Subgraph Decomposition for Multi-Target Tracking

Siyu Tang[1] Bjoern Andres[1] Mykhaylo Andriluka[1,2] Bernt Schiele[1]
[1]Max Planck Institute for Informatics, Saarbrücken, Germany [2]Stanford University, USA

Abstract

Tracking multiple targets in a video, based on a finite set of detection hypotheses, is a persistent problem in computer vision. A common strategy for tracking is to first select hypotheses spatially and then to link these over time while maintaining disjoint path constraints [14, 15, 24]. In crowded scenes multiple hypotheses will often be similar to each other making selection of optimal links an unnecessary hard optimization problem due to the sequential treatment of space and time. Embracing this observation, we propose to link and cluster plausible detections jointly across space and time. Specifically, we state multi-target tracking as a Minimum Cost Subgraph Multicut Problem. Evidence about pairs of detection hypotheses is incorporated whether the detections are in the same frame, neighboring frames or distant frames. This facilitates long-range re-identification and within-frame clustering. Results for published benchmark sequences demonstrate the superiority of this approach.

1. Introduction

Multi-target tracking can be formulated as an optimization problem with respect to a graph whose nodes correspond to detection hypotheses and whose edges connect detection hypotheses that hypothetically describe the same target. A commonly employed objective of the optimization is to select a subset of nodes and edges in such a graph to maximize similarity of connected detection hypotheses, while maintaining constraints that prevent splits and merges of tracks.

By far the most common approach is to choose the initial graph such that detection hypotheses are connected only across time (not within the same time frame) and to constrain the solution such that connected components of selected detection hypotheses are paths (that do not branch). With respect to a linear objective function, this problem is a Minimum Cost Disjoint Paths Problem with respect to the initial graph. It is used, explicitly or implicitly, in many modern tracking algorithms including [14, 15, 1, 25].

While being intuitive, the Disjoint Paths formulation has a notable caveat: Typical target detectors yield, for each time frame, many similar (and typically equally plausible) detections of the same target. Within the Disjoint Paths formulation, it becomes necessary to choose, for each time frame and target, one best out of many similar (and plausible) hypotheses. Various recipes are proposed in the literature to address this challenge. E.g., [14, 1] rely on a greedy iterative procedure that finds one track at a time and then removes corresponding hypotheses, or [25] performs several rounds of optimization that merge detections into tracklets and then into full tracks. Unfortunately, all these methods depend on parameters that need to be tuned carefully, as noted in [14, 1, 25].

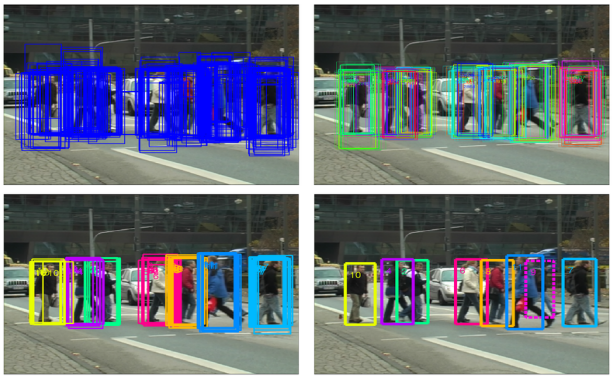


Figure 1. Overview of the Subgraph Multicut tracking method: (clockwise) detection hypotheses, overlapping tracklet hypotheses, hypotheses decomposition (clustering jointly across space and time) and final tracks (dotted rectangles are interpolated tracks).

Embracing the possibility of having multiple plausible hypotheses per target and frame motivates us to formulate multi-target tracking as a Minimum Cost Subgraph Multicut Problem. The feasible solutions of this formulation are such that possibly multiple hypotheses per track and time frame are selected and clustered, resulting in an overall rigorous and elegant approach to link, cluster and track targets *jointly* across space and time. To illustrate the similarities and differences to prior work we implement a version of a tracking algorithm based on the Minimum Cost Disjoint Path Problem. Although conceptually simple, its output is already on par with the state of the art for public benchmark sequences, as we show in Sec. 6.

This paper makes the following contributions: *First*, to our knowledge, our work is the first to propose a Subgraph

Multicut model for the multi-target tracking problem jointly solving the spatial *and* temporal associations of detection hypotheses. *Second*, we provide an in-depth analysis and comparison of the Subgraph Multicut and the Disjoint Paths models. Our results suggest that the Subgraph Multicut model has considerable advantages due to the fact that state-of-the-art object detectors output multiple hypotheses per target. *Third*, besides proposing an exact solver, we also provide a heuristic solution based on the Kernighan-Lin algorithm [13], which makes the method applicable to large sequences. Finally we perform extensive experiments and present superior results compared to the state-of-the-art.

2. Related Work

Much of the recent literature on multi-target tracking follows the tracking-by-detection strategy using target detectors to establish an initial state-space of detection hypotheses in each frame. Given such an initial state-space a popular approach to tracking is to formulate it as a combinatorial optimization problem of linking detection hypotheses across frames. Various ways to link hypotheses are proposed such as methods based on network flow [25, 24], or integer linear programming [16, 20]. Other approaches are iteratively finding one track at a time by iteratively solving the MAP estimation problem [14] or jointly finding a set of tracks with continuous optimization [3]. The majority of these approaches employ some strategy to reduce the search complexity by performing early grouping or non-maximum suppression of hypotheses. For example, it is common to first group single-frame hypotheses into tracklets spanning several adjacent frames, and then combine them into complete tracks [12, 19, 24, 11, 22, 23, 2].

Various strategies are proposed to deal with the variable number of tracking targets. [14, 24] rely on a greedy approach that recovers tracks one at a time by iteratively reducing the state space. [3] jointly optimizes tracking trajectories and the number of tracking targets. [15] implicitly encodes the number of tracks by linking individual detection hypothesis between neighboring frames.

Literature on multi-target tracking is vast but several key properties reappear in a number of successful approaches: leveraging long-range associations to prevent ID switches and recover missing detections caused by long-term occlusion [3, 24]; jointly inferring the number of tracks and solving the data association problem [15, 3]; exploring appearance information and combine it with long-range associations [24, 15]; integrate non-maximum suppression with tracking [1, 14]. Our Subgraph Multicut formulation allows to combine all these in the same framework.

Approaches of [15, 24] are perhaps closest to ours. Similarly to [15] we implicitly encode the number of tracks by linking tracklet hypotheses. However our approach jointly reasons about connectivity of groups of hypotheses, whereas

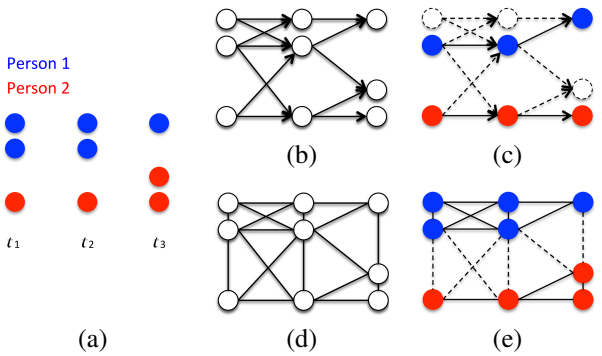


Figure 2. Two person detection hypotheses in three consecutive frames (ground truth assignment depicted in color) **(a)**; The disjoint paths **(c)** obtained by solving a Minimum Cost Disjoint Paths Problem with respect to a directed graph **(b)**; The decomposition **(e)** obtained by solving a Minimum Cost Subgraph Multicut Problem with respect to an undirected graph **(d)**.

they connect individual hypotheses only. Our approach incorporates long-range connections between hypotheses, and we show that our approach achieves better experimental results compared to [15]. [24] also introduces long-range connections between hypotheses and uses an iterative greedy procedure finding tracks one at a time, whereas we jointly solve for all tracks. [11] aims to delay resolution of local ambiguities by introducing "tree-tracklets" that delay locally ambiguous decision until more information is available. Our approach achieves the same goal by jointly associating groups of detections.

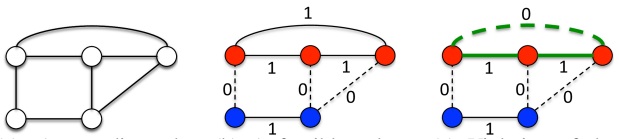
3. Formulations of Multi-Target Tracking

Before introducing the formulations for the Subgraph Multicut Problem and the Disjoint Paths Problem, we illustrate the difference between them by visualizing a toy example in Fig. 2: (c) shows a solution of the Minimum Cost *Disjoint Paths* problem that finds disjoint trajectories for all targets in a directed graph; and (e) shows a solution to the Minimum Cost *Subgraph Multicut* problem that corresponds to a decomposition of an undirected graph.

3.1. Disjoint Paths Problem

We now summarize the formulation of multi-target tracking as a Minimum Cost Disjoint Paths Problem (Def. 1). The formulation is with respect to a *directed* graph $G = (V, E)$ whose nodes V are all hypothesized detections of an entire video and whose edges E connect pairs of detection hypotheses that hypothetically describe the same target in the different frames. More specifically, every edge $vw \in E$ points forward in time, i.e., the frame of the detection v is strictly smaller than the frame of the detection w .

The feasible solutions of the Minimum Cost Disjoint Paths Problem (Def. 1) are subgraphs $G' = (V', E')$ of G which are encoded by $x \in \{0, 1\}^V$, the characteristic



(a) An undirected graph G (b) A feasible solution (c) Violation of the cycle constraint

Figure 3. **(a)** An undirected graph G; **(b)** A feasible solution of the Minimum Cost Subgraph Multicut Problem (Def. 2) on G, two connected components are in red and blue respectively, the set of edges with value 0 (dotted lines) is a multicut of the graph G; **(c)** The cycle constraint (9) is violated for the cycle depicted in green.

function of the subset $V' = \{v \in V \mid x_v = 1\} \subseteq V$ of nodes, and $y \in \{0,1\}^E$, the characteristic function of the subset $E' = \{vw \in E \mid y_{vw} = 1\} \subseteq E$ of edges. More specifically, the subgraph G' is constrained (by Def. 1) to be a set of disjoint paths in G. The objective function is linear in the coefficients of x and y:

Definition 1 With respect to a directed graph $G = (V, E)$, $c \in \mathbb{R}^V$ and $d \in \mathbb{R}^E$, the 01-linear program written below is called an instance of the *Minimum Cost Disjoint Paths Problem*.

$$\min_{\substack{x\in\{0,1\}^V \\ y\in\{0,1\}^E}} \quad \sum_{v\in V} c_v x_v + \sum_{e\in E} d_e y_e \tag{1}$$

$$\text{subject to} \quad \forall e = vw \in E: \quad y_{vw} \leq x_v \tag{2}$$

$$\forall e = vw \in E: \quad y_{vw} \leq x_w \tag{3}$$

$$\forall v \in V: \quad \sum_{vw\in E} y_{vw} \leq 1 \tag{4}$$

$$\forall w \in V: \quad \sum_{vw\in E} y_{vw} \leq 1 \tag{5}$$

Here, c_v and d_e correspond to the unary and pairwise costs. The constraints (2) and (3) state that an edge can only be selected if both its nodes are selected. The constraints (4) and (5) state that every node has at most one incoming edge and at most one outgoing edge, respectively, effectively implementing the Disjoint Paths constraint.

3.2. Subgraph Multicut Problem

We now formulate multi-target tracking as a Minimum Cost Subgraph Multicut Problem (Def. 2). The formulation is with respect to an *undirected* graph $G = (V, E)$ whose nodes V are all hypothesized detections of an entire video and whose edges E connect pairs of detection hypotheses that hypothetically describe the same target, including pairs in the same video frame.

The feasible solutions of the Minimum Cost Subgraph Multicut Problem (Def. 2) define subgraphs $G' = (V', E')$ of G which are encoded by $x \in \{0,1\}^V$, the characteristic function of the subset $V' = \{v \in V \mid x_v = 1\} \subseteq V$ of nodes, and $y \in \{0,1\}^E$, a characteristic function defining the subset $E' = \{vw \in E \mid y_{vw} = 1\} \subseteq E$ of edges. More specifically, the subgraph G' is constrained (by Def. 2) such that each connected component (V'', E'') of G' contains all edges $E'' = \binom{V''}{2} \cap E$. We show an example graph and a feasible solution in Fig. 3.

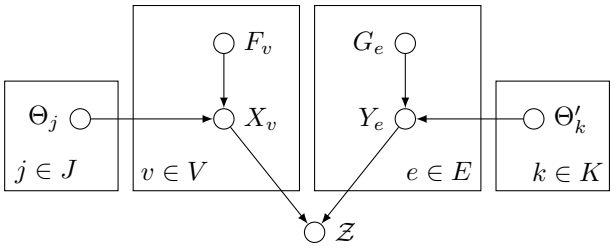


Figure 4. A Bayesian Network of probability measures of characteristic functions of subgraphs.

The objective function of the Minimum Cost Subgraph Multicut Problem is linear in the coefficients of x and y:

Definition 2 With respect to an undirected graph $G = (V, E)$, $c \in \mathbb{R}^V$ and $d \in \mathbb{R}^E$, the 01-linear program written below is called an instance of the *Minimum Cost Subgraph Multicut Problem*.

$$\min_{\substack{x\in\{0,1\}^V \\ y\in\{0,1\}^E}} \quad \sum_{v\in V} c_v x_v + \sum_{e\in E} d_e y_e \tag{6}$$

$$\text{subject to} \quad \forall e = vw \in E: \quad y_{vw} \leq x_v \tag{7}$$

$$\forall e = vw \in E: \quad y_{vw} \leq x_w \tag{8}$$

$$\forall C \in \text{cycles}(G)\ \forall e \in C: \quad (1 - y_e) \leq \sum_{e'\in C\setminus\{e\}} (1 - y_{e'}) \tag{9}$$

Here, the constraints (7) and (8) state that an edge can only be selected if both its nodes are selected. The cycle constraints (9) state, firstly, that every component of the selected subgraph G' is also a component of G and, secondly, that every edge of G whose nodes are in the same component of G' is also in G. An example of violation is shown in Fig. 3(c). In the context of multi-target tracking this implies that if a detection hypothesis is connected (spatially or temporally) to another detection hypothesis, all neighbors of the first hypothesis have to be connected to all spatial and temporal neighbors of the second hypothesis as well.

3.3. Probabilistic Model

Toward the goal of learning and inferring the parameters c and d of both optimization problems (Def. 1 and 2) from video data and toward the goal of comparing the two formulations of the multi-target tracking problem, we now define a probability measure on subgraphs of a graph $G = (V, E)$ such that a maximally probable set of disjoint paths is precisely a solution of the Minimum Cost Disjoint Path Problem (Def. 1) and such that a maximally probable subgraph multicut is precisely a solution of the Minimum Cost Subgraph Multicut Problem (Def. 2).

More specifically, we define a probability measure on the characteristic functions $x \in \{0,1\}^V$ and $y \in \{0,1\}^E$ with respect to the Bayesian Network depicted in Fig. 4. Realizations of the random variables X and Y are the characteristic functions x and y. For a finite index set J and every $v \in V$, a realization of the random variable F_v is a vector $f^v \in \mathbb{R}^J$ of features of the node v. For a finite index set K and every $e \in E$, a realization of the random variable G_e is a vector $g^e \in \mathbb{R}^K$ of features of the edge e. A realization of the random variable Θ (Θ') is a vector $\theta \in \mathbb{R}^J$ ($\theta' \in \mathbb{R}^K$) of model parameters. Finally, a realization of the random variable $\mathcal{Z}$ is a set $Z \subseteq \{0,1\}^{V \cup E}$ of feasible characteristic functions.

From the conditional independencies enforced by the Bayesian Network (Fig. 4) follows that a probability measure of the conditional probability of characteristic functions x of nodes and y of edges and model parameters θ and θ', given features f and g and given a feasible set Z, factorizes according to

$$\begin{aligned} & p(x, y, \theta, \theta' | f, g, Z) \\ \propto \quad & p(Z|x,y) \cdot \prod_{v \in V} p(x_v | f^v, \theta) \cdot \prod_{j \in J} p(\theta_j) \\ & \cdot \prod_{e \in E} p(y_e | g^e, \theta') \cdot \prod_{k \in K} p(\theta'_k) \ . \end{aligned} \tag{10}$$

In order to constrain the characteristic functions x and y jointly to the feasible set Z, the first term, the probability density of a feasible set Z, given x and y, is defined to be 0 if $(x,y) \notin Z$; It is defined to be positive and constant, otherwise:

$$p(Z|x,y) \propto \begin{cases} 1 & \text{if } (x,y) \in Z \\ 0 & \text{otherwise} \end{cases} \ . \tag{11}$$

The second and third term in Eq. 10 are a probabilistic model for the independent 01-classification of nodes (detections). The fourth and fifth term are a probabilistic model for the independent 01-classification of edges (pairs of detections). Specifically, we consider a linear logistic model and a Gaussian prior with $\sigma \in \mathbb{R}^+$. These are stated below for nodes. The definition for edges is independent and analogous.

$$p(x_v = 1 | f^v, \theta) = \frac{1}{1 + \exp(-\langle \theta, f^v \rangle)} \tag{12}$$

$$p(\theta_j) = \mathcal{N}(0, \sigma^2) \tag{13}$$

Estimation (Learning and Inference). Estimating maximally probable model parameters θ, θ' from training data x, y, f, g requires the solution of two (convex) logistic regression problems, one for nodes and one for edges.

Estimating maximally probable characteristic functions x and y for previously unseen data f, g, given a feasible set Z and given (learned) model parameters θ, θ' amounts to solving the 01-linear problem stated in Lemma 1. This problem specializes to the problems in Definitions 1 and 2 for the respective feasible sets Z and motivates our choice of the parameters c and d.

Lemma 1 *Given a graph $G = (V, E)$, a feasible set Z, feature vectors f, g, and model parameters θ, θ', all as defined above with respect to G, a pair (x, y) with $x \in \{0,1\}^V$ and $y \in \{0,1\}^E$ is maximally probable with respect to the measure defined above if and only if it is a solution of the 01-linear program written below, with $c_v = -\langle \theta, f^v \rangle$ and $d_e = -\langle \theta', g^e \rangle$.*

$$\min_{\substack{x \in \{0,1\}^V \\ y \in \{0,1\}^E}} \quad \sum_{v \in V} c_v x_v + \sum_{e \in E} d_e y_e \tag{14}$$

$$\text{subject to} \quad (x, y) \in Z \tag{15}$$

Certified Optimal Solutions. The Minimum Cost Disjoint Paths Problem has polynomial time complexity. We solve instances of this problem by Branch-and-Cut. The Minimum Cost Subgraph Multicut Problem is NP-hard [4] and APX-hard [7]. In order to solve instances of this problem exactly, we make use of the Branch-and-Cut loop of the closed-source commercial software Gurobi[1] which represents the state of the art in integer linear programming (ILP).

In every iteration of an outer cutting plane loop, we consider a relaxed ILP with the full objective function and a subset of the cycle inequalities (none in the first iteration). In order to solve this relaxed ILP to optimality, in an inner loop, we resort to the general classes of branches and cuts implemented in Gurobi. Once a solution of the relaxed ILP is found, we separate violated cycle inequalities, by breadth-first-search, and add these to the relaxed ILP, thus tightening the relaxation. The procedure stops when all cycle inequalities are satisfied and, thus, the full problem has been solved to optimality.

Heuristic Solutions. Alternatively, we propose a heuristic solution for the unconstrained set partition problem by making use of the Kernighan Lin (KL) algorithm as defined in [13], which uses the KL for the bi-partition problem, also defined in [13], as a subroutine. The procedure starts from an initial decomposition defined, in our case, by the components of the graph containing precisely the edges $e \in E$ for which $d_e > 0$. In every iteration, an attempt is made to strictly improve the current decomposition via a sequence of transformations: In an outer loop, every pair of adjacent components is considered. For any such pair, it is assessed, in an inner loop, whether moving nodes from one set to the other improves the objective value. In every iteration of this inner loop, an optimal move is chosen and saved, together with the difference of the objective value caused by this move. Having ordered all possible moves in this way, the smallest k is chosen such that the first k moves, carried out in order, improve the objective value maximally. If the

[1] Version 6.0, http://www.gurobi.com

improvement is positive, the moves are made and thus, the current decomposition is improved. If the improvement is not positive, the procedure terminates.

4. Tracking Details

In this section, we describe our tracklet hypotheses generation method in Sec. 4.1, definitions of the unary feature f and the pairwise feature g in Sec. 4.2 and further implementation details about the Disjoint Paths and Subgraph Multicut tracking model in Sec. 4.3.

4.1. Tracklet Generation

We start with person detections produced by the Deformable Part Model (DPM) [9]. Instead of using the detections as person hypotheses directly, we generate overlapping tracklet hypotheses by the method proposed in [1]. Let the length of a tracklet be M, the set of all detections in frame t is denoted by $h^t = [h^t_1, \dots, h^t_{nt}]$. Then a tracklet $H = [h^1_{t1}, \dots, h^M_{tM}]$ is optimal given all the detections in M frames if H maximizes the following probability:

$$p(H) = p(h^1_{t1}) \cdot \prod_{k=2}^{M} p(h^k_{tk}) \cdot p(h^k_{tk}, h^{k-1}_{tk-1}). \tag{16}$$

where $p(h^k_{tk})$ denotes the probability of detection h_{tk} being true, and $p(h^k_{tk}, h^{k-1}_{tk-1})$ is the transition probability which models a simple Gaussian position dynamics. In our implementation, $M = 5$ for sequences which are shorter than 300 frames and $M = 9$ for others due to the computation cost.

Overlapping Tracklets. For all the detections in every M consecutive frames, we apply the Viterbi algorithm to maximize Eq. (16) to obtain the optimal sequence of detections - our tracklet hypotheses. We remove the selected detections from the set of detections and maximize Eq. (16) iteratively until all the detections are considered. Our tracklets are obtained in an over-complete fashion in two aspects (1) Non-Maximum Suppression (NMS) is not applied for the detections and (2) we compute overlapping tracklets starting at every frame of the sequence. Each strong detection contributes M times to different tracklets (which have different starting frames). Our overlapping tracklets contain a sufficient number of good ones which is arguably a good basis for a tracking algorithm.

4.2. Unary and Pairwise Features

Each tracklet contains the following information: spatial-temporal location, speed, scale, appearance and confidence (tracklet score). Here, with respect to the detection in the middle frame of a tracklet, we use x and y to denote the tracklet center; t is the frame index; v_x and v_y is the velocity fo the tracklet along x and y coordinate respectively; h and a denotes the scale and appearance of the tracklet; s is the tracklet score. Given two tracklets $(x, y, t, v_x, v_y, h, a, s)$ and $(x', y', t', v'_x, v'_y, h', a', s')$, the unary feature is simply the tracklet score and we define the following auxiliary variables for the pairwise feature:

$$m_1 = x' - x \quad m_2 = v_x(t' - t) \quad m_3 = v'_x(t' - t)$$
$$n_1 = y' - y \quad n_2 = v_y(t' - t) \quad n_3 = v'_y(t' - t)$$

which are all further normalized by $\bar{h}$ where $\bar{h} = \max(h, h')$. The pairwise features are defined as

$$\begin{aligned} g_1 &= |t - t'| & g_4 &= |m_1 - m_2| & g_7 &= |n_1 - n_2| \\ g_2 &= \frac{|h - h'|}{\bar{h}} & g_5 &= |m_2 - m_3| & g_8 &= |n_2 - n_3| \\ g_3 &= D(a, a') & g_6 &= |m_1 - m_3| & g_9 &= |n_1 - n_3| \end{aligned} \tag{17}$$

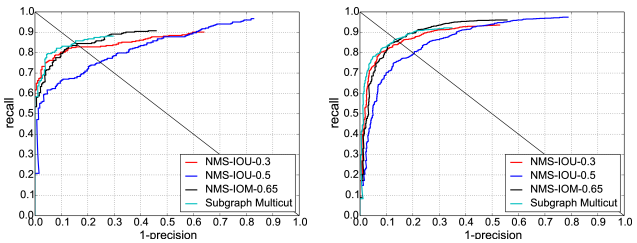
where g_1 denotes temporal distance between two tracklets, g_2 is the normalized scale difference, $g_4 \dots g_9$ describe the relations between speed and temporal-spatial locations of two tracklets, g_3 is the euclidean distance between two tracklets' dColorSIFT features proposed in [26].

We introduce a non-linear mapping from the feature space to the cost space by extending our unary and pairwise features to quadratic and exponential terms. Unary feature f^v is extended as $(f_1, f_1^2, e^{(-f_1)})$ and pairwise feature g^e is $(g_1, \dots, g_9, g_1^2 \dots g_9^2, e^{(-g_1)} \dots e^{(-g_9)})$.

4.3. Further Details

NMS for the Disjoint Paths Model. The above technical details are identical for the Subgraph Multicut model and the Disjoint Paths model. However, pre-selection of tracklet hypotheses (tracklet NMS) and post-processing of the final tracks (tracks NMS) are necessary steps for the Disjoint Paths model. In our implementation, these two steps are performed in a standard way: the tracklet NMS is performed in full analogy to a greedy NMS for people detection, with respect to the middle frame of the tracklet. For the NMS of the final tracks, the suppression is performed on the overlapping fragment of each track, which means that if the optimal track of a target is obtained, it suppresses all other suboptimal redundant tracks of the target. The extensive evaluation described in Sec. 6.3 shows that our Disjoint Paths model with the standard NMS technics achieves results which are on par with state-of-the-art, indicating that our Disjoint Paths model is a good baseline to conduct valid analyses and comparisons.

Tracks from the Subgraph Multicut Model. While the Disjoint Paths model directly produces tracks for each target by its definition, our Subgraph Multicut model produces a connected component for each target. Generating tracks from connected components is straight-forward: in each frame, for all the hypotheses which belong to the same component, we obtain representative location x,y and scale s in this frame by averaging all the connected hypotheses weighted by their probability defined in Eq.12. The final track of the target is a smoothed trajectory which links the representative hypotheses across all the frames.



(a) TUD-Campus (b) TUD-Crossing

Figure 5. Performance comparison between the Subgraph Multicut model and local greedy NMS methods.

5. Subgraph Multicut for Detection NMS

Our Subgraph Multicut model has the property of jointly addressing the problem of spatial (within-frame) *and* temporal (across-frame) associations. Non-Maximum Suppression (NMS) for detections in single frames, on the other hand, is a spatial association problem. Therefore, it is straight-forward to apply the Subgraph Multicut model to the NMS problem.

In full analogy to our Subgraph Multicut tracking model, for detection hypotheses, the unary feature is the detection score, the pairwise feature is derived from Eq.17. Given that we have $|t' - t| = 0$, the pairwise feature g is defined as $(|h - h'|, |m_1|, |n_1|)$. The final representation of each target is obtained by weighted averaging of all the detections which are associated together.

Results. We evaluate the Subgraph Multicut NMS method on the TUD-Campus and TUD-Crossing datasets [1], which are challenging for pedestrian detection due to partial occlusions. Given the detections obtained from DPM [9], two state-of-the-art NMS methods are used as baselines. *(1)* NMS intersection over union (NMS-IOU) [10] and *(2)* NMS intersection over minimum (NMS-IOM) [8].

In Fig. 5(a), NMS-IOU with threshold 0.3 gets better precision and NMS-IOU with threshold 0.5 obtains higher recall. For NMS-IOM, we use threshold 0.65 which is the best setting for this method [8]. Our Subgraph Multicut model is able to improve the performance comparing to all the NMS methods evaluated here. In Fig. 5(b), our Subgraph Multicut model is on par with NMS-IOM at equal-error-rate, and outperforms others at high precision. The parameters used in the Subgraph Multicut NMS model for TUD-Crossing are learned from TUD-Campus and vice versa.

Summary. Only spatial relations between two detections are considered in the current pairwise feature, which is a fair comparison between our Subgraph Multicut model and local greedy NMS methods. Our model performs better because *(1)* associations of detections are obtained in a globally optimal fashion and *(2)* different spatial relations between two detections are learned for associations. Note that, our Subgraph Multicut model has the potential of leveraging other information in the pairwise term, e.g., appearance and prior knowledge about object layout.

6. Tracking Evaluation

We evaluate the performance of the proposed Subgraph Multicut model on three publicly available sequences: TUD-Campus, TUD-Crossing [1] and ParkingLot [24]. We perform extensive experiments and analysis on TUD-Crossing and present quantitative, superior results compared to other competitive methods on three sequences.

We use standard CLEAR MOT as evaluation metrics that include recall (Rcll), precision (Prcsn), multiple object tracking accuracy (MOTA), and multiple object tracking precision (MOTP) [5]. MOTA is a cumulative measure that combines missed targets (FN), false alarms (FP), and identity switches (IDs). MOTP measures overlap between the ground truth and estimated trajectory. We also report mostly tracked (MT), partly tracked (PT), mostly lost (ML) and fragmentation (FM) for measuring track completeness.

We analyze the performance of the proposed methods in four aspects. *(1)* We compare the exact integer linear programming (ILP) solver and the heuristic Kernighan Lin (KL) solver in terms of run time and MOTA. For the same tracklet hypotheses, KL obtains nearly the same MOTA compared to ILP, but much faster (Sec. 6.1). *(2)* We evaluate the influence of long-term associations both for the Disjoint Paths model and the Subgraph Multicut model. By associating tracklet hypotheses that are temporally far from each other (up to 30 frames), MOTA is improved for both models and the number of ID switches is substantially reduced (Sec. 6.2). *(3)* We provide an in-depth analysis of the Disjoint Paths model and the Subgraph Mutlicut model. Extensive experimental results indicate that the properties of leveraging multiple hypotheses per target within and across frames facilitate the Subgraph MultiCut model to obtain a more robust association (Sec. 6.3). *(4)* We show that our Subgraph Multicut model obtains superior results over the state-of-the-art (Sec. 6.4).

Training sequences. For the Subgraph Multicut and Disjoint Paths models, we need training data to learn the model parameters θ and θ' (Sec 3.3). In our experiments, we use the parameters learned from TUD-Crossing for the experiments on TUD-Campus and ParkingLot. For TUD-Crossing, we use the parameters learned on TUD-Campus.

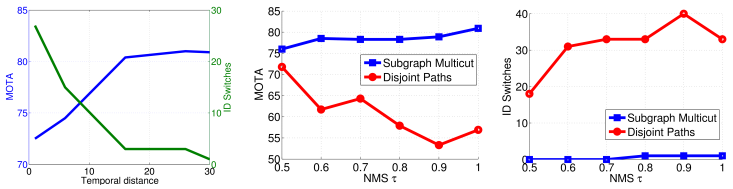
6.1. Solver Comparison

We start by comparing the performance of the Subgraph Multicut model optimized by the KL and ILP solvers on TUD-Campus. In this experiment we vary the number of initial person hypotheses $|V|$ by adjusting the threshold τ of NMS and report tracking performance and convergence speed of each solver. Results are shown in Tab. 6(a).

Setting τ to 0.5 results in 277 tracklet hypotheses and 4835 pairwise terms. Both solvers achieve the same MOTA (79.4%) within comparable runtime (0.86 sec. vs. 0.48 sec.). Increasing τ to 0.7 results in 616 tracklet hypotheses. In this

		KL solver		ILP solver	
$\|V\|$	$\|E\|$	Run time (s)	MOTA	Run time (s)	MOTA
277	4835	0.86	79.4	0.48	79.4
616	35424	1.82	80.8	76.39	83.3
1453	199333	12.49	83.3	79986.01	83.3

(a) Solver comparison



(b) Long-term influence (c) MOTA (d) ID switches

Figure 6. **(a)** Comparison of tracking performance and convergence speed of KL and ILP solvers on TUD-Campus; **(b)** Long-term association for the Subgraph Multicut model on TUD-Crossing; MOTA **(c)** and ID switches **(d)** comparison for the Subgraph Multicut model and the Disjoint Paths model on TUD-Crossing.

regime ILP achieves better MOTA, but is 40 times slower than KL. Omitting NMS further increases the number of tracklet hypotheses to 1453. KL achieves the same MOTA as ILP in 12.5 seconds, compared to ILP that takes 22 hours. These results indicate that the KL algorithm achieves results comparable to ILP but significantly faster. For efficiency, we apply the KL solver for the Subgraph Multicut Problem in the following experiments. Note that reducing amount of NMS leads to improved performance, likely because NMS makes local decisions on the level of individual frames that are potentially suboptimal in the context of global optimization.

6.2. Long-Term Association

Next, we evaluate the robustness of the Disjoint Paths and Subgraph Multicut models with respect to long-term associations between hypotheses. To that end we apply both models to graphs that connect each tracklet hypothesis to every other tracklet hypothesis within a neighborhood of 30 frames. Intuitively enabling such long-range connectivity should be helpful for misdetection and occlusion cases that otherwise result in ID switches. We conduct this experiment on TUD-Crossing that has a large number of people that frequently occlude each other.

The baseline model in this comparison corresponds to a graph in which each hypothesis is connected to hypotheses in the next and previous frames only. This baseline model for the Disjoint Paths formulation results in 66.8% MOTA. Adding long-range connections improves the performance of the Disjoint Paths model to 71.8% and reduces the number of ID switches from 34 to 18. The results for the Subgraph Multicut model are shown in Fig. 6(b). The performance improves from 72.5% to 80.9% MOTA, and the number of ID switches is reduced from 27 to 1. This result indicates the importance of long-term associations across frames which the Subgraph Multicut model can leverage.

6.3. Subgraph Multicut vs. Disjoint Paths Models

The Disjoint Paths model achieves results on par with the state-of-the-art, as shown in Tab. 2 and Tab. 3 (71.8% MOTA for TUD-Crossing, 86.6% MOTA for ParkingLot). This suggests that the Disjoint Paths model is a good baseline to conduct a detailed analysis. Note that both models are based on the same set of tracklet hypotheses as well as unary and pairwise terms as detailed above.

An important difference between the Disjoint Paths and Subgraph Multicut models is that the Disjoint path model imposes mutual exclusion constraints when connecting tracklet hypotheses. This is in contrast to the Subgraph Multicut model that allows each tracklet hypothesis to associate with an appropriate number of tracklet hypotheses in the same and other frames resulting in more robust associations.

When the tracklet hypotheses are pre-selected by performing NMS, as shown in Fig 6(c), with $\tau = 0.5$, the Disjoint Paths model performs best. However, the model is sensitive to the NMS threshold. Decreasing the level of NMS or skipping the NMS step altogether results in a substantial performance drop for MOTA (from 71.8% to 56.9%). Additionally, the number of ID switches increases from 18 to 33 (red line in Fig 6(d)). This is an inherent limitation of the Disjoint Paths model resulting from the mutual exclusion constraints. This and similar models require both preprocessing of person hypotheses (detection/tracklets-NMS) as well as post-processing of tracks (tracks-NMS) to obtain good performance.

In contrast, decreasing the level of NMS improves the performance of the Subgraph Multicut model constantly from 76.0% MOTA to 80.9% (blue curve in Fig. 6(c)). This is due to the ability of the Subgraph Multicut model to associate hypotheses jointly across space and time, thereby aggregating information about the targets which results in more robust associations over the whole sequence.

With respect to ID switches, the Subgraph Multicut model constantly outperforms the Disjoint Paths model for all NMS thresholds by a large margin as shown in Fig. 6(d). This performance difference is explained by the fact that finding a disjoint path for a target precisely in a graph across all frames is a substantially harder problem than clustering nodes that correspond to the same target.

6.4. Comparison to the State-of-the-art

We now compare our approach to recent approaches on TUD-Crossing, TUD-Campus, and "Parking Lot" datasets. TUD-Campus and TUD-Crossing show people from the cam-

[2]Result on frame 346-989 using the parameters trained on frame 1-345.

Method	Rcll	Prcsn	FAR	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP	MOTAL
Pirsiavash et al. [14]	66.6	95.5	0.15	8	3	4	1	11	118	10	13	60.6	78.2	63.2
Breitenstein et al. [6]	-	-	-	-	-	-	-	-	-	2	-	73.3	67.0	-
Segal et al. [15]	-	-	-	-	5	-	-	-	-	0	3	82.0	74.0	-
Subgraph Multicut	83.8	99.3	0.03	8	5	2	1	2	58	0	1	83.3	76.9	83.3

Table 1. Tracking performance on TUD-Campus.

Method	Rcll	Prcsn	FAR	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP	MOTAL
Pirsiavash et al. [14]	73.9	95.0	0.21	13	6	7	0	43	286	50	42	65.5	76.8	69.9
Breitenstein et al. [6]	-	-	-	-	-	-	-	-	-	2	-	84.3	71.0	-
Segal et al. [15]	-	-	-	-	7	-	-	-	-	2	12	74.0	76.0	-
Tang et al. [18]	82.7	93.9	-	-	7	-	1	-	-	-	-	76.0	78.6	-
Zamir et al. [24]	88.4	96.2	0.19	13	9	4	0	38	128	2	5	84.8	74.5	84.9
Disjoint Paths	74.5	98.6	0.06	13	6	7	0	12	281	18	18	71.8	77.7	73.3
Subgraph Multicut	82.0	98.8	0.05	13	8	3	2	11	198	1	1	80.9	78.0	81.0

Table 2. Tracking performance on TUD-Crossing.

Method	Rcll	Prcsn	FAR	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP	MOTAL
Pirsiavash et al. [14]	69.4	97.8	0.16	14	2	1	-	39	754	52	60	65.7	75.3	-
Shu et al. [17]	-	-	-	-	-	-	-	-	-	-	-	79.3	74.1	-
Wen et al. [21]	90.8	98.4	0.16	14	11	-	0	39	227	21	23	88.4	81.9	-
Tang et al. [18]	91.0	98.5	-	-	-	-	-	-	-	-	-	89.3	77.7	-
Zamir et al. [24]	85.3	98.2	0.02	14	-	-	-	-	-	-	-	90.4	74.1	-
Disjoint Paths	89.0	98.5	0.14	14	11	3	0	34	272	25	24	86.6	76.7	87.5
Subgraph MultiCut	96.1	95.4	0.45	14	13	1	0	113	95	5	18	91.4	77.4	91.5
Subgraph MultiCut[2]	96.9	97.0	0.37	14	13	1	0	46	47	1	6	93.8	78.3	93.9

Table 3. Tracking performance on ParkingLot.

era at low viewpoint resulting in frequent occlusions, and TUD-Campus also includes substantial variation in people scale. The Parking Lot sequence is captured in a surveillance setting with a camera elevated above the ground that results in pedestrians' walking patterns substantially different compared to TUD-Campus and TUD-Crossing. Tables 1, 2 and 3 show results for TUD-Campus, TUD-Crossing and "Parking Lot" respectively. The ground truth tracks used in all experiments are from [3]. Our Subgraph Multicut model achieves state-of-the-art MOTA results overall. In particular the number of ID switches is substantially improved compared to other approaches. [24] also reports tracking results on TUD-Crossing. Based on the tracking results they provided to us, we obtain 84.8% MOTA and 2 ID Switches on the ground truth from [3].

On the ParkingLot sequence, the Disjoint Paths model again performs on par with state-of-the-art (86.6% MOTA), suggesting that it is a good baseline to conduct comparison and analysis. With our Subgraph Multicut Model and parameters learned from TUD-Crossing, we achieve 91.4% MOTA and 5 ID Switches. To evaluate sensitivity of our model to particular training set we split the "Parking Lot" sequence into to training(1-345) and testing(346-989) sequences, and retrain parameters of our pairwise and unary terms on the training subset. This results in slight improvement in performance compared to the model with parameters trained on TUD-Crossing. We obtain 93.8% MOTA and ID switches are reduced to 1, as shown in the last row of Tab. 3.

7. Conclusion

In this work, we propose to formulate multi-target tracking as a Minimum Cost Subgraph Multicut Problem. In contrast to the Minimum Cost Disjoint Paths formulation, which selects a set of disjoint paths as tracks and which is similar in spirit to many state-of-the-art methods, the Subgraph Multicut model selects and clusters all suitable hypotheses for each target jointly in space and time. Experiments show that our Subgraph Multicut model improves the multi-target tracking performance on several datasets underlying both the usefulness as well as the applicability of the proposed formulation. We also show initial results to the classic problem of Non-Maximum Suppression that without any changes achieves performance on par with top-performing NMS-schemes. In the future we will explore more powerful unary and pairwise terms to further improve NMS and tracking performance.

Acknowledgements. This work has been supported by the Max Planck Center for Visual Computing and Communication.

References

[1] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *CVPR 2008*. 1, 2, 5, 6

[2] M. Andriluka, S. Roth, and B. Schiele. Monocular 3d pose estimation and tracking by detection. In *CVPR*, June 2010. 2

[3] A. Andriyenko, K. Schindler, and S. Roth. Discrete-continuous optimization for multi-target tracking. In *CVPR 2012*. 2, 8

[4] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Machine Learning*, 56(1–3):89–113, 2004. 4

[5] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *Image and Video Processing*, 2008(1):1–10, May 2008. 6

[6] M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Online multiperson tracking-by-detection from a single uncalibrated camera. *IEEE T. Pattern Anal. Mach. Intell.*, 33(9):1820–1833, 2011. 8

[7] E. D. Demaine, D. Emanuel, A. Fiat, and N. Immorlica. Correlation clustering in general weighted graphs. *Theoretical Computer Science*, 361(2–3):172–187, 2006. 4

[8] P. Dollár, Z. Tu, P. Perona, and S. Belongie. Integral channel features. In *BMVC*, 2009. 6

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE T. Pattern Anal. Mach. Intell.*, 32(9):1627–1645, 2010. 5, 6

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 6

[11] R. Henschel, L. Leal-Taixe, and B. Rosenhahn. Efficient multiple people tracking using minimum cost arborescences. In *GCPR*, 2014. 2

[12] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. In *ECCV 2008*. 2

[13] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell Systems Technical Journal*, 49:291–307, 1970. 2, 4

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

[15] A. V. Segal and I. Reid. Latent data association: Bayesian model selection for multi-target tracking. In *ICCV*, 2013. 1, 2, 8

[16] H. B. Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Tracking multiple people under global appearance constraints. In *ICCV 2011*. 2

[17] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah. Part-based multiple-person tracking with partial occlusion handling. In *CVPR*, 2012. 8

[18] S. Tang, M. Andriluka, A. Milan, K. Schindler, S. Roth, and B. Schiele. Learning people detectors for tracking in crowded scenes. In *ICCV*, 2013. 8

[19] S. Tang, M. Andriluka, and B. Schiele. Detection and tracking of occluded people. *International Journal of Computer Vision (IJCV)*, 2014. 2

[20] X. Wang, E. Turetken, F. Fleuret, and P. Fua. Tracking interacting objects optimally using integer programming. In *ECCV*, 2014. 2

[21] L. Wen, W. Li, J. Yan, Z. Lei, D. Yi, and S. Li. Multiple target tracking based on undirected hierarchical relation hypergraph. In *CVPR)*, 2014. 8

[22] C. Wojek, S. Roth, K. Schindler, and B. Schiele. Monocular 3d scene modeling and inference: Understanding multi-object traffic scenes. In *ECCV*, 2010. 2

[23] C. Wojek, S. Walk, S. Roth, K. Schindler, and B. Schiele. Monocular visual scene understanding: Understanding multi-object traffic scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2013. 2

[24] A. R. Zamir, A. Dehghan, and M. Shah. GMCP-Tracker: Global multi-object tracking using generalized minimum clique graphs. In *ECCV 2012*. 1, 2, 6, 8

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

[26] R. Zhao, W. Ouyang, and X. Wang. Unsupervised salience learning for person re-identification. In *CVPR*, 2013. 5