# Multi-Commodity Network Flow for Tracking Multiple People

Horesh Ben Shitrit, Jérôme Berclaz, François Fleuret, and Pascal Fua, *Fellow, IEEE*

**Abstract**—In this paper, we show that tracking multiple people whose paths may intersect can be formulated as a multi-commodity network flow problem. Our proposed framework is designed to exploit image appearance cues to prevent identity switches. Our method is effective even when such cues are only available at distant time intervals. This is unlike many current approaches that depend on appearance being exploitable from frame to frame. Furthermore, our algorithm lends itself to a real-time implementation.
We validate our approach on three publicly available datasets that contain long and complex sequences, the APIDIS basketball dataset, the ISSIA soccer dataset and the PETS'09 pedestrian dataset. We also demonstrate its performance on a newer basketball dataset that features complete world championship basketball matches. In all cases, our approach preserves identity better than state-of-the-art tracking algorithms.

**Index Terms**—Multi-object tracking, Multi-Commodity Network Flow, MCNF, Tracklet association, Linear Programming

## 1 Introduction

In this paper, we address the problem of tracking multiple people whose paths may intersect repeatedly over long periods of time while retaining their individual identities. We assume that a time-independent people detector is available and provides probabilities of presence at various possible spatial locations. The problem is therefore reduced to linking these detections into consistent trajectories, which is a sub-domain of tracking known as tracking-by-detection [1].

A standard approach to doing this is to recursively track from frame to frame, which may easily lead to irrecoverable errors if a person is undetected in a frame or if two detections made at different times are inappropriately linked. To overcome this weakness, the recursive tracking approach can be replaced by global trajectory optimization over batches of frames. Dynamic Programming [2], [3] or Linear Programming [4], [5] have been successfully applied to solve such global optimization frameworks. While both methods operate on directed graphs whose nodes represent places where people have been detected, the latter tends to be more robust than the former but scales poorly for large problems and long batches. This limitation can be alleviated by linking detections over a few frames into trajectory fragments, or *tracklets*, which become the graph nodes to be linked [6], [7], [8]. This reduces the computational complexity and increases robustness but still relies on heuristics such as introducing occlusion nodes and the proper setting of many parameters, such as those controlling arc-costs in the graph. In a recent paper [9], we showed that this problem could be addressed by formulating the multi-object tracking as minimum-cost maximum-flow problem. This is a global optimization problem whose objective function is convex and depends on very few parameters. Furthermore, it can be efficiently solved using the K-Shortest Paths algorithm (KSP) [10].

However, in our earlier approach [9], we completely ignored appearance, which can result in unwarranted identity switches in complex scenes. In this paper, we therefore extend it to allow the exploitation of *sparse* appearance information to keep track of people's identity, even when their paths come close to each other or intersect. By sparse we mean that the appearance needs only be discriminative in a very limited number of frames. For example, in the basketball sequence of Fig. 1, all teammates wear the same uniform and the numbers on the back of their shirts can only be read once in a long while. Furthermore, the appearance models are most needed when the players are bunched together. However, it is precisely then, that they are the least reliable [11]. Our algorithm, which we first introduced in a conference paper [12], can disambiguate such situations using the information from temporally distant frames. This is in contrast with many state-of-the-art approaches that depend on associating appearance models across *successive* frames [5], [3], [13], [14].

We formulate multi-object tracking with appearance constraints as a multi-commodity minimum-cost maximum-flow problem. This involves working with a layered graph such as the one of Fig. 2(b), which contains several grid cells at each possible spatial location, one for each possible identity group. It is much larger than the one of our original approach [9] that contains a single layer and is depicted by Fig. 2(a). However,

- *H. Ben Shitrit and P. Fua are with the École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland. E-mail: {horesh.benshitrit, pascal.fua}@epfl.ch*
- *J. Berclaz is now with Microsoft, Sunnyvale, CA, 94089, USA. E-mail: jerome.berclaz@microsoft.com; This work was done while he was at EPFL.*
- *F. Fleuret is with the Idiap Research Institute, CH-1920 Martigny, Switzerland, and the École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland. E-mail: francois.fleuret@idiap.ch*

*This work was supported in part by the CTI project "Image Based Object Tracking and Identification in Team Sports Environments" and by a Swiss National Science Foundation project. François Fleuret was supported in part by the European Community's Seventh Framework Programme FP7 - Challenge 2 - Cognitive Systems, Interaction, Robotics - under grant agreement No 247022 - MASH.*

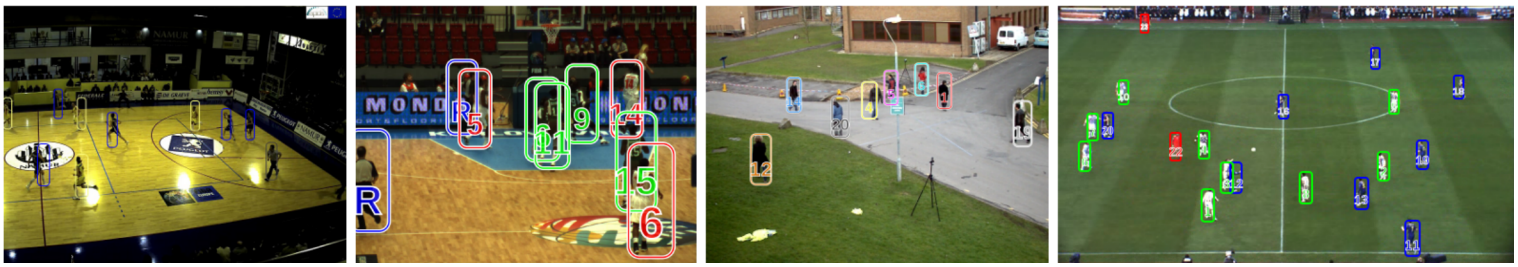


Fig. 1. Representative tracking results from the four tested datasets Basketball-APIDIS, Basketball-FIBA, Pedestrians-PETS'09 and Soccer-ISSIA

by first running the KSP method [9] on this smaller graph, we can eliminate all nodes that are consistently empty and run our algorithm on a much reduced layered graph, thus making the problem tractable. The computation can be further sped up by grouping unambiguously connected graph nodes into tracklets, which results in the further reduced graph of Fig. 5 and allows for real-time performance.

The contribution of this paper is therefore both a reformulation of the identity-preserving multiple target tracking problem as one of maximizing a convex objective function and a real-time algorithm for doing so. We validate our new method on multiple datasets featuring basketball players, soccer players, and pedestrians and demonstrate a significant improvement over earlier approaches.

## 2 Related Work

In this section, we first review generic approaches to multiple target tracking and then briefly discuss those that are specifically designed to track multiple players in team sports.

### 2.1 Multiple Target Tracking

Multiple target tracking has a long tradition, going back many years for applications such as radar tracking [15]. These early approaches to data association usually relied on gating and Kalman filtering, which have later made their way into our community [16], [17], [18], [19], [20]. Because of their recursive nature, when used to track people in crowded scenes, they are prone to identity switches that are difficult to recover from. Particle-based approaches such as [21], [22], [23], [24], [25], [26], [1], among many others, partially address this issue by simultaneously exploring multiple hypotheses. However, they can handle only relatively small batches of temporal frames without their state space becoming unmanageably large and often require careful parameters setting to converge.

Another approach is to perform many simple operations on trajectory fragments or tracklets, such as adding new detections, splitting, or merging. Those that minimize a global energy function are retained, as in Multi Hypothesis Tracking [27], Markov Chain Monte Carlo Data Association [28], [29], [30], [31], or Discrete-Continuous Optimization [32] . These algorithms may take advantage of large frame batches but are quite difficult to implement and involve many parameters, which need to be learned and tuned.

Another recent approach in multi object tracking is to first construct reliable short tracklets, and then concatenate them into long trajectories. The tracklet association problem can be formulated as one of optimally connecting a bipartite graph [33], [34], [35], [36], [37] using the Hungarian algorithm. The main differences between these algorithms reside in the way the tracklets are constructed and the similarity measure between them. They include appearance, motion, and even social and activity terms. These methods, however, tend to depend on appearance information being available for all tracklets, which ours does not.

Alternatively, a "belief matrix" that represents tracklet association probabilities can be maintained [38]. Unlike the algorithms described above, this one decouples tracking from identification and its online implementation only propagates appearance-based probabilities forward as the information becomes available. This work was extended to include information of the relationship between tracks [39], [40]. However, the authors only used simulated data and manually provided relationships between people. It is not clear how these could be estimated automatically.

One important feature of our approach is that we solve the tracklet association problem by minimizing a global objective function. and other recent approaches also do this. For example, in [41], tracks corresponding to pairs of people are split and merged. Although effective for tracking pedestrians and preventing collisions, this does not apply to tracking multiple people who tightly interact. In [42], [43], Conditional Random Field (CRF) are used to model occlusions and motion dependencies between tracklets and in [44] Belief Propagation is used of this purpose. However, in situations where people move unpredictably as in team sports, it is not clear how to model the motion and the occlusions terms and further research would be needed to use this approach in such scenarios.

Dynamic and Linear Programming approaches have emerged as powerful alternatives. They operate on graphs whose nodes can either be all the spatial locations of potential people presence [2], [3], only those where a detector has fired [4], [5], or short temporal sequences of consecutive detections that are very likely to correspond to the same person [6], [7], [8]. On average, they are much more robust than the earlier methods but typically require the careful setting of edge costs in the graph, the introduction of special purpose nodes to handle occlusions, and an assumption that the appearance of people remains both unchanged and discriminative from frame to frame. This last assumption is damaging in cases where the lighting changes quickly or where the appearance is only

distinctive at long intervals, such as when tracking ballplayers who all wear the same uniform and whose number can only be read occasionally.

In our recent work [9], we overcame this limitation by directly working on the graph of all potential locations over time and solving the data association problem using the K-Shortest Paths (KSP) algorithm [10]. Our algorithm completely ignores appearance, does not require any heuristics regarding occlusion nodes, and has a comparatively low computation complexity in the order of $O(k(m + n \log n))$, where $n$, $m$, and $k$ are the number of graph nodes, edges, and trajectories. And, yet, it has been shown to outperform many state-of-the-art methods on the PETS'09 data set [45]. Its main limitation is that, because it does not exploit appearance, it cannot prevent identity switches when people come close to each other. This is the problem we address in this paper.

Recent approaches attempted coupling the detection and tracking steps [46], [47], instead of dissociating them as we do. In [47], 2D detections from each camera are used as input and coupled for 3D localization and tracking purposes. In [46], the inputs are background-foreground binary masks and the pedestrian detection and tracking are performed directly in 3D. These methods are shown to improve performance slightly in terms of miss detections and false detections over KSP [9] in some cases. However, like KSP , they do not incorporate appearance information and, therefore, cannot recover from identity mismatches. We could thus have used one of those algorithms instead of KSP to perform the first step of our approach as described below.

### 2.2 Tracking Multiple Players

Reliable tracking of players engaged in team sports involves challenges that do not exist when tracking pedestrians because the players tend to constantly change their motion pattern and, sometimes, to tackle their opponent. This makes their trajectories much more erratic and less predictable.

In [48], a framework for tracking multiple basketball player similar to ours is presented. Player detection is achieved using multiple cameras, while occasional reading of the numbers printed on their shirts is used for identification purposes. The algorithm relies on simple tracking to propagate the players' identities over time. The authors themselves write that this part should be improved and only present detection results without tracking ones.

More recently [49], Conditional Random Fields (CRF) have been used to track multiple basketball players from the same team in sequences acquired with a single moving camera. However, CRF require modeling and learning an objective function, which is then optimized. Therefore, extensive training is required and it is not clear how much new training is required from one match to the next.

## 3 Algorithm

In this section, we assume that the ground plane is represented by a discrete grid and that, at each time step over a potentially long period of time, we are given as input a Probabilistic Occupancy Map [3] (POM) containing probabilities of presence of people in each grid cell, which can be generated by any people detector. While informative, the resulting probability maps may contain both miss-detections and false positives, especially when the scene becomes crowded.

To infer identity-preserving trajectories from these potentially noisy POMs, we first extend the formalism introduced in [9] to account for individual identities, which the original formulation did not do. In practice, we compute flows on Direct Acyclic Graphs (DAGs) such as those of Fig. 2. This results in our multi-target tracking problem being reformulated as an Integer Program, which can be relaxed into a Linear Program and solved using standard optimization packages. It has been shown that multi-commodity network flow (MCNF) problems on DAGs can be solved in a polynomial time [50]. In general, for multi-commodity network flow (MCNF) problems with more than two commodities—groups of people in our case—the solution of the Linear Program is not necessarily that of the corresponding Integer Program because its constraint matrix is not totally unimodular [51]. However, in practice it almost always is [52].

Solving a full graph with Linear Programming might be very slow for long sequences, as the graphs tend to be large. To address this, we implemented a two-step process: First, we run the K-Shortest Paths (KSP) algorithm [9], which can be seen as a single commodity network flow. This produces trajectories that may include identity switches but tell us which are the grid cells in which we can expect to find people at any given time. We then eliminate most of the other grid cells and run our Linear Program on a significantly smaller graph, which saves both time and memory. Pruning the graph results does not guarantee that the solution we find is optimal but allows the batch processing of substantial numbers of frames relatively quickly. Furthermore, in practice, we have not seen any significant degradation in tracking performance.

For a further speed up, necessary for real-time applications, we retain the two-step process but, in addition to pruning the graph, we group nodes that are unambiguously connected into tracklets. We treat these tracklets as the nodes of a reduced graph, which is now small enough for the Linear Program to be solved in real-time.

In the remainder of this section, we first formulate our data association problem as a multi-commodity network flow problem on a DAG. We then show that doing so requires solving an Integer Program, which we relax to a Linear Program. Finally, we discuss the above-mentioned approaches for reducing the computational cost of finding the optimal solution.

### 3.1 Formulation

As in [9], we model people's trajectories as continuous flows going through an area of interest. Preserving the identities of the tracked people means that the flows should not be allowed to mix. In addition, available soft biometrics or appearance cues should be used to assign flows to people. These two elements are missing from the original formulation. By formulating the tracking problem as a multi-commodity network flow problem we manage to include these elements in

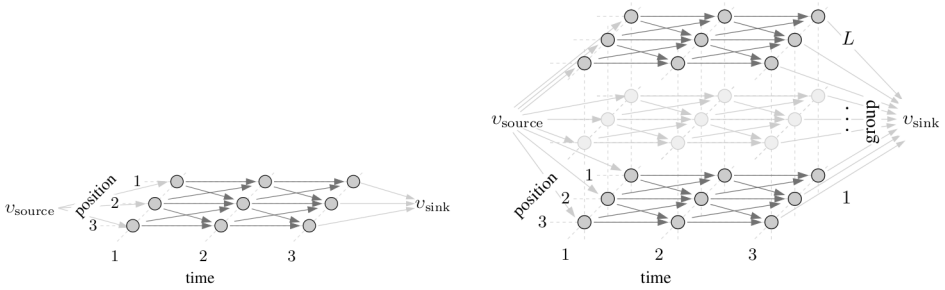



Fig. 2. Our tracking algorithm involves computing flows on a Directed Acyclic Graph (DAG). (a) In [9], the DAG includes source and sink nodes that allow people to enter and exit at selected locations, such as the boundaries of the playing field. This can be interpreted as a single-commodity network flow. (b) To take image-appearance into account, we formulate the tracking problem as a multi-commodity network flow problem which can be illustrated as a duplication of the graph for each appearance-group. The obtained problem has a much larger graph, which makes it intractable. In this work, we present two methods for reducing this complexity, and solving the problem in real-time.

our tracking algorithm. In our formulation, we use the notation summarized in Table 1.

TABLE 1
Notation

| | |
|---|---|
| $T$ | number of time steps. |
| $\mathbf{I} = (\mathbf{I}^1, \dots, \mathbf{I}^T)$ | captured images. |
| $K$ | number of locations on the ground plane. |
| $L$ | number of *labeled* groups of people |
| $N_l$ | maximum number of people in group $l$. |
| $\mathcal{N}_\delta(k) \subset \{1, \dots, K\}$ | neighborhood of location $k$, all locations which can be reached within $\delta$ time instants. |
| $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ | full graph with $\lvert\mathcal{V}\rvert = K \times T \times L$. |
| $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ | a pruned graph, subset of $\mathcal{G}$. |
| $v_i^l(t)$ | node in the graph, represents group $l$ in location $i$ at time $t$. |
| $m_i^l(t)$ | number of people from group $l$ in location $i$ at time $t$. |
| $e_{i,j}^l(t)$ | directed edge in the graph. |
| $f_{i,j}^l(t)$ | number of people moving from location $i$ to location $j$ at time $t$ in group $l$. |
| $Q_i(t)$ | r.v. standing for the true identity group of a person in location $i$, at time $t$. |
| $X_i(t)$ | r.v. standing for the true occupancy of location $i$ at time $t$. |
| $\varphi_i^l(t)$ | estimated probability of a location $i$ to be occupied by a person from group $l$, according to the appearance model. |
| $\rho_i(t)$ | estimated probability of location $i$ to be occupied by an unidentified person according to the pedestrian detector. |
| $\delta$ | maximum distance between locations, used to define a neighborhood of a location or a trajectory. |
| $\mathfrak{F}$ | set of occupancy maps physically possible. |
| $\mathcal{T}$ | set of trajectories obtained by the KSP algorithm. |
| $\tau_q \subset \mathcal{T}$ | a trajectory obtained by the KSP algorithm. |
| $\tau_q^i \subset \tau_q$ | a tracklet, a fragment of trajectory $\tau_q$. |
| $\mathcal{G}^* = (\mathcal{V}^*, \mathcal{E}^*)$ | the graph of tracklets. |
| $v_i^{*l}$ | node in the graph of tracklets, represents a tracklet of group $l$ with index $i$. |
| $e_{i,j}^{*l}$ | directed edge in the graph of tracklets. |

To this end, we represent the ground plane as a discrete grid containing $K$ cells and compute POMs at $T$ consecutive time instants. We partition the total number of tracked people into $L$ groups and assign a separate appearance model to each group. In a constrained scene, such as a ball game, we can restrict each group $l$ to include at most $N_l$ people, but in general cases, $N_l$ is left unbounded. The groups can be made of individual people, in which case $N_l = 1$. They can also be composed of several people that share a common appearance, such as members of the same team or referees, in sports games.

We introduce a Directed Acyclic Graph (DAG) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with number of vertices $|\mathcal{V}| = K \times T \times L$ such as the one of Fig. 2(b), in which every node $v_i^l(t)$ represents a location $i$ at a given time instant $t$ and for a particular identity group $l$. Each edge represents admissible motion between two locations, in two consecutive time instants. Since groups cannot exchange their identity, there are no edges linking groups, and therefore no vertical edge in Fig. 2(b). The resulting graph is made of disconnected layers, one per each identity group. This is in contrast to the approach of [9], which relied on a single-layer graph such as the one of Fig. 2(a).

Let $\mathcal{N}_\delta(k) \subset \{1, \dots, K\}$ be the neighborhood of location $k$, that is, the set of all physical locations that can be reached from $k$ in $\delta$ time instants. In practice, for the graph construction we set $\delta = 1$, which is already longer than the distance a person can travel in $1/25^{\text{th}}$ of a second, the typical frame duration in our experiments.

There is an edge $e_{i,j}^l(t)$ from location $i$ to location $j$ if and only if $j \in \mathcal{N}(i)$. We associate to every node of the graph a variable $m_i^l(t)$ standing for the number of people of group $l$ present on location $i$ at time $t$. Similarly, a variable $f_{i,j}^l(t)$ corresponds to every edge $e_{i,j}^l(t)$, and encodes the number of people of group $l$ moving from node $i$ to $j$ at time $t$.

We now define a set of constraints to ensure that every flow through the graph is physically possible. First, we enforce flow continuity by making sure that the sum of flows arriving at one node at time $(t-1)$ is equal to the sum of flows leaving the same location at time $t$

$$\forall t, l, i \quad \underbrace{\sum_{k:i\in\mathcal{N}(k)} f_{k,i}^l(t-1)}_{\text{Arriving at } i} = m_i^l(t) = \underbrace{\sum_{j\in\mathcal{N}(i)} f_{i,j}^l(t)}_{\text{Leaving from } i}. \quad (1)$$

Second, our grid cells are sufficiently small for one not to be occupied by more than one person at a time, hence

$$\forall t, i, \quad \sum_{j \in \mathcal{N}(i)} \sum_{l=1}^{L} f_{i,j}^{l}(t) \leq 1 \ . \tag{2}$$

Third, the flows have to be positive and we have

$$\forall t, l, i, j \ f_{i,j}^{l}(t) \geq 0 \ . \tag{3}$$

In case we have a precise knowledge about the number of people we track, we can use an optional constraint to ensure that no more than the allowed number of people is present in each group

$$\forall t, l \sum_{i=1}^{K} m_i^l(t) \leq N_l \ . \tag{4}$$

Our model as described so far can only handle a fixed number of people. In practice, however, the number of people in the scene, may vary over time. We therefore introduce a source and a sink nodes, $\upsilon_{\text{source}}$ and $\upsilon_{\text{sink}}$. The source node is connected to every node from the first frame and the sink to every node from the last frame, as shown in Fig. 2(b). Additionally, both nodes are connected to all locations in the set $\mathcal{N}(k) \subset \{1, \ldots, K\}$ of locations susceptible to act as entry or exit points throughout the whole sequence. This last part is not illustrated in Fig. 2 to avoid overloading it. The source and sink nodes are also subject to a constraint that enforces all the flows starting in $\upsilon_{\text{source}}$ to end in $\upsilon_{\text{sink}}$

$$\forall l \quad \sum_{j \in \mathcal{N}(\upsilon_{\text{source}})} f_{\upsilon_{\text{source}},j}^{l} = \sum_{k:\upsilon_{\text{sink}} \in \mathcal{N}(k)} f_{k,\upsilon_{\text{sink}}}^{l} \ . \tag{5}$$

### 3.2 Linear Program

Let us now assume that we have access to a person detector that estimates the probability of presence of someone at every position $i$

$$\rho_i(t) = \hat{P}(X_i(t) = 1 \,|\, \mathbf{I}) \ , \tag{6}$$

where $X_i(t)$ is a random variable standing for the true occupancy of location $i$ at time $t$, and $\mathbf{I}$ represents the input images. Let us further assume that we can compute an appearance model and that we use it to estimate

$$\varphi_i^l(t) = \hat{P}(Q_i(t) = l \,|\mathbf{I}, X_i(t) = 1) \,, \tag{7}$$

the probability that the identity of a person occupying location $i$ at time $t$ is $l$, given that the location is indeed occupied. Here, $Q_i(t)$ is a random variable standing for the true identity group of a person in location $i$ at time $t$. Let there be $L$ identity groups, hence $Q_i(t) \in \{1, \ldots, L\}$. The appearance model can rely on various cues, such as color similarity or shirt numbers of sports players. In Section 4.3, we describe in details the ones we use for different datasets.

Since we are seeking a set of physically possible trajectories that best explain the observed image evidence, we look for

$$\hat{\mathbf{m}} = \arg\max_{\mathbf{m} \in \mathfrak{F}} P(\mathbf{X} = \mathbf{x}, \mathbf{Q} = \mathbf{q} \,|\, \mathbf{I}) \ , \tag{8}$$

where $\mathbf{m}$ is a set of occupancy maps and $\mathfrak{F}$ is the space of occupancy maps satisfying constraints from Eqs. 1 to 5.

As shown in the appendix supplied as supplementary material, this can be expressed as a function of $\rho_i(t)$ and $\varphi_i^l(t)$ as

$$\hat{\mathbf{m}} = \arg\max_{\mathbf{m} \in \mathfrak{F}} \sum_{t=1}^{T} \sum_{i=1}^{K} \sum_{l=1}^{L} m_i^l(t) \log \left( \frac{\rho_i(t) \varphi_i^l(t) L}{1 - \rho_i(t)} \right) . \tag{9}$$

Note that when no appearance information is available, we set $\forall l, \varphi_k^l(t) = \frac{1}{L}$ and the appearance term cancels the $L$ coefficient in the objective function. In this case, this simplifies to the original one of [9]. This property makes it possible to process sparse appearance information, such as shirt numbers that can be read only once in a while. The spatial extent of trajectories is mostly based on the occupancy information, while the sparse appearance places a trajectory in the correct identity group and avoids switches at intersections.

Maximizing Eq. 9 under the constraints of Eqs. 1 to 5 can be formulated as an Integer Program, which is optimized with respect to the flows $f_{i,j}^{l}(t)$

$$\begin{aligned}
\text{maximize} \quad & \sum_{t=1}^{T} \sum_{i=1}^{K} \sum_{l=1}^{L} \log \left( \frac{\rho_i(t) \varphi_i^l(t) L}{1 - \rho_i(t)} \right) \sum_{j \in \mathcal{N}(i)} f_{i,j}^{l}(t) \\
\text{subject to} \quad & \forall t, l, i, \ \sum_{j \in \mathcal{N}(i)} f_{i,j}^{l}(t) - \sum_{k:i \in \mathcal{N}(k)} f_{k,i}^{l}(t-1) \leq 0 \\
& \forall t, i, \ \sum_{j \in \mathcal{N}(i)} \sum_{l=1}^{L} f_{i,j}^{l}(t) \leq 1 \\
& \forall t, l, i, j, \ f_{i,j}^{l}(t) \geq 0 \\
& \forall t, l, \ \sum_{i=1}^{K} \sum_{j \in \mathcal{N}(i)} f_{i,j}^{l}(t) \leq N_l \\
& \forall l, \ \sum_{j \in \mathcal{N}(\upsilon_{\text{source}})} f_{\upsilon_{\text{source}},j}^{l} - \sum_{k:\upsilon_{\text{sink}} \in \mathcal{N}(k)} f_{k,\upsilon_{\text{sink}}}^{l} \leq 0 \,.
\end{aligned} \tag{10}$$

In practice, since Integer Programming is NP-complete, we relax the problem of Eq. 10 into a Linear Program (LP) of polynomial complexity by making the variables real numbers between zero and one.

As the constraints matrix of a multi-commodity network flow problem is not totally unimodular [51], the LP results are not guaranteed to be integral and real values that are far from either zero or one may occur. In practice this only happens very rarely, and typically when two or more targets are moving so close to each other that appearance information is unable to disambiguate their respective identities, as depicted by Fig. 3. These non-integer results can be interpreted as an uncertainty about identity assignment by our algorithm. This represents valuable information that could be dealt with accordingly if necessary. However, as this happens rarely, we simply round off non-integer results in our experiments.

### 3.3 Optimization

Even though turning the Integer Program of Eq. 10 into a Linear Program substantially reduces its computational complexity, the large number of variables and constraints involved still results in too large a problem to be directly handled by regular solvers for real-life cases such as those presented in

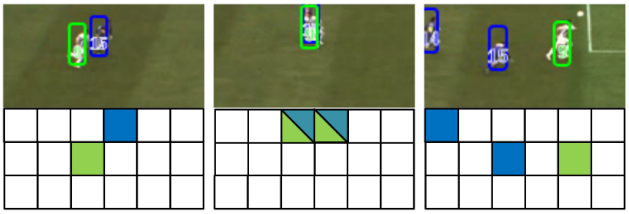


Fig. 3. The LP solver might produce non-integer values when two or more people intersect. In this example, our tracking algorithm assigns non-integer values for the identities of the two soccer players at two adjacent locations. After the intersection, the algorithm recovers and assigns again integer value to each identity.

Section 4. It is thus necessary to further simplify the problem. In the remainder of this section, we present two different ways to achieve this.

#### 3.3.1 Pruning the Graph

The simplest way to reduce the computational complexity is to remove unnecessary nodes from the graph of Fig. 2(b). To this end, we first run our earlier algorithm [9] on a graph with only one commodity ($L = 1$), such as the one in Fig. 2(a). The algorithm tracks all the people in the scene very efficiently but ignores appearance and is therefore prone to identity switches. We obtain a set of trajectories $\mathcal{T}$, each trajectory $\tau_q \in \mathcal{T}$ is a subset of vertices on this graph

$$\tau_q = \{v_i(t_q), v_j(t_q + 1), \ldots, v_h(T_q)\} \tag{11}$$

where the vertices belong to consecutive time instants starting at time $t_q$ and ending at time instant $T_q$. We take $\mathcal{N}_\delta(\tau_q) \subset \mathcal{V}$ to be the neighborhood of trajectory $\tau_q$, which is the set of all vertices that can be reached within $\delta$ time instances from a node $v_i(t) \in \tau_q$. Formally, we write

$$\mathcal{N}_\delta(\tau_q) = \bigcup_{v_i(t) \in \tau_q} \mathcal{N}_\delta(v_i(t)), \tag{12}$$

where $\delta$ can be larger than one, unlike for graph construction, as explained in section 3.1. We take the set of *shared* vertices $\mathcal{S}$ to be those who are included in the neighborhood of more than one trajectory. We write

$$\mathcal{S} = \{v_j(t) \in \mathcal{V}, \text{ s.t. } \sum_{\tau_q \in \mathcal{T}} \mathbb{1}(v_j(t) \in \mathcal{N}(\tau_q)) > 1\}. \tag{13}$$

The vertices $\mathcal{V}'$ in our pruned graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ are taken to be

$$\mathcal{V}' = \mathcal{T} \cup \mathcal{S}, \tag{14}$$

the union of the vertices in all trajectories and the shared vertices.

In other words, we eliminate most of the nodes in which nobody was found. However, we retain the shared vertices $\mathcal{S}$ because KSP produces trajectories with very good spatial accuracy, except where people meet and separate. There, it may erroneously link bits of trajectories belonging to different individuals and ignore the vertices through which the true trajectories pass. By adding the shared nodes, we give our algorithm the degrees of freedom it requires to avoid such mistakes by using appearance information. After pruning, we set $L$ to be the true number of identity groups, and keep only the set of constraints from equations Eq. 1 to Eq. 5, which involves vertices and edges of the pruned graph. In our experiments, the pruning reduces the number of variables and constraints by two to three orders of magnitude.

In the remainder of this paper, we will refer to this approach as Multi-Commodity Network Flow (MCNF).

#### 3.3.2 Grouping Nodes into Tracklets

A more radical approach to reducing the computational complexity is to not only remove obviously empty nodes from the graph but, in addition, to group obviously connected ones into *tracklets*. The Linear Program of Eq. 10 can then be solved on a reduced graph such as the one of Fig. 2(c) whose nodes are the tracklets instead of individual locations. In the remainder of this paper, we will refer to this approach as Tracklet-based Multi-Commodity Network Flow (T-MCNF).

Tracklets have been extensively used for people tracking [34], [36] and they are usually created on the basis of appearance being preserved over consecutive frames. In this work, we assume that appearance information may be unavailable over long periods of time and we therefore create our tracklets without reference to it.

To this end and as before, we start with the output of the KSP algorithm [9] and obtain a set of trajectories $\mathcal{T}$ and a set of shared vertices $\mathcal{S}$ as in section 3.3.1. In practice, these shared vertices are the only points where an identity switch could occur and we therefore take the trajectory fragments connecting them to be our tracklets, as depicted by Fig. 4.

Formally, each trajectory $\tau_q$ is split into tracklets $\tau_q^j$ $j = 1, ..., j_q$, which are connected fragments of the trajectory. We therefore have

$$\forall q \ \tau_q = \bigcup_j \tau_q^j. \tag{15}$$

We split a trajectory into tracklets by ensuring that, except for the end points, none of their vertices is in the neighborhood of a shared vertex. Let $T_q^j$ be the last time instant of tracklet $\tau_q^j$, then,

$$\forall q, i, j, t = T_q^j \quad \mathcal{N}(v_i(t) \in \tau_q^j) \cap \mathcal{S} \neq \emptyset. \tag{16}$$

We now construct a graph $\mathcal{G}^* = (\mathcal{V}^*, \mathcal{E}^*)$ whose nodes

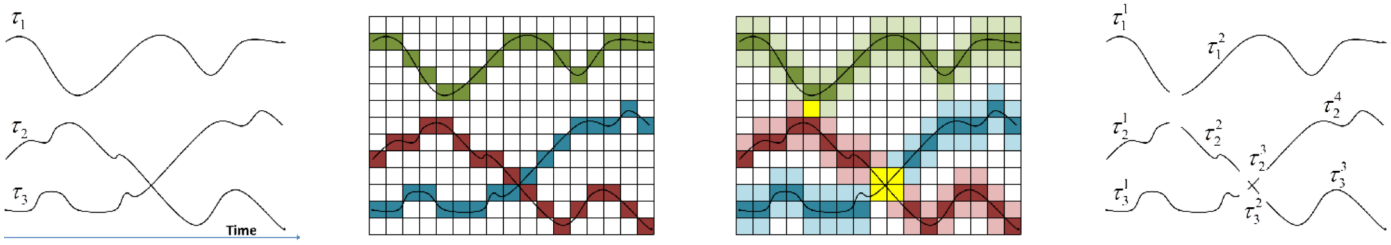
$$\mathcal{V}^* = \{\tau_q^j \,,\, q = 1, ..., |\mathcal{T}| \,, j = 1, ..., j_q\} \tag{17}$$

are the tracklets. In addition, we introduce an edge between each pair of tracklets such that the first vertex of one tracklet is included in the neighborhood of the last one of the other. We write

$$\begin{aligned} \mathcal{E}^* = \{(\tau_q^j \,, \tau_{q'}^{j'}) \ \exists\, v_i(T_q^j) \in \tau_q^j \,, v_{i'}(t_{q'}^{j'}) \in \tau_{q'}^{j'} \\ \text{s.t.} \ \ (t_{q'}^{j'} = T_q^j + 1) \ \wedge \ (v_{i'}(t_{q'}^{j'}) \in \mathcal{N}(v_i(T_q^j))\}. \end{aligned} \tag{18}$$

Here, $t_{q'}^{j'}$ is the starting time instant of tracklet $\tau_{q'}^{j'}$ and $T_q^j$ is the last time instant of tracklet $\tau_q^j$.

The resulting graph of tracklet $\mathcal{G}^*$ is a DAG. We formulate a set of linear constraints on this graph similar to those of Eq.




(a) (b) (c) (d)

Fig. 4. Splitting trajectories into tracklets. (a) For simplicity, we represent the trajectories as being one-dimensional and assume that we have three of them. (b) Each trajectory is a set of vertices from successive time instants. We assigned a different color to each. (c) The neighborhoods $\mathcal{N}$ of the trajectories with a distance $\delta = 1$ are shown in a color similar to that of the trajectory but less saturated. The shared vertices $\mathcal{S}$ are those that are included in more than one neighborhood appear in yellow. They are used as splitting points. (d) The resulting tracklets. Note that two trajectories do not necessarily have to cross to be split; it is enough that they come close to each other.

1 to 5 in section 3.1. An illustration of the resulting graph is shown in Fig. 5. The only difference is that in the MCNF version, each node represents one physical location at one time instant, whereas in the T-MCNF version, each node represents a tracklet, which is a set of successive locations in a batch of time frames.

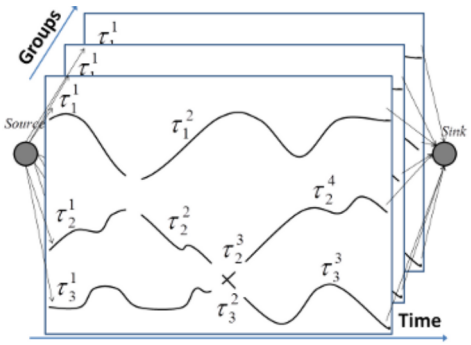



Fig. 5. The tracklet-based multi-commodity network flow algorithm can be interpreted as finding paths from the source node to the sink node, on a multiple layer graph whose nodes are the tracklets. This graph has the same properties as in Fig. 2 (b), however the number of vertices and edges is significantly smaller.

We therefore modify the meaning of variable $m_j^{*l}$ to stand for the number of people of group $l$ present in tracklet $j$. Similarly, the variable $f_{i,j}^{*l}$ now corresponds to the edge $e_{i,j}^{*l}$, and encodes the number of people of group $l$ moving from tracklet $i$ to tracklet $j$.

To maximize the flow on this graph, for each tracklet $\tau_q^{j,l}$, we define a score $S(\tau_q^{j,l})$ as the sum of the appearance scores of the vertices of the full graph $\mathcal{G}$ that form the tracklet. Formally, we write

$$\forall j,q,l \; S(\tau_q^{j,l}) = \sum_{v_k^l(t) \in \tau_q^{j,l}} \log(\varphi_k^l(t)L). \qquad (19)$$

Note that this score is similar to the sum of the scores of all the positions that compose the tracklet in the original objective function of the full layered grid (Eq. 10). However, in the tracklet case we do not take detection probabilities into account, as they have already been used to construct the tracklets.

Our Tracklet-based Multi-Commodity Network Flow (T-MCNF) problem then becomes

$$\begin{aligned}
&\text{maximize} \sum_{l=1}^{L} \sum_{i:j\in\mathcal{N}(i)} S(i) f_{i,j}^{*l} \\
&\text{subject to } \forall l,i, \sum_{j\in\mathcal{N}(i)} f_{i,j}^{*l} - \sum_{k:i\in\mathcal{N}(k)} f_{k,i}^{*l} \leq 0 \\
&\qquad \forall i, \sum_{j\in\mathcal{N}(i)} \sum_{l=1}^{L} f_{i,j}^{*l} \leq 1 \\
&\qquad \forall l,i,j \;\; f_{i,j}^{*l} \geq 0 \\
&\qquad \forall t,l \sum_{i:\text{ time } t\in i} \sum_{j\in\mathcal{N}(i)} f_{i,j}^{*l} \leq N_l \\
&\qquad \forall l, \sum_{j\in\mathcal{N}(v_{\text{source}})} f_{v_{\text{source}},j}^{*l} - \sum_{i:v_{\text{sink}}\in\mathcal{N}(i)} f_{i,v_{sink}}^{*l} \leq 0 \; .
\end{aligned} \qquad (20)$$

To obtain a feasible solution, we relax the IP problem of Eq. 20 to a LP problem, by allowing each flow to be assigned a value between 0 to 1 instead of a boolean value.

The T-MCNF graph is between one to two orders of magnitude smaller than the MCNF one. This results in a reduction of both computing time and memory consumption, while providing similar tracking results, as shown in Sec. 5.

## 4 Experiments

We use multi-camera sequences acquired during soccer and basketball matches to validate our approach and compare it against both the approach we extend [9], which completely ignores image appearance, and a modified version of it that takes frame-to-frame appearance into account, using simple features patterned after those used in [14].

To highlight the impact of the Linear Programming formulation, we implemented a Dynamic Programming approach to linking tracklets, inspired by those of [3], [8].

| Numbers<br>Name | Camera | Frame | People | Groups | Locations | Entry/Exit points |
|---|---|---|---|---|---|---|
| APIDIS | 7 | 1500 | 12 | 3 | 9216 | 17 |
| FIBA(MS) | 1,6,8 | 4000 | 14 | 3 | 9216 | 0 |
| FIBA(CB) | 6,8 | 5500 | 15 | 3 | 9216 | 65 |
| ISSIA | 6 | 3000 | 25 | 5 | 33480 | 0 |
| PETS'09 | 6 | 795 | 10 | 2 | 10400 | 55 |

TABLE 2
Names and characteristics of the sequences used in our experiments

To compare our approach against other state-of-the-art ones, we tested it on the PETS'09 benchmark dataset, which features pedestrians. We also ran our algorithm on single-camera sequences and compare our results to those of the recent Successive Shortest Path algorithm [8].

In the remainder of this section, we first describe these video sequences. We then discuss how we obtain image evidence and present our results[1].

### 4.1 Datasets

Team-sports players are hard to track reliably because they tend to converge towards the ball, often change their direction of travel abruptly, and wear the same uniforms when they belong to the same team. The only reliable way to identify them is to read the numbers on their shirts but, given the resolution of the images, this can only be done in relatively few frames. Furthermore, even though the color of the uniforms can be used to tell the teams apart, this information is hard to exploit at the most critical times, that is, when several players are bunched together.

Therefore, team sports sequences are among the most challenging in people tracking and we tested our approach on both basketball and soccer sequences, along with a standard pedestrian benchmarking dataset. The datasets are described in more detail below, and their parameters are summarized in Table 2

#### *4.1.1 Basketball - APIDIS*

This publicly available 1500-frame sequence [2] was acquired using 7 cameras—5 ground cameras and two "fish-eye" cameras looking from above as shown in Fig. 6—all synchronized at 25 fps. There are 12 people on the court, 2 referees and two 5-player teams. This dataset is very challenging because the lighting conditions are difficult and there are many reflections and shadows. In our experiments we use the jersey colors to provide the appearance information and use three identity groups, which consist of the two teams and the referees.

#### *4.1.2 Basketball - FIBA MS and CB*

We acquired several sequences at the 2010 FIBA World Championship for Women, using 8 cameras—4 wide-angle ones, 2 looking from above, and 2 providing close-ups—filming at 25 fps. In a typical basketball match there are two teams of 5 players, 3 referees, and 2 coaches. However, as players and referees may enter and exit the field of view of the cameras, there might be fewer people in each group at any given time. In general, the closer to the final there are, the more challenging the matches become from a tracking point of view.

We performed our experiments on long sequences from the Mali vs. Senegal match and the semi-final Czech Republic vs. Belarus match. In the remainder of this paper, we denote the first as **MS** and the second as **CB** for short. In each case, we operated in two different manners: First, we used color as the sole source of appearance information and the 3 identity groups then consisted of the two teams and referees. Second, we also attempted to read the numbers on the players' shirts whenever possible, which allows us to handle 11 groups, one per player plus one for the referees and coaches.

Fig. 6. Tracking basketball players; T-MCNF results on the publicly available APIDIS dataset. The dataset is filmed by seven cameras, two "fish-eye" camera and five side-view camera. Note that the lighting conditions are quite challenging.

#### *4.1.3 Soccer - ISSIA*

We use the publicly available ISSIA dataset [53]. It is made of 3,000 frames acquired by six cameras at a soccer match. There are 25 people, 3 referees, and two teams of 11 players, including the goal keepers whose uniform is different from that of their teammates. Due to the dataset resolution, the shirt numbers are unreadable. Hence, the appearance is based on shirt colors only. Similarly to [54], we use 5 identity groups that we denote as *referees*, *team 1*, *team 2*, *goal keeper 1* and *goal keeper 2*. A sample of the dataset is presented in Fig. 7.

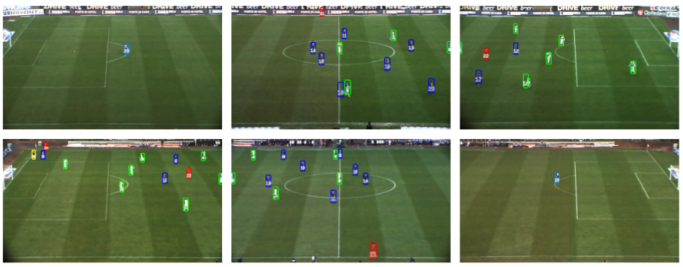


Fig. 7. Tracking results on the soccer ISSIA dataset [53]. Note that there is little overlap between cameras on the same side of the field.

1. For the supplementary material and videos, please visit: http://cvlabwww.epfl.ch/%7ebenshitr/supplementary/
2. APIDIS http://www.apidis.org/Dataset/

#### 4.1.4 Pedestrians - PETS'09

We use the publicly available PETS'09[3] dataset, for which the performance of other algorithms has been published [45]. More specifically, we tested our method on the 800-frame sequence `S2/L1`, which is filmed by 7 cameras at 7 fps, and features 10 people. In this sequence, the density of people is lower than in the sport datasets but most of the pedestrians wear similar dark clothes, which makes appearance-based identification very challenging. We therefore used only two appearance groups, one for people wearing dark clothes and the other for those wearing reddish ones.

### 4.2 Appearance Information

We exploit two distinct sources of image information, the color of the uniforms and the numbers on the players shirts. This is done as follows.

#### 4.2.1 Color Similarity

In order to estimate the correct number of groups and to have prototypes for each identity groups, we clustered the pedestrian detector results into groups. For each camera, based on the background subtraction, we extracted the foreground pixel in the bounding boxes corresponding to the detector results. We convert the foreground pixels within each box to the CIE-LAB color space, and use them to populate a $20 \times 20 \times 20$ color histogram. We repeat this process independently for each camera because they are not color calibrated.

Extracting color information from closely spaced people is unreliable because it is often difficult to correctly segment individuals. Thus, for each camera and at each time frame, we first compute an occlusion map based on the raw probability occupancy map: If a specific location is occluded with high probability in a given camera view, we do not use it in the clustering process. Similarly, at run time, we do not evaluate the color similarity of occluded persons.

We define the similarity between two color histograms using the Kullback-Leibler divergence

$$s(a,b) = \exp(-\mathrm{KL}(H_a, H_b)) \ . \tag{21}$$

Based on this similarity, we cluster the histograms into groups and obtain prototypes for each group (Templates). Finally, the similarity between this observed color histogram $O_{\text{colors}}$ and the templates $T_{\text{colors}}$ is computed as the average over the maximum matching scores from the non-occluded views $v$. We normalize this term in order to get a probability between 0 and 1

$$\varphi_i^{t,l} \propto \frac{\sum_v \exp(-\mathrm{KL}(T_{\text{colors}}, O_{\text{colors}}))}{|v|} \ . \tag{22}$$

If no appearance cue is available, due to occlusions for example, $\varphi_i^{t,l}$ is set to $\frac{1}{L}$.

3. PETS 2009: http://www.cvg.rdg.ac.uk/PETS2009

(a) (b) (c) (d)

Fig. 8. Color projections. (a) Color image. (b) Gray–scale image. (c) Color projection using the two colors of the green team. (d) Color projection using the two colors of the white team.

#### 4.2.2 Reading the Numbers

The numbers on the back of sports players are unique identifiers, and can be used to unambiguously recognize them. Within a team, the printed numbers usually share a unique color, which is well separated from the shirt color. Here we use this observation to develop a specific image binarization that improves number recognition. For every team, the shirt color $c_s$ and number color $c_n$ are obtained by clustering a shirt color patch into two clusters. Then, for each pixel we measure the distance between its color $c_p$ and these two colors: $d_s = ||c_s - c_p||, d_n = ||c_n - c_p||$. The converted gray-level pixel is defined as $255\frac{d_n}{d_s+d_n}$, which produces a white number on a black shirt, as illustrated by Fig. 8.

As for group classification, we manually extract a template for every player beforehand. At run time, applying number recognition at every position of an image would be much too expensive. Instead, we rely on people detection to select candidate positions for number reading. For each candidate position, we trim the upper $1/5$ part and the lower $1/5$ part of the bounding box, which roughly correspond to the head of the player and his legs respectively. We then search for number candidates inside the reduced bounding box, by using XOR operation between the templates and observation patches with the same size.

We select the observation patch that gives us the maximum normalized sum of pixel-wise XOR between the template and the observation and write

$$\varphi_i^{t,l} \propto \frac{T_{\text{numbers}} \oplus O_{\text{numbers}}}{|T_{\text{numbers}}|} \ . \tag{23}$$

Since numbers cannot be read often, we favor highly confident detections. Therefore, we only keep scores that are higher than a threshold, $0.8$ in our case. In other cases, we set $\varphi_i^{t,l}$ to a neutral value of $\frac{1}{L}$.

### 4.3 Implementation Details

Our system is implemented in C++ using standard libraries. To produce the Probability Occupancy Maps (POMs) we need as input, we use the publicly available POM software package [3]. It implements an algorithm that estimates ground plane occupancy from the binary output of a background

subtraction algorithm in multiple images acquired simultaneously using calibrated cameras [3]. The LP problems were formulated and optimized using the MOSEK solver [55]. We used a 3GHz PC and utilized a single core, the running time and memory consumption are summarized in Table 3 in the results section. Using our T-MCNF algorithm it is practical to process whole batches at once, on a regular PC.

### 4.4 Baselines

We use four algorithms as baselines, three for multi-camera case and one for the single-camera case.

**KSP tracker [9].** The publicly available KSP algorithm completely ignores appearance. Nevertheless, it has been shown to outperform many state-of-the-art methods on the PETS'09 dataset [45]. Note that for some of our datasets, we could not fit the full KSP grid in memory and had to prune it to reduce the number of grid cells. Given the POM detection results, we only kept locations with a probability of presence greater than $0.75$ and locations within $4$ grid cells in time or space from those.

**C-KSP tracker.** This modified version of KSP incorporates frame-to-frame appearance information by modifying the edge costs of Eq. 9. We take them to be

$$-\log\left(\frac{\rho_i(t)\zeta_{i,j}(t)L}{1-\rho_i(t)}\right) \text{ instead of } -\log\left(\frac{\rho_i(t)\varphi_i^l(t)L}{1-\rho_i(t)}\right),$$

where $\zeta_{i,j}(t)$ is an appearance term similar to the one used in [14]. It measures the probability that two locations in successive frames correspond to the same person based on the corresponding color histograms.

**DP tracker.** We use a Dynamic Program (DP) algorithm that iteratively finds and removes the shortest path from our graph of tracklets until no more path can be found from source to sink. This algorithm is similar to the Dynamic Programming of [3], [8]. The main conceptual difference with our approach is that this algorithm is greedy and, unlike ours, cannot backtrack if it makes a mistake.

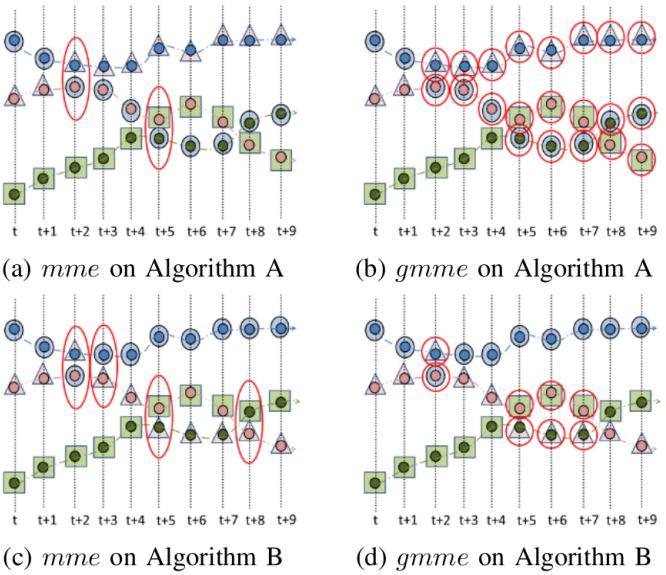
**SSP tracker [8].** The Successive Shortest Path (SSP) algorithm is a publicly available algorithm for tracking in single videos. It links detections into tracklets, uses them to form a graph, and finds a near-optimal set of trajectories using Dynamic Programming. The key difference with our approach is that, even though we also end-up connecting tracklets, the objective function we optimize is expressed in terms of occupancy probabilities in the ground plane.

### 4.5 Evaluation Metrics

A standard metric for evaluating object trackers is the Multiple Object Tracking Accuracy [56] (MOTA), whose exact definition is

$$\text{MOTA} = 1 - \frac{\sum_t (c_f(fp_t) + c_m(fn_t) + c_s(mme_t))}{\sum_t g_t}, \quad (24)$$

where $g_t$ is the number of ground truth detections, $fp_t$ the false positive count, $fn_t$ the false negative count (notated as $m_t$ - the number of miss detections in the original paper) and $mme_t$ the number of *instantaneous* identity switches. According to [57], the weighting functions are set to $c_m = c_f = 1$, and $c_s = \log_{10}$. While providing a reliable performance measure for generic tracking systems, this metric is not appropriate to evaluate applications for which identity preserving is crucial. Its $mme$ term penalizes only instantaneous identity switches, that is the frame at which two trajectories are switched, but does not account for the proportion of a trajectory that is correctly labeled over a whole sequence.




(a) $mme$ on Algorithm A (b) $gmme$ on Algorithm A

(c) $mme$ on Algorithm B (d) $gmme$ on Algorithm B

Fig. 9. Illustration of the difference between identity mismatch score $mme$ and global identity mismatch score $gmme$. We apply the two scores on two fictional tracking results A and B. The mismatches are circled in red. As can be seen, algorithm B manages to recover from its tracking mistakes. However, its $mme$ score is worse than the one of Algorithm A. Our proposed $gmme$ score favors algorithms that preserve identities.

Therefore, we introduce a new term $gmme$ for measuring the proportion of identity switches in a global manner. For every detection at every frame, the $gmme$ term is incremented if the detection label does not correspond to the ground truth identity. Thus, a trajectory with an identity switch in the middle will be counted wrong for half of its length, instead of just once for the $mme$, as explained on Fig. 9. We generate a new metric GMOTA standing for Global Multiple Object Tracking Accuracy, by replacing $mme$ by $gmme$ in MOTA, which yields

$$\text{GMOTA} = 1 - \frac{\sum_t (c_f(fp_t) + c_m(fn_t) + c_s(gmme_t))}{\sum_t g_t}. \quad (25)$$

For the sake of clarity, we will show results using both metrics and on each of their components $fp$, $fn$, $mme$ and $gmme$, in the results section.

In addition, the evaluation protocol [56] we rely on is usually used in image-space. As we use multi-camera setups, we evaluate the accuracy of the tracking in the ground plane, instead of the camera planes. To generate the plots presented in the next section, we set our distance threshold for a True

Positive to be the distance between three grid cells, which is 0.75m in the pedestrian and basketball cases and 1m in the soccer case. As shown in the supplementary material, using different distance thresholds yields the same trends.

## 5 Results

We ran the two versions of our algorithm, MCNF of Section 3.3.1 and T-MCNF of Section 3.3.2, along with the KSP, C-KSP, and DP baselines on our five sets of sequences. Fig. 11 depicts the results evaluated using both the standard MOTA metric of Eq. 24 and our modified GMOTA metric of Eq. 25. The acronyms below the graphs refer to those given to each sequence as they appear in table 2.

### 5.1 Multi-Camera Results

Using either metric, both MCNF and T-MCNF always do as well or better than KSP, which has itself been shown to outperform state-of-the-art methods on the PETS'09 sequence [45]. However, the difference is much more obvious on the GMOTA score, which is explicitly designed to penalize identity switches. Fig. 14 gives a more detailed picture of the algorithms' performance by individually ploting the components of MOTA and GMOTA $fp$, $fn$, $mme$, and $gmme$. The first three are very similar for all four methods but the latter, which is the global identity mismatch score introduced in Section 4.5, is much lower for MCNF and T-MCNF than for the others.

C-KSP behaves more erratically. It improves over KSP on the basketball data set, but performs slightly worse on the other ones. This suggests that frame-to-frame appearance cannot be depended upon to preserve identities over long periods of time. DP performs significantly worse than the other baselines and our algorithm. It tends to assign detections to a few long trajectories and cannot recover from erroneous assignments.

In short, even though the $mme$ term is low for all methods, which explains the similarity of the MOTA results, the more accurate $gmme$ metric clearly indicates that our approach preserves identity much better than the two baselines. Note, however, that the improvement is more significant on the sport sequences than in the pedestrian ones. This is because, in the latter people all wear clothes of relatively uniform color and without distinguishing marks, which makes appearance information less discriminative and failures such as those of Fig. 10 more likely. In the extreme case, if no appearance information were available, the result produced by our algorithm would be the same one as that of KSP.


Fig. 10. Failure case: Despite the global appearance model, individuals 5 and 8 are switched because of similarly colored clothes.

At the other end of the spectrum, Fig. 15 depicts what happens when using not only the color of the uniforms but also when being able to read the numbers on the players' shirts once in a while. In our experiments, the numbers can only be read once in every few hundreds of frames, in bursts of approximately 15 frames. In addition, based on the role of the players, some numbers are facing the zoomed camera much more than the others. Unsurprisingly, even though the number recognition information is only sparsely obtained, the already high GMOTA scores of Fig. 11(b) rise even further.

### 5.2 Single-Camera Results

Our approach can be also applied to single videos and we compare its performance to that of SSP [8] and KSP. For a fair comparison, we first ran the POM people detector and used its results as input to all three approaches. Since SSP expects detections and not probabilities of occupancy, we thresholded the POMs as explained at the beginning of Section 4.4. We checked that this yields better results than using the output of state-of-the-art people detectors on these sequences, in part because they feature uniform backgrounds and in part because the players are performing much more complicated motions than those of typical pedestrians.

As can be seen in Fig. 12, SSP yields a lower false positive rate at the cost of a higher false negative rate than KSP and T-MCNF. In the end, our algorithm preserves identities better than the baselines even though the results are logically less good than those obtained using multiple cameras.

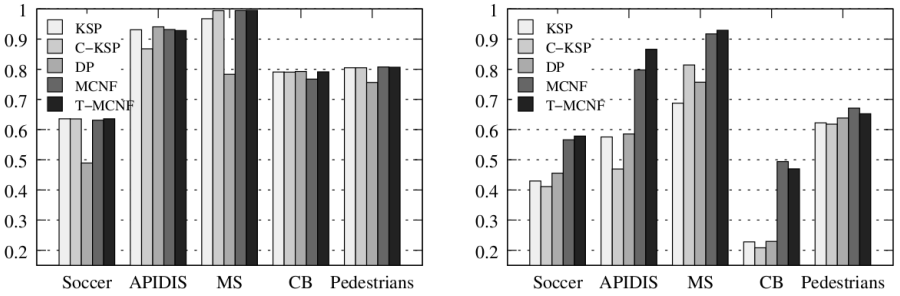
### 5.3 Computational Cost

As shown in Figs. 14 and 15. there is almost no difference between the tracking accuracy of the MCNF and the T-MCNF algorithms, However, T-MCNF is much faster and consumes far less memory, as can be seen in Table 3. The variations in computational costs are mainly due to differences in the graph sizes and the number of entry and exit points.

| Sequence | MCNF | | T-MCNF | |
|---|---|---|---|---|
| | Speed(fps) | Memory(MB) | Speed(fps) | Memory(MB) |
| PETS'09 | 255 | 2642 | 1370 | 7.5 |
| APIDIS | 44.2 | 677 | 1153 | 30 |
| FIBA MS | 3.1 | 4320 | 1277 | 104 |
| FIBA CB | 22.5 | 6332 | 863 | 1534 |
| ISSIA | 3.95 | 5285 | 187.5 | 117 |

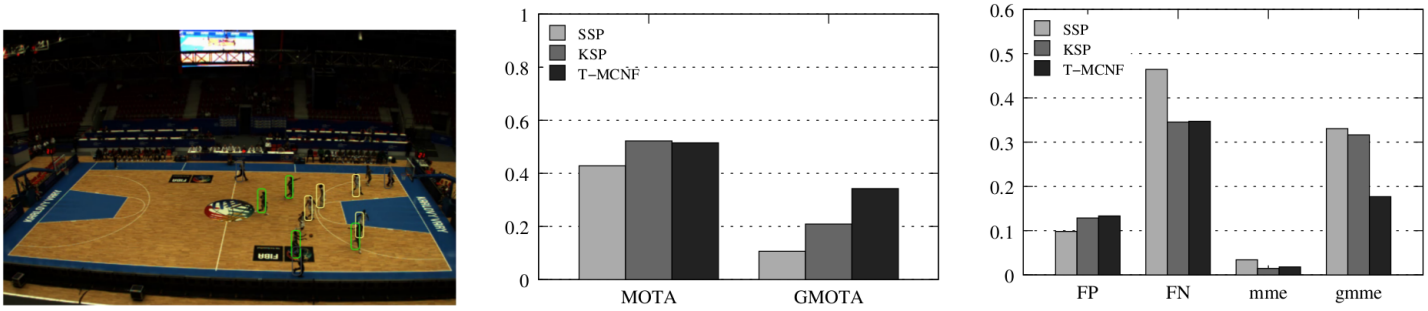
TABLE 3
Comparison between running time and memory consumption of MCNF and T-MCNF algorithms on the examined datasets. The T-MCNF algorithm is faster and consumes significantly less memory

All the results presented so far have been obtained by optimizing on whole sequences, which precludes real-time operations. If near real-time performance is required, for example so that results can be presented to spectators during a break in the action, the algorithm can be run on shorter but overlapping batches of frames. We experimented this approach on batches of 50,100,200,500 and 1000 frames, which implies a constant




(a) MOTA scores (b) GMOTA scores

Fig. 11. Comparative performance of our methods (MCNF and T-MCNF) against the baselines (KSP, C-KSP and DP) using the standard MOTA metric (a) and the new GMOTA metric (b). (a) The MOTA scores are almost the same for all methods. This is because MOTA only considers instantaneous identity switches, and weights them by $\log_{10}$. (b) The GMOTA metric penalizes identity switches more than the MOTA metric. The results clearly indicates that our methods (MCNF and T-MCNF) preserve the identities much better than the other baseline methods (KSP, C-KSP and DP). Note that higher values are better and the maximum is 1.




(a) (b) (c)

Fig. 12. Single-camera results. Comparing our approach against SSP and KSP when using only one of the video sequences from the Mali vs. Senegal FIBA basketball match.

delay of 2,4,8,20 and 40 seconds respectively. We stitched the resulting trajectories using the Hungarian algorithm [58]. As shown in Fig. 13, this results in a slight degradation with respect to the scores of Fig. 11. However, having batches with more than 200 frames does not improve the results much. This suggests that applying our approach to short overlapping batches is sufficient for obtaining excellent results, at the cost of few seconds delay.

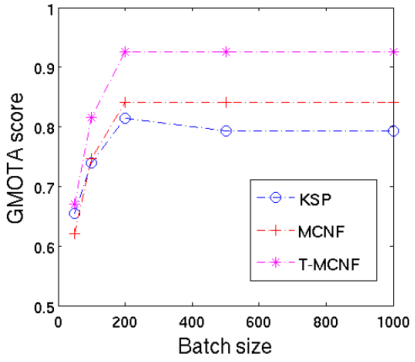



Fig. 13. Tracking performances of varying batches sizes. All methods present better tracking results using longer batches. However, short batches allows us to meet constraints on the time delay.

## 6 Conclusion

In this paper, we introduced a global optimization framework for multi-persons tracking that takes image-appearance cues into account, even if they are only available at distant time intervals. We have shown that by formalizing people's displacements as flows along the edges of a graph of spatio-temporal locations and appearance groups, we can reduce this difficult estimation problem to a standard Linear Programming one.

As a result, it does better at preserving identity over very long sequences than previous approaches, while properly handling entrances into the scene and exits from it. Furthermore, by grouping spatio-temporal locations intro tracklets, we can substantially reduce the size of the Linear Program. This yields processing times on an ordinary computer that are short enough for practical applications, such as producing statistics of team-sport players' performance during actual matches. In future work, we will focus on using these statistics for behavioral analysis and automated understanding of tactics.

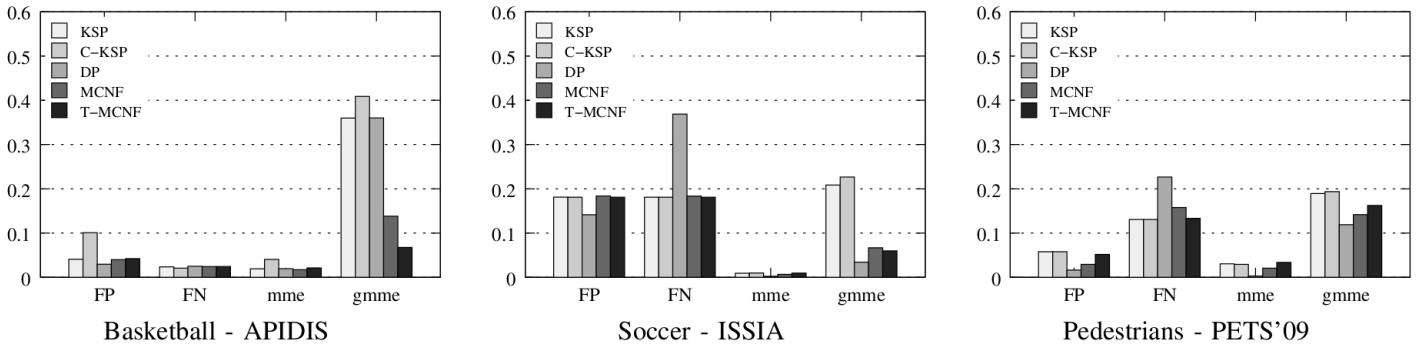



Fig. 14. Comparative performance of our methods (MCNF and T-MCNF) against the baseline (KSP, C-KSP and DP) using color as the sole source of appearance information. We plot separately each component of MOTA: the false positive and false negative rates $fp$ and $fn$, the rate of instantaneous identity switches $mme$, and the rate of global identity mismatch $gmme$. While MOTA components ($fp$, $fn$, $mme$) are similar among all the four algorithms, our methods are much better at preserving identities, as reflected by the low $gmme$ rates. In addition, the difference between the two suggested method is not significant. Note that, for these scores, lower is better.

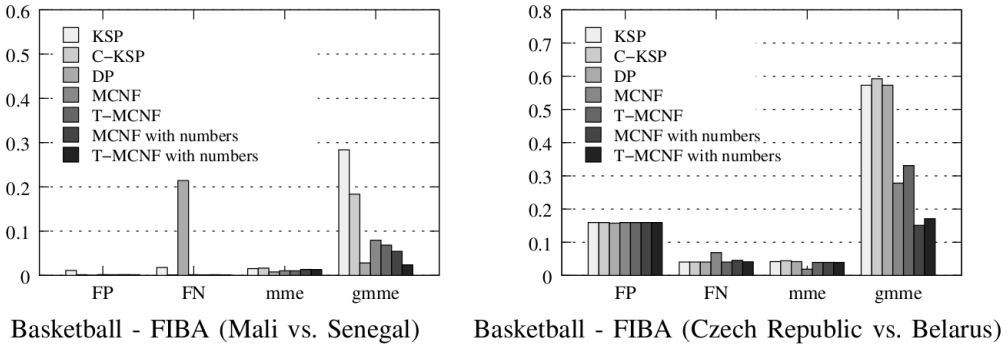



Fig. 15. Using different appearance models. MCNF and T-MCNF performance numbers when using both color similarity and number recognition against MCNF and T-MCNF numbers with color similarity only and those of the KSP and C-KSP baselines. We plot separately each component of MOTA: the false positive and false negative rates $fp$ and $fn$, the rate of instantaneous identity switches $mme$, and the rate of global identity mismatch $gmme$. Clearly, adding distinguishing identifiers, even sparsely, improves the preserving of the identities, as reflected by the low $gmme$ rates. Note that, for these scores, lower is better.

## REFERENCES


[1] M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Online Multi-Person Tracking-By-Detection from a Single Uncalibrated Camera," *PAMI*, 2010.

[2] J. Wolf, A. Viterbi, and G. Dixon, "Finding the Best Set of K Paths through a Trellis with Application to Multitarget Tracking," *IEEE Transactions on Aerospace and Electronic Systems*, 1989.

[3] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, "Multi-Camera People Tracking with a Probabilistic Occupancy Map," *PAMI*, 2008, code available at http://cvlab.epfl.ch/software/pom.

[4] P. Storms and F. Spieksma, "An LP-Based Algorithm for the Data Association Problem in Multitarget Tracking," *Computers and Operations Research*, 2003.

[5] H. Jiang, S. Fels, and J. Little, "A Linear Programming Approach for Multiple Object Tracking," in *CVPR*, 2007.

[6] A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and H. Wensheng, "Multi-Object Tracking through Simultaneous Long Occlusions and Split-Merge Conditions," in *CVPR*, 2006.

[7] L. Zhang, Y. Li, and R. Nevatia, "Global Data Association for Multi-Object Tracking Using Network Flows," in *CVPR*, 2008.

[8] H. Pirsiavash, D. Ramanan, and C. Fowlkes, "Globally-Optimal Greedy Algorithms for Tracking a Variable Number of Objects," in *CVPR*, 2011, code available at http://www.ics.uci.edu/%7edramanan/.

[9] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua, "Multiple Object Tracking Using K-Shortest Paths Optimization," *PAMI*, 2011, code available at http://cvlab.epfl.ch/software/ksp.

[10] J. W. Suurballe, "Disjoint Paths in a Network," *Networks*, vol. 4, pp. 125–145, 1974.

[11] T. Misu, A. Matsui, S. Clippingdale, M. Fujii, and N. Yagi, "Probabilistic Integration of Tracking and Recognition of Soccer Players," in *Advances in Multimedia Modeling*, 2009.

[12] H. BenShitrit, J. Berclaz, F. Fleuret, and P. Fua, "Tracking Multiple People Under Global Apperance Constraints," in *ICCV*, 2011.

[13] M. Andriluka, S. Roth, and B. Schiele, "People-Tracking-By-Detection and People-Detection-By-Tracking," in *CVPR*, 2008.

[14] A. Andriyenko and K. Schindler, "Globally Optimal Multi-Target Tracking on a Hexagonal Lattice," in *ECCV*, 2010.

[15] S. Blackman, *Multiple-Target Tracking with Radar Applications*. Artech House, 1986.

[16] J. Black, T. Ellis, and P. Rosin, "Multi-View Image Surveillance and Tracking," in *IEEE Workshop on Motion and Video Computing*, 2002.

[17] A. Mittal and L. Davis, "M2Tracker: A Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene," *IJCV*, 2003.

[18] S. Iwase and H. Saito, "Parallel Tracking of All Soccer Players by Integrating Detected Positions in Multiple View Images," in *ICPR*, 2004.

[19] M. Xu, J. Orwell, and G. Jones, "Tracking Football Players with Multiple Cameras," in *ICIP*, 2004.

[20] D. R. Magee, "Tracking Multiple Vehicles Using Foreground, Background and Motion Models," *Image and Vision Computing*, 2004.

[21] J. Giebel, D. Gavrila, and C. Schnorr, "A Bayesian Framework for Multi-Cue 3D Object Tracking," in *ECCV*, 2004.
[22] K. Smith, D. Gatica-Perez, and J.-M. Odobez, "Using Particles to Track Varying Numbers of Interacting People," in *CVPR*, 2005.
[23] K. Okuma, A. Taleghani, N. de Freitas, J. Little, and D. Lowe, "A Boosted Particle Filter: Multitarget Detection and Tracking," in *ECCV*, May 2004.
[24] Z. Khan, T. Balch, and F. Dellaert, "MCMC-Based Particle Filtering for Tracking a Variable Number of Interacting Targets," *PAMI*, 2005.
[25] C. Yang, R. Duraiswami, and L. Davis, "Fast Multiple Object Tracking via a Hierarchical Particle Filter," in *ICCV*, 2005.
[26] T. Mauthner, M. Donoser, and H. Bischof, "Robust Tracking of Spatial Related Components," in *ICPR*, 2008.
[27] D. B. Reid, "An Algorithm for Tracking Multiple Targets," *IEEE Transactions on Automatic Control*, 1979.
[28] S. Oh, S. Russell, and S. Sastry, "Markov Chain Monte Carlo Data Association for General Multiple-Target Tracking Problems," in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, 2004.
[29] W. Ge and R. T. Collins, "Multi-Target Data Association by Tracklets with Unsupervised Parameter Estimation," in *BMVC*, 2008.
[30] C. Wojek, S. Walk, S. Roth, and B. Schiele, "Monocular 3D Scene Understanding with Explicit Occlusion Reasoning," in *CVPR*, 2011.
[31] C. Wojek, S. Walk, S. Roth, , K. Schindler, and B. Schiele, "Monocular Visual Scene Understanding: Understanding Multi-Object Traffic Scenes," *PAMI*, 2013.
[32] A. Andriyenko, K. Schindler, and S. Roth, "Discrete-Continuous Optimization for Multi-Target Tracking," in *CVPR*, 2012.
[33] S. W. Joo and R. Chellappa, "A Multiple-Hypothesis Approach for Multiobject Visual Tracking," *IEEE Transactions on Image Processing*, 2007.
[34] V. K. Singh, B. Wu, and R. Nevatia, "Pedestrian Tracking by Associating Tracklets Using Detection Residuals," *IEEE Workshop on Motion and Video Computing*, 2008.
[35] Y. Li, C. Huang, and R. Nevatia, "Learning to Associate: Hybridboosted Multi-Target Tracker for Crowded Scene," in *CVPR*, 2009.
[36] C.-H. Kuo, C. Huang, and R. Nevatia, "Multi-Target Tracking by On-Line Learned Discriminative Appearance Models," *CVPR*, 2010.
[37] C.-H. Kuo and R. Nevatia, "How Does Person Identity Recognition Help Multi-Person Tracking?" *CVPR*, 2011.
[38] J. Shin, L. J. Guibas, and F. Zhao, "A Distributed Algorithm for Managing Multi-Target Identities in Wireless Ad-Hoc Sensor Networks Information Processing in Sensor Networks," in *IPSN*, 2003.
[39] R. Kondor, A. Howard, and T. Jebara, "Multi-Object Tracking with Representations of the Symmetric Group," in *In AISTATS*, 2007.
[40] J. Huang, C. Guestrin, and L. Guibas, "Fourier Theoretic Probabilistic Inference Over Permutations," *JMLR*, 2009.
[41] J. F. Henriques, R.Caseiro, and J. Batista, "Globally Optimal Solution to Multi-Object Tracking with Merged Measurements," in *ICCV*, 2011.
[42] B. Yang, C. Huang, and R. Nevatia, "Learning Affinities and Dependencies for Multi-Target Tracking Using a CRF Model," in *CVPR*, 2011.
[43] B. Yang and R. Nevatia, "An Online Learned CRF Model for Multi-Target Tracking," in *CVPR*, 2012.
[44] W. Choi and S. Savarese, "A Unified Framework for Multi-Target Tracking and Collective Activity Recognition," in *ECCV*, 2012.
[45] A. Ellis, A. Shahrokni, and J. Ferryman, "Pets 2009 and Winter Pets 2009 Results, a Combined Evaluation," in *PETS*, December 2009.
[46] Z. Wu, A. Thangali, S. Sclaroff, and M. Betke, "Coupling Detection and Data Association for Multiple Object Tracking," in *CVPR*, 2012.
[47] L. Leal-Taixe, G. Pons-Moll, and B. Rosenhahn, "Branch-And-Price Global Optimization for Multi-View Multi-Target Tracking," in *CVPR*, 2012.
[48] D. Delannay, N. Danhier, and C. Vleeschouwer, "Detection and Recognition of Sports(wo)men from Multiple Views," in *ICDSC*, 2009.
[49] W.-L. Lu, J.-A. Ting, K. P. Murphy, and J. J. Little, "Identifying Players in Broadcast Sports Videos Using Conditional Random Fields," in *CVPR*, 2011.
[50] C. Li, S. T. McCormick, and D. Simchi-Levi, "Finding Disjoint Paths with Different Path-Costs: Complexity and Algorithms," *Networks*, 1992.
[51] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear Programming and Network Flows*. John Wiley & Sons, 2010.
[52] D. Choi and I. Choi, "On the Effectiveness of the Linear Programming Relaxation of the 0-1 Multi-Commodity Minimum Cost Network Flow Problem," in *COCOON*, 2006.
[53] T. D'Orazio, M. Leo, N. Mosca, P. Spagnolo, and P. L. Mazzeo, "A Semi-Automatic System for Ground Truth Generation of Soccer Video Sequences," in *International Conference on Advanced Video and Signal Based Surveillance*, 2009.
[54] C. Poppe, S. D. Bruyne, S. Verstockt, and R. V. de Walle, "Multi-Camera Analysis of Soccer Sequences," in *International Conference on Advanced Video and Signal Based Surveillance*, 2010.
[55] "The Mosek Optimization Tools," 2010, http://www.mosek.com/.
[56] K. Bernardin and R. Stiefelhagen, "Evaluating Multiple Object Tracking Performance: the Clear Mot Metrics," *EURASIP Journal on Image and Video Processing*, 2008.
[57] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, M. Boonstra, V. Korzhova, and J. Zhang, "Framework for Performance Evaluation of Face, Text, and Vehicle Detection and Tracking in Video: Data, Metrics, and Protocol," *PAMI*, 2009.
[58] H. Kuhn, "The Hungarian Method for the Assignment Problem," *Naval Research Logistics Quarterly*, 1955.

**Horesh Ben Shitrit** received the BSc and the MSc degrees in Electrical and Electronics Engineering from Ben Gurion University in 2006 and 2009, respectively. He is currently pursuing his PhD at EPFL's Computer Vision Laboratory (CVLAB), Switzerland. His research interests include people detection, identification and tracking using multiple video cameras.

**Jérôme Berclaz** received a MS degree in Communication Systems in 2004 and a PhD in Computer Vision in 2010 from EPFL. He is now a Scientist at Microsoft, Sunnyvale, California. Prior to that, he held research positions at EPFL and Nokia Research Center, Palo Alto. His research interests include object tracking and 3D urban modeling.

**François Fleuret** received a PhD in probability from the University of Paris VI in 2000, and the habilitation degree in Applied Mathematics from the University of Paris XIII in 2006. He is the head of the Computer Vision and Learning group at IDIAP, Switzerland. Prior to that, he held positions at the University of Chicago, at INRIA, France, and at EPFL, Switzerland. He is a PAMI Associate Editor.

**Pascal Fua** received a degree from Ecole Polytechnique, Paris, in 1984 and a Ph.D. in Computer Science from the University of Orsay in 1989. He joined the Faculty of EPFL in 1996. Before that, he worked at SRI International and at INRIA Sophia-Antipolis as a Computer Scientist. His research interests include shape modeling and motion recovery from images, analysis of microscopy images, and Augmented Reality. He has (co)authored over 150 publications in refereed journals and conferences. He is an IEEE fellow and has been a PAMI associate editor. He often serves as program committee member, area chair, or program chair of major vision conferences.