Re-Identification for Improved People Tracking

F. Fleuret, H. Ben Shitrit, and P. Fua*

Abstract

Re-identification is usually defined as the problem of deciding whether a person currently in the field of view of a camera has been seen earlier either by that camera or another. However, a different version of the problem arises even when people are seen by multiple cameras with overlapping fields of view. Current tracking algorithms can easily get confused when people come close to each other and merge trajectory fragments into trajectories that include erroneous identity switches. Preventing this means re-identifying people across trajectory fragments.

In this chapter, we show that this can be done very effectively by formulating the problem as a minimum-cost maximum-flow linear program. This version of the re-identification problem can be solved in real-time and produces trajectories without identity switches.

We demonstrate the power of our approach both in single- and multi-camera setups to track pedestrians, soccer players, and basketball players.

1 Introduction

Person re-identification is often understood to mean determining whether the same person has been seen at different locations in non-overlapping camera views and other chapters in this book deal with this issue. However, a different version of the problem arises when attempting to track people over long periods of time to provide long-lived and persistent characterizations. Even though the problem

F. Fleuret
IDIAP, CH-1920 Martigny, Switzerland. e-mail: francois.fleuret@idiap.ch

P. Fua and H. Ben Shitrit
EPFL, CH-1015 Lausanne, Switzerland. e-mail: pascal.fua@epfl.ch,horesh.benshitrit@epfl.ch

* This work was funded in part by the Swiss National Science Foundation.

may seem easier than the traditional re-identification one, state-of-the-art algorithms [29, 26, 23, 36, 4, 28, 6] are still prone to producing trajectories with identity switches, that is, that combine trajectory fragments of several individuals into a single path. Preventing this and guaranteeing that the resulting trajectories are those of a single person can therefore be understood as a re-identification problem since the algorithm must understand which trajectory fragments correspond to the same individual.

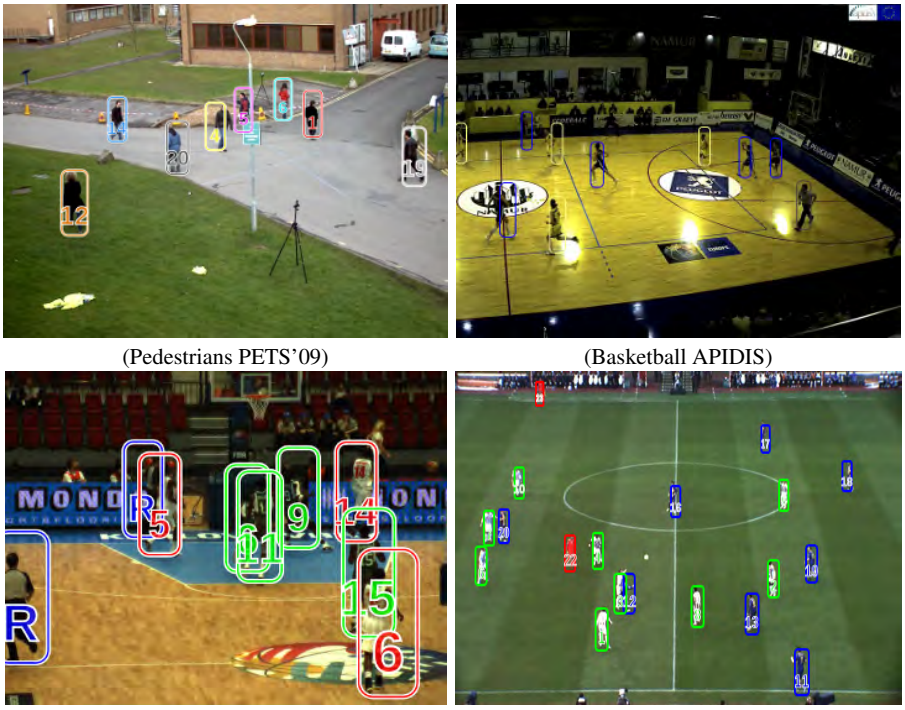
This is the re-identification problem we address in this chapter. We will show that by formulating the multi-object tracking as a minimum-cost maximum-flow linear program, we can make appearance-free tracking robust enough so that relatively simple appearance cues, such as using color histograms or simple face-recognition technology, yield real-time solutions that produce trajectories free from the above-mentioned identity switches.

More specifically, we have demonstrated in earlier work [12] that, given probabilities of presence of people at various locations in individual time frames, finding the most likely set of trajectories 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) [30]. However, this formulation completely ignores appearance, which can result in unwarranted identity switches in complex scenes. We therefore later extended it [10] to allow the exploitation of *sparse* appearance information to keep track of people's identities, 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 and soccer sequences 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, and it is precisely then that they are the least reliable [25]. Our algorithm 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 [23, 22, 3, 5].

In this chapter, we first introduce our formulation of the multi-target tracking problem as a Linear Program. We then discuss our approach to estimating the required probabilities, and present our results, first without re-identification and then with it.

2 Tracking as Linear Programming

In this section, we formulate the multi target tracking as an integer program (IP), which can be relaxed to a Linear Program (LP) and efficiently solved. We begin with the case where appearance can be ignored and then extend our approach to take it into account.



(Pedestrians PETS'09) (Basketball APIDIS)

(Basketball FIBA) (Soccer ISSIA)

Fig. 1 Representative detection results on four different datasets. The pedestrian results were obtained using a single camera while the others were obtained with multiple cameras.

2.1 Tracking without Using Appearance

We represent the ground plane by a discrete grid and, at each time step over a potentially long period of time, we compute a *Probability Occupancy Map* (POM) that associates to each grid cell a probability of presence of people, as will be discussed in § 3. We then formulate the inference of trajectories from these often noisy POMs as a Linear Program (LP) [12], which can be solved very efficiently using the K-Shortest Paths algorithm (KSP) [30]. In this section, we first introduce our LP and then our KSP approach to solving it.

2.1.1 Linear Program Formulation

We model people's trajectories as continuous flows going through an area of interest. More specifically, we first discretize the said area into K grid locations, and the time interval into T instants. Let $\mathbf{I}$ stand for the images we are processing. For any $(i,t) \in \{1,\ldots,K\} \times \{1,\ldots,T\}$ let $X_i(t)$ be a Boolean random variable standing for the presence of someone at location i at time t, and

$$\rho_i(t) = P(X_i(t) = 1 \mid \mathbf{I}) \tag{1}$$

be the posterior probability that someone stands at location k at time t, given the images.

For any location i, let $\mathscr{N}(i) \subset \{1,\ldots,K\}$ denote its neighborhood, that is, the locations a person located at i at time t can reach at time $t+1$. To model occupancy over time, let us consider a labeled directed acyclic graph with $K \times T$ vertices such as the one depicted by Fig. 2(a), which represent every location at every instant. As shown in Fig. 2(b), these locations represent spatial positions on successive grids, one for every instant t. The edges connecting locations correspond to admissible motions, which means that there is one edge $e^t_{i,j}$ from (t,i) to $(t+1,j)$ if, and only if, $j \in \mathscr{N}(i)$. Note that to allow people to remain static, we have $\forall i, i \in \mathscr{N}(i)$. Hence, there is always an edge from a location at time t to itself at time $t+1$.

As shown in Fig. 2(b), each vertex is labeled with a discrete variable m^t_i standing for the number of people located at i at time t. Each edge is labeled with a discrete variable $f^t_{i,j}$ standing for the number of people moving from location i at time t to location j at time $t+1$. For instance, the fact that a person remains at location i between times t and $t+1$ is represented by $f^t_{i,i} = 1$. These notations and those we will introduce later are summarized in Table 1.

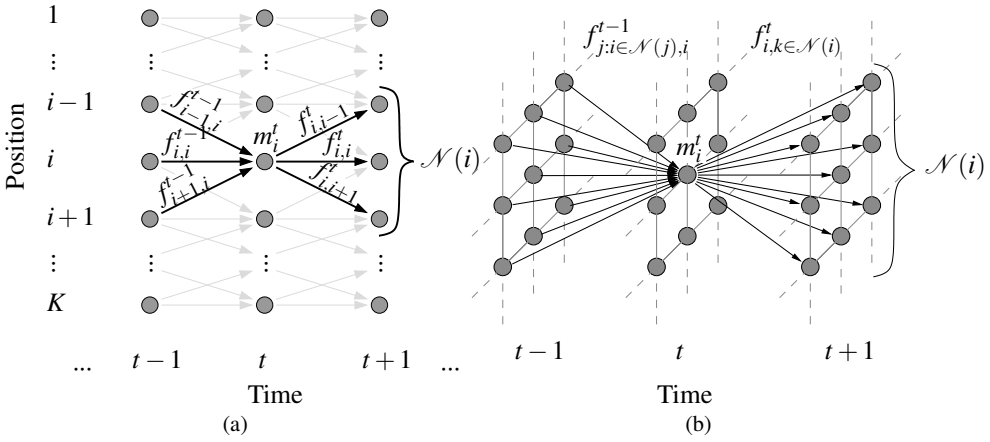


Fig. 2 Directed Acyclic Graph and corresponding flows. (a) Positions are arranged on one dimension and edges created between vertices corresponding to neighboring locations at consecutive time instants. (b) Basic flow model used for tracking people moving on a 2D grid. For the sake of readability, only the flows to and from location i at time t are printed.

In general, the number of people being tracked may vary over time, meaning some may appear inside the tracking area and others may leave. Thus, we introduce two additional nodes υ_{source} and υ_{sink} into our graph. They are linked to all the nodes representing positions through which people can respectively enter or exit the area, such as doors and borders of the cameras' fields of view. In addition, edges connect υ_{source} to all the nodes of the first frame to allow the presence of people

T	: Number of time steps.
$\mathbf{I} = (\mathbf{I}^1, \ldots, \mathbf{I}^T)$	: 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.
$\mathcal{N}(i) \subset \{1, \ldots, K\}$	: Neighborhood of location i, all locations which can be reached in one time step.
m_i^t	: Number of people at location i at time t.
$e_{i,j}^t$	: Directed edge in the graph.
$f_{i,j}^l$	: 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.

Table 1 Notations used in this chapter. When appearance is ignored as in Section 2.1, L the number of groups is equal to one and the ^{l} superscripts are omitted.

anywhere in that frame, and reciprocally edges connect all the nodes of the last frame to υ_{sink}, to allow for people to still be present in that frame. As an illustration, consider the case of a small area of interest that can be modeled using only three locations, one of which is both an entrance and exit, over three time steps. This yields the directed acyclic graph (DAG) depicted by Fig. 3. υ_{source} and υ_{sink} are *virtual locations*, because, unlike the other nodes of the graph, they do not represent any physical place.

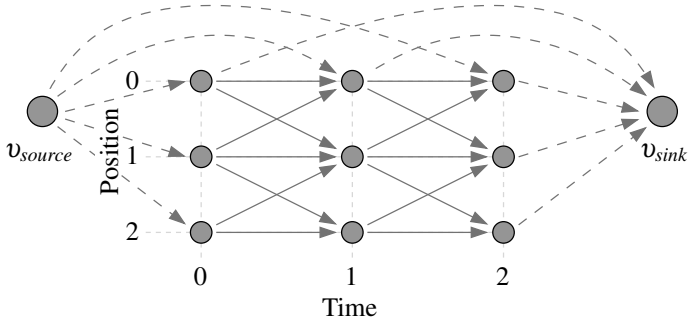


Fig. 3 Complete graph for a small area of interest consisting only of 3 positions and 3 time frames. Here, we assume that position 0 is connected to the virtual positions and therefore a possible entrance and exit point. Flows to and from the virtual positions are shown as dashed lines while flows between physical positions are shown as solid lines.

Under the constraints that people may not enter or leave the area of interest by any other location than those connected to υ_{sink} or υ_{source} and that there can never be more than one single person at each location, we showed in [12] that the flow with the maximum a posteriori probability is the solution of the Integer Program

$$
\begin{aligned}
\text{Maximize} \quad & \sum_{t,i} \log\left(\frac{\rho_i(t)}{1-\rho_i(t)}\right) \sum_{j \in \mathcal{N}(i)} f_{i,j}(t) \ , \\
\text{subject to} \quad & \forall t,i,j, \ f_{i,j}(t) \geq 0 \ , \\
& \forall t,i, \quad \sum_{j \in \mathcal{N}(i)} f_{i,j}(t) \leq 1 \ , \\
& \forall t,i, \quad \sum_{j \in \mathcal{N}(i)} f_{i,j}(t) - \sum_{k : i \in \mathcal{N}(k)} f_{k,i}(t-1) \leq 0 \ , \\
& \sum_{j \in \mathcal{N}(\upsilon_{\text{source}})} f_{\upsilon_{\text{source}},j} - \sum_{k : \upsilon_{\text{sink}} \in \mathcal{N}(k)} f_{k,\upsilon_{\text{sink}}} \leq 0 \ ,
\end{aligned}
\tag{2}
$$

where $\rho_i(t)$ is the probability that someone is present at location i at time t computed from either one or multiple-images, as will be discussed in § 3.

2.1.2 Using the K-Shortest Path Algorithm

The constraint matrix of the Integer Program of Eq. 2 can be shown to be *totally unimodular*, which means it could be solved exactly by relaxing the integer assumption and solving a Linear Program instead. However, most available solvers rely on variants of the Simplex algorithm [17] or interior point based methods [24], which do not make use of the specific structure of our problem and have very high worst case time complexities.

In [12] however, we showed that the LP of Eq. 2 could be reformulated as a *k shortest node-disjoint paths* problem on a DAG and solved by the computationally efficient K-Shortest Paths algorithm (KSP) [30]. Its worst case complexity is $O(k(m+n\cdot\log n))$, where k is the number of objects appearing in a given time interval, m is the number of edges and n the number of graph nodes. This is more efficient than the min-cost flow method of [36], which exhibits a worst case complexity of $O(kn^2 m \log n)$. Furthermore, due to the acyclic nature of our graph, the average complexity is almost linear with the number of nodes and we have observed 1,000-fold speed gains over general LP solvers.

As a result, we have been able to demonstrate real-time performance on realistic scenarios by splitting sequences in overlapping batches of 100 frames. This results in a constant 4-second delay between input and output, which is acceptable for many applications.

2.2 Tracking with Sparse Appearance Cues

The KSP algorithm of § 2.1.2 completely ignores appearance, which can result in unwarranted identity switches when people come close and separate again. If reliable appearance cues were available, this would be easy to avoid but such cues are often undependable, especially when people are in close proximity. For example, in

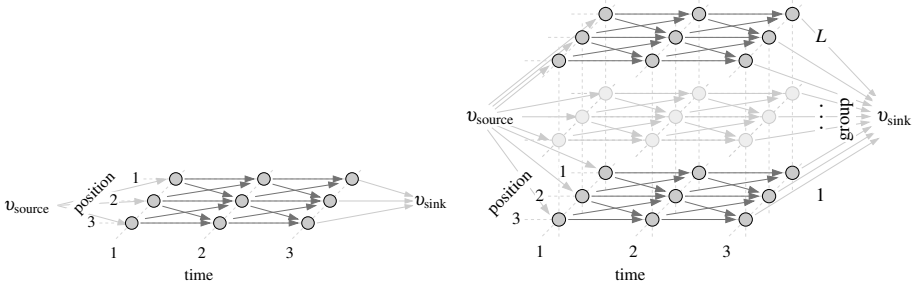


Fig. 4 Our tracking algorithm involves computing flows on a Directed Acyclic Graph (DAG). (a) The DAG of Section 2.1.1 includes source and sink nodes that allow people to enter and exit at selected locations, such as the boundaries of the playing field. This can be interpreted as a single-commodity network 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 case of the basketball players, the appearance of teammates is very similar and they can only reliably be distinguished by reading the numbers on the back of their jerseys. In practice, this can only be done at infrequent intervals.

2.2.1 Multi-Commodity Network Flow Formulation

To take advantage of this kind of sparse appearance information, we extend the framework of § 2.1.1 by computing flows on the expanded DAG of Fig. 4(b). It is obtained by starting from the graph of Fig. 4(a), which is the one we used before, and duplicating it for each possible appearance group.

More precisely, we partition the total number of tracked people into L groups and assign a separate appearance model to each. 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.

The resulting expanded DAG has $|\mathscr{V}| = K \times T \times L$ nodes. Each one represents a location i at time t occupied by a member of identity group l. Edges represent admissible motions between locations at consecutive times. Since individuals cannot change their identity, there are no edges linking groups, and therefore no vertical edge in Fig. 4(b). The resulting graph is made of disconnected layers, one per identity group. This is in contrast to the approach of § 2.1.1, which relies on a single-layer graph such as the one of Fig. 4(a).

As before, let us assume that we have access to a person detector that estimates the probability of presence $\rho_i(t)$ of someone at every location i and time t. Let us further assume that we can compute an appearance model that we use to estimate

$$\varphi_i^l(t) = \hat{P}(Q_i(t) = l \mid \mathbf{I}, X_i(t) = 1)\,, \tag{3}$$

the probability that the identity of a person occupying location i at time t is l, given that the location is indeed occupied. Here, $X_i(t)$ is a Boolean random variable standing for the actual presence of someone at location i and time t, and $Q_i(t)$ is a random variable on $\{1,\ldots,L\}$, standing for the true identity of that person. The appearance model can rely on various cues, such as color similarity or shirt numbers of sports players. In § 4, we describe in details the ones we use for different datasets.

We showed in [10, 11] that, given these appearance terms, the flows $f_{i,j}^l(t)$ with the maximum a posteriori probability are the solution of the Integer Program

$$\begin{aligned}
\text{Maximize} \quad & \sum_{t,i,l} \log\left(\frac{\rho_i(t)\varphi_i^l(t)L}{1-\rho_i(t)}\right) \sum_{j\in\mathcal{N}(i)} f_{i,j}^l(t) \\
\text{subject to} \quad & \forall t,l,i,j,\ f_{i,j}^l(t) \geq 0\,. \\
& \forall t,i,\ \sum_{j\in\mathcal{N}(i)} \sum_{l=1}^{L} f_{i,j}^l(t) \leq 1\,, \\
& \forall t,l,i,\ \sum_{j\in\mathcal{N}(i)} f_{i,j}^l(t) - \sum_{k:i\in\mathcal{N}(k)} f_{k,i}^l(t-1) \leq 0\,, \\
& \sum_{j\in\mathcal{N}(\upsilon_{\text{source}})} f_{\upsilon_{\text{source}},j} - \sum_{k:\upsilon_{\text{sink}}\in\mathcal{N}(k)} f_{k,\upsilon_{\text{sink}}} \leq 0\,, \\
& \forall t,l,\ \sum_{i=1}^{K} \sum_{j\in\mathcal{N}(i)} f_{i,j}^l(t) \leq N_l\,.
\end{aligned} \tag{4}$$

Since Integer Programming is NP-complete, we relax the problem of Eq. 4 into a multi-commodity network flow (MCNF) problem of polynomial complexity as in § 2.1.2 by making the variables real numbers between zero and one. However unlike the one of Eq. 2, this new problem is not totally unimodular. As a result, the LP solution is not guaranteed to be integral and real values that are far from either zero or one may occur [9]. In practice this only happens 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. These non-integer results can be interpreted as an uncertainty about identity assignment by our algorithm. This represents valuable information that could be used. However, as this happens rarely, we simply round off non-integer results in our experiments.

2.2.2 Making the Problem Computationally Tractable

A more severe problem is that the graphs that we have to deal with are much larger than those of § 2.1.2. The massive number of variables and constraints involved usually results in too large a problem to be directly handled by regular solvers for

real-life cases. Furthermore, the problem cannot be solved anymore using the efficient K-Shortest Paths algorithm [30].

In practice we address this problem by removing unnecessary nodes from the graph of Fig. 4(b). To this end, we first ignore appearance and run the K-Shortest Path algorithm on the DAG of Fig. 4(a). The algorithm tracks all the people in the scene very efficiently but is prone to identity switches. We account for this by eliminating all graph nodes except those that belong to trajectories found by the algorithm plus those that could be used to connect one trajectory to the other, such as the yellow vertices of Fig. 5(c). We then turn the pruned graph into a multi-layer one and solve the multi-commodity network flow problem of Eq. 4 on this expanded graph, which is now small enough to be handled by standard solvers.

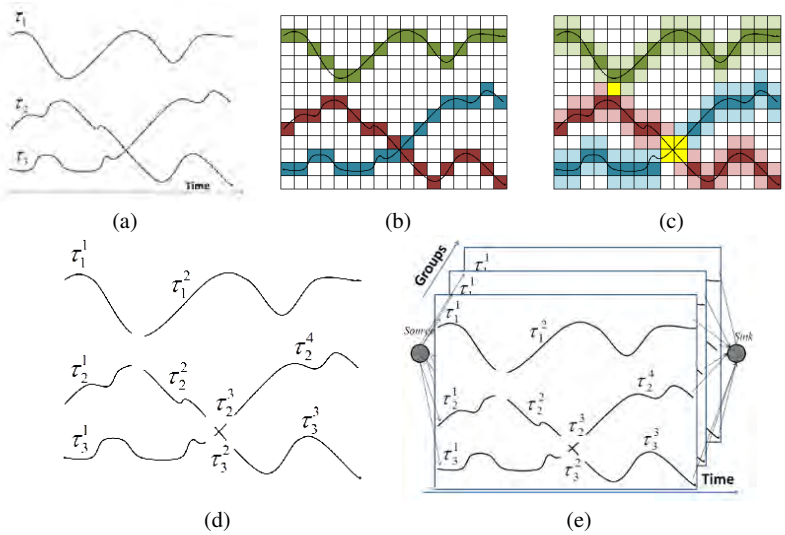


Fig. 5 Pruning the graph and splitting trajectories into tracklets. (a) For simplicity, we represent the trajectories as being one-dimensional and assume that we have three of them. (b) Each trajectory is a set of vertices from successive time instants. We assigned a different color to each. (c) The neighborhoods of the trajectories within a distance of 1 are shown in a color similar to that of the trajectory, but less saturated. The vertices that are included in more than one neighborhood appear in yellow and are used along with those on the trajectories themselves to build the expanded graph. (d) The yellow vertices are also used as trajectory splitting points to produce tracklets. Note that two trajectories do not necessarily have to cross to be split; it is enough that they come close to each other. (e) 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.

The computational complexity can be further reduced by not only removing obviously empty nodes from the graph but, in addition, by grouping obviously connected ones into *tracklets*, such as those of Fig. 5(d). The Linear Program of Eq. 4 can then be solved on a reduced graph such as the one of Fig. 5(e) whose nodes are the tracklets instead of individual locations. It is equivalent to the one of Fig. 4(b),

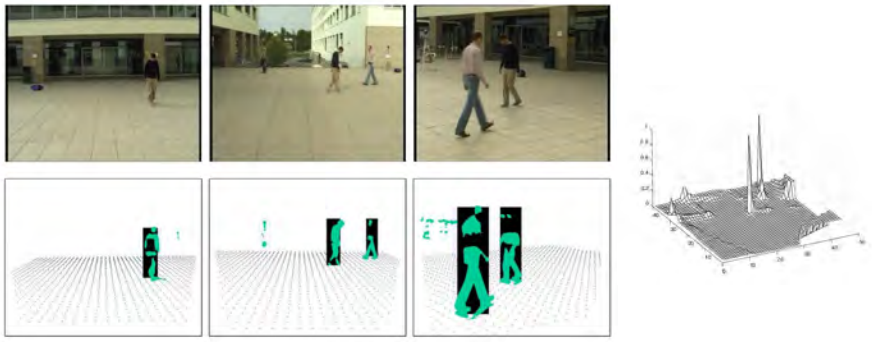
but with a much reduced number of vertices and edges [11]. In practice, this makes the computation fast enough so that taking appearance into account only represents a small overhead over not using and we can still achieve real-time performance.

3 Computing the Probabilities of Presence

The LP programs of Eqs. 2 and 4 both depend on the estimated probabilities $\rho_i(t)$ that someone is present at location i at time t. In this section, we explain how we compute these probabilities. We will discuss the appearance-based probabilities $\varphi_i^l(t)$ that the person belongs to group $l \in L$ that the program of Eq. 4 also requires in the following section.

We describe here two alternatives to estimating the probabilities of presence $\rho_i(t)$ depending on whether the background is static or not. In the first case, we can rely on background subtraction and in the second on people detectors to compute the Probability Occupancy Maps (POMs) introduced in § 2.1, that is, values of $\rho_i(t)$ for all locations i.

3.1 Detecting People against a Static Background



(a) (b)

Fig. 6 Computing probabilities of occupancy given a static background. (a) Original images from three cameras and corresponding background subtraction results shown in green. Synthetic average images computed from them by the algorithm of § 3.1 are shown in black. (b) Resulting occupancy probabilities $\rho_i(t)$ for all locations i.

When the background is static, a background subtraction algorithm can be used to find the people moving about the scene. As shown in Fig. 6(a), this results in very rough binary masks $\mathbf{B}_c$, one per image, where the pixels corresponding to the

moving people are labeled as ones and the others as zeros. Our goal then is to infer a POM such as the one of Fig. 6(b) from these. A key challenge is to account for the fact that people often occlude each other.

To this end, we introduced a generative model-based approach [22] that has been shown to be competitive against start-of-the-art ones [19]. We represent humans as cylinders that project to rectangles in individual images, as depicted by the black rectangles in Fig. 6(a). If we knew the true state of occupancy $X_i(t)$ at location i and time t for all locations, this model could be used to generate synthetic images such as those in the bottom row of Fig. 6(a). Given probability estimates $\rho_i(t)$ for the $X_i(t)$, we consider the *average* synthetic image these probabilities imply. We select them to minimize the distance between the average synthetic image and the background subtraction results in all images simultaneously. In [22], we showed that under a mean-field assumption this amounts to minimizing the Kullback-Leibler divergence between the resulting product law, and the "true" conditional posterior distribution of occupancy given the background subtraction output under our generative model.

In practice, given the binary masks $\mathbf{B}_1,\ldots,\mathbf{B}_C$ from the one or more images acquired at time t and omitting the time indices in the remainder of this section, this allows us to compute the corresponding ρ_i as the fixed point of a large system of equations of the form

$$\rho_i = \frac{1}{1+\exp\left(\lambda_i + \sum_c \Psi(\mathbf{B}_c, \mathbf{S}_c^{X_i=1}) - \Psi(\mathbf{B}_c, \mathbf{S}_c^{X_i=0})\right)}, \tag{5}$$

with λ_k a small constant that accounts for the *a priori* probability of presence in individual grid cells, $\mathbf{S}_c^{X_k=b}$ is the average synthetic image in view c given all the ρ_k for $k \neq i$ and assuming that $X_k = b$. Ψ measures the dissimilarity between images and is defined as

$$\Psi(B,S) = \frac{1}{\sigma}\frac{\|B\otimes(1-S)+(1-B)\otimes S\|}{\|S\|}, \tag{6}$$

where $\otimes$ denote the pixel-wise product of two images and σ accounts for the expected quality of the background subtraction.

Eq. 5 is one of a large system of equations whose unknowns are the ρ_i values. To compute their values, we iteratively update them in parallel until we find a fixed point of the system, which typically happens in 100 iterations given a uniform initialization of the ρ_i. Computationally, the dominant term is the estimation of the synthetic images, which can be done very fast using integral images. As a result, computing POMs in this manner is computationally inexpensive and using them to instantiate the LPs of Eqs. 2 and 4 is key to a real-time people-tracking pipe-line.

Vanila DPM Trained DPM (ours)

Fig. 7 Detection results of the DPM trained using only the INRIA pedestrian database (left) vs. our retrained DPM (right). In both cases, we use the same parameters at run-time and obtain clearly better results with the retrained DPM.

3.2 Detecting People against a Dynamic Background

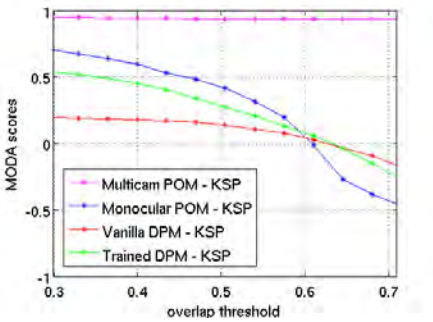
If the environment changes or if the camera moves, we replace the background subtraction based estimation of the marginal probabilities of presence $\rho_i(t)$ of Section 3.1 by the output of a modified Deformable Part Model object detector (DPM) [21]. We chose it because it has consistently been found to be competitive against other state-of-the-art approaches but we could equally well have used another one, such as [16, 8, 27].

Given a set of high-scoring detections, we assign a large occupancy probability value to the corresponding ground locations. In practice, when using the DPM detector, the top of the head tends to be the most accurately detected body part. We therefore estimate ground locations by projecting the center of the top of the bounding boxes, assumed to be at a pre-specified height above ground. The occupancy probability at locations where no one has been detected are set to a low value to account for the fact that the detector could have failed to detect somebody who was actually there. Note that these probabilities could also be learned in an automated fashion given sufficient amounts of training data.

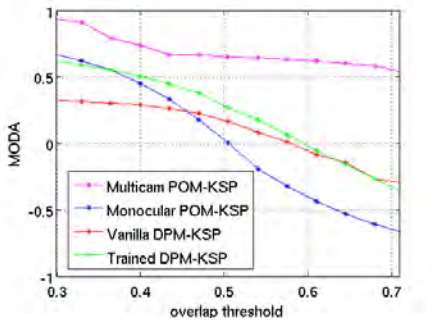
Re-training the DPM Model. We performed most of our experiments with dynamic backgrounds on tracking basketball players and found that in such a context the performance of the original DPM model [21] is insufficient for our purposes. This is due in large part to the fact that it is trained using videos and images of pedestrians whose range of motion is very limited. By contrast and as shown in Fig. 7, the basketball players tend to perform large amplitude motions.

To overcome this difficulty, we used our multi camera setup [10] to acquire additional training data from two basketball matches for which we have multiple synchronized views, which we add to the standard INRIA pedestrian database [16]. We use the bounding boxes corresponding to un-occluded players as positive examples and images of empty courts as negative ones.

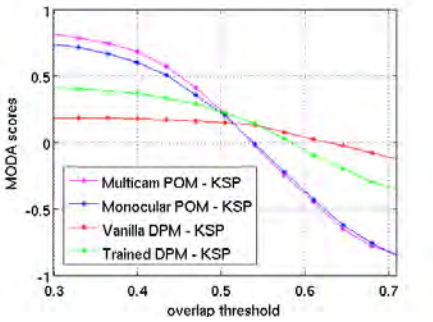
Geometric Constraints and Non-Maximum Suppression. It is well known that imposing geometric consistency constraints on the output of a people detector significantly improves detection accuracy [34, 14]. In the specific case of basketball,



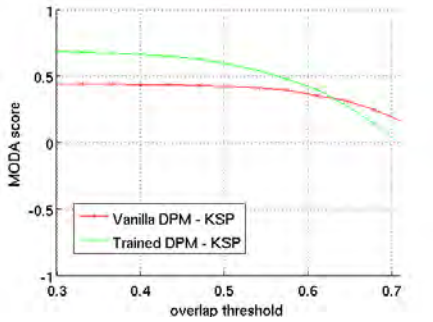
FIBA (Czech Republic vs. Belarus, Static Cameras)



FIBA (Mali vs. Senegal, Static Cameras)



APIDIS (Static Cameras)



FIBA (Moving Camera)

Fig. 8 MODA scores obtained for two different FIBA matches and one APIDIS using one or more static cameras and the different approaches to people detection of Section 3. The corresponding curves are labeled as **Multicam POM**, multi-camera generative-model; **Monocular POM**, single-camera generative-model; **Vanilla DPM**, DPM trained only with INRIA pedestrian dataset; **Trained DPM**, DPM trained using both pedestrian and basketball datasets. The MODA scores were calculated as functions of the bounding-box overlap value used to decide whether two detections correspond to the same person.

we can use the court markings to accurately compute the camera intrinsic and extrinsic parameters [31]. This allows us to reject all detections that are clearly out of our area of interest, the court in this case.

Non-Maximum Suppression (NMS) is widely used to post-process the output of object detectors that rely on a sliding window search. This is necessary because their responses for windows translated by a few pixels are virtually identical, which usually results in multiple detections for a single person. In the specific case of the DPM we use, the head usually is the most accurately detected part and, in the presence of occlusions, it is not uncommon for detection responses to correspond to the same head but different bodies. In our NMS procedure, we therefore first sort the detections based on their score. We then eliminate all those whose head overlaps by more than a fraction with that of a higher scoring one or whose body overlaps by more than a similar fraction.

3.3 Appearance-Free Experimental Results

By using the approaches described above at every time-frame independently, we obtain the $\rho_i(t)$ probabilities the KSP algorithm of § 2.1.2 requires. We tested them on two very different basketball datasets:

- The FIBA dataset comprises several multi-view basketball sequences captured during matches at the 2010 women's world championship. We manually annotated the court locations of the players and the referees on 1000 frames of the Mali vs. Senegal match and 6000 frames of the Czech Republic vs. Belarus match. Individual Frames from these matches are shown in Figs. 1 and 7. They were acquired either by one of six stationary synchronized cameras or by a single moving broadcast camera.
- The APIDIS dataset [7] is a publicly available set of video sequences of a basketball match captured by seven stationary unsynchronized cameras placed above and around the court. It features challenging lighting conditions produced by the many direct light sources that are reflected on the court while other regions are shaded. We present results using either all seven cameras or only Camera #6, which captures half of the court as shown at the top right of Fig. 1.

Fig. 8 depicts our results. They are expressed in terms of the standard MODA CLEAR metric [13], which stands for *Multiple Object Detection Accuracy* and is defined as

$$\text{MODA} = 1 - \frac{\sum_t (m_t + fp_t)}{\sum_t g_t} \,, \tag{7}$$

where g_t is the number of ground truth detections at time t, m_t the number of misdetections, fp_t the false positive count.

Following standard Computer Vision practice, we decide whether two detections correspond to the same person on the basis of whether the overlap of the corresponding bounding boxes is greater or smaller than a fraction of their area, which is usually taken to be between 0.3 and 0.7 [19]. In Fig. 8, we therefore plot our results as functions of this threshold.

When background-subtraction can be used, the generative-model approach of § 3.1 yields excellent results with multiple cameras. Even when using a single camera, it outperforms the people detector-based approach of § 3.2, in part because the generative-model explicitly handles occlusion. However, when the camera moves it becomes impractical whereas people detectors remain effective.

4 Using Appearance-Based Clues for Re-Identification Purposes

The KSP approach of § 2.1.2, which has been used to obtain the results of § 3.3, does not take appearance cues into account. Thus, it does not preclude identity switches. In other words, trajectory segments corresponding to different people can be mis-

takenly joined into a single long trajectory. This typically happens when two people come close to each other and then separate again.

In this section, we show how we can use MCNF approach of § 2.2.1 to take appearance cues into account and re-identify people from one tracklet to the next. This involves first computing the appearance-based probabilities $\varphi_i^l(t)$ of Eq. 4 that a person belongs to group $l \in L$. Note that, even though values of $\varphi_i^l(t)$ have to be provided for all locations and all times, they do not have to be informative in every single frame. If they are in a few frames and uniform in the rest, this suffices to reliably assign identities to these because we reason in terms of whole trajectories.

In other words, we only have to guarantee that the algorithms we use to process appearance return usable results once in a while, which is much easier than doing it in every frame. In the remainder of this section, we introduce three different ways of doing so and present the corresponding results.

4.1 Color Histograms

Since our sequences feature groups of individuals, such as players of the same team or referees whose appearance is similar, the simplest is to use color distribution as a signature [10]. We use a few temporal frames at the beginning of the sequence to generate representative templates for each group by manually selecting a few bounding boxes such as the black rectangles of Fig. 6 that correspond to members of that group, converting the foreground pixels within each box to the CIE-LAB color space, and generating a color histogram for each view.

Extracting color information from closely spaced people is unreliable because it is often difficult to correctly segment them. Thus, at run time, 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 to compute color similarity. Within a detection bounding box, we use the background subtraction result to segment the person. The segmented pixels are inserted into a color histogram, in the same way as for template generation. Finally, the similarity between this observed color histogram and the templates is computed using the Kullback-Leibler divergence, and normalized to get a value between 0 and 1 to be used as a probability. If no appearance cue is available, for example because of occlusions, $\varphi_i^l(t)$ is set to $\frac{1}{L}$.

4.2 Number Recognition

In team-sports, the numbers on the back of players are unique identifiers that can be used to unambiguously recognize them. However, performing number recognition at every position of an image would be much too expensive at run-time.

Instead, we manually extract templates for each player's number early in the matches when all players are standing still while the national anthem is played. Since within a team the printed numbers usually share a unique color, which is well separated from the shirt color, we create distinct shirt and number color prototypes by grouping color patches on the shirts into two separate clusters. For each prototype and each pixel in the images we want to process, we then compute distances to that prototype as shown in Fig. 9, and binarize the resulting distance image.

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

Fig. 9 Reading Numbers. (a) Color image. (b) Gray-scale image. (c,d) Distances to color prototypes for the green and white team respectively.

At run-tine, we only attempt to read numbers at locations where the probability of presence is sufficiently high. For each one, we trim the upper and lower $1/5$ of the corresponding bounding box to crop out the head and legs. We then binarize the corresponding image window as described above and search for number candidates within it by XORing the templates with image-patches of the same size. We select the patches that maximize the number of ones and take $\varphi_i^l(t)$ to be the normalized matching score. For reliability, we only retain high-scoring detections. In all other frames, we assume a uniform prior and set $\varphi_i^l(t)$ to $\frac{1}{L}$.

4.3 Face Recognition

Our third approach relies on face-detection and recognition. After estimating the probability of occupancy at every location, we run a face detector in each camera view, but only at locations whose corresponding probability of occupancy $\rho_i(t)$ is large. The face detector relies on Binary Brightness Features [1] and a cascade of strong classifiers built using a variant of AdaBoost, which has proved to be faster than the standard Viola-Jones detector [32] with comparable detection performance. For each detected face, we then extract a vector of histograms of Local Binary Pattern (LBP) features [2].

In some cases, such as when a limited number of people are known to be present, L can be assumed to be given *a priori* and representative feature vectors, or prototypes, learned offline for each person. However, in more general surveillance settings, both L and the representative feature vectors must be estimated online. We have therefore implemented two different scenarios.

- **Face Identification.** When the number of people that can appear is known *a priori*, our run-time system estimates the $\varphi_i^l(t)$ probabilities by comparing the feature vectors it extracts from the images to prototypes. These are created by acquiring sequences of the L people we expect our system to recognize and run our face-detection procedure. We then label each resulting feature vector as corresponding to one of the L people and train a multi-class RBF SVM [15] to produce a L-dimensional response vector [35]. At run-time, at each location i and time t where a face is detected, the same L-dimensional vector is computed and converted into probability $\varphi_i^l(t)$ for $1 \leq l \leq L$ [33]. In the absence of a face detection, we set $\varphi_i^l(t)$ to $\frac{1}{L}$ for all l.
- **Face Re-Identification.** When the number of people can be arbitrary, the system creates the prototypes and estimates L at run-time by first clustering the feature vectors [20, 35] and only then estimating the probabilities and finally computing the probabilities as described above.

In the Face Identification case, people's identities are known while in Face Re-Identification all that can be known is that different tracklets correspond to the same person. The second case is of course more challenging than the first.

We deployed a real-time version of our algorithm in one room of our laboratory. The video feed is processed in 50-frame batches at a framerate of 15 Hz on a quad-core 3.2 GHz PC [35]. In practice, this means that the result is produced with a constant $3.4s$ delay, making it completely acceptable for many broadcasting or even surveillance applications.

4.4 Appearance-Based Experimental Results

In this section, we demonstrate that using the appearance-based information does improve our results by significantly reducing the number of identity switches. To this end, we present here results on the FIBA and APIDIS datasets introduced in § 3.3 as well as three additional ones.

- The ISSIA soccer dataset [18] is a publicly available set of 3000-frame sequences captured by six stationary cameras placed at two sides of the stadium. They feature 25 people, 3 referees and 11 players per team, including the goal keepers whose uniforms are different from those of their teammates. Due to the low image resolution, the shirt numbers are unreadable. Hence, we consider $L = 5$ appearance groups and only use color-based cues.
- The PETS'09 pedestrian dataset features 10 people filmed by 7 cameras at 7 fps and has been used in a computer vision challenge to compare tracking algorithms.

Even though it does not use appearance cues, the KSP approach of § 2.1.2 was shown to outperform the other approaches on this data [19] and constitutes therefore a very good baseline for testing the influence of the appearance terms, which we did on the 800-frame sequence `S2/L1`. Most of the pedestrians wear similar dark clothes, which makes appearance-based identification very challenging. We therefore used only $L = 2$ appearance groups, one for people wearing dark clothes and the other for those wearing reddish ones.

- We designed the CVLab dataset to explore the use of face recognition in the context of people tracking. We used 6 synchronized cameras filming a $7m \times 8m$ room at 30 fps to acquire a training set of $L = 30$ sequences, each featuring a single person looking towards the 6 cameras, and a 7400-frame test set featuring 9 of the thirty people we trained the system for entering and leaving the room. In all these frames, 2379 instances of faces were recognized and used to compute the appearance-based probabilities.

Our results are depicted by Figs. 10 and 11 and expressed in terms of a slightly modified version of the MOTA CLEAR metric [13], which, unlike the MODA metric we used in § 3.3, is designed to evaluate performance in terms of identity preservation. MOTA stands for *Multiple Object Tracking Accuracy* and is defined as

$$\text{MOTA} = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t} \quad , \tag{8}$$

where g_t is the number of ground truth detections, m_t the number of misdetections, fp_t the false positive count and mme_t the number of *instantaneous* identity switches. In all our experiments, both KSP and MCNF algorithms yield similarly high scores [10] because this metric is not discriminative enough. To see why, consider a case where the identities of two subjects are switched in the middle of a sequence. The MOTA score is decreased because mme_t is one instead of zero, but not by much even though the identities are wrong half of the time. To remedy this, we define the metric GMOTA as

$$\text{GMOTA} = 1 - \frac{\sum_t (m_t + fp_t + gmme_t)}{\sum_t g_t} \quad , \tag{9}$$

where $gmme_t$ now is the number of times in the sequence where the identity is wrong. The GMOTA values are those we plot in Figs. 10 and 11 as function of the ground-plane distance threshold we use to assess whether a detection corresponds to a ground-truth person.

In all our experiments, computing the appearance probabilities on the basis of color improves tracking performance. Moreover, For the FIBA and CVLab dataset we show that incorporating unique identifiers such as numbers or faces is even more effective. The sequences vary in their difficulty, and this is reflected in the results. In the PETS'09 dataset, most of the pedestrians wear similar natural colors and the MCNF algorithm only delivers a small gain over KSP. The APIDIS dataset is very challenging due to strong specular reflections and poor lighting. As a result, KSP performs relatively poorly in terms of identity preservation but using color helps

greatly. In the ISSIA dataset, the soccer field is big and we use large grid cells to keep the computational complexity low. Thus, localization accuracy is less and we need to use bigger distance thresholds to achieve good scores. The best tracking results are obtained on the FIBA dataset when simultaneously using color-information and the numbers, and on the CVLab sequence when using face recognition. This is largely because the images are of a much higher-resolution, yielding better background subtraction masks. Note that since people can enter or leave the room or the court, they are constantly being identified and re-identified. The corresponding videos are available on our website at http://cvlab.epfl.ch/research/body/surv/.

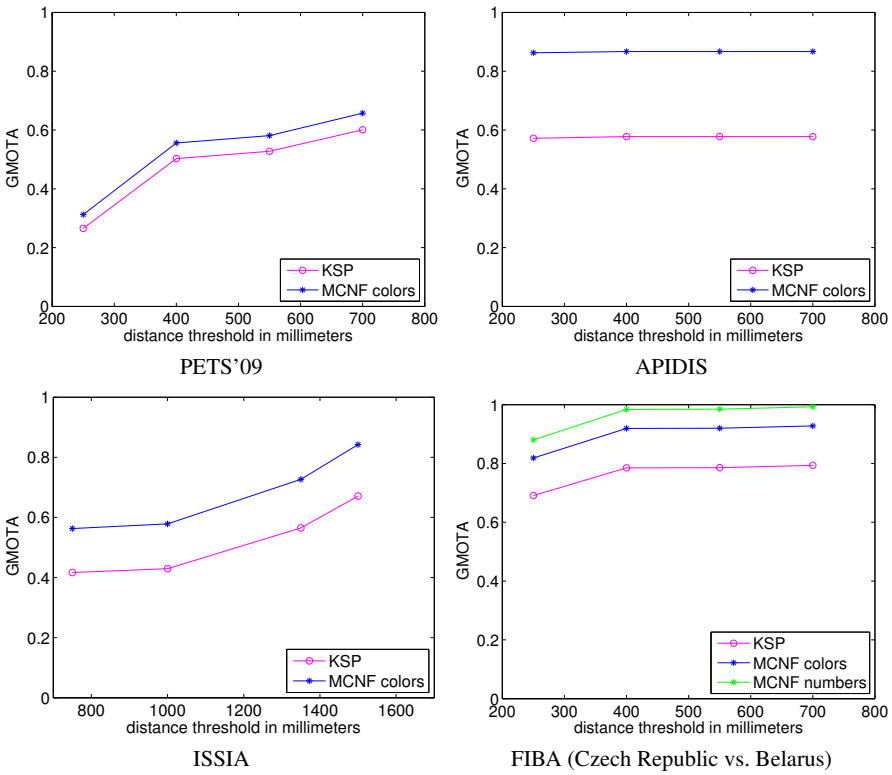
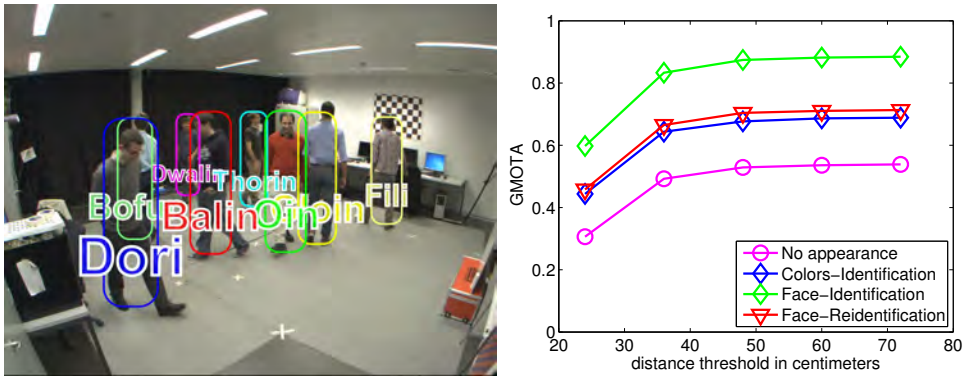


Fig. 10 Tracking results of four datasets depicted by Fig. 1 expressed in terms of GMOTA values. We compare KSP, which does not use appearance, against MCNF using color cues. For the FIBA dataset, we also show results using number recognition. The appearance information significantly improves performance in all cases.



(a) (b)

Fig. 11 Tracking results on the CVLab sequence. (a) Representative frame with detected bounding boxes and their associated identities. For surveillance purposes, the fact that names can now be associated to detections is very relevant. (b) GMOTA values. We compare KSP against MCNF using either color prototypes or face recognition. In the latter case, we give results both for the identification and re-identification scenarios. Since we use color prototypes, the color results are to be compared to face-identification ones, showing that facial cues are much more discriminative than color ones.

5 Conclusion

In this chapter, we have described a global optimization framework for multi-people tracking that takes image-appearance cues into account, even if they are only available at infrequent 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, our algorithm can identify and re-identify people reliably enough to preserve identity over very long sequences, while properly handling entrances and exits. This only requires using simple appearance-cues that can be computed easily and fast. Furthermore, by grouping spatio-temporal locations intro tracklets, we can substantially reduce the size of the Linear Program. This allows real-time processing on an ordinary computer and opens the door for practical applications, such as producing statistics of team-sport players' performance during matches.

In future work, we will focus on using these statistics for behavioral analysis and automated understanding of tactics.

References

1. Abramson, Y., Steux, B., Ghorayeb, H.: YEF Real-Time Object Detection. In: International Workshop on Automatic Learning and Real-Time (ALaRT) (2005)

2. Ahonen, T., Hadid, A., Pietikïinen, M.: Face Description with Local Binary Patterns: Application to Face Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **28**(12), 2037–2041 (2006)
3. Andriluka, M., Roth, S., Schiele, B.: People-Tracking-By-Detection and People-Detection-By-Tracking. In: Conference on Computer Vision and Pattern Recognition (2008)
4. Andriluka, M., Roth, S., Schiele, B.: Monocular 3D Pose Estimation and Tracking by Detection. In: Conference on Computer Vision and Pattern Recognition (2010)
5. Andriyenko, A., Schindler, K.: Globally Optimal Multi-Target Tracking on a Hexagonal Lattice. In: European Conference on Computer Vision (2010)
6. Andriyenko, A., Schindler, K., Roth, S.: Discrete-Continuous Optimization for Multi-Target Tracking. In: Conference on Computer Vision and Pattern Recognition (2012)
7. APIDIS European Project FP7-ICT-216023: (2008–2010). www.apidis.org
8. Barinova, O., Lempitsky, V., Kohli, P.: On Detection of Multiple Object Instances Using Hough Transforms. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(9) (2012)
9. Bazaraa, M.S., Jarvis, J.J., Sherali, H.D.: Linear Programming and Network Flows. John Wiley & Sons (2010)
10. BenShitrit, H., Berclaz, J., Fleuret, F., Fua, P.: Tracking Multiple People Under Global Apperance Constraints. In: International Conference on Computer Vision (2011)
11. BenShitrit, H., Berclaz, J., Fleuret, F., Fua, P.: Multi-Commodity Network Flow for Tracking Multiple People. IEEE Transactions on Pattern Analysis and Machine Intelligence (2013). In Press
12. Berclaz, J., Fleuret, F., Türetken, E., Fua, P.: Multiple Object Tracking Using K-Shortest Paths Optimization. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**, 1806–1819 (2011). Code available at http://cvlab.epfl.ch/software/ksp
13. Bernardin, K., Stiefelhagen, R.: Evaluating Multiple Object Tracking Performance: the Clear Mot Metrics. EURASIP Journal on Image and Video Processing (2008)
14. Bimbo, A.D., Lisanti, G., Masi, I., Pernici, F.: Person Detection Using Temporal and Geometric Context with a Pan Tilt Zoom Camera. In: International Conference on Pattern Recognition (2010)
15. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology (2011). Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm
16. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: Conference on Computer Vision and Pattern Recognition (2005)
17. Dantzig, G.B.: Linear Programming and Extensions. Princeton University Press (1963)
18. D'Orazio, T., Leo, M., Mosca, N., Spagnolo, P., Mazzeo, P.L.: A Semi-Automatic System for Ground Truth Generation of Soccer Video Sequences. In: International Conference on Advanced Video and Signal Based Surveillance (2009)
19. Ellis, A., Shahrokni, A., Ferryman, J.: Pets 2009 and Winter Pets 2009 Results, a Combined Evaluation. In: IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (2009)
20. Ester, M., Kriegel, H., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Knowledge Discovery and Data Mining (1996)
21. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object Detection with Discriminatively Trained Part Based Models. IEEE Transactions on Pattern Analysis and Machine Intelligence (2010)
22. Fleuret, F., Berclaz, J., Lengagne, R., Fua, P.: Multi-Camera People Tracking with a Probabilistic Occupancy Map. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(2), 267–282 (2008). Code available at http://cvlab.epfl.ch/software/pom
23. Jiang, H., Fels, S., Little, J.: A Linear Programming Approach for Multiple Object Tracking. In: Conference on Computer Vision and Pattern Recognition (2007)
24. Karmarkar, N.: A New Polynomial Time Algorithm for Linear Programming. Combinatorica **4**(4), 373–395 (1984)

25. Misu, T., Matsui, A., Clippingdale, S., Fujii, M., Yagi, N.: Probabilistic Integration of Tracking and Recognition of Soccer Players. In: Advances in Multimedia Modeling (2009)
26. Perera, A., Srinivas, C., Hoogs, A., Brooksby, G., Wensheng, H.: Multi-Object Tracking through Simultaneous Long Occlusions and Split-Merge Conditions. In: Conference on Computer Vision and Pattern Recognition (2006)
27. Pirsiavash, H., Ramanan, D.: Steerable Part Models. In: Conference on Computer Vision and Pattern Recognition (2012)
28. Pirsiavash, H., Ramanan, D., Fowlkes, C.: Globally-Optimal Greedy Algorithms for Tracking a Variable Number of Objects. In: Conference on Computer Vision and Pattern Recognition (2011). Code available at http://www.ics.uci.edu/%7edramanan/
29. Storms, P., Spieksma, F.: An LP-Based Algorithm for the Data Association Problem in Multitarget Tracking. Computers and Operations Research (2003)
30. Suurballe, J.W.: Disjoint Paths in a Network. Networks **4**, 125–145 (1974)
31. Tsai, R.: A Versatile Cameras Calibration Technique for High Accuracy 3D Machine Vision Metrology Using Off-The-Shelf TV Cameras and Lenses. Journal of Robotics and Automation **3**(4), 323–344 (1987)
32. Viola, P., Jones, M.: Rapid Object Detection Using a Boosted Cascade of Simple Features. In: Conference on Computer Vision and Pattern Recognition (2001)
33. Wu, T., Lin, C., Weng, R.C.: Probability Estimates for Multi-Class Classification by Pairwise Coupling. Journal of Machine Learning Research (2004)
34. Yuan, L., Bo, W., Nevatia, R.: Human Detection by Searching in 3D Space Using Camera and Scene Knowledge. In: International Conference on Pattern Recognition (2008)
35. Zervos, M., BenShitrit, H., Fleuret, F., Fua, P.: Facial Descriptors for Identity-Preserving Multiple People Tracking. Tech. Rep. EPFL-REPORT-187534, EPFL (2013)
36. Zhang, L., Li, Y., Nevatia, R.: Global Data Association for Multi-Object Tracking Using Network Flows. In: Conference on Computer Vision and Pattern Recognition (2008)