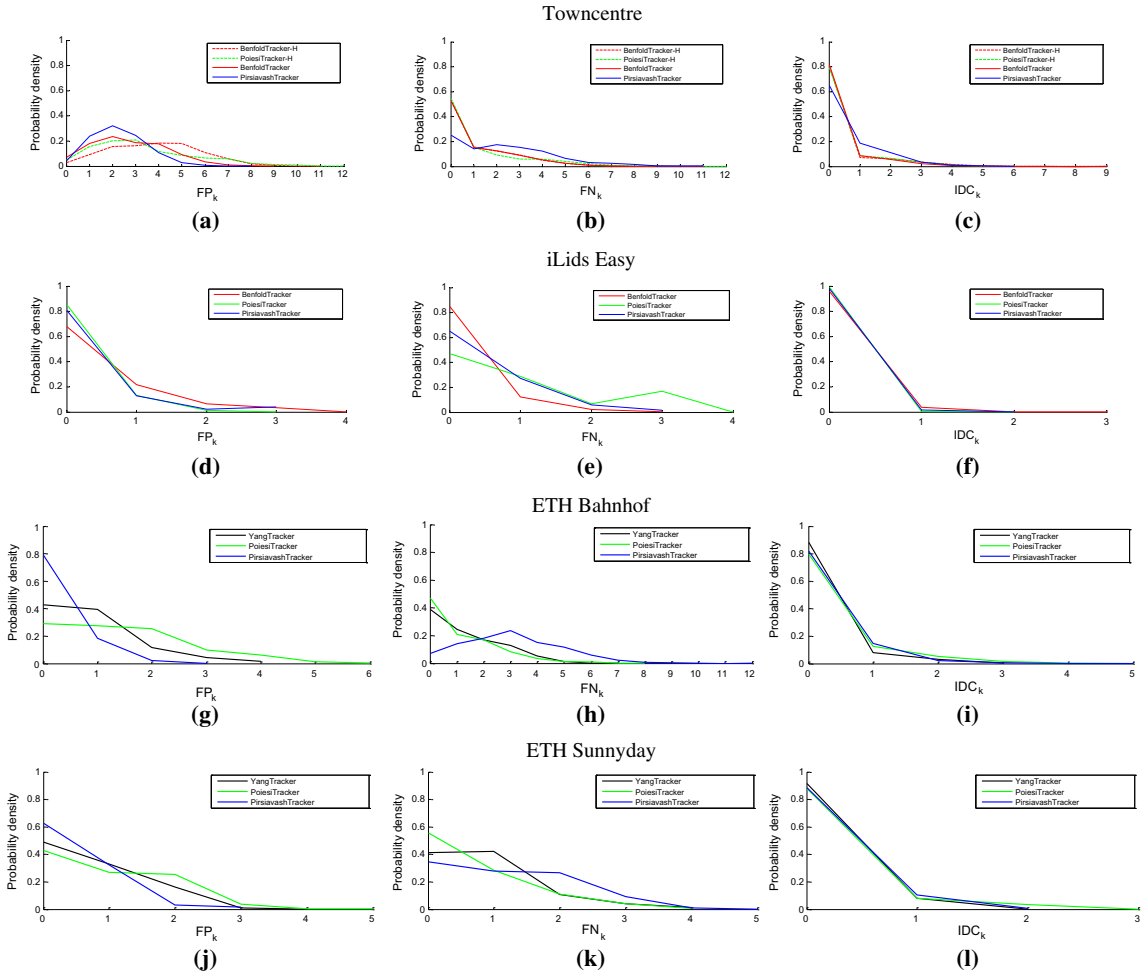


Fig. 2 Performance diagnosis in terms of the probability density functions (PDFs) of trackers on the Towncentre [3], iLids Easy [7], ETH Bahnhof [8], and ETH Sunnyday [8] datasets for fault types: false positives (*first column*); false negatives (*second column*); ID changes (*third column*). On Towncentre, the legends, 'BenfoldTracker-H' and 'PoiesiTracker-H,' refer to the use of these trackers for head tracking

To show the usefulness of analyzing PDFs, consider the case of the Towncentre dataset [3], where PDFs for false positives of a pair of trackers (a tracker from Benfold and Reid (BenfoldTracker) [3] and a tracker from Pirsiavash et al. (PirsiavashTracker) [17]) are generated for full-body tracking in Fig. 2a (PDFs are shown with solid lines). The total number of false positives in the sequence ($\sum_{k=1}^{K} \text{FP}_k$) only reveals that PirsiavashTracker (10,118 false positives) is better than BenfoldTracker (12,162 false positives); however, their corresponding PDFs (Fig. 2a) provide a deeper insight as follows. PDFs reveal that $Pr[\text{FP}_k = 0]$ for BenfoldTracker is higher (better) than $Pr[\text{FP}_k = 0]$ for PirsiavashTracker. This shows an enhanced robustness of BenfoldTracker than PirsiavashTracker because of the presence of more frames where the former did not produce any false positives. The PDFs further reveal that, on the contrary, BenfoldTracker shows a greater tendency than PirsiavashTracker of producing a higher concentration of false positives (i.e., for $\text{FP}_k > 3$) in a frame (see Fig. 2a), i.e., $Pr[\text{FP}_k > 3]$ for BenfoldTracker is higher (worse) than that for PirsiavashTracker. Therefore, analysis of a PDF offers a more detailed and dissected picture of a tracker's performance by revealing its *robustness* and *per frame concentration* for an individual fault type, which is not explicitly available by a simple fault count. To further aid the analysis and to facilitate end performance evaluation comparison of trackers, we next define two performance scores that account for the two aspects above for each fault type.

3.2 Performance evaluation

The first score tells the ability of a tracker to track without producing a fault across a sequence, and is called robustness to a fault type (R): $R_{\text{fp}} = 1 - \frac{K_{\text{fp}}}{K}$; $R_{\text{fn}} = 1 - \frac{K_{\text{fn}}}{K}$; $R_{\text{idc}} = 1 - \frac{K_{\text{idc}}}{K}$; such that K_{fp} is the number of frames containing false positive(s), K_{fn} is the number of frames containing false

negative(s), and K_{idc} is the number of frames containing ID change(s). $R_{\text{fp}} \in [0, 1]$, $R_{\text{fn}} \in [0, 1]$, $R_{\text{idc}} \in [0, 1]$: the higher the value ($R_{\text{fp}}/R_{\text{fn}}/R_{\text{idc}}$), the better the ability.
NB: $R_{\text{fp}}/R_{\text{fn}}/R_{\text{idc}}$ differs in formulation from MOTA [4]: $\text{MOTA} = 1 - \frac{\sum_k^K (c_1\text{FP}_k + c_2\text{FN}_k + c_3\text{IDC}_k)}{\sum_{k=1}^K \tilde{n}_k}$. If $c_1 = 1, c_2 = 0, c_3 = 0$, or $c_1 = 0, c_2 = 1, c_3 = 0$, or $c_1 = 0, c_2 = 0, c_3 = 1$, MOTA reduces to providing the performance separately in terms of false positives or false negatives or ID changes, respectively. Unlike MOTA that, in 'reduced' form, uses information about the *number* of false positives, false negatives, or ID changes, R_{fp}, R_{fn} and R_{idc} instead use information about the *number of frames* having false positive(s), false negative(s), and ID change(s), respectively, to provide robustness.

The second score tells the tendency of a tracker to produce a fault type per frame, and is called per frame concentration of a fault type (PFC): $\text{PFC}_{\text{fp}} = \frac{1}{K}\sum_{k=1}^K \text{FP}_k$, $\text{PFC}_{\text{fn}} = \frac{1}{K}\sum_{k=1}^K \text{FN}_k$, and $\text{PFC}_{\text{idc}} = \frac{1}{K}\sum_{k=1}^K \text{IDC}_k$. $\text{PFC}_{\text{fp}} \geq 0$, $\text{PFC}_{\text{fn}} \geq 0$, and $\text{PFC}_{\text{idc}} \geq 0$: the lower the value ($\text{PFC}_{\text{fp}}/\text{PFC}_{\text{fn}}/\text{PFC}_{\text{idc}}$), the lower the tendency of producing a fault type per frame.

NB: PFC_{idc} differs from NIDC [15], such that the latter penalizes the number of ID changes by length of the track in which they occur, whereas the former quantifies per frame concentration by averaging the number of ID changes across the whole sequence. Likewise, PFC_{fp} and PFC_{fn} differ from MELT [15] that encapsulates lost-track ratio information (as explained in Sect. 2). The lost-track value could indeed reflect the number of frames having false positives and/or false negatives in a track. Unlike MELT, $\text{PFC}_{\text{fp}}(\text{PFC}_{\text{fn}})$ quantifies per frame concentration by averaging the number of false positives(false negatives) across the whole sequence.

3.3 Advantages

This section shows the advantages of using the proposed method over the widely used measure, MOTA. To this end, for clarity, we plot in Fig. 3 the numerator term of MOTA (that we here refer to as MOTA_k: $\text{MOTA}_k = \sum_k^K (c_1\text{FP}_k + c_2\text{FN}_k + c_3\text{IDC}_k)$) for BenfoldTracker [3] on a segment of the Towncentre dataset. MOTA_k combines contributions of FP_k, FN_k, and IDC_k at frame k. Indeed the same value of MOTA_k could be caused by different combinations of FP_k, FN_k, and IDC_k; for example, $\text{MOTA}_k = 8$ at $k = 1348$ and $k = 1393$, although values of FP_k, FN_k, and IDC_k are different in these frames (see Fig. 3). Therefore, MOTA alone might not be revealing enough, as it does not provide an explicit insight into the individual fault types (false positives, false negatives, ID changes) that could be beneficial for a deeper understanding of performance. Differently, the proposed method enables a separate analysis of the behavior of individual fault types for a tracker in terms of respective PDFs (Sect. 3.1), as well as its end performance in terms of extracted robustness (R) and per frame concentration (PFC) scores for each fault type (Sect. 3.2).

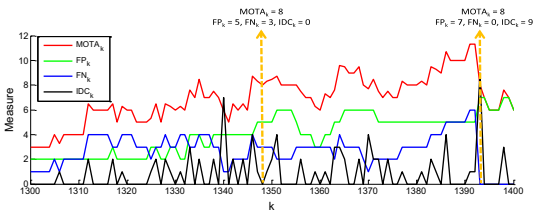


Fig. 3 The numerator term of MOTA (here referred to as MOTA_k) is plotted across a segment of the Towncentre dataset for BenfoldTracker. Additionally, FP_k, FN_k and IDC_k are also plotted alongside

4 Experimental validation

This section demonstrates the usefulness of the proposed method using state-of-the-art trackers on real-world publicly available datasets. Section 4.1 describes the setup including trackers and datasets, followed by the performance analysis of trackers using the proposed method and (for comparison) existing measures in Sect. 4.2, and a discussion in Sect. 4.3.

4.1 Trackers and datasets

Table 1 provides a summary of trackers and datasets used in the experiments. We used available ground truth generated for every frame of the sequences. We used trackers from Pirsiavash et al. (PirsiavashTracker) [17], Yang and Nevatia (YangTracker) [21], Benfold and Reid (BenfoldTracker) [3], and Poiesi *et al.* (PoiesiTracker) [18]. The parameters of trackers are the same as in the original papers. We use head and full-body tracks in experiments. Moreover, we chose four challenging datasets: Towncentre [3], iLids Easy [7], ETH Bahnhof [8], and ETH Sunnyday [8]. Towncentre and iLids Easy are recorded from a static camera, whereas ETH Bahnhof and Sunnyday involves a moving camera. On Towncenter, trackers are tested for head tracking (BenfoldTracker-H, PoiesiTracker-H) and full-body tracking (BenfoldTracker, PirsiavashTracker); on iLids Easy, trackers are used for full-body tracking (BenfoldTracker, PoiesiTracker, PirsiavashTracker); and on ETH Bahnhof and Sunnyday, trackers are tested for full-body tracking (YangTracker, PoiesiTracker, PirsiavashTracker). We use $\tau = 0.25$ for head tracking, and $\tau = 0.5$ for full-body tracking [3]. Figure 2 shows PDFs of trackers for each fault type on all datasets. Table 2 presents performance scores (PFC_{fp}, PFC_{fn}, PFC_{idc}, R_{fp}, R_{fn}, R_{idc}) and existing measures (MOTA, mean METE, MELT) for trackers on all datasets, as well as their number of false positives, false negatives, and ID changes to aid analysis.

Table 1 Summary of datasets

Dataset	K	Frame size	I	Challenges	Trackers	TT
Towncentre	4491	1080 × 1920	231	Occ, SC, Cr	[3,17,18]	H, P
iLids Easy	5220	576 × 720	17	Occ, SC, IC	[3,17,18]	P
ETH Bahnhof	998	480 × 640	95	Occ, SC, Cr, IC, CM	[17,18,21]	P
ETH Sunnyday	353	480 × 640	30	Occ, SC, Cr, IC, CM	[17,18,21]	P

K, number of frames; I, number of trajectories; TT, target type(s) under consideration; H, 'head' target; P, 'full person body' target; Occ, occlusion; SC, scale changes; Cr, crowdedness; IC, illumination changes; CM, camera motion

4.2 Performance analysis of trackers

4.2.1 Towncentre

For full-body tracking, existing measures (MOTA, mean METE, MELT) are primarily limited to only showing that BenfoldTracker outperforms PirsiavashTracker (Table 2). Differently, the proposed method provides a greater understanding of the trackers' performance, by enabling a separate analysis of their per frame concentration and robustness to faults using corresponding PDFs, as well as PFC and R scores, as follows. Based on false positives, PDFs of BenfoldTracker and PirsiavashTracker (shown with solid lines in Fig. 2a) reveal that there are more frames in which PirsiavashTracker produced false positive(s) than BenfoldTracker. This shows that the latter is more robust to false positives than the former, which is also confirmed by a better R_{fp} for BenfoldTracker (Table 2); see also qualitative results where PirsiavashTracker produced false positives, but BenfoldTracker did not (Fig. 4b). At the same time, it can also be noticed that BenfoldTracker shows a greater tendency of producing a higher per frame concentration of false positives than PirsiavashTracker (i.e., for $\text{FP}_k > 3$, Fig. 2a), that is also shown by a better PFC_{fp} for PirsiavashTracker (Table 2). Based on false negatives and ID changes, BenfoldTracker outperforms PirsiavashTracker, as shown in general in their PDFs (Fig. 2b, c), PFC_{fn}, PFC_{idc}, R_{fn}, and R_{idc} (Table 2).

Likewise, for head tracking on this dataset, the values of MOTA, mean METE and MELT simply show that PoiesiTracker-H is better than BenfoldTracker-H (Table 2). The proposed method enables a more detailed analysis of trackers' performance as follows. PDFs of trackers (BenfoldTracker-H, PoiesiTracker-H) are shown with dotted lines in Fig. 2a–c. For false positives (Fig. 2a), the results show that $Pr[\text{FP}_k = 0]$ for PoiesiTracker-H is higher than that for BenfoldTracker-H, showing an enhanced robustness of the former to false positives that is also noticeable by its superior R_{fp} (Table 2). As for per frame concentration, from PDFs, there is no clear winner between BenfoldTracker-H and PoiesiTracker-H for $\text{FP}_k > 0$ (Fig. 2a): BenfoldTracker-H outperforms PoiesiTracker-H for $0 < \text{FP}_k \leq 3$, PoiesiTracker-H is better than BenfoldTracker-H for $3 < \text{FP}_k < 7$, and both trackers generally perform comparably thereafter across their PDFs. Overall, in terms of PFC_{fp}, PoiesiTracker-H, however, shows a superior performance than BenfoldTracker-H (Table 2). For false negatives, PoiesiTracker-H shows more robustness than BenfoldTracker-H, that is noticeable by higher $Pr[\text{FN}_k = 0]$ (Fig. 2b) and R_{fn} (Table 2) of former. As for per frame concentration, BenfoldTracker-H shows a better PFC_{fn} than PoiesiTracker-H (Table 2); this is also reflected by mostly a better performance of the former across their PDFs, i.e., for $\text{FN}_k > 3$ (Fig. 2b). See qualitative results in a sample frame showing several false negatives for PoiesiTracker-H, and a fewer for BenfoldTracker-H (Fig. 4c). For ID changes, overall the results based on PDFs (Fig. 2c) and PFC_{idc}, R_{idc} (Table 2) reveal that PoiesiTracker-H is better based on per frame concentration of ID changes, whereas BenfoldTracker-H is more robust to producing ID changes.

4.2.2 iLids Easy

Based on MOTA, mean METE and MELT, BenfoldTracker is the best followed by PirsiavashTracker and PoiesiTracker (Table 2). The proposed method produces a different ranking based on false positives (PFC_{fp}, R_{fp}) and ID changes (PFC_{idc}, R_{idc}) by ranking PoiesiTracker as the best, followed by PirsiavashTracker and BenfoldTracker (Table 2). Indeed, Fig. 2d, f also shows that PoiesiTracker generally outperforms PirsiavashTracker and BenfoldTracker across their PDFs of false positives and ID changes. On the other hand, based on false negatives, the performance trends of trackers using the proposed method are similar to those produced by MOTA, mean METE and MELT, i.e., BenfoldTracker outperforms PirsiavashTracker and PoiesiTracker across their PDFs (Fig. 2e), as well as based on their PFC_{fn}, R_{fn} (Table 2). The qualitative results show that BenfoldTracker produces more false positives (Fig. 4d, f) and ID changes (Fig. 4d, e) than others, and PoiesiTracker produces more false negatives (Fig. 4d) than others.

Table 2 Performance evaluation of trackers on different datasets using PFC_{fp}, PFC_{fn}, PFC_{idc}, R_{fp}, R_{fn}, and R_{idc}

Tracker	Dataset	PFC_{fp}	PFC_{fn}	PFC_{idc}	R_{fp}	R_{fn}	R_{idc}	FP	FN	IDC	MOTA	Mean METE	MELT
BenfoldTracker-H	Towncentre	3.836	**1.192**	0.426	0.029	0.526	**0.804**	17,228	**5354**	1913	0.678	0.622	0.656
PoiesiTracker-H		**3.368**	1.283	**0.400**	**0.050**	**0.546**	0.796	**15,125**	5762	**1798**	**0.701**	**0.534**	**0.557**
BenfoldTracker		2.708	**1.190**	**0.338**	**0.072**	**0.530**	**0.817**	12,162	**5345**	**1519**	**0.750**	**0.329**	**0.386**
PirsiavashTracker		**2.253**	2.418	0.587	0.046	0.253	0.650	**10,118**	10,861	2637	0.697	0.479	0.542
BenfoldTracker	iLids Easy	0.458	**0.177**	0.043	0.679	**0.850**	0.959	2392	**926**	227	**0.743**	**0.356**	**0.428**
PoiesiTracker		**0.164**	0.939	**0.010**	**0.852**	0.472	**0.992**	**858**	4900	**54**	0.562	0.530	0.541
PirsiavashTracker		0.301	0.440	0.020	0.805	0.652	0.981	1571	2296	105	0.704	0.404	0.523
YangTracker	ETH Bahnhof	0.826	1.264	**0.158**	0.427	0.388	**0.883**	824	1262	**158**	**0.744**	**0.394**	**0.419**
PoiesiTracker		1.426	**1.119**	0.308	0.292	**0.472**	0.796	1423	**1117**	307	0.685	0.443	0.461
PirsiavashTracker		**0.234**	3.065	0.229	**0.792**	0.073	0.818	**234**	3059	229	0.597	0.529	0.568
YangTracker	ETH Sunnyday	0.705	0.821	**0.088**	0.490	0.414	**0.915**	249	290	**31**	**0.711**	0.460	**0.390**
PoiesiTracker		0.940	**0.660**	0.159	0.428	**0.555**	0.881	332	**233**	56	0.694	0.468	0.461
PirsiavashTracker		**0.439**	1.153	0.122	**0.626**	0.346	0.887	**155**	407	43	0.697	**0.444**	0.560

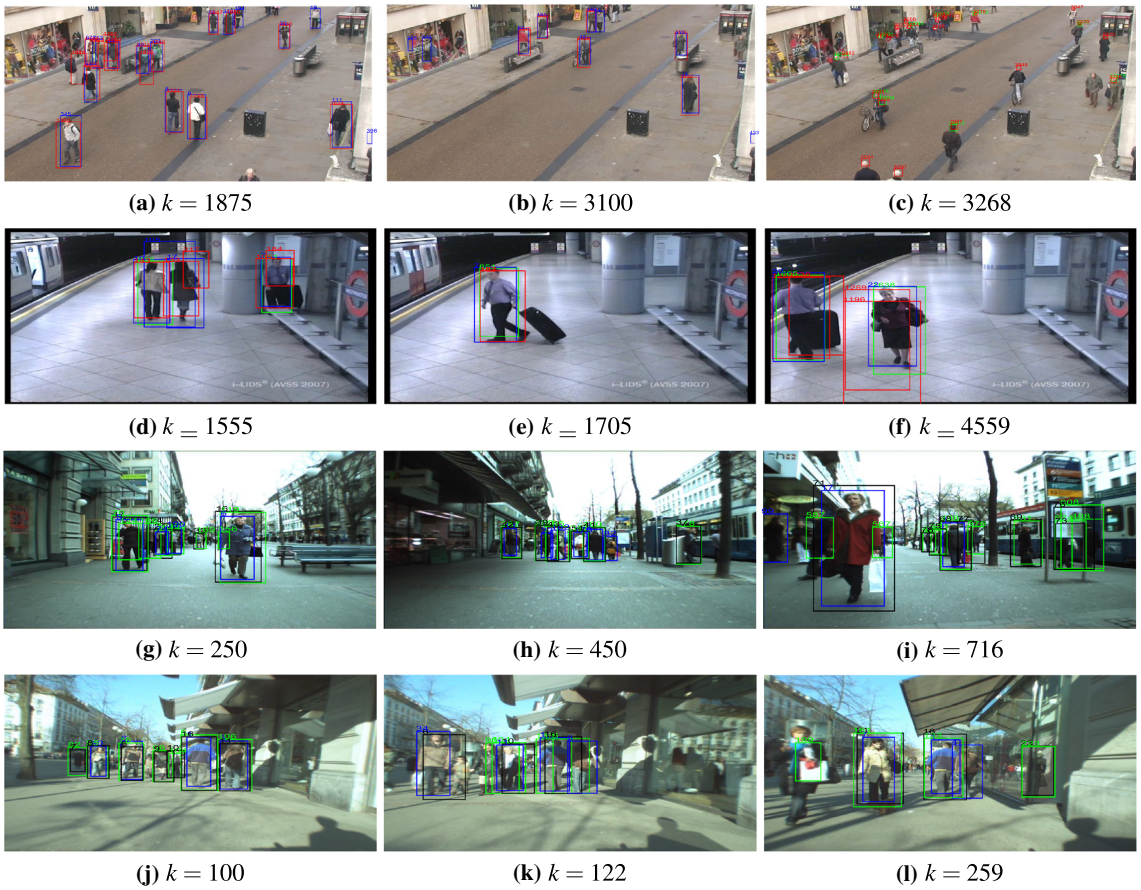
The total number of false positives (FP), false negatives (FN), ID changes (IDC), MOTA, Mean METE and MELT of trackers are also listed. For Towncentre, 'BenfoldTracker-H' and 'PoiesiTracker-H' refer to the use of these trackers for head tracking. On each dataset, the best tracking scores are shown in bold

4.2.3 ETH Bahnhof and Sunnyday

On ETH Bahnhof, existing measures (MOTA, mean METE, MELT) consider YangTracker as the best, followed by PoiesiTracker and PirsiavashTracker. The proposed method provides additional information and useful insights into the performance based on different fault types as follows. Based on false positives, PirsiavashTracker is found to be the most robust and shows the least per frame concentration, followed by YangTracker and PoiesiTracker, as confirmed by their PFC_{fp} and R_{fp} scores (Table 2), and across their PDFs (Fig. 2g). For example, Fig. 4g, i shows the qualitative tracking results with more false positives for PoiesiTracker than others. Based on false negatives, PoiesiTracker is the best, followed by YangTracker and PirsiavashTracker; this is reflected by their PFC_{fn} and R_{fn} (Table 2), as well as in general across their PDFs (Fig. 2h). See, for example, qualitative results in a sample frame with more false negatives for PrisiavashTracker than others (Fig. 4i). Based on ID changes, YangTracker shows an increased robustness and a better per frame concentration, as compared to PirsiavashTracker and PoiesiTracker, as confirmed by their PFC_{idc} and R_{idc} scores (Table 2), and generally across their PDFs (Fig. 2i). On ETH Sunnyday, the trends and rankings of trackers (YangTracker, PirsiavashTracker, PoiesiTracker) based on PFC and R scores (Table 2), and PDFs (Fig. 2j–l) for all fault types are interestingly similar to those reported above for ETH Bahnhof. See also qualitative results on ETH Sunnyday in Fig. 4j–l.

4.3 Discussion

The proposed method could be used to provide formative feedback that could help researchers in addressing shortcomings in tracking algorithms. In fact, the analysis based on false positives could enable analyzing the impact on tracking performance originating from the detection stage. For example, BenfoldTracker has generally shown inferior PFC_{fp} and R_{fp} on Towncentre and iLids Easy than others, and PoiesiTracker has shown inferior PFC_{fp} and R_{fp} on ETH Bahnhof and Sunnyday than others. Indeed, on ETH Bahnhof and Sunnyday, the possible reason of the worst PFC_{fp} and R_{fp} scores of PoiesiTracker is that its person detector has a limited ability to deal with varying illumination conditions in these datasets [15]. Therefore, the results show a particular need of improvement at the detection stage of BenfoldTracker and PoiesiTracker. Similarly, inferior scores related to false negatives (i.e., PFC_{fn}, R_{fn}) can point toward improving the detection stage, and/or inability to temporally link small tracks ('tracklets') in an effective manner. For example, PirsiavashTracker has shown inferior PFC_{fn} and R_{fn} on most datasets (Towncentre, ETH Bahnhof, ETH Sunnyday) than others. This is likely due to the absence



(a) $k = 1875$ **(b)** $k = 3100$ **(c)** $k = 3268$

(d) $k = 1555$ **(e)** $k = 1705$ **(f)** $k = 4559$

(g) $k = 250$ **(h)** $k = 450$ **(i)** $k = 716$

(j) $k = 100$ **(k)** $k = 122$ **(l)** $k = 259$

Fig. 4 Qualitative results of trackers on the Towncentre [3] (**a–c**), iLids Easy [7] (**d–f**), ETH Bahnhof [8] (**g–i**), and ETH Sunnyday [8] (**j–l**) datasets. Key—*Red* BenfoldTracker, *blue* PirsiavashTracker, *green* PoiesiTracker, *black* YangTracker

of an effective dedicated strategy to link tracklets in PirsiavashTracker [17], which other trackers (e.g., PoiesiTracker, YangTracker) possess. In fact, it is due to this limited ability that PirsiavashTracker also reported the highest cardinality error (that can be caused by false negatives) on these datasets in an earlier study [15]. Likewise, the analysis based on ID changes provide a formative feedback vis-a-vis the tracking stage. For instance, YangTracker consistently shows better PFC_{idc} and R_{idc} than PoiesiTracker and PirsiavashTracker on ETH Bahnhof and Sunnyday, which also confirms the conclusions of [15] that YangTracker outperforms other trackers in terms of ID changes on the same datasets. Indeed, this is because YangTracker uses an effective ID management strategy, employing motion and appearance affinities to avoid confusion between IDs of targets that are close to each other [21]. Hence, a researcher could pay more attention on improving the ID management in PoiesiTracker and PirsiavashTracker.

5 Conclusions

We presented a new method that, instead of just providing usual end performance evaluation, also aims at performance diagnosis of a multi-target tracker in a video sequence. Existing tracking evaluation proposals generally focus only on end performance assessment that is important for drawing performance comparison. Instead, the proposed approach enables a more detailed performance analysis using probability density functions (PDFs) of key frame-level faults that a tracker can make (i.e., false positives, false negatives, ID changes). To complement this analysis, the extracted performance scores further offer a separate evaluation in terms of per frame concentration and robustness of trackers for each fault type. We used real-world publicly available datasets using state-of-the-art trackers to validate the proposed method by showing its effectiveness over existing proposals, and its use in identifying algorithmic shortcomings of trackers.

While the proposed method accounts for multi-target tracking, it could still be partly suitable for single-target trackers; however, for single-target tracking, ID change is generally not an issue and, hence, could be ignored. Additionally, the proposed method could be applied for any target type, provided the target model contains the position and occupied area (on a 2D image plane) as parameters. Moreover, the proposed method is based on analyzing frame-level

faults. An explicit inclusion of track-level faults could also be of interest and is left to future work.

References

1. Allili, M.S., Ziou, D.: Active contours for video object tracking using region, boundary and shape information. SIVP **1**(2), 101–117 (2007)
2. Bashir, F., Porikli, F.: Performance evaluation of object detection and tracking systems. In: Proceedings of IEEE PETS Workshop (2006)
3. Benfold, B., Reid, I.: Stable multi-target tracking in real-time surveillance video. In: Proceedings of IEEE Conference on CVPR, Colorado Springs, USA (2011)
4. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: the clear mot metrics. JIVP **2008**, 1–10 (2008)
5. Black, J., Ellis, T., Rosin, P.: A novel method for video tracking performance evaluation. In: Proceedings of IEEE PETS Workshop (2003)
6. Brown, L.M., Senior, A.W., Tian, Y.L., Connell, J., Hampapur, A., Shu, C.F., Merkl, H., Lu, M.: Performance evaluation of surveillance systems under varying conditions. In: Proceedings of IEEE PETS Workshop (2005)
7. http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html. Accessed Jan 2016
8. http://www.vision.ee.ethz.ch/~aess/iccv2007/. Accessed Jan 2016
9. Kasturi, R., Goldgof, D., Soundararajan, P., Manohar, V., Garofolo, J., Bowers, R., Boonstra, M., Korzhova, V., Zhang, J.: Framework for performance evaluation of face, text, and vehicle detection and tracking in video: data, metrics, and protocol. IEEE Trans. PAMI **31**(2), 319–336 (2009)
10. Koohzadi, M., Keyvanpour, M.: OTWC: an efficient object-tracking method. SIVP **9**(6), 1235–1247 (2015)
11. Kristan, M., Pflugfelder, R., Leonardis, A., Matas, J., Porikli, F., Cehovin, L., Nebehay, G., Fernandez, G., Vojir, T.: The VOT2013 challenge: overview and additional results. In: CVWW (2014)
12. Li, H., Shen, C., Shi, Q.: Real-time visual tracking using compressive sensing. In: Proceedings of IEEE Conference on CVPR (2011)
13. Nawaz, T., Cavallaro, A.: PFT: a protocol for evaluating video trackers. In: Proceedings of ICIP, Brussels (2011)
14. Nawaz, T., Cavallaro, A.: A protocol for evaluating video trackers under real-world conditions. IEEE Trans. Image Process. **22**(4), 1354–1361 (2013)
15. Nawaz, T., Poiesi, F., Cavallaro, A.: Measures of effective video tracking. IEEE Trans. Image Process. **23**(1), 376–388 (2014)
16. Nawaz, T.H.: Ground-truth-based trajectory evaluation in videos. Ph.D. thesis, Queen Mary University of London, UK (2014)
17. Pirsiavash, H., Ramanan, D., Fowlkes, C.C.: Globally-optimal greedy algorithms for tracking a variable number of objects. In: Proceedings of IEEE Conference on CVPR, Colorado Springs, USA (2011)
18. Poiesi, F., Mazzon, R., Cavallaro, A.: Multi-target tracking on confidence maps: an application to people tracking. Comput. Vis. Image Underst. **117**(10), 1257–1272 (2013)
19. Ristic, B., Vo, B.N., Clark, D., Vo, B.T.: A metric for performance evaluation of multi-target tracking algorithms. IEEE Trans. Signal Process. **59**(7), 3452–3457 (2011)
20. Wu, H., Liu, N., Luo, X., Su, J., Chen, L.: Real-time background subtraction-based video surveillance of people by integrating local texture patterns. SIVP **8**(4), 665–676 (2014)
21. Yang, B., Nevatia, R.: An online learned CRF model for multi-target tracking. In: Proceedings of IEEE Conference on CVPR, Providence, Rhode Island (2012)
22. Yin, F., Makris, D., Velastin, S.A.: Performance evaluation of object tracking algorithms. In: Proceedings of IEEE PETS Workshop (2007)