LMGP: Lifted Multicut Meets Geometry Projections for Multi-Camera Multi-Object Tracking

Duy M. H. Nguyen[1,4], Roberto Henschel[2], Bodo Rosenhahn[2], Daniel Sonntag[3,4], Paul Swoboda[1]

[1] Max Planck Institute for Informatics, Saarland Informatics Campus, [2] Institute for Information Processing, Leibniz University Hannover
[3] Oldenburg University [4] German Research Center for Artificial Intelligence, Saarbrücken

Abstract

Multi-Camera Multi-Object Tracking is currently drawing attention in the computer vision field due to its superior performance in real-world applications such as video surveillance with crowded scenes or in wide spaces. In this work, we propose a mathematically elegant multi-camera multiple object tracking approach based on a spatial-temporal lifted multicut formulation. Our model utilizes state-of-the-art tracklets produced by single-camera trackers as proposals. As these tracklets may contain ID-Switch errors, we refine them through a novel pre-clustering obtained from 3D geometry projections. As a result, we derive a better tracking graph without ID switches and more precise affinity costs for the data association phase. Tracklets are then matched to multi-camera trajectories by solving a global lifted multicut formulation that incorporates short and long-range temporal interactions on tracklets located in the same camera as well as inter-camera ones. Experimental results on the WildTrack dataset yield near-perfect performance, outperforming state-of-the-art trackers on Campus while being on par on the PETS-09 dataset. We will release our implementations at this link https://github.com/nhmduy/LMGP.

1. Introduction

Multiple object tracking (MOT), i.e., extracting motions of objects moving through a scene, is a fundamental primitive for high-level understanding information in videos. The most common approach to MOT is the tracking-by-assignment paradigm, in which first detection boxes are computed for the objects of interest in each timeframe, and second, a data association is performed by linking detections of the same objects to each other. In the most popular setting, a single camera faces a scene and the data association links detections in different timeframes to each other [1, 3, 9, 43]. However, even though a large body of research has been devoted to MOT with a single camera, large and
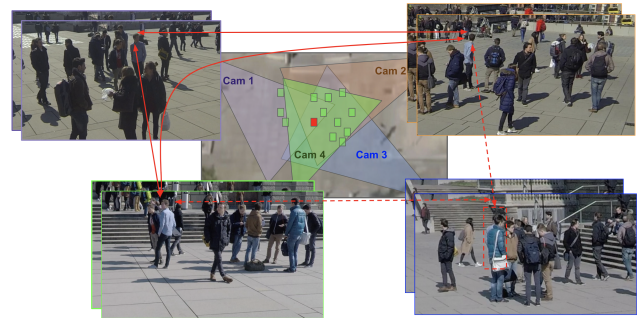

Figure 1. Multi-camera tracking with four overlapping cameras. A target object (red rectangle) is occluded at Cam 3 but is still observed at Cam 1, Cam 2, and Cam 4. Taking this correspondence into account (red arrow), we can recover a missing bounding box at Cam 3 (red dashed arrow).

crowded scenes still cannot be tracked faithfully, and errors occur mainly in the data association step. These errors most often are caused by partial visibility (or even occlusion) and indistinguishability of objects.

One possibility of improving performance has been to use multiple cameras facing the same scene but from different angles (Figure 1). In this setting, partial visibility and indistinguishability are less severe since an object may be occluded in a single camera but may still be fully observed by another camera. Leveraging this property, recent papers have pursued two principal approaches: *single view-based* and *centralized representation* methods. In the first strategy [16, 28, 33, 45], a two-step procedure is followed: 1) generating local tracklets of all the targets within each camera; 2) matching local tracklets that belong to the same target across cameras through computing affinity costs and using a global optimization framework. While this framework brings benefits via a reduced hypothesis space and allows to design motion-based features, its main drawbacks lie in ID-Switch errors contained inside local tracklets, i.e., detections of distinct objects are grouped into the same trajectory (Figure 3a). As a result, these errors will propagate throughout the tracking graph, affecting the total performance. *The centralized representation* approach [47, 49]

on the other hand, is not plagued by such obstacles since each node in the tracking graph is an occupancy map (not a tracklet), which is estimated from all detections at each timeframe. Unfortunately, the cost of the data association step is increased due to a huge state space of variables and integrating advances from single-camera methods is more complicated.

In this work, we propose a method that follows the *single view-based approach* but integrates concepts from the centralized representation paradigm. Our motivation for this design choice is to harness the great progress made in single-camera tracking while at the same time effectively addressing limitations encountered in prior studies such as ID-Switch errors by exploiting *centralized representation* ideas through our novel pre-clustering step. Specifically, the corresponding images from the pre-clustering step (i.e. our occupancy map) allow us to break up initial tracklets generated by single-camera trackers at ID-Switch errors and establish precise affinity costs for both temporal and spatial affinities (Subsection 4.5). On top of that, a novel spatial-temporal optimization model for the data association is employed, which takes into account both short- and long-range temporal interactions of objects detected by a single camera as well as spatial interactions between cameras in a single framework (Table 4). The experimental findings show that, given the right conditions, with a multiple-camera environment and precise boundary detections, our method leads to a nearly optimal solution for multiple object tracking using multiple cameras (Table 1).

Contributions Our main contributions can be summarized as follows. First, we introduce a new pre-clustering algorithm driven by 3D geometry projections to group detections at each timestep across cameras. This effectively eliminates tracklet errors from single-camera trackers and provides highly accurate affinity costs for the data association step. Second, we propose a novel spatial-temporal lifted multicut formulation for the multi-camera setting, jointly optimizing both intra- and inter-camera as well as short- and long-range interaction in a single global formulation. Finally, we obtain nearly perfect performance on the large-scale WILDTRACK [7] dataset, outperform state-of-the-art on Campus [44] and are on par with the PETS-09 [12] dataset.

2. Related Work

There has been a large body of research on single-camera MOT. These methods focus on the data association step, for which the (lifted) multicut problem [38, 39, 40], the lifted disjoint paths problem [21, 22], maximum clique [10, 48], multigraph-matching [23], and binary quadratic optimization [17, 18, 19, 41] was used. Another area is building end-to-end differentiable frameworks for both detector and data association [1, 9, 43, 50, 53]. For an exhaustive survey of MOT we refer to [11].

Multi-camera MOT has recently received increasing attention. The related work can be categorized into two different approaches:

Single View-Based Methods [44] propose a Hierarchical Composition of Tracklet (HCT) framework to match local tracklets by utilizing multiple cues of objects such as appearances and their 3D positions. In [45] the matching problem is solved using a Bayesian formulation with a Spatio-Temporal Parsing (STP)-based tracking graph to prune matching candidates by exploiting semantic attribute targets. Similarly, [42] formulate a dense sub-hypergraph search (SVTH) on the space-time-view graph using a sampling-based approach. Recent approaches include a semi-online Multi-Label Markov Random Field (MLMRF) method [28], where the ensuing optimization problem over single detections is solved through alpha-expansion [4] and a non-negative matrix factorization approach (TRACTA) for grouping tracklets across cameras [16]. In another direction, DyGLIP [33] formulates the data association problem for multi-camera as a link prediction on a graph whose nodes are tracklets. While these methods have demonstrated promising performance in some datasets, they are affected by ID-switch errors in the tracklet proposal generation, especially in cluttered or crowded scenes such as [7].

Centralized Representation-Based Methods To estimate the occupancy map (2D) or occupancy volume (3D), occlusion relationships among different detections have been explicitly modeled. The works in [14, 31] construct occupancy maps by using the foreground map after a background subtraction step. Ground plane homographs are another technique introduced in [26], that generates a voting map from the foreground pixels in each view for occupancy map construction. Toward the probabilistic approach [13], GMLP [32] jointly uses CNNs and Conditional Random Fields to model explicitly an occupancy volume map given detections estimated from multiple cameras. More recently [47] (DMCT) propose deep learning to directly compute the occupancy volume by fusing feature maps extracted from CNNs at multi-camera views.

Differences w.r.t. Previous Work Our work, denoted as LMGP, is at the intersection of single-view-based and centralized representation methods. We use single-camera tracklets but improve them by eliminating ID-switch errors using centralized representation concepts with multi-camera information derived from a novel 3D geometry-based occupancy map. This factor sets us apart from competing approaches (Table 5 Appendix). Moreover, we are the first to formulate a global lifted multicut method for multi-camera settings. So far, lifted multicut was only applied to the single-camera setting [38, 39, 40]. We argue

that our model is an elegant abstraction capturing the full range of interactions in multi-camera MOT.

3. Method

Our tracking pipeline is illustrated in Figure 2. Below we describe each of its steps in detail, i.e., pre-clustering for removing ID-switch errors and improving the subsequent affinity cost computation that is utilized in the global lifted multicut problem for computing multi-camera trajectories.

Notation Before describing each part of our approach in detail, we introduce notation used throughout the paper. Let B be the set of detections and $B^{t,j}$ the detections at timestep t observed by camera j. Each detection $b \in B$ is observed by camera $\mathrm{cam}(b)$ and in timeframe $\mathrm{time}(b)$. Each single camera tracklet τ consists of a set of bounding boxes at specific timepoints, i.e. $\tau = (b_1^\tau, b_2^\tau, ..., b_{|\tau|}^\tau)$, where b_l^τ is the l-th detection of trajectory τ. Tracklets only contain detections from a single camera, i.e. $\mathrm{cam}(b_1^\tau) = ... = \mathrm{cam}(b_{|\tau|}^\tau)$. We extend the functions cam and time to tracklets by $\mathrm{cam}(\tau) = \mathrm{cam}(b_1)$ and $\mathrm{time}(\tau) = \{\mathrm{time}(b_1), \mathrm{time}(b_2), \ldots, \mathrm{time}(b_{|\tau|})\}$. Detections of two tracklets that cover the same timepoint are denoted as

$$O(\tau, \tau') = \{(b, b') \in \tau \times \tau' : \mathrm{time}(b) = \mathrm{time}(b')\} . \quad (1)$$

We denote by f a feature extractor that, given a bounding box b, produces an embedding vector $f(b)$ representing its appearance features. h denotes a map that takes a bounding box and computes the 3D coordinates of the foot point (center of bottom edge) on the ground plane ($z = 0$) [15] (Section B Appendix).

3.1. 3D Geometry Based Pre-Clustering

The pre-clustering step aims at bringing into correspondence detections of the same object observed by different cameras at each timeframe (Figure 2-b). This enables us to overcome occlusions observed by a single camera. In particular, if some object is occluded, we will be able to continue tracking the same object at different views (Figure 1). Unlike prior works [14, 31, 47] that applied foreground subtraction or fusing image features from multiple cameras, our algorithm exploits 3D geometry constraints of detection projections of the same object. In particular, we project the bottom edge center of each bounding box to obtain its 3D coordinates (ground point) via map h. Two detections observed by different cameras potentially belong to the same person if, after transformation to 3D, the Euclidean distance of the two ground points is less than the diameter of a typical person, which is approximately the human width average (Figure 6-a Appendix).

The pre-clustering works as follows: for each detection b, we consider the set of nearby detections $B^{t,j}(b) = \{b' \in$

$B^{t,j} : \mathrm{dist}(h(b), h(b')) \leq r\}$ observed by the same camera $j = \mathrm{cam}(b)$ at the same timepoint $t = \mathrm{time}(b)$ with $\mathrm{dist}(,)$ and r being Euclidean distance and radius to scan respectively. Likewise for camera $j' \neq \mathrm{cam}(b)$, we consider the set of detections $B^{t,j'}(b)$ observed by camera j' close to the 3D-position $h(b)$. We next compute a matching between detections of $B^{t,j}(b)$ and $B^{t,j'}(b)$ via a linear assignment problem [6] with costs being the Euclidean distance. If b is matched to a detection $b' \in B^{t,j'}(b)$ through the matching between $B^{t,j}(b)$ and $B^{t,j'}(b)$, and vice versa, if b' is matched to $b \in B^{t,j}(b')$ through another matching between $B^{t,j'}(b')$ and $B^{t,j}(b')$, we record that match as it represents a confident connection. We denote the resulting cluster for each detection b as C_b. The whole algorithm is detailed in Algorithm 1 in the Appendix.

Visible Detection Clustering Let $b \in B$ be a detection and C_b be its cluster obtained after pre-clustering. Since our algorithm uses only geometric coordinates of bounding boxes, the cluster C_b can include both visible and occluded detections (Figure 3-b). Let h_j be the 3D camera position of camera j (Equation (19) in Section B Appendix). First, given the detection b, we compute a detection $\mathrm{visible}(b)$ nearest to the camera by

$$\mathrm{visible}(b) = \begin{cases} \arg\min_{b'} \mathrm{dist}(h(b'), h_j) \\ \text{s.t. } b' \in B^{\mathrm{time}(b), \mathrm{cam}(b)} : \mathrm{IoU}(b', b) \geq 0.6 \end{cases} \quad (2)$$

and then use it to refine the pre-cluster to contain only visible detections by

$$C_b' = \{b' \in C_b : b' = \mathrm{visible}(b')\} . \quad (3)$$

3.2. Spatial-Temporal Tracking Graph

We formulate a global spatial-temporal tracking graph $G = (V, E)$, where each node $v \in V$ corresponds to a tracklet τ in a single camera and edges represent data associations between tracklets across space and time (Figure 2-c). A trajectory output will correspond to a cluster of nodes in the tracking graph G. To benefit from current advances in single-camera MOT, each node (tracklet) at each camera is derived from a state-of-the-art tracker. We use Center-Track [53] in our experiments, but it can be replaced by other trackers.

While recent works [16, 28, 45] directly compute affinity costs and solve the data association on the graph with nodes generated by single-camera trackers, we further correct ID-Switch errors in tracklet proposals (Figure 3-a). The ID switches in the original tracklets severely harm the total performance, especially in crowded or cluttered scenes. To this end, we leverage bounding box correspondences from the pre-clustering and conduct the following steps.

Feature Extension on Detection Clusters Given a pair of detections $b, b' \in B$ at different timepoints of poten-
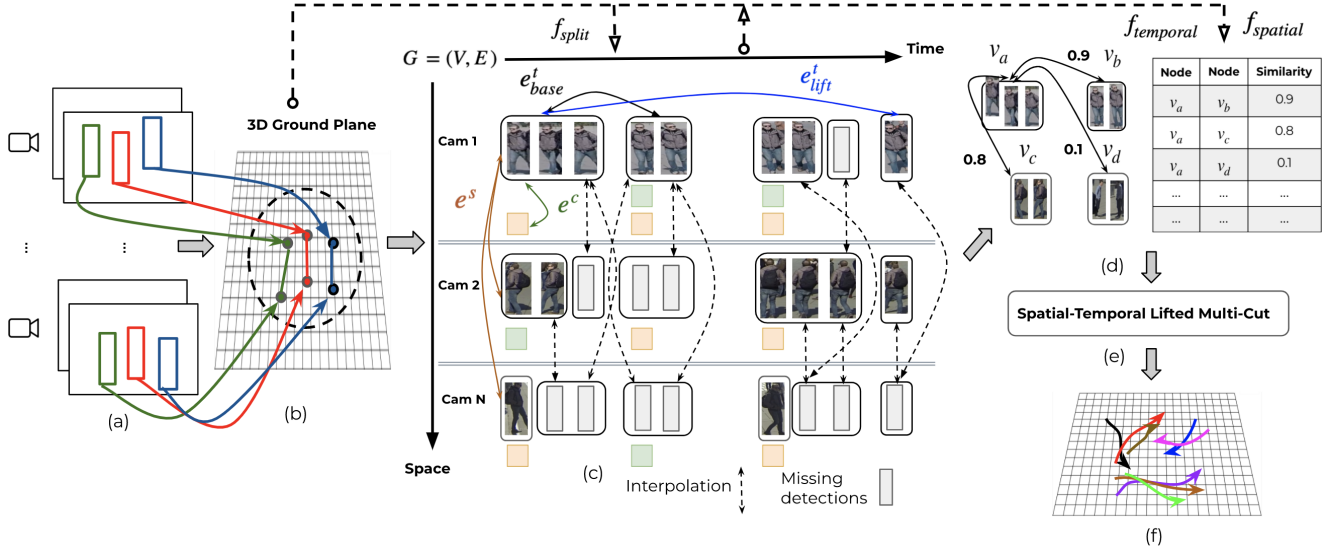
Figure 2. Illustration of our LMGP framework. (a) Input bounding boxes are given at each camera and each timepoint. (b) Bounding boxes observed from different cameras at the same time point are preliminary brought into correspondence through our 3D geometry based pre-clustering step. (c) A spatial-temporal tracking graph is constructed. Nodes in the graph correspond to tracklets generated by a single-camera tracker that are split up at likely ID-switch positions based on features estimated from (b) step via an ID-Error predictor f_{split}. Edges correspond to possibly associations between tracklets with temporal edges e^t (base and lifted for short and long range interaction in the same camera), spatial edges e^s (different cameras, overlapping timeframes), and trajectory constraint edges e^c (same camera, overlapping timeframes). (d) Association costs between nodes are computed between pre-processed tracklets using two networks $f_{\text{temporal}}, f_{\text{spatial}}$ for intra- and inter-camera edges in the tracking graph. (e) Tracklets in the tracking graph (c) are clustered together into trajectories via our lifted multicut optimization problem using affinity costs in (d). (f) 3D coordinates of trajectories from (e) are generated.

tially the same object and an embedding feature f (we use DG-Net [52]), we aim to obtain robust association features by considering relations between detections in the visible cluster C'_b (Equation 3). To this end, we first compute for all pairs of detections in $(\bar{b}, \bar{b}') \in C'_b \times C'_{b'}$ their cosine similarity w.r.t. features extracted by f, that is $D_{\bar{b},\bar{b}'} = \langle f(\bar{b}), f(\bar{b}') \rangle$. Next, we solve a linear assignment problem between C'_b and $C'_{b'}$ with costs $D_{\bar{b},\bar{b}'}$ for each pair $(\bar{b}, \bar{b}') \in C'_b \times C'_{b'}$. Let the set of matches be $M = \{(\bar{b}, \bar{b}') \in C'_b \times C'_{b'} : \bar{b} \text{ and } \bar{b}' \text{ are matched}\}$. On the matches we estimate the following statistical similarities:

$$c_{b,b'}^{\text{best}} = \min_{(\bar{b},\bar{b}') \in C'_b \times C'_{b'}} D_{\bar{b},\bar{b}'}, \quad c_{b,b'}^{\text{min}} = \min_{(\bar{b},\bar{b}') \in M} D_{\bar{b},\bar{b}'},$$

$$c_{b,b'}^{\text{max}} = \max_{(\bar{b},\bar{b}') \in M} D_{\bar{b},\bar{b}'}, \quad c_{b,b'}^{\text{mean}} = \frac{\sum_{(\bar{b},\bar{b}') \in M} D_{\bar{b},\bar{b}'}}{|M|},$$

$$c_{b,b'}^{\text{var}} = \sum_{(\bar{b},\bar{b}') \in M} (D_{\bar{b},\bar{b}'} - c_{b,b'}^{\text{mean}})^2. \quad (4)$$

Splitting Tracklets We now construct a network f_{split} (see Appendix D.2 for the architecture details) for correcting ID-Switch errors. Specifically, for each tracklet τ, f_{split} scans over all consecutive detections $(b, b') \subset \tau$ (Figure 3-a), takes their respective similarity values from Equations 4 using the visible detections $C'_b, C'_{b'}$ (Figure 3-b) and returns

a probability score indicating whether or not they belong to the same tracklet. τ is split into sub-tracklets at the predicted ID-switch error positions that become new nodes in our spatial-temporal tracking graph (Figure 3-c).
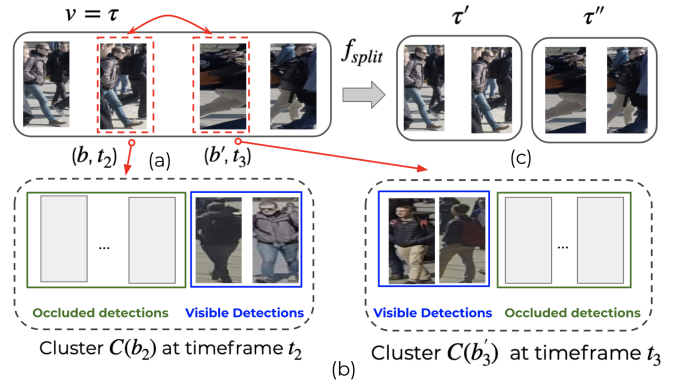

Figure 3. (a) A tracklet (node) τ with ID-Switch error (two red dashed rectangles) in an initial tracking graph, (b) Using visible clusters of detections at two consecutive frames, we can cut at the error positions, (c) Two new sub-tracklets τ', τ'' of distinct objects are generated.

3.3. Learning Affinities with Multi-Camera Setting

Given the tracking graph $G = (V, E)$ after pre-processing for ID-Switch error removal, we compute for

pairs of tracklets $\tau = (b_1, \ldots, b_{|\tau|})$ and $\tau' = (b'_1, \ldots, b'_{|\tau'|})$ affinities representing the probability that both τ and τ' track the same object. To this end, we consider standard cues based on motion information as in [32, 42, 45]. We additionally propose novel appearance and 3D position-based similarity costs which harness image correspondences from our 3D pre-clustering step.

3.3.1 Temporal Affinites

Let τ and τ' be two tracklets, observed by the same camera and with $\text{time}(b_{|\tau|}) < \text{time}(b'_1)$, i.e. following each other temporally. We compute their similarity based on motion and appearance.

Forward/Backward Motion Affinities Motion extrapolation of τ to subsequent or extrapolation of τ' to previous timeframes can provide evidence of whether the two tracklets belong to the same object. We adopt features from [42, 45] for our settings by using the first/last m frames from each tracklet to estimate an average velocity and then predict the forward velocity $\vec{v}_\tau$ of τ and the backward velocity $\overleftarrow{v}_{\tau'}$ of τ'. The forward $c^{\text{fw,t}}$ and backward $c^{\text{bw,t}}$ affinity are computed by:

$$c^{\text{fw,t}}(\tau, \tau') = \|h(b_{|\tau|}) + \text{disp}(\tau, \tau') - h(b'_1)\|.$$
$$c^{\text{bw,t}}(\tau, \tau') = \|h(b'_1) - \text{disp}(\tau, \tau') - h(b_{|\tau|})\|. \quad (5)$$

where $\text{disp}(\tau, \tau') = \vec{v}_\tau \cdot t(\tau, \tau')$; $t(\tau, \tau') = (\text{time}(b'_1) - \text{time}(b_{|\tau|}))$.

Multi-view Appearance Affinities Tracklets belonging to the same object should share a similar appearance across time and cameras. We measure this by computing

$$c^{\text{app}}_{\text{index}}(\tau, \tau') = \sum_{b \in \tau} \sum_{b' \in \tau'} \frac{c^{b,b'}_{\text{index}}}{|\tau||\tau'|}, \quad (6)$$

where $\text{index} \in \{\text{best}, \text{min}, \text{max}, \text{mean}, \text{std}\}$ and each score $c_{\text{index}}(b, b')$ is computed as in Equation 4. Note that we associate each detection with its visible image correspondences derived from the pre-clustering step in Equations 2, which sets us apart from prior works [28, 32, 42] as both intra- and inter-camera information can be associated simultaneously. An ablation study showing the benefit of this new cost can be found in Table 6 Appendix.

3.3.2 Spatial Affinities

For each pair of tracklets τ and τ' observed by different cameras and overlapping in time, we compute affinities based on similarity of motion and average 3D distance of their projected detections as in [42, 45]. We additionally propose novel pre-clustering agreement based-similarity scores for each timestep on which they overlap.

Forward/Backward Motion Affinities Similar to temporal forward resp. backward affinities, we will also exploit

motion information in the spatial setting. If, for example $\max\{\text{time}(\tau)\} < \max\{\text{time}(\tau')\}$, we let $\vec{v}_\tau$ be the forward velocity of τ. The spatial forward affinity is defined as

$$c^{\text{fw,s}}(\tau, \tau') = \|h(b_{|\tau|}) + \vec{v}_\tau - h(b'_{|\tau|+1})\|. \quad (7)$$

The spatial backward affinity $c^{bw,t}$ is defined analoguously.

Average 3D Distance Affinity Given two tracklets of the same object, the bounding boxes at timepoints covered by both should have a small distance w.r.t. 3D projections. We capture this property by defining the average 3D distance affinity as

$$c^{\text{avg3D}}(\tau, \tau') = \sum_{(b,b') \in O(\tau,\tau')} \frac{\|h(b) - h(b')\|}{|O(\tau, \tau')|}. \quad (8)$$

Pre-clustering Agreement Affinity The pre-clustering of two tracklets of the same object on shared timesteps should coincide. We define a novel metric to quantify this by

$$c^{\text{pc}}(\tau, \tau') = \sum_{(b,b') \in O(\tau,\tau')} \frac{p \cdot \mathbb{1}_{[C_b = C_{b'}]} + (1-p)\mathbb{1}_{[C_b \neq C_{b'}]}}{|O(\tau, \tau')|} \quad (9)$$

for some prior probability p (we choose 0.8). The contributions of Equation (9) in capturing similarity among spatial tracklets can be seen in Table 7 Appendix.

3.4. Lifted Multicut for Multi-Camera Tracking

We first recapitulate the lifted multicut problem and use it to formulate the multi-camera MOT problem. Our formulation extends the (lifted-) multicut works for single-camera MOT [38, 39, 40] to multiple cameras.

Lifted Multicut The multicut optimization problem [8] is to partition nodes of a given graph $G = (V, E)$ with edge weights $c : E \to \mathbb{R}$ into a number of clusters $\Pi = (\Pi_1, \ldots, \Pi_k)$, where the number of clusters k is determined as part of the optimization problem. The clusters form a partition of V, i.e. $\Pi_i \cap \Pi_j = \varnothing \; \forall i \neq j$ and $\Pi_1 \cup \ldots \cup \Pi_k = V$ where each component $\Pi_i \in \Pi$ implies a connected subgraph of G. A multicut is an edge indicator vector $y : E \to \{0, 1\}$ defined by $y_e = 0$ iff the endpoints of e are in the same component. The multicut problem can be written as an integer linear program:

$$\min_{y \in \{0,1\}^E} \sum_{e \in E} c_e \, y_e \quad (10)$$

$$\text{s.t.} \; \forall C \in \text{cycles}(G), \; \forall e \in C : y_e \leq \sum_{\bar{e} \in C \setminus \{e\}} y_{\bar{e}}. \quad (11)$$

The lifted multicut problem [25] augments the original multicut problem by introducing a second set of lifted

edges E' together with lifted edge costs $c' : E' \to \mathbb{R}$. For any edge $e' = (i, j) \in E'$ and clusters $(\Pi_1, \ldots, \Pi_k)$, a lifted edge indicator vector is defined as

$$y_{e'} = 0 \Leftrightarrow \exists l \text{ and a path } P \in ij\text{-paths}(G) \text{ s.t. } P \in \Pi_l. \quad (12)$$

In words, the label of lifted edge is 0 iff there exists a path connecting its endpoints through base edges in E and through nodes that all are in a single cluster.

Spatial-Temporal Lifted Tracking Graph We now propose our lifted multicut for multiple object tracking given a spatial-temporal graph. Specifically, three types of edges are employed: temporal, spatial, and constraint edges (Figure 2-c). The temporal edges $E_{t_{\max}}^t$ connect tracklet nodes observed by the same camera at different timesteps up to some maximal time threshold $t_{\max}$ (10 seconds in our experiments), spatial edges E^s connect tracklets observed by different cameras at overlapping timeframes, and tracklet nodes that are observed by the same camera and have overlapping timeframes are forbidden to end up in the same trajectory through constraint edges E^c. The formulations for these edge sets are:

$$E_{t_{\max}}^t = \left\{ (\tau, \tau') : \begin{array}{c} \mathrm{cam}(\tau) = \mathrm{cam}(\tau'), \\ \max\{\mathrm{time}(\tau')\} - \min\{\mathrm{time}(\tau)\} \in (0, t_{max}] \end{array} \right\}$$
$$E^s = \left\{ (\tau, \tau') : \begin{array}{c} \mathrm{cam}(\tau) \neq \mathrm{cam}(\tau'), \\ \mathrm{time}(\tau) \cap \mathrm{time}(\tau') \neq \varnothing \end{array} \right\}$$
$$E^c = \left\{ (\tau, \tau') \in V \times V : \begin{array}{c} \mathrm{cam}(\tau) = \mathrm{cam}(\tau'), \\ \mathrm{time}(\tau) \cap \mathrm{time}(\tau') \neq \varnothing \end{array} \right\}. \quad (13)$$

Given the above edges, we divide them into base edges E and lifted edges E':

$$E = E_{5\mathrm{sec}}^t \cup E^c \cup E^s; E' = E_{10\mathrm{sec}}^t \backslash E_{5\mathrm{sec}}^t. \quad (14)$$

That is, all edges such that tracklets have temporal distance less than 5 seconds are base edges (e_{base}^t), while temporal edges with larger time distance $t_{\max}$ are lifted ones (e_{lift}^t). The lifted edges are used here to incorporate long-range interactions of objects, i.e. handling the case when an object disappears due to occlusion and reappears again.

The edge costs c_{e^t} and c_{e^s} for temporal $e^t \in E_{t_{\max}}^t$ and spatial edges $e^s \in E^s$ in our framework are produced by two neural networks f_{temporal} and f_{spatial}, which take input features described in Section 3.3 and return a similarity score (Subsection G.2 Appendix). We also assign a large negative value $c_{e^c} = M \ll 0$ for each $e^c \in E^c$ to guarantee that clusters are trajectories.

Lifted Multicut Formulation We state our lifted multicut optimization for the multi-camera tracking as:

$$\min_{y \in \{0,1\}^{E \cup E'}} \sum_{e^t \in E_{t_{\max}}^t} c_{e^t} y_{e^t} + \sum_{e^s \in E^s} c_{e^s} y_{e^s} + \sum_{e^c \in E^c} M y_{e^c}$$
$$\text{s.t. } y \text{ obeys Equations (11) and (12).} \quad (15)$$

Since this formulation is NP-Hard [25], we resort to efficient heuristic solvers presented in Section E Appendix.

In summary, our formulation has the following advantages. First, we optimize intra-camera (temporal edges) and inter-camera (spatial edges) simultaneously. Additionally, long-range interactions are also incorporated through lifted-edges in E'. This results in a solution informed by cues from all cameras at once. Second, the optimal number of trajectories is determined during the optimization. Third, our nodes in the spatial-temporal graph are tracklets, resulting in decreased execution time as the hypothesis space is significantly reduced.

4. Experimental Analysis

Below we detail our experimental setup and results, including implementation details of our approach, considered datasets and baselines, quality metrics, and ablations.

4.1. Implementation Details

Tracking Graph We use CenterTrack [53], one of current state-of-the-art single-camera trackers, to create tracklets for each camera. These tracklets will be aligned again with the provided public detections to eliminate duplicated ones and retain detections that were not tracked, e.g. due to occlusion. We apply DG-Net [52] for extracting embedding vectors for detections and train it on visible detections (Eq. 2) obtained by running the proposed pre-clustering algorithm over training video sequences. Finally, we train three basic multi-layer deep networks for the three networks f_{split}, f_{spatial}, and f_{temporal} which handle splitting up initial tracklets, generating affinity costs for spatial and temporal edges respectively.

Lifted Multicut Solver For solving the proposed lifted multicut problem (Eq. 15), we use the efficient GAEC+KLj heuristic solver from [25]. We adapt a two-stage optimization approach. First, the initial tracklets are used as nodes in the tracking graph and compute trajectories through the lifted multicut. Second, we initialize the computed trajectories again as nodes in the tracking graph and recompute costs to obtain the final trajectories. The second step improves the trajectories computed in the first pass by allowing to compute costs on longer trajectories and joining trajectories together where a connection was not initially detected. For a long-time video, this process can be iterated until converged. More details of our implementations are described in Subsection G.1 Appendix.

4.2. Datasets & Metrics

We perform experiments on three datasets with a wide range of different camera configurations, densities, and video/bounding box qualities. For all datasets, we use the provided detections for a fair comparison.

- *WILDTRACK* [7]: The largest-scale dataset for the multi-camera setting is currently with a dense group of 313 pedestrians standing and walking. There is a total of 400 frames observed from seven cameras that are annotated with 3D positions. The first 360 of those frames is used for training and the rest for testing.

- *Campus* [44]: We have chosen the two sequences *Garden 1* and *Parking Lot* which had camera calibration parameters and a ground plane for 3D-projection. There are 15 - 25 pedestrians in each video observed by four cameras captured at a 30fps rate. For each video, we use the first 10% for training and the remaining frames for testing.

- *PETS-09* [12]: Contains three sequences with increasing level of difficulty: low density (S2.L1) - 19 objects in 795 frames, medium density (S2.L2) - 43 pedestrians spreading in 436 frames and high density (S2.L3) - 44 pedestrians moving together in 240 frames. While PETS-09 is not as dense as WILDTRACK, its main challenge lies in its poor video acquisition conditions with cameras far away from targets and low-quality bounding boxes.

We report results for the following metrics:

- MOTA [2]: the multiple objects tracking accuracy measuring a number of false negatives (FN), false positives (FP), and identity switches (IDs) focusing on the coverage of detections.

- MOTP [2]: the multiple object tracking precision penalizes the overall dissimilarity between true positives and the ground truth objects.

- IDF1 [35]: measure through the F1 score agreement between computed and ground truth trajectories.

- MT [30]: number of mostly tracked objects for at least 80% of its life span.

- ML [30]: number of mostly lost objects for at most 20% of its life span.

4.3. Algorithms

We compare against state-of-the-art baselines for each respective dataset. To study the effectiveness of our approach and judge the effectiveness of each proposed component, we run experiments on multiple configurations. All algorithms are listed below.

`Baselines`: For each dataset, we compare against the most recent approaches for which we found experimental results. In particular, we compare against KPS-DO, KSP-DO-ptrack [7], GLMB-YOLOv3, GLMB-DO [32], DMCT, DMCT Stack [47], HJMV [20], STVH [42], MLMRF [28], HCT [44], STP [45], TRACTA [16] and DyGLIP [33]. All baseline performances are taken from the respective original papers.

Our configurations are:

– `LMGP w/o Pre-Clustering`: see Section 3.1.

– `LMGP w/o Tracklet Split`: no tracklet splitting using information from pre-clustering (Section 3.2, Splitting Tracklets).

– `LMGP w/o Enhanced Affinities`: do not use pre-clustering in affinity cost computation (Eq. 6, 9).

– `LMGP`: our approach with everything enabled. For the *Campus* dataset we also provide as comparison `LMGP-DeepSort` using DeepSort [43] for generating tracklets. Note that CenterTrack outperforms DeepSort on single camera benchmarks.

4.4. Results

Method	IDF1 ↑	MOTA ↑	MT ↑	ML ↓	FP ↓	FN ↓	IDs ↓
KSP-DO [7]	73.2	69.6	28.7	25.1	1095	7503	85
KSP-DO-ptrack [7]	78.4	72.2	42.1	14.6	2007	5830	103
GLMB-YOLOv3 [32]	74.3	69.7	79.5	21.6	424	1333	104
GLMB-DO [32]	72.5	70.1	93.6	22.8	960	990	107
DMCT [47]	77.8	72.8	61.0	4.9	91	126	42
DMCT Stack [47]	81.9	74.6	65.9	4.9	114	107	21
LMGP w/o Pre-Clustering	79.4	73.6	76.4	24.3	570	506	76
LMGP w/o Tracklet Split	92.4	89.7	92.6	8.3	285	119	45
LMGP w/o Enhanced Affinities	95.1	94.8	95.5	5.7	95	103	27
LMGP	**98.2**	**97.1**	**97.6**	**1.3**	71	7	**12**

Table 1. Our LMGP performance compared to state-of-the-art baselines on *WILDTRACK*.

Sequence	Method	MOTA ↑	MOTP ↑	MT ↑	ML ↓	IDs ↓
	HJMV [20]	91.7	79.4	94.7	0.0	45
	STVH [42]	95.1	79.8	**100.0**	0.0	13
S2-L1	MLMRF [28]	96.8	79.9	**100.0**	0.0	**2**
	LMGP w/o Pre-Clustering	97.5	79.6	96.3	0.0	6
	LMGP w/o Tracklet Split	97.3	82.1	97.2	0.0	6
	LMGP w/o Enhanced Affinities	97.6	82.1	98.1	0.0	4
	LMGP	**97.8**	**82.4**	**100.0**	0.0	**2**
	HJMV [20]	58.9	66.0	30.2	2.3	388
	STVH [42]	65.2	61.8	44.2	0.0	249
S2-L2	MLMRF [28]	**72.1**	58.3	**72.1**	2.3	142
	LMGP w/o Pre-Clustering	66.8	64.3	67.4	2.1	158
	LMGP w/o Tracklet Split	70.1	68.5	67.2	2.1	97
	LMGP w/o Enhanced Affinities	71.6	**73.2**	68.2	2.0	81
	LMGP	70.4	**73.2**	69.6	1.7	**75**
	HJMV [20]	40.2	49.5	29.6	25.0	123
	STVH [42]	49.8	63.0	29.6	20.5	92
S2-L3	MLMRF [28]	**54.4**	54.9	36.4	6.8	82
	LMGP w/o Pre-Clustering	49.1	55.7	33.2	13.6	92
	LMGP w/o Tracklet Split	52.7	63.6	33.4	11.2	86
	LMGP w/o Enhanced Affinities	53.5	64.7	31.8	8.7	82
	LMGP	**54.4**	**66.5**	**40.2**	**5.3**	**78**

Table 2. Our LMGP performance compared to other baselines in *PETS-09*.

Sequence	Method	MOTA ↑	MOTP ↑	MT ↑	ML ↓
	HCT [44]	49.0	71.9	31.3	6.3
	STP [45]	57	75	-	-
Garden 1	TRACTA [16]	58.5	74.3	30.6	1.6
	DyGLIP [33]	71.2	91.6	31.3	**0.0**
	LMGP-DeepSort	75.6	93.4	46.7	1.6
	LMGP	**76.9**	**95.9**	**62.9**	1.6
	HCT [44]	24.1	66.2	6.7	26.6
	STP [45]	28	68	-	-
Parking Lot	TRACTA [16]	39.4	74.9	15.5	10.3
	DyGLIP [33]	72.8	**98.6**	26.7	**0.0**
	LMGP-DeepSort	76.7	98.0	51.7	5.1
	LMGP	**78.1**	97.3	**62.1**	**0.0**

Table 3. Our LMGP performance compared to other baselines on *Campus*.

We report quantitative results for *WILDTRACK* in Table 1, for *PETS-09* in Table 2 and for *Campus* in Table 3. Some qualitative results are presented in Figures 8, 9 Appendix.

On *WILDTRACK* we obtain almost perfect metric scores with `LMGP`. Our results on *Campus* significantly outperform the state-of-the-art on both sequences even when using weak tracklets (DeepSort). On *PETS-09* we achieve comparable results. We argue that the differing performance of our algorithms is mostly due to poor bounding boxes and camera calibration for *PETS-09* and, to a lesser extent, for *Campus*.

4.5. Efficacy of Individual Components & Ablations

To assess the performance of individual parts in our approach and their contribution to our overall performance, we give more experimental details.

Ablations w.r.t. Pre-Clustering/Enhanced Affinities/Tracklet Splitting In Tables 1 and 2 we report results of the ablated versions `LMGP w/o Pre-Clustering`, `LMGP w/o Tracklet Split` and `LMGP w/o Enhanced Affinities` of our solver. In most cases (except for *S2-L1* of *PETS-09*, where nearly perfect results can be obtained with any baseline), we see significant improvements w.r.t. all ablated versions of our solvers, validating the efficacy of all steps. *The greatest performance drop can be observed by turning of the pre-clustering*, showing its important role for obtaining improved results. More fine-grained ablations for the enhanced affinity costs can be found in Table 6 and 7 Appendix.

Pre-Clustering Accuracy In Figure 4, we report performance of our Pre-Clustering (Section 3.1) for all considered datasets. Since our Pre-Clustering is purely based on geometry, we also see experimentally that its performance is dependent upon the accuracy of bounding box coordinates. In the high-quality *WILDTRACK* dataset, we obtain almost perfect results. For *Campus* and *PETS-09* we achieve precision higher than 80%. We argue that the *confident connection* in our pre-clustering helps in these more challenging and noisy settings. More details on the reduction of ID-switch errors can be found in Table 8 Appendix.
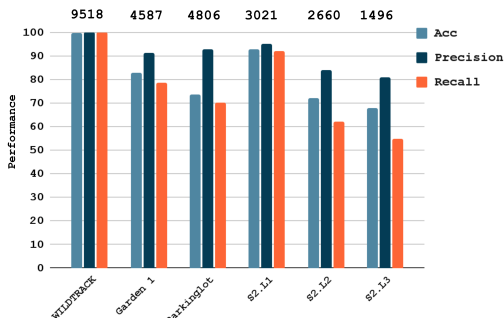

Figure 4. Our pre-clustering performance measured by accuracy, precision and recall w.r.t. correctly estimated correspondences between detections across cameras. On the top we give the total detections of objects in each dataset.

Joint Optimization Model In order to assess the performance of our joint spatio-temporal optimization model

(Eq.15) as compared to a stage-wise optimization, we provide an experiment in Table 4. The variant `LMGP w/o spatial edges` first solves an ordinary single view MOT problem. After obtaining computed single-camera trajectories, they are linked across cameras in a second step. To validate the effect of lifted long-range edges vs. a simpler model without them, we also provide an ablation `LMGP w/o lifted edges`. These ablations are tested on *WILDTRACK* and *S2-L1* of *PETS-09*. The results indicate that significant improvement can be gained by optimizing jointly over temporal and camera affinities, especially in dense scenes like WILDTRACK. Also, the long-range edges contribute to better performance in both cases.

Dataset	Method	MOTA ↑	IDF1 ↑	MT ↑	ML ↓	IDs ↓
WILDTRACK	LMGP full	**97.1**	**98.2**	**97.6**	**1.3**	**12**
	LMGP w/o lifted edges	95.4	96.4	93.6	2.7	41
	LMGP w/o spatial edges	93.2	94.1	91.7	6.9	85
S2-L1	LMGP full	**97.8**	**82.4**	**100.0**	**0.0**	**2**
	LMGP w/o lifted edges	96.2	81.1	98.7	1.3	5
	LMGP w/o spatial edges	95.6	80.2	98.7	1.3	8

Table 4. An ablation study of different edge types

4.6. Discussion

We have demonstrated that nearly perfect multiple object tracking results can be obtained in crowded scenes given the right conditions (as for *WILDTRACK*), i.e. high bounding box quality and a large enough number of well-calibrated cameras observing the same scene. Even when these conditions are not met, as in *Campus* and *PETS-09* which use old detectors, our `LMGP` still delivers better results than competitors in most cases. Specifically, the *pre-clustering is crucial in our framework* because it allows exploiting multi-camera information through repairing efficiently trajectories computed by single-camera trackers and can be used to enhance affinity costs. Also, our lifted multicut model jointly optimizing over inter- and intra-camera affinities, short- and long-range interactions is essential and can effectively correct erroneous associations and continue trajectories which would be lost when only a single camera is used and severe occlusions are present.

5. Conclusion

We have shown that integrating single view-based approaches and centralized representation-based methods for multi-camera tracking can lead to improvements w.r.t. strategies that fall into only one of these paradigms. Given a good enough input data, this strategy can deliver almost optimal results even in crowded scenes. We conjecture that in noisier settings, significantly better results can be achieved by making our 3D pre-clustering more robust, e.g., by employing it in an end-to-end training framework [5, 46].

6. Acknowledgements

References

[1] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixé. Tracking without bells and whistles. In *IEEE International Conference on Computer Vision*, pages 941–951, 2019.

[2] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008.

[3] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, pages 3464–3468. IEEE, 2016.

[4] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11):1222–1239, 2001.

[5] Guillem Brasó and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6247–6257, 2020.

[6] Rainer E Burkard and Eranda Cela. Linear assignment problems and extensions. In *Handbook of combinatorial optimization*, pages 75–149. Springer, 1999.

[7] Tatjana Chavdarova, Pierre Baqué, Stéphane Bouquet, Andrii Maksai, Cijo Jose, Timur Bagautdinov, Louis Lettry, Pascal Fua, Luc Van Gool, and François Fleuret. Wildtrack: A multi-camera hd dataset for dense unscripted pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5030–5039, 2018.

[8] Sunil Chopra and Mendu R Rao. The partition problem. *Mathematical Programming*, 59(1):87–115, 1993.

[9] Peng Chu and Haibin Ling. FAMNet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking. In *IEEE International Conference on Computer Vision*, pages 6172–6181, 2019.

[10] Afshin Dehghan, Shayan Modiri Assari, and Mubarak Shah. GMMCP tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4091–4099, 2015.

[11] Patrick Dendorfer, Aljosa Osep, Anton Milan, Konrad Schindler, Daniel Cremers, Ian Reid, Stefan Roth, and Laura Leal-Taixé. Motchallenge: A benchmark for single-camera multiple target tracking. *International Journal of Computer Vision*, 129(4):845–881, 2021.

[12] Anna Ellis, Ali Shahrokni, and James Michael Ferryman. Pets2009 and winter-pets 2009 results: A combined evaluation. In *2009 Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 1–8. IEEE, 2009.

[13] Francois Fleuret, Jerome Berclaz, Richard Lengagne, and Pascal Fua. Multi-camera people tracking with a probabilistic occupancy map. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):267–282, 2007.

[14] Dirk Focken and Rainer Stiefelhagen. Towards vision-based 3-d people tracking in a smart room. In *Proceedings. Fourth IEEE International Conference on Multimodal Interfaces*, pages 400–405. IEEE, 2002.

[15] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004.

[16] Yuhang He, Xing Wei, Xiaopeng Hong, Weiwei Shi, and Yihong Gong. Multi-target multi-camera tracking by tracklet-to-target assignment. *IEEE Transactions on Image Processing*, 29:5191–5205, 2020.

[17] Roberto Henschel, Laura Leal-Taixé, Daniel Cremers, and Bodo Rosenhahn. Fusion of head and full-body detectors for multi-object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2018.

[18] Roberto Henschel, Laura Leal-Taixé, Bodo Rosenhahn, and Konrad Schindler. Tracking with multi-level features. *arXiv preprint arXiv:1607.07304*, 2016.

[19] Roberto Henschel, Timo von Marcard, and Bodo Rosenhahn. Simultaneous identification and tracking of multiple people using video and imus. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[20] Martin Hofmann, Daniel Wolf, and Gerhard Rigoll. Hypergraphs for joint multi-view reconstruction and multi-object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3650–3657, 2013.

[21] Andrea Hornakova, Roberto Henschel, Bodo Rosenhahn, and Paul Swoboda. Lifted disjoint paths with application in multiple object tracking. In *The 37th International Conference on Machine Learning (ICML)*, July 2020.

[22] Andrea Hornakova, Timo Kaiser, Paul Swoboda, Michal Rolinek, Bodo Rosenhahn, and Roberto Henschel. Making higher order mot scalable: An efficient approximate solver for lifted disjoint paths. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6330–6340, 2021.

[23] Weiming Hu, Xinchu Shi, Zongwei Zhou, Junliang Xing, Haibin Ling, and Stephen Maybank. Dual L1-normalized context aware tensor power iteration and its applications to multi-object tracking and multi-graph matching. *International Journal of Computer Vision*, Oct 2019.

[24] Justin M Johnson and Taghi M Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):1–54, 2019.

[25] Margret Keuper, Evgeny Levinkov, Nicolas Bonneel, Guillaume Lavoué, Thomas Brox, and Bjoern Andres. Efficient decomposition of image and mesh graphs by lifted multicuts. In *IEEE International Conference on Computer Vision*, 2015.

[26] Saad M Khan and Mubarak Shah. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *European Conference on Computer Vision*, pages 133–146. Springer, 2006.

[27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[28] Long Lan, Xinchao Wang, Gang Hua, Thomas S Huang, and Dacheng Tao. Semi-online multi-people tracking by re-identification. *International Journal of Computer Vision*, pages 1–19, 2020.

[29] Evgeny Levinkov, Alexander Kirillov, and Bjoern Andres. A comparative study of local search algorithms for correlation clustering. In *German Conference on Pattern Recognition*, pages 103–114. Springer, 2017.

[30] Yuan Li, Chang Huang, and Ram Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2953–2960. IEEE, 2009.

[31] A Mittal, LS Davis, and M Tracker. Amulti-view approach to segmenting and tracking people in a cluttered scene using region-based stere. In *Proc. of the 7th ECCV*, 2002.

[32] J. Ong, B. T. Vo, B. N. Vo, D. Y. Kim, and S. Nordholm. A bayesian filter for multi-view 3d multi-object tracking with occlusion handling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020.

[33] Kha Gia Quach, Pha Nguyen, Huu Le, Thanh-Dat Truong, Chi Nhan Duong, Minh-Triet Tran, and Khoa Luu. Dyglip: A dynamic graph model with link prediction for accurate multi-camera multiple object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13784–13793, 2021.

[34] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European conference on computer vision*, pages 17–35. Springer, 2016.

[35] Ergys Ristani and Carlo Tomasi. Tracking multiple people online and in real time. In *Asian Conference on Computer Vision*, pages 444–459. Springer, 2014.

[36] Ing G Sagerer, Ilker Savas, and Dipl-Inform Joachim Schmidt. Entwicklung eines systems zur visuellen positionsbestimmung von interaktionspartnern. 2005.

[37] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv preprint arXiv:1805.00123*, 2018.

[38] Siyu Tang, Bjoern Andres, Miykhaylo Andriluka, and Bernt Schiele. Subgraph decomposition for multi-target tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5033–5041, 2015.

[39] Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Multi-person tracking by multicut and deep matching. In *European Conference on Computer Vision*, pages 100–111. Springer, 2016.

[40] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. Multiple people tracking by lifted multicut and person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[41] Timo von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 4d human pose in the wild using imus and a moving camera. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 601–617, 2018.

[42] Longyin Wen, Zhen Lei, Ming-Ching Chang, Honggang Qi, and Siwei Lyu. Multi-camera multi-target tracking with space-time-view hyper-graph. *International Journal of Computer Vision*, 122(2):313–333, 2017.

[43] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017.

[44] Yuanlu Xu, Xiaobai Liu, Yang Liu, and Song-Chun Zhu. Multi-view people tracking via hierarchical trajectory composition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4256–4265, 2016.

[45] Yuanlu Xu, Xiaobai Liu, Lei Qin, and Song-Chun Zhu. Cross-view people tracking by scene-centered spatio-temporal parsing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[46] Yihong Xu, Aljosa Osep, Yutong Ban, Radu Horaud, Laura Leal-Taixé, and Xavier Alameda-Pineda. How to train your deep multi-object tracker. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6787–6796, 2020.

[47] Quanzeng You and Hao Jiang. Real-time 3d deep multi-camera tracking. *arXiv preprint arXiv:2003.11753*, 2020.

[48] Amir Roshan Zamir, Afshin Dehghan, and Mubarak Shah. GMCP-tracker: Global multi-object tracking using generalized minimum clique graphs. In *European Conference on Computer Vision*, pages 343–356. Springer, 2012.

[49] Qi Zhang and Antoni B Chan. Wide-area crowd counting via ground-plane density maps and multi-view fusion cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8297–8306, 2019.

[50] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, pages 1–19, 2021.

[51] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *IEEE International Conference on Computer Vision*, pages 1116–1124, 2015.

[52] Zhedong Zheng, Xiaodong Yang, Zhiding Yu, Liang Zheng, Yi Yang, and Jan Kautz. Joint discriminative and generative learning for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[53] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European Conference on Computer Vision*, pages 474–490. Springer, 2020.