Discriminative Label Propagation for Multi-Object Tracking with Sporadic Appearance Features

Amit Kumar K.C. and Christophe De Vleeschouwer
ISPGroup, ELEN Department, ICTEAM Institute
Université catholique de Louvain
Louvain-la-Neuve, B-1348, Belgium
{amit.kc, christophe.devleeschouwer}@uclouvain.be

Abstract

Given a set of plausible detections, detected at each time instant independently, we investigate how to associate them across time. This is done by propagating labels on a set of graphs that capture how the spatio-temporal and the appearance cues promote the assignment of identical or distinct labels to a pair of nodes. The graph construction is driven by the locally linear embedding (LLE) of either the spatio-temporal or the appearance features associated to the detections. Interestingly, the neighborhood of a node in each appearance graph is defined to include all nodes for which the appearance feature is available (except the ones that coexist at the same time). This allows to connect the nodes that share the same appearance even if they are temporally distant, which gives our framework the uncommon ability to exploit the appearance features that are available only sporadically along the sequence of detections.

Once the graphs have been defined, the multi-object tracking is formulated as the problem of finding a label assignment that is consistent with the constraints captured by each of the graphs. This results into a difference of convex program that can be efficiently solved. Experiments are performed on a basketball and several well-known pedestrian datasets in order to validate the effectiveness of the proposed solution.

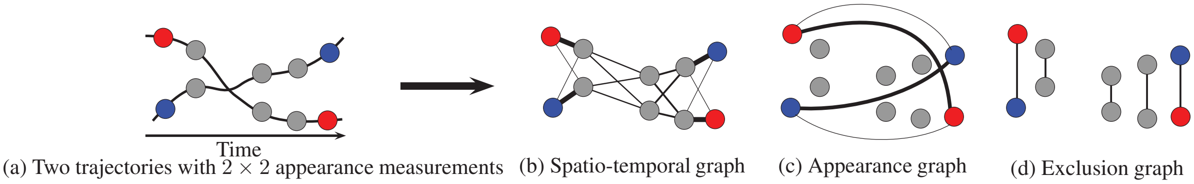
1. Introduction

In this paper, we address the problem of multi-object tracking. We assume that the targets have been detected at each time instant and their appearance features (if available) have been extracted. Then, our objective is to link these detections into consistent trajectories using a graph-based formalism.

A graph-based formalism assigns a node to each detection. Edges are then defined to connect the nodes, and each edge gets a cost that reflects the dissimilarity between the two nodes, it connects. Afterwards, a (K)-shortest path algorithm [10] is typically used to find the trajectories of the (K) targets. Alternatively, other approaches use network flow [21], maximum weighted independent set [12], etc. to solve the same problem. These approaches have been proven to be effective in scenarios for which the features are collected with the same level of accuracy and reliability for each detection. With such a stationary measurement process, the likelihood that the detections along a path correspond to the same physical object can be reasonably estimated based on the accumulation of dissimilarities (similarities) between consecutive nodes in the path. In contrast, these approaches are not appropriate in cases for which appearance features cannot be measured with same accuracy and reliability in every space and time co-ordinates. Such problems are prevalent in many real-life situations. For example, color histograms tend to be noisy in presence of occlusions. In some cases, highly discriminative features are available only sporadically. This happens, for example, while imaging biological cells in varying illuminations in which each illumination level highlights certain features of the cell. As another example, a digit, printed on the jersey of a player, is available only when it faces the camera. In such cases, the task of tracking multiple objects, while exploiting such features, becomes non-trivial.

Recently, there have been some efforts to address this problem. In [23], the authors assume that a discrete set of L possible appearances is known beforehand, which allows the creation of a L-layered graph. In the i-th layer, running through a node is penalized when the appearance of the node is available and differs from the i-th presumed appearance. Afterwards, a K-shortest path algorithm is applied in order to find the K shortest paths across L-layers. This method demonstrates that exploiting sporadic features can significantly improve the tracking performance. How-

Part of this work is funded by Belgian NSF (FNRS) and Walloon region project SPORTIC.



(a) Two trajectories with 2×2 appearance measurements (b) Spatio-temporal graph (c) Appearance graph (d) Exclusion graph

Figure 1. **Best viewed in color.** (a) An example with two targets (red and blue) with associated detections at each time. Gray detections mean that no appearance feature is available. (b) Spatio-temporal graph that depicts the spatio-temporal association between the nodes, (c) Appearance graph that connects nodes even if they are far in time. (d) Exclusion graph in which edges connect nodes that coexist at the same time. In graphs (b) and (c), thickness of an edge is proportional to its weight.

ever, it is restricted to cases for which the number and the appearance of the targets are known *a priori*.

In contrast, [5] proposes an iterative hypothesis testing strategy to exploit appearances features that are corrupted by non-stationary noise or are only sporadically available. In short, the authors iteratively consider each node in the graph as a key-node, and investigate how to link this key-node with other nodes in its neighborhood, under the assumption that the target appearance is defined by the key-node appearance. This is done through a shortest-path computation in the temporal neighbourhood, while promoting the nodes that have an appearance similar to that of the key-node. This not only allows to handle the cases for which the discrete set of possible appearance is not known, but also alleviates the construction of the L-layered graph. The greedy and iterative nature of the algorithm makes it computationally more efficient. Its main disadvantage is that it is greedy and consequently there is no guarantee about some global optimality of the solution.

In this paper, we propose an alternate formulation of the problem, for which an optimal solution can be computed and does not require to know the possible appearances beforehand. We adopt a graph-based label propagation framework. For this, we construct a number of distinct graphs, one for each appearance feature, apart from the usual spatio-temporal graph. Additionally, we also construct an exclusion graph in order to reflect the fact that two detections that occur at the same time should be assigned to distinct labels. Therefore, we construct $K + 2$ graphs (one spatio-temporal, K appearance, one exclusion), where $K \ll L$ is the number of appearance features. An example is shown in Figure 1. In case of a sport game, for example, the jersey color and the digit, printed on it, can be considered as two appearance features, and result in two distinct appearance graphs. The framework is scalable in that it allows incorporating as many appearance features as needed. Graph construction is described in Section 2.1. In short, a node is assigned to each detection. An edge connects two nodes and has a weight that increases with the similarity between the nodes in terms of space, time and appearance. The higher the weight, the more likely the two nodes correspond to the same physical target. Exceptionally, in case of the exclusion graph, the weights among the nodes, which occur at the same time, are equal.

Given these graphs, the tracking problem is then formulated as finding a label assignment that jointly and consistently labels the nodes. Here, the notion of consistent labelling means that (i) the nodes that are sufficiently close in space and time are labelled similarly, (ii) the nodes that are close (respectively, far) in appearance are labelled similarly (respectively, differently), and (iii) the nodes that coexist at the same time are labelled differently. The quality of labelling is measured by the labelling error, that accumulates the difference in the labels between a node and other nodes that are connected to it. If the nodes are more likely to have similar labels, then the labelling error is small and vice versa. Due to the definition of weights in our graph, a good labelling should minimize the labelling error in the spatio-temporal and the appearance graphs while maximizing the error due to the exclusion graph.

The rest of the paper is organized as follows: the construction of graphs and the formulation of the problem are presented in Section 2. A brief review of the related work is presented in Section 3. Experimental results are presented in Section 4. Finally, Section 5 concludes our paper.

2. Algorithm

This section first describes the construction of the associated graphs. Afterwards, the multi-object tracking is formulated as a consistent labelling problem in the graphs. Finally, the optimal solution to the proposed formulation is presented.

2.1. Graph construction

As presented earlier, we consider three distinct types of the graphs. Hence, each graph should be constructed separately. Nevertheless, the constructions of spatio-temporal and appearance graphs are similar. We derive those graphs from the locally linear embedding (LLE) technique [22]. It assumes that data points can be accurately reconstructed by a weighted combination of their local neighbors. If two data points are close (respectively, far) in some space, then the weight to reconstruct/approximate one point from the other is high (respectively, low). The number of neighbors, n, is a design parameter, and should thus be chosen according to the type of the graph.

In the following, we represent the i-th data point by $\boldsymbol{x}_i$ and its n-neighbours by $\boldsymbol{X}^{(i)} = (\boldsymbol{x}_1^{(i)}, ..., \boldsymbol{x}_n^{(i)})$. As $\boldsymbol{x}_i$ and n depend on the type of the graph, they should be defined separately for each graph. Afterwards, the graph construction can be formulated as the problem of finding the vector of reconstruction weights $\boldsymbol{w}_i^\star$ that minimizes following optimization problem

$$\begin{array}{ll} \text{minimize} & \left\| \boldsymbol{x}_i - \boldsymbol{X}^{(i)} \boldsymbol{w}_i \right\|_2^2 + \frac{\delta}{2} \| \boldsymbol{w}_i \|_2^2 \\ \text{subject to} & \mathbf{1}^\top \boldsymbol{w}_i = 1, \boldsymbol{w}_i \succeq \mathbf{0}. \end{array} \tag{1}$$

where $\delta > 0$ ensures that the objective function is strongly convex resulting in a unique $\boldsymbol{w}_i^\star$. We use $\delta = 10^{-2}$. The reason to choose the weights to be non-negative and to sum to unity is two-fold. On the one hand, the matrix $W = \left(\boldsymbol{w}_1, ..., \boldsymbol{w}_{|V|}\right)^\top$ can be considered as a transition matrix of a Markov chain and thus it has a random walk interpretation. On the other hand, the weight vector is sparse.

Once the weights for each data point are computed, we gather them into a graph $G = (V, E, W)$, where

- V is the set of nodes, with i-th node corresponding to the i-th sample. We denote the number of nodes by $|V|$.
- $E \subseteq V \times V$ defines the connectivity between the samples. We create an edge from node j to node i if $W_{ij} > 0$.
- W assigns a weight to each edge, W_{ij} being equal to the j-th component of $\boldsymbol{w}_i^\star$, and being non-zero when the j-th data point contributes to the approximation of the i-th data point in Equation 1.

Now, we explain the specific issues in the construction of each graph.

Spatio-temporal graph. In the case of the spatio-temporal graph, the data point $\boldsymbol{x}_i$ is defined by the time instant t_i and the location information (*e.g.*, bounding box) $\boldsymbol{c}_i$ as $\boldsymbol{x}_i = (\gamma t_i, \boldsymbol{c}_i)^\top$, where γ affects the relative importance of the time difference compared to the location difference between the data points. We use $\gamma = 3$. The neighbors $\boldsymbol{X}^{(i)}$ are defined to be the samples whose time indices fall within a temporal window of size $T > 1$, centered around t_i. Therefore, n depends on T. The temporal window T makes the system robust to missed detections. Since the window includes the samples from both the past and the future, a linear motion model is implicitly embedded.

Appearance graph. In the case of the appearance graph, a data point $\boldsymbol{x}_i$ corresponds to an appearance feature (*e.g.* color histograms, etc.). Since we are considering the fact that a feature might occur only sporadically, we consider all other samples as the neighbors, except the ones that cooccur with the i-th sample.

Exclusion graph. The exclusion graph captures the constraint associated to the fact that the detections that occur at the same time instant should have different labels. Hence, the neighborhood of the i-th sample for the exclusion graph is defined to comprise all other samples that occur at the same time instant, and their weights are taken uniformly, *i.e.*, $W_{ij} = \mathbf{1}/n$ if j is neighbor of i and $W_{ij} = 0$ otherwise, where n is the size of the neighborhood so defined,

2.2. Tracking problem formulation

In this section, we formulate the multi-object tracking as a consistent labelling problem in a set of associated graphs.

Given a graph $G = (V, E, W)$, we consider a label assignment $Y = (\boldsymbol{y}_1, ..., \boldsymbol{y}_{|V|})^\top$ that assigns a label distribution $\boldsymbol{y}_i \in [0,1]^{|V|}$ to each node i. It should be noted that Y is a row-stochastic matrix, with each row summing to unity. Therefore, we write $Y \in \mathcal{P}$, where $\mathcal{P}$ is the set of all row-stochastic matrices of size $|V| \times |V|$. In the following, we often refer to $\mathcal{P}$ as the probability simplex.

In order to measure the inconsistency between the labels with respect to the graph G, we adopt the harmonic function approach, introduced in [28], and define the labelling error as

$$\mathcal{E}_G(Y) = \frac{1}{2} \sum_{i=1}^{|V|} \sum_{j \in \mathcal{N}_i} W_{ij} \parallel \boldsymbol{y}_i - \boldsymbol{y}_j \parallel^2 = \mathbf{Tr}(Y^\top L Y), \tag{2}$$

where $\mathbf{Tr}$ is the trace of a matrix, $\mathcal{N}_i$ is the neighbourhood of node i, and $L = D - W$ is the graph Laplacian where $D = \mathbf{diag}(d_1, d_2, ..., d_{|V|})$ is a diagonal matrix with its diagonal elements defined as $d_i = \sum_{j \in \mathcal{N}_i} W_{ij}$. By the definition of our graphs, we have $D = I$, where I is $|V| \times |V|$ identity matrix. From spectral graph theory, L is positive-semi definite and therefore the labelling error $\mathbf{Tr}(Y^\top L Y)$ is convex in Y.

In the sequel, we frequently refer to a graph by its Laplacian L. In our framework, we have $K + 2$ distinct graphs. We represent the exclusion graph by $L^{(-)}$, and other graphs by $L_p^{(+)}, p = 0, ..., K$, where $p = 0$ corresponds to the spatio-temporal graph and $1 \leq p \leq K$ corresponds to the p-the appearance graph. We explicitly introduce the minus (respectively, plus) superscript in order to emphasize that we would like to maximize (respectively, minimize) the labelling error on the corresponding graph.

Given the measure of labelling error on each graph, we want to define a label assignment $Y^\star$ that minimizes the labelling errors due to $L_p^{(+)}$ and maximizes the labelling error due to $L^{(-)}$. Mathematically, we have

$$\begin{aligned} Y^\star \quad := \quad & \underset{Y \in \mathcal{P}}{\operatorname{argmin}} \sum_{p=0}^{K} \alpha_p \mathbf{Tr}(Y^\top L_p^{(+)} Y) - \mathbf{Tr}(Y^\top L^{(-)} Y) \\ = \quad & \underset{Y \in \mathcal{P}}{\operatorname{argmin}} \, \mathbf{Tr}(Y^\top L_{\text{eff}}^{(+)} Y) - \mathbf{Tr}(Y^\top L^{(-)} Y) \end{aligned} \tag{3}$$

where $L_{\text{eff}}^{(+)} = \sum_{p=0}^{K} \alpha_p L_p^{(+)}$, and $\alpha_p \geq 0$ weighs the contribution of labelling error due to p-th graph. Since $\alpha_p \geq 0$ and $L_p^{(+)}$ is positive semi-definite, $L_{\text{eff}}^{(+)}$ is also positive semi-definite. Given $Y^\star$, the i-th node is assigned the label that corresponds to the largest entry in $\boldsymbol{y}_i^\star$.

2.3. Optimization

In this section, we describe an algorithm to compute the optimal solution $Y^\star$. Let us rewrite Equation 3 as

$$\begin{aligned} Y^\star &= \operatorname*{argmin}_{Y\in\mathcal{P}} \mathbf{Tr}(Y^\top L_{\text{eff}}^{(+)} Y) - \mathbf{Tr}(Y^\top L^{(-)} Y) \\ &:= \operatorname*{argmin}_{Y\in\mathcal{P}} f(Y) - g(Y) \end{aligned} \tag{4}$$

As $L_{\text{eff}}^{(+)}$ and $L^{(-)}$ are positive semi-definite matrices, both $f(Y) := \mathbf{Tr}(Y^\top L_{\text{eff}}^{(+)} Y)$ and $g(Y) := \mathbf{Tr}(Y^\top L^{(-)} Y)$ are convex in Y, whereas $f(Y) - g(Y)$ is non-convex. However, it belongs to a family of problems, called difference of convex (DC) programming and efficient algorithms have been developed to solve such problems [24]. An iterative algorithm to solve the problem in Equation 4 is presented in Algorithm 1. Starting with a random label distribution $Y^{(0)} \in \mathcal{P}$, the algorithm iteratively linearizes $g(Y)$ around the k-th iterate $Y^{(k)}$ and solves the resulting convex function $f(Y) - \nabla g^\top \left(Y^{(k)}\right) Y$ until convergence. The convergence tolerance, ϵ, is set to 10^{-4} in our experiments. As shown in Appendix 1, solving a sequence of such convex programs solves the original problem.

Algorithm 1 Iterative algorithm to solve Equation 3

```
Input: Graph Laplacians $\{L_p^{(+)}, p = 0, ..., K\}$, $L^{(-)}$, a
set of weights $\{\alpha_p, p = 0, ..., K\}$, tolerance $\epsilon$
Output: Optimal solution $Y^\star$.
Procedure:
Choose the initial solution $Y^{(0)}$ as a random point in $\mathcal{P}$.
Set $k = 0$.
repeat
    1. Compute $\nabla g(Y^{(k)})$, gradient of $g(Y)$ at $Y^{(k)}$.
    2. Solve the convex optimization problem
       $Y^{(k+1)} = \text{argmin}_{Y\in\mathcal{P}} \left[f(Y) - \nabla g^\top (Y^{(k)}) Y\right]$
       by projected gradient method [13].
    3. $k = k + 1$.
until $\| Y^{(k+1)} - Y^{(k)} \|_F < \epsilon$.
Return $Y^\star = Y^{(k)}$.
```

3. Related work

In this section, we provide a brief review of the recent and related works under the following categories:

Label propagation in graphs. Propagation of labels in a graph has been extensively studied in machine learning as a semi-supervised learning approach, and a concise survey of recent developments can be found in [18]. In short, most of these approaches assume that the label of a node is approximated as the linear combination of the labels of its neighbours [26]. In [25], the authors use a mixed label propagation in which (i) they measure the bipolar similarity (*e.g.*, Karl Pearson's correlation coefficient that lies in the range [-1,1]) between the samples, and (ii) construct a 'positive' and a 'negative' graph based on the sign of the coefficient. Afterwards, they minimize the ratio between labelling errors due to the positive and negative graphs. This is done by semi-definite relaxation in order to assign a binary label to each node of the graph. Our method differs from [25] both in the definition of the graph similarities, and the label propagation method. Specifically, since we use multi-class labels instead of binary labels, and impose that the label distribution at each node should lie on a probability simplex, our problem is difficult to cast into their formalism. Therefore, we adopt difference of convex programming approach to solve our problem.

Message passing. Message passing approaches have been used to label the nodes in a graph [19, 6]. In [19], a subset of the nodes are initially labelled and then a conditional random field is used to infer the label of the remaining nodes. For this, the authors compute various appearance features and assume that the features are always available with similar accuracies. Hence, their approach cannot exploit appearance features that are sporadic or affected by non-stationary noise. In [6], the authors utilized such non-stationary and sporadic features in order to prioritize the propagation of belief, related to the label probability distribution. Even though this approach allows to exploit sporadically available appearance features, it relies on the assumption that the target appearances are known beforehand, which is not the case of our approach.

Mutual exclusion. The exploitation of a specific constraint associated to the structure of the graph (*e.g.*, the exclusivity constraint associated to the detections that coexist in time) has been considered in [16, 17] in order to learn discriminative appearance features. In these papers, first of all, a low-level but reliable tracker is used to connect unambiguous detections into tracklets. Afterwards, positive samples are defined by pairs of detections that belong to the same tracklet, while negative samples correspond to pairs that belong to tracklets that likely correspond to distinct objects (because they overlap in time). Lastly, these samples are used to train an AdaBoost [15], which in turn selects the discriminative appearances. Our approach could benefit from the above approach in order to select the discriminative features, while defining the appearance graph(s).

In [7, 8], the authors define a mutual exclusion term based on the physical distance between two detections that occur at the same time. The term goes to infinity as the distance goes to zero. This is motivated by the fact that two objects cannot occupy the same space simultaneously. Our formulation is different in that our mutual exclusion term is defined in terms of the similarity in the label distribution rather than the position.

4. Evaluation

The proposed algorithm has been evaluated on the following well-known and challenging datasets: APIDIS [1], PETS-2009 S2/L1 [2] and TUD Stadtmitte [3]. APIDIS is a multi-camera sequence acquired during a basketball match, whereas the other two are monocular sequences.

In the remainder of the section, we first describe these datasets. We then discuss the evaluation metrics and the implementation details. Finally, we present our results and compare with several state-of-the-art methods.

4.1. Datasets

APIDIS dataset. This 1 minute dataset is generated by 7 cameras, distributed around a basketball court. The candidate detections are computed independently at each time instant based on a ground occupancy map, as described in [14]. For each detection, the jersey color and its digit are computed to define the appearance features. In short, the jersey color is computed as the average blue component divided by the sum of average red and green components, over the foreground silhouette of the player within the detected rectangular box. The digit feature is obtained by running a digit-recognition algorithm in the same rectangular region. The digit feature is inherently sporadic as it is available only when the digit on the jersey faces the camera. The ground-truth is obtained from [1].

Pedestrian datasets. In order to evaluate the performance of our method in monocular views, we use publicly available PETS-2009 S2/L1 and TUD Stadtmitte datasets. The PETS dataset is 795-frames long, with moderate target density. However, the pedestrians wear similar dark clothes, which makes appearance comparison very challenging. TUD Stadtmitte is 179 frames long but the targets frequently occlude each other because of the low viewpoint. Detection results and the ground-truth are obtained from [4]. Afterwards, 8-bin CIE-LAB color histograms are computed for each channel of each bounding box, resulting in a 24-bin appearance vector. We ignore the histogram(s) if the overlap ratio between any two bounding boxes exceeds 5%. This makes the features sporadic over time.

4.2. Evaluation metrics

We adopt the widely used CLEAR MOT metrics[11] to evaluate our approach. The Multi-Object Tracking Accuracy (MOTA) combines missed targets (MS), false positives (FP) and identity switches (SW) into one number that varies from 0 to 100%. A tracker output and the ground-truth are defined to be matched if their intersection-over-union ratio exceeds 50% (respectively, if the distance < 30 cm for APIDIS). The Multi-Object Tracking Precision (MOTP) averages the bounding box overlap (respectively, distance between the ground truth and the tracker output for APIDIS) over all tracked targets, as a measure of localization accuracy.

4.3. Implementation details

The algorithm has been implemented on MATLAB running on a 2.4 GHz dual core CPU with 4 GB RAM.

Pedestrian datasets. For these datasets, a node is assigned to each individual detection. The size of the temporal neighborhood in spatio-temporal graph is chosen to be 10 frames. Thus, $T = 10$. In the current implementation, the graph construction step takes around 3 minutes for the PETS dataset and 2 minutes for the TUD Stadtmitte and the label propagation step is still the bottleneck. It takes around 1/2 hour for the TUD Stadmitte dataset and 1 hour for PETS for the label propagation step. When processing time is an issue, we can envision processing the dataset in batches or running a low-level but reliable tracker first to reduce the complexity (which we perform in the APIDIS dataset).

APIDIS dataset. We first pre-process the data by aggregating some of the detections into tracklets based on a spatio-temporally local but reliable tracker. The advantages are twofold. On the one hand, it reduces the number of nodes in the graph, thereby reducing the complexity of the algorithm. On the other hand, it helps to aggregate the appearance feature(s) along the tracklet in order to infer the appearance more accurately. The local but reliable tracker associates unambiguous detections between successive frames. Two detections are supposed to be unambiguous if the distance between them is less than 15 cm, and if there are no other detections within that distance. The resulting tracklets define the nodes in our graphs. The neighborhood of the spatio-temporal graph is defined to extend the tracklet size by 100 frames on each side, which allows us to connect tracklets that are up to 4 seconds apart. In the exclusion graph, the neighborhood of a node consists of all the nodes that overlap in time. Finally, the appearance features of a tracklet is inferred by averaging the appearance features of the detections along the tracklet. The low-level tracker takes 15 seconds, graph construction takes 1 minute and label optimization step takes 5 minutes. With an optimized implementation, it is possible to reach the real-time performance.

4.4. Results

To better compare with the literature, we consider two versions of our proposed method. The first one does not take appearance into consideration. It uses only the spatio-temporal information in order to label the nodes in the graph. Thus, we construct only the spatio-temporal and the exclusion graphs. This is equivalent to setting $\alpha_0 = 1$

and $\alpha_p = 0, \forall p \neq 0$ in our algorithm. In contrast, the second one considers both the spatio-temporal and the appearance features. In this case, we use $\alpha_0 > \alpha_1$ (α_0 for the spatio-temporal graph and α_1 for the appearance graph) for the TUD and PETS datasets. This constrains the spatio-temporal consistency more strictly than the appearance consistency. The reason is that the targets wear similar clothes and therefore have similar appearances in the datasets. In the experiments, we use $\alpha_0 = 1$ and $\alpha_1 = 0.5$.

The tracking results for the TUD Stadtmitte dataset are presented in Table 1. We can see that our method is much better than [7, 8] and [27] in terms of both MOTP and MOTA. This is because our approach is able to connect the detections even if they are far in time, resulting in longer and consistent tracks. However, our method is slightly worse than [27] in terms of ID switches. This might be because [27] uses motion information in a global manner in order to ensure a smooth motion while connecting the tracklets, which is not the case in our formalism.

Method	MOTA	MOTP	SW
Continuous energy [7]	60.5	65.8	7
Discrete-continuous [8]	61.8	63.2	4
GMCP tracker [27]	77.7	63.4	0
Our method (without appearance)	62.6	73.5	17
Our method (with appearance)	79.3	73.9	4

Table 1. Performance on the TUD Statdmitte dataset. The results are extracted from [8] and [27].

The results on the PETS-2009 S2/L1 dataset are presented in Table 2. The k-shortest paths [10], the continuous optimization [7] and the discrete-continuous optimization [8] approaches do not use the appearance features. Therefore, we compare them to the first version of our approach, which does not use appearance information. Similarly, the global appearance approach [23], the iterative hypothesis testing [5] and the GMCP tracker [27] use appearance features. We compare them to the second version of our approach, which uses appearance features. Specifically, since the global appearance approach [23] and the iterative hypothesis testing [5] also consider the fact that the color histograms are sporadic, the comparison with them is more relevant. From Table 2, we can see clearly that our proposed approach outperforms several contemporary approaches. When the appearance features are ignored, the MOTA metric is better than [10] but worse than [8]. This might be because of the fact that [8] exploits higher-order motion models, whereas our formalism does not. We assert the fact that a linear motion is implicit in our formalism in order to justify our superior performance against [10] and [23], which do not take the motion information into account. When the appearance information is incorporated, the performance is improved significantly from 82% to 91%. Moreover, the switching error is drastically reduced.

Method	MOTA	MOTP	SW
Discrete-continuous [8]	89.30	56.40	-
Continuous energy [7]	81.84	73.93	15
K-shortest paths [10]	80.00	58.00	28
GMCP tracker [27]	90.30	69.02	8
Global appearance [23]	81.46	58.38	19
Iterative hypothesis [5]	83.0	74.0	-
Our method (without appearance)	82.75	71.21	25
Our method (with appearance)	91.01	70.99	5

Table 2. Tracking results on the PETS 2009-S2/L1 dataset. The results are obtained from [8, 5, 27].

The result for APIDIS dataset is presented in Table 3. As before, first we computed the results without using any appearances. This is done by setting $\alpha_0 = 1, \alpha_1 = 0, \alpha_2 = 0$, where the indices 0, 1 and 2 correspond to the spatio-temporal,the color and the digit graphs respectively. Afterwards, we use both the digit and the color features. As the color feature is less discriminant (because the players from the same team wear jersey of the same color) than the digit feature, we set $\alpha_1 < \alpha_2$. Empirically, we use $\alpha_0 = 1, \alpha_1 = 0.1, \alpha_2 = 0.5$. Even though our approach performs significantly better than [23], the results are slightly worse than [5]. The reason might be because of the fact that iterative hypothesis testing framework associates two nodes only when the connection is sufficiently reliable than alternative connections. This prevents potential track switches. This is well-reflected by the switching errors.

Method	MOTA	MOTP	SW
Iter. hypothesis (no app.) [5]	85.83	60.83	18
Iter. hypothesis (color+digit) [5]	86.19	60.90	12
Global app. (no app.) [23]*	72.91	53.13	108
Global app. (color+digit) [23]*	73.07	53.15	110
Our method (no app.)	81.25	57.13	49
Our method (color+digit)	83.80	60.01	45

Table 3. Results on the APIDIS dataset. The tracking results have been provided by the authors of [5, 23]. [*] Since the detection results for [23] are different than that for the [5] and ours, we relax the distance threshold to 40 cm (from 30 cm) for the tracking results of [23].

Some sample frames of our tracking results are presented in Figure 2. In case of the APIDIS dataset, the frames from camera 1 and 6 are stitched in order to provide an entire view of the field. Two typical failure cases in our tracking system are depicted below. In Figure 3, an identity switch is depicted. The identities of two targets are momentarily switched. This might be because of the fact that we do not consider the appearance feature if the overlap between their

We varied $\alpha_1 \in [0.1, 1]$ but did not observe significant performance changes.

We performed a grid search on for various values of α_1 and α_2.

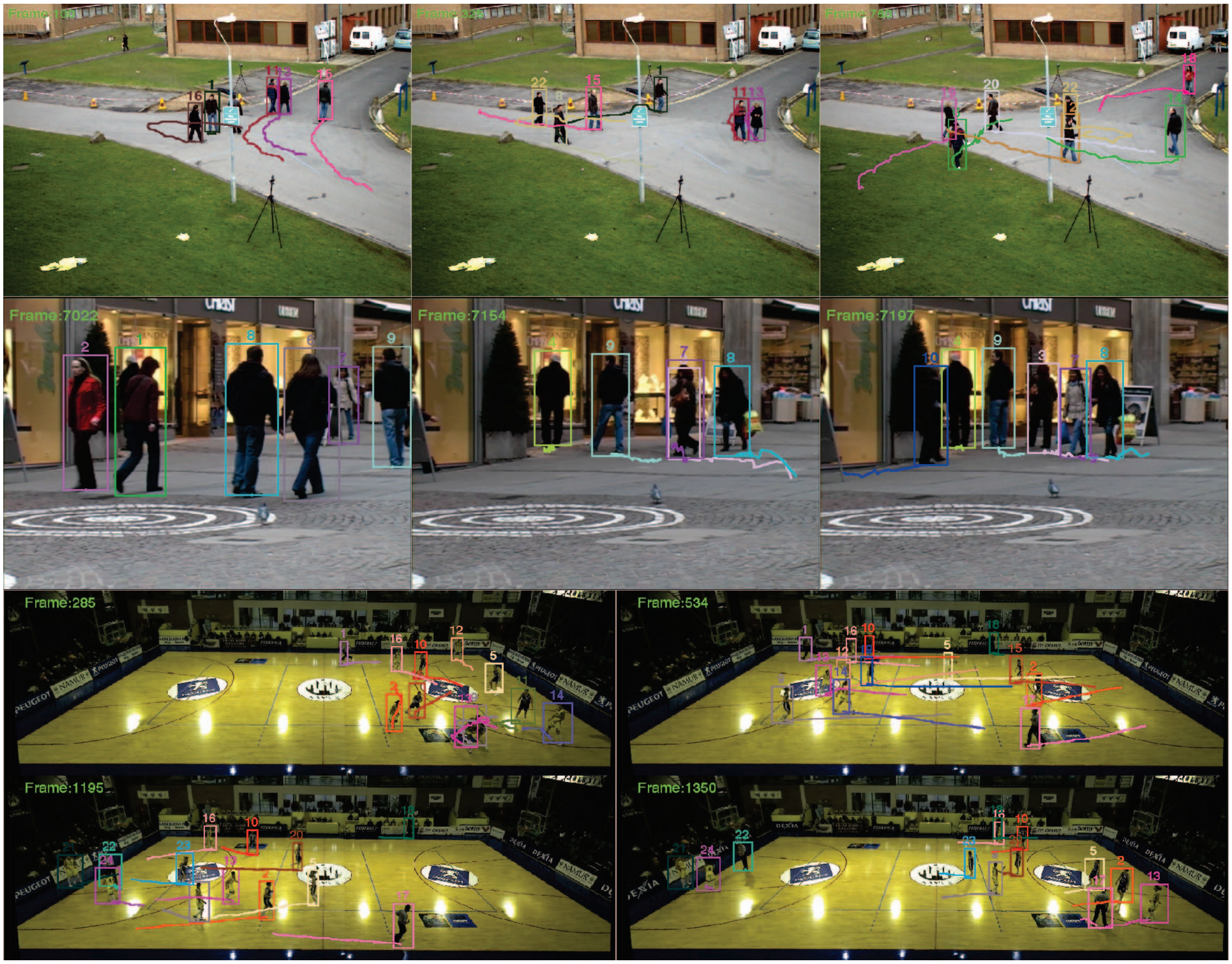


Figure 2. **Sample frames** from the PETS2009-S2/L1 (first row), the TUD Statdmitte (second row) and the APIDIS (third and fourth rows) datasets. For the sake of clarity, a tail of 50 frames is added. The numbers represent the distinct IDs of the tracks.

bounding boxes exceeds 5%. Therefore, neither the position nor the appearance disambiguates the identities of the targets. Later on, when the targets are separated, the algorithm is able to assign the correct label to the targets.

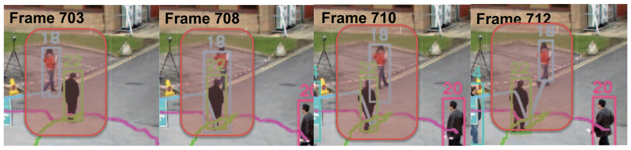


Figure 3. **Instantaneous identity switch.** In frame 703, targets 18 and 22 come close. Their bounding boxes overlap significantly in the frame 708 and their identities are momentarily switched. Afterwards, the targets separate and their identities are retained in frames 710 and 712.

Figure 4 shows an example of false positive. This happens because of the spurious detections provided by the object detector. Our current approach does not model such spurious detections, which are typically characterized by their low confidence values, explicitly.

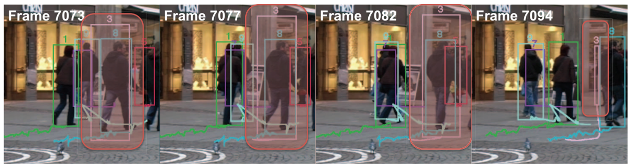


Figure 4. **False positive.** A false target 3 appears in frame 7073 and lasts until frame 7094.

5. Conclusion

In this paper, we focus on the problem of multi-object tracking under sporadic appearance features. For this, a number of distinct graphs are constructed in order to capture the spatio-temporal (including the exclusivity constraint), and the appearance information. Afterwards, we formulate the multi-object tracking as a consistent labelling problem in the associated graphs, and provide an efficient solution based on DC (Difference of Convex) functions programming. The effectiveness of the proposed approach has been demonstrated with several challenging datasets. One limitation of the approach is the scalability of the method with respect to the number of nodes. To some extent, it has been taken care by pre-processing the detections into tracklets. Nevertheless, algorithmic improvements are possible. This issue will be investigated in the future works.

References

[1] http://www.apidis.org/Dataset/. 5

[2] http://www.cvg.rdg.ac.uk/PETS2009/. 5

[3] http://www.d2.mpi-inf.mpg.de/node/428. 5

[4] http://www.gris.informatik.tu-darmstadt.de/~aandriye/data.html. 5

[5] Amit Kumar K.C., D. Delannay, L. Jacques, and C. D. Vleeschouwer. Iterative hypothesis testing for multi-object tracking with noisy/missing appearance features. In *Detection and Tracking in Challenging Environments Workshop in ACCV*, 2012. 2, 6

[6] Amit Kumar K.C. and C. D. Vleeschouwer. Prioritizing the propagation of identity beliefs for multi-object tracking. In *British Machine Vision Conference (BMVC)*, 2012. 4

[7] A. Andriyenko and K. Schindler. Multi-target tracking by continuous energy minimization. In *CVPR*, 2011. 4, 6

[8] A. Andriyenko, K. Schindler, and S. Roth. Discrete-continuous optimization for multi-target tracking. In *CVPR*, 2012. 4, 6

[9] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, January 2009.

[10] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *PAMI*, 2011. 1, 6

[11] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *Journal on Image and Video Processing*, 2008(1), February 2008. 5

[12] W. Brendel, M. Amer, and S. Todorovic. Multi-object tracking as maximum weight independent set. In *CVPR*, 2011. 1

[13] P. H. Calamai and J. J. Moré. Projected gradient methods for linearly constrained problems. *Mathematical programming*, 39:93–116, 1987. 4, 8

[14] D. Delannay, N. Danhier, and C. D. Vleeschouwer. Detection and recognition of sports(wo)men from multiple views. In *ICDSC, Como, Italy*, 2009. 5

[15] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55, 1997. 4

[16] C.-H. Kuo, C. Huang, and R. Nevatia. Multi-target tracking by online learned discriminative appearance models. In *CVPR*, 2010. 4

[17] C.-H. Kuo and R. Nevatia. How does person identity recognition help multi-person tracking. In *CVPR*, 2011. 4

[18] W. Lu, J. Wang, and S.-F. Chang. Robust and scalable graph-based semisupervised learning. *Proceedings of the IEEE*, 100(9), September 2012. 4

[19] W.-L. Lu, J.-A. Ting, J. J. Little, and K. P. Murphy. Learning to track and identify players from broadcast sports videos. *PAMI*, 2012. 4

[20] Y. Nesterov. *Introductory Lectures on Convex Optimization. A Basic Course*. 2004.

[21] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, 2011. 1

[22] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, December 2000. 2

[23] H. B. Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Tracking multiple people under global appearance constraints. In *ICCV*, 2011. 1, 6

[24] B. K. Sriperumbudur and G. R. G. Lankriet. On the convergence of the concave-convex procedure. In *NIPS*, 2009. 4, 8

[25] W. Tong and R. Jin. Semi-supervised learning by mixed label propagation. In *Proceedings of the 22nd national conference on Artificial intelligence - Volume 1*, AAAI'07, pages 651–656. AAAI Press, 2007. 4

[26] F. Wang and C. Zhang. Label propagation through linear neighborhoods. In *ICML06, 23rd International Conference on Machine Learning*, 2006. 4

[27] A. R. Zamir, A. Dehghan, and M. Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *ECCV*, 2012. 6

[28] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 2003. 3

A. Appendix 1

Our problem can be written as

$$\begin{aligned}\phi(Y) &= f(Y) - g(Y) \\ &\leq f(Y) - \left[g(Y^{(k)}) + \nabla g^{\top}(Y^{(k)})(Y - Y^{(k)})\right] \\ &= f(Y) - \nabla g^{\top}(Y^{(k)})Y - g(Y^{(k)}) \\ &\quad + \nabla g^{\top}(Y^{(k)})Y^{(k)} \\ &:= \hat{\phi}(Y, Y^{(k)}) \qquad (5)\end{aligned}$$

where the inequality in Equation 5 is due to the convexity of $g(Y)$. We can see that $\phi(Y) \leq \hat{\phi}(Y, Y^{(k)}), \forall Y \in \mathcal{P}$ with the equality holding only when $Y = Y^{(k)}$. In the literature, $\hat{\phi}(Y, Y^{(k)})$ is called the *majorization* of $\phi(Y)$ [24]. The solution

$$Y^{(k+1)} = \operatorname*{argmin}_{Y \in \mathcal{P}} \hat{\phi}(Y, Y^{(k)}) \qquad (6)$$

follows the inequality

$$\phi(Y^{(k+1)}) \leq \hat{\phi}(Y^{(k+1)}, Y^{(k)}) \leq \hat{\phi}(Y^{(k)}, Y^{(k)}) = \phi(Y^{(k)}),$$

where the first inequality and the last equality follow from Equation 5 and the second inequality follows from Equation 6. Therefore, above iterate in Equation 6 monotonically decreases $\phi(Y)$. In order to solve the convex problem in Equation 6, we use the projected gradient method [13] as

$$Y^{(l+1)} = \textbf{Proj}_{\mathcal{P}}\left(Y^{(l)} - \beta_l \nabla \hat{\phi}(Y^{(l)}, Y^{(k)})\right), \qquad (7)$$

where **Proj** is the projection onto the probability simplex $\mathcal{P}$, and β_l is the step size. We use a fixed step size $\beta_l = 1/\lambda_{\max}$ where $\lambda_{\max}$ is the largest eigenvalue of $L_{\text{eff}}^{(+)}$.