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

{smangenfr, 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 intra-class 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-
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. [21]. Between different time frame detections is given by the over-
global data association.

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

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 is 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 \( X = \{x_i\} \). Each detection \( x_i = \{x_i, s_i, t_i, \beta_i\} \), 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. \( \{x_{i4}, x_{i10}, x_{i13}, x_{i20}\} \). 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):

\[
\begin{align*}
T & = \underset{T}{\text{argmin}} \sum_i C_i^{en} f_i^{en} + \sum_{i,j} C_{ij} f_{ij} + \sum_i C_i^{ex} f_i^{ex} + \sum_i C_i f_i \\
\text{s.t.} & \quad f_i^{en}, f_{ij}, f_i^{ex}, f_i \in \{0, 1\} \\
& \quad f_i^{en} + \sum_j f_{ij} = f_i^{ex} + \sum_j f_{ij}
\end{align*}
\]

where \( T \) is the set of trajectories that minimize the total cost, \( C_i^{en} \) and \( C_i^{ex} \) are the costs of \( x_i \) being, resp., the entry and exit points of a trajectory, \( C_i \) is the cost of considering \( x_i \) a true positive and \( C_{ij} \) is the cost of linking detections \( x_i \) and \( x_j \). 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, \( C_i = \log ((1 - \beta_i)/\beta_i) \), is positive if the detection confidence \( \beta_i \) is over 0.5 (and negative oth-
These confident detections are the ones that encourage the global optimization to associate detections and generate trajectories. Typically, and also in our work, $C'^n_i$ and $C'^E_i$ are considered constant, i.e., without an entry or exit point prior. We indicate in Section 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 detections $x_i$ and $x_j$ 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 $x_i$ and $x_j$. $A_{ij}$ indicates how likely it is that detections $x_i$ and $x_j$ belong to the same object. As briefly explained above, we propose to compute $A_{ij}$ based on trajectories predicted from $x_i$ and $x_j$ using a visual tracker (see Fig. 2).

For each detection $x_i$ in the video, we initialize a visual tracker and track the object hypothesis forwards and backwards in time for a certain amount of time $\tau$. By this means, we obtain a relatively short trajectory $\nu_i$ 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 $\nu_i$ to make a more extensive prediction of the trajectory of detection $i$. We denote this refined trajectory prediction with $T_i$ (see Fig. 2).

Note that fitting a motion model to $\nu_i$ 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 $\tau$ and models $\mathcal{M}$ in the experiments and show how the optimal complexity of the model is directly related to the visual tracking duration $\tau$, i.e., a higher $\tau$ 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 $T_i$ 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 $\Phi_{ij}$ extracted from the trajectory predictions $T_i$ and $T_j$ (see Fig. 2).

The feature vector $\Phi_{ij}$ consists of three concatenated vectors:

1. **Time-wise overlap:** The overlap of $T_i$ and $T_j$ in each frame between $x_i$ and $x_j$. 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 $x_i$ and $x_j$ 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 $x_i$ and $x_j$, 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 $x_i$ and $x_j$, whereas we compute them at each time-frame $t$ between the detections, using the intermediate detections from the trajectory predictions $T_i$ and $T_j$. 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 $\kappa$. That is, we only allow positive affinities $A_{ij}$ between two detections $x_i$ and $x_j$ 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 $\kappa$. We use 4 seconds (100 frames) in our experiments. Note that given a certain neighbourhood $\kappa$ each trajectory prediction $T_i$ only needs to be accurate up to $\kappa$ 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 $\Phi_{ij}$:

$$A_{ij} = \frac{1}{1 + e^{-(\omega^T \Phi_{ij} + b)}}$$

(2)
where \( w \) are the weights indicating the importance of each feature and \( b \) is a bias.

We learn \( w \) 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 \( x_i \), we use the same visual tracker to obtain a short trajectory \( \nu_i \), to which we fit the same motion model \( M \) that we use at test time.

In our experiments, we train the weights of the regressor \( w \) in a separate set of trajectories with 5,842 positive object examples and equal number of negative examples. We use the same weights, \( w \) 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-
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] for 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.

**Implementation details** We do not consider scene-specific entry or exit priors. Instead we use a constant entry \(C_i^{in}\) and exit cost \(C_i^{off}\) of 10, as in [32], for all detections, keeping the tracker scene-generic. We use the visual tracker ASLA [16] to obtain the trajectory \(\nu_i\) for each detection \(x_i\). 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 \(\nu_i\) of 4 frames of length \(\tau\) forwards and backwards in time and use a constant velocity motion model \(\mathcal{M}\) to obtain the trajectory prediction \(T_i\). 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 \(\kappa\) of 4 seconds. We show in Sec. 4.3 how a larger the tracking duration \(\tau\) allows to fit a more flexible motion model \(\mathcal{M}\) and associate more distant detections.

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

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

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

<table>
<thead>
<tr>
<th>Affinity</th>
<th>FP rate</th>
<th>TP rate</th>
<th>ID Sw. (per frame)</th>
<th>MOTP</th>
<th>MOTA</th>
<th>ML</th>
<th>MT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appearance</td>
<td>0.23</td>
<td>0.38</td>
<td>0.21</td>
<td>0.64</td>
<td>0.15</td>
<td>0.23</td>
<td>0.27</td>
</tr>
<tr>
<td>[43]</td>
<td>0.07</td>
<td>0.42</td>
<td>0.05</td>
<td>0.62</td>
<td>0.35</td>
<td>0.24</td>
<td>0.30</td>
</tr>
<tr>
<td>[32]</td>
<td>0.05</td>
<td>0.42</td>
<td>0.02</td>
<td>0.62</td>
<td>0.38</td>
<td>0.23</td>
<td>0.31</td>
</tr>
<tr>
<td>Ours (ours)</td>
<td>0.03</td>
<td>0.40</td>
<td>0.00</td>
<td>0.63</td>
<td>0.37</td>
<td>0.31</td>
<td>0.24</td>
</tr>
<tr>
<td>Ours</td>
<td>0.03</td>
<td>0.48</td>
<td>0.01</td>
<td>0.61</td>
<td>0.44</td>
<td>0.19</td>
<td>0.33</td>
</tr>
</tbody>
</table>
Recall

Precision

out-of-the-box [10] (AP:0.27)

BSOD (AP:0.43)

Felzenszwalb et al. [10] (AP:0.57)

(a) Detection 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 through 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.
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 cope 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.

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 $M$ to use? 2) How far can we associate detections, i.e. what temporal neighbourhood $\kappa$ can we choose?

Motion model $M$ As explained in Sec. 3, for each detection $i$, we track forwards and backwards for $\tau$ frames with a visual tracker in order to obtain $\nu_i$ and then fit a motion model $M$ to obtain a more extensive trajectory prediction $T_i$. We test three motion models $M$ 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 $\tau$ of the visual tracks to which they are applied. The results are quite intuitive. Tracking for a longer duration $\tau$ provides more information, so we can employ a more complex and flexible motion model. The constant acceleration and constant jerk models need a $\tau$ of 12 and 80 frames, respectively, to improve over a constant velocity model. Note that a long tracking duration $\tau$ 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 $\kappa$ An important parameter in most data-association approaches is the temporal neigh-

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig6.png}
\caption{We compare in (a) several motion models for different trajectory lengths $\tau$. (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 Hosnedales3 and Kuettel1.}
\end{figure}
bourhood \( \kappa \), i.e., maximum time-difference between two detections to be connected. We show in Fig. 6b the influence of the visual tracking duration \( \tau \) on the tracking MOTA for different temporal neighbourhoods \( \kappa \). We conclude that: 1) Up to jumps of 50 frames (2s) it does not really matter what \( \tau \) and motion model is used, so an efficient \( \tau \) of 4 frames is convenient. 2) After 50 frames, the tracking accuracy increases as we use more single object tracking information, by increasing \( \tau \). 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


