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 [9, 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 [17, 5, 19, 2, 8, 22]. 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 [17, 5, 8] 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, 22] to be the solution, while others argue that this complexity may affect the long-term robustness [21]. For instance, in large crowds people appearance is rarely informative. As a consequence, tracking robustness is often achieved by focusing on spatial features [19], finding them more reliable than visual ones.

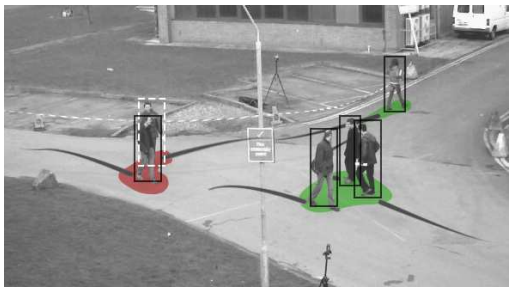


Figure 1: The scene is partitioned in local zones. Green zones is where the same number of tracks and detections are present. Red zones, where miss and false detections (white dashed contours) are discovered and solving the associations may call for complex appearance or motion features.

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 [11], 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, 17, 12, 16] 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 [16, 12, 23]. 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 [22] 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.

		C	A	T	L	M
Offline methods						
Berclaz *et al.* [5]	2011	✓				
Milan *et al.* [17]	2014	✓	✓	✓		✓
Hoffman *et al.* [12]	2013		✓	✓	✓	
Li *et al.* [16]	2009			✓	✓	
Online methods						
Yang and Nevatia [23]	2014			✓	✓	
Breitenstein *et al.* [7]	2009		✓			
Bae and Yoon [2]	2014	✓	✓			✓
Possegger *et al.* [19]	2014	✓				
Wu *et al.* [22]	2013		✓			
Our proposal	2015	✓	½		✓	✓

Table 1: Overview of offline and online related works in terms of code availability (C), appearance models (A), tracklets computation (T), associations learning (L) and presence in the MOT Challenge competition (M). In our method, use of appearance set to ½ means only when needed.

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.* [19] 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 [13, 20, 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 [13]. They proposed the theory of Object Files to understand the dominant

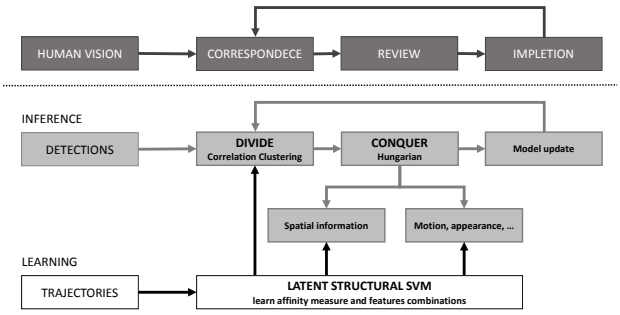


Figure 2: First row shows the human tracking process according to Kahneman, Treisman and Gibbs theory [13]. Below a schematic view of the inference and learning steps underpinning our method.

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 [13] 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.

Our proposal relies on a similar scheme but re-designed in a larger context to deal with a multitude of targets as in the case of MTT problem.

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 associating coherent elements in the *where* (spatial) domain and incoherent ones in the *what* (visual) domain.

The core of the proposal is twofold. First, a method to divide potential associations between detections and tracks into local clusters or zones. A zone can be either simple or complex, calling for different features to complete the association. Targets can be directly associated to their closest detections if they are inside a simple zone (*e.g.* when we have the same number of tracks and detections, green area in Fig. 3b). Conversely, targets inside complex areas (red in Fig. 3b) are subject to a deeper evaluation where appearance, motion and other features may be involved.

Second, we cast the problems of splitting potential associations and solving them by selecting and weighting the features inside a unified structural learning framework that aims at the best set of partitions and adapts from scene to scene.

4.1. Problem formulation

Online MTT is typically solved by optimizing, at frame k, a generic assignment function for a set of tracks $\mathcal{T}$ and current detections $\mathcal{D}_k$:

$$h(\mathcal{T}, \mathcal{D}_k) = \arg\min_{\mathbf{y}} \sum_{i=1}^{n} \mathbf{C}(i, \mathbf{y}^i), \tag{1}$$

where $\mathbf{y}$ is a permutation vector of $\{1, 2, \ldots, n\}$ and $\mathbf{C} \in \mathbb{R}^{n \times n}$ is a cost matrix. The cost matrix $\mathbf{C}$ is designed to include dummy rows and columns to account for new detected objects ($\mathbf{D}_{\text{in}}$) or leaving targets ($\mathbf{T}_{\text{out}}$). More formally, if matrix $\mathbf{A} : \mathcal{T} \times \mathcal{D}_k \rightarrow \mathbb{R}$ contains association costs for currently tracked targets and detections, the cost matrix is:

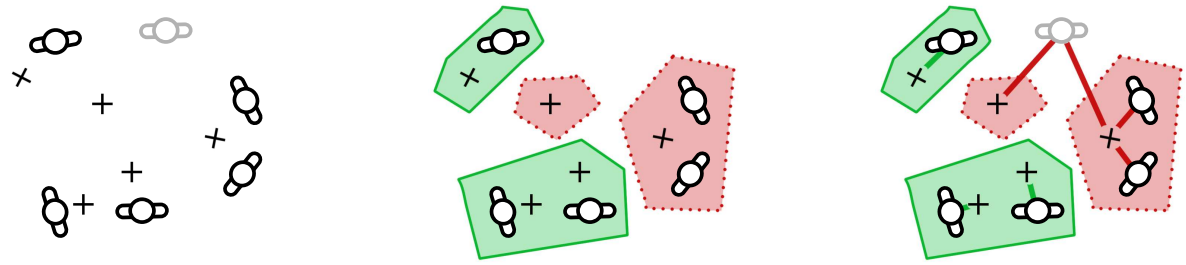
$$\mathbf{C} = \begin{bmatrix} \mathbf{A} & \mathbf{T}_{\text{out}} \\ \mathbf{D}_{\text{in}} & \boldsymbol{\Xi} \end{bmatrix} \tag{2}$$

where $\mathbf{D}_{\text{in}}$, $\mathbf{T}_{\text{out}}$ contain the cost ξ of creating a new track on the diagonal and $+\infty$ elsewhere. Similarly, $\boldsymbol{\Xi}$ is a full matrix of value ξ.

The formulation in Eq. (1) evaluates all the associations through the same cost function, built upon a preferred set of features. In order to consider different cost functions for specific subsets of associations, we reformulate Eq. (1) as:

$$h(\mathcal{T}, \mathcal{D}_k) = \arg\min_{\mathbf{y}, \mathcal{Z}} \sum_{\substack{(i,\mathbf{y}^i) \in \mathbf{z} \\ \mathbf{z} \in \mathcal{Z}_{\text{s}}}} \mathbf{C}_{\mathbf{s}}(i, \mathbf{y}^i) + \sum_{\substack{(i,\mathbf{y}^i) \in \mathbf{z} \\ \mathbf{z} \in \mathcal{Z}_{\text{c}}}} \mathbf{C}_{\mathbf{c}}(i, \mathbf{y}^i) \tag{3}$$

where we explicit the different contribution of trivial and difficult associations, whose costs are given by the functions $\mathbf{C}_{\mathbf{s}}$ and $\mathbf{C}_{\mathbf{c}}$ respectively. Associations are locally partitioned in zones $\mathbf{z} \in \mathcal{Z}$ as shown in Fig. 3b. Hereinafter, we seamlessly refer to a zone $\mathbf{z}$ as a portion of the scene or the set of detections and tracks that lie onto it. A zone can be simple $\mathbf{z} \in \mathcal{Z}_{\text{s}}$ or complex to solve $\mathbf{z} \in \mathcal{Z}_{\text{c}}$ depending on the set of associations it involves.



(a) frame input (b) *divide* associations (c) *conquer* associations

Figure 3: Overview of the inference procedure. (a) In the image targets are represented by bird eye view sketches (shaded when occluded) and detections by crosses. (b) In the *divide* step detections and non-occluded targets are spatially clustered into zones. A zone with an equal number of targets and detections is simple (solid green contours), complex otherwise (dashed red contours). (c) Associations in simple zones are independently solved by means of distance features only. Complex zones are solved by considering more complex features such as appearance or motion and accounting for potentially occluded targets, which are shared across all the complex zones.

5. Learning to divide

In this section, we propose a method to generate zones $\mathbf{z}$ and decide whether associations in those zones are simple $\mathbf{z} \in \mathcal{Z}_s$ or difficult $\mathbf{z} \in \mathcal{Z}_c$. A zone $\mathbf{z}$ 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 $\mathbf{A}$ suitably modified to obtain an affinity matrix $\bar{\mathbf{A}}$ as required by the CC algorithm. To move from cost features (distances) in $\mathbf{A}$ to affinity features in $\bar{\mathbf{A}}$, 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 $\boldsymbol{\theta}$:

$$\bar{\mathbf{A}}(i,j) = \boldsymbol{\theta}^T(\underbrace{|t_x^i - d_x^j|, |t_y^i - d_y^j|}_{\text{cost features}}, \underbrace{1 - |t_x^i - d_x^j|, 1 - |t_y^i - d_y^j|}_{\text{similarity features}})^T, \tag{4}$$

where t^i and d^j are the i-th track and j-th detection respectively. The $\boldsymbol{\theta}$ 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 $\boldsymbol{\theta}$ are provided in the following sections.

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

$$\bar{\mathbf{A}}_{\text{sym}} = \begin{bmatrix} \mathbf{0} & \bar{\mathbf{A}} \\ \bar{\mathbf{A}}^T & 0 \end{bmatrix} \tag{5}$$

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 $\bar{\mathbf{A}}_{\text{sym}}$, 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:

$$\arg\max_{\mathcal{Z}} \sum_{\mathbf{z} \in \mathcal{Z}} \sum_{(i,j) \in \mathbf{z}} \bar{\mathbf{A}}_{\text{sym}}(i,j). \tag{6}$$

Eventually, a zone $\mathbf{z}$ 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 $\mathbf{z} \in \mathcal{Z}_c$ 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 $\mathbf{z} \in \mathcal{Z}_c$ 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 $\mathbf{H}$ accounts for potential associations between occluded tracks and current detections:

$$\widehat{\mathbf{C}} = \begin{bmatrix} \hat{\mathbf{A}} & +\infty & \mathbf{T}_{\text{out}} \\ \mathbf{H} & \mathbf{H}_{\text{occ}} & +\infty \\ \mathbf{D}_{\text{in}} & \boldsymbol{\Xi} & \boldsymbol{\Xi} \end{bmatrix}. \tag{7}$$

$\mathbf{H}_{\text{occ}}$ is a ξ-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 $\widehat{\mathbf{C}}$, obtained by applying the Hungarian algorithm, provides the final tracking associations for this frame.

More precisely, thanks to the peculiar block structure of $\widehat{\mathbf{C}}$ 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 $\widehat{\mathbf{C}}$, 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 $\mathbf{H}$.

By casting the problem using the cost matrix $\widehat{\mathbf{C}}$, 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 $\mathbf{w}$-parametrization on $\widehat{\mathbf{A}}$ and $\mathbf{H}$ with a mask vector $\boldsymbol{\pi}_{\mathcal{Z}}$ that selects the features according to the complexity of the belonging zone :

$$\widehat{\mathbf{C}}(i,j) = \mathbf{w}^T \boldsymbol{\pi}_{\mathcal{Z}}(i,j) \circ \mathbf{f}(i,j), \tag{8}$$

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

$$\begin{aligned}\mathbf{f}(i,j)^T = \; & (\underbrace{1}_{\xi}, \underbrace{|t_x^i - d_x^j|, |t_y^i - d_y^j|}_{\text{features for } \mathbf{z} \in \mathcal{Z}_s}, \\ & \underbrace{1-|t_x^i - d_x^j|, 1-|t_y^i - d_y^j|}_{\text{features for } \textit{divide} \text{ step}}, \\ & \underbrace{|t_x^i - d_x^j|, |t_y^i - d_y^j|, g_1(t^i,d^j), g_2(t^i,d^j), \dots}_{\text{features for } \mathbf{z} \in \mathcal{Z}_c}).\end{aligned} \tag{9}$$

where $g_1, g_2, \dots$ are distance functions between track i and detection j on complex features 1 and 2 respectively. Precisely, $\boldsymbol{\pi}_{\mathcal{Z}}$ selectively activates features according to the following rules:

$$\boldsymbol{\pi}_{\mathcal{Z}}(i,j)^T = \begin{cases} (\;0,1,1,0,0,0,0,0,0,0,\dots) & \text{if (a)} \\ (\;0,0,0,0,0,1,1,1,1,1,\dots) & \text{if (b)} \\ (\infty,0,0,0,0,0,0,0,0,0,\dots) & \text{if (c)} \end{cases} \tag{10}$$

where the pair target-detection in $\widehat{\mathbf{C}}_{\mathcal{Z}}(i,j)$ 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 $\mathbf{f}(i,j)$ is computed only on pairs of (possibly occluded) tracks and detections. To extend the parametrization to the whole matrix $\widehat{\mathbf{C}}$, it is sufficient to set $\boldsymbol{\pi}_{\mathcal{Z}} = (1,0,0,\dots)^T$ outside $\widehat{\mathbf{A}}$ and $\mathbf{H}$. Analogously, for elements $\widehat{\mathbf{C}}(i,j)$ outside $\widehat{\mathbf{A}}$ or $\mathbf{H}$, we set $\mathbf{f}(i,j) = (\infty,0,0,\dots)^T$ and $\mathbf{f}(i,j) = (1,0,0,\dots)^T$ when $\widehat{\mathbf{C}}(i,j) = +\infty$ and $\widehat{\mathbf{C}}(i,j) = \xi$ respectively. The learning procedure in Sec. 7 computes the best weight vector $\mathbf{w}$ and consequently ξ is learnt as a bias term. Recall that ξ governs tracks initiation and termination. Eq. (3) becomes a linear combination of the weights $\mathbf{w}$ and a *feature map* Φ:

$$\begin{aligned} h(\mathcal{T}, \mathcal{D}_k; \mathbf{w}) &= \arg\max_{\mathbf{y},\mathcal{Z}} -\mathbf{w}^T \sum_{i=1}^{n} \boldsymbol{\pi}_{\mathcal{Z}}(i,\mathbf{y}^i) \circ \mathbf{f}(i,\mathbf{y}^i) \\ &= \arg\max_{\mathbf{y},\mathcal{Z}} \mathbf{w}^T \Phi(\mathcal{T}, \mathcal{D}_k, \mathbf{y}, \mathcal{Z}). \end{aligned} \tag{11}$$

The feature map Φ is a function evaluating how well the set of zones $\mathcal{Z}$ and the proposed tracking solution $\mathbf{y}$ for frame k fit on the input data $\mathcal{T}$ and $\mathcal{D}_k$.

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

7. Online subgradient optimization

The problem of Eq. (11) requires to identify the complex structured object $(\mathbf{y}, \mathcal{Z}) \in \mathcal{Y} \times \mathbf{Z}$ such that $\mathcal{Z}$ is the set of zones that best explain the k-th frame tracking solution $\mathbf{y}$ for an input $(\mathcal{T}, \mathcal{D}_k)$. Zones $\mathbf{z} \in \mathcal{Z}$ are modelled as latent variables, since they remain unobserved during training. To this end, we learn the weight vector $\mathbf{w}$ in $h(\mathcal{T}, \mathcal{D}_k; \mathbf{w})$ through Latent Structural SVM [24] by solving the following unconstrained optimization problem over the training set $\mathcal{S} = \{(\mathcal{T}, \mathcal{D}_k, \mathbf{y}_k)\}_{k=1\dots K}$:

$$\min_{\mathbf{w}} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{K}\sum_{k=1}^{K} \tilde{H}_k(\mathbf{w}), \tag{12}$$

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

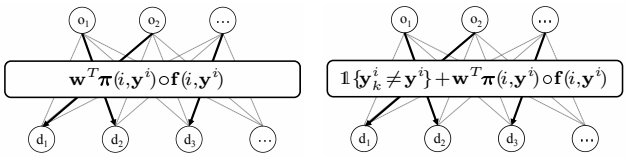
$$\begin{aligned} \tilde{H}_k(\mathbf{w}) &= \max_{\mathbf{y},\mathcal{Z}} H_k(\mathbf{y}, \mathcal{Z}; \mathbf{w}) \\ &= \max_{\mathbf{y},\mathcal{Z}} \Delta_k(\mathbf{y}, \mathcal{Z}) - \langle \mathbf{w}, \psi_k(\mathbf{y}, \mathcal{Z}) \rangle, \end{aligned} \tag{13}$$

where $\Delta_k(\mathbf{y}, \mathcal{Z}) = \Delta(\mathbf{y}_k, \mathcal{Z}_k, \mathbf{y}, \mathcal{Z})$ is a loss function that measures the error of predicting the output $\mathbf{y}$ instead of the correct output $\mathbf{y}_k$ while assuming $\mathcal{Z}$ to hold instead of $\mathcal{Z}_k$, and we defined $\psi_k(\mathbf{y}, \mathcal{Z}) = \Phi(\mathcal{T}, \mathcal{D}_k, \mathbf{y}_k, \mathcal{Z}_k) - \Phi(\mathcal{T}, \mathcal{D}_k, \mathbf{y}, \mathcal{Z})$ for notation convenience.

Solving Eq. (13) is equivalent to finding the output-latent pair $(\mathbf{y}, \mathcal{Z})$ generating the most violated constraint, for a given input $(\mathcal{T}, \mathcal{D}_k)$ and a latent setting $\mathcal{Z}_k$. Despite the generality of the learning framework, the loss function Δ is

Algorithm 1 Block-Coordinate Primal-Dual Frank-Wolfe Algorithm for learning $\mathbf{w}$ on a sequence of K frames

1: Let $\mathbf{w}^{(0)} \leftarrow \mathbf{0}, \mathbf{w}_k^{(0)} \leftarrow \mathbf{0}, l^{(0)} \leftarrow 0, l_k^{(0)} \leftarrow 0$ for $k = 1, \dots, K$
2: **for** $k \leftarrow 1$ **to** K **do**
3: Compute simple features for learning to *divide* Eq. (9)
4: **Latent completion**: $\mathcal{Z}_k = \arg\max_{\mathcal{Z}} \mathbf{w}^T \Phi(\mathcal{T}, \mathcal{D}_k, \mathbf{y}_k, \mathcal{Z})$ through *Correlation Clustering* on $\bar{\mathbf{A}}_{\text{sym}}$ of Eq. (5)
5: Compute complex features for learning to *conquer* Eq. (9)
6: **Max Oracle**: $(\bar{\mathbf{y}}_k, \bar{\mathcal{Z}}_k) = \arg\max_{\mathbf{y},\mathcal{Z}} H_k(\mathbf{y}, \mathcal{Z}; \mathbf{w})$ through *Hungarian* on Eq. (14)
7: Let $\mathbf{w}_s \leftarrow \frac{1}{\lambda K} \psi_k(\bar{\mathbf{y}}, \bar{\mathcal{Z}})$ and $l_s \leftarrow \frac{1}{n} \Delta_k(\bar{\mathbf{y}}, \bar{\mathcal{Z}})$
8: Let $\gamma \leftarrow [\lambda(\mathbf{w}_k^{(r)} - \mathbf{w}_s)^T \mathbf{w}^{(r)} - l_k^{(r)} + l_s] / [\lambda \|\mathbf{w}_k^{(r)} - \mathbf{w}_s\|^2]$ and clip to $[0, 1]$
9: Update $\mathbf{w}_k^{(r+1)} \Leftarrow (1-\gamma)\mathbf{w}_k^{(r)} + \gamma \mathbf{w}_s$ and $l_k^{(r+1)} \Leftarrow (1-\gamma) l_k^{(r)} + \gamma l_s$
10: Update $\mathbf{w}^{(r+1)} \Leftarrow \mathbf{w}^{(r)} + \mathbf{w}_k^{(r+1)} - \mathbf{w}_k^{(r)}$ and $l^{(r+1)} = l^{(r)} + l_k^{(r+1)} - l_k^{(r)}$
11: **end for**



(a) Inference Step (b) Maximization Oracle

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:

$$\tilde{H}_k(\mathbf{w}) = \max_{\mathbf{y},\mathcal{Z}} \sum_{i=1}^{n} \mathbb{1}\{\mathbf{y}_k^i \neq \mathbf{y}^i\} + \mathbf{w}^T \boldsymbol{\pi}_{\mathcal{Z}}(i, \mathbf{y}^i) \circ \mathbf{f}(i, \mathbf{y}^i) \tag{14}$$

where $\Phi(\mathcal{T}, \mathcal{D}_k, \mathbf{y}_k, \mathcal{Z}_k)$ was dropped as not dependent on either $\mathbf{y}$ or $\mathcal{Z}$.

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 [15] 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 [14]. To deal with latent variables during optimization, we added the latent completion process (line 4) where, given an input/output pair, the latent variable $\mathcal{Z}_k$ which best explain the solution $\mathbf{y}_k$ 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 [18]. All the detections, where not provided by authors, have been computed using the method in [9] as suggested by the protocol in [18]. 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 reassociated 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 $\mathbf{d} \in \mathcal{D}_k$ and a single track $\mathbf{t} \in \mathcal{T}$, 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

Figure 5: Tracking results on PETS09-S2L3, 1shatian3 and GVEII from the MCD dataset (top row). AVG-TownCentre, ADL-Rundle-3 and Venice-1 from the MOT Challenge sequences (bottom). Next to images, simple (green) and complex (red) zones are displayed.

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 [10].

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 [25], 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

	app	MOTA	MOTP	MT	ML	IDS	FRG
LDCT	*w.n.*	**47.7**	**68.8**	**88**	**26**	**209**	**103**
LDCT (all features)	✓	40.6	66.3	61	43	446	193
LDCT (only simple)		36.4	64.7	58	50	586	276
Bae and Yun [2]	✓	39.0	65.8	84	35	637	289
Possegger *et al.* [19]		38.7	65.0	79	37	455	440
Milan *et al.* [17]		40.6	66.7	64	42	242	141

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

	MOTA	MOTP	MT	ML	FP	FN	IDS	FRG
Online								
LDCT	**43.1**	**74.5**	**9**	**10**	682	**2780**	161	187
RMOT	30.4	70.2	2	27	1011	3259	74	125
TC_ODAL	24.2	70.9	1	31	1047	3528	75	152
Offline								
MotiCon	32.0	70.6	2	30	777	3280	110	105
SegTrack	32.3	72.1	3	38	520	3454	80	76
CEM	28.1	71.2	5	24	1256	3088	87	97
SMOT	23.9	71.7	2	27	706	3627	120	208
TBD	28.0	71.3	3	25	1233	3083	192	193
DP_NMS	22.7	71.4	3	17	1062	3052	529	325

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.

are presented in Tab. 2, while visual results are in Fig. 5. We compared against two very recent online methods [19, 2] that focus either on target motion or appearance. Moreover, the offline method [17] 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 [19] 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

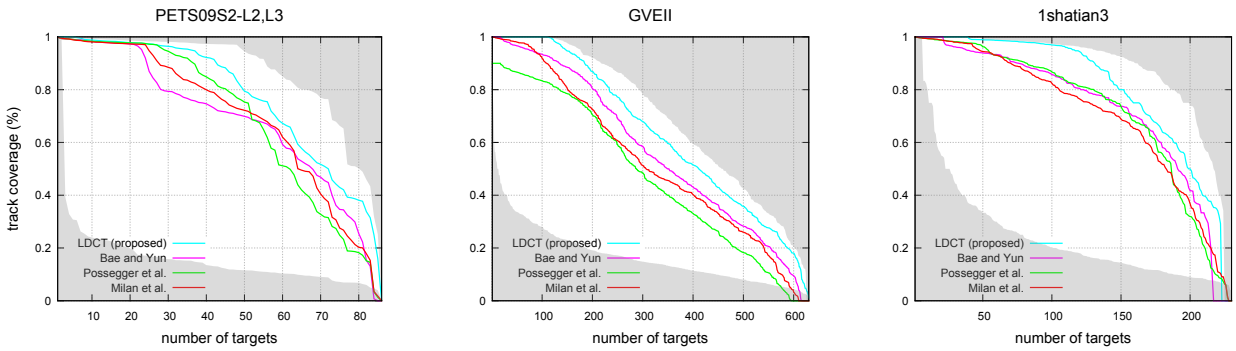


Figure 6: Tracks length curves (TL) on MCD sequences. The gray shaded area indicates the performances reached by a simple global NN algorithm (lower bound) and the highest score obtained for each track combining all different methods results (upper bound).

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/disapperared 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 $\mathcal{O}(N+N_o)^3$ 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 $N\mathcal{O}(\frac{\alpha}{2}) + \mathcal{O}(N\beta + N_o)^3$. The first term applies for simple zones and is linear in N being dominated by α 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 β 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 $0.3N$ and, if the percentage of complex zones β is 10%, the algorithm is $50\times$ 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.

References

[1] G. A. Alvarez and S. L. Franconeri. How many objects can you track?: Evidence for a resource-limited attentive tracking

[1] Although analogy with cognitive theory holds for spatial features only.

mechanism. *Journal of Vision*, 7(13), Oct. 2007. 2

[2] S.-H. Bae and K.-J. Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1218–1225, June 2014. 1, 2, 7

[3] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Machine Learning*, 56:89–113, July 2004. 4

[4] R. Benenson, M. Omran, J. Hosang, and B. Schiele. Ten years of pedestrian detection, what have we learned? In *European Conference on Computer Vision Workshops*, pages 613–627. 2015. 1

[5] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1806–1819, Sept. 2011. 1, 2

[6] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008(1):246309, 2008. 6

[7] M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1515–1522, Sept 2009. 2

[8] C. Dicle, O. Camps, and M. Sznaier. The way they move: Tracking multiple targets with similar appearance. In *2013 IEEE International Conference on Computer Vision (ICCV)*, pages 2304–2311, Dec. 2013. 1

[9] P. Dollar, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1532–1545, Aug. 2014. 1, 6

[10] D. Gong, X. Zhao, and G. G. Medioni. Robust multiple manifold structure learning. In *ICML*, 2012. 7

[11] M. A. Goodale and A. Milner. Separate visual pathways for perception and action. *Trends in Neurosciences*, 15(1):20 – 25, 1992. 2

[12] M. Hofmann, M. Haag, and G. Rigoll. Unified hierarchical multi-object tracking using global data association. In *2013 IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, pages 22–28, Jan. 2013. 2

[13] D. Kahneman, A. Treisman, and B. J. Gibbs. The reviewing of object files: Object-specific integration of information. *Cognitive Psychology*, 24(2):175–219, Apr. 1992. 2, 3

[14] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, Mar. 1955. 6

[15] S. Lacoste-Julien, M. Jaggi, M. Schmidt, and P. Pletscher. Block-coordinate frank-wolfe optimization for structural SVMs. In *International Conference on Machine Learning*, 2013. 6

[16] 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. 2

[17] A. Milan, S. Roth, and K. Schindler. Continuous energy minimization for multitarget tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1):58–72, Jan. 2014. 1, 2, 7

[18] L. Milan, A. Leal-Taix, K. Schindler, S. Roth, and I. Reid. MOT Challenge. http://www.motchallenge.net, 2014. 6

[19] H. Possegger, T. Mauthner, P. M. Roth, and H. Bischof. Occlusion geodesics for online multi-object tracking. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1306–1313, June 2014. 1, 2, 7

[20] Z. Pylyshyn. The role of location indexes in spatial perception: a sketch of the FINST spatial-index model. *Cognition*, 32(1):65–97, June 1989. 2

[21] A. Smeulders, D. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual tracking: An experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1442–1468, July 2014. 1

[22] Z. Wu, J. Zhang, and M. Betke. Online motion agreement tracking. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2013. 1, 2

[23] B. Yang and R. Nevatia. Multi-target tracking by online learning a crf model of appearance and motion patterns. *Int. J. Comput. Vision*, 107(2):203–217, Apr. 2014. 2

[24] C.-N. J. Yu and T. Joachims. Learning structural SVMs with latent variables. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1169–1176, New York, NY, USA, 2009. ACM. 5

[25] B. Zhou, X. Tang, H. Zhang, and X. Wang. Measuring crowd collectiveness. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(8):1586–1599, Aug 2014. 7