A Protocol for Evaluating Video Trackers Under Real-World Conditions

Tahir Nawaz and Andrea Cavallaro

Abstract—The absence of a commonly adopted performance evaluation framework is hampering advances in the design of effective video trackers. In this paper, we present a single-score evaluation measure and a protocol to objectively compare trackers. The proposed measure evaluates tracking accuracy and failure, and combines them for both summative and formative performance assessment. The proposed protocol is composed of a set of trials that evaluate the robustness of trackers on a range of test scenarios representing several real-world conditions. The protocol is validated on a set of sequences with a diversity of targets (hexagonal vehicle, person) and challenges (occlusions, background clutter, pose changes, scale changes) using six state-of-the-art trackers, highlighting their strengths and weaknesses on more than 187,000 frames. The software implementing the protocol and the evaluation results are made available online and new results can be included, thus facilitating the comparison of trackers.

Index Terms—Performance evaluation, video trackers, evaluation measure, protocol, trials.

I. INTRODUCTION

Unlike other areas of image processing and computer vision such as disparity estimation [1], optical flow computation [2] and video coding [3] that consistently use commonly accepted evaluation procedures, video tracking still lacks a standard way to evaluate and compare algorithmic performance.

Although a number of efforts have been made toward performance evaluation of trackers in the form of evaluation campaigns (ETISEO, CLEAR, PETS, i-LIDS, CAVIAR) and small-scale evaluation frameworks ([4], [5], [6], [7], [8]), the performance of trackers is still tested using different evaluation criteria and varying datasets, thus hindering an effective evaluation and comparison. Moreover, because of the complexity of the evaluation task, many performance criteria contain multiple measures [4], [6], [7], which are difficult to combine in order to rank various algorithms. A single-score evaluation criterion that can comprehensively encapsulate the overall tracking performance would be desirable to simplify the performance comparison task.

Performance evaluation may involve the computation of the discrepancy between the estimated and the ground-truth position and size of the target [4], [6], [9], [10]. The discrepancy is computed based on a distance-based criterion [9], [10], [11] or an overlap-based criterion [4], [7], [12]. Distance-based criteria use the concept of distance minimization between estimation and ground truth to evaluate performance. Naive distance-based evaluation may not include target size variations in the evaluation procedure [8] and may not effectively reflect instances of tracking failure [13] that refers to the case of no-overlap between estimated and ground-truth states. Overlap-based criteria compute the amount of overlap between estimation and ground truth. Overlap-based evaluation mostly takes into account target size variations (with some exceptions [4], [5], [6]) and can therefore detect instances of tracking failure. However, existing overlap-based criteria [5], [7], [8], [14] use hard thresholds or fixed parameters that restrict their use to application-specific tracking performance assessment.

In this paper, we propose a threshold-independent overlap-based criterion that summarizes tracking performance based on a new evaluation measure, which takes into account target size variations. The proposed measure quantifies how accurately and how long a target is tracked across a sequence. Moreover, we propose a protocol with a comprehensive set of trials that evaluate trackers on a wide range of test scenarios representing several real-world operational conditions. The trials quantify the robustness of a tracker to noisy inputs, processing and communication delays, video compression and varying scene conditions such as illumination changes. To the best of our knowledge, this is the first initiative that enables evaluating the robustness of the performance of tracking algorithms under such a wide variety of real-world conditions. The resulting performance evaluation tool is made available online as an open source software.

The organization of the paper is as follows. The proposed evaluation criterion and the protocol are discussed in Sec. II and Sec. III, respectively. This is followed by the experimental validation in Sec. IV. Section V concludes the paper.

II. COMBINED TRACKING PERFORMANCE SCORE

Let an estimated trajectory \( R = \{ (x_k, y_k, A_k) \}_{k=1}^{N_R} \), (1)

where \( (x_k, y_k) \) is the estimated target position (e.g. its centroid), \( A_k \) is the information about the estimated target area in the \( k \)-th frame and \( N_R \) is the total number of frames for which the tracker generated an output. Let the corresponding ground-truth trajectory be:

\[
G = \{ (\hat{x}_k, \hat{y}_k, \hat{A}_k) \}_{k=1}^{N_G},
\]

(2)

where \( (\hat{x}_k, \hat{y}_k) \) is the ground-truth target position, \( \hat{A}_k \) is the ground-truth target area in the \( k \)-th frame and \( N_G \) is the total number of frames in which the target exists. \( A_k \) and \( \hat{A}_k \) may be in the form of a bounding box, a bounding ellipse or a bounding contour. Without loss of generality, let \( A_k \) and \( \hat{A}_k \) be bounding boxes that define the width and the height of the target.

We firstly compute the amount of overlap, \( O_k \), across \( R \) as follows [15]:

\[
O_k = \frac{|TP_k|}{|TP_k| + |FP_k| + |FN_k|},
\]

(3)

where \( O_k \in [0, 1] \) and \( |\cdot| \) represents the cardinality of a set. \( TP_k \), \( FP_k \) and \( FN_k \) are the sets of true positive (correctly estimated), false positive (incorrectly estimated) and false negative (missed) pixels of a target at frame \( k \). Note that \( O_k = 0 \) if the tracker does not produce a bounding box when the target is present or if a bounding box is produced when no target is present.

The tracking accuracy quantifies the extent to which the estimated trajectory overlaps the ground-truth trajectory, considering only frames with \( O_k \neq 0 \) (Fig. 1(a)) and is computed as [15]:

\[
\hat{\lambda} = \frac{\hat{N}_f}{\hat{N}},
\]

(4)

where \( \hat{N}_f = |\hat{F}| \) and \( \hat{F} = \{ f_k : O_k \in (0, \hat{\tau}), \hat{\tau} \in (0, 1), \forall k \} \); and \( \hat{N} = |\hat{F}| \), with \( \hat{F} = \{ f_k : O_k \neq 0, \forall k \} \), is the number of frames with \( O_k \neq 0 \).

Computing \( \hat{\lambda} \) for a fixed value of \( \hat{\tau} \) necessitates an application-dependent decision, since different values of \( \hat{\tau} \) may be appropriate for different tracking tasks. To overcome this limitation, instead of computing \( \hat{\lambda} \) for a fixed value of \( \hat{\tau} \), we accumulate its value over

1http://www.eecs.qmul.ac.uk/~andrea/pft2
the full range of $\hat{\tau}$ values. In particular, we use an increment of $\Delta \hat{\tau} = 0.01$ to obtain $\hat{\lambda}(\hat{\tau})$ and therefore, the score that quantifies tracking accuracy across the sequence, $\Omega$, is computed as

$$\Omega = \Delta \hat{\tau} \sum_{\tau \in [0,1]} \hat{\lambda}(\hat{\tau}), \quad (5)$$

where $\Omega \in [0,1]$. The smaller $\Omega$, the higher the tracking accuracy. $\Omega$ can be regarded as an approximation of the area under the curve of $\hat{\lambda}(\hat{\tau})$.

Tracking failures correspond to instances of target loss. The tracking failure score, $\lambda_0$ ($\lambda_0 \in [0,1]$), is defined as

$$\lambda_0 = \frac{N^0}{N}, \quad (6)$$

where $N^0 = |F^0|$, with $F^0 = \{f_k : O_k = 0, \forall k\}$, and $N = |F|$, with $F = \{f_k : \forall k\}$. The smaller $\lambda_0$, the smaller the tracking failure score.

We combine the information on tracking accuracy and tracking failure in a single score to facilitate performance ranking. The proposed Combined Tracking Performance Score, CoTPS ($\text{CoTPS} \in [0,1]$), is computed as follows:

$$\text{CoTPS} = \beta \Omega + (1 - \beta) \lambda_0, \quad (7)$$

where $\beta$ is a penalty, with $\beta \in [0,1]$. The smaller CoTPS, the better the tracking performance. Figure 1(b) plots $O_k$ for two tracking results whose comparison is shown using CoTPS.

Note that a preset value of $\beta$ may lead to incorrect performance evaluation (see Fig. 1(c)). $\beta$ is computed adaptively:

$$\beta = \frac{N - \hat{N}}{N}, \quad (8)$$

where $\hat{N}$ is the number of frames in which the tracker has partially or completely tracked the target ($O_k > 0$), thus restricting the inclusion of any extra influence of $\Omega$ in the computation of $\text{CoTPS}$. Similarly, $(1 - \beta)$ applied to $\lambda_0$ is proportional to $(N - N)$, i.e. the number of frames in which the tracker has failed ($O_k = 0$), which are also the same frames used in the estimation of $\lambda_0$, thus restricting the inclusion of any extra influence of $\lambda_0$ in the computation of $\text{CoTPS}$.

Let us consider the result of the Mean-Shift tracker (MS) [16] in Fig. 1(b). In this example, a penalty of $\beta = 0.328$ (computed using Eq. (8)) is applied to $\Omega$ since the tracker is successful ($O_k > 0$) in 32.8% frames ($\hat{N} = 79$ and $N = 241$). Similarly, a penalty of $(1 - \beta) = 0.672$ is applied to $\lambda_0$ since the tracker has failed ($O_k = 0$) in 67.2% frames ($N - \hat{N} = 162$ and $N = 241$). The adaptive computation of $\beta$ allows us to include accurate contributions of $\Omega$ and $\lambda_0$ in the estimation of $\text{CoTPS}$.

### III. Evaluation Protocol

Trackers operate under various challenges in real-world applications and therefore these challenges should be considered when evaluating and comparing performance. Tracking challenges include initialization errors caused by a detector; sensor noise; latency due to the transmission of video data over a channel or due to the delayed generation of results by the tracker; changing illumination in the scene; and compression of the video data (Fig. 2). The proposed evaluation protocol enables evaluation under these challenges to quantify the robustness of trackers.

Given a set of $M$ trackers$^2$ $T = \{T_j\}_{j=1}^M$, we aim to evaluate tracker $T_j$ on a set of trials $P = \{P_i\}_{i=1}^n$, where each trial simulates a specific real-world operational condition.

**Trials**: 1, 2, 3 ($P_1, P_2, P_3$) evaluate the robustness of trackers to **initialization errors** possibly introduced by a detector. These errors are simulated by perturbing the position of the initial bounding box in **Trial 1 ($P_1$)**, the size (width and height) of the bounding box in **Trial 2 ($P_2$)**, and both the position and size in **Trial 3 ($P_3$)**. The amount of perturbation is added while ensuring at least an overlap of $\%O$ between the bounding boxes of the original (ground-truth) initialization and the perturbed initialization. The number of perturbed initializations generated on $P_1, P_2$ and $P_3$ are $n_1, n_2$ and $n_3$, respectively.

**Trial 4 ($P_4$)** evaluates robustness to **noisy video data** generated by low-cost sensors. On $P_4$, a set of $n_4$ test sequences are generated by adding to the original sequence a Gaussian noise of a low-quality webcam (Creative webcam VF0330). The estimated standard deviations of its red, green and blue channels are $\sigma_1 = 8.59$, $\sigma_2 = 8.40$ and $\sigma_3 = 11.96$, respectively.

**Trial 5 ($P_5$)** evaluates robustness to **latency** due to transmission over a channel or due to the delayed generation of the results by the tracker. On $P_5$, the protocol generates a set of $n_5$ test sequences by periodically dropping $m - 1$ frames from the original sequence.

**Trial 6 ($P_6$)** evaluates robustness to changing illumination in the scene. On $P_6$, a set of $n_6$ test sequences are generated by synthetically...
increasing (+\(\Delta L\)) or decreasing (−\(\Delta L\)) illumination over time (in the original sequence) with saturation by adding (subtracting) \(\Delta L = 0, 1, \ldots, L\) to (from) the pixel values of frames \(k = 1, 2, \ldots, K\), respectively. If the number of frames in the sequence is \(K > (L+1)\), a value of \(\Delta L = L\) is maintained for the remaining frames.

Trials 7, 8 (\(P_7, P_8\)) evaluates robustness to bandwidth reduction of the video data. On \(P_7\), test sequences are generated by gradually increasing the compression ratio of the original sequence. We chose Motion JPEG compression because of its suitability for video tracking applications. In Motion JPEG, the extent of compression ratio depends on a quality parameter \(C\). The higher \(C\), the better the visual quality and the lower the compression ratio, where \(C \in [0, 100]\). To ensure evaluation under strong compression ratios, a set of \(n_T\) test sequences are generated on this trial by gradually reducing \(C\). On \(P_8\), a set of \(n_T\) test sequences are generated by reducing the resolution of the original video frames by \(\rho\%\).

On each trial \(P_i\), \(T_j\) is tested with the original (ground truth) initialization \(I_1\) and the original video sequence \(V\) which contains a target \(H\), where \(H = \{H_1\}_{i=1}\) is a set of targets. To study its variation in performance, each tracker \(T_j\) is tested with the initialization \(I_{1,i}\) and test sequence \(V_{i,t}\) which are generated on trial \(P_{j}\) by modifying \(I_1\) or \(V_1\) in a pre-defined manner such that the applied modification simulates a specific real-world scenario: \(I_{1,i} = P_i(I_1)\), and \(V_{i,t} = P_i(V_1)\).

Let \(R_{1,i}^t\) be the trajectory of target \(H\) estimated by testing tracker \(T_j\) on trial \(P_{j}\) with \(V_{i,t}\) and \(I_{1,i}\): \(R_{1,i}^t = T_j(V_{i,t}, I_{1,i})\). The performance of tracker \(T_j\) is computed by evaluating the estimated trajectory \(R_{1,i}^t\) of the target with respect to its ground-truth trajectory \(G_t\) using the proposed evaluation criterion (Sec. II) thus obtaining the performance score: \(CoTPS_{1,i}^t = \Psi(R_{1,i}^t, G_t)\), where \(\Psi(\cdot)\) represents the procedure involved in the evaluation criterion (see Sec. II). Based on \(CoTPS_{1,i}^t\), we compare the performance of the trackers under consideration.

Table I summarizes the trials and the values of the corresponding parameters (these parameters accomplished statistically significant results, as discussed at the end of Sec. IV). Using the proposed protocol, a tracker is tested on each sequence of the dataset in original form and in its 24 variations generated on different trials. Each tracker is tested with 60 perturbations of the initialization on the original video sequence. A deterministic tracker is therefore run 85 times, whereas a probabilistic tracker is run \(85 \times n\) times for its evaluation using the protocol, where \(n\) denotes the number of runs for each test of a trial.

We selected the dataset by taking into account the diversity of targets and test scenarios, their availability and the challenges involved. The dataset contains three target classes, namely head, vehicle and person. The sequences are chosen from PETS, CAVIAR, AVSS and SPEVI datasets, which are publicly available. A range of tracking challenges are present in the dataset such as partial/total occlusions, pose changes, background clutter and small/large scale changes. The selected sequences include two head targets \(H_1\) and \(H_2\) from SPEVI [18], two vehicle targets \(H_3\) and \(H_4\) from PETS 2000 [19] and AVSS 2007 [20], respectively, and two person targets \(H_5\) and \(H_6\) from PETS 2010 [21] and CAVIAR [22], respectively.

Table II summarizes the dataset in terms of initial target size \((C_{ini}^h)\), maximum and minimum sizes of the visible part of target \((C_{max}^h, C_{min}^h)\), number of frames \((K)\), frame size \((C'f)\) and the challenges present in the sequence.

### IV. Experimental analysis and validation

We demonstrate the effectiveness of the proposed score and protocol by evaluating and comparing six state-of-the-art trackers. The selected trackers can be divided into two categories: standard trackers and boosting-based trackers. Standard trackers are Mean Shift (MS) [16], the fragments-based tracker (FragTrack) [23], and Particle Filter (PF) [24]. Boosting-based trackers are Boost [17], the semi-supervised on-line boosting-based tracker (SemiBoost) [25], and beyond semi-supervised boost (BeyondSemiBoost) [26]. The parameters of all trackers are fixed throughout the experiments.

We discuss the performance comparison on each trial \(P_i\), on each target \(H_j\) and on each target class, and verify the statistical significance of the obtained results. Each tracker is tested on a total of 187144 frames. PF, being a probabilistic tracker, is run \(n = 10\) times on each test of each trial and the mean value of its \(CoTPS\) on the \(n\) runs is considered. The choice of \(n = 10\) is made based on the analysis of the behavior of the mean \(CoTPS\) of PF computed by running it with each of the six targets \(H_1, H_2, \ldots, H_6\) for a variation of \(n\); the fluctuation in the mean \(CoTPS\) tends to decrease after \(n = 5\) and becomes stable for \(n \rightarrow 10\).
of tracking errors over time. Among the remaining trackers, Boost shows closer performance to MS on these trials (Fig. 3(a)) followed by PF and the other two boosting-based trackers. Additionally, in terms of robustness to initialization errors, MS and PF outperform the boosting-based trackers and FragTrack (smaller $d_C$ of MS and PF in Fig. 3(b)). The reason of the increased sensitivity of the boosting-based trackers is that any perturbation to initialization may affect their learning process. The performance of the BeyondSemiBoost decreased the most with noisy video data (highest $\mu_C$ on $P_3$). PF is more robust to deal with noise (smaller $d_C$ than the remaining trackers). On $P_5$, Boost shows the best performance (smallest $\mu_C$) followed by MS, BeyondSemiBoost, SemiBoost, PF and FragTrack, respectively. Frame dropping may result in abrupt movements of target: standard trackers are more robust to increasing levels of frame dropping than boosting-based trackers. PF is the most robust tracker (smallest $d_C$ on $P_8$) and BeyondSemiBoost is the least robust. Boost has the best performance under changing illumination (smallest $\mu_C$ on $P_9$), because of its ability to adapt to appearance changes [17]. PF is the most robust with changing illumination (smallest $d_C$ on $P_8$). The $d_C$ of MS is the closest to that of PF. An interesting observation regarding the performance of boosting-based trackers on $P_9$ is that both $\mu_C$ and $d_C$ increase from Boost to SemiBoost and from SemiBoost to Beyond-SemiBoost, which suggests that the evolution of the boosting-based trackers has resulted in a decreased ability to cope with changing illumination. The results also highlight the sensitivity of FragTrack to deal with changing illumination (the highest $\mu_C$ and the highest $d_C$ on $P_8$). MS has the best performance on $P_7$ both in terms of $\mu_C$ and the robustness ($d_C$) to cope with the compressed video data. In terms of $\mu_C$, Boost shows the closest performance to MS followed by PF, SemiBoost, BeyondSemiBoost and FragTrack, respectively; and in terms of $d_C$, PF has the closest performance to MS followed by Boost, FragTrack, BeyondSemiBoost and SemiBoost, respectively. Finally, on $P_8$, MS again outperforms other trackers in terms of $\mu_C$ and the robustness ($d_C$) to deal with resolution changes. The performance of Boost is closer to that of MS in terms of $\mu_C$ as compared to remaining trackers. Moreover, the performance of PF is closer to that of MS in terms of $d_C$ than the other trackers. Based on the performance analysis on $P_7$ and $P_8$, the performance of MS is the least affected by compression or reduction in the resolution.

**A. Trial-wise comparison**

Figure 3 shows the mean CoTPS ($\mu_C$) of trackers on each $P_i$ computed with all targets and their robustness in terms of the dispersion of their CoTPS ($d_C$) computed with all targets as $d_C = \text{CoTPS}_{\text{max}} - \text{CoTPS}_{\text{min}}$, where $\text{CoTPS}_{\text{max}}$ and $\text{CoTPS}_{\text{min}}$ are the maximum and the minimum values of CoTPS of a tracker on a trial, respectively.

MS consistently tracks more accurately in the presence of initialization errors than other trackers (smaller $\mu_C$ on $P_1, P_2, P_3$ in Fig. 3(a)); whereas FragTrack shows inferior performance than other trackers in the presence of initialization errors. In fact, unlike MS, FragTrack uses a fragment-based representation of the target [23] and a perturbation in its initialization can lead to the inclusion of non-target patches in the target model thus resulting in the accumulation

of tracking errors over time. Among the remaining trackers, Boost shows closer performance to MS on these trials (Fig. 3(a)) followed by PF and the other two boosting-based trackers. Additionally, in terms of robustness to initialization errors, MS and PF outperform the boosting-based trackers and FragTrack (smaller $d_C$ of MS and PF in Fig. 3(b)). The reason of the increased sensitivity of the boosting-based trackers is that any perturbation to initialization may affect their learning process. The performance of the BeyondSemiBoost decreased the most with noisy video data (highest $\mu_C$ on $P_3$). PF is more robust to deal with noise (smaller $d_C$ than the remaining trackers). On $P_5$, Boost shows the best performance (smallest $\mu_C$) followed by MS, BeyondSemiBoost, SemiBoost, PF and FragTrack, respectively. Frame dropping may result in abrupt movements of target: standard trackers are more robust to increasing levels of frame dropping than boosting-based trackers. PF is the most robust tracker (smallest $d_C$ on $P_8$) and BeyondSemiBoost is the least robust. Boost has the best performance under changing illumination (smallest $\mu_C$ on $P_9$), because of its ability to adapt to appearance changes [17]. PF is the most robust with changing illumination (smallest $d_C$ on $P_8$). The $d_C$ of MS is the closest to that of PF. An interesting observation regarding the performance of boosting-based trackers on $P_9$ is that both $\mu_C$ and $d_C$ increase from Boost to SemiBoost and from SemiBoost to Beyond-SemiBoost, which suggests that the evolution of the boosting-based trackers has resulted in a decreased ability to cope with changing illumination. The results also highlight the sensitivity of FragTrack to deal with changing illumination (the highest $\mu_C$ and the highest $d_C$ on $P_8$). MS has the best performance on $P_7$ both in terms of $\mu_C$ and the robustness ($d_C$) to cope with the compressed video data. In terms of $\mu_C$, Boost shows the closest performance to MS followed by PF, SemiBoost, BeyondSemiBoost and FragTrack, respectively; and in terms of $d_C$, PF has the closest performance to MS followed by Boost, FragTrack, BeyondSemiBoost and SemiBoost, respectively. Finally, on $P_8$, MS again outperforms other trackers in terms of $\mu_C$ and the robustness ($d_C$) to deal with resolution changes. The performance of Boost is closer to that of MS in terms of $\mu_C$ as compared to remaining trackers. Moreover, the performance of PF is closer to that of MS in terms of $d_C$ than the other trackers. Based on the performance analysis on $P_7$ and $P_8$, the performance of MS is the least affected by compression or reduction in the resolution.

**B. Target-wise comparison**

Figure 4 shows the mean CoTPS of trackers ($\mu_C$) on each target ($H_1, H_2, ..., H_6$) and their robustness in terms of dispersion of their CoTPS ($d_C$) computed in all trials.

The performance of SemiBoost is the best on $H_1$ in terms of $\mu_C$, followed by BeyondSemiBoost, MS, Boost, PF and FragTrack (Fig. 4(a)). In terms of $d_C$, the results show a smaller variation in performance of the standard trackers compared to the boosting-based trackers (Fig. 4(b)). There is a pose change of the target ($H_1$) around frame 107 of the sequence (Fig. 5(a)), where the boosting-based trackers lose the target (Boost only tracks a very small part of target in this frame). $H_2$ presents challenges such as partial occlusions, pose changes and scale changes. PF has the best performance in terms of $\mu_C$. The $\mu_C$ of Boost is the closest to PF. Moreover, MS and PF have a smaller variation ($d_C$) in their performance on $H_2$ compared to the remaining trackers. There is a significant pose change of target (360° turning) around frame 145 (Fig. 5(b)), where only MS, PF and Boost have tracked. On $H_3$, MS outperforms the other trackers as shown by its smallest $\mu_C$. This is because the appearance of target $H_3$ is very bright and well-distinguished from the background, and
and MS have similar performance on $H_5$ with MS showing the best tracking followed by PF (Fig. 5(d)). Boost is challenging to track. All trackers have struggled to track this target. CoTPS of trackers on each target computed with all trials; Boost (red), SemiBoost (green), BeyondSemiBoost (blue), MS (black), FragTrack (cyan) and PF (magenta). The dispersion value ($d_C$) for a tracker is computed as difference between its maximum and minimum CoTPS values on a target.

The use of color distribution enables MS to track well on the various generated test sequences containing $H_5$. SemiBoost has the smallest variation in performance on $H_5$ (smallest $d_C$). $H_5$ undergoes gradual change in its scale and pose across the sequence. MS deals with these challenges and tracks consistently well, followed by PF (Fig. 5(c)). $H_4$ is challenging due to the presence of background clutter, similar objects (vehicles) and scale changes, and all trackers have obtained high $\mu_C$ on it. MS has the best performance on $H_4$ in terms of $\mu_C$ followed by PF, Boost, FragTrack, BeyondSemiBoost and SemiBoost, respectively. In terms of variation in performance, although $d_C$ for SemiBoost is smaller than for the other trackers, this is less important as its CoTPS is mostly very high. Among the remaining trackers, $d_C$ of standard trackers is smaller than Boost and BeyondSemiBoost. The appearance of $H_4$ is very similar to that of the road making it challenging to track. All trackers have struggled to track this target with MS showing the best tracking followed by PF (Fig. 5(d)). Boost and MS have similar performance on $H_5$ in terms of $\mu_C$ ($\mu_C$ of PF and SemiBoost are also comparable to them). SemiBoost shows a smaller variation ($d_C$) in its performance on $H_5$ as compared to other trackers. $H_5$ faces a severe occlusion around frame 51 where only PF can track the target after the occlusion (Fig. 5(e)). $H_6$ has challenges such as the presence of targets of the same class (person), partial occlusions and small pose changes. FragTrack outperforms the other trackers in terms of $\mu_C$ as it can deal well with pose changes and partial occlusions [23]. The sequences containing $H_2$ and $H_5$ also involve pose changes and partial occlusions but FragTrack has not performed as well on them (Fig. 4(a)). $H_2$ involves significant pose changes and $H_5$ involves severe occlusions, suggesting that FragTrack can cope better with small pose changes and partial occlusions. Figure 5(f) shows frame 359 involving partial occlusion where FragTrack performs well (PF also tracks a small part of the target).

C. Discussion

Figure 6(a) shows the performance of trackers for each target class (head, vehicle, person). Each tracker has its best performance on head, followed by person and vehicle. The overall best performance on head and person tracking is by Boost. The performance of PF is closer to Boost on head tracking. The overall best performance on vehicle tracking is by MS. There is an inconsistency in the performance of FragTrack on person tracking: while it has achieved the best performance on $H_6$, its performance reduces significantly on $H_5$ (Fig. 4(a)), as discussed earlier.

Figure 6(b) shows the cumulative performance of trackers on all trials and all targets. MS has the best performance in terms of $\mu_C$ followed by Boost, PF, SemiBoost, BeyondSemiBoost and FragTrack, respectively. PF is more robust than the remaining trackers as shown by its smaller $d_C$. Finally, overall, the standard trackers are more robust when dealing with various test scenarios than the boosting-based trackers (smaller $d_C$ of the former set of trackers in Fig. 6(b)). MS handles better initialization errors and outperforms other trackers with compressed videos and resolution reductions. Boost copes well with noise, with frame dropping and with changing illumination. Among standard trackers, MS and PF can handle small as well as large pose changes; whereas FragTrack can only deal with small pose changes. Among boosting-based trackers, Boost outperforms SemiBoost and BeyondSemiBoost in handling pose changes. PF can handle partial and total occlusions better than all the other trackers.

To conclude, we tested the statistical significance of CoTPS using the Welch ANOVA test [27], a modified version of the one-way ANalysis Of Variance (ANOVA) test [28], commonly employed to test statistical significance of multiple groups of data (in our case, there are six groups each containing a set of $H_5$ of a tracker) whose variances are unequal [29]. Statistical significance was achieved on each trial, on each target and on each target class at the standard significance level $\alpha = 0.05$.

V. Conclusion

We introduced a new overlap-based criterion for the performance evaluation of video trackers on extended targets. The proposed criterion quantifies performance by combining tracking accuracy and tracking failure scores. We also presented a new evaluation protocol that quantifies the robustness of trackers under various real-world conditions, which are encapsulated in a series of trials. An extensive experimental analysis and validation is presented in the form of a statistically significant performance comparison of six state-of-the-art trackers. The implementation of the protocol is available online to provide the research community with a platform to present and compare the performance of their trackers.
Our future work involves extending the proposed evaluation criterion to multi-target tracking performance evaluation. Moreover, as the proposed trials are generic and not designed specifically for a ground-truth-based evaluation, we aim to use them in combination with standalone evaluation criteria.

REFERENCES


