Learning to Divide and Conquer for Online Multi-Target Tracking

Francesco Solera  Simone Calderara  Rita Cucchiara

Department of Engineering
University of Modena and Reggio Emilia
name.surname@unimore.it

Abstract

Online Multiple Target Tracking (MTT) is often addressed within the tracking-by-detection paradigm. Detections are previously extracted independently in each frame and then objects trajectories are built by maximizing specifically designed coherence functions. Nevertheless, ambiguities arise in presence of occlusions or detection errors. In this paper we claim that the ambiguities in tracking could be solved by a selective use of the features, by working with more reliable features if possible and exploiting a deeper representation of the target only if necessary. To this end, we propose an online divide and conquer tracker for static camera scenes, which partitions the assignment problem in local subproblems and solves them by selectively choosing and combining the best features. The complete framework is cast as a structural learning task that unifies these phases and learns tracker parameters from examples. Experiments on two different datasets highlights a significant improvement of tracking performances (MOTA +10%) over the state of the art.

1. Introduction

Multiple Target Tracking (MTT) is the task of extracting the continuous path of relevant objects across a set of subsequent frames. Due to the recent advances in object detection [10, 4], the problem of MTT is often addressed within the tracking-by-detection paradigm. Detections are previously extracted independently in each frame and then objects trajectories are built by maximizing specifically designed coherence functions [19, 5, 21, 2, 9, 24]. Tracking objects through detections can mitigate drifting behaviors introduced by prediction steps but, on the other hand, it forces the tracker to work in adverse conditions, due to the frequent occurrence of false and miss detections.

The majority of approaches address MTT offline, i.e. by exploiting detections from a set of frames [19, 5, 9] through global optimization. Offline methods benefit from the bigger portion of video sequence they dispose of to establish spatio-temporal coherence, but can not be used in real-time applications. Conversely, online methods track the targets frame-by-frame; they have a larger spectra of application but must be both accurate and fast despite working with less data. In this context, the robustness of the features play a major role in the online MTT task. Some approaches claim the adoption of complex targets models [2, 24] to be the solution, while others argue that this complexity may affect the long-term robustness [23]. For instance, in large crowds people appearance is rarely informative. As a consequence, tracking robustness is often achieved by focusing on spatial features [21], finding them more reliable than visual ones.

We do believe that many of the ambiguities in tracking could be solved by a selective use of the features, by working with more reliable features if possible and exploiting a deeper representation of the target only if necessary. In fact, a simple spatial association is often sufficient while, as clutter or confusion arise, an improved association scheme on more complex features is needed (Fig. 1).

In this paper a novel approach for online MTT in static camera scenes is proposed. The method selects the most
suitable features to solve the frame-by-frame associations depending on the surrounding scene complexity. Specifically, our contributions are:

- an online method based on Correlation Clustering that learns to divide the global association task in smaller and localized association subproblems (Sec. 5),
- a novel extension to the Hungarian association scheme, flexible enough to be applied to any set of preferred features and able to conquer trivial and complex subproblems by selectively combining the features (Sec. 6),
- an online Latent Structural SVM (LSSVM) framework to combine the divide and conquer steps and to learn from examples all the tracker parameters (Sec. 7).

The algorithm works by alternating between (a) learning the affinity measure of the Correlation Clustering as a latent variable and (b) learning the optimal combinations for both simple and complex features to be used as cost functions by the Hungarian. Results on public benchmarks underline a major improvement in tracking accuracy over current state of the art online trackers (+10% MOTA).

The work takes inspiration from the human perceptive behavior, further introduced in Sec. 3. According to the widely accepted two-streams hypothesis by Goodale and Milner [12], the use of motion and appearance information is localized in the temporal lobe (what pathway), while basic spatial cues are processed in the parietal lobe (where pathway). This suggests our brain processes and exploits information in different and specific ways as well.

2. Related works

Tab. 1 reports an overview of recent tracking-by-detection literature approaches separating online and offline methods and indicates the adoption of tracklets (T), appearance models (A), and complex learning schemes (L). Offline methods [5, 19, 13, 18] are out of the scope of the paper and are reported for the sake of completeness.

Tracklets are the results of an intermediate hierarchical association of the detections and are commonly used by both offline and online solutions [18, 13, 25]. In these approaches, high confidence associations link detections in a pre-processing step and then optimization techniques are employed to link tracklets into trajectories. Nevertheless tracklets creation involves solving a frame by frame assignment problem by thresholding the final association cost and errors in tracklets affect the tracking results as well.

In addition, online methods often try to compensate the lack of spatiotemporal information through the use of appearance or other complex features model. Appearance model is typically handled by the adoption of a classifier for each tracked target [24] and data associations is often finalized through an averaged sum of classifiers scores, [7, 2]. As a consequence, learning is on targets model, not on associations. Moreover, online methods also need to cope with drifting when updating their targets model. One possible solution is to avoid model updating when uncertainties are detected in the data, i.e. a detection cannot be paired to a sufficiently close previous trajectory [2]. Nevertheless, any error introduced into the model can rapidly lead to tracking drift and wrong appearance learning. Building on these considerations, Possegger et al. [21] does not consider appearance at all and only work with distance and motion data.

Differently from the aforementioned online learning methods, our approach is not hierarchical and we do not compute intermediate tracklets because errors in the tracklets corrupt the learning data. Similarly to [2], we model a score of uncertainty but based on distance information only and not on the target model, since distance can not drift over time. This enables us to invoke appearance and other less stable features only when truly needed as in the case of missing detections, occluded objects or new tracks.

3. Related perception studies

The proposed method is inspired by the human cognitive ability to solve the tracking task. In fact, events such as eye movements, blinks and occlusions disrupt the input to our vision system, introducing challenges similar to the ones encountered in real world video sequences and detections. Perception psychologists have studied the mechanisms employed in our brain during multiple object tracking since the ’80s [14, 22, 1], though only recently RMI experiments have been used to confirm and validate proposed theories. One
of these preeminent theories is given in a seminal work by Kahneman, Treisman and Gibbs in 1992 [14]. They proposed the theory of Object Files to understand the dominant role of spatial information in preserving target identity. The theory highlights the central role of spatial information in a paradigm called Spatio-Temporal Dominance. Accordingly, target correspondence is computed on the basis of spatio-temporal continuity and does not consult non-spatial properties of the target. If spatio-temporal information is consistent with the interpretation of a continuous target, the correspondence will be established even if appearance features are inconsistent. “Up in the sky, look: It’s a bird. It’s a plane. It’s Superman!” - this well known quote, from the respective short animated movie (1941), suggests that the people pointing at Superman changed their visual perception of the target to the extent of giving him a completely different meaning, while they never had any doubt they kept referring to the same object. Nevertheless, when correspondence cannot be firmly established on the basis of spatial information, appearance, motion, and other complex features can be consulted as well. In particular, in [14] the tracking process is divided into a circular pipeline of three steps (Fig 2, top row). The correspondence uses only positional information and aims at establishing if detected objects are either a new target or an existing one appearing at a different location. The review activates when ambiguity in assignments arises, and recomputes uncertain target links by also taking into account more complex features. Eventually, the impletion is the final task to assess and induce the perception of targets temporal coherence.

4. The proposal

As depicted in Fig. 2, the proposed method relates the 3 steps of correspondence, review and impletion to a divide and conquer approach. Targets are divided in the where pathway by checking for incongruences in spatial coherence. Eventually, the tracking solution is conquered by associat-
5. Learning to divide

In this section, we propose a method to generate zones \( z \) and decide whether associations in those zones are simple \( z \in \mathcal{Z}_s \) or difficult \( z \in \mathcal{Z}_c \). A zone \( z \) can be defined as an heterogeneous set of tracks and detections characterized by spatial proximity. Even if simple, the concept of proximity may vary across sequences, and the importance of distances on each axis depends on targets dominant flows in the scene. Zones are computed through the Correlation Clustering (CC) method [3] on the cost matrix \( A \) suitably modified to obtain an affinity matrix \( \hat{A} \) as required by the CC algorithm. To move from cost features (distances) in \( A \) to affinity features in \( \hat{A} \), the cost features vector is augmented with their similarity counterpart and the affinity value is computed as the scalar product between this vector and a parameter vector \( \theta \):

\[
\hat{A}(i, j) = \theta^T (|t_x^i - d_x^j|, |t_y^i - d_y^j|, 1 - |t_x^i - d_x^j|, 1 - |t_y^i - d_y^j|)^T,
\]

\( \text{cost features} \quad \text{similarity features} \) (4)

where \( t^i \) and \( d^j \) are the \( i \)-th track and \( j \)-th detection respectively. The \( \theta \) vector has the triple advantage of weighting differently distances on each axis, avoiding to set thresholds in the affinity computation and controlling the compactness and the balancing of clusters. Further detail on learning \( \theta \) are provided in the following sections.

To prevent the creation of clusters composed only of detections or tracks, a symmetric version of \( \hat{A} \) is created having a zero block diagonal structure:

\[
\hat{A}_{\text{sym}} = \begin{bmatrix} 0 & \hat{A} \\ \hat{A}^T & 0 \end{bmatrix}
\]

(5)

Through this shrewdness, two tracks (detections) can be in the same cluster only if close to a common detection (track). The CC algorithm, applied on \( \hat{A}_{\text{sym}} \), efficiently partition the scene in a set of zones \( \mathcal{Z} \) so that the sum of the affinities between track-detection pairs in the same zone is maximized:

\[
\arg \max_{\mathcal{Z}} \sum_{z \in \mathcal{Z}} \sum_{(i,j) \in z} \hat{A}_{\text{sym}}(i, j).
\]

(6)

Eventually, a zone \( z \) is defined as simple if it contains an equal number of targets and detections, otherwise is complex. As previously stated, associations in a complex zone \( z \in \mathcal{Z}_c \) cannot be solved with the use of distance information only (Fig. 3b), but require more informative features to disambiguate the decision.

6. Learning to conquer

The divide mechanism brings the advantage of splitting the problem into smaller local subproblems. Associations belonging to simple zones can be independently solved trough any bipartite matching algorithm. The complete tracking problem must deal also with occluded target as well. We consider a target as occluded when it is not associated to a detection (e.g. a miss detection in frame \( k \) occurred, shaded people in Fig. 3). Since occluded targets are representation of disappeared objects, they are not included in the zones at the current frame. All the subproblems related to complex zones \( z \in \mathcal{Z}_c \) are consequently connected by sharing the whole set of occluded targets. In order to simultaneously solve the whole set of subproblems, we construct an augmented version of the matrix in Eq. (2) where the block \( H \) accounts for potential associations between occluded tracks and current detections:

\[
\hat{C} = \begin{bmatrix} \hat{A} & +\infty & T_{\text{out}} \\ H & H_{\text{occ}} & +\infty \\ D_{\text{in}} & \Xi & \Xi \end{bmatrix}.
\]

(7)

\( H_{\text{occ}} \) is a \( \xi \)-diagonal matrix (\( +\infty \) elsewhere) used to keep occluded tracks still occluded in the current frame. The
solution of the optimization problem in Eq. 1 on matrix \( \hat{C} \), obtained by applying the Hungarian algorithm, provides the final tracking associations for this frame.

More precisely, thanks to the peculiar block structure of \( C \) a single call to Hungarian results in solving the partitioned association problem in Eq. (3), subject to the constraint that each occluded element can be inserted in a single complex zone subproblem solution. In \( \hat{C} \), simple zones subproblems are isolated by setting the association cost outside the zone to \(+\infty\). Similarly, complex zones results in independent blocks as well, but are connected through the presence of occluded elements, i.e. non-infinite entries in \( H \).

By casting the problem using the cost matrix \( \hat{C} \), it is possible to learn, in a joint framework, to combine features in order to obtain a suitable cost for both the association (either in simple or complex zones) and the partition in zone as well. To this end we introduce a linear \( w \)-parametrization on \( A \) and \( H \) with a mask vector \( \pi_Z \) that selects the features according to the complexity of the belonging zone:

\[
\hat{C}(i, j) = w^T \pi_Z(i, j) \circ f(i, j),
\]

being \( \circ \) the Hadamard product. The feature vector contains both simple and complex information between the \( i \)-th track and the \( j \)-th detection:

\[
f(i, j)^T = \begin{cases} 
1, |t_x^i - d_x^j|, |t_y^i - d_y^j|, & \xi \text{ features for } x \in Z, \\
1 - |t_x^i - d_x^j|, 1 - |t_y^i - d_y^j|, & \text{features for divide step} \\
|t_x^i - d_x^j|, |t_y^i - d_y^j|, g_1(t^i, d^j), g_2(t^i, d^j), \ldots. & \text{features for } x \in Z_c 
\end{cases}
\]

where \( g_1, g_2, \ldots \) are distance functions between track \( i \) and detection \( j \) on complex features 1 and 2 respectively. Precisely, \( \pi_Z \) selectively activates features according to the following rules:

\[
\pi_Z(i, j)^T = \begin{cases} 
(0, 1, 1, 0, 0, 0, 0, 0, 0, \ldots) & \text{if (a)} \\
(0, 0, 0, 0, 1, 1, 1, 1, \ldots) & \text{if (b)} \\
(\infty, 0, 0, 0, 0, 0, 0, 0, 0, \ldots) & \text{if (c)}
\end{cases}
\]

where the pair target-detection in \( \hat{C}_Z(i, j) \) may (a) belong to the same simple zone, (b) be composed by elements belonging to complex zones and (c) have elements belonging to different zones.

The feature vector \( f(i, j) \) is computed only on pairs of (possibly occluded) tracks and detections. To extend the parametrization to the whole matrix \( \hat{C} \), it is sufficient to set \( \pi_Z = (1, 0, 0, \ldots)^T \) outside \( \hat{A} \) and \( H \). Analogously, for elements \( \hat{C}(i, j) \) outside \( \hat{A} \) or \( H \), we set

\[
f(i, j) = (\infty, 0, 0, \ldots)^T \text{ and } f(i, j) = (1, 0, 0, \ldots)^T \text{ when } \hat{C}(i, j) = +\infty \text{ and } \hat{C}(i, j) = \xi \text{ respectively.}
\]

The learning procedure in Sec. 7 computes the best weight vector \( w \) and consequently \( \xi \) is learnt as a bias term. Recall that \( \xi \) governs tracks initiation and termination. Eq. (3) becomes a linear combination of the weights \( w \) and a feature map \( \Phi \):

\[
h(T, D_k; w) = \arg \max_{y, Z} -w^T \sum_{i=1}^n \pi_Z(i, y^i) \circ f(i, y^i) = \arg \max_{y, Z} w^T \Phi(T, D_k, y, Z).
\]

The feature map \( \Phi \) is a function evaluating how well the set of zones \( Z \) and the proposed tracking solution \( y \) for frame \( k \) fit on the input data \( T \) and \( D_k \).

Given a set of weights \( w \), the tracking problem in Eq. (11) can be solved by first computing the zones \( Z \) through the divide step on matrix \( \hat{A}_{sym} \) of Eq. (5) and then by conquering the associations in each zone through the Hungarian method on matrix \( \hat{C} \). Note that now \( \hat{A}_{sym}(i, j) = w^T (0, 1, 1, 1, 0, 0, \ldots)^T \circ f(i, j) \) and \( \theta \) is a subset of \( w \).

### 7. Online subgradient optimization

The problem of Eq. (11) requires to identify the complex structured object \((y, Z) \in Y \times Z\) such that \( Z \) is the set of zones that best explain the \( k \)-th frame tracking solution \( y \) for an input \((T, D_k)\). Zones \( z \in Z \) are modelled as latent variables, since they remain unobserved during training. To this end, we learn the weight vector \( w \) in \( h(T, D_k; w) \) through Latent Structural SVM [26] by solving the following unconstrained optimization problem over the training set \( S = \{(T, D_k, y_k)\}_{k=1}^K \):

\[
\min_w \frac{\lambda}{2} ||w||^2 + \frac{1}{K} \sum_{k=1}^K \tilde{H}_k(w),
\]

with \( \tilde{H}_k(w) \) being the structured hinge-loss. \( \tilde{H}_k(w) \) results from solving the loss-augmented maximization problem

\[
\tilde{H}_k(w) = \max_{y, Z} H_k(y, Z; w) = \max_{y, Z} \Delta_k(y, Z) - \langle w, \psi_k(y, Z) \rangle,
\]

where \( \Delta_k(y, Z) = \Delta(y_k, Z_k, y, Z) \) is a loss function that measures the error of predicting the output \( y \) instead of the correct output \( y_k \) while assuming \( Z \) to hold instead of \( Z_k \), and we defined \( \psi_k(y, Z) = \Phi(T, D_k, y_k, Z_k) - \Phi(T, D_k, y, Z) \) for notation convenience.

Solving Eq. (13) is equivalent to finding the output-latent pair \((y, Z)\) generating the most violated constraint, for a given input \((T, D_k)\) and a latent setting \(Z_k\). Despite the generality of the learning framework, the loss function \( \Delta \) is
Algorithm 1 Block-Coordinate Primal-Dual Frank-Wolfe Algorithm for learning w on a sequence of K frames

1: Let $w^{(0)} \leftarrow 0, w_k^{(0)} \leftarrow 0, l^{(0)} \leftarrow 0, i_k^{(0)} \leftarrow 0$ for $k = 1, \ldots, K$
2: for $k \leftarrow 1$ to $K$ do
3: Compute simple features for learning to divide Eq. (9)
4: **Latent completion**: $Z_k = \arg \max_Z w^T \Phi(T, D_k, y_k, Z)$ through Correlation Clustering on $A_{\text{sym}}$ of Eq. (5)
5: Compute complex features for learning to conquer Eq. (9)
6: **Max Oracle**: $(\hat{y}_k, \hat{Z}_k) = \arg \max_{y, Z} H_k(y, Z; w)$ through Hungarian on Eq. (14)
7: Let $w_s \leftarrow \frac{1}{\lambda_k} \psi_k(\hat{y}, \hat{Z})$ and $l_s \leftarrow \frac{1}{\gamma} \Delta_k(\hat{y}, \hat{Z})$
8: Let $\gamma \leftarrow [\lambda (w_k^{(r)} - w_s)^T w_k^{(r)} - l_k^{(r)} + l_s]/[\lambda \|w_k^{(r)} - w_s\|^2]$ and clip to [0, 1]
9: Update $w_k^{(r+1)} \leftarrow (1 - \gamma) w_k^{(r)} + \gamma w_s$ and $i_k^{(r+1)} \leftarrow (1 - \gamma) i_k^{(r)} + \gamma l_s$
10: Update $w^{(r+1)} \leftarrow w^{(r)} + w_k^{(r+1)} - w_k^{(r)}$ and $l^{(r+1)} \leftarrow l^{(r)} + i_k^{(r+1)} - i_k^{(r)}$
11: end for

Figure 4: Thanks to the choice of the Hamming loss, the maximization oracle is reduced to an assignment problem efficiently solved through the Hungarian algorithm, as for the inference step.

Problem dependent and must be accurately chosen. In particular, we adopted the Hamming loss function that, substituted in Eq. (13), behaves linearly making the maximization oracle solvable as a standard assignment problem, Fig. 4b:

$$\hat{H}_k(w) = \max_{y, Z} \sum_{i=1}^{n} 1\{y_i^k \neq y^i\} + w^T \pi_Z (i, y^i) \circ f(i, y^i)$$

where $\Phi(T, D_k, y_k, Z_k)$ was dropped as not dependent on either $y$ or $Z$.

The learning step of Eq. (12) can be efficiently solved online, under the premise that the dual formulation of LSSVM results in a continuously differentiable convex objective after latent completion. We designed a modified version of the Block-Coordinate Frank-Wolfe algorithm [16] presented in Alg. 1. The main insight here is to notice that the linear subproblem employed by Frank-Wolfe (line 5) is equivalent to the loss-augmented decoding subproblem of Eq. (14), which can be solved efficiently through the Hungarian algorithm [15]. To deal with latent variables during optimization, we added the latent completion process (line 4) where, given an input/output pair, the latent variable $Z_k$ which best explains the solution $y_k$ to the observed data is found. Through the latent completion step, the objective function optimized by Frank-Wolfe has guarantees to be convex.

8. Experimental results

In this section we present two different experiments that highlight the improvement of our method over state of the art trackers in static camera sequences. The first experiment is devoted to stress the method in clutter scenarios where moderate crowd occurs and our divide and conquer approach gives its major benefits in terms of both computational speed and performances. The second experiment is on the publicly available MOT Challenge dataset that is becoming a standard for tracking by detection comparison. Test were evaluated employing the CLEAR MOT [6] measures and trajectory based measures (MT,ML,FRG) as suggested in [20]. All the detections, where not provided by authors, have been computed using the method in [10] as suggested by the protocol in [20]. Results are averaged per experiment in order to have a quick glimpse on the tracker performances. Individual sequences results are provided in the additional material. To train the parameters acting on the complex zones, the LSSVM have been trained with ground truth (GT) trajectories and the addition of different levels of random noise simulating miss and false detections. In all the tests, occluded objects locations are updated in time using a Kalman Filter with a constant velocity state transition model, and discarded if not reassigned after 15 frames.

8.1. Features

The strength of the proposal is the joint LSSVM framework that learns to weight features for both partitioning the scene and associating targets. On these premises, we purposely adopted standard features. Without loss of generality, the method can be expanded through additional and more complex features as well. The features always refer to a single detection $d \in D_k$ and a single track $t \in T$, occluded or not, and its associated history, in compliance with Eq. (9).

In the experiments, the appearance of the targets is modeled through a color histogram in the RGB space. Every time a new detection is associated to a track, its appearance
information is stored in the track history. The appearance feature $g_1$ is then computed as the average value of the Kullback-Leibler distance of the detection histogram from track previous instances. Additionally, we designed tracks to contain their full trajectories over time. By disposing of the trajectories, we modeled the motion coherence $g_2$ of a detection w.r.t a track by evaluating the smoothness of the manifold fitted on the joint set of the new detected point and the track spatial history. More precisely, given a detected point, an approximate value of the Ricci curvature is computed by considering only the subset of detections of the trajectory lying inside a given neighborhood of the detected point. An extensive presentation of this feature is in [11].

8.2. Datasets and Settings

Midly Crowded Dataset (MCD): the dataset is a collection of moderately crowded videos taken from both public benchmarks with the addition of ad-hoc sequences. This dataset consists of 4 sequences: the well-known PETS09 S2L2 and S2L3 sequences, and 2 new sequences. GVEII is characterized by a high number of pedestrian crossing the scene (up to 107 people per frame), while 1shatian3, captured by [27], is a sequence characterized by a high density and clutter (up to 227 people per frame). A single training stage was performed by gathering the first 30% of each video. These frames have not been used at test time.

MOT Challenge: the dataset consists of several public available sequences in different scenarios. Detections and annotations are provided by the MOTChallenge website. In our test we consider the subset of the sequences coming from fixed cameras since distances are not meaningful in the moving camera settings: TUD-Crossing, PETS09-S2L2, AVG-TownCentre, ADL-Rundle-3, KITTI-16 and Venice-1. Learning was performed on a distinct set of sequences provided on the website for training.

8.3. Comparative evaluation

Results on MCD: Quantitative results of our proposal on the MCD dataset compared with the state of the art trackers are presented in Tab. 2, while visual results are in Fig. 5. We compared against two very recent online methods [21, 2] that focus either on target motion or appearance. Moreover, the offline method [19] has been considered being one of the most effective MTT methods up to now. In the MCD challenging sequences, we outperform the competitors in terms of MT values having also the lowest number of IDS and FRAG. This is basically due to the selective use of the proper features depending on the outcomes of the divide phase of our algorithm. This solution allows our tracker to take the best of both worlds against [21] and [2]. MOTA measure is higher as well testifying the overall quality of the proposed tracking scheme. Additionally, in Fig. 6 we reported the track lenght curves (TL) on the MCD dataset. TL curve is computed by considering the length of the correctly tracked GT trajectories plotted in descending order. The plot

Table 2: Averaged results on MCD. In the appearance column, w.n. is when needed. More details on the light gray baselines in the text.

<table>
<thead>
<tr>
<th>Method</th>
<th>MOTA</th>
<th>MOTP</th>
<th>MT</th>
<th>ML</th>
<th>FF</th>
<th>FN</th>
<th>IDS</th>
<th>FRG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LDCT</td>
<td>43.1</td>
<td>74.5</td>
<td>9</td>
<td>10</td>
<td>68.2</td>
<td>2780</td>
<td>161</td>
<td>187</td>
</tr>
<tr>
<td>LDCT (all features)</td>
<td>✓</td>
<td>40.6</td>
<td>66.3</td>
<td>61</td>
<td>43</td>
<td>446</td>
<td>193</td>
<td></td>
</tr>
<tr>
<td>LDCT (only simple)</td>
<td>✓</td>
<td>36.4</td>
<td>64.7</td>
<td>58</td>
<td>50</td>
<td>586</td>
<td>276</td>
<td></td>
</tr>
<tr>
<td>Bar and Yun [2]</td>
<td>39.0</td>
<td>65.8</td>
<td>84</td>
<td>55</td>
<td>637</td>
<td>289</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Possegger et al. [21]</td>
<td>38.7</td>
<td>65.0</td>
<td>79</td>
<td>37</td>
<td>455</td>
<td>440</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Milan et al. [19]</td>
<td>40.6</td>
<td>66.7</td>
<td>64</td>
<td>42</td>
<td>242</td>
<td>141</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Offline</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MotCon</td>
<td>32.0</td>
<td>70.6</td>
<td>2</td>
<td>30</td>
<td>777</td>
<td>3280</td>
<td>110</td>
<td>105</td>
</tr>
<tr>
<td>SegTrack</td>
<td>32.3</td>
<td>72.1</td>
<td>3</td>
<td>38</td>
<td>520</td>
<td>3454</td>
<td>80</td>
<td>76</td>
</tr>
<tr>
<td>CEM</td>
<td>28.1</td>
<td>71.2</td>
<td>5</td>
<td>24</td>
<td>1256</td>
<td>3088</td>
<td>87</td>
<td>97</td>
</tr>
<tr>
<td>SMOJ</td>
<td>23.9</td>
<td>71.7</td>
<td>2</td>
<td>27</td>
<td>706</td>
<td>3627</td>
<td>120</td>
<td>208</td>
</tr>
<tr>
<td>TID</td>
<td>28.0</td>
<td>71.3</td>
<td>3</td>
<td>25</td>
<td>1233</td>
<td>3083</td>
<td>192</td>
<td>193</td>
</tr>
<tr>
<td>DP.NMS</td>
<td>22.7</td>
<td>71.4</td>
<td>3</td>
<td>17</td>
<td>1062</td>
<td>3052</td>
<td>529</td>
<td>325</td>
</tr>
</tbody>
</table>

Table 3: Averaged results of our method (LDCT) and the other MOT Challenge competitors on the 6 fixed camera sequences. See: http://www.motchallenge.net for detailed results.
gives information on the ability of the tracker to build continuous and correct tracks for all the ground truth elements in the scene, neglecting the amount of false tracks inserted. Our AUC is always greater than competitors’ thanks to the adoption of complex zones that effectively deals with occluded/disappeared objects and keep the tracks longer.

To evaluate the improvement due to the adoption of the divide and conquer steps, which is the foundation of our tracker, in Tab. 2 we also test two baselines: when either all features or spatial features only were used for all the assignments independently of the zone type. In both tests, the divide step, the parameter learning and occlusion handling remain as previously described. Improvement of the complete method (dark gray) over these baselines (light gray) suggests that complex features are indeed more beneficial when used selectively.

Results on MOT Challenge: Tab 3 summarizes the accuracy of our method compared to other state of the art algorithms on the MOT Challenge dataset. Similarly to the MCD experiment, we observe that our algorithm outperforms the other state of the art methods. Our method achieves best results in most of the metrics, keeping IDS and FRG relatively low as well. In turn, our method records the highest MOTA compared to others with a significant margin (+10%). Excellent results on this dataset highlight the generalization ability of our method, which was trained on sequences different (although similar) from the ones in the test evaluation. Fig. 5 shows some qualitative examples of our results. Furthermore, our online tracker has been designed to perform considerably fast. We report an average performances of 10 fps on the MOT Challenge sequences. The runtime is strongly influenced by the number of detections as well as by the number of tracks created up to a specific frame. The performances are in line or faster than the majority of the current methods that report an average of 3-5 fps.

The computational complexity of solving Eq. (1) using the Hungarian algorithm is \( O(N^3) \) with \( N \) the number of tracks and detections to be associated and \( N_o \) the number of occluded tracks. Since the complexity of the divide step is linear in the number of targets, our algorithm reduced the assignment complexity to \( N O\left(\frac{N_o}{2}\right) + O(N^\beta + N_o)^3 \). The first term applies for simple zones and is linear in \( N \) being dominated by \( \alpha \) that is the average number of detections in every partition (\( \alpha << N \)). The second term modulates the complexity of the association algorithm in complex zones by the \( \beta \) factor, i.e. is the percentage of complex zones in the scene. Eventually the \( N_o \) term is related to the recall of the chosen detector. As an example \( N_o \) can be realistically set to 0.3\( N \) and, if the percentage of complex zones \( \beta \) is 10%, the algorithm is 50× faster than its original counterpart.

9. Conclusion

In this work, we proposed an enhanced version of the Hungarian online association model to match recent features advancement and cope with different sequences peculiarities. The algorithm is able to learn to effectively partition the scene and choose the proper feature combination to solve simple and complex association in an online fashion. As observed in the experiments, the benefits of our divide and conquer approach are evident in terms of both computational complexity of the problem and tracking accuracy.

The proposed tracking framework can be extended/enriched with a different set of simple and complex features and it can learn to identify the relevant ones for the specific scenario\(^1\). This can open a major room for improvement by allowing the community to test the method with more complex and sophisticated features. We invite the reader to download the code and to test it by adding her favorite features.

\(^1\)Although analogy with cognitive theory holds for spatial features only.
Appendix A: Block-Coordinate Frank-Wolfe optimization of Latent Structural SVM

In a recent paper by Lacoste-Julien et al. [16] the efficient use of Block-Coordinate Frank-Wolfe optimization for the training of structural SVM was demonstrated. They noted that by disposing of a maximization oracle, subgradient methods could be applied to solve the non-smooth unconstrained problem of Eq. (15). The notation follows the one used in the paper.

\[
\min_{\mathbf{w}} \frac{\lambda}{2} \|\mathbf{w}\|^2 + \frac{1}{K} \sum_{k=1}^{K} \hat{H}_k(\mathbf{w}),
\]

where \(\hat{H}_k(\mathbf{w})\) is exactly the optimal value for the necessary max oracle. The Lagrange dual of the above \(K\)-slack formulation of Eq.(15) has \(n = \sum_{k} |\mathcal{Y}_k|\) potential support vectors. Writing \(\alpha_k(y)\) for the dual variables associated with the \(k\)-th training example and potential output \(y\), the dual problem is given by

\[
\min_{\alpha \in [0,1]^n} f(\alpha) = \frac{\lambda}{2} \|A\alpha\|^2 - \mathbf{b}^T \alpha
\]

\[
\text{s.t. } \sum_{y \in \mathcal{Y}_k} \alpha_k(y) = 1, \forall k \in \mathbb{N}_K
\]

Here matrix \(A\) and vector \(\mathbf{b}\) are constructed by simple Lagrangian derivation. The only two requirements that need to be satisfied in order to apply Frank-Wolfe algorithm on the problem of Eq.(16) are:

- the domain \(\mathcal{M}\) of \(f(\alpha)\) has to be compact, and
- the convex objective \(f\) has to be continuously differentiable.

Observe that the domain \(\mathcal{M}\) of Eq. (16) is the product of \(K\) probability simplices, \(\mathcal{M} = \Delta_{|\mathcal{Y}_1|} \times \cdots \times \Delta_{|\mathcal{Y}_k|}\). It is thus compact by the geometrical definition of simplex. We now present matrix \(A\) and vector \(\mathbf{b}\) for the latent formulation of SSVM and check that \(f\) is continuously differentiable. Recalling that for LLSVM the loss augmented decoding subproblem is expressed as

\[
\hat{H}_k(\mathbf{w}) = \max_{y,z} H_k(y, z; w)
\]

\[
= \max_{y,z} \Delta_k(y, z) - \langle w, \psi_k(y, z) \rangle,
\]

and omitting the lagrangian dual derivation as it is a simple mathematical procedure, we obtain \(A_{d \times m}\) to be composed of a set of \(m\) columns:

\[
A(i, \cdot) = \left\{ \frac{\psi_k(y, z)(i)}{\lambda K}, \forall k \in \mathbb{N}_K, y \in \mathcal{Y}_k, z \in \mathbf{Z}_k \right\},
\]

where \(i\) indicates the row index. Analogously, the column vector \(b_{m \times 1\)} is built as follows.

\[
\mathbf{b} = \left\{ \frac{L_k(y, z)}{K}, \forall k \in \mathbb{N}_K, y \in \mathcal{Y}_k, z \in \mathbf{Z}_k \right\}
\]

The function \(f\) is now differentiated by

\[
\nabla f(\alpha) = \lambda A^T A \alpha - \mathbf{b}^T = \lambda A^T \mathbf{w} - \mathbf{b}^T,
\]

where \(w = A\alpha\) is the stationarity KKT condition that has to hold in order to make the duality strong. By substituting the definition of \(A\) and \(\mathbf{b}\) for specific values of \((k, y, z)\) we obtain

\[
\nabla f(\alpha)_{k,y,z} = \frac{\lambda}{\lambda K} \psi_k(y, z)^T \mathbf{w} - \frac{1}{K} \Delta_k(y, z)
\]

\[
= -\frac{1}{K} \left[ \Delta_k(y, z) - \psi_k(y, z)^T \mathbf{w} \right]
\]

\[
= -\frac{1}{K} \Delta_k(y, z).
\]

Which is the same hinge loss of Eq. (17). So once again the intuition that the linear subproblem that the Frank-Wolfe algorithm has to solve is strongly connected to the loss augmented decoding subproblem is true. Nevertheless, as opposed to the non-latent case, here the hinge loss is also dependent on the latent variable \(z\) which makes the problem non-convex. Thus, before computing the \(H_k(y, Z)\), a latent completion step (line 4 of Alg. 1) is needed in order to ensure \(f\) to be continuously differentiable over all the domain except for a finite number of points (sufficient condition). Once we attend these precautions, the latent formulation reduces to the standard SSVM case, and as such, all convergence results also apply to the latent case.

Appendix B: Computational complexity details

In this section we provide some details on the computational complexity results we presented in the paper. Recall that if the problem of data association is tackled as a whole, the complexity of finding a perfect minimum matching amounts to

\[
\mathcal{O}((N + N_o)^3),
\]

according to widely used Hungarian implementations, where \(N\) is the number of currently active tracks and detections and \(N_o\) is the number of occluded tracks that can still be considered for association.

When considering our method, we have to consider two distinct contributions to the complexity of the overall algorithm:

- divide which is accomplished through a correlation clustering (CC) step on \(N\) elements.
• conquer or the application of the Hungarian to the generated sub-problems.

The division step can be linear or even sublinear with respect to the number of elements to be split $N$. This can be obtained through the many recent approximate solutions of the CC (which is NP-hard optimally), e.g. [17, 8]. The conquer step has to be evaluated considering that complex and simple zones are solved differently. Simple zones result in independent subproblems that can be solved directly through the Hungarian. Conversely, the complex zones are not independent and occluded targets have to be considered as well during the overall association subproblem.

Let $K$ be the number of clusters created by the CC and suppose a uniform partition of tracks and detections among these clusters. To simplify the notation, let us call $\alpha = \frac{N}{K}$ the average number of elements (tracks and detections) inside a zone. If $N$ is the overall number of active tracks and detections, each cluster has approximately $\frac{N}{2K}$ tracks and $\frac{N}{2K}$ detections that need to be associated. Independently of the zone, these two quantities must coincide for the zone to be simple, so the complexity of solving a simple zone is $O\left(\frac{N}{2K}^3\right) = O\left(\frac{1}{2}\right)^3$. Note that for large $N$, $\alpha$ will typically be much smaller than $N$. If simple clusters are a fraction $\beta K$ of the overall number of clusters $K$, the final complexity of solving simple zones is $\beta N O\left(\frac{1}{2}\right)^3$ which reduces to $\beta N O\left(\frac{1}{2}\right)^3$ in the worst case hypotheses. Note that these subproblem can be solved in parallel and typically $\alpha << N$ when $N$ is large. This is because $\alpha$, the number of interfering tracks/detections, is limited by the non-maxima suppression response of a detector.

Complex zones have to be solved altogether due to the $N_o$ shared occluded tracks. If $\beta K$ is the number of simple groups, the number of complex groups is $(1 - \beta)K$ or $\beta K$ for notation convenience. Since the number of tracks and detections are not equal anymore, the number of rows/columns to consider for each group is $\frac{N}{K}$. We thus obtain a complexity of $O\left(\beta K \frac{N}{K}^3\right)$ and due to the addition of occluded targets, the complexity increases to $O\left((\beta N + N_o)^3\right)$. Summing up we obtain that the overall complexity of the conquer step is

$$ (1 - \beta) N O\left(\frac{\alpha}{2}\right)^3 + O\left((\beta N + N_o)^3\right). $$

Note that when $\beta = 0$, i.e. all zones are simple, only the first term matters; while if $\beta = 1$ than the contribution of the first term vanishes and the second term reduces to a stand hungarian over all the tracks/detections as in Eq. (22).

Appendix C: Kalman filtering

In order to propagate tracks position over occlusions, we employed a simple Kalman Filter predictor with a constant velocity measurement model. This basically means that while unobserved, tracks keep moving by assuming their velocity will not change over time. More formally, the standard discrete Kalman Filter formulation, when no input is considered, is:

$$ x(k) = A x(k - 1) + w(k - 1) $$

$$ z(k) = H x(k) + v(k), $$

being the first equation the state equation and the second one the measurement equation. Here $x(k)$ represents the state of a track at time $k$, while $z(k)$ its measured position. $H$ is the matrix which relates these two variables, namely relates the state and the measurement. $A$ is called state space matrix and explain how the model should evolve over time by means of its physical intrinsic peculiarities. $v(k)$ and $w(k - 1)$ are the measurement and state noise random variables. During occlusions, the observation cannot be directly measured so we need to rely on the second relation $z(k) = H x(k) + v(k)$, and cannot correct the model. A state $x$ for a track is usually represented by a four dimensional vector containing its position and velocity as follows:

$$ x(k) = [x_x(k), x_y(k), \dot{x}_x(k), \dot{x}_y(k)]^T $$

To describe a constant velocity linear model, we need to specify $A$ and $H$ as follows:

$$ A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} $$

By substituting the equations, and ignoring the noises just for the sake of simplicity, we obtain the measurement vector:

$$ z(k) = [x_x(k - 1) + \dot{x}_x(k - 1), x_y(k - 1) + \dot{x}_y(k - 1)]^T, $$

which corresponds exactly to a constant velocity model due to the identity 2x2 submatrix in the lower right corner of $A$.

Appendix D: Detailed experimental results

In the paper we had to omit some detail on the experimental results. Due to space limitations we presented the results only averaged over the whole sequence set of the Mildly Crowded Dataset (MCD) and the fixed camera sequences of the MOT Challenge (MOT) benchmark. In Tab. 4 and Tab. 5 we report per sequence results. In particular, for MCD we also report results for the considered competitors; while for the MOT benchmark results are reported for our method only and we let the reader refer to the benchmark site for competitors results: [http://motchallenge.net/results_detail](http://motchallenge.net/results_detail).

References

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Method</th>
<th>onl.</th>
<th>app.</th>
<th>MOTA (%)</th>
<th>MOTP (%)</th>
<th>GT</th>
<th>MT</th>
<th>ML</th>
<th>FP</th>
<th>FN</th>
<th>IDS</th>
<th>FRG</th>
</tr>
</thead>
<tbody>
<tr>
<td>PETS09-S2-L2</td>
<td>LDCT (our)</td>
<td>✓</td>
<td>w.n.</td>
<td>47.4</td>
<td>70.8</td>
<td>6</td>
<td>3</td>
<td>63</td>
<td>3</td>
<td>297</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td></td>
<td>42 pedestrian</td>
<td>✓</td>
<td>✓</td>
<td>41.3</td>
<td>69.7</td>
<td>4</td>
<td>7</td>
<td>44</td>
<td>7</td>
<td>411</td>
<td>425</td>
<td></td>
</tr>
<tr>
<td></td>
<td>up to 33 for frame</td>
<td>✓</td>
<td>✓</td>
<td>35.7</td>
<td>68.8</td>
<td>3</td>
<td>9</td>
<td>40</td>
<td>3</td>
<td>497</td>
<td>323</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bae and Yun [2]</td>
<td>✓</td>
<td>✓</td>
<td>30.2</td>
<td>69.2</td>
<td>1</td>
<td>8</td>
<td>32</td>
<td>2</td>
<td>284</td>
<td>499</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Possegger et al. [21]</td>
<td>✓</td>
<td></td>
<td>40.0</td>
<td>68.6</td>
<td>8</td>
<td>3</td>
<td>21</td>
<td>11</td>
<td>211</td>
<td>432</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Milan et al. [19]</td>
<td></td>
<td></td>
<td>44.9</td>
<td>70.2</td>
<td>5</td>
<td>6</td>
<td>15</td>
<td>5</td>
<td>150</td>
<td>165</td>
<td></td>
</tr>
<tr>
<td>PETS09-S2-L3</td>
<td>LDCT (our)</td>
<td>✓</td>
<td>w.n.</td>
<td>35.2</td>
<td>66.7</td>
<td>6</td>
<td>15</td>
<td>24</td>
<td>22</td>
<td>120</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td></td>
<td>44 pedestrian</td>
<td>✓</td>
<td>✓</td>
<td>30.6</td>
<td>65.1</td>
<td>1</td>
<td>20</td>
<td>31</td>
<td>6</td>
<td>235</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td></td>
<td>up to 42 for frame</td>
<td>✓</td>
<td>✓</td>
<td>26.1</td>
<td>63.2</td>
<td>1</td>
<td>25</td>
<td>31</td>
<td>6</td>
<td>235</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bae and Yun [2]</td>
<td>✓</td>
<td>✓</td>
<td>28.8</td>
<td>62.3</td>
<td>8</td>
<td>17</td>
<td>96</td>
<td>150</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Possegger et al. [21]</td>
<td>✓</td>
<td></td>
<td>32.2</td>
<td>64.1</td>
<td>5</td>
<td>12</td>
<td>79</td>
<td>111</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Milan et al. [19]</td>
<td></td>
<td></td>
<td>31.3</td>
<td>64.6</td>
<td>7</td>
<td>23</td>
<td>71</td>
<td>56</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GVEII</td>
<td>LDCT (our)</td>
<td>✓</td>
<td>w.n.</td>
<td>65.6</td>
<td>73.5</td>
<td>208</td>
<td>63</td>
<td>285</td>
<td>71</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>630 pedestrian</td>
<td>✓</td>
<td>✓</td>
<td>55.6</td>
<td>70.5</td>
<td>172</td>
<td>101</td>
<td>548</td>
<td>320</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>up to 107 for frame</td>
<td>✓</td>
<td>✓</td>
<td>50.9</td>
<td>67.9</td>
<td>151</td>
<td>113</td>
<td>755</td>
<td>418</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bae and Yun [2]</td>
<td>✓</td>
<td>✓</td>
<td>57.9</td>
<td>71.1</td>
<td>200</td>
<td>75</td>
<td>1023</td>
<td>320</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Possegger et al. [21]</td>
<td>✓</td>
<td></td>
<td>51.1</td>
<td>69.8</td>
<td>153</td>
<td>98</td>
<td>844</td>
<td>652</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Milan et al. [19]</td>
<td></td>
<td></td>
<td>49.3</td>
<td>71.2</td>
<td>147</td>
<td>87</td>
<td>312</td>
<td>244</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ishitian3</td>
<td>LDCT (our)</td>
<td>✓</td>
<td>w.n.</td>
<td>42.6</td>
<td>61.0</td>
<td>133</td>
<td>23</td>
<td>137</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>239 pedestrian</td>
<td>✓</td>
<td>✓</td>
<td>34.7</td>
<td>59.9</td>
<td>68</td>
<td>45</td>
<td>592</td>
<td>154</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>up to 227 for frame</td>
<td>✓</td>
<td>✓</td>
<td>32.8</td>
<td>58.9</td>
<td>79</td>
<td>52</td>
<td>776</td>
<td>301</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bae and Yun [2]</td>
<td>✓</td>
<td>✓</td>
<td>38.9</td>
<td>60.7</td>
<td>150</td>
<td>40</td>
<td>1146</td>
<td>185</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Possegger et al. [21]</td>
<td>✓</td>
<td></td>
<td>31.5</td>
<td>57.4</td>
<td>110</td>
<td>35</td>
<td>686</td>
<td>654</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Milan et al. [19]</td>
<td></td>
<td></td>
<td>36.8</td>
<td>60.8</td>
<td>98</td>
<td>50</td>
<td>435</td>
<td>98</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Comparison of the proposed method (dark grey) with the state of the art methods on the MCD dataset. In the appearance column, w.n. means when needed. For each sequence, we also run our code by always associating based on the whole feature set and simple features only (light grey baselines).

<table>
<thead>
<tr>
<th>Sequence</th>
<th>MOTA (%)</th>
<th>MOTP (%)</th>
<th>GT</th>
<th>MT</th>
<th>ML</th>
<th>FP</th>
<th>FN</th>
<th>IDS</th>
<th>FRG</th>
</tr>
</thead>
<tbody>
<tr>
<td>TUD-Crossing</td>
<td>67.7</td>
<td>82.9</td>
<td>13</td>
<td>9</td>
<td>1</td>
<td>100</td>
<td>205</td>
<td>161</td>
<td>30</td>
</tr>
<tr>
<td>PETS09-S2L2</td>
<td>47.4</td>
<td>70.8</td>
<td>42</td>
<td>6</td>
<td>3</td>
<td>995</td>
<td>3779</td>
<td>187</td>
<td>201</td>
</tr>
<tr>
<td>AVG-TownCentre</td>
<td>31.7</td>
<td>72.2</td>
<td>226</td>
<td>36</td>
<td>24</td>
<td>1878</td>
<td>2608</td>
<td>395</td>
<td>509</td>
</tr>
<tr>
<td>ADL-Rundle-3</td>
<td>25.2</td>
<td>73.4</td>
<td>44</td>
<td>2</td>
<td>27</td>
<td>453</td>
<td>7039</td>
<td>110</td>
<td>101</td>
</tr>
<tr>
<td>KITTI-16</td>
<td>53</td>
<td>79</td>
<td>17</td>
<td>2</td>
<td>3</td>
<td>91</td>
<td>665</td>
<td>44</td>
<td>57</td>
</tr>
<tr>
<td>Venice-1</td>
<td>33.5</td>
<td>68.4</td>
<td>17</td>
<td>0</td>
<td>5</td>
<td>578</td>
<td>2384</td>
<td>73</td>
<td>129</td>
</tr>
</tbody>
</table>

Table 5: Per sequence results of our method on the MOT Challenge fixed camera sequences. Last row contains mean values and is the one reported in the paper for comparison. Refer to the benchmark website (http://motchallenge.net/results_detail) for competitors detailed results.


