On Pairwise Costs for Network Flow Multi-Object Tracking

Visesh Chari* Simon Lacoste-Julien† Ivan Laptev* Josef Sivic*

INRIA and Ecole Normale Supérieure, Paris, France

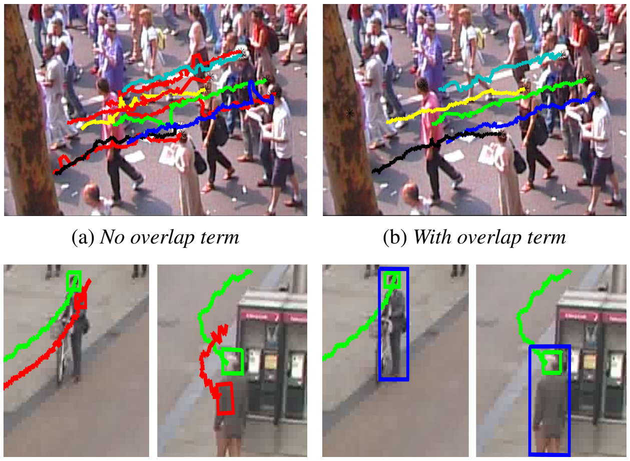
Abstract

Multi-object tracking has been recently approached with the min-cost network flow optimization techniques. Such methods simultaneously resolve multiple object tracks in a video and enable modeling of dependencies among tracks. Min-cost network flow methods also fit well within the "tracking-by-detection" paradigm where object trajectories are obtained by connecting per-frame outputs of an object detector. Object detectors, however, often fail due to occlusions and clutter in the video. To cope with such situations, we propose to add pairwise costs to the min-cost network flow framework. While integer solutions to such a problem become NP-hard, we design a convex relaxation solution with an efficient rounding heuristic which empirically gives certificates of small suboptimality. We evaluate two particular types of pairwise costs and demonstrate improvements over recent tracking methods in real-world video sequences.

1. Introduction

The task of visual multi-object tracking is to recover spatio-temporal trajectories for a number of objects in a video sequence. Tracking multiple objects, like people or vehicles, has a wide range of applications from Robotics to video surveillance [28]. Despite recent progress in the field [3, 5, 8, 20, 21, 22, 27], tracking remains a challenging problem especially in crowded and cluttered scenes.

With the advances in object detection, "tracking-by-detection" have recently become a popular paradigm for object tracking [5, 8, 13, 17]. Given object detections in every frame of a video sequence, the tracking is formulated as selection and clustering of corresponding object detections over time. Such selection and clustering problems can be solved in an optimization framework using carefully designed cost functions. Given an appropriate cost function, tracking-by-detection is typically setup as a MAP estimation problem [29]. Among different formulations of this problem, min-cost network flow [2] is particularly attractive as it allows for optimal and efficient solutions [22].


(a) *No overlap term* (b) *With overlap term*

(c) *No co-occurrence term* (d) *With co-occurrence term*

Figure 1: Results of network flow tracking using cost functions with/without pairwise terms. (a)-(b): a pairwise term that penalizes the overlap between different tracks helps resolving ambiguous tracks (shown in red) in crowded scenes. (c)-(d): a pairwise term that encourages the consistency between two signals (here head detections and body detections) helps eliminating failures (shown in red) of object detectors.

The energy minimization approach to tracking enables global solutions to track selection and avoids early and error-prone local decisions. Moreover, it also enables for a principled modeling of interactions among different tracks. In the past, models of track interactions have been shown to improve human tracking in crowds [21], to identify unusual behavior [15] as well as to resolve ambiguous tracks [20, 22]. Such previous methods, however, either resort to local *non-convex* optimization [21, 15, 20], or use greedy methods to enforce interactions [22].

Unlike previous work, we here propose to model track interactions within the min-cost network flow tracking approach. We introduce pairwise costs to the objective function and design a *convex* relaxation solution with an efficient rounding heuristic. Although our final integer solution can be suboptimal, our method is generic and empirically pro-

[*] WILLOW project-team, Départment d'Informatique de l'Ecole Normale Supérieure, ENS/INRIA/CNRS UMR 8548, Paris, France

[†] SIERRA project-team, Départment d'Informatique de l'Ecole Normale Supérieure, ENS/INRIA/CNRS UMR 8548, Paris, France

978-1-4673-6964-0/15/$31.00 ©2015 IEEE

vides certificates of small suboptimality. Tracking results using two particular examples of pairwise costs discussed in this paper are illustrated in Figure 1.

In summary, this paper makes the following contributions:

- We propose a new *non-greedy* approach to optimize pairwise terms within a min-cost network flow framework. Our solution is generic and allows the simultaneous optimization of any type of pairwise costs.
- We propose a global optimization strategy with a convex relaxation that allows us to minimize pairwise costs using linear optimization, and a principled Frank-Wolfe style rounding procedure to obtain integer solutions with a certificate of suboptimality. The optimization procedure is empirically stable, allowing the practitioner to focus on modeling.
- To illustrate our method, we propose two particular examples of pairwise costs: the first discourages significant overlaps between distinct tracks; the second models the spatial co-occurrence of different types of detections. This allows us to better model complex dynamic scenes with substantial clutter and partial occlusions.
- Using our method, we show improved tracking results on several real-world videos. In addition, we propose a new strategy to evaluate tracking results that better measures the longevity of overlap between output tracks and ground truth.

This paper is organized as follows. Section 2 presents related work and the overview of our approach. Section 3 summarizes min-cost flow tracking. Section 4 describes our optimization framework with pairwise costs presented in Sections 4.1.1 and 4.1.2. The optimization strategy is described in Section 5, with initial quadratic optimization formulation in Section 5.1 and subsequent linear relaxation in Section 5.3. Finally we present results of our method and compare them to the state of the art on challenging datasets in Section 6, and conclude with a discussion in Section 7.

2. Related work

Recent approaches have formulated multi-frame, multi-object tracking as a min-cost network flow optimization problem [29, 22, 5], where the optimal flow in a connected graph of detections encodes the selected tracks. While earlier min-cost network flow optimization methods have used linear programming, recently proposed solutions to the min-cost flow optimization include push-relabel methods [29], successive shortest paths [22, 5], and dynamic programming [22]. To ensure globally optimal and efficient solutions, previous methods have often restricted the cost to unary terms over all edges. While non-unary terms break the optimality of solutions in general, dependencies between detections have been enforced by greedy approaches, such as greedily eliminating the overlapping detections after each step of a sequential selection of distinct tracks in [22]. This non-global optimization approach, however, cannot recover from early suboptimal decisions.

Additional dependencies among detections can also be incorporated into the min-cost network flow tracking by modifying the underlying graph structure. Butt and Collins [8] follows this approach and minimizes the modified objective using Lagrangian methods. While the method works well for the particular type of introduced cost, generalizing this method to the new types of pairwise costs would require appropriate modifications of the graph structure which is non-trivial in general. Moreover, combining multiple costs within such a framework would be difficult. In contrast, our framework allows addition of terms without any modification to the underlying optimization framework.

Brendel et al. [7] and Milan et al. [20, 19] formulate the problem in a framework that first selects *tracklets* and then connects them using a learned distance measure [7] or a CRF [20, 19]. Long term occlusions are handled in [7] by merging appearance and motion similarity. While [20, 19] propose to alternate between discrete and continuous optimizations in order to minimize several cost functions, the presence of two levels of optimization makes theoretical or empirical guarantees of optimality hard to give. Unlike this work, we use a convex relaxation in our approach that allows us to give an empirical guarantee of optimality to our solutions.

Other methods [17, 26, 27] use offline or online training to learn a similarity measure between tracklets. These methods do not provide any optimality guarantee, though. In addition, training might be difficult in some conditions. For example, online training to discriminate appearances might be erroneous when objects move very close to each other (Figure 1). We avoid such problems by using pairwise terms to robustify the tracker to detection errors.

Incorporation of pairwise terms into the min-cost network flow formulation has been previously attempted by Choi and Savarese [9]. Their work, however, is focused on jointly optimizing tracking and activity recognition. In contrast, we focus on tracking in particular, and propose a generic framework enabling inclusion of multiple types of pairwise costs and providing empirical measures of small suboptimality.

2.1. Overview of our approach

We propose an algorithm that incorporates quadratic pairwise costs into the traditional min-cost flow network. Unlike previous methods [5, 17], which either build on top of min-cost flow solutions [20] or change the network structure [8], we propose a modification to the standard optimization algorithm. Such quadratic costs can represent several useful properties like similar motion of people in a rally, co-occurrence of tracks for different parts of the same object instance and others.

While in such a case obtaining the global optimum is NP-hard [18], we outline an approach to obtain near optimum solutions, while we empirically verify its optimality. We present a linear relaxation to the quadratic term that is fast to optimize, followed by a Frank-Wolfe based rounding heuristic to obtain an integer solution.

3. Background: Min-cost flow tracking

In this section, we describe the traditional formulation of multi-object tracking as a min-cost flow optimization problem [29]. We extend this framework in Section 4.

Given a video with objects in motion, the goal is to simultaneously track K moving objects in a "detect-and-track" framework [29]. The input to the approach is two-fold. First a set of candidate object locations is assumed to be given, provided, for example, as output of an object detector. Henceforth we refer to these locations as *detections*. The approach also requires a measure of correspondence between detections across video frames. This could be obtained for example from optical flow, or using some other form of correspondence. Based on these inputs, the tracking problem is setup as a joint optimization problem of simultaneously selecting detections of objects and connections between them across video frames. Such a problem can be modeled through a MAP objective [29] with specific constraints encoding the structure of the tracks. The MAP optimization problem can be cast as the following integer linear program (ILP):

$$\min_{\mathbf{x}} \qquad \textstyle\sum_i c_i x_i + \sum_{ij\in E} c_{ij} x_{ij} \tag{1}$$

$$\text{s.t.} \quad \left.\begin{array}{c} 0 \le x_i \le 1\,,\, 0 \le x_{ij} \le 1 \\ \displaystyle\sum_{i\,:\,ij\in E} x_{ij} = x_j = \sum_{i\,:\,ji\in E} x_{ji} \\ \displaystyle\sum_i x_{it} = K = \sum_i x_{si} \end{array}\right\} \mathbf{x} \in \mathrm{FLOW}_K$$

$$x_i\,,\, x_{ij} \quad \text{are} \quad \text{integer}.$$

The above formulation encodes the joint selection of K tracks using the following selection variables: $x_i \in \{0,1\}$ is a binary indicator variable taking the value 1 when the *detection* i is selected in some track; $x_{ij} \in \{0,1\}$ is a binary indicator variable taking the value 1 when detection i and detection j are *connected* through the *same* track in nearby time frames. The index i ranges over possible detections across the whole video. c_i denotes the cost of selecting detection i in a specific frame (and represents the negative detection confidence) while c_{ij} represents the negative of the correspondence strength between detections i and j. The set of possible connections between detections is represented by E and could be a subset of all pairs of detections in nearby frames by using choice heuristics (such as spatial proximity). The quality of track selection is quantified by the objective in (1).

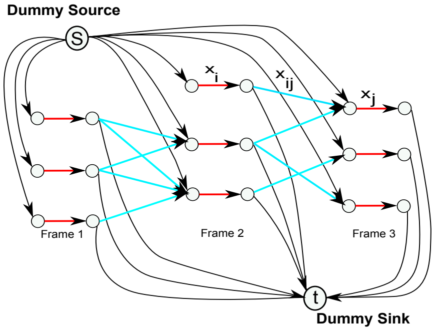


Figure 2: Illustration of a graph used in traditional min-cost flow. The detection (red) and connection (cyan) variables are marked as edges and have unit capacity. Every track is a unit flow that starts at a source S and ends at a sink t. S and t are connected to all detections.

The constraint $\sum_{i\,:\,ij\in E} x_{ij} = x_j = \sum_{i\,:\,ji\in E} x_{ji}$, which has the structure of a *flow conservation constraint* [2], encodes the correct claimed semantic that x_{ij} can take the value 1 if and only if both x_i and x_j take the value 1, and moreover, that each detection belongs to *at most one track*, enforcing the fact that two objects cannot occupy the same space. Finally, the constraint $\sum_i x_{it} = K = \sum_i x_{si}$ ensures that exactly K tracks are selected (dummy "source" and "sink" variables with the fixed value $x_s = x_t = K$ are added; the connection variables x_{si} and x_{it} represent the start and end of tracks respectively).

We have grouped the linear constraints in (1) under the name FLOW_K as they actually correspond to constraints in a min-cost network flow problem where one would like to push K units of flow with minimum cost in a network with unit capacity edges. In fact, these linear constraints have the property of being *totally unimodular* [2]. This implies that the polytope they determine has only vertices with *integer* coordinates, and so relaxing the integer constraints in (1) and solving it as a linear program is still guaranteed to produce *integer solutions*, making it a tight relaxation. Figure 2 illustrates the correspondence between a network flow structure and the formulation (1).

To summarize, the above optimization problem with relaxed integer constraint can be solved efficiently using existing network flow or linear algebra packages [2], and provides a convenient framework to transform the tracking problem into a *track selection* problem. We use this conversion as a starting point to add additional constraints and costs on the selection process to influence it in desirable ways to address challenging scenarios that are shown in later sections.

4. Modeling pairwise costs with an IQP

The above formulation in (1) represents a linear objective with linear equality constraints (where the integer constraint is not needed). While linear terms are both simple and easy to minimize, higher order models can represent more useful properties [21]. We suggest to add a quadratic cost between pairs of selection variables. To simplify the notation for the optimization sections, we collect the x_i and x_{ij} variables in a long vector $\mathbf{z}$. The product $z_i z_j$ then encodes *joint* selection of z_i and z_j – these choices could correspond to a pair of connections, a pair of detections, or even a connection and a detection. A term of the form $Q_{ij} z_i z_j$ can then either encourage (or discourage) the joint selection of z_i and z_j by having Q_{ij} negative (or positive), respectively. Our approach is to consider a small set $\mathcal{Q}$ of such joint selections, and add the term $\sum_{ij \in \mathcal{Q}} z_i z_j Q_{ij}$ to the objective. Our new optimization problem can thus be expressed as the integer quadratic program (IQP):

$$\begin{aligned} \min_{\mathbf{z}} \quad & \mathbf{c}^\top \mathbf{z} + \mathbf{z}^\top \mathbf{Q} \mathbf{z} \\ \text{s.t.} \quad & \mathbf{z} \in \text{FLOW}_K \\ & \mathbf{z} \quad \text{integer}\,, \end{aligned} \tag{2}$$

where the $\mathbf{Q}$ matrix is *sparse* with $Q_{ij} \neq 0$ for $ij \in \mathcal{Q}$.

Unfortunately, the above formulation can encode the quadratic assignment problem which is NP-hard to optimize in general [18]. Nevertheless, we propose an efficient (convex) linear relaxation in Section 5 as well as a powerful rounding heuristic that provides empirical certificates of suboptimality. Our main modeling strategy is thus twofold: first, we encode our prior knowledge about the joint selection of variables using the sparse cost matrix $\mathbf{Q}$ (which can be arbitrary); second we add additional constraints to the IQP as long as they can be encoded as network flow constraints (this is a requirement of our rounding heuristic presented in Section 5.3). In the rest of this section, we provide two examples of pairwise costs used in our experiments. We then focus on the optimization aspects in Section 5.

4.1. Designing pairwise costs

In the following subsections, we show how some traditional constraints [15, 21] could be incorporated in our quadratic min-cost network flow framework. We focus on elements that cannot be simply encoded with traditional linear terms in (1).

4.1.1 Overlap penalty

Object detectors often produce multiple responses per object. This issue is typically addressed by the *Non Maxima Suppression* (NMS) step, which retains most confident detections within spatial neighborhoods. While NMS works well for tracking isolated objects, independent decisions produced by NMS for each object and frame often become suboptimal in crowded scenes where multiple objects may occupy the same spatial neighborhood. To address this problem, we avoid taking independent decisions and propose to discourage overlapping detections within the network flow tracking framework. For this purpose we extend the cost function with the following *pairwise overlap cost*:

$$\begin{gathered} q_{ij}^{\text{ov}} x_i x_j \\ \text{for } \; (i,j) \;\text{ s.t. }\; \text{ov}(\text{box}(x_i), \text{box}(x_j)) \geq o_{\text{thres}} \end{gathered} \tag{3}$$

where x_i and x_j represent two selection variables associated with sufficiently overlapping[1] detections and $q_{ij}^{\text{ov}} > 0$.

In previous approaches like [22], NMS was implemented in a *greedy* fashion. Greedy approaches, however, have the disadvantage of making non-reversible decisions in the early stages of optimization. In contrast, our approach of incorporating the cost (3) into the overall cost function ensures that NMS is optimized *simultaneously* with other tracking objectives. As a result, overlapping detections may become tolerated, for example, in situations when two tracks intersect. On the other hand, continuously overlapping tracks resulting from multiple outputs of detectors will be discouraged.

4.1.2 Enforcing consistency between two signals

In many tracking scenarios, multiple signals are available for use. For example, we might have a body detector as well as a head detector. In case they give complementary information about the presence of the object, we can be more robust to detection noise by ensuring that the two tracks are consistent using a pairwise cost.

For example, let z_i^h and z_i^b denote the selection variables (detection or connection) for the head and body respectively. Each set can be associated with its own flow feasible set FLOW_K^h and FLOW_K^b. We can *encourage* the consistent "co-occurrence" of the two flows by adding the following negative cost:

$$\begin{gathered} -q_{ij}^{\text{co}} z_i^h z_j^b \\ \text{for } (i,j) \text{ s.t. } z_i^h \text{ and } z_j^b \text{ are } \textit{consistent}. \end{gathered} \tag{4}$$

In our experiment, we say that z_i^h and z_j^b are *consistent* in two scenarios. Either z_i^h and z_j^b are detection variables such that their corresponding boxes[2] overlap more than o_{thres}. Or we have a head detection z_i^h with a box that intersects the edge z_j^b connecting its respective body detection boxes (and similarly for a body detection and head edge). The idea behind the latter possibility is to be more robust to missing detections on some frames: it corresponds to a situation where a head and body detection would have overlapped if we were interpolating detections along an edge that skips frames. Note that the cost (4) is difficult to minimize greedily, since both head and body tracks need to be optimized *simultaneously*.

[1] The overlap threshold o_{thres} is set to 0.5 in our experiments.

[2] For the body detection box, we only consider its top 25% region when computing overlap or looking at intersection.

5. Optimization

In the previous section, we presented examples of quadratic cost functions that we could include in our extension to the min-cost flow network formulation to encourage co-occurrence preferences for individual variables in the minimization. Finding a global minimum is NP-hard [18] if we keep the integer constraints on the variables (which is necessary to ensure the correct track encoding). Our suggested strategy is to instead find a global solution to the *relaxed* version of the problem with the integer constraints removed, and then use a powerful heuristic to search for nearby integer solution that satisfies the flow constraints (see Section 5.3). By comparing the objective value between the "rounded" integer solution and the global solution to the relaxed problem, which provides a lower bound, we obtain a *certificate of optimality*. In our experiments, we observed that suboptimality upper bounds were quite small, thus indicating that our optimization framework is stable and we can instead focus on designing good cost functions. We now describe several approaches to optimize (2).

5.1. Quadratic optimization

If $\mathbf{Q}$ is positive definite, then the quadratic program (QP) in (2) with relaxed integer constraint is convex and can be robustly optimized using interior point methods implemented in commercial solvers such as MOSEK/CPLEX. These methods can scale to medium-size problems[3] by exploiting the sparseness of $\mathbf{Q}$ suggested in Section 4.1.

In our general formulation, $\mathbf{Q}$ is not necessarily positive definite. We can nevertheless use a standard trick to make it positive definite by defining its diagonal entries to be $Q_{ii}^{\text{new}} = \sum_{j \neq i} |Q_{ij}^{\text{old}}|$, while using $c_i - Q_{ii}^{\text{new}} + Q_{ii}^{\text{old}}$ as the linear coefficient for z_i in the objective. As $z_i^2 = z_i$ for binary variables, this transformation sill yields an (equivalent) IQP. $\mathbf{Q}^{\text{new}}$ is now positive semidefinite [11, Thm. 6.1.10], and so the relaxation gives a convex problem.

In order to scale to very large scale datasets (billions of variables), one could use the Frank-Wolfe algorithm [12] which is a first order gradient based method that iteratively minimizes a linearization of the quadratic objective. An advantage of this approach is that each step of the Frank-Wolfe algorithm reduces in our case to the minimization of a min-cost network flow problem, which can scale to much larger sizes than a generic linear program solver. Moreover, each step of this algorithm yields an integer solution. Thus, while optimizing the relaxed objective (which will provide a lower bound certificate), we can keep track of which integer iterate had the best objective thus far. This perspective also motivates a powerful rounding heuristic that we describe in Section 5.3. Building on a preliminary version of our paper, [14] used this approach successfully for performing efficient co-localization in videos, where the constraint set also had a network flow structure.

5.2. Equivalent integer linear program

Another way to make the approach more scalable is to transform the integer QP (2) into an equivalent integer linear program (ILP) by introducing well-chosen additional variables and constraints. We present such an approach in this section, which generalizes the line of reasoning from [16].

We introduce a new set of variables u_{ij} that encode the joint selection of the edge z_i and z_j, and thus we would like to enforce $u_{ij} = z_i z_j$. The quadratic cost component $Q_{ij} z_i z_j$ could then be replaced with a linear cost $Q_{ij} u_{ij}$. An equivalent integer linear program is thus the following:

$$
\begin{array}{lcc}
\min\limits_{\mathbf{z},\mathbf{u}} & \mathbf{c}^\top \mathbf{z} + \mathbf{q}^\top \mathbf{u} & \\
 & \mathbf{z} \in \mathrm{FLOW}_K & \\
\text{s.t.} & \left.\begin{array}{c} 0 \le u_{ij} \le 1,\ \forall ij \in \mathcal{Q} \\ u_{ij} \le z_i\,,\ u_{ij} \le z_j \\ z_i + z_j \le 1 + u_{ij} \end{array}\right\} & \begin{array}{c} (\mathbf{z},\mathbf{u}) \in \\ \mathrm{LOCAL}(\mathcal{Q}) \end{array} \\
 & \mathbf{z},\mathbf{u} \quad \text{integer}\,. &
\end{array}
\qquad (5)
$$

Here $\mathbf{u}$ and $\mathbf{q}$ represents the vector whose elements are u_{ij} and Q_{ij} respectively. The new constraint $z_i + z_j \le 1 + u_{ij}$ enforces that u_{ij} should be 1 if z_i and z_j are both 1; while the pair of constraints $u_{ij} \le z_i$ and $u_{ij} \le z_i$ enforce $u_{ij} = 0$ if either z_i or z_j is zero. We call these constraints '$\mathrm{LOCAL}(\mathcal{Q})$' as it turns out that they define a polytope which can be obtained by a projection of the *local marginal consistency* polytope for the over-complete representation of a discrete Markov random field (MRF) [24, (4.6)] with edges defined by the non-zero entries of $\mathbf{Q}$[4]. Removing the integer constraint in (5) thus yields a LP relaxation that is similar to one for MAP inference in MRFs, but with additional FLOW_K constraints, yielding a crucial structural difference with the previous works.

An advantage of this formulation is that its relaxed form is a LP, which can usually be optimized by MOSEK or CPLEX to larger scale than the QP formulation, even though there is an increase in the number of variables and constraints. Note though that the number of new variables u_{ij} created is the same as the number of non-zero coefficients in the sparse $\mathbf{Q}$, which was indicated by the set $\mathcal{Q}$ in (5) to stress that we do not need to look at all pairs of edges. In exploratory experiments, we observed that the LP relaxation yielded similar quality solutions as the QP relaxation, but was faster to optimize; we have thus focused on the LP relaxation in our experiments. Another advantage of (5) is that we can easily generalize it to handle higher

[3] A few millions variables, which translates to several hundreds frames with a high number of detections for our datasets.

[4] More specifically, this representation defines one indicator variable per possible joint assignment of values on the cliques of the MRF. If we do Fourier-Motzkin elimination [6, 24] on the local consistency polytope to eliminate the extra variables and to only keep the three variables z_i, z_j, u_{ij} for each edge, then we obtain back the constraints for $\mathrm{LOCAL}(\mathcal{Q})$.

order terms in the objective. For a clique C of decision variables that we want to encourage or discourage jointly, we introduce a new variable $u_C := \prod_{i \in C} z_i$. This semantic can be readily enforced with the constraints $u_C \leq z_i$ for all $i \in C$, and $\sum_{i \in C}(z_i - 1) + 1 \leq u_C$, which generalizes $\mathrm{LOCAL}(\mathcal{Q})$ for higher order terms and yields another ILP that can be relaxed to a LP.

5.3. Frank-Wolfe rounding heuristic

The solution of the LP relaxation of (5) can have fractional components because the additional linear constraints from $\mathrm{LOCAL}(\mathcal{Q})$ essentially violate the *total unimodularity* property, in contrast to FLOW_K which yields a polytope with only integer vertices. Since naively rounding the obtained fractional variables to the nearest integer might not result in a feasible point (in other words a valid flow), we need a strategy to obtain an integer solution with cost similar to the minimum. Given the relaxed global solution $\mathbf{z}^*$, the simplest approach would be to look for the point closest in Euclidean norm in FLOW_k which is an integer. As $z_i^2 = z_i$ for binary variables, we have $||\mathbf{z} - \mathbf{z}^*||^2 = (\mathbf{1} - 2\mathbf{z}^*)^\top \mathbf{z} + ||\mathbf{z}^*||^2$ which is a linear function of $\mathbf{z}$. We can thus obtain the closest integer point by solving a LP over FLOW_k, as all its vertices are integers. We call this approach *Hamming rounding* as $d_H(\mathbf{z}, \mathbf{z}') := ||\mathbf{z} - \mathbf{z}'||^2$ reduces to the Hamming distance when evaluated on pair of binary vectors. On the other hand, the closest point in Euclidean norm does not necessarily yield a good objective value (as the search was agnostic to the objective). Inspired by the Frank-Wolfe algorithm, our suggested heuristic is to minimize instead the first-order linear under-estimator of the quadratic objective constructed with the gradient at the relaxed global solution $\mathbf{z}^*$. Specifically, we obtain the following LP, which has the usual network flow constraint structure and thus can be solved very efficiently:

$$\begin{aligned} \min_{\mathbf{z}} \quad & \left(\mathbf{c} + (\mathbf{Q} + \mathbf{Q}^\top)\mathbf{z}^*\right)^\top \mathbf{z} \\ \text{s.t.} \quad & \mathbf{z} \in \mathrm{FLOW}_K \,. \end{aligned} \tag{6}$$

The objective here can be interpreted as modifying the distance function on binary vectors to take the cost function in consideration. As previously mentioned, the relaxed LP solution provides a lower bound on the true ILP (which is equivalent to the IQP) solution. The difference between the objective evaluated on *any* feasible integer solution and the lower bound is thus an upper bound certificate on its suboptimality. In our experiments, we obtained small suboptimality certificates ($\approx 10^{-3}$) for our returned integer solutions, indicating that our rounding heuristic was effective at returning near-global optimal solutions (we note that we define $\mathbf{c}$ and $\mathbf{Q}$ so that the objective is normalized between 0 and 1). We also observed that Hamming rounding generally produced a suboptimality that was around 3 to 4 times *worse* than the solution produced by Frank-Wolfe rounding. These worse objective values also translated in worse tracking accuracy (see Appendix A in the supplementary material[5]). We finally note that in contrast to the previous work [8] which could not guarantee that their algorithm would converge to an integer solution, our approach will always give *some* integer solution (by solving a simple min-cost network flow problem), and can provide a certificate of suboptimality a-posteriori.

6. Experiments

In this section, we evaluate our approach on several real world videos and compare results to the state-of-the-art methods [4, 20, 22]. First we illustrate the effect of the two pairwise costs proposed in Section 4.1 and evaluate improvement over the basic min-cost network flow tracking. We also argue that the standard MOTA score is often insufficient to capture the quality of tracking results and propose a new measure for tracking evaluation, termed *re-detection measure* (Section 6.2.1).

Second, we evaluate our method on six videos from the two standard datasets PETS and TUD. For both of these datasets, we obtain part of the input (person detections) from Milan et al. [20], and show improvements over their approach using the standard MOTA metrics.

6.1. Tracking datasets

We test our algorithm on several publicly available videos. The first video MarchingRally corresponds to a crowd walking in a rally along a street (see Figure 1, top row). The video consists of 120 frames recorded at 25 fps, and has about 50 people. This video is challenging due to the high number of people moving close to each other. We have manually annotated ground truth tracks for all people in this video for the purpose of tracking evaluation[6].

The second video illustrated in Figure 1 (bottom row) is called TownCenter [4] and consists of 4500 frames recorded at 25 fps. The video shows approximately 230 people walking across the street. Finally, we use videos from the well-known PETS and TUD datasets. These videos depict frequently occluded people moving in multiple directions.

Preprocessing. We run a "head" detector [23] to detect heads of people in every frame of the MarchingRally and PETS videos. While we use only head detections for the MarchingRally sequence, for PETS we use our head detections in combination with readily-available body detections from [20]. Head detections complement frequently overlapping body detections and help resolving partial occlusions as well as ID-switches. For each of these videos, we run a KLT tracker after initializing features within detection bounding boxes. Finally, for every pair of nearby frames (< 10 frames apart), we connect pairs of detections

[5] The supplementary material (with videos and code) is available at [1].

[6] The original MarchingRally video and the corresponding ground truth tracks are available from [1].

with high correspondence strength. The strength of correspondence between two detections is the ratio of their common KLT tracks and the total number of KLT tracks passing through both detections.

6.2. Tracking in video experiment

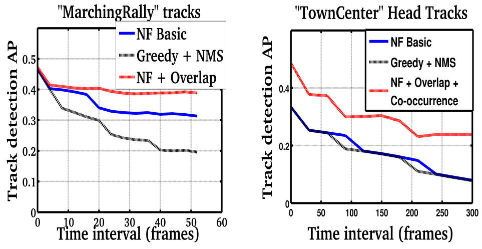
6.2.1 Evaluation strategy

Evaluating results of multi-object tracking is non-trivial because errors might be present in various forms including ID switches, broken tracks, imprecisely localized tracks and false tracks. Measures such as MOTA [4, 20] combine different errors into a single score and enable the global ranking of tracking methods. Such measures, however, lack interpretability. On the other hand, independent assessment of different errors can also be misleading. For example, in dense crowd videos such as in Figure 1(a), tracks may have relatively low localization error while being incorrect due to switches between neighboring people. Similarly, low error of ID switches can be a consequence of many broken tracks.

We argue that a meaningful evaluation of tracking methods should be related to a task. One task with particular relevance to crowd videos is to detect the location of a given person after Δt frames. To evaluate the performance of tracking methods on such a task we propose the re-detection measure as described below.

Re-detection measure. The proposed re-detection measure evaluates the ability of a tracker to find the correct location of a given object after Δt frames. The measure is inspired by the common evaluation procedure for object detection in still images [10] and extends it to tracking. For each pair of detections A_t and $B_{t+\Delta t}$ associated to the same track by a tracker, we check if there exists a ground truth track that overlaps with A_t and $B_{t+\Delta t}$ on frames t and $t + \Delta t$ respectively.[7] If the answer is negative, the subtrack $(A_t, B_{t+\Delta t})$ is labeled as false positive. Otherwise, it is labeled as true positive unless there exist multiple subtracks overlapping with the same ground truth. To avoid multiple responses, in the latter case only one subtrack is labeled as true positive while others are declared as false positives.

For the given Δt we collect subtracks from all video intervals $(t, t + \Delta t)$ and sort them according to their confidence.[8] Given the subtrack labels defined above, we evaluate Precision-Recall and Average Precision (AP). High AP values indicate the good performance of the tracker in the re-detection task. On the other hand, common errors such as ID switches and imprecise localization reduce AP values. Note that in the case of $\Delta t = 0$, our measure reduces to the standard measure for object detection. Larger values of Δt enable evaluation of re-detection for longer time intervals.



(a) (b)

	MOTA	Prec	Recall
NF + Ov. + Co-oc.	**55.9%**	**93.1%**	60.6%
Ben[4] Head	45.4%	73.8%	**71.0%**

Figure 3: (Top) Re-detection results for including overlap and co-occurrence terms in the linear relaxed formulation (5), vs state of the art. The maximum Δt considered on the x-axis corresponds to the median length of tracks in the video. (a) & (b) Show performance for the MarchingRally and TownCenter sequences respectively. In (a), the overlap term significantly improves over NF basic. Interestingly, NF basic outperforms the greedy baseline as opposed to the claim in [22] where the difference between the two methods was measured only at $\Delta t = 0$ and thus was not visible. In (b), adding a co-occurrence term to the network flow formulation also provides a significant improvement over the baselines. (Bottom) Tracking results for TownCenter evaluated in terms of MOTA measures, compared with results of [4].

To compare different methods, we plot the re-detection AP for different values of Δt as illustrated in Figure 3.

6.2.2 Experimental results

We compare our algorithm with the state-of-the-art approaches on several video sequences. For the MarchingRally and TownCenter sequences, the baseline approaches for comparison are a greedy implementation of the basic min-cost network flow algorithm with the greedy NMS heuristic from [22], and a network flow (NF) implementation as a linear program. In all graphs in Figure 3, the corresponding results are represented by black ("Greedy + NMS") and blue ("NF Basic") curves. We note that we perform a careful grid search over the parameter space for all three algorithms and show the results corresponding to the best parameters, to make sure the differences observed are not arising from different parameter choices, but rather from limitations of the framework. On the other hand, we have used only one fixed set of parameter values to produce the results on the different sequences in the PETS and TUD datasets given in Table 1. See [1] for the parameters used and information about the runtime.

In the MarchingRally video sequence, several people are moving in a crowd in a similar direction. The angle of viewing and the number of people alleviate the issue of clutter, which leads to failure of tracking algorithms that tend to confuse tracking identities. Our algorithm with overlap con-

[7] The overlap between ground truth and detections is measured by the standard Jaccard similarity of corresponding bounding boxes.

[8] The confidence for a subtract in this paper is given by the sum of its constituent detection confidences and correspondence strengths.

		Rcll	Prcn	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP
TUD Stadtmitte	NF	67.9	72.0	10	4	6	0	305	371	26	26	39.3	59.5
	NF+pairwise	59.6	**89.9**	10	2	8	0	**77**	467	15	22	51.6	**61.6**
	Milan [20]	**69.1**	85.6	10	**4**	**6**	0	134	**457**	15	**13**	**56.2**	**61.6**
PETS S2L1	NF	93.7	83.4	19	17	2	0	870	293	64	66	73.6	72.9
	NF+pairwise	92.4	**94.3**	19	**18**	**1**	0	**262**	354	56	74	85.5	**76.2**
	Milan [20]	**96.8**	94.1	19	18	**1**	0	282	**148**	**22**	**15**	**90.3**	74.3
PETS S2L2	NF	47.7	87.6	43	1	37	5	693	5383	291	531	38.1	60.7
	NF+pairwise	60.6	88.6	43	6	**34**	3	807	4050	244	379	50.4	**60.6**
	Milan [20]	**65.1**	**92.4**	43	**11**	31	**1**	**549**	**3592**	**167**	**153**	**58.1**	59.8
PETS S2L3	NF	44.5	92.2	44	9	15	20	164	2428	121	189	38.0	69.3
	NF+pairwise	**45.5**	91.2	44	**12**	15	**17**	155	**2125**	44	50	**40.3**	61.2
	Milan [20]	43.0	**94.2**	44	8	**17**	19	**115**	2493	**27**	**22**	39.8	**65.0**
PETS S1L1-2	NF	62.9	89.1	44	18	15	11	295	1425	289	140	47.8	65.2
	NF+pairwise	**68.9**	92.0	44	20	**16**	**8**	230	**1198**	35	74	**62.0**	**62.1**
	Milan [20]	64.9	**92.4**	44	**21**	12	11	**169**	1349	**22**	**19**	60.0	61.9
PETS S1L2-1	NF	31.3	87.4	42	4	15	23	208	3501	101	243	23.7	57.9
	NF+pairwise	**37.9**	89.6	42	**4**	**20**	**18**	223	**3141**	67	122	**32.2**	55.0
	Milan [20]	30.9	**98.3**	42	2	19	21	**27**	3494	**42**	**34**	29.6	**58.8**

Table 1: Table summarizing results over PETS and TUD sequences. Bold indicates best value for each column for each dataset. Abbreviations are as follows GT - ground truth tracks. MT - Mostly tracked. PT - partially tracked. ML - mostly lost. FP - false positives. FN - false negatives. IDs - ID swaps. FM - fragmentation.

straints (red curve) outperforms the state of the art by a large margin. Figure 3(a) shows the re-detection accuracy results with/without the overlap constraints. Note that the difference in performance between our algorithm and [22] grows together with the re-detection time interval. In fact, for the intervals of 40 frames or more, our algorithm outperforms the baseline by over 20% AP.

The TownCenter sequence is a video with two complementary sets of detections corresponding to heads and upper bodies. While head detections are noisy but have high recall, body detections are more precise but are also prone to more clutter. In such a case, as shown in Figure 3(b) we leverage body detections to improve noisy head tracks. Again in this case, there is more than 20% improvement in AP over the head baseline. Finally, the table in Figure (3) compares our method with a state-of-the-art [4] algorithm in terms of traditional MOTA evaluation measure. Note that while we compare with a "greedy" version of the overlap term [22], designing a greedy version of the co-occurrence term is not obvious.

For the PETS and TUD sequence, we compare the results of our method based on MOTA metrics with those presented in Milan et al. [20]. These sequences are challenging for a variety of reasons. First, there is a crowd of people walking in different directions and criss-crossing each other, which makes sustained tracking difficult. Second, few full body detections are available per frame in each video, which makes adding new terms to the objective function difficult. Third, since people walk side-by-side there is a lot of overlap between detections that belong to two different persons, hence enforcing the overlap criterion is difficult. However, as can be seen in Table 1, our method generally has comparable MOTA, MOTP and recall scores with [20]. This shows that our method is able to address complex scenarios effectively and our cost function is easy to adapt to general scenarios. Note also that the camera angle in PETS and TUDS are very different from each other, which means that our algorithm is sufficiently robust to these changes. Thus, we estimate trajectories better (sum of MT and PT of our method is usually high). This also results from the use of both overlap and co-occurrence terms in our approach, which can take into account head detections as additional information.

7. Discussion and conclusion

We have presented a generic optimization procedure enabling addition of quadratic costs to the min-cost network flow tracking methods. Our method enables modeling of track interactions in a principled way and provides empirical certificates of small suboptimality. We have shown practical benefits of our method for two particular examples of pairwise costs on challenging video sequences.

Combining different types of pairwise costs into a single (linear) cost opens up the possibility of tracking complicated motions. Moreover, while complex cost functions have more tunable parameters, they could be learnt from labeled data using structured output learning [16]. This opens up the possibility of learning quadratic costs for specific *crowd actions* such as panic, street crossing or stampede.

8. Acknowledgements

This research was supported in part by the projects FluidTracks, EIT ICT Labs, Google research award, ERC grant Activia (no. 307574) and ERC grant LEAP (no. 336845). We thank Patrick Perez for discussions on the multi-target tracking evaluation.

References

[1] http://www.di.ens.fr/willow/research/flowtrack/. 6, 7, 10

[2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice-Hall, Inc., 1993. 1, 3

[3] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *CVPR*, 2008. 1

[4] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. In *CVPR*, 2011. 6, 7, 8

[5] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *TPAMI*, 33(9):1806–1819, 2011. 1, 2

[6] D. Bertsimas and J. N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997. 5

[7] W. Brendel, M. Amer, and S. Todorovic. Multiobject tracking as maximum weight independent set. In *CVPR*, 2011. 2

[8] A. A. Butt and R. T. Collins. Multi-target tracking by Lagrangian relaxation to min-cost network flow. In *CVPR*, 2013. 1, 2, 6

[9] W. Choi and S. Savarese. A unified framework for multi-target tracking and collective activity recognition. In *ECCV*, 2012. 2

[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Pascal visual object classes (VOC) challenge. *IJCV*, 2010. 7

[11] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985. 5

[12] M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *ICML*, 2013. 5

[13] H. Jiang, S. Fels, and J. Little. A linear programming approach for multiple object tracking. In *CVPR*, 2007. 1

[14] A. Joulin, K. Tang, and L. Fei-Fei. Efficient image and video co-localization with Frank-Wolfe algorithm. In *ECCV*, 2014. 5

[15] L. Kratz and K. Nishino. Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In *CVPR*, 2009. 1, 4

[16] S. Lacoste-Julien, B. Taskar, D. Klein, and M. I. Jordan. Word alignment via quadratic assignment. In *NAACL*, 2006. 5, 8

[17] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *CVPR*, 2009. 1, 2

[18] E. M. Loiola, N. M. M. de Abreu, P. O. Boaventura-Netto, P. Hahn, and T. Querido. A survey for the quadratic assignment problem. *European Journal of Operational Research*, 176(2), 2007. 3, 4, 5

[19] A. Milan, S. Roth, and K. Schindler. Continuous energy minimization for multi-target tracking. *TPAMI*, 36(1):58–72, 2014. 2

[20] A. Milan, K. Schindler, and S. Roth. Detection- and trajectory-level exclusion in multiple object tracking. In *CVPR*, 2013. 1, 2, 6, 7, 8

[21] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *CVPR*, 2009. 1, 4

[22] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, 2011. 1, 2, 4, 6, 7, 8, 10, 12

[23] M. Rodriguez, I. Laptev, J. Sivic, and J.-Y. Audibert. Density-aware person detection and tracking in crowds. In *ICCV*, 2011. 6

[24] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008. 5

[25] B. Wu and R. Nevatia. Tracking of multiple, partially occluded humans based on static body part detection. In *CVPR*, 2006.

[26] B. Yang, C. Huang, and R. Nevatia. Learning affinities and dependencies for multi-target tracking using a CRF model. In *CVPR*, 2011. 2

[27] B. Yang and R. Nevatia. An online learned CRF model for multi-target tracking. In *CVPR*, 2012. 1, 2

[28] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4), 2006. 1

[29] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*, 2008. 1, 2, 3

Supplementary Material

A. Superiority of Frank-Wolfe rounding heuristic vs. Hamming rounding

In Section 5.3, we described two approaches to round the fractional solution $\mathbf{z}^*$ obtained after optimizing the LP relaxation (5). "Rounding" here meant finding a valid track encoding for prediction, i.e. a $\mathbf{z} \in \mathrm{FLOW}_k$ with integer coordinates. The first approach was to find the vertex (binary vector) in FLOW_k with minimal Euclidean distance to $\mathbf{z}^*$. We called this approach *Hamming rounding* and is standard for problems operating on binary vectors. We also proposed a novel alternative rounding heuristic called *Frank-Wolfe rounding* which instead minimizes the linear approximation of the quadratic objective (2), and is given by problem (6). In our experiments, we observed that Frank-Wolfe rounding yielded solutions with better objective values, as well as better tracking accuracy, than Hamming rounding. We illustrate these observations in this section.

For the MarchingRally experiment (where we only have head detections), we parameterized the objective with two parameters: a multiplicative constant in front of the detection confidences, and the value of the overlap penalty q_{ij}^{ov} mentioned in (3) (set to a constant).[9] In Table 2, we compare the suboptimality certificate values for Frank-Wolfe rounding vs. Hamming rounding for 6 different parameter settings on the MarchingRally dataset. More specifically, for each parameter setting, we first obtain the global relaxed solution $\mathbf{z}^*$ to the LP relaxation (5), then we either round by Frank-Wolfe rounding or by Hamming rounding and compare their suboptimality certificates. We also compare their re-detection accuracy in Figure 4, which shows that Frank-Wolfe rounding systematically yields better results than Hamming rounding.

Case 1 in Table 2 is the reference case where we use the best parameter values found by grid search, which were used to produce the results in Figure 3(a) in the paper. For Case 2 and 3, we vary the overlap penalty weight. Case 2 is a very low value for the overlap term encouraging tracks to criss-cross each other, while Case 3 has a very high overlap weight which means even small amount of overlap is unacceptable. Results for these cases are shown in the first row of Figure 4. The next three cases vary the weight for detection confidence. In particular in Case 6, the presence of negative weight actually "discourages" any detections from being picked unless they are connected to edges with extremely high connection strength. This results in poor performance as shown in Figure 4 but note that even here, Hamming rounding results are worse than the Frank-Wolfe rounding ones. Also note that worse suboptimality certificates usually result in worse tracking.

	Detection	Overlap	FW	Ham.
Case1	0.1	0.0223	4.7e-03	1.4e-02
Case2	0.1	0.0007	8.7e-06	9.3e-03
Case3	0.1	2.23	4.3e-03	1.0e-01
Case4	3.0	0.0223	9.3e-06	8.9e-03
Case5	0.074	0.0223	3.1e-02	1.0e-01
Case6	-1.0	0.0223	1.0e-01	1.3e-01

Table 2: Suboptimality certificates for Frank-Wolfe rounding vs. Hamming rounding on the MarchingRally sequence for different parameter value settings of the objective. The first two columns give the parameter value for the detection confidences and the overlap penalty respectively for each case. The last two columns give the suboptimality certificate for Frank-Wolfe rounding and Hamming rounding (lower is better).

B. Video Results

The following images in Figure 5 shows the tracks overlaid on top of the first frame of the MarchingRally sequence. Each track is shown in a separate color. The output on the top illustrates our result (NF+Overlap) and the one on the bottom illustrates the results of [22] (Greedy + NMS). Note how in our case one gets non-overlapping tracks while in the case of [22] there are places where tracks overlap and criss-cross. We highlight this in videos available from [1] by drawing cyan colored boxes at places where such ID swaps happen. See Figure 3(a) for the corresponding re-detection curves. For the more classical metrics, the (MOTA, MOTP, IDswap) numbers for NF+Overlap are (27.7%, 66.5%, 11) vs. (22.5%, 66.0%, 24) for Greedy+NMS.

C. Runtime and Constraints

For the PETS and TownCenter dataset, typically we have approximately 10–40 detections per frame. For PETS data, each detection is connected to a detection in another frame (with a pairwise term) if they are less than 6 frames apart. On average, each detection is connected to about 10 other detections for pairwise terms (overlap+CO), which means the number of pairwise terms is linear in the number of unary terms. For TownCenter data, we connect detections over 30 frames to account for slower motion of people and missing detections, resulting in about 15 pairwise terms (overlap+CO) per detection on average. While our algorithm runs in about 5–10 seconds on the PETS dataset, it takes about 30–45 minutes on the TownCenter dataset. This

[9] We suppose a multiplicative constant of one in front of the correspondence strengths; changing it as well would just amount to multiply the whole objective by a constant, which would not change the solution.

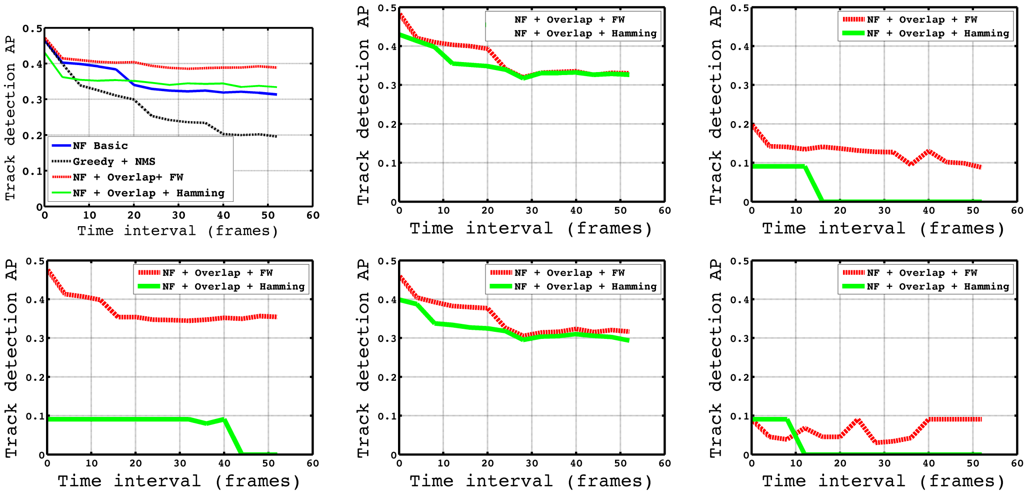


Figure 4: Re-detection accuracy for the cases presented in Table 2 on the MarchingRally sequence. Top row is Case 1 to 3 (varying the overlap penalty); bottom row is Case 4 to 6 (varying the detection confidence weight); Case 1 is Figure 3(a) in the paper with the additional Hamming rounding curve. Hamming rounding yields systematically worse re-detection accuracy than Frank-Wolfe rounding.

difference is due to the larger number of frames in the TownCenter dataset (one order of magnitude greater than for the PETS videos), and also the larger number of pairwise terms per detection on average, resulting in a LP with about 5 million variables in comparison to about 50 thousand for the PETS sequences.

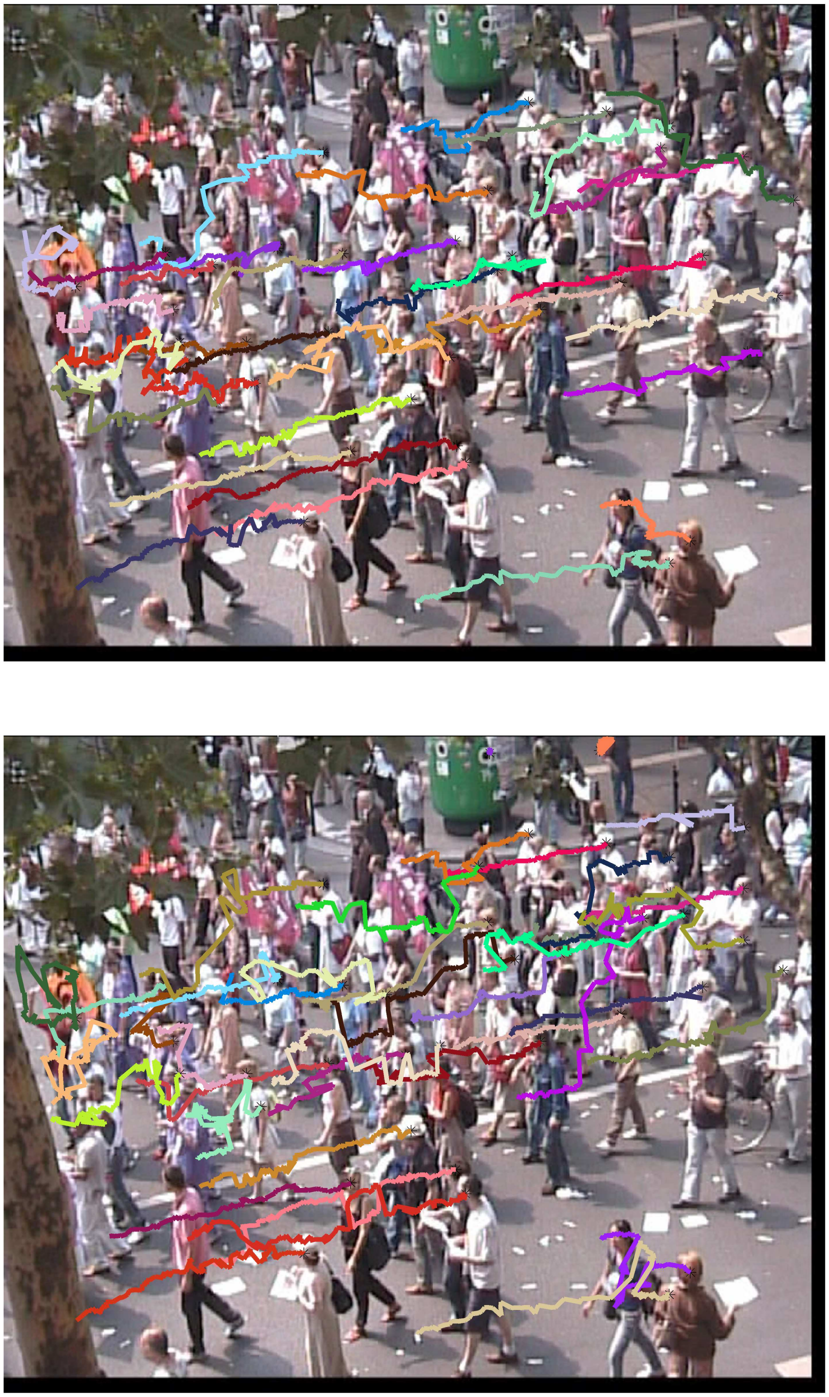


Figure 5: First image of MarchingRally sequence overlaid with tracks. Each track is set to a different color. Tracks on top show our result (NF+Overlap), while tracks on the bottom show the result of [22] (Greedy + NMS)