Tracking Multiple People Online and in Real Time

Ergys Ristani, Carlo Tomasi

Duke University

Abstract. We cast the problem of tracking several people as a graph partitioning problem that takes the form of an NP-hard binary integer program. We propose a tractable, approximate, online solution through the combination of a multi-stage cascade and a sliding temporal window. Our experiments demonstrate significant accuracy improvement over the state of the art and real-time post-detection performance.

1 Introduction

In many surveillance or monitoring applications, one or more cameras view several people that move in an environment. Multi-person tracking amounts to using the videos from these cameras to determine who is where at all times.

Abstractly, a multi-person tracker takes as its input a set of *observations.* At the lowest level, these are *detections* computed on each video frame by a person detection algorithm, and consist of a bounding polygon that encloses the person, together with an image position, a time stamp, an estimate of the person's velocity, and an appearance descriptor. For the sake of efficiency, intermediate stages within the tracker may form higher-level observations by grouping detections. The tracker then partitions the input observations into sets, with the intent that each set corresponds to one person and *vice versa.* The literature calls these sets *identities* or—when the linear time ordering of detections in a set needs emphasis—*trajectories.*

Two major pairs of conflicting challenges make multi-person tracking hard. Observations are *ambiguous* in that different people that look alike may be confused with each other. Conversely, changing lighting, viewpoint, and other circumstances may cause *variance of appearance* for a given person, which may not be recognized to be the same in different observations. In the other pair of challenges, person *occlusions*—whether caused by limited field of view, visual obstacles between camera and person, or algorithm failure—generate gaps in the input observations that make tracking harder. Conversely, overactive person detectors may generate *spurious observations* that confuse the tracker. All of these challenges already show up in the convenient nutshell of the single-camera case, on which this paper is focused.

1.1 Overview of the Proposed Approach

Our tracker reasons about evidence for or against any two observations being *co-identified*, that is, being assigned to the same identity. If the appearance descriptors of two observations are similar and their times and locations are consistent with typical walking speeds, evidence for their co-identity is positive. If two observations look different or occur at nearby time instants at faraway locations, evidence is negative. In the limit, "hard" evidence may be available: Simultaneous observations at faraway locations cannot possibly correspond to the same person, and yield "infinitely negative" evidence. "Infinitely positive" evidence, on the other hand, denotes an irreversible commitment to a co-identification. Infinite evidence, positive or negative, is equivalent to hard constraints on the solution.

We associate evidence to the edges of an *evidence graph* that has one node per observation and one edge for each pair of observations for which co-identity evidence is available. Multi-person tracking then becomes a *graph partitioning* problem.[1] Specifically, the set of nodes is partitioned into subsets—one subset per person identity—such that edges within sets accumulate high positive evidence, and edges between sets accumulate high negative evidence. Casting multi-person tracking as a graph partitioning problem is one of the main contributions of our paper. As described later, the resulting problem is a Binary Integer Program (BIP) when the input is finite.

Solving a BIP is NP hard. To address complexity, we introduce a cascade with two phases that operate respectively on a short (one second) and a longer (several seconds) time horizon. Each phase is in turn composed of two stages. The first stage forms groups of weakly related observations safely, reasoning about space and time in the first phase, and about appearance in the second. Once these opportunistic and conservative groups are inexpensively formed, the second stage in each phase partitions each group of observations into identities by solving the corresponding BIP, enforcing *both* space-time and appearance criteria optimally and simultaneously.

The input video in most surveillance or monitoring applications has no bounded duration. Because of this, we embed the cascade above into an online algorithm that slides a temporal window over the video stream and works in real-time.

We compare our approach with closely related literature in Section 2. Section 3 describes our approach, Section 4 discusses experimental results, and Section 5 closes with concluding remarks and a discussion of future work.

2 Relationship to Prior Work

Multi-person tracking in video has a long history. In this Section, we compare our method with other approaches based on evidence graphs.

Problem Formulation. Several methods limit evidence to pairs of observations that are consecutive in time and can achieve polynomial complexity. These formulations involve maximum-weight vertex-disjoint path cover [1, 2], a maximum

[1] Graph partitioning is often called *graph clustering* in the literature. We avoid this term to prevent confusion with other types of clustering we do in this paper.

weight independent set problem [3], bipartite matching [4–7] or some variant of network flow [8–12]. Methods that use stronger and more comprehensive evidence have demonstrated superior performance. These methods consider evidence from all observation pairs [13], observation triplets [14] or higher order relationships [15–18]. The better performance however comes at a cost of increased computational complexity due to the problem's combinatorial nature.

Problem Decomposition. Despite these difficulties, several strategies have been proposed in the literature to address computational complexity, such as (i) approximating the problem formulation, (ii) approximating the solution of the problem [1, 3, 13, 19, 20], (iii) limiting the number of edges in the observation graph [9, 10], (iv) relaxing constraints in the BIP [21–23] or (v) solving the problem in stages for efficiency [3, 11, 13, 19, 20].

We stick to the original problem formulation and propose a tractable, approximate solution for real-time multi-person tracking from a single video stream of indefinite duration. Our approximation decomposes the full problem into smaller subproblems and in two separate phases that examine short- and long-term time horizons respectively. These smaller problems can be solved exactly with a BIP solver, and the resulting solutions are pieced together *post facto*. This decomposition results into a theoretically suboptimal solution. The key challenge is then to define the decomposition so as to make partition errors unlikely. As we show in Section 3.3, many opportunities arise to perform a preliminary partitioning of observations based on position and velocity in the short-term phase, and on visual appearance in the long-term phase. If these opportunities are taken conservatively, partition errors are infrequent, as we demonstrate empirically in Section 4.

Online Algorithm. Several real-time multi-person tracking methods have been proposed [3, 9, 10, 20], but they buy speed at a noticeable cost in terms of tracking accuracy. Similarly to previous work [1, 2, 20], we use a sliding temporal window to work online. However, our extended cascade lets us process much longer windows, and thereby achieve significantly higher accuracy and resilience to occlusion. We can process video in real time at 25 frames per second on a single PC while simultaneously improving over the state of the art on existing data benchmarks. We share several technical aspects of our solution with existing approaches. Nonetheless, the multi-stage formulation and how the specific algorithms are used in each stage make our cascade novel.

The work most similar to ours is that of Zamir *et al.* [13], who formulate multi-person tracking as a sequential Generalized Minimum Clique Problem (GMCP) decomposition of a complete graph of observations. As it is common in the literature, they also decompose computation into two stages for computational efficiency, and can incorporate hard constraints on co-identity. We both allow for evidence edges between *any* two observations, and both formulations are NP-hard. However, we differ in two aspects. First, we formulate multi-person tracking jointly for all identities, while Zamir *et al.* handle identities sequentially and greedily. Second, the formulation of Zamir *et al.* mandates that one person must have exactly one observation per time frame in order to form a clique. GMCP adds one hypothetical node for every time frame where the person is missing. In scenarios of lengthy video and short person presence in the scene, most observations in a person's clique are hypothetical nodes outside the field of view, unnecessarily increasing the problem complexity. In contrast, our formulation is more general and considers only available evidence, and as a result complexity depends solely on the number of observations, not sequence length.

In summary, the contributions of this paper consist of (i) a new, general graph partitioning problem formulation which considers all the evidence between any pair of observations at once; (ii) a generic approximation method for real-time, online processing; and (iii) thorough experiments with an analysis of the trade-offs in multi-person tracking and improvements in the state of the art.

3 Multi-person Tracking

Section 3.1 introduces a batch formulation of the multi-person tracking problem in terms of a Binary Integer Program (BIP) that partitions the nodes of a graph we call the *evidence graph*. Section 3.2 discusses how to compute the weights on the edges of this graph. These weights measure evidence for or against two observations being of the same identity ("co-identity"), and the intent is that each of the sets in the computed partition corresponds to one identity. The BIP in Section 3.1 is too large to solve for realistic input, and Section 3.3 describes a four-stage cascade that computes the partition through successive refinements from a single input set. The input set contains all the person detections from an off-the-shelf detector. Section 3.4 then introduces a sliding-window method for transforming the batch solution into an online algorithm that can process video for an unbounded amount of time.

3.1 Tracking as a Binary Integer Program

Consider a set V of n observations that could be individual outputs from a person detector, or the results of aggregating co-identical detections with some other method. For a pair of observations u, v in V, let w_{uv} be a measure of the evidence for or against the hypothesis that u and v are co-identical. Section 3.2 shows how this evidence can be computed from data. For now, it suffices to say that evidence is quantified by a *correlation*, that is, a number in the set $\{-\infty, [-1, 1], +\infty\}$. Positive values indicate evidence for co-identity, negative values indicate evidence against, zero denotes indifference, and infinite values correspond to definitive evidence (hard constraints).

Let the *evidence graph* $G = (V, E, W)$ be a weighted graph[2] on V. If a correlation is available for a pair of observations in V, an edge is added to set E

[2] The graph can be directed from past to future in time, if simultaneous observations cannot be co-identical, or undirected otherwise.

for that pair, and its correlation is added to set W. In the following, we think of G as being a complete graph, but nothing in our formulation depends on this.

A multi-person tracker partitions V into sets believed to refer to distinct identities. Specifically, the partition maximizes the sum of the rewards w_{uv} assigned to edges that connect co-identical observations and the penalties $-w_{uv}$ assigned to edges that straddle identities. This graph partition problem can be rephrased as the following BIP:

$$\arg\max_{X} \sum_{(u,v)\in E} w_{uv} x_{uv} \tag{1}$$

subject to

$$x_{uv} \in \{0,1\} \qquad \forall (u,v) \in E \tag{2}$$

$$x_{uv} + x_{vt} \leq 1 + x_{ut} \qquad \forall (u,v),(v,t),(u,t) \in E\ . \tag{3}$$

The set X is the set of all possible combinations of assignments to the binary variables x_{uv}, with the interpretation that x_{uv} is 1 iff the observations u and v are co-identical. The constraints in Equation (3) enforce co-identity to be transitive: If u and v are co-identical and so are v and t, then u and t must be co-identical as well.

Considering all pairwise correlations has the advantage that even when some edges carry negative correlation, the nodes they connect can still be co-identical if the overall reward of the set is positive, and *vice versa*.

Finding an optimal solution to this BIP is NP-hard [24] and the problem is also hard to approximate [25]. The best known approximation algorithm achieves an approximation ratio of 0.7664 [26], but its semi-definite program formulation makes it slow for practical consideration. Other algorithms exist and we describe one of them in Section 4.4, but they come with no quality guarantees. These results suggest that one needs to look at the special properties of the multi-person tracking problem to find an efficient solution, as we do in Section 3.3. In Section 3.2, we first consider how to compute the correlations w_{uv}.

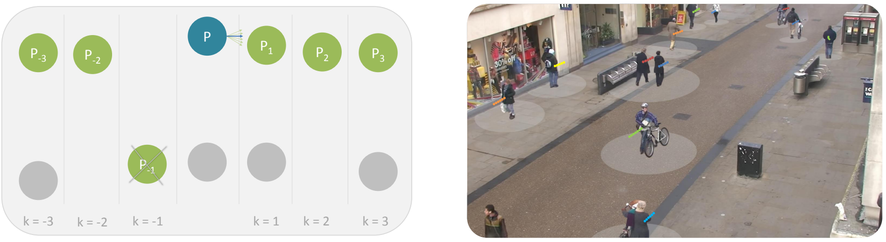
3.2 Features and Measures of Evidence

To manage computational complexity, the BIP defined in Section 3.1 is solved first over individual person detections within small time horizons, and then within longer time horizons and over short trajectories called "tracklets," as explained in Section 3.3 below. In this section, we show how observations of both types are described, and how correlations are computed for pairs of them.

Each person detection $D = (\boldsymbol{\phi}, \mathbf{p}, t, \mathbf{v})$ is described by its appearance feature $\boldsymbol{\phi}$, position $\mathbf{p}$, time stamp t, and estimated velocity[3] $\mathbf{v}$. We use an HSV color histogram to describe a person's appearance, but different descriptors can be used with no other modification of the proposed methods.

Co-identity evidence from space and time information comes mainly from the assumption that people are limited in their speed, and reasoning about person

[3] Velocity is a vector, and its norm is called the *speed*.



(a) (b)

Fig. 1. (a) Velocity estimation of the blue detection for $m = 3$. Circles are detections, the horizontal dimension is time, and the vertical one stands for 2D space. Green detections are the nearest detections in space to the blue detection for each k. Detections in grey are not considered for velocity estimation. Detection $\mathbf{p}_{-1}$ is discarded because the speed required to reach the blue detection from it exceeds a predefined limit. The green vectors are the velocities computed for each blue-green detection pair and the blue vector is the estimated velocity. (b) Circles enclose disjoint space-time groups, found from assumed bounds on walking speed.

speed requires converting image coordinates to world coordinates. To this end, we assume that people move on a planar region and that a homography is available between the world and the image.

The velocity of a detection at position $\mathbf{p}$ in video frame i is estimated as follows. For each frame k in $[i-m, i+m]$ (where m is a small integer) and $k \neq i$, determine the detection $\mathbf{p}_k$ that is nearest (in space) to $\mathbf{p}$. Compute the velocities from each pair $(\mathbf{p}, \mathbf{p}_k)$, and discard those that violate a predefined speed limit. The velocity estimate for the detection at $\mathbf{p}$ is then the component-wise median of the remaining velocities. See Figure 1(a).

Tracklets (short trajectories of detections) have somewhat more complex descriptors than individual detections, because they extend over time. Specifically, a tracklet descriptor $\tilde{T} = \{\tilde{\boldsymbol{\phi}}, \tilde{\mathbf{p}}^s, \tilde{\mathbf{p}}^e, \tilde{t}^s, \tilde{t}^e, \tilde{\mathbf{v}}\}$ contains an appearance feature $\tilde{\boldsymbol{\phi}}$ that is equal to the median appearance of its detections. The descriptor also contains the start point $\tilde{\mathbf{p}}^s$ and end point $\tilde{\mathbf{p}}^e$ of the tracklet, its start time $\tilde{t}^s$ and end time $\tilde{t}^e$, and its velocity $\tilde{\mathbf{v}}$. Since tracklets are short, we assume that their detections are on a straight line and we approximate the velocity of the tracklets as follows:

$$\tilde{\mathbf{v}} = \frac{\tilde{\mathbf{p}}^e - \tilde{\mathbf{p}}^s}{\tilde{t}^e - \tilde{t}^s} \ . \tag{4}$$

Given two detections $D_1 = (\boldsymbol{\phi}_1, \mathbf{p}_1, t_1, \mathbf{v}_1)$ and $D_2 = (\boldsymbol{\phi}_2, \mathbf{p}_2, t_2, \mathbf{v}_2)$, we first define two simple space-time and appearance affinity measures for them in $[0, 1]$, and then combine the affinities into a single correlation measure.

Specifically, the space-time affinity of D_1 and D_2 is:

$$s_{st} = \max[1 - \beta\,(e(D_1, D_2) + e(D_2, D_1)), 0] \tag{5}$$

where $e(D_1, D_2) = \|\mathbf{q}_1 - \mathbf{p}_2\|_2$ measures the error between the position $\mathbf{p}_2$ of detection D_2 and the estimated position $\mathbf{q}_1 = \mathbf{p}_1 + \mathbf{v}_1(t_2 - t_1)$ of detection D_1

at time t_2. The parameter β controls how much error we are willing to tolerate. Setting a lower value for β is helpful for handling long occlusions. We use $\beta = 1$.

The appearance affinity between D_1 and D_2 is:

$$s_a = \max[1 - \alpha\, d(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2), 0] \tag{6}$$

where $d(\cdot)$ is a distance function in appearance space. We use the earth mover's distance [27] in our experiments to compare HSV histograms, and set $\alpha = 1$.

A sigmoid function maps affinities to correlations smoothly, except in extreme cases:

$$w = \begin{cases} -\infty & \text{if } s_a s_{st} = 0 \\ +\infty & \text{if } s_a s_{st} = 1 \\ -1 + \frac{2}{1+\exp(-\lambda(s_a s_{st} - \mu))} & \text{otherwise} \end{cases} . \tag{7}$$

The parameter λ determines the width of the transition band between negative and positive correlation, and μ is the value that separates them. We use $\mu = 0.25$, assuming that $s_a = s_t = 0.5$ indicates indifference.

The definition of appearance affinity remains the same for tracklets, once appearance descriptors are modified as explained earlier. For space-time affinities, the position error $e(\cdot, \cdot)$ is redefined to measure the discrepancy between a tracklet's start point and the estimated start point as determined from the end point of the other tracklet: $e(\tilde{T}_1, \tilde{T}_2) = \|\tilde{\mathbf{q}}_1^s - \tilde{\mathbf{p}}_2^s\|_2$ where $\tilde{\mathbf{q}}_1^s = \tilde{\mathbf{p}}_1^e + \tilde{\mathbf{v}}_1(\tilde{t}_2^s - \tilde{t}_1^e)$.

3.3 The Cascade

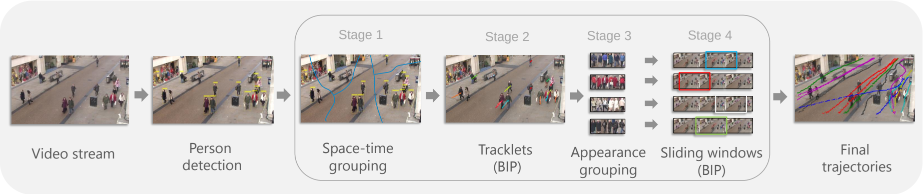


Fig. 2. The proposed processing pipeline.

In preparation for the online method of Section 3.4, we describe a cascade that allows solving the graph partitioning problem defined in Section 3.1 approximately and efficiently over a *temporal window* several seconds long. The longer the window, the longer the occlusions through which identities can be retained. Although we lose theoretical guarantees of optimality, we exploit the special structure of multi-person tracking to decompose the large BIP problem from Section 3.1 into manageable chunks that are unlikely to take us far from the optimal solution.

Our cascade has two simpler phases divided into two stages each. The first phase partitions detections over short time horizons and results into *tracklets*, short sequences of detections that can be safely connected to each other based on both appearance and space-time affinities. The second phase reasons over the entire temporal window, and partitions tracklets into identities (a.k.a. trajectories). Each phase has in turn a first stage that does a preliminary partitioning done safely by simple means in order to reduce the size of the BIP in that phase, and a second stage that solves a BIP exactly to utilize all evidence optimally. The four stages are now described in turn.

Space-Time Groups. The first stage divides the entire video sequence into 1-second intervals and uses hierarchical agglomeration [28] to group detections within each interval into *space-time groups* (Figure 1(b)). Initially, each detection is in a separate group. The algorithm then repeatedly merges the pair of groups that are closest to each other in space until k_i space-time groups are formed for time interval i. We set k_i to one half of the expected number of visible people in the given time interval, estimated as the ratio between the total number of detections and the number of frames in the interval. Because of the conservative choice of k_i, it is unlikely that observations that belong together end up in different groups. Even if they do, one person will end up split into different identities, and the trajectory stage, described later, has an opportunity to undo the split.

Tracklets. The second stage solves a BIP exactly for the observations of each space-time group, using the correlations (7) for evidence. The resulting partitions are called *tracklets*, and are at most one second long by construction. Solving exact BIPs on space-time groups ensures that both appearance and space-time evidence are used optimally within this short time horizon. Missing detections are recovered using interpolation or extrapolation and tracklets shorter than 0.2 seconds are discarded as false positives.

Appearance Groups. The third stage reasons in appearance space and groups tracklets from the entire temporal window into appearance groups that will be processed independently of each other in the fourth stage. Non-parametric methods for discovering appearance groups [29] are a good fit for this stage. However, we use k-means and set the number k of clusters manually for simplicity.

The wholesale splitting of identities across different appearance groups is an irrecoverable error. However, appearance grouping is again conservative, in that two observations are grouped whenever they are even just loosely similar. The main assumptions in this stage are that a person's appearance can have only short-lived variations (*e.g.*, partial occlusions or shadows) and that person appearance does not change suddenly and dramatically (*e.g.*, a person putting on a rain coat while hidden behind an obstacle). The conservative nature of this stage typically prevents identity-split errors, and a few incorrectly assigned observations can be handled similarly to false positives and false negatives.

Trajectories. The last stage in the cascade solves a separate BIP (exactly) for all the tracklets in each appearance group and within the entire temporal window, again using both space-time consistency and appearance similarity as evidence. Missing tracklets for each trajectory are inferred using interpolation, and very short trajectories (shorter than 2 seconds) are discarded as false positives. The reduction in the size of the BIPs in the second and fourth stage of our cascade allows processing long temporal windows of data in real time, as Section 4 illustrates experimentally.

3.4 Unlimited Time Horizon

Typical surveillance video streams are unbounded in length and require real-time, online processing. To turn the method described so far into an online algorithm we employ a sliding temporal window. The temporal extent of the window is set ahead of time—and depending on application—so that the observations in it can be processed in real time. Video frames stream in continuously, and an off-the-shelf person detector provides the needed detections. One-second-long tracklets are continuously formed by stages 1 and 2 of the cascade, and added to the input data. Once a window is processed completely as explained next, it is advanced by half its temporal extent.

All the tracklets that are at least partially contained in the first window are fed to the second phase of the cascade. Stages 3 and 4 form partial trajectories, and missing and spurious observations are handled as explained in Section 3.3. Partial trajectories are never undone, but they can be extended from data in subsequent windows. In windows after the first, the elementary input observations for stage 3 are all the tracklets and all the partial trajectories whose temporal extents overlap the current window. Except for this difference, the computations are the same as in the first window. This process repeats forever. Figure 2 illustrates the complete processing pipeline.

The experiments in Section 4 illustrate that the cascade allows processing rather long windows. As a consequence, we can often successfully connect people identities across much longer occlusions than in previous literature, because corresponding tracklets before and after an occlusion are more likely to occur together in some window. For instance, our experiments show that for a medium crowded scene the sliding window can be as long as 32 seconds if tracklets are split into ten appearance groups, while still achieving real-time performance.

4 Experiments

4.1 Datasets and Performance Measures

We evaluate our algorithm on three standard single-camera datasets for multi-person tracking: PETS2009 [30], Town Center [20] and Parking Lot [13]. We used the PETS2009-S2L1 View 1 sequence, which has a resolution of 768 x 576 pixels and consists of 798 frames at 7 fps (117 seconds). The scene is not heavily crowded, but the low predictability in people's motion and a few long occlusions behind a lamp post makes the sequence challenging. The Town Center sequence is more challenging because it is longer, more crowded, and has longer occlusions. Occlusions in this sequence are mainly caused by people walking very close to each other. The sequence has a resolution of 1920 x 1080 pixels and consists of

4500 frames at 25 fps (180 seconds). The Parking Lot sequence consists of 998 frames at 30 fps (33.26 seconds) and has a resolution of 1920 x 1080 pixels. This sequence is challenging because it is filmed from an oblique angle and several people have similar appearance. Also, people walk close to each other in parallel causing long occlusions, both partial and full.

We use the standard Multiple Object Tracking Accuracy (MOTA) score [31] to evaluate the performance of our algorithm. This score combines the number of false positives $f_p(t)$, false negatives $f_n(t)$, and identity switches $id(t)$ over all frame indices t as follows:

$$MOTA = 1 - \frac{\sum_t (f_p(t) + f_n(t) + id(t))}{\sum_t g(t)} \tag{8}$$

where $g(t)$ is the ground-truth number of people in frame t. MOTA is widely accepted in the field as one of the principal indicators of a tracker's performance.

	PETS2009	
	MOTA	IDsw
Berclaz [10]	80.00	28
Shitrit [9]	81.46	19
Andriyenko [19]	81.84	15
Henriques [6]	87.95	10
Izadinia [11]	90.70	-
Zamir [13]	91.50	8
Ours	**93.34**	**1**

	Town Center	
	MOTA	IDsw
Benfold [20]	64.9	259
Zhang [12]	65.7	114
Leal-Taixe [32]	67.3	86
Izadinia [11]	75.70	-
Zamir [13]	75.59	-
McLaughlin [33]	76.46	-
Ours	**78.43±0.29**	**68**

	Parking Lot	
	MOTA	IDsw
Izadinia [11]	88.90	-
Zamir [13]	92.27	1
Ours	**94.20**	**1**

Table 1. Multi Object Tracking Accuracy (MOTA) and ID switches on three standard datasets. MOTA variance for the Town Center sequence was caused by differences in appearance groups in different runs and resulting from the randomness of the k-means clustering algorithm.

In Table 1 we present results for all sequences. We outperform state of the art methods in MOTA and identity switches. For a fair comparison, we use the detections used in previous work [13], courtesy of the authors. All evaluations are done using the CLEAR MOT evaluation script [34] and we use the standard 1 meter acceptance threshold.

In the PETS2009 sequence we use a long temporal window of 20 seconds and one appearance group since the scene is not crowded. We allow tracklets to be at most 10 frames in this sequence due to its low frame rate. The total running time of our method, not accounting for person detection, is 38 seconds. In the Town Center sequence we use a temporal window of 12 seconds and 5 appearance groups because the sequence is more crowded. Tracklets have lengths of at most 20 frames. The total running time on this sequence is 176 seconds, 120 of which were spent finding all tracklets. In the Parking Lot sequence we use a temporal window of 6 seconds and tracklets are at most 20 frames long. We used one appearance group in this sequence since it is short. The total running time on this sequence is 34 seconds.

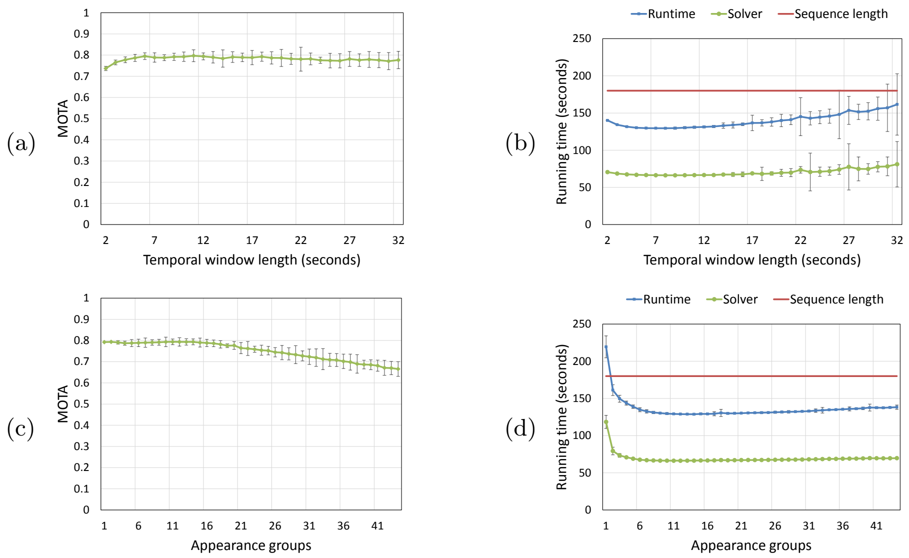


Fig. 3. MOTA scores (a, c) and running times (b, d) as functions of the length of the sliding window (a, b) and the number of appearance groups (c, d) for the Town Center sequence. Solver time indicates how much time was spent for assembling and solving all the Binary Integer Programs. The total running time also includes the time for computing correlations, but does not account for person detection. Figures (a) and (b) are for ten appearance groups, and Figures (c) and (d) are for 8-second sliding windows. Best viewed on screen.

4.2 Window Length, Accuracy, and Runtime

Figures 3(a) and 3(b) show the dependency of tracking accuracy and running time on the length of the sliding window for the Town Center Sequence with 10 appearance groups. We ran several experiments on this sequence by progressively elongating the temporal window.

Figure 3(a) shows that after the temporal window length is increased beyond 3 seconds, which corresponds to the typical occlusion length in the scene, there is no significant improvement in the quality of the solution. The variations in the graph are caused by differences in the appearance groups that the k-means algorithm finds in each window. The slight decrease in the scores for windows longer than 19 seconds is because the parameter β in Equation (5) also influences how large partitions can grow in time.

Figure 3(b) shows how the sliding window length affects the running time. Appearance grouping allows us to achieve an unprecedented temporal window length for real-time computation.

4.3 Appearance Groups, Accuracy, and Runtime

Figures 3(c) and 3(d) show the dependency of tracking accuracy and running time on the number of appearance groups for the Town Center sequence with a temporal window of 8 seconds.

Figure 3(c) shows that even a moderately high number of appearance groups, around 20, has negligible harmful effects on the accuracy of the tracker. When the number of appearance groups is increased further, the accuracy measure starts to decay because identities are split into separate groups. The fluctuations in the graph are again caused by the k-means algorithm, which over-clusters in windows that contain few tracklets.

Figure 3(d) shows that the overall running time is greatly reduced when we go from 1 to about 5 appearance groups, while the MOTA score only drops from 79% to 78.4% (Figure 3(c)). Increasing the number of appearance groups further yields marginal reductions in running time. The slight increase in total runtime for more than 20 appearance groups is caused by the k-means algorithm, whose complexity increases with k.

4.4 Approximate and Exact Graph Partitioning Solvers

We explore the trade-off between accuracy and runtime for different combinations of solvers for graph partitioning. We demonstrate that approximating the solution of multi-person tracking by piecing together exact solutions of small subproblems is qualitatively better than algorithms with no optimality guarantees, while still achieving real-time performance.

Three algorithms for graph partitioning have been recently proposed in the literature [35], namely: Expand-and-Explore, Swap-and-Explore, and Adaptive Label Iterative Conditional Modes (AL-ICM). We use the latter in our experiment because of its speed and ability to scale to large problems. Given a labeling vector $L = \{1, 2, \ldots\}^n$ the algorithm assigns a label l_u to observation u so as to minimize the following energy function:

$$E(L) = \sum_{uv} w_{uv} \mathbf{1}_{[l_u \neq l_v]} \tag{9}$$

where $\mathbf{1}_{[P]}$ is 1 when P is true and 0 otherwise. Minimizing this energy function is equivalent to maximizing rewards and minimizing penalties in Equation (1). This energy is lowered when observations supported by negative correlation are labeled differently and when observations supported by positive correlation are labeled identically. This discrete energy minimization formulation has the advantage that the labeling vector L consists of n variables whereas the co-identity matrix X in our formulation consists of n^2 variables. This allows AL-ICM to scale to $n \geq 100,000$ observations.

AL-ICM is a greedy search algorithm. In each iteration, every variable is assigned the label that minimizes the energy, conditioned on the current label of the other variables. While ICM requires a fixed number of labels [36], AL-ICM handles a varying number of labels as follows: conditioned on the current

		PETS2009			Town Center			Parking Lot		
	Method	MOTA	Runtime	Solver	MOTA	Runtime	Solver	MOTA	Runtime	Solver
	Izadinia [11]	90.70	-	-	75.70	-	-	88.90	-	-
	Zamir [13]	91.50	-	-	75.59	-	-	92.27	-	-
Ours	**AL-ICM**	91.34	**9.33**	**0.31**	77.78 ± .35	**107.54**	**3.52**	93.33	**15.69**	**0.39**
	AL-ICM-NoGroup	92.20	10.68	0.45	78.46	284.73	18.86	93.92	28.00	1.11
	BIP	**93.18**	17.06	8.06	78.43 ± .29	177.17	73.10	**94.20**	33.59	20.40
	BIP-NoGroup	**93.18**	27.43	16.87	**78.87**	25725.82	23444.58	**94.20**	334.45	307.39

Table 2. Different combinations of solvers evaluated on three standard datasets. The length of each sequence is 117, 180 and 33.26 seconds respectively. Solver time indicates how many seconds were spent for solving graph partitioning problems in each sequence. The total running time also includes the time for computing correlations, but does not account for person detection.

labeling, each observation is assigned to the most rewarding partition, or to a new partition if penalized by all current partitions. The algorithm terminates either when the energy cannot be minimized further or when a predefined number of iterations is reached.

We construct two methods based on this algorithm. Method AL-ICM uses the greedy algorithm in stages 2 and 4 of the cascade, and space-time and appearance grouping in stages 1 and 3. Method AL-ICM-NoGroup uses the greedy algorithm but no grouping, thus only stages 2 and 4 of the full cascade.

We refer to our full algorithm as BIP, and we compare it also to a method we call BIP-NoGroup, that is, stages 2 and 4 of the cascade without space-time and appearance grouping. Performance metrics for all methods on three sequences are presented in Table 2.

Accuracy. All our methods consistently outperform the state of the art. Even method AL-ICM is on par, if not better than the state of the art, although it can be penalized by mistakes due to grouping heuristics and the suboptimal greedy algorithm. The differences in accuracy between our methods that use grouping and their corresponding version without grouping is minimal. This confirms that stages 1 and 3 of our cascade can be used in practice, safely and with negligible harmful effects. It is also worth noting that piecing together optimal solutions of small problems is superior to combining approximate solutions of small problems, which is common in the literature: Both BIP and BIP-NoGroup perform better than AL-ICM and AL-ICM-NoGroup, respectively.

Runtime. It is not surprising that the AL-ICM algorithm is much faster than the BIP solver. AL-ICM is a greedy algorithm and does not require assembling and solving a BIP with a quadratic number of variables and a combinatorial number of constraints. We note that the use of grouping heuristics is crucial for improving runtime performance; methods that do not use heuristics need to compute large and full correlation matrices. While the best time performance is that of AL-ICM, our BIP method is also fast enough to work in real-time at 25 fps.

Trade-offs. Considering the trade-offs between accuracy and runtime, the BIP approach is appropriate when accuracy is important and the scene has medium crowd density. The AL-ICM variant is more appropriate for time-critical applications or more crowded scenes, but comes with a cost in terms of accuracy. In the absence of heuristics, which are not useful when the scene is crowded or all appearances look the same, AL-ICM-NoGroup is the only method from the above set that can be used to meet weaker time constraints.

4.5 Implementation

We implemented our algorithm in MATLAB and we used the Gurobi Optimizer to solve the Binary Integer Programs. All experiments were done on a PC with Intel i7-3610 2.3 GHz processor and 6 GB of memory. The results for the BIP-NoGroup method in Table 2 were produced on a Linux machine with Intel Xeon E5540 2.53 GHz processor and 96 GB memory in order to solve very large Binary Integer Programs. The code and data to reproduce the above results are available on the authors' website.

5 Concluding Remarks and Future Work

We developed a general, efficient, online method for multiple person tracking that outperforms the state of the art and can be used with any person detector, appearance feature, space-time heuristic, or similarity metric. Our graph partitioning formulation accounts properly for evidence both for and against co-identity, and can both force and forbid co-identity through hard constraints. A cascade of stages that reason over short- and long-term time horizons exploits safe groupings by space-time and appearance opportunistically to reduce computational complexity. We improve over the state of the art and achieve real-time, online performance thanks to a sliding window approach. The windows can be made long enough to handle very significant occlusions successfully.

Future improvements include replacing the k-means algorithm for appearance grouping with clustering-forest techniques, which are more appropriate for online data association; replacing appearance and space-time affinities with more sophisticated metrics that depend on context; and making the temporal window length adapt to data complexity in real time. We are also looking at how additional constraints related to people entering and exiting the scene affect our algorithm. In the long term, we plan to extend our methods to tracking from multiple cameras.

Acknowledgements. This work was supported by the Army Research Office under Grant No. W911NF-10-1-0387 and by the National Science Foundation under Grants IIS-10-17017 and IIS-14-20894.

References

1. Shafique, K., Shah, M.: A noniterative greedy algorithm for multiframe point correspondence. In: PAMI. (2005)

2. Javed, O., Shafique, K., Rasheed, Z., Shah, M.: Modeling inter-camera space-time and appearance relationships for tracking across non-overlapping views. CVIU (2008)
3. Brendel, W., Amer, M., Todorovic, S.: Multiobject tracking as maximum weight independent set. In: CVPR. (2011)
4. Breitenstein, M.D., Reichlin, F., Leibe, B., Koller-Meier, E., Van Gool, L.: Robust tracking-by-detection using a detector confidence particle filter. In: ICCV. (2009)
5. Shu, G., Dehghan, A., Oreifej, O., Hand, E., Shah, M.: Part-based multiple-person tracking with partial occlusion handling. In: CVPR. (2012)
6. Henriques, J.F., Caseiro, R., Batista, J.: Globally optimal solution to multi-object tracking with merged measurements. In: ICCV. (2011)
7. Bae, S.H., Yoon, K.J.: Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. (2014)
8. Pirsiavash, H., Ramanan, D., Fowlkes, C.C.: Globally-optimal greedy algorithms for tracking a variable number of objects. In: CVPR. (2011)
9. Ben Shitrit, H., Berclaz, J., Fleuret, F., Fua, P.: Tracking multiple people under global appearance constraints. In: ICCV. (2011)
10. Berclaz, J., Fleuret, F., Turetken, E., Fua, P.: Multiple object tracking using k-shortest paths optimization. In: PAMI. (2011)
11. Izadinia, H., Saleemi, I., Li, W., Shah, M.: (mp)2t: Multiple people multiple parts tracker. In: ECCV. (2012)
12. Zhang, L., Li, Y., Nevatia, R.: Global data association for multi-object tracking using network flows. In: CVPR. (2008)
13. Amir Roshan Zamir, A.D., Shah, M.: Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In: ECCV. (2012)
14. Butt, A.A., Collins, R.T.: Multiple target tracking using frame triplets. In: ACCV. (2012)
15. Andriyenko, A., Schindler, K., Roth, S.: Discrete-continuous optimization for multi-target tracking. In: CVPR. (2012)
16. Arora, C., Globerson, A.: Higher order matching for consistent multiple target tracking. In: ICCV. (2013)
17. Yang, B., Nevatia, R.: An online learned crf model for multi-target tracking. In: CVPR. (2012)
18. Wen, L., Li, W., Yan, J., Lei, Z., Yi, D., Li, S.Z.: (Multiple target tracking based on undirected hierarchical relation hypergraph)
19. Andriyenko, A., Schindler, K.: Multi-target tracking by continuous energy minimization. In: CVPR. (2011)
20. Benfold, B., Reid, I.: Stable multi-target tracking in real-time surveillance video. In: CVPR. (2011)
21. Jiang, H., Fels, S., Little, J.J.: A linear programming approach for multiple object tracking. In: CVPR. (2007)
22. Leibe, B., Schindler, K., Van Gool, L.: Coupled detection and trajectory estimation for multi-object tracking. In: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, IEEE (2007) 1–8
23. Butt, A.A., Collins, R.T.: Multi-target tracking by lagrangian relaxation to min-cost network flow. In: CVPR. (2013)
24. Bansal, N., Blum, A., Chawla, S.: Correlation clustering. In: Foundations of Computer Science. (2002)
25. Tan, J.: A note on the inapproximability of correlation clustering. (2008)
26. Swamy, C.: Correlation clustering: maximizing agreements via semidefinite programming. In: ACM-SIAM symposium on Discrete algorithms. (2004)

27. Rubner, Y., Tomasi, C., Guibas, L.J.: A metric for distributions with applications to image databases. In: ICCV. (1998)
28. Anderberg, M.R.: Cluster analysis for applications. Technical report, DTIC Document (1973)
29. Liu, C., Gong, S., Loy, C.C., Lin, X.: Person re-identification: What features are important? In: ECCV Workshops. (2012)
30. Ferryman, J.: Proceedings (pets 2009). In: Eleventh IEEE International Workshop on Performance Evaluation of Tracking and Surveillance. (2009)
31. Keni, B., Rainer, S.: Evaluating multiple object tracking performance: the clear mot metrics. EURASIP JIVP (2008)
32. Leal-Taixé, L., Pons-Moll, G., Rosenhahn, B.: Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. In: ICCV Workshops. (2011)
33. McLaughlin, N., del Rincon, J.M., Miller, P. In: Online Multiperson Tracking With Occlusion Reasoning and Unsupervised Track Motion Model. (2013)
34. Bagdanov, A.D., Del Bimbo, A., Dini, F., Lisanti, G., Masi, I.: Posterity logging of imagery for video surveillance. (2012)
35. Bagon, S., Galun, M.: Large scale correlation clustering optimization. arXiv preprint arXiv:1112.2903 (2011)
36. Besag, J.: On the statistical analysis of dirty pictures. Journal of the Royal Statistical Society. Series B (Methodological) (1986) 259–302