Leveraging single for multi-target tracking using a novel trajectory overlap affinity measure

Santiago Manen[1] Radu Timofte[1] Dengxin Dai[1] Luc Van Gool[1,2]

[1] Computer Vision Laboratory, ETH Zurich
[2] ESAT - PSI / iMinds, KU Leuven
{smanenfr, Radu.Timofte, dai, vangool}@vision.ee.ethz.ch

Abstract

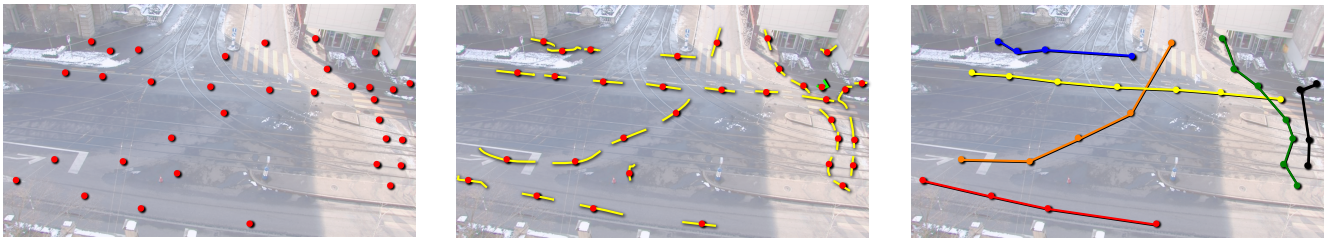
*Multi-target tracking (MTT) is the task of localizing objects of interest in a video and associating them through time. Accurate affinity measures between object detections is crucial for MTT. Previous methods use simple affinity measures, based on heuristics, that are unable to track through occlusions and missing detections. To address this problem, this paper proposes a novel affinity measure by leveraging the power of single-target visual tracking (VT), which has proven reliable to locally track objects of interest given a bounding-box initialization. In particular, given two detections at different frames, we perform VT starting from each of them and towards the frame of the other. We then learn a metric with features extracted from the behaviours (*e.g. *overlaps and distances) of the two tracking trajectories. By plugging our learned affinity into the standard MTT framework, we are able to cope with occlusions and large amounts of missing or inaccurate detections. We evaluate our method on public datasets, including the popular MOT benchmark, and show improvements over previously published methods.*

1. Introduction

Multi-target tracking (MTT) aims to infer target trajectories in a video. This is challenging in the presence of low image quality, target deformations, occlusions, visual and motion similarity among the targets, and/or cluttered environments. Frame-by-frame solutions, such as (visual) target tracking, are prone to failure as they usually do not cope well with ambiguous or noisy observations and the overlap of target trajectories. Multi-frame solutions use more information and reach improved results by considering all the observations of the video in one holistic optimization. From the latter category a popular approach is the so-called Data-Association-based Tracking (DAT) [43, 37, 24, 3, 32, 33], in which detections or short track fragments are associated into trajectories using a variety of features.

The MTT literature flourished with the advances in object detection [12, 4, 10]. The 'tracking-by-detection' paradigm became a dominant direction for both visual tracking (VT) [36] and MTT research [26]. MTT takes as input a (dense) set of detections in an image sequence. With object detections, most MTT solutions rely on two main components [26]: an affinity measure and an association optimization formulation. The affinity model provides the likelihood that two detections/tracklets belong to the same target, while the optimization framework determines the linkage between detections/tracklets based on their affinity. Popular optimization frameworks include the Hungarian algorithm [37], Linear Program [3] and cost-flow network [43]. Detection errors force MTT approaches not only to solve the association problem, but also to figure out false positives and missing detections. Accurate affinity measures are crucial for both tasks: assigning detections of the same object together and singling out false detections for the final optimization to prune them.

The affinity between detections can be defined based on different properties. 1) *Appearance models* (based usually on pixels, gradients or visual words) are useful when handling known object classes with low intraclass variability. However, for challenging scenarios (our focus) with clutter, occlusions, target ambiguity, inaccurate or missing detections and targets with large variability in appearance, appearance models are often less reliable sources of discriminant information than motion and/or interaction between the targets. 2) *Interaction models* assume a fair amount of prior knowledge about the targets and/or large annotated training materials, which limits their applicability [31]. 3) *Motion models*, on the other hand, in their simplest form of direction preservation, can help recovery from trajectory crossings and/or (short-term) occlusions. Pirsiavash *et al*. [32] consider that targets move slowly, the affinity be-



(1) Object detection (2) Single-target tracking (3) Multi-target tracking

Figure 1: We (1) compute object detections, (2) from which we initialize a visual tracker to obtain short trajectories, which we use for (3) global data association.

tween different time frame detections is given by the overlap in the image space. Lasdas *et al*. [20] use motion priors in static-camera scenes to improve over Pirsiavash *et al*. Some approaches do not globally optimize over detections directly, but rather first build track fragments (tracklets) employing optical flow [44], greedy local assignments [35], RANSAC-based clustering [25], or the optimization framework with conservative settings [19, 39, 15], to then optimize over them. These methods are complementary to ours, since our affinity measure can be trivially extended to associate tracklets instead of individual detections. Xing *et al*. [37] and Yang and Nevatia [40] employ a constant velocity model to extend pairs of tracklets in the overlapping time frames. The difference between the predicted positions is their used affinity measure. A variant is the dynamic model of Andriyenko and Schindler [1]. Kuo and Nevatia [19] use the acceleration besides the position and velocity. [5] proposed an affinity measure based on the amount of KLT tracks that are consistent between two detections. When the targets are moving freely, then sets of motion patterns or non-linear motion models could help [40, 39]. Andriyenko *et al*. [2] employ a multi-model fitting in discrete-continuous optimization. More recently, [8] proposed the association of tracklets whose union can be explained with low order motion models. For defining the affinity measure some works use handcrafted combinations of the aforementioned models [34, 19, 26], while others learn metrics and/or boosted combinations [33, 24].

Our main contribution is the 'trajectory overlap' (TO) affinity measure. It generalizes the standard overlap (Intersection-over-Union, *IoU*) [9] often used to measure the affinity between two detections. Our TO measures the likelihood that two detections belong to the same target. In particular, given two detections at different frames, we perform VT starting from each of them and towards the frame of the other. We then learn a metric with features extracted from the behaviours (e.g overlaps and distances) of the two tracking trajectories to obtain our TO affinity measure. The main steps of TO are illustrated in Fig. 1. Our approach is similar in spirit to the point trajectory generation of [17]. The difference is that our affinity is based on VT results and used in an MTT framework. [38] also proposed using VT results to improve MTT. But they simply use visual tracklets to extend the pool of detections and solve the association in a frame-to-frame greedy manner, whereas we integrate VT in a globally optimal MTT framework. The superior performance of TO are ascribed to 1) the power of VT, which has proven reliable to track objects for at least a short period of time [36, 16], and 2) the power of metric learning. By integrating VT and metric learning into MTT, our method is able to handle more challenging datasets where missing detections (gaps) or inaccurate detections are common. We use as global data association framework the cost-flow network of [43].

2. Global Data Association Framework

The data association in MTT is usually formulated as a *Maximum a Posteriori* (MAP) problem. We review here the framework we use, as proposed by Zhang *et al*. [43]. The goal is to find the most likely set of trajectories hypothesis given a set of object detections $\mathcal{X} = \{\mathbf{x}_i\}$. Each detection $\mathbf{x}_i = \{x_i, s_i, t_i, \beta_i\}$, is represented by its position x_i, scale s_i, time t_i, and confidence measure $\beta_i \in [0, 1]$. Each individual hypothesis T_i consists of a set of detections sorted by time, e.g. $\{\mathbf{x}_4, \mathbf{x}_{10}, \mathbf{x}_{13}, \mathbf{x}_{20}\}$. Assuming 1) trajectory independence, 2) non-overlapping trajectories and 3) using costs (negative log-likelihoods), the MAP inference can be re-written [32] as an Integer Linear Program (ILP):

$$\begin{aligned} T = \quad & \textbf{argmin}_{\mathcal{T}} \sum_i C_i^{en} f_i^{en} + \sum_{i,j} C_{ij} f_{ij} + \sum_i C_i^{ex} f_i^{ex} + \sum_i C_i f_i \\ \text{s.t.} \quad & f_i^{en}, f_{ij}, f_i^{ex}, f_i \in \{0, 1\} \\ & f_i^{en} + \sum_j f_{ji} = f_i = f_i^{ex} + \sum_j f_{ij} \end{aligned} \tag{1}$$

where T is the set of trajectories that minimize the total cost, C_i^{en} and C_i^{ex} are the costs of $\mathbf{x}_i$ being, resp., the entry and exit points of a trajectory, C_i is the cost of considering $\mathbf{x}_i$ a true positive and C_{ij} is the cost of linking detections $\mathbf{x}_i$ and $\mathbf{x}_j$. As in any Integer Program, f_i^{en}, f_{ij}, f_i^{ex} and f_i are variables that indicate if their respective cost should be applied.

The detection cost, $C_i = \log\left((1 - \beta_i)/\beta_i\right)$, is positive if the detection confidence β_i is over 0.5 (and negative oth-

erwise). These confident detections are the ones that encourage the global optimization to associate detections and generate trajectories. Typically, and also in our work, C_i^{en} and C_i^{ex} are considered constant, i.e., without an entry or exit point prior. We indicate in section Sec. 4.1 the value we use for these parameters.

Eq. (1) can be solved with the *push-relabel* maximum flow algorithm of [7] if the number of trajectories K is known. K can be determined by finding the solution with lowest cost using bisection. The overall complexity of the global association is $O(N^3 \log^2(N))$, assuming a uniform distribution of the detections $\mathcal{X}$ throughout the video. We use the push-relabel implementation from [32].

In the rest of the paper, we work with the affinity measure $A_{ij} \in [0, 1]$, which is used to compute the pairwise linking cost as $C_{ij} = -\log(A_{ij})$. A_{ij} is an intuitive measure representing the likelihood that $\mathbf{x}_i$ and $\mathbf{x}_j$ should belong to the same object and should be associated.

3. Trajectory Overlap Affinity Measure

In this section we introduce the main contribution of this paper – our TO affinity similarity measure. Let the real value $A_{ij} \in [0, 1]$ be the affinity measure between detections $\mathbf{x}_i$ and $\mathbf{x}_j$. A_{ij} indicates how likely it is that detections $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same object. As briefly explained above, we propose to compute A_{ij} based on trajectories predicted from $\mathbf{x}_i$ and $\mathbf{x}_j$ using a visual tracker (see Fig. 2).

For each detection $\mathbf{x}_i$ in the video, we initialize a visual tracker and track the object hypothesis forwards and backwards in time for a certain amount of time τ. By this means, we obtain a relatively short trajectory ν_i centered around the detection ($t_i - \tau \leq t \leq t_i + \tau$), see Fig. 2. Tracking for a longer duration will provide more information, but it will also be more susceptible to drifting and efficiency issues. Therefore, we fit a motion model $\mathcal{M}$ to ν_i to make a more extensive prediction of the trajectory of detection i. We denote this refined trajectory prediction with $\mathcal{T}_i$ (see Fig. 2). Note that fitting a motion model to ν_i introduces motion dynamics to the classical multi-target tracking framework. Indeed, fitting such a motion model helps when tracking through big gaps of missing detections and occlusions. We test different combinations of tracking durations τ and models $\mathcal{M}$ in the experiments and show how the optimal complexity of the model is directly related to the visual tracking duration τ, i.e., a higher τ can accommodate a more complex motion model. In practice, the trajectory predictions only need to be accurate up to the next detection of the same object.

Each trajectory prediction $\mathcal{T}_i$ is the result of the target-specific appearance model and the motion model of visual tracking. Detections that are associated should have similar trajectory predictions, whereas false positives of the object detector should be uncorrelated with respect to other detections. Based on these valuable properties, we propose to learn A_{ij} using a feature vector Φ_{ij} extracted from the trajectory predictions $\mathcal{T}_i$ and $\mathcal{T}_j$ (see Fig. 2).

The feature vector Φ_{ij} consists of three concatenated vectors:

1. **Time-wise overlap:** The overlap of $\mathcal{T}_i$ and $\mathcal{T}_j$ in each frame between $\mathbf{x}_i$ and $\mathbf{x}_j$. These overlaps are uniformly quantized in 10 bins to obtain a constant feature length. We use the PASCAL VOC [9] *intersection-over-union* (*IoU*) to compute the overlaps. The feature takes into account the relative location and size of the objects.

2. **Time-wise normalized distance:** The distance between the centers of the detections $\mathbf{x}_i$ and $\mathbf{x}_j$ for each frame, which is also uniformly quantized with 10 bins. We use the euclidean distance normalized with the average size of the detections at each frame.

3. **Time difference:** The time difference between detections $\mathbf{x}_i$ and $\mathbf{x}_j$, i.e. $|t_i - t_j|$. This feature helps to encourage connectivity, in order to avoid skipping valuable detections and duplicated trajectories.

In a sense, the time-wise overlap and distance features extend the affinities used in [32] and [43]. The main difference is that these previous works perform the computation directly using $\mathbf{x}_i$ and $\mathbf{x}_j$, whereas we compute them at each time-frame t between the detections, using the intermediate detections from the trajectory predictions $\mathcal{T}_i$ and $\mathcal{T}_j$. This makes our similarities more accurate in the presence of false negatives and occlusions, cf. Fig. 5.

Similarly to most global-association tracking-by-detection methods, we only consider connections within a certain temporal neighbourhood κ. That is, we only allow positive affinities A_{ij} between two detections $\mathbf{x}_i$ and $\mathbf{x}_j$ if $|t_i - t_j| \leq \kappa$. This neighbourhood maintains the method efficient and avoids very long jumps that could potentially introduce many false positives. Since our affinity is quite conservative and reliable even across big gaps, we can afford to have large jumps κ. We use 4 seconds (100 frames) in our experiments. Note that given a certain neighbourhood κ each trajectory prediction $\mathcal{T}_i$ only needs to be accurate up to κ frames as shown in Sec. 4.3. We motivate our selection of the visual tracker in the implementation details section, Sec. 4.1.

3.1. Learning the TO affinity measure

We model our affinity measure using logistic regression from the feature vector Φ_{ij}:

$$A_{ij} = \frac{1}{1 + e^{-(\mathbf{w}^T \Phi_{ij} + b)}}, \tag{2}$$

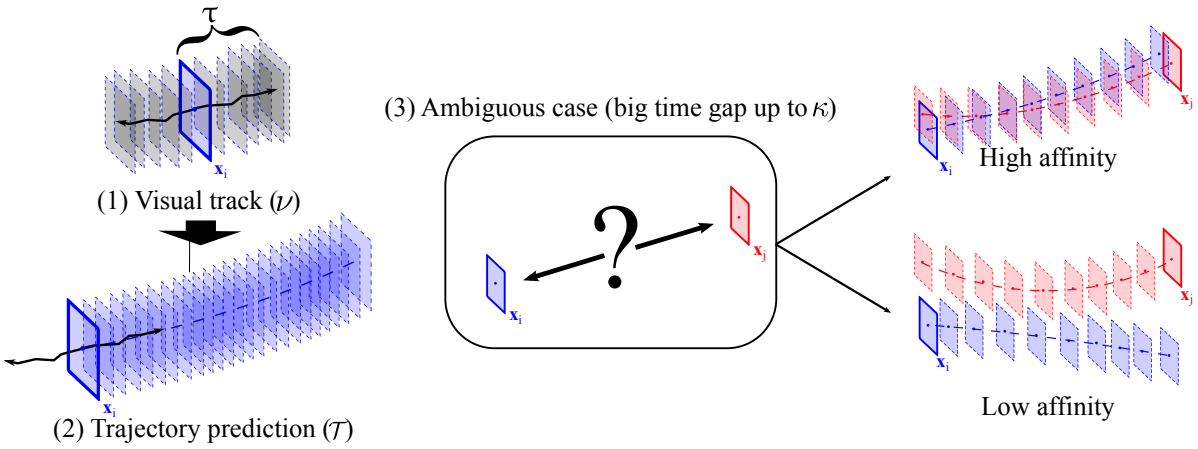


Figure 2: Illustration of the Trajectory Overlap (TO) affinity measure. For each detection we (1) run a visual tracker to obtain short trajectories (ν) that we use to (2) make trajectory predictions ($\mathcal{T}_i$). (3) TO can handle ambiguous cases in which two detections are separated by a number of frames (gap), by measuring the overlap of the predicted trajectories.

where $\mathbf{w}$ are the weights indicating the importance of each feature and b is a bias.

We learn $\mathbf{w}$ and b from a set of training trajectory annotations in a different training sequence. In order to have the most similar settings at training and test time, we use the same object detector to obtain an initial set of detections. Each pair of detections with an overlap of at least 0.25 *IoU* with the same ground truth trajectory is a positive example ($y_i = 1$). And the pairs of detections that do not overlap the same trajectory, those which belong to different trajectories, are negative examples ($y_i = 0$). We use the previously described feature extraction process both at train and test time. For each detection $\mathbf{x}_i$, we use the same visual tracker to obtain a short trajectory ν_i, to which we fit the same motion model $\mathcal{M}$ that we use at test time.

Time-wise overlap weights (10 bins):									
1.31	1.38	1.46	1.54	1.60	1.60	1.54	1.45	1.38	1.31
Time-wise normalized distance weights (10 bins):									
-0.35	-0.35	-0.36	-0.36	-0.36	-0.36	-0.36	-0.35	-0.35	-0.35
Time difference weight: -0.33									
Bias (b): 4.51									

Table 1: Learned weights ($\mathbf{w}$, b) for TO.

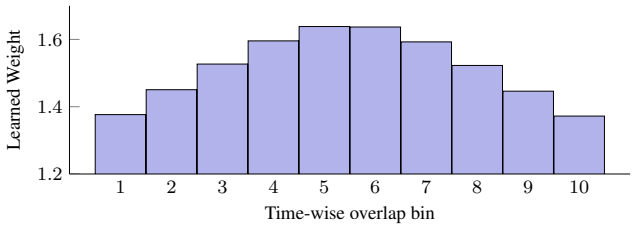


Figure 3: Learned overlap weights.

In our experiments, we train the weights of the regressor $\mathbf{w}$ in a separate set of trajectories with 5,842 positive object examples and equal number of negative examples. We use the same weights, $\mathbf{w}$ and b, for all the experiments, showing how the same weights can be used across datasets. Tab. 1 shows the learned weights. The learning yields very intuitive results: 1) A higher overlap and a lower distance between the trajectories increases the affinity. 2) Interestingly, a negative weight for the time difference encourages connectivity, i.e., the affinity between two detections is higher if they are closer together in time. The complementary plot in Fig. 3 focuses on the weights of the time-wise overlap measure. This plot shows how the importance of the overlap is the highest right at the middle, time-wise, of two detections and gradually diminishes in both directions outwards. This follows the intuition that the trajectories will most likely overlap close to the middle of the two detections if they actually belong to the same object. Since the measure is symmetric, the largest distance is in the middle, and the bin weights are symmetric as well.

4. Experiments

We present in this section our experimental setup and results. Sec. 4.1 details the sequences and metrics we use and implementation details. Sec. 4.2 presents a comparison of our method with the current state-of-the-art. In Sec. 4.3 we give more insight into the TO affinity measure by analyzing various properties and features.

4.1. Datasets and experimental protocol

We evaluate our method on two datasets: i) we use the *MOT Benchmark* [21] to compare with state-of-the-art methods and ii) provide additional baseline and state-of-the-

art comparisons on two publicly available surveillance sequences (*Hospedales3* and *Kuettel1*).

The MOT Benchmark [21] consists of 11 sequences with 721 pedestrian trajectory annotations. These sequences are very diverse. They have different lengths, frame rates, some are taken at street-level and others from a higher viewpoint. The dataset also includes static and moving cameras. It enforces learning only from a training set and aims to be the standard benchmark in multi-target tracking. We use this benchmark to present a state-of-the-art comparison.

The second datasets we use contains 329 long trajectories, with 231,853 object instances, distributed over two sequences: *Hospedales3* (see Fig. 5 top), 256 trajectories, and *Kuettel1* (see Fig. 5 bottom), 73 trajectories. Both are 2000 frames long and have a frame rate of 25 frames per second. These sequences were first introduced by [14] and [18]. They are very challenging due to heavy occlusions, small objects and low image quality, hindering the accuracy of the initial object detections. Moreover, we do not only evaluate on vehicle tracking, but also on tracking of pedestrians in *Kuettel1*, which often are < 50 pixels tall. We use the trajectory annotations provided by [27] for *Hospedales3* and provide on our website annotations for *Kuettel1*, for which we used the annotation tool [42].

It is challenging to evaluate MTT with just one metric. Therefore, we use a combination of the popular CLEAR MOT Metrics [6], especially designed for MTT, and the classical Precision-Recall (PR) curve, which provide a retrieval perspective. These performance metrics require an overlap threshold to consider an object as being properly detected. Since *Hospedales3* and *Kuettel1* contain many small objects, which are more difficult to annotate and detect accurately, we use a loose Intersection-over-Union (*IoU*) criterion of 0.25. We use the standard 0.5 *IoU* for the MOT Benchmark.

Initial detections Global-association MTT requires an initial set of object detections. To test the robustness of our affinity measure with respect to detection accuracy, we use 3 different object detection setups, see Fig. 4a and Fig. 6c. First we use the detections of Felzenszwalb *et al*. [10] with the *out-of-the-box* models trained on PASCAL VOC [9]. These detections are poor, because the appearance of the vehicles and pedestrians in our datasets is very different than in VOC. To retrain [10] with more similar objects, we follow the setup of [27] of training a scene-specific detection model with a small set of 30 trajectory annotations in each sequence, but in a completely separate span of time. This setup has practical applications, since such a small number of trajectories are fast to annotate for any sequence of interest. Positive examples are extracted from the trajectory annotations and hard negatives are mined from the background image. Objects of interest are trained together using 3 components. Fig. 4a shows the large improvement (0.3 AP) of retraining. We also run a background-segmentation-based object detector (BSOD) with 30 trajectory annotations of each sequence. BSOD starts from a set of object proposals, foreground blobs, which are scored by a trained regression forest based on location, scale and color contrast features. Fig. 4b shows how it yields better results than [10] with the *out-of-the-box* model, but worse than with the retrained model. The retrained version of [10] is our default detector for *Hospedales3* and *Kuettel1*. We use the provided detections for evaluation on the MOT Benchmark to ensure a fair comparison.

Table 2: Comparison in the MOT Benchmark. Best in bold.

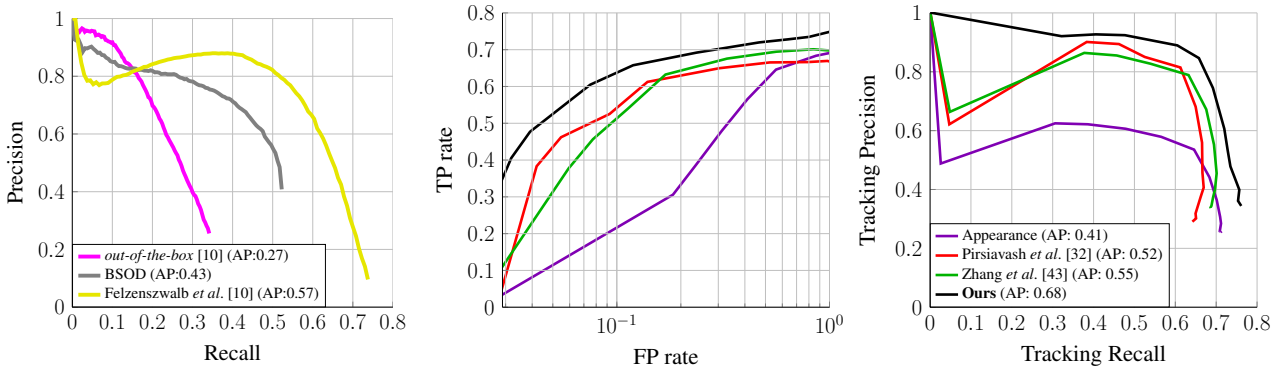
Tracker	MOTA	MOTP	MT	ML	ID Sw.	Frag
TBD [11]	0.16	0.71	0.06	0.48	1,939	1,963
SMOT [8]	0.18	0.71	0.03	0.55	1,148	2,132
RMOT [41]	0.19	0.70	0.05	0.53	684	1,282
CEM [30]	0.19	0.71	**0.08**	0.46	813	1,023
LP2D [22]	0.20	0.71	0.07	**0.41**	1,649	1,712
SDT	0.22	0.71	0.04	0.56	566	1,496
SegTrack [29]	0.22	**0.72**	0.06	0.64	697	737
MotiCon [23]	0.23	0.71	0.05	0.52	1,018	1,061
ELP [28]	0.25	0.71	0.07	0.44	1,396	1,805
TO (ours)	**0.26**	**0.72**	0.04	0.57	**383**	**600**

Table 3: Comparison in *Hospedales3* and *Kuettel1*. Best in bold.

Affinity	FP rate	TP rate	ID Sw. (per frame)	MOTP	MOTA	ML	MT
Appearance	0.23	0.38	0.21	**0.64**	0.15	0.23	0.27
[43]	0.07	0.42	0.05	0.62	0.35	0.24	0.30
[32]	0.05	0.42	0.02	0.62	0.38	0.23	0.31
Ours (unlearned)	0.03	0.40	**0.00**	0.63	0.37	0.31	0.24
Ours	**0.03**	**0.48**	0.01	0.61	**0.44**	**0.19**	**0.33**

Implementation details We do not consider scene-specific entry or exit priors. Instead we use a constant entry (C_i^{en}) and exit cost (C_i^{ex}) of 10, as in [32], for all detections, keeping the tracker scene-generic. We use the visual tracker ASLA [16] to obtain the trajectory ν_i for each detection $\mathbf{x}_i$. This tracker has state-of-the-art accuracy [36] and can track scale changes.

We explore in Sec. 4.3 different motion models $\mathcal{M}$ to make trajectory estimations. But, unless otherwise stated, for each detection i we compute a visual track ν_i of 4 frames of length (τ) forwards and backwards in time and use a constant velocity motion model $\mathcal{M}$ to obtain the trajectory prediction $\mathcal{T}_i$. An advantage of our affinity measure is that it can associate detections across big gaps of missing detections or occlusions. Therefore, we extract for our method detections at a rate of 2 frames per second and use a temporal neighbourhood κ of 4 seconds. We show in Sec. 4.3 how a larger the tracking duration τ allows to fit a more flexible motion model $\mathcal{M}$ and associate more distant detections.



(a) Detection results

(b) Tracking results

Figure 4: (a) Precision-Recall curve of the 3 object detectors considered and (b) the proposed TO affinity measure vs. other measures and tracking frameworks using the retrained [10] detector for *Hospedales3* and *Kuettel1*.

4.2. Comparison to the state of the art

MOT Benchmark Tab. 2 compares the performance of the proposed tracker (TO) against the top performing trackers published in previous conferences. We also include a comparison (SDT) with an affinity measure between detections similar to the one presented in [5], which measures the consistency of KLT tracks between two detections. More specifically, it is defined as the number of tracked interest points shared by the detections normalized by the total number of interest points they contain, i.e., the intersection-over-union criterion. Our approach has favorable performance, with a higher MOTA and MOTP and low id switches and fragments. The lower *mostly tracked* (MT) and higher *mostly lost* (ML) scores show that it is more conservative than previous approaches. Note that MT and ML are incomplete metrics since they do not take into account false positives. These results are available in the webpage of the benchmark [21].

Hospedales3 and Kuettel1 Tab. 3 shows the comparison of our TO-based MTT approach with respect to state-of-the-art methods ([43, 32]) and an appearance-based affinity measure. The appearance-based affinity is computed with the Bhattacharyya distance between the color histograms (8 bins of RGB space) of the two detections of interest. We use the most accurate detector, the retrained Felzenszwalb *et*

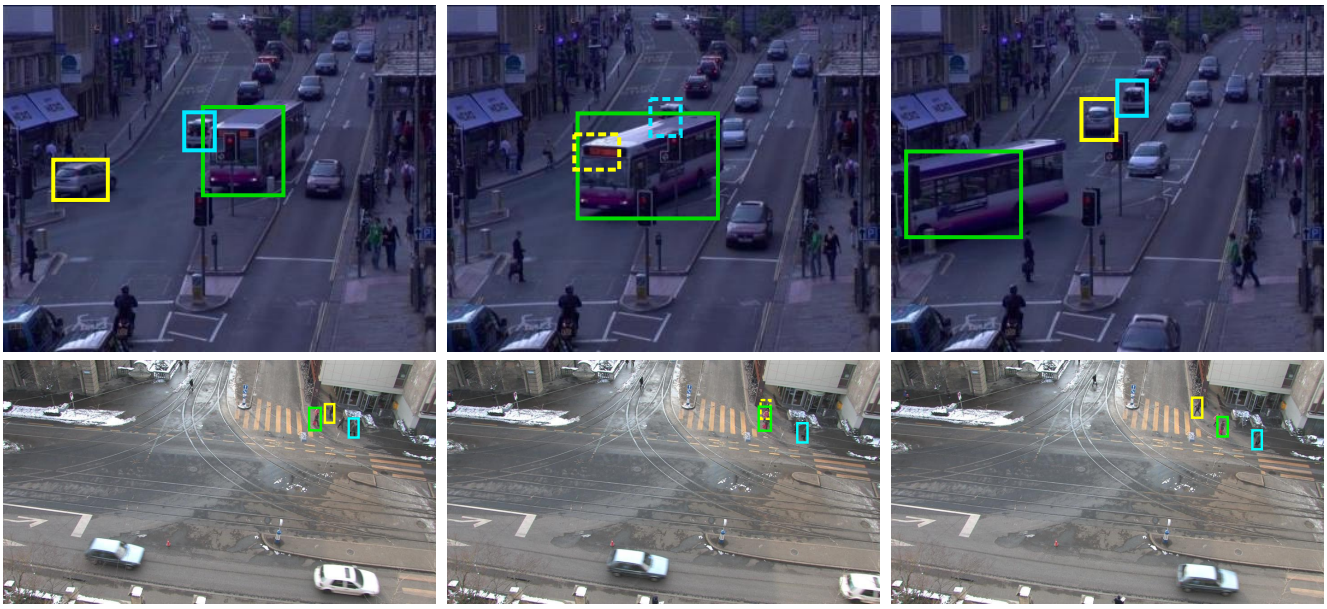
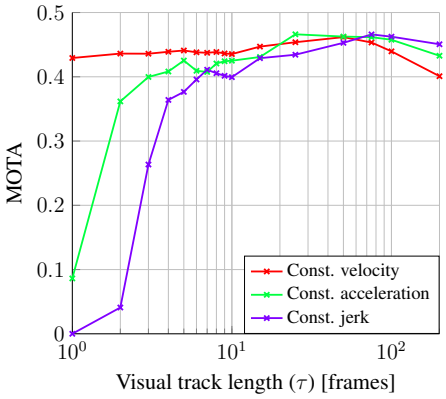
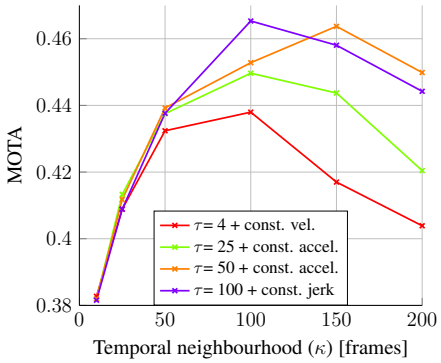


Figure 5: Examples where our TO affinity measure allows tracking trough full occlusions and trajectory crossings. Only selected tracks are shown to avoid clutter. Dashed windows indicate objects that we manage to track during occlusions, when detections are not available. More results in the accompanying video.



(a) $\mathcal{M}$ comparison



(b) κ analysis

	Object detectors		
Method	VOC [10]	BSOD	[10]
[32]	0.26	0.28	0.52
[43]	0.28	0.31	0.55
Ours	**0.33**	**0.44**	**0.68**

(c) Tracking robustness

Figure 6: We compare in (a) several motion models for different trajectory lengths τ. (b) shows the MOTA for different neighbourhoods and motion models. (c) Tracking performance comparison (measured in AP) for different object detectors. These evaluations refer to the Sequences *Hospedales3* and *Kuettel1*.

al. [10]. Our approach achieves a MOTA 16% higher, relatively, than the second best. Indeed, it manages to detect the largest amount of objects (TP), while keeping the lowest false positive rate (FP). Also, our methods copes well with occlusions and trajectory crossings (see Fig. 5). This shows the discriminative power of the TO affinity measure. As expected, due to the small size of the objects and the low quality of the videos, the appearance-based model yields the worst results, except for a high MOTP that is due to a low true positive rate (TP). For a more in-depth comparison, we also provide the Precision-Recall curve and the True Positive and False Positive rate curve in Fig. 4b. These plots are obtained by applying a threshold to the detection costs C_i, i.e. it is equivalent to the process of changing the detection threshold to obtain a PR curve in object detection. Note that our TO affinity similarity consistently yields more true positives for the same number of false positives. Our affinity particularly excels when very precise trajectories ($> 95\%$ precision) are needed. Tab. 3 also provides the results of our method directly using the average overlap along the trajectories instead of learning the feature weights, as described in Sec. 3. It can be seen how learning the feature vector significantly improves performance.

Robustness to less reliable detections A good affinity measure is important when object detections are unreliable. To test this use the three detectors evaluated in Fig. 4a as input for the MTT methods. Fig. 6c summarizes the Average Precision (AP) results of the tracking results. Our affinity measure consistently yields better results than [32, 43]. This was expected, since the TO affinity measure is particularly useful when associating through big gaps of occlusions and missing detections.

4.3. Further insights

In this section, we provide baseline comparisons to answer the two main questions posed by the TO affinity measure: 1) What is an adequate motion model $\mathcal{M}$ to use? 2) How far can we associate detections, i.e. what temporal neighbourhood κ can we choose?

Motion model $\mathcal{M}$ As explained in Sec. 3, for each detection i, we track forwards and backwards for τ frames with a visual tracker in order to obtain ν_i and then fit a motion model $\mathcal{M}$ to obtain a more extensive trajectory prediction $\mathcal{T}_i$. We test three motion models $\mathcal{M}$ that assume, from lower to higher complexity: 1) constant velocity, 2) constant acceleration, and 3) constant jerk, i.e., a constant variation of acceleration. Ideally, we would model motion on the ground plane. However, we avoid using the ground plane and define the motion models on image space. This keeps the affinity more generally applicable. We present in Fig. 6a the performance of the different models with respect to the length τ of the visual tracks to which they are applied. The results are quite intuitive. Tracking for a longer duration τ provides more information, so we can employ a more complex and flexible motion model. The constant acceleration and constant jerk models need a τ of 12 and 80 frames, respectively, to improve over a constant velocity model. Note that a long tracking duration τ is more informative, but also more susceptible to drifting and more time costly. Tracking each detection for only 4 frames forwards and backwards takes for our sequences an average of 50 ms per frame with a real-time single target tracker and 8 cores. This can be further parallelized and sped up using more recent visual trackers, such as [13], which runs at hundreds of frames per second while achieving top accuracy.

Temporal neighbourhood κ An important parameter in most data-association approaches is the temporal neigh-

bourhood κ, i.e., maximum time-difference between two detections to be connected. We show in Fig. 6b the influence of the visual tracking duration τ on the tracking MOTA for different temporal neighbourhoods κ. We conclude that: 1) Up to jumps of 50 frames (2s) it does not really matter what τ and motion model is used, so an efficient τ of 4 frames is convenient. 2) After 50 frames, the tracking accuracy increases as we use more single object tracking information, by increasing τ. 3) Our TO affinity measure allows tracking through detections as much as 4s apart, this is a significant improvement over other methods, which consider much smaller temporal neighbourhoods.

5. Conclusion

We proposed a novel Trajectory Overlap affinity measure (TO) that improves multi-target tracking performance, by leveraging the power of single-target visual tracking. For each detection we used a visual tracker to obtain trajectory predictions. The likelihood of two such detections to belong to the same target is measured based on the overlap in the predicted trajectories. Our TO combines overlap and distance between predicted positions and time with offline learned weights. By plugging our TO into the standard cost-flow network data association framework we obtained a robust solution capable to cope with missing and inaccurate detections and significantly to improve over top methods on challenging datasets.

Acknowledgements This work was supported by the European Research Council (ERC) under the project VarCity (#273940).

References

[1] A. Andriyenko and K. Schindler. Multi-target tracking by continuous energy minimization. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1265–1272, June 2011.

[2] A. Andriyenko, K. Schindler, and S. Roth. Discrete-continuous optimization for multi-target tracking. In *CVPR*, pages 1926–1933. IEEE, 2012.

[3] H. Ben Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Tracking multiple people under global appearance constraints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 137–144, Nov 2011.

[4] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool. Pedestrian detection at 100 frames per second. In *CVPR*, 2012.

[5] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 3457–3464, Washington, DC, USA, 2011. IEEE Computer Society.

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

[7] B. V. Cherkassky and A. V. Goldberg. On implementing the push-relabel method for the maximum flow problem. *Algorithmica*, 19(4):390–410, 1997.

[8] C. Dicle, O. I. Camps, and M. Sznaier. The way they move: Tracking multiple targets with similar appearance. In *ICCV*, 2013.

[9] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision*, 88(2):303–338, June 2010.

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[11] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun. 3d traffic scene understanding from movable platforms. *PAMI*, 2014.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 580–587. IEEE, 2014.

[13] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(3):583–596, 2015.

[14] T. Hospedales, S. Gong, and T. Xiang. A markov clustering topic model for mining behaviour in video. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1165–1172, Sept 2009.

[15] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. In *Proceedings of the 10th European Conference on Computer Vision: Part II*, ECCV '08, pages 788–801, Berlin, Heidelberg, 2008. Springer-Verlag.

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

[17] Z. Kalal, K. Mikolajczyk, and J. Matas. Forward-backward error: Automatic detection of tracking failures. In *ICPR*, 2010.

[18] D. Kuettel, M. Breitenstein, L. Van Gool, and V. Ferrari. What's going on? discovering spatio-temporal dependencies in dynamic scenes. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1951–1958, June 2010.

[19] C.-H. Kuo and R. Nevatia. How does person identity recognition help multi-person tracking? In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1217–1224, June 2011.

[20] V. Lasdas, R. Timofte, and L. Van Gool. Non-parametric motion-priors for flow understanding. In *Applications of Computer Vision (WACV), 2012 IEEE Workshop on*, pages 417–424, Jan 2012.

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

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

[23] L. Leal-Taix, M. Fenzi, A. Kuznetsova, B. Rosenhahn, and S. Savarese. Learning an image-based motion context for multiple people tracking. In *CVPR*, 2014.

[24] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2953–2960, June 2009.

[25] J. Liu, P. Carr, R. Collins, and Y. Liu. Tracking sports players with context-conditioned motion models. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1830–1837, June 2013.

[26] W. Luo, X. Zhao, and T.-K. Kim. Multiple object tracking: A review. *arXiv preprint arXiv:1409.7618*, 2014.

[27] S. Manen, J. Kwon, M. Guillaumin, and L. Van Gool. Appearances can be deceiving: Learning visual tracking from few trajectory annotations. In *European Conference on Computer Vision (ECCV)*, volume 8693. Springer International Publishing, 2014.

[28] N. McLaughlin, J. Martinez Del Rincon, and P. Miller. Enhancing linear programming with motion modeling for multi-target tracking. In *WACV*, 2015.

[29] A. Milan, L. Leal-Taix, K. Schindler, and I. Reid. Joint tracking and segmentation of multiple targets. In *CVPR*, 2015.

[30] A. Milan, S. Roth, and K. Schindler. Continuous energy minimization for multitarget tracking. *PAMI*, 2014.

[31] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 261–268, Sept 2009.

[32] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[33] B. Wang, G. Wang, K. L. Chan, and L. Wang. Tracklet association with online target-specific metric learning. In *CVPR*, 2014.

[34] W. Wang, R. Nevatia, and B. Yang. Beyond pedestrians: A hybrid approach of tracking multiple articulating humans. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 132–139, Jan 2015.

[35] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, 75(2):247–266, 2007.

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

[37] J. Xing, H. Ai, and S. Lao. Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1200–1207, June 2009.

[38] X. Yan, X. Wu, I. A. Kakadiaris, and S. K. Shah. To track or to detect? an ensemble framework for optimal selection. In *ECCV*, 2012.

[39] B. Yang and R. Nevatia. Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1918–1925. IEEE, 2012.

[40] B. Yang and R. Nevatia. An online learned crf model for multi-target tracking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2034–2041, June 2012.

[41] J. H. Yoon, M.-H. Yang, J. Lim, and K.-J. Yoon. Bayesian multi-object tracking using motion context from multiple objects. In *WACV*, 2015.

[42] J. Yuen, B. C. Russell, C. Liu, and A. Torralba. Labelme video: Building a video database with human annotations. In *ICCV*, 2009.

[43] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[44] X. Zhao, D. Gong, and G. Medioni. Tracking using motion patterns for very crowded scenes. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Computer Vision - ECCV 2012*, Lecture Notes in Computer Science, pages 315–328. Springer Berlin Heidelberg, 2012.