On Pairwise Cost for Multi-Object Network Flow Tracking

Vishes Chari  
INRIA - Willow Project-team  
Ecole Normale Supérieure, Paris, France

Simon Lacoste-Julien  
INRIA - Sierra Project-team  
Ecole Normale Supérieure, Paris, France

Ivan Laptev  
INRIA - Willow Project-team  
Ecole Normale Supérieure, Paris, France

Josef Sivic  
INRIA - Willow Project-team  
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 an approach that regularizes the tracker by adding second order costs to the min-cost network flow framework. While solving such a problem with integer variables is NP-hard, we present a convex relaxation with an efficient rounding heuristic which empirically gives certificates of small suboptimality. Results are shown on real-world video sequences and demonstrate that the new constraints help selecting longer and more accurate tracks improving over the baseline tracking-by-detection method.

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 [1]. Despite recent progress in the field, tracking under challenges of clutter or occlusion (e.g. in crowded scenes) is still a tall order. In such scenarios, tracks computed using methods that, for example, entirely rely on feature correspondences tend to “drift” away from the actual object over time. Moreover, partial overlap between different targets severely restricts tracking of individual objects in a standalone manner, and mandates group tracking.

To deal with the above problems, “tracking-by-detection” approaches relying on object detectors have recently become popular [2, 3, 4, 5]. Given object detections per frame in a video sequence, such methods connect detections across frames into tracks using some measure of correspondence. This minimizes the “drift” experienced by feature-based trackers, and provides robustness to partial occlusion. Given object detections in every frame, the tracking problem can be formulated as selection and clustering of corresponding detections over time. Moreover, track selection can be solved globally without early decisions that are typical for on-line trackers. Tracking-by-detection is typically formulated as a MAP estimation problem [6] that reduces to a min-cost network flow problem [7] which can be efficiently solved [8].

Another advantage of the “global” track selection approach is that it enables a principled way of modeling interactions between tracks. Motion models of person interactions have been shown to improve human tracking in crowds [9], or identify unusual behavior [10]. Some models have also
Figure 1: Pairwise terms can be used to remove overlapping tracks caused by clutter (a) & (b), and “ghost” tracks attached to shirts of people (c) & (d).

helped to resolve ambiguities in label assignment of tracks \cite{11}, impose constraints of constant velocity to handle partial occlusions \cite{5} or overcome handicaps arising out of skewed camera viewpoints, clutter or indeed complex motions.

In this work, we propose to encode relations among tracks by modeling the spatial “co-occurrence” of detections and object motion between frames using pairwise costs. We show that such characteristics tend to “bias” final solutions towards tracks that satisfy the considered “co-occurrence” criteria. We show that such a bias gives more sustained tracking in the presence of clutter or partial occlusions, and can even incorporate multiple sources of information for robust tracking in videos. We further show that these costs can be minimized in a robust manner through a linear optimization relaxation followed by a rounding heuristic that provides suboptimality certificates. Results on real-world videos demonstrate a significant improvement over the state-of-the-art tracking-by-detection methods.

2 Related work

Recent approaches have formulated multi-frame, multi-object tracking as a min-cost network flow optimization problem \cite{6, 5, 11}, where the optimal flow in a connected graph of detections encodes the selected tracks. Proposed approaches to solve for the optimal min-cost flow include push-relabel methods \cite{6}, successive shortest paths \cite{8, 2}, and dynamic programming in a greedy framework \cite{8}. While solving for the flow greedily is an efficient algorithm because of its simple nature, the cases presented in these papers restrict the cost to unary terms over all edges. Thus they are unsuitable as a framework for minimizing pair-wise costs without extensive modification of algorithms. Indeed, adding any new term might require re-designing algorithms to include its effect on the overall minimization.

Brendel et al. \cite{12} and Milan et al. \cite{11, 13} formulate the problem in a framework that first selects tracklets and then connects them using a learned distance measure \cite{12} or a CRF \cite{11, 13}. Long term occlusions are handled in \cite{12} by merging appearance and motion similarity. While \cite{11, 13} propose to alternate between discrete and continuous optimizations in order to minimize several cost functions, the presence of two levels of optimization makes the minimization non-convex and so theoretical or empirical guarantees of optimality are hard to give. This eventually means that every cost needs to be separately verified for its performance on large datasets, and combination of costs need to be treated as different minimizations altogether.

Other approaches \cite{3, 14, 15} use offline or online training to learn a similarity measure between tracklets. The proposed optimization methods don’t provide any optimality guarantee, though. In addition, training might be difficult in some important conditions. For example, online training to discriminate appearances might be erroneous when objects move very close to each other (Figure 1).

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 \cite{2, 3}, which either build on top of min-cost flow solutions \cite{11} or
while still solving the quadratic costs with particular models in mind in Section 3.1. We then discuss optimization strategies in 4.2. Finally we outline strategies to achieve near-optimal integer solutions in Section 4.3 and show results on challenging datasets to validate our theory in Section 5.

2.2 Background: min-cost flow tracking

In this section, we describe the traditional formulation of multi-object tracking as a min-cost flow optimization problem [6]. We extend this framework in Section 3.1.

Suppose that we wish to track several objects simultaneously in a video by using the “detect-and-track” framework [6]. That is, we run object detectors independently on every frame of the video and then connect detections over time to extract object tracks. Connecting might be done by using some correspondence criteria based from KLT or SIFT features between pair of detections from consecutive frames. The joint selection problem of multiple tracks from observed features can be formalized through a MAP objective [6] with specific constraints encoding the structure of the tracks. The MAP problem can be equivalently cast as the integer linear program (ILP) in equation (1).

The correspondence with a network flow structure is summarized in Figure 2. The joint selection of $K$ tracks has been encoded 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. The quality of multiple tracks is quantified by the cost function. $c_i$ 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 constraint $\sum_{i,j \in E} x_{ij} = x_j = \sum_{i,j \in E} x_{ji}$, which has the structure of a flow conservation constraint [7], 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 $\sum_i x_{it} = K = \sum_j x_{si}$ 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 [7]. 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.

To summarize, the above optimization problem can be solved efficiently using existing network flow or linear algebra packages [7] when the integer constraint is relaxed, 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.
3 Modeling pairwise costs with an integer quadratic program

The above formulation in equation (1) represents a linear objective with linear equality constraints (where the integer constraint is not needed). While linear equations are both simple and easy to minimize, higher order models can represent more useful properties [9]. 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 \( z \). 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 \( Q \) of such joint selections, and add the term \( \sum_{ij \in Q} z_i z_j Q_{ij} \) to the objective. Our new optimization problem can thus be expressed as the integer quadratic program (IQP) in (2), where the \( Q \) matrix is sparse and has non-zero entries \( Q_{ij} \) for \( ij \in Q \). Unfortunately, the above formulation can encode the quadratic assignment problem which is NP-hard to optimize in general [16]. Nevertheless, we propose an efficient (convex) linear relaxation in Section 4 as well as a powerful rounding heuristic that provides empirical certificates of suboptimality. Our main modeling strategy is thus twofold: 1) we encode our prior knowledge about the joint selection of variables using the sparse cost matrix \( Q \) (which can be arbitrary); 2) we can also 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 4.3). For the rest of this section, we provide examples of prior knowledge that can be encoded in our framework; we will focus on the optimization aspects in the following section.

3.1 Designing pairwise costs

In the following subsections, we show how some of the traditional constraints [10, 9] could be incorporated in our quadratic min-cost flow network framework. We focus on elements that cannot be simply encoded with traditional linear cost terms in (1).

3.1.1 Overlap penalty

One frequently used pruning strategy while dealing with detections is Non Maxima Suppression (NMS) [8]. Since object detectors fire repeatedly around objects in an image, this constraint enforces only one detection to be selected from an overlapping bunch of detections. It is thus an important tool to make sure multiple tracks are not selected for a single object. Previously algorithms like Pirsiavash et. al [8] have implemented this condition greedily by first selecting a track and then pruning all overlapping detections before selecting the next track. Others like Andriyenko et. al [17] have encoded it as a hard constraint in the optimization problem. Our approach is to design a new term that ensures NMS is performed simultaneously with tracking in a soft (scored) approach. Let
\(x_i\) and \(x_j\) represent two overlapping detections. We add a positive penalty \(Q_{ij}\) when both \(x_i\) and \(x_j\) are selected simultaneously. Such a penalty can also be added for connection-detection overlaps and connection-connection overlaps. One advantage of our soft approach is that when two people cross each other, overlapping detections might be allowed because of the upside of having longer tracks. Thus only tracks with repeated overlaps are discouraged by our method.

### 3.1.2 Enforcing consistency between two signals

In many tracking scenarios, multiple signals are available for use. For example, we might have separately trained detectors available for different regions of the body like head and torso. In case they give complementary information about the presence of the object, we can achieve better performance by combining them using a pairwise cost. Let \(x^h_i\) and \(x^t_i\) denote detections of head and torso for example. Each set can be associated with its own flow feasible set \(\text{FLOW}^h_K\) and \(\text{FLOW}^t_K\). If \(z^h\) and \(z^t\) are variables corresponding to individual flows, we can encourage "co-occurrence" of the two flows by adding the negative term \(-z^h_i z^t_i\) to the cost. We could encourage co-occurrence of detections \(r_{ij} x^h_i x^t_j\), connections \(r_{ijkl} x^h_{ij} x^t_{kl}\), or connections and detections \(r_{ijk} x^h_i x^t_j\).

### 4 Optimization strategies

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 \(2\) to encourage co-occurrence preferences for individual variables in the minimization. Finding a global minimum is NP-hard \([16]\) 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 4.3). 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 the 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 the problem \(2\).

#### 4.1 Quadratic optimization

If \(Q\) is positive definite, then the relaxed quadratic program (QP) in \(2\) obtained after removing the integer constraint is convex and can be robustly optimized using interior point methods implemented in standard commercial solvers such as MOSEK and CPLEX. These methods can scale to medium-size problems \(^1\) by exploiting the sparseness of \(Q\) suggested in Section 3.1.

In our general formulation, \(Q\) is not necessarily positive definite though. We can nevertheless use a standard trick to make it positive definite by defining its diagonal entries to be \(Q_{ii} = \sum_{j \neq i} |Q_{ij}|\), while using \(c_i - Q_{ii}\) as the linear coefficient for \(z_i\) in the objective. As \(z_i^2 = z_i\) for binary variables, this transformation yields an equivalent IQP. On the other hand, \(Q\) is then diagonally dominant and thus positive semidefinite \([18,\text{Thm. 6.1.10}]\), and so the relaxation gives a convex problem.

In order to scale to very large scale datasets (billions variables), one could use the Frank-Wolfe algorithm \([19]\) 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 4.3.

---

\(^1\)A few millions variables, which translates to several hundreds frames with a high number of detections for our datasets.
4.2 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 follows a similar line of reasoning as in \([20]\).

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 now shown in \((3)\).

\[
\begin{align*}
\min_{z, u} & \quad c^\top z + z^\top Q z \\
\text{s.t.} & \quad z \in \text{FLOW}_K \\
& \quad z \text{ integer}, \\
& \quad u_{ij} \leq z_i, u_{ij} \leq z_j \\
& \quad z_1 + z_j \leq 1 + u_{ij} \\
& \quad z, u \text{ integer}. 
\end{align*}
\]

Here \(u\) and \(q\) represents the vector whose elements are \(u_{ij}\) and \(Q_{ij}\) respectively. The new constraint \(z_1 + z_j \leq 1 + u_{ij}\) enforces that \(u_{ij}\) should be 1 if \(z_1\) and \(z_j\) are both 1; while the pair of constraints \(u_{ij} \leq z_i\) and \(u_{ij} \leq z_j\) enforce that \(u_{ij}\) should be zero if either \(z_i\) or \(z_j\) is zero. We call these additional constraints 'LOCAL(Q)' as it turns out that they define a polytope which can be obtained by a projection of the \textit{local marginal consistency} polytope for the over-complete representation of a discrete Markov random field (MRF) \([21]\) \((4.6)\) with edges defined by the non-zero entries of \(Q\).

Removing the integer constraint in \((3)\) thus yields a LP relaxation that is similar to a standard one for \textit{MAP} inference in MRFs, but with the additional \(\text{FLOW}_K\) constraints, yielding a crucial structural difference with the previous work.

An advantage of the formulation \((3)\) 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 \(Q\), which was indicated by the set \(Q\) in \((3)\) to stress that we don’t 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 the formulation \((3)\) is that we can easily generalize it to handle higher 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 LOCAL(Q) for higher order terms and yield another ILP that can be relaxed to a LP.

4.3 Frank-Wolfe rounding heuristic

The solution of the LP relaxation of \((3)\) can have fractional components because the additional linear constraints from LOCAL(Q) essentially violate the \textit{total unimodularity} property, in contrast to \(\text{FLOW}_K\) 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 \(z^*\), the simplest approach would be to look for the point closest in Euclidean norm in \(\text{FLOW}_K\) which is an integer. As \(z^*_i = z_i\) for binary variables, we have \(||z - z^*||^2 = (1 - 2z^*)^\top z + ||z^*||^2\) which is a linear function of \(z\). We can thus obtain the closest integer point by solving a LP over \(\text{FLOW}_K\), as all its vertices are integers. We call this approach \textit{Hamming rounding} as \(d_H(z, z^*) := ||z - z^*||^2\) reduces to the Hamming distance when evaluated on pair of binary vectors. On the other hand, the closest point in Euclidean norm doesn’t necessarily yield a good

\[2\] 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 \([22][21]\) 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 LOCAL(Q).
Figure 3: (Top) Re-detection results of including overlap and co-occurrence terms in the linear relaxed formulation, vs state-of-the-art. x-axis in these results represent median length of tracks in the video. (a) & (b) Show performance for the “MarchingRally” and “TownCenter” sequences respectively, where clutter and noise cause tracking problems. Notice that in (a) overlap term significantly improves basic network flow which in turn outperforms the greedy baseline. In (b) adding a co-occurrence term to the network flow formulation provides significant improvement over the greedy baseline, as well as the basic network flow baseline. (Bottom) Tracking results for “TownCenter” evaluated in terms of MOTA/MOTH/Precision/Recall measures, compared with results of [23].

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 $z^*$. Specifically, we obtain the following LP, which has the usual network flow constraint structure and thus can be solved very efficiently:

$$\min_{z} \quad (c + (Q + Q^T)z^*)^T z$$
$$\text{s.t.} \quad z \in \text{FLOW}_K. \quad (4)$$

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. Empirically, we obtained very small suboptimality certificates ($\approx 10^{-3}$) for our returned integer solutions, indicating that our rounding heuristic was effective at returning near-global optimal solutions. Hamming rounding was not as effective. We also note that in contrast to the previous work [5] 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.

5 Experiments

In this section, we evaluate our approach on real world videos through experiments on two challenging datasets (Section 5.2). The first video of a crowd moving through a street is challenging because of clutter. Head detectors fire repeatedly in and around groups of people, and detections often overlap significantly. This generally throws off trackers as they find tracks that slowly drift
from one person to another. For the second video, we have two signals in the form of head and torso detectors for each frame. While head detectors are noisy (low precision) but have high recall, torso detectors have high precision. Thus we demonstrate the use of our “co-occurrence” term by increasing the precision of head tracks by coupling them to torso tracks.

5.1 Tracking datasets

We test our algorithms on two datasets. The first sequence shown in Figure 1 (top row) is called “MarchingRally” sequence and is that of a crowd walking in a rally along a street. The video consists of 120 frames at 25 fps, and has around 50 people. This video is challenging because issues of clutter get amplified due to the high number of people present, and their proximity to each other while walking. We run a “head” detector [24] to detect heads of people in every frame. We then run a KLT tracker after initializing features within the bounding boxes of heads. Finally, for every pair of frames that are less than a certain threshold apart (10 frames), we connect pairs of detections with high values of correspondence strength. The strength of correspondence between two detections is defined by the number of their common KLT tracks divided by the total number of KLT tracks passing through both detections. Ground truth tracks are marked for all the 50 people in order to evaluate tracking results.

The second video shown in Figure 1 (bottom row) is called “TownCenter” [23] sequence and consists of 4500 frames at 25 fps, and around 230 people who are walking across the street. We have head and torso detections for every frame, and we run a KLT tracker to separately link head and torso detections across frames. Correspondence strength is then computed similarly as for the previous video.

5.2 Tracking in video experiment

5.2.1 Evaluation strategy

Evaluating a multi-object tracking algorithm can be difficult because errors might be present in various forms like ID swaps, broken tracks for a single object, or false positives. It is thus difficult to glean any information from scores like MOTA [11, 23] which combine all these errors into a single number.

Some methods in the past [11] have focused on metrics like distance between selected and ground truth tracks, count of ID swaps occurring in a video etc., they might not be a suitable metric for evaluation in all scenarios. For example, in crowd videos Figure 1(a), it is easy to have a low distance measure for tracks even if they continuously swap between neighboring people. On the other hand, ID swaps can be minimal when tracks are fragmented. Thus, we need an integrated measure that penalizes ID swaps while measuring accuracy at the same time. In order to do this, we propose a measure that extends the precision recall (PR) evaluation strategy of object detection as used in [25].

Re-detection measure. We propose to measure the “re-detection” accuracy of a track, which evaluates correct tracking for at least \( n \) frames where \( n \) varies from 0 to several frames. To do this, we first extract all subtracks of length \( n \) from selected tracks. Each subtrack is then marked true positive or false positive if its first and last detections both overlap with a single ground truth track in corresponding frames. Consequently, we can plot a PR curve for the entire video using this approach. Each PR curve is summarized by the average precision, which is computed as the area under the PR curve [25]. Finally, the collection of average precisions for varying values of \( n \) summarizes how accurately and continuously a track is selected. Figure 3 shows two such graphs for various optimizations. Such a measure has several advantages. An ideal score in such a scenario is a value of 1 for all values of \( n \). It signifies that there are no ID swaps or track fragmentation and distances between ground truth and detected tracks are limited because of overlap with ground truth bounding boxes. It also accounts for variations in detection sizes to which distance computation measures might be sensitive. Finally, such a graph shows the performance of an algorithm as tracks get longer, which measures the ability of a tracker to remain accurate.
5.2.2 Experimental results

We compare our algorithm with the state-of-the-art approaches on two video sequences. The baseline approach is a greedy implementation of the basic min-cost network flow algorithm \[8\], 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.

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 in the video alleviate the issue of clutter, which leads to failure of tracking algorithms that tend to confuse tracking identities repeatedly. Our algorithm with overlap constraints (red curve) outperforms the state of the art by a large margin. Figure 3(a) shows the re-detection accuracy results with and without the overlap constraints. Notice that as the evaluation time interval increases, the difference between our algorithm and that of \[8\] increases. In fact, for 40 frames or more, our algorithms outperforms the baseline by over 20%.

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 in Figure 3(b) we leverage body detections to improve noisy head tracks. Again in this case, there is more than a 20% improvement over the head baseline. Finally, the table in Figure 3 compares our method with a state-of-the-art \[23\] algorithm in terms of traditional MOTA evaluation measure.

6 Discussion and conclusion

We have shown how to minimize the traditional “detect-and-track” cost function in the presence of pairwise cost terms. The proposed method can incorporate several second order models that translate to useful constraints on tracks. We have experimentally demonstrated that adding the second order terms significantly improves over the state-of-the-art baseline tracker, and produces more consistent and long-term tracks.

Combining different types of pairwise terms into a single (linear) cost opens up the possibility of tracking complicated motions. In contrast, using greedy algorithms is tedious since different algorithms have to be engineered for different costs. Finally, while more complex cost functions have more parameters that have to be tuned, the parameters could be learnt from labeled data using structured output learning \[20\]. This opens up the possibility of learning second order tracking costs for specific crowd actions such as panic, street crossing or stampede.

7 Acknowledgements

We thank Prof. Patrick Perez for conversations over email about multi-target tracking evaluation algorithms. This work is partly supported by the Quaero Programme, funded by OSEO, the MSR-INRIA laboratory, ERC grant Activia, and the EIT ICT Labs.

References

[1] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4), 2006.
[2] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. In *PAMI*, 2011.
[3] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *CVPR*, 2009.
[4] H. Jiang, S. Fels, and J. Little. A linear programming approach for multi-object tracking. In *CVPR*, 2007.
[5] A. A. Butt and R. T. Collins. Multi-target tracking by Lagrangian relaxation to min-cost network flow. In *CVPR*, 2013. ISBN 978-0-7695-4989-7.
[6] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*, 2008.
[7] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice-Hall, Inc., 1993. ISBN 0-13-617549-X.
[8] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, 2011. ISBN 978-1-4577-0394-2.
[9] 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.
[10] L. Kratz and K. Nishino. Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In CVPR, 2009.
[11] A. Milan, K. Schindler, and S. Roth. Detection- and trajectory-level exclusion in multiple object tracking. In CVPR, 2013.
[12] W. Brendel, M. Amer, and S. Todorovic. Multiobject tracking as maximum weight independent set. In CVPR, 2011.
[13] A. Milan, S. Roth, and K. Schindler. Continuous energy minimization for multi-target tracking. PAMI, 2013.
[14] B. Yang, C. Huang, and R. Nevatia. Learning affinities and dependencies for multi-target tracking using a CRF model. In CVPR, 2011.
[15] B. Yang and R. Nevatia. An online learned CRF model for multi-target tracking. In CVPR, 2012.
[16] E. M. Loula, 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.
[17] A. Andriyenko and K. Schindler. Globally optimal multi-target tracking on a hexagonal lattice. In ECCV, 2010.
[18] R. A. Horn and C. R. Johnson. Matrix Analysis. Cambridge University Press, 1985.
[19] M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In ICML, 2013.
[20] S. Lacoste-Julien, B. Taskar, D. Klein, and M. I. Jordan. Word alignment via quadratic assignment. In NAACL, 2006.
[21] 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.
[22] D. Bertsimas and J. N. Tsitsiklis. Introduction to Linear Optimization. Athena Scientific, 1997.
[23] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. In CVPR, pages 3457–3464, June 2011.
[24] M. Rodriguez, I. Laptev, J. Sivic, and J.-Y. Audibert. Density-aware person detection and tracking in crowds. In ICCV. IEEE Computer Society, 2011. ISBN 978-1-4577-1101-5.
[25] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. IJCV, 2010.
[26] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In CVPR, 2008.
[27] B. Wu and R. Nevatia. Tracking of multiple, partially occluded humans based on static body part detection. In In CVPR, 2006.