Assessing Tracking Performance in Complex Scenarios using Mean Time Between Failures

Peter Carr
Disney Research
carr@disneyresearch.com

Robert T. Collins
The Pennsylvania State University
rcollins@cse.psu.edu

Abstract

Existing measures for evaluating the performance of tracking algorithms are difficult to interpret, which makes it hard to identify the best approach for a particular situation. As we show, a dummy algorithm which does not actually track scores well under most existing measures. Although some measures characterize specific error sources quite well, combining them into a single aggregate measure for comparing approaches or tuning parameters is not straightforward. In this work we propose 'mean time between failures' as a viable summary of solution quality — especially when the goal is to follow objects for as long as possible. In addition to being sensitive to all tracking errors, the performance numbers are directly interpretable: how long can an algorithm operate before a mistake has likely occurred (the object is lost, its identity is confused, etc.)? We illustrate the merits of this measure by assessing solutions from different algorithms on a challenging dataset.

1. Introduction

Characterizing the performance of algorithms is critical for determining which approach is best for a particular situation. Typically, object tracking algorithms are evaluated by comparing a set $\mathcal{E} = \{E_1, E_2, \dots\}$ of *estimated object tracks* to the set $\mathcal{A} = \{A_1, A_2, \dots\}$ of *actual object tracks* established from a ground truth data source. The results are tabulated as false alarms, missed occurrences and identity swaps; and a collection of measures such as precision, recall and multi-object tracking accuracy (MOTA) are derived from these base statistics [3, 8] (see Fig. 1).

Ideally, these individual measures are somehow combined into a single score to enable direct comparison between algorithms, as well as searching for optimal parameter configurations through cross validation [7]. When combining into a single score, the relative importance of each measure can be tailored to the scenario. For instance, surveillance for abandoned luggage would require virtually no missed detections. However, in the majority of the literature (and especially when comparing different algorithms), the standard approach is to treat all errors equally. An additional complication is that each measure typically has different units (and often different magnitudes), so treating each measure equally without some sort of normalization will result in a poor aggregate measure.

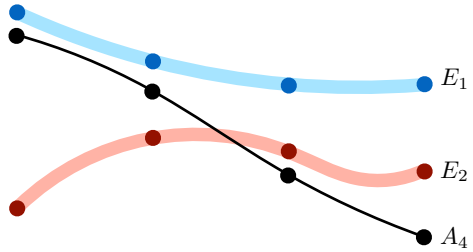

Figure 1. **Assocation**. An object A_4 moves throughout the scene, and a tracking algorithm generates estimated object tracks E_1 and E_2 (the input detections are not shown). The sampled spatial locations of all tracks over the four frame duration are shown as dots, and one-to-one data association is performed independently at each frame. The corresponding sequences of associations are $\mathbf{A}_4 = \langle E_1, E_1, E_2, \emptyset \rangle$, $\mathbf{E}_1 = \langle A_4, A_4, \emptyset, \emptyset \rangle$ and $\mathbf{E}_2 = \langle \emptyset, \emptyset, A_4, \emptyset \rangle$. The quality of this tracking solution can be derived from base statistics (true positives, false positive, false negatives and identity switches) tabulated from these associations.

Surprisingly, the majority of multi-object tracking performance measures do not take the duration of estimated tracks into account! This omission is striking because the general tracking problem is to follow a target for as long as possible without making a mistake. One would expect the amount of time an algorithm can operate until an error occurs to be an important factor in determining performance. Many of the established performance measures are not suited for this purpose because they have a bias towards shorter tracks (fewer opportunities to make mistakes). Because of this bias, low quality tracking solutions may achieve good performance scores. For example, con-

Method	Avg. Dur. [frames]	FP [-]	FN [-]	ID [-]	Prec. [%]	Rcll [%]	MT [%]	PT [%]	PL [%]	ML [%]	Avg. Frag. [-]	Pur. est [%]	Pur. act [%]	MOTA [-]	MTBF std [frames]	MTBF mono [frames]
Null	1.0	8458	11888	0	85.1	80.3	64.5	26.1	8.4	2.0	40.1	85.1	1.4	66.3	1.0	0.8
[1]	181.5	3444	18267	71	92.4	69.7	42.9	27.1	14.3	15.8	2.2	81.9	58.7	63.9	129.2	6.7
[2]	139.0	11023	9809	109	82.1	83.8	68.0	15.8	8.4	7.9	2.3	68.4	65.2	65.3	93.5	4.6
[4]	176.5	6857	10931	232	87.8	81.9	67.5	23.6	6.4	2.5	3.0	71.5	65.6	70.2	74.9	5.4
[12]	225.6	7333	11767	474	86.9	80.5	62.6	26.6	7.4	3.4	6.8	60.6	62.3	67.6	51.0	4.8

Table 1. **Performance Ambiguity**. The performance of various algorithms on the *Town Centre* dataset. Standard metrics are reported: average estimated track length (frames), the number of false positives, false negatives and identity switches; precision, recall, {mostly/partially}-{tracked/lost}, average fragmentations per track (switching from 'tracked' to 'not tracked'), purity (predominance of a single identity) for both estimated and actual tracks, and multi-object tracking accuracy (MOTA). The proposed 'mean time between failures' measure (in standard and monotonic forms) is also reported. Algorithm parameters were not tuned extensively. Remarkably, the 'null' algorithm (tracks that are a single frame in duration) achieves very competitive results for all measures except fragmentation and actual track purity. It is not clear how the standard measures should be interpreted relative to estimated track length.

sider the extreme case of a null tracking algorithm[1] which assigns a unique track identifier to each detection. Each estimated track is one frame in duration, making it impossible to have any identity switches. Alarmingly, the majority of the popular tracking measures [7, 8] give a competitive score to the result of the null algorithm (see Table 1):

- Precision and recall measure false positives and false negatives relative to true positives. A good tracking algorithm should filter out the false detections and fill in false negatives.
- 'Mostly tracked', 'partially tracked', 'partially lost' and 'mostly lost' are four categories by which all ground truth tracks are classified. The category is assigned based on the fraction of the track that was detected — i.e. 'mostly tracked' means at least 80% of the ground truth track was detected. These measures do not take into account the duration of the estimated track, or the consistency of the inferred identity.
- Fragmentation measures the number of times a ground truth track switches from being 'tracked' to 'not tracked', and vice versa. There is no direct dependence on duration, but similar to identity swaps, the amount of fragmentation tends to increase as tracks get longer.
- Purity measures the frequency of the predominant label. Tracks with high purity correspond to a single associated object, while tracks with low purity associate to multiple objects. This measure can be applied to either the estimated or actual object tracks.
- MOTA describes the aggregate error from false and missed detections, as well as identity swaps. Although identity swaps do not directly reflect the length of an estimated track, the number of identity swaps generally increases with longer estimated tracks.

[1]The null tracking algorithm is analogous to a null detection algorithm that simply predicts the predominant class (such as a patient not having a rare disease).

The existing measures are rarely able to distinguish a clear winner amongst competing algorithms. In part, this is because none of the measures are adequately sensitive to all error sources. Furthermore, there is no agreed upon method for combining the different measures. Successfully following an object for a sustained period of time may not be of primary importance in all tracking scenarios, and in these situations, established performances measures may suffice. However, as benchmark data sets have grown in size and complexity, the existing measures have struggled to differentiate the performances of various algorithms. As research progresses in sustained long term object tracking, a performance measure which can assess the duration under which objects can be followed successfully is needed.

In addition to being discriminative, a good performance measure should also be predictive. Ground truth data may be available in laboratory settings, but it is unavailable in the field. Ideally, a measure should say something about how an algorithm is expected to perform on future unseen (but representative) data. For example, the precision of an object detector describes the probability that an object is actually present if a detection is made on new data. Ideally, measurements of tracking performance should have similar predictive interpretations.

To address these shortcoming, we propose a new evaluation technique which combines all of the tracking error sources into a single number that reflects the average amount of time a tracking algorithm can successfully follow an object without making a mistake: the *mean time between failures* (MTBF). The term is borrowed from the field of reliability engineering [9]. Our empirical results show MTBF is an effective performance measure which generates clear differentiations between various tracking solutions. Furthermore, the measure has a clear interpretable meaning, and more importantly, is predictive about future performance: MTBF is an estimate of how long a tracking algorithm should be able to follow an object before it drifts away or confuses it with a different target. Compared to the

standard measures, computing MTBF requires an additional run length encoding stage which is trivial to implement. Finally, we prove that a variant of MTBF is monotonic; guaranteeing that a reduction in tracking errors (false positives, false negatives and identity switches) results in a better (or at least unchanged) performance measure.

2. Mean Time Between Failures

In the case of single object tracking, the performance measure is fairly straightforward: how long can a target be followed successfully? The idea of *failure rate* (the number of times a tracker must be manually re-initialized) was recently proposed for single object tracking [11]. In multi-target tracking, initializations and terminations occur automatically, making failure rate analogous to fragmentation. Instead, we propose the idea of "errorless duration", which is applicable to both single and multi-target tracking domains. For simplicity, we assume all errors are equally important and compute the mean time between any type of failures. But, we will also highlight variants of the measure for different subsets of errors (identity switches, and identity switches and fragmentations).

Computing the mean time between failures for object tracking requires two inputs: a set $\mathcal{A} = \{A_1, A_2, \dots\}$ of actual object tracks (the ground truth), and a set $\mathcal{E} = \{E_1, E_2, \dots\}$ of estimated object tracks (a tracking solution). Both sets use the same representation of a track: a unique identifying label paired with a sequence of timestamped spatial locations (which could be on the image plane or ground plane). Like [5], we require each track to be a single continuous temporal span.

Similar to other tracking measures, the first step for determining the mean time between failures is to perform data association between the estimated and actual object trajectories. Usually, this is accomplished by evaluating the spatial discrepancy between the sampled locations of each estimated track and actual track. Unlike [10], our formulation does not allow many-to-one or many-to-many associations, so we employ the Kuhn-Munkres (Hungarian) algorithm at each time instant independently. Furthermore, unlike [3], we do not include a greedy tracking stage before solving the linear assignment problem to minimize the number of identity switches. Instead, we assume a good tracking solution should generate an estimated track which is indeed closest to the actual object track. Once the bi-directional matching between the estimated tracks and the actual object tracks has been established at each time instant, the base statistics (the number of true positives, false positives, false negatives, and identity switches) for the standard tracking measures can be deduced directly from the association data. In our proposed approach, we first extract a useful intermediary representation from the associations, and then compute the standard statistics.

From the association data, one can extract the sequence $\mathbf{A}_i$ of labels representing the mapping between the i^{th} ground truth actual object track and its associated estimated object track at each time instant (Smith *et al*. [10] call these sequences of labels "configuration maps"). For example, Figure 1 shows a tracking result for a hypothetical scenario. The sequence of labels $\mathbf{A}_4 = \langle E_1, E_1, E_2, \emptyset \rangle$ means ground truth track A_4 was associated to estimated track E_1 for the first two frames, estimated track E_2 for the third frame, and was not associated to any estimated track in the fourth frame (i.e. a false negative). In a similar fashion, a sequence $\mathbf{E}_j$ of labels can be computed for each estimated object track, where these labels represent which actual object track was associated at each time instant. In this direction, the null association $\emptyset$ represents false positives. From these label sequences, the standard counts of true positives, false negatives and false positives can be computed (which are respectively 3, 1 and 5 for Fig. 1).

All of the classic tracking measures can be derived from an arbitrary set $\mathcal{L} = \{\mathbf{L}_1, \mathbf{L}_2, \dots\}$ of associated labels. Generally, the measured performance will be different depending on which set of tracks is analyzed: $\mathcal{A}$ or $\mathcal{E}$. We will discuss this aspect further, after explaining how standard measures and our proposed mean time between failures can be derived from $\mathcal{L}$.

For simplicity, we will explain how performance measures are calculated for a single sequence $\mathbf{L}_i$ of association labels. The aggregate performance across the entire set $\mathcal{L}$ is straightforward (often summing or averaging over the set). For an input sequence of associated labels $\mathbf{L}_i$, the periods of consistent identity assignment are easily determined from the run length encoding

$$\mathsf{RLE}(\mathbf{L}_i) = \langle \mathbf{R}_{i,1}, \mathbf{R}_{i,2}, \dots, \mathbf{R}_{i,K} \rangle, \tag{1}$$

where $\mathbf{R}_{i,k} = [\ell_k, D_k]$ is the k^{th} run and represents label ℓ_k repeated D_k times. The core tabulated measures can be computed directly from the run lengths

$$\textbf{True Positives} = \sum_k D_k[\ell_k \neq \emptyset], \tag{2}$$

$$\textbf{False Positives/Negatives} = \sum_k D_k[\ell_k = \emptyset], \tag{3}$$

$$\textbf{Identity Transitions} = K - 1. \tag{4}$$

The number of true positives is the same regardless of whether $\mathcal{A}$ or $\mathcal{E}$ is analyzed. However, only $\mathcal{E}$ determines the number of false positives, and only $\mathcal{A}$ determines the number of false negatives.

Identity transitions occur whenever two temporally consecutive associations do not have the same label. If one of the labels is null, then the transition is a **tracking fragmentation** (where the tracking solution transitions from 'tracking' to 'not tracking' or vice versa); otherwise it is

an **identity switch**. Identity switches can also occur in non-consecutive frames if there are null labels in between. For example, the sequence $\langle 1, \emptyset, 2 \rangle$ has two fragmentations and one identity switch, while the sequence $\langle 1, \emptyset, 1 \rangle$ has two fragmentations and no identity switches. Identity transitions are the upper limit for both identity switches and tracking fragmentations (identity switches and tracking fragmentations are not independent subsets)

$$\mathsf{Id\ Trans} = \max(\mathsf{Id\ Switches}, \mathsf{Tracking\ Frags}). \quad (5)$$

Often, fragmentation is computed for the ground truth tracks $\mathcal{A}$. From a prediction point of view, it is more useful to gauge identity swaps in terms of $\mathcal{E}$ because that reflects how often tracks outputted by a tracking algorithm confuse the identity of one target for another. To illustrate how these measures are dependent on which set of tracks is analyzed, consider the situation in Figure 1. The actual object tracks $\mathcal{A}$ have 1 fragmentation and 1 identity switch, whereas the estimated tracks $\mathcal{E}$ have 3 fragmentations and 0 identity switches. Reporting "identity switches" is ambiguous unless the reference set of tracks is specified. In our case, the values in Table 1 were computed using identity switches computed for the estimated object tracks.

From these base statistics, the standard measures of precision, recall, {mostly/partially}-{tracked/lost} and MOTA can be calculated. However, as we will now explain, the distribution of the run lengths provides insightful information about performance.

2.1. Properties of Mean Time Between Failures

Each run with a non-null label represents a period of error free tracking. The multiset $\mathcal{F}(\mathbf{L}_i) = \{D_k | \ell_k \neq \emptyset\}$ describes the observed times between errors. For convenience, one can summarize the distribution of times by its mean value[2]

$$\mathsf{MTBF} = \frac{1}{|\mathcal{F}|} \sum_k D_k[\ell_k \neq \emptyset], \quad (6)$$

but other moments could be computed as well. By definition, the MTBF of an empty set is zero.

The example sequence $\mathbf{L}_i = \langle 1, 1, 1, 2, \emptyset \rangle$ has a run length encoding $\mathsf{RLE}(\mathbf{L}_i) = \langle (1, 3), (2, 1), (\emptyset, 1) \rangle$. The mean time between failures would be $\frac{3+1}{2} = 2$ frames (runs with $\ell_k = \emptyset$ are omitted).

Sensitivity to All Error Sources MTBF measures the average duration of consistent identity associations (which may be null or non-null labels). Identity transitions occur because of instantaneous identity switches or tracking fragmentations (which may coincide with indirect identity switches). To the best of our knowledge, MTBF is the first performance measure that is simultaneously sensitive to both identity switches and tracking fragmentations. The standard measures are sensitive to only one of these error sources at most. If desired, a variant which is only sensitive to identity swaps can be computed: when omitting runs with null labels, consecutive runs with the same label are merged — eliminating the effect of fragmentations).

Normalized Mean time between failures for a single track is limited to the duration of the track. As a result, one can normalize MTBF to $[0.0, 1.0]$ (if desired) by scaling by the average track length. Normalized MTBF may be useful for comparing performance across datasets with different underlying motion characteristics — *i.e.* targets which are visible in the scene for long periods of time compared to other scenarios where targets are only visible briefly.

Asymmetric The computed mean time between failures will generally be different for $\mathcal{A}$ and $\mathcal{E}$. When analyzing the actual object tracks $\mathcal{A}$ from the ground truth, MTBF characterizes how well we expect the tracking algorithm to be able to follow the object without losing it or misidentifying it. When analyzing $\mathcal{E}$, MTBF describes the expected amount of time the tracker can follow an object before drifting away or misidentifying it. The distinction is subtle, and is analogous to the relationship between precision/recall. Typically, the harmonic mean of precision and recall is used to specify the aggregate performance (since they are rates). Mean time between failures is inversely proportional to the *failure rate* λ [9]. The harmonic mean of $\lambda_\mathcal{A}$ and $\lambda_\mathcal{E}$ is the arithmetic average of mean times between failures for $\mathcal{A}$ and $\mathcal{E}$

$$\mathsf{MTBF}_{\mathcal{AE}} = \frac{1}{\lambda_{\mathcal{AE}}}, \quad (7)$$

$$= \frac{\lambda_\mathcal{A} + \lambda_\mathcal{E}}{2\lambda_\mathcal{A}\lambda_\mathcal{E}}, \quad (8)$$

$$= \frac{1}{2}\mathsf{MTBF}_\mathcal{A} + \frac{1}{2}\mathsf{MTBF}_\mathcal{E}. \quad (9)$$

As a result, the best way to concisely summarize the temporal reliability of a tracking algorithm is to quote the arithmetic mean of the MTBF measures calculated for estimated object tracks and actual object tracks. However, more thorough descriptions about a tracking algorithm's performance can be computed from the distribution of errorless tracking intervals (such as higher order moments of the distribution).

Not Monotonic Leichter and Krupka [6] argue that a good performance measure should be monotonic with respect to error rates, or more precisely, the reduction of false positives, false negatives or identity switches should not cause the tracking performance measure to decrease. Although this relationship tends to hold for MTBF in practice, there are circumstances where a reduction in errors could cause MTBF to decrease.

If the number of false negatives is reduced, then some null elements in one or more $\mathbf{A}_i$s will become non-null. Of-

[2] A reference C++ implementation is included as supplemental material.

Scenario Associations	True Pos.	False Neg.	Id. Sw.	Total Errors	MOTA	MT/PT/ PL/ML	Frag.	Purity	MTBF standard	MTBF monotonic
$\mathbf{A}_1 = [E_1\ E_1\ E_1\ E_1\ E_1]$	5	0	0	0	100.0	MT	0	1.0	5.00	5.00
$\mathbf{A}_2 = [E_1\ E_1\ E_1\ E_2\ E_2]$	5	0	1	1	80.0	MT	0	0.6	2.50	2.50
$\mathbf{A}_3 = [E_1\ E_1\ E_1\ E_2\ \emptyset\ ]$	4	1	1	2	60.0	MT	1	0.6	2.00	1.33
$\mathbf{A}_4 = [E_1\ E_1\ E_2\ E_1\ E_2]$	5	0	3	3	40.0	MT	0	0.6	1.20	1.20
$\mathbf{A}_5 = [E_1\ E_1\ \emptyset\ E_2\ \emptyset\ ]$	3	2	1	3	40.0	PT	3	0.4	1.50	0.75
$\mathbf{A}_6 = [\ \emptyset\ E_1\ \emptyset\ E_2\ \emptyset\ ]$	2	3	1	4	20.0	PL	4	0.2	1.00	0.40
$\mathbf{A}_7 = [\ \emptyset\ \emptyset\ \emptyset\ \emptyset\ \emptyset\ ]$	0	5	0	5	0.0	ML	0	0.0	0.00	0.00

Table 2. **Synthetic Examples**. Performance measures for seven scenarios involving a single actual object trajectory (ground truth) and two estimated object trajectories (tracking solution). Identity transitions have been partitioned into identity swaps and fragments. Multi-object tracking accuracy (MOTA) exhibits a clear trend in quality of the tracking solutions. The {mostly/partially}-{tracked/lost} classification as well as the fragmentation count are both insensitive to misidentification errors, and do not adequately characterize the quality of the tracking solution. Purity does not take into account the internal cohesiveness of labels, giving tracks $\mathbf{A}_2$ and $\mathbf{A}_4$ the same score (where $\mathbf{A}_2$ is clearly a better tracking solution). The standard formulation of mean time between failures gives a similar quality ranking as MOTA, but with a slight variation in preferences between low quality solutions. The monotonic variant of MTBF tends to over penalize false positives and negatives, and has a different preference for the ordering of $\mathbf{A}_4$ and $\mathbf{A}_5$ (the monotonicity property places no constraint on scores when the number of errors is equal; hence $\mathbf{A}_3$ has a higher MTBF than both $\mathbf{A}_4$ and $\mathbf{A}_5$, and $\mathbf{A}_6$ has a lower MTBF than both).

ten, the switch to a non-null label is consistent with its immediate predecessor or successor, which results in a slightly longer run and an increase in MTBF. However, if the new non-null label is not the same as its predecessor and successor, then a new run of length 1 is created, which most likely will make MTBF go down (unless it is ≤ 1 already).

If monotonicity is desired, a more strict definition of MTBF can be used. Instead of run length encoding all labels, the monotonic variant only run length encodes non-null labels. For example, the sequence $\langle 1, 1, \emptyset, \emptyset \rangle$ would be run length encoded as $\langle (1, 2), (\emptyset, 1), (\emptyset, 1) \rangle$. When constructing the multiset $\mathcal{F}$, the value D_k is added for every run involving a non-null label, and the value 0 is inserted for every run involving a null label (which by definition is always a single frame in duration because null labels were not run length encoded). Effectively, null labels result in errorless durations of zero frames.

Proposition 1. *MTBF is a monotonic measure if each null label corresponds to zero time between failures.*

Proof. If the number of false negatives decreases, then one or more errorless durations of zero will be replaced by either: (1) extending an existing non-null labeled errorless duration by one frame or (2) creating a new non-null run with an errorless duration of one frame (which may induce additional identity switches). In both cases, the MTBF will not decrease. The same arguments apply to $\mathcal{E}$ for the case of reducing the number of false positives. Finally, if the number of identity swaps is reduced, then either: (1) consecutive non-null runs are merged into a single longer run (increasing MTBF) or (2) non-null runs separated by a null run change to the same label (keeping MTBF the same). □

We have found the more strict definition of MTBF to be less useful in practice because its distribution of errorless durations is heavily skewed towards zero, and these errors are well represented in established measures such as precision and recall. Furthermore, because runs of null labels are omitted from the computation of standard MTBF, normalizing by the average track length will still reflect errors arising from false positives and false negatives. For example, consider two tracking algorithms where one is tuned to be more conservative at terminating tracks. When the object is lost, the first algorithm will terminate its track immediately, while the second will extrapolate forwards in time and suffer false positives until it is convinced it is necessary to terminate the track. Assuming both algorithms had the same solution up to the point that the object was lost, the absolute MTBF scores would be the same. However, since the second algorithm generated a longer track (with false positives at the end), when interpreting MTBF relative to estimated track length, the second algorithm will generate a lower fractional length.

Empirical Table 2 contains performance measures for seven toy examples involving a single actual object trajectory (from the ground truth) and two estimated object trajectories (from a tracking algorithm). The standard measures give reasonable assessments to each of the solutions, but few have sufficient discrimination power to reflect which tracking solutions are better than others.

The multiple object tracking accuracy (MOTA) gives a reasonable measure of performance for each of the tracks (in this case there are no false positives because the example is only evaluating the quality by which the tracking algorithm was able to follow the actual objects). Because MOTA is indifferent to fragmentations it assigns the same scores to $\mathbf{A}_4$ and $\mathbf{A}_5$.

The classification of {mostly/partially}-{tracked/lost} is invariant to the number of identity swaps, and so is the count of fragmentations (switching from 'tracked' to 'not tracked' and vice versa). As a result, these measures are not always reliable, such as giving situations $\mathbf{A}_1$ and $\mathbf{A}_4$ the same score. Similarly, the purity measure [10] does not take into account the cohesiveness of associations within the track, and gives equal preference to $\mathbf{A}_2$ and $\mathbf{A}_4$.

The standard formulation of MTBF mostly agrees with the rankings of MOTA. However, MTBF is sensitive to fragmentation errors (MOTA is not) and both the standard and monotonic variants have preferences for how $\mathbf{A}_4$ and $\mathbf{A}_5$ should be ordered. The monotonic variant tends to over emphasize errors from false negatives and positives (which result in zero frame errorless durations) which is why it ranks $\mathbf{A}_4$ ahead of $\mathbf{A}_5$.

3. Experiments

For the hypothesized situations in Table 2, both MOTA and MTBF give reasonable scores, although MOTA is indifferent to tracking fragmentation errors. We now compare MTBF to the established tracking performance measures on a larger dataset from a real scenario using a variety of solutions generated by different data association algorithms. Our experiments are designed to assess a measure's ability to distinguish the quality of different tracking solutions. We are not focusing on which algorithm generates the best solution, and have not adjusted parameters to maximize each algorithm's performance. The quality of the generated solutions should not be considered a good proxy for algorithm performance. Before presenting each measure's assessments of the various solutions, we first describe the scenario and how the actual object tracks are established.

3.1. Ground Truth

We use the publicly available *Town Centre* dataset [1]. We are interested in estimating the trajectories of objects on the ground plane, and not bounding boxes of heads in the image plane. As a result, we define a region of interest on the ground plane in which all individuals should be fully visible unless they are occluded by another person. To translate the published ground truth of bounding box head locations to (x, y) locations on the ground plane, we manually estimated the height of each individual. During this process we noticed errors in the published annotations: paths of some individuals were annotated as unconnected segments, and identities of some pedestrians changed midtrack. We corrected annotations as necessary[3] which reduced the total number of tracks from 228 to 203. Furthermore, the average track length increased from 262.4 frames to 297.5 frames (the video is 25 fps).

[3]Our revised ground truth is included as supplementary material.

Figure 2. **Projective Uncertainty**. The region of interest is demarcated with a red line. (left) Within the ROI, any person's head and feet should be visible (unless occluded by another object). The uncertainty of a spatial location on the image plane is modeled as an isotropic 2D normal distribution, exemplified by concentric circles. (right) Propagating the uncertainty through the projective transform results in an error distribution on the ground plane which is not normally distributed (the ellipses would need to be concentric). Because Mahalanobis distances are only valid for normal distributions, we associate actual object locations to estimated object locations based on image plane distance.

3.2. Associations

Although we track objects on the ground plane, we perform data association on the image plane. As Figure 2 illustrates, the non-linear projective transform of the camera means errors that are normally distributed on the image plane are not normally distributed on the ground plane (the ellipses on the ground plane are not concentric).

The first step in performance assessment is to match the input set $\mathcal{E}$ of estimated object tracks to a reference set $\mathcal{A}$ of actual object tracks (from the ground truth). In our approach, we sample both sets of tracks over all time instants to generate a set $\mathcal{E}_t$ of estimated object locations at each time t, as well as a set $\mathcal{A}_t$ of actual object locations. Both sets of ground plane locations are then projected into the image plane. We then compute the image plane distances between all possible pairings of estimated and actual image plane locations. We use the Kuhn-Munkres (Hungarian) algorithm to determine the optimal association of estimated image plane locations to/from actual image plane locations based on the squared image plane distances (since the distances are distributed according to a Rayleigh distribution). This scheme is used for computing both MTBF and MOTA.

The matchings are subject to an upper feasible distance limit. If no good matching can be found then the sampled location is either a false positive (if an estimated location has no suitable actual location) or a false negative (if an actual location has no suitable estimated location). In practice, the spatial uncertainty of a detection algorithm should be characterized through repeated trials under the same stimulus. Because that methodology isn't applicable in this situation, we employ a heuristic to estimate the maximum feasible distance. We assume the detector is reasonably reliable, which means the estimated/actual object location pair with the smallest image plane distance is probably the cor-

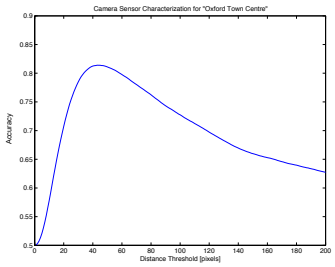

Figure 3. **Feasible Distance Limit**. In order to associate an estimated object location with an actual object location, the image plane distance must be below the feasible distance limit. To determine that value, we search for the decision threshold in a binary classifier which maximizes the classifier's accuracy when predicting outcomes for the smallest and second smallest distances.

rect association, and the pair with the next smallest distance is usually not a correct association. We can define a binary classifier using an above/below threshold test to predict whether a particular pair of estimated and actual object locations should be associated. For overly small thresholds, both the smallest and second smallest distances will not be considered feasible, whereas for overly large thresholds, both the smallest and second smallest distances will be considered feasible. In practice, we want the smallest distance to be feasible, and the second smallest distance to be infeasible. We determined a feasible distance limit of 43 pixels by searching for the threshold which maximized the binary classifier's accuracy (see Fig. 3).

3.3. Performance Measures

Table 1 lists the performance of different tracking solutions that have been published for the *Town Centre* dataset. The MOTA scores are very similar — including the solution of the null tracking algorithm! The average length of the estimated tracks is somewhat consistent as well (except for the null solution, obviously). From these two measures, it is difficult to determine whether one solution is better than another. The mean time between failures has a fairly big spread. At the low end, one tracking algorithm can operate for about 4s before likely making a mistake, while at the upper end, another algorithm is expected to operate for 6s before making an error. The monotonic variant suggests [1] and [4] have nearly equivalent performance, while [2] appears to be better.

Figure 4 shows one minus the cumulative distributions of errorless durations for the three tracking solutions. One minus the cumulative distribution illustrates how the solution from [1] is able to track an object for a substantially longer time for a given error tolerance. For example, after about 75 frames, solutions [2] and [4] have a 40% chance of making at least one error. The method from [1], on the other hand, is able to operate for approximately 125 frames before reaching the same 40% chance of making at least one error. To make this distinction more clear, we can plot the *reliability* function which describes the probability of no error occurring after t seconds assuming a fixed error rate $\lambda = \frac{1}{\text{MTBF}}$

$$R(t) = \exp(-\lambda t). \tag{10}$$

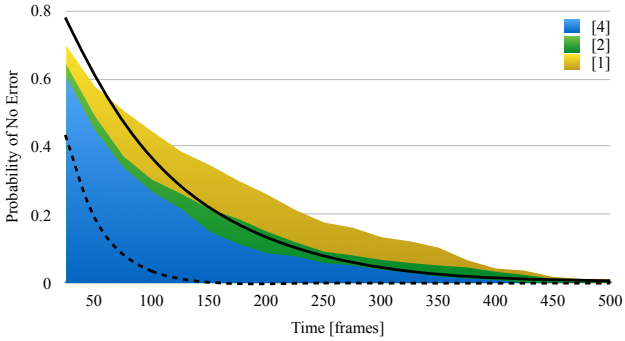

Figure 4. **Reliability**. One minus the cumulative distribution of errorless durations is a useful way for visualizing the reliability of an algorithm, and is equivalent to the probability of tracking for time t without making an error. Ideally, the data should follow an exponential distribution. For reference, reliability curves of corresponding to MTBFs of 100 (solid line) and 30 (dotted line) are plotted. The predictive nature of MTBF is directly interpretable from the plot. For example, [1] can operate for approximately 125 frames before there is a only 40% chance of not making a mistake; whereas [2] and [4] can only operate for about 75 frames before reaching the same probability of an error having occurred.

For reference, two reliability curves are plotted: the solid line represents the expected distribution for a MTBF of 100 frames, and the dashed line represents 30 frames. These values roughly correspond to the performances of the three tracking solutions in Table 1 for both the standard version of MTBF and its monotonic variant. The MTBF for 30 frames does not represent the actual distribution of the non-zero errorless durations, because it is accounting for a large spike at $t = 0$ (not shown) arising from false negatives and false positives. In reliability engineering terms, MTBF is used to characterize the expected failure rate during the *useful life* period [9]. It specifically ignores the higher error rates during the *early life* and *wear out* periods, which in object tracking terms is analogous to track initialization and termination. By interpreting standard MTBF scores relative to the average duration of an actual object track (297.5 frames in this case), the errors arising from false positives and false negatives are still accounted for. The advantage of this perspective (normalized standard MTBF) is that it avoids the over emphasis of false positives and negatives present in the monotonic variant, and treats identity switches and tracking fragmentations as equally important errors compared to not following a target or accidentally following something that is not an object of interest.

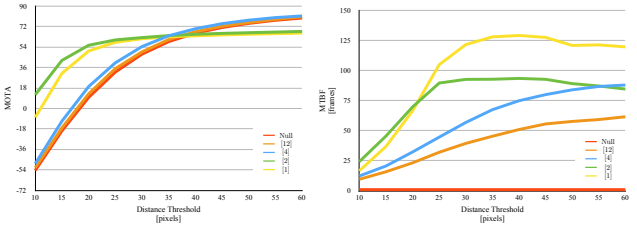

Figure 5. **Precision Sensitivity**. The tracking solutions of Table 1 are re-evaluated for different spatial error tolerances (see Fig. 3). MOTA scores (left) are mostly determined by the input set of detections, whereas MTBF (right) provides good separation between solutions generated from the same set of detections.

Spatial Precision When [11] proposed *failure rate* as a performance measure for single-target trackers, the authors noted that spatial precision (bounding box overlap) is a complementary measure. Because multi-object tracking automatically determines initializations and failures, we plot curves of MOTA and MTBF as a function of feasible image plane distance (see Fig. 5). For example, the MTBF of [2] plateaus at a spatial precision of 25 pixels, whereas [4] achieves a similar performance, but at 55 pixels, implying the localization accuracy of these detections is much lower.

3.4. Correlation Analysis

In addition to the tracking solutions listed in Tab. 1, we generate additional solutions by varying the parameters of [12] (true positive rate, true negative rate and initialization/termination probability) such that the average duration of estimated tracks was uniformly sampled. The correlation between different measures across all tracking solutions is shown in Fig. 6. Compared to MOTA, MTBF exhibits significant negative correlation with all three fundamental error sources. There is strong positive correlation with precision, recall and mostly tracked; and negative correlation with partially tracked, partially lost and mostly lost, as well as fragmentation. There is positive correlation with purity and MOTA. Only the anti-correlations with partially lost and mostly lost are not significant ($p > 0.05$). In contrast, MOTA and purity have minimal correlation with false positives and identity switches (because the number of false negatives is substantially larger). Furthermore, MOTA only has significant correlation ($p < 0.05$) with precision, recall, mostly tracked and purity (estimated and actual).

4. Summary

Although it is important to have specific measures for each type of error [6], a single aggregate measure is useful for comparing algorithms, as well as tuning parameters through cross validation. When feasible, individual measures should be aggregated through a loss function crafted for the particular scenario. However, customized loss functions are not always practical. Our approach is to think of

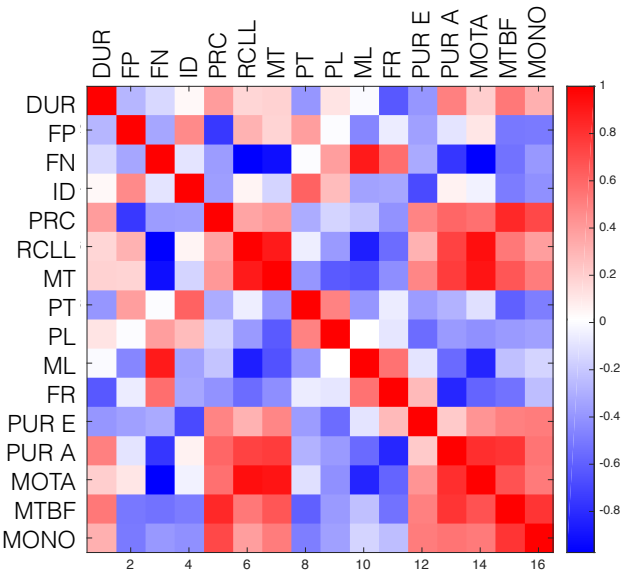

Figure 6. **Correlation Analysis**. The correlation between the different measures (see Tab. 1) aggregated over various tracking solutions. MTBF has significant correlation ($p < 0.05$) with all measures except PL and ML. MOTA has negligible correlation with track duration.

errors as events, and to characterize the expected amount of time between events. As we have shown, *mean time between failures* has many useful properties:

- The computed numbers are intuitive, and have a predictive 'reliability' interpretation (see Fig. 4).
- Our empirical evidence suggests the measure can provide good discriminability between different tracking solutions (see Fig. 5).
- The measure is equally sensitive to all types of tracking errors (see Fig. 6).
- If desired, variants are available with sensitivity specific to different subsets of errors: (1) identity switches only, (2) identity switches and fragmentations, and (3) all error sources. The full measure is monotonic with respect to the three fundamental errors source: false positives, false negatives and identity switches; making it useful for cross-validation.

Mean time between failures addresses many of the weaknesses that are present in the established set of measures for assessing tracking performance — especially if the ultimate goal is to follow targets for as long as possible. As research in visual object tracking continues to pursue more complex scenarios, we believe MTBF and an inspection of the distribution of errorless durations will be a useful tool for understanding how well algorithms perform in different situations.

References

[1] B. Benfold and I. Reid. Guiding visual surveillance by tracking human attention. In *BMVC*, 2009. 2, 6, 7

[2] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. In *CVPR*, 2011. 2, 7, 8

[3] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *EURASIP Journal on Image and Video Processing*, 2008. 1, 3

[4] R. Collins and P. Carr. Hybrid stochastic / deterministic optimization for tracking sports players and pedestrians. In *ECCV*, 2014. 2, 7, 8

[5] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler. Motchallenge 2015: Towards a benchmark for multi-target tracking. *arXiv:1504.01942*, 2015. 3

[6] I. Leichter and E. Krupka. Monotonicity and error type differentiability in performance measures for target detection and tracking in video. Technical Report MSR-TR-2012-23, Microsoft Research, 2012. 4, 8

[7] A. Milan, K. Schindler, and S. Roth. Challenges of ground truth evaluation of multi-target tracking. In *CVPR Workshops*, 2013. 1, 2

[8] T. Nawaz, F. Poiesi, and A. Cavallaro. Measures of effective video tracking. *Image Processing, IEEE Transactions on*, 23(1):376–388, 2014. 1, 2

[9] E. Nikolaidis, D. Ghiocel, and S. Singhal. *Engineering Design Reliability Handbook*. CRC Press, 2004. 2, 4, 7

[10] K. Smith, D. Gatica-Perez, J. Odobez, and S. Ba. Evaluating multi-object tracking. In *CVPR Workshops*, 2005. 3, 6

[11] L. Čehovin, M. Kristan, and A. Leonardis. Is my new tracker really better than yours? In *WACV*, 2014. 3, 8

[12] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*, 2008. 2, 8

Revisions from WACV Round 1

Reviewer #1

Temporary loss is not a big deal A variant of MTBF which is not sensitive to fragmentation errors is now described around line 380.

Report Identity Switches Table 1 has been revised to include this measure.

MOTChallenge Already cited as [5].

Only consider good labels MTBF is sensitive to identity switches (these result in separate runs of consistent labels). The measure you are referring to is analogous to *purity* which we have now included in Table 1 as well.

Wrong Object Tracked The tracking algorithm has no idea it has made a mistake. We treat identity switches like manual re-initializations. When a tracker switches to an incorrect object, how well long can it follow this new object before making an additional mistake? Penalizing everything after the first mistake would produce overly conservative measures.

Evaluation in Ground Plane An error tolerance of 1m is very generous in the foreground, and overly conservative in the background. Image plane distance is not affected by non-linear projection.

Reviewer #2

Nawaz et al. Added this relevant reference (thanks!). The core metrics in Nawaz are equivalent to precision, recall and the other standard measures.

Two measures for spatiotemporal consistency Because initialization/termination happens automatically in multi-object tracking, the effects of spatial precision are easier to visualize by plotting the impact of the decision threshold on the other metrics. See Fig. 5.

Null labels and switches A variant which is only sensitive to identity switches is now described around line 380.

Predictive Interpretation Interpretation is similar to precision: given a detection, what is the probability of it being correct, or a false alarm. MTBF describes the number of frames a tracker can operate before an error has most likely occurred. We have revised the caption of Fig. 4 with an explicit example.

Experiments showing correlation See Sec. 3.2.

Reviewer #3

Association scheme Same scheme is used for MTBF and MOTA. Explicitly stated on line 635.

Table 2 discussion Added final line to caption of Table 2 to mention that monotonicity only applies to changes in the number of errors. If the number of errors stays the same, the score can go up down or stay the same (but remains within the bounds of scores with more or fewer errors).

MTBF vs Precision and Recall All measures are now specified in Table 1. See additional analysis in Sec. 3.4.