# Appearance features for online multiple camera multiple target tracking

Quoc Cuong Le, Moncef Hidane

# Appearance Features for Online Multiple Camera Multiple Target Tracking*

Quoc Cuong Le
Université de Tours
Laboratoire d'Informatique Fondamentale et Appliquée de Tours - EA 6300
Tours, France
quoccuong.le@etu.univ-tours.fr

Moncef Hidane
INSA Centre Val de Loire
Laboratoire d'Informatique Fondamentale et Appliquée de Tours - EA 6300
Blois, France
moncef.hidane@insa-cvl.fr

## ABSTRACT

Multiple object tracking methods in the state-of-the-art are challenged by appearance variation, environment changes and long-term occlusions. Exploiting multiple calibrated and frame synchronized cameras holds the promise of alleviating these problems, in particular, the one pertaining to occlusion. The practical realization of this idea faces the problem that the appearance of the same target can change through different cameras. Thus, particular care should be taken in order to enhance the computation of appearance distances between targets in multiple cameras. In this paper, we tackle the problem of multiple object multiple camera tracking by adopting a Markov Decision Process framework. We concentrate on the effect of the affinity function by discussing different possible implementations and validating their performance, in terms of the MOT metric and the ID measure, on the PETS 2009 and EPFL datasets. Our experimental result shows a significant improvement of multiple cameras approaches with a sufficiently large overlapping zone compared to single camera ones.

## KEYWORDS

Multiple Object Tracking (MOT), Multiple Target Multiple Camera Tracking (MTMCT), Tracking-by-Detection, Data Association, Appearance Feature Extraction

**ACM Reference Format:**
Quoc Cuong Le and Moncef Hidane. 2020. Appearance Features for Online Multiple Camera Multiple Target Tracking. In *SAC '20: The 35th ACM/SIGAPP Symposium on Applied Computing (SAC '20), March 30-April 3, 2020, Brno, Czech Republic.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3341105.3373960

## 1 INTRODUCTION

As one of main subcategories of Visual Object Tracking, Multiple-object tracking (MOT) is a fundamental and largely studied problem in computer vision. Its applications involve many real-world problems such as visual surveillance, traffic monitoring, person identification and autonomous driving. The main objective of MOT is to determine the trajectories of a number of targets in a video.

A natural, and naive, way to handle MOT is to consider multiple single-object trackers (SOT). Practical implementations of this strategy reveal a number of crucial problems such as tracker initialization on every frame to detect potential targets, interactions between the targets causing frequent mutual occlusions, similar appearances resulting in identity switches or losing identity of lost targets when reappearing on the scene.

Unlike SOT-based MOT approaches, recent MOT algorithms, especially those targeted at pedestrians tracking, follow a *tracking-by-detection* strategy [2, 23, 41]. This is due to the efficiency of recent detectors [8, 10] which have proved their capability in detecting people with high accuracy, even in various illumination conditions and in cluttered backgrounds. Having all detections through video frames, a data association algorithm gathers all detections belonging to the same targets, which is based on their affinity, in order to achieve complete trajectories. The usual distinction in this setting revolves around the online/offline dichotomy: while online methods solely use results form previous detections, offline ones consider the whole (or batch) time-sequence in order to compute data associations. Obviously, non-causal systems are not suited for time-critical applications such as robot navigation and autonomous driving. In most cases, tracking-by-detection MOT methods are formulated as global optimization in a graph-based representation whose vertices represent detections and edges weighted by given distances (or affinities) [2, 28, 33, 41].

Both SOT-based and association-based methods need an efficient way to manage the state of the targets in order to avoid missing tracks and identity switches. This is especially true for applications requiring tracking in online mode. The Markov Decision Processes (MDP) framework [39] has been leveraged to tackle these issues. The MDP model helps the tracking process control the beginning/ending and temporal appearance/disappearance of the targets in a principled way and in an online context. This is achieved by explicitly modeling the lifetime of a given target and by devising an optimal policy that determines the sequence of states of each target. This approach has been extended to an overlapping multiple camera setting in [19] and has shown capability in allowing the individual cameras to recapture/re-identify their lost targets.

In this paper, we use the multi-camera MDP framework [19] to implement various appearance features as a robust distinctive function to re-identify targets within multi-camera setting. The goal is to use the different views provided by *overlapping*, *calibrated* and

*Produces the permission block, and copyright information

*frame-synchronized* cameras in order to increase the robustness to occlusions and the performance of targets' identity recovery, thus improving the overall ID-measure score. Indeed, we propose a new robust distance function relying on both trajectory and appearance features, and use it to associate targets in different views. We review some appearance features in common use in the state-of-the-art and analyze the impact of these features on the performance of the proposed method. Furthermore, we conduct exhaustive experiments to study the impacts of different appearance and trajectory features on tracking results. Additionally, our experimental study shows that using multiple cameras makes recapturing lost targets more flexible and efficient, and that it also helps reveal the missing trajectories due to occlusions.

Our paper is organized as follows. Section 2 is dedicated to related work containing both single-camera and multiple-camera MOT approaches. We recall the MDP-based Multi-camera Multi-object tracking approach proposed in [19] in Section 3 and then conduct a study of different distance functions and their impact on the final tracking result in Section 4. The experimental validation of our method is addressed in Section 5. We conclude the paper in Section 6.

## 2 RELATED WORK

### 2.1 Single-Camera Multiple-Object Tracking

Since the management of the different targets is the main challenge for MOT, the tracking-by-detection paradigm has evolved as the main approach. This is especially true since the advent of high-performing category detectors. Tracking-by-detection approaches can be separated into two categories: online and offline methods.

*2.1.1 Offline approaches.* Following the tracking-by-detection paradigm [2, 41], graph optimization problems are formulated to link the detections of targets, between successive frames, in order to form complete trajectories. These methods have become popular because they simplified the classic issues mentioned above such as trackers management, interaction, initialization and update. The formulation of the data association problem relies on a graph whose nodes represent the detections/features and whose edges are weighted by the distance (or affinity) between detections. The association methods usually collect all detections/features over the video, the current position of a target (a node of the graph) being thus determined by adjacent nodes that represent past and future detections. The goal of data association methods is to optimize the cost made by the edges of the graph. There are various methods using global and flow network optimization algorithms [42], and relying on criteria such as Graph Clique [36, 41], Graph Multicut [16, 27, 34, 35], Network Flow [2, 5, 23, 42], Maximum Weight Independent Set [4, 6, 18].

*2.1.2 Online approaches.* To fulfill the need for immediate tracking results in many applications, numerous papers proposed online tracking methods [1, 9, 23, 28, 32, 40, 43]. Within the tracking-by-detection paradigm, only detections in the current and previous frames are used to form targets' trajectories. One of the most popular approaches to associate detections is the bipartite matching formulation [24, 28, 32, 43], usually solved by using the Hungarian algorithm or heuristic approaches. Some offline methods, which can perform online when their optimization process only uses detections from the last frames to current, can be considered as "near-online" methods such as [6, 27]. Alternatively, the tracking-by-detection strategies and multiple SOT algorithms are combined to benefit from the SOT trackers and the ability of recovering lost targets of data association approaches in [39, 40, 43].

### 2.2 Multiple-Camera Multiple-Object Tracking

MOT approaches based on a single camera have recently been extended to multiple cameras. These approaches have been proposed in an attempt to fully cover the observation of the objects. Multiple-camera tracking can solve the problem of occlusion where the interesting targets are frequently occluded by the environment or by other targets. First attempts in using multiple (non-overlapping) cameras dealt with the re-identification problem, in order to track objects between cameras [37]. Following this approach, many researchers studied the problem of collaboratively using overlapping cameras for tracking. Almost all authors use the hypothesis that the exact position of each camera is known and camera calibration has been done before applying tracking. In the tracking phase, the trackers implemented on different cameras usually pool their results into a 3-D coordinate system via projection from image plane to ground plane in real world [21, 22, 31]. This allows combining the different results, and in particular reconnecting missing trajectories.

Besides of the above generic multi-camera tracking approaches, the methods based on the tracking-by-detection arises as an alternative. These methods inherit from most of the global optimization methods of MOT in single view such as graph multicuts [16, 27, 33–35], graph cliques [7, 41], network flow [5, 23, 42]. Meanwhile, the other data association methods including bipartite matching [1, 28, 32] and independent set [4, 18] do not address multi-camera tracking problem, because the tracklets are formed through the detections in consecutive frames (i.e. a short time window) of a single view, whereas tracking with multiple cameras is to connect trajectories of targets at different times. Some other approaches [17, 36] generalize multi-camera tracking into 2 main steps: MOT on each single view, then linking the trajectories across cameras. Unfortunately, none of those mentioned methods is online. Recently, Le et al. [19] introduced an online multi-camera tracking based on data association on each processing frame. In the next section, we introduce this multiple-camera multiple-object tracking framework to handle hard occlusions and prevents identity switches.

## 3 PRELIMINARY

### 3.1 Object management in SOT-based MOT

As mentioned previously, occlusion caused by their targets themselves or environment is a crucial point while implementing SOT trackers on *online* MOT applications. The very first attempt proposed by Xiang *et al.* in [39] introduced the Markov Decision processes to model multiple targets during tracking.

In their framework, the lifetime of a target is represented by the state of a Markov Decision Process (MDP) which consists of a target state space $\mathcal{S}$, a set of possible actions $\mathcal{A}$ and a state transition function $\mathcal{T}$. Using the idea of separating tracking process of an individual target into multiple tracking states such as tracked, lost, active, the most recent MOT papers [19, 40, 43] use this initiative to

naturally deal with the present/absence of multiple targets. In the context of multiple camera tracking, Le et al. [19] adapt this object management strategy to a multi-camera setting by introducing a target association method across cameras. The following section presents the basic notions of the data association across cameras approach.

## 3.2 Data Association Across Camera Views

The data association across camera views aims to link together the trackers of a specific target on multi-camera at every frame. Therefore, the following formulation is set at a single frame instant. Given a graph $G = (V, E, w)$, where $V$, $E$ and $w$ respectively denote the set of nodes, set of edges and the weights of the edges, the "alive" targets *of all cameras* are considered as the set $V$ of nodes. Let $C_k = \{v_1^k, ..., v_M^k\}$ be the cluster of targets in the camera $k$. The cluster $C_k$ includes, in particular, all detections which are not considered as being false positives. The edges of the graph are defined as $E = \{(v_m^k, v_n^l) | m, n \geq 1 \text{ and } k \neq l\}$, with the condition $k \neq l$ indicating that two nodes in same camera cannot be connected. A node $v_m^k$ has a *trajectory feature-vector* $\mathbf{x}_m^k = \{x_m^{k,1} ..., x_m^{k,F}\}$, where $x_m^{k,i}$ correspond to the 2-dimensional coordinates, on the 3-D world ground plane $z = 0$, in the previous $i$ frame. The number $F$ corresponds to the number of past frames retained. These 2-D coordinates are obtained by projecting the tracking result from image plane by using the homography matrix obtained from calibration data.

A node $v_m^k$ also has a bounding box $\Phi_m^k$ that will serve to compute *appearance feature-vector*, to be detailed later. The weight of an edge between two nodes is then defined by the following equation:

$$w(v_m^k, v_n^l) = \alpha \, f_{app}(\Phi_m^k, \Phi_n^l) + \beta f_{traj}(\mathbf{x}_m^k, \mathbf{x}_n^l). \tag{1}$$

The computation of the distance function $f_{traj}$ involves the trajectories between two targets in the last $L$ frames:

$$f_{traj}(\mathbf{x}_m^k, \mathbf{x}_n^l) = g\left(x_m^{k,F-L+1:F}, x_n^{l,F-L+1:F}\right), \tag{2}$$

where $L = \min\left(\min\left(|\mathbf{x}^k|, |\mathbf{x}_n^l|\right), 20\right)$ and $g$ is a function, to be detailed later, that quantifies the *average distortion* between trajectories.

As mentioned in [41], the process of matching a target in different views requires identifying correspondences of the target in all different views. Hence, the solution of the problem can be described as a connected subgraph of $G$ in which each node (target) is selected from only one cluster (view). Therefore, the subgraph for a particular tracked person can be denoted by $G_s = (V_s, E_s, w_s)$. The set of nodes $V_s$ has a general form $V_s = \{v_m^k | k \in \{1, ..., K\}\}$, $E_s = \{E(p, q) | p, q \in V_s\}$ and $w_s = \{w(p, q) | p, q \in V_s\}$. Fig. 1 shows some examples of connected subgraphs representing some targets through the views. Additionally, a maximal distance constraint on the edges is imposed to ensure that targets on all views must get close enough to confirm the identity of an *unique* person. More precisely, given a particular target in a particular view, all targets whose distance to the chosen one is greater than a $\epsilon$ value are removed.

Associating targets across cameras now amounts to finding a connected subgraph having a maximum number of targets gathered from all views, and having the following minimum cost:

$$\min_{G_s \subset G} \mathcal{F}(G_s) = \sum_{m,n,k,l} w(v_m^k, v_n^l) \; s.t. \; k \neq l, \tag{3}$$

where

$$G_s = \left\{V_s, E_s, w_s | \forall (v_p, v_q) \in E_s, w(v_p, v_q) \leq \epsilon \in \mathbb{R}\right\}. \tag{4}$$

Unlike the similar data association problem of [41], this method do not consider the data association through time but across cameras, at each frame. Without adding temporal constraints, the goal is to find a subgraph corresponding to each identity *in every frame*. Since this problem is NP-hard [41], Le et al. [19] proposed also a simple fast heuristic technique targeting an approximate solution.

With the result of subgraphs linking targets from same identity across views, the tracking process in each single camera can be improved and stabilized. The next section details how single views can benefit from multi-camera target association results.

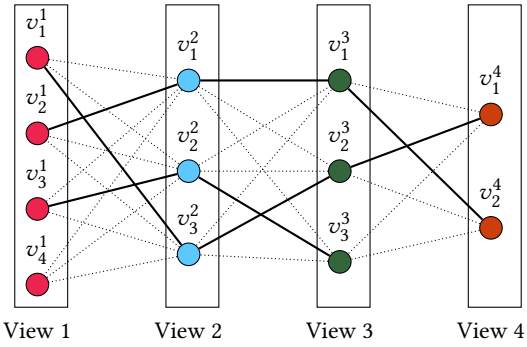


**Figure 1: Finding the corresponding targets in different views. For visualization purposes, only edges between 2 next views are shown in the figure, while edges connect targets between *all* the views. Edges in solid line connect a target corresponding to the *same identity*, in different views. In this example, there are three connected subgraphs indicating three people in the tracking process[19].**

## 3.3 Exploiting Multiple Cameras to improve robustness

After identifying targets on all views, this information is used to make views collaborate to deal with occlusions. When occlusion occurs on one view, it will request tracking result from other views. With this multi-camera tracking approach, there are two scenarios to be considered: partly occlusion (soft occlusion) and total and long-term occlusion (hard occlusion).

*3.3.1 Soft occlusion.* In practice, partly occlusion of target causes the *failure* while matching characteristics between the templates of target and detections during tracking process. To profit from detection results, from the views in which the target is being occluded, the representing points of all detections (foot position) are projected into the common ground plane. Then all the targets from all views that belong to the same identity give their position on the ground plane to get an average position of the target. The detection which stays closest to that position and is also not further than a fixed

threshold will be added to the result of the tracker which missed its target. These techniques help tracker perform more efficiently in the cases of partial occlusion as illustrated in Fig. 2.

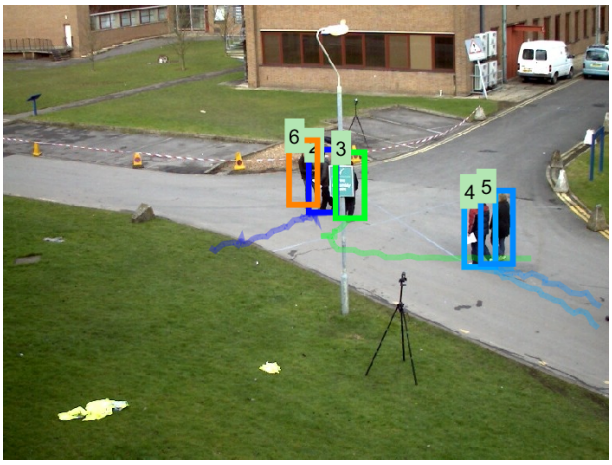


**Figure 2: The identity 3 is hidden behind the road sign in the displayed view. By using multiple cameras, our method is able to link the detection with the correct identity.**

*3.3.2 Hard occlusion.* In order to address short-term occlusions, a common association strategy searches the detection of lost targets around their last positions within a given time. As a simple and quick identity recovery technique, expanding the search area or/and the searching time period might be possible solutions. These are nonetheless not suited for all tracking scenes, particularly, in complex tracking scenarios where obstacles on the scene often conceal targets' movements which are already complex.

In a network of cameras with *overlapping fields of view*, trackers can benefit from multiple cameras to recapture their missing targets. In our implementation, when a camera detects a new target that persists in subsequent frames, it is compared with targets that are in "lost" state. Having the lost target's estimate from other views, the new target's position which is nearest to that estimation and less than a pre-defined threshold will be considered as the lost target.

The next section describes the our main contribution on appearance features for affinity measures between targets in multi-camera views.

# 4 APPEARANCE FEATURES AND TRAJECTORY DISTANCES

As aforementioned, the appearance of targets is an important cue to determine whether two targets in two different views are identical or not. Generally, an algorithm has access to the appearance cues of a target through a bounding box. The bounding box usually contains both the target and the background, the latter being dependent on the particular view/camera. In order to increase robustness, we also include a distance function between targets' trajectories, which is a consistent and pertinent factor that is invariant to the view, since each target moves on a unique path in a specific period of time during the videos. We now present a variety of features and distances used in our algorithm and dedicate a subsection to the delicate issue of combining distances based on different features.

## 4.1 Trajectory-Based Distances

The distance introduced in this section measures the disparity between the paths followed by targets *in different views*. We consider here 2-D paths connecting points whose coordinates correspond to the projection, via the homography matrix, onto the ground plane. In our approach, we selected two relevant methods: point-wise (averaged) Euclidean distance, and the Dynamic Time Warping distance [30]. For the point-wise distance, we consider trajectories involving $L$ points, given by:

$$f_{traj}(\mathbf{x}_m^k, \mathbf{x}_n^l) = \frac{1}{L}\sum_{i=0}^{L-1}\left\| x_m^{k,F-i} - x_n^{l,F-i} \right\|. \tag{5}$$

One of the issues in multiple-camera tracking systems is the *imperfection of frame synchronization*. Indeed, a multiple-camera system can only ensure that the movement of targets on the ground, captured by different cameras, is identical up to a certain tolerance. Thus, the path of a target can be dissimilar in different camera views. The Dynamic Time Warping algorithm [29] allows to alleviate this issue. In this setting, the point-wise distance is replaced by

$$f_{traj}(\mathbf{x}_m^k, \mathbf{x}_n^l) = \frac{1}{L} d_{DTW}(\mathbf{x}_m^{k,F-L+1:F}, \mathbf{x}_n^{l,F-L+1:F}), \tag{6}$$

## 4.2 Appearance Features

*4.2.1 Histogram of colors.* Color is an important cue that allows us to distinguish between different targets. In many cases, the color of the same target can change from camera to camera due to sensors differences. Another limitation pertaining to the use of color is related to the fact that the background behind targets can vary from on view to another. This leads to targets confusion and strongly affects the performance of tracking. In our implementation, after evaluating the histogram of all three RGB channels from the bounding box, we concatenate them into one feature vector and use the Euclidean distance to compare them:

$$f_{app}(\Phi_m^k, \Phi_n^l) = \left\| f_{CH}(\Phi_m^k) - f_{CH}(\Phi_n^l) \right\|_2. \tag{7}$$

*4.2.2 Median LK backward forward error.* We implement the appearance feature used in [39] that has shown its effectiveness to track a target by matching the points between two frames. Concretely, the algorithm performs a matching between the densely sampled points of optical flow calculated from the target's current appearance and the considered detection on the next frame. An optical flow is then computed backward in time. The median value of the backward-forward (BF) errors is used as an appearance distance [15]. We use the same idea in our multi-camera setting by computing the distances between the targets in different views:

$$f_{app}(\Phi_m^k, \Phi_n^l) = e_{medBF}(\Phi_m^k, \Phi_n^l), \tag{8}$$

where $e_{medBF}$ is the median of Backward-Forward errors, as defined in [15].

*4.2.3 DeepMatching.* The DeepMatching pairwise feature has been introduced in [34]. This method densely finds the matching points from an image to another. Thus the approach is devoted to the

quantity of the matching points rather than their quality. In tracking problems, it becomes more appealing when it is used to match detections that belong to the same person, the matching points on the body being more easily found than those on the background as the scene behind permanently changes while the target moves. In our implementation, when the DeepMatching algorithm [25] is deployed on two distinct views, we observe that the number of matching points decreasing dramatically, so the pairwise feature used in [34] becomes inaccurate. In order to resolve this issue, we propose a variant of deep matching inspired from the Lucas-Kanade backward-forward distance mentioned above. But instead of taking the BF distance, we measure the average displacement of the keypoints of the target's appearance $u_i \in R^2, 1 \leq i \leq K$ while matching them to those $u'_i \in R^2, 1 \leq i \leq K$ on other detections in *other views*, where K is the number of matching points found. A matching between two distinct individuals will amplify the displacement on average, while the correct matching of the same identity on multiple cameras will result in having a small displacement. With this choice, the distance function in Eq. (2) is defined as follows:

$$f_{app}(\Phi_m^k, \Phi_n^l) = e_{DM}(\Phi_m^k, \Phi_n^l) = \frac{1}{N}\sum_{i=1}^{N} ||u_i - u'_i||_2, \qquad (9)$$

*4.2.4 CNN-based Learning Appearance Features.* The features introduced so far are hand-crafted. A current trend, fueled by the recent success in deep learning, considers the features corresponding to the output produced by the hidden layers of a deep neural network. In [14], the authors introduced a novel CNN to extract useful appearance features, while eliminating unnecessary factors such as background and moving body parts. The output of the CNN (excluding the fully connected layers) embeds the input bounding box into a space where *Euclidean distance allows to disambiguate identity*. Letting $F : \Phi_m^k \in S \rightarrow f_{net}(\Phi_m^k) \in R^n$ denote the learned embedding provided by the CNN, we can define an appearance distance as:

$$f_{app}(\Phi_m^k, \Phi_n^l) = \left\| f_{net}(\Phi_m^k) - f_{net}(\Phi_n^l) \right\|_2. \qquad (10)$$

In this paper, we test the pre-trained CNN [14], named triNet, and another CNN feature extractor, the resNet50 [13] without the last fully connected layers.

## 4.3 Combining appearance and trajectory distances

The final distance measure we retain consists in a linear combination of the appearance and trajectory distances:

$$w(v_m^k, v_n^l) = \alpha\, f_{app}(\Phi_m^k, \Phi_n^l) + \beta f_{traj}(\mathbf{x}_m^k, \mathbf{x}_n^l). \qquad (11)$$

Because of the (slight) temporal misalignment between cameras, there is always a certain uncertainty level in the position and appearance of the same object in different views, when projected onto the ground plane. Fig. 3 a) and c) show, for two different datasets, the (empirical) standard deviation of the $\ell_2$ errors between the positions of the computed projections, seen from different cameras, and the ground truth. The figures show the evolution of these uncertainties through time (frames) and for each particular identity. Fig. 3 b) and d) illustrate the same uncertainties, but this time, based on the $\ell_2$ errors between the appearance features. The particular feature vector used in Fig. 3 b) and d) is the color histogram. The dataset used are *PETS09-S2L1* and *terrace1*.

We propose to relate the values of $\alpha$ and $\beta$ to these location and appearance uncertainties, respectively. More precisely, we consider the following distance:

$$w(v_m^k, v_n^l) = \frac{1}{\mu_1} f_{app}(\Phi_m^k, \Phi_n^l) + \frac{1}{\mu_2} f_{traj}(\mathbf{x}_m^k, \mathbf{x}_n^l), \qquad (12)$$

where $\mu_1$ and $\mu_2$ are the average, through all the identities present, of location and appearance uncertainties, as discussed earlier.

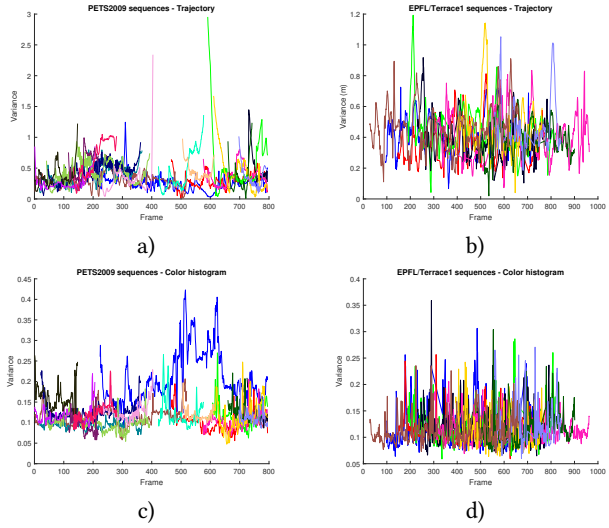


**Figure 3: The trajectory uncertainties in PETS09-S2L1 sequence (a) and terrace1 sequence (b). Figures (c) and (d) depict the uncertainties related to color histogram features. Each color corresponds to a particular identity.**

# 5 EXPERIMENTAL EVALUATION

## 5.1 Benchmarking Performance

**Tracking quality metrics.** In this paper, we evaluate our method by using the CLEAR MOT [3] and ID measure [26]. It is important to note that there is an important difference between these two metrics. The CLEAR MOT metric evaluates a tracker based on how often mismatches happen, while the ID metric measures how long the tracker maintains the identity of targets and how many times the tracker gives the prediction within an offset boundary value $\Delta$ from the ground-truth.

In this section, we present our experimental results verifying the efficiency of the multi-camera MOT algorithm with diverse appearance features. To evaluate MOT performance, the benchmark MotChallenge [20] has been released with 2 datasets (MOT15 and MOT16), which contain a number of single-view video sequences recorded by static or dynamic cameras, and the evaluation metrics of CLEAR MOT [3] and ID measure [26] are used. Additionally, the MotChallenge also provides multi-camera video sequences, but with cameras with mostly non-overlapping fields of view. Unfortunately, these datasets do not fit to this case study that focuses on

using multiple overlapping views to tackle the missing of targets by occlusions. Since the CLEAR MOT and ID measures are still appropriate for our current setting, the performance of our method is evaluated by these scores. As the multi-camera method aims to improve identity robustness in single views, we will emphasize ID scores in the sequel.

**Datasets.** The datasets in our experiments include the well-known PETS2009 [11] and EPFL Multi-camera Pedestrian Videos [2] datasets. Among all sequences of PETS2009, the most relevant and suitable for our multiple-camera tracking system is "PETS09-S2L1" with 7 views from 7 synchronized and calibrated cameras. For our experiment, only 1 main view (from the camera 1) and 4 close-up views (from the cameras 5, 6, 7 and 8) are used. Meanwhile, the EPFL dataset provides multiple indoor and outdoor video sequences, recording pedestrians by 4 different cameras. Because of the affinity between sequence scenarios, the sequences "Terrace1" and "Basketball" are selected for the experiments. In terms of camera topology, only about $15-20\%$ of the observable zones are covered by all cameras in most of the sequences. The detections of the other views of PETS2009 and those of "Terrace1" are obtained by the same public detector used on MotChallenge[1], which is detailed in the section 5.2. The ground truth and detection data on all views are available online[2].

**Evaluation metric.** To validate the efficiency of our various settings on the multi-camera MOT approach, we adopt the CLEAR MOT metric and ID measures and in particular the following scores: MOTA (multiple-object tracking accuracy), MOTP (multiple-object tracking precision), IDs (identity switches), IDF1 (ID F1-score), IDP (ID precision), IDR (ID Recall), False Positive (FP) and False Negative (FN). For further details on the metric, we recommend the MOTChallenge website[1].

## 5.2 Implementation

The experiments with the different distance functions and configurations are conducted with the implementation of the paper [19].

**Parameter.** In order to analyze the impact of each term in the weight function (1), we set up 2 setting for our experiments as following:

- Setting 1: $w(v_m^k, v_n^l) < \epsilon_1$
- Setting 2: $w(v_m^k, v_n^l) < \epsilon_1$ and $f_{traj} < \epsilon_2$,

where $\epsilon_1 = 4.2$ and $\epsilon_2 = 3.5\mu_1$, where $\mu_1$ is the position uncertainty level, as defined in the section 4.3. In other words, we test our algorithm without and with a spatial distance constraint to analyze the contribution of appearance features in the tracking results.

**Detection.** In all tracking-by-detection approaches, the detector plays an important role in tracking performance. We employ the widely used, public ACF (Aggregate Channel Features) detector [8] on all views of the sequences "PETS09-S2L1" and "Terrace1", using the pre-trained Caltech model [38].

**Competing methods.** To evaluate our approach, we compare our methods with the original MDP single-camera method and the multi-camera K-Shortest path (KSP) from the state-of-the-art. The KSP algorithm outputs quantized positions in the probabilistic occupancy map (POM) [12]. Our implementation uses a POM grid of size $36 \times 36$. The test is carried out with respect to different trajectory distances and appearance features detailed previously:

- Pointwise distance and Dynamic Time Warping distance
- Color histogram, median Backward-Forward LK error (LK), DeepMatching (DM) and learned appearance features using triNet and resNet50.

## 5.3 Performance analysis

**Overview.** The results shown in the following tables are the average values from all views. Concretely, the overall tracking results of the PETS sequence can be seen in the Table 1 and Table 2. Each score column has either an ↑ or a ↓ indicating whether better corresponds to higher or lower, respectively.

Primarily, the multi-camera tracking method focuses on tracking targets in the hard occlusion case. It leads to an important reduction of identity switches and a significant improvement of ID measures. In detail, with the overlapping zones, on average in the best setting, our approach on the considered PETS sequence increased about 14.9% for IDF1, 14.3% for IDP and reduced 36.7% of ID switches. In terms of CLEAR MOT scores, our approach slightly improves both MOTA and MOTP scores, because the CLEAR MOT metric does not focus on re-identification ability of tracking algorithms [26].

On the EPFL/terrace1 sequence, we observe the same effect. More precisely, the multiple cameras tracking method remarkably improved the ID measure scores in the Table 3 and Table 4. The score increases by 27.7% for IDF1 and 27.2% for IDP. In contrast, the MOT scores and ID switch number slightly decreased, but these changes are not notable. Meanwhile, EPFL/basketball sequence records a basketball matching where the overlapping zone on the basketball court is relatively small. In addition, the players are often seen in this zone with more intense and complex movements, these flash appearances of players on the overlapping zone renders the algorithm impossible to collaborate and share tracking results between views. As a result, the performance scores remain unchanged in overall (see Tabs. 6 5). However, due to the lack of the probabilistic occupancy map (POM) on this sequence, KSP method was not conducted.

The KSP method performs poorly on the sequence PETS09-S2L1, but achieved a good score on EPFL/terrace1. This result can be explained by the fact that KSP method was developed on the EPFL multiple- camera Pedestrian Dataset. In fact, the authors assume that the targets have to finish their complete trajectories before leaving the scene. The in/out position of targets is also known and fixed on the scene, so we can see the actors walking in and out at the same place. On EPFL/terrace1, the algorithm found 8 paths that exactly correspond to the 8 targets in the video, thus leading to a high IDF1 score. On the sequence PETS09, the algorithm cannot deal with the targets that usually went out and then returned into the scene. It just found the longest paths and completely ignored the targets which appeared in short period of time and regularly get confused by other targets at the boundary. Moreover, in PETS09 database, there is no constraints on where people will appear and disappear on the scene. Consequently, KSP receives a negative score on MOTA. It indicates that the KSP algorithm cannot handle the enter/exit of targets. Another problem with KSP is that the tracking process occurs on Probabilistic Occupancy Map (POM), whose

[1] https://motchallenge.net
[2] https://github.com/quoccuongLE/MDP_MTMC_Tracking

| Method | IDF1↑ | IDP ↑ | IDs↓ | MOTA↑ | MOTP↑ |
|---|---|---|---|---|---|
| MDP | 57.49 | 62.24 | 333 | 68.44 | 68.83 |
| MDPmv + triNet | 62.34 | 67.11 | 270 | 69.14 | 68.94 |
| MDPmv + resNet50 | 62.35 | 67.15 | 250 | 69.54 | 68.84 |
| MDPmv + LK | 63.76 | 68.80 | 252 | 69.50 | 68.84 |
| MDPmv + DM | 58.97 | 63.75 | 260 | 68.88 | 68.94 |
| MDPmv + CH | 64.14 | 69.11 | 251 | 69.15 | 68.99 |
| KSP | 21.51 | 18.16 | 812 | -29.63 | 64.27 |

**Table 1: Scores on "PETS09-S2L1" sequence (setting 1).**

| Method | IDF1↑ | IDP ↑ | IDs↓ | MOTA↑ | MOTP↑ |
|---|---|---|---|---|---|
| MDP | 57.49 | 62.24 | 333 | 68.44 | 68.83 |
| MDPmv + triNet | 65.76 | 70.91 | 213 | 70.54 | 69.04 |
| MDPmv + resNet50 | 65.93 | 71.09 | 212 | 69.94 | 68.91 |
| MDPmv + LK | 64.84 | 69.77 | 230 | 70.24 | 68.90 |
| MDPmv + DM | 66.00 | 71.00 | 207 | 70.67 | 68.99 |
| MDPmv + CH | 67.69 | 72.85 | 192 | 70.70 | 68.96 |
| KSP | 25.85 | 23.51 | 695 | 19.57 | 62.26 |

**Table 2: Scores on "PETS09-S2L1" sequence (setting 2).**

| Method | IDF1↑ | IDP ↑ | IDs↓ | MOTA↑ | MOTP↑ |
|---|---|---|---|---|---|
| MDP | 12.29 | 16.36 | 656 | 47.14 | 72.65 |
| MDPmv + triNet | 15.46 | 20.60 | 736 | 46.57 | 72.42 |
| MDPmv + resNet50 | 17.87 | 23.92 | 741 | 47.64 | 72.62 |
| MDPmv + LK | 15.76 | 20.87 | 768 | 46.49 | 72.53 |
| MDPmv + DM | 13.62 | 18.37 | 730 | 46.78 | 72.48 |
| MDPmv + CH | 16.00 | 21.12 | 761 | 46.85 | 72.44 |
| KSP | 25.85 | 23.51 | 695 | 19.57 | 62.26 |

**Table 3: Scores on "terrace1" sequence (setting 1).**

| Method | IDF1↑ | IDP ↑ | IDs↓ | MOTA↑ | MOTP↑ |
|---|---|---|---|---|---|
| MDP | 12.29 | 16.36 | 656 | 47.14 | 72.65 |
| MDPmv + triNet | 14.41 | 19.08 | 710 | 46.57 | 72.42 |
| MDPmv + resNet50 | 14.62 | 19.35 | 692 | 47.64 | 72.62 |
| MDPmv + LK | 14.93 | 19.85 | 681 | 46.49 | 72.53 |
| MDPmv + DM | 17.89 | 23.82 | 689 | 46.78 | 72.48 |
| MDPmv + CH | 16.60 | 21.98 | 699 | 46.85 | 72.44 |
| KSP | 25.85 | 23.51 | 695 | 19.57 | 62.26 |

**Table 4: Scores on "terrace1" sequence (setting 2).**

unit size directly affects the accuracy of the tracker. Unfortunately, increasing the resolution of the POM required more iterations to make sure the occupancy map converged correctly.

**Analysis of different features.** According to the IDF1 score, and excluding the KSP method, the results shows that there is no dominant appearance feature showing the best scores for all the tests. We remark that the LK feature and the two learned appearance features using triNet and resNet50 display a stable performance in the entire experiment. In contrast, the hand-crafted DeepMatching

| Method | IDF1↑ | IDP ↑ | IDs↓ | MOTA↑ | MOTP↑ |
|---|---|---|---|---|---|
| MDP | 4.648 | 6.847 | 4540 | 5.612 | 67.369 |
| MDPmv + triNet | 4.648 | 6.847 | 4540 | 5.612 | 67.369 |
| MDPmv + resNet50 | 4.648 | 6.847 | 4540 | 5.612 | 67.369 |
| MDPmv + LK | 4.495 | 6.863 | 4670 | 4.765 | 67.17 |
| MDPmv + DM | 4.648 | 6.847 | 4540 | 5.612 | 67.369 |
| MDPmv + CH | 4.648 | 6.847 | 4540 | 5.612 | 67.369 |
| KSP | - | - | - | - | - |

**Table 5: Scores on "basketball" sequence (setting 1).**

| Method | IDF1↑ | IDP ↑ | IDs↓ | MOTA↑ | MOTP↑ |
|---|---|---|---|---|---|
| MDP | 4.648 | 6.847 | 4540 | 5.612 | 67.369 |
| MDPmv + triNet | 4.648 | 6.847 | 4540 | 5.612 | 67.369 |
| MDPmv + resNet50 | 4.648 | 6.847 | 4540 | 5.612 | 67.369 |
| MDPmv + LK | 4.495 | 6.863 | 4670 | 4.765 | 67.17 |
| MDPmv + DM | 4.648 | 6.847 | 4540 | 5.612 | 67.369 |
| MDPmv + CH | 4.648 | 6.847 | 4540 | 5.612 | 67.369 |
| KSP | - | - | - | - | - |

**Table 6: Scores on "basketball" sequence (setting 2).**

feature has an inconsistent results with the two settings. Indeed, it performs poorly in the Setting 1, but reaches the performance level of the other features in Setting 2. Meanwhile, the color histogram appears to be a simple, stable and good appearance feature, reaching high scores in all settings. The increase on MOTA scores in the case of multi-camera tracking can be seen, but the impact of the multi-camera algorithm including all the features variants considered on the CLEAR MOT metric is not significant. This was somehow expected as the main motivation behind the proposed method is to increase identity robustness.

**Comparison of different trajectory distances.** We study trajectory distance functions in both settings 1 and 2. The IDF1 and MOTA mean values, over all appearance features, are shown in Table 7. The reported values mark that the Setting 2 notably helps the multiple-camera approach increase its performance, as opposed to Setting 1. Within the setting 2, our algorithm shows the relatively same scores with different appearance features. That means the additional spatial constraint, imposed in Setting 2, produces a stabilization with respect to other features. The Tab. 7 also displays the average scores, over all views, obtained when using the pointwise and DTW trajectory distances. The overall results indicate, in this case, that the improvement provided by DTW is not significant.

## 6 CONCLUSION

In conclusion, we introduced a novel distance function combining the trajectory and appearance affinities in order to collaboratively exploit the tracking results between cameras. Moreover, we studied the impact of multiple appearance features, including hand-crafted and deep learning approaches, on tracking performance. Our approach is validated on the well-known PETS2009 and EPFL datasets with the experimental results showing a significant improvement in preserving the identity of targets on the condition that the camera system acquires sufficiently overlapping zone.

| Sequence | Path error | Setting | IDF1 | MOTA |
|---|---|---|---|---|
| PETS09-S2L1 | pointwise | 1 | 62.31 | 69.24 |
| | | 2 | 66.05 | 70.42 |
| | DTW | 1 | 62.77 | 69.36 |
| | | 2 | 65.89 | 70.59 |
| EPFL/terrace1 | pointwise | 1 | 15.74 | 45.64 |
| | | 2 | 15.69 | 46.87 |
| | DTW | 1 | 15.69 | 45.53 |
| | | 2 | 16.27 | 46.52 |
| EPFL/basketball | pointwise | 1 | 5.44 | 5.61 |
| | | 2 | 5.44 | 5.61 |
| | DTW | 1 | 5.44 | 5.61 |
| | | 2 | 5.44 | 5.61 |

**Table 7: Overall comparison of different settings.**

# REFERENCES


[1] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. 2008. People-tracking-by-detection and people-detection-by-tracking. In *2008 IEEE Conference on computer vision and pattern recognition*. IEEE, 1–8.

[2] Jerome Berclaz, Francois Fleuret, Engin Turetken, and Pascal Fua. 2011. Multiple object tracking using k-shortest paths optimization. *IEEE transactions on pattern analysis and machine intelligence* 33, 9 (2011), 1806–1819.

[3] Keni Bernardin and Rainer Stiefelhagen. 2008. Evaluating multiple object tracking performance: the CLEAR MOT metrics. *Journal on Image and Video Processing* 2008 (2008), 1.

[4] William Brendel, Mohamed Amer, and Sinisa Todorovic. 2011. Multiobject tracking as maximum weight independent set. In *CVPR 2011*. IEEE, 1273–1280.

[5] Visesh Chari, Simon Lacoste-Julien, Ivan Laptev, and Josef Sivic. 2015. On pairwise costs for network flow multi-object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5537–5545.

[6] Wongun Choi. 2015. Near-online multi-target tracking with aggregated local flow descriptor. In *Proceedings of the IEEE International Conference on Computer Vision*. 3029–3037.

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

[8] Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perona. 2014. Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 8 (2014), 1532–1545.

[9] Loïc Fagot-Bouquet, Romaric Audigier, Yoann Dhome, and Frédéric Lerasle. 2016. Improving multi-frame data association with sparse representations for robust near-online multi-object tracking. In *European Conference on Computer Vision*. Springer, 774–790.

[10] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. 2010. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence* 32, 9 (2010), 1627–1645.

[11] J Ferryman and A Shahrokni. 2009. Pets2009: Dataset and challenge. In *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on*. IEEE, 1–6.

[12] Francois Fleuret, Jerome Berclaz, Richard Lengagne, and Pascal Fua. 2008. Multicamera people tracking with a probabilistic occupancy map. *IEEE transactions on pattern analysis and machine intelligence* 30, 2 (2008), 267–282.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[14] Alexander Hermans, Lucas Beyer, and Bastian Leibe. 2017. In Defense of the Triplet Loss for Person Re-Identification. *CoRR* abs/1703.07737 (2017). arXiv:1703.07737 http://arxiv.org/abs/1703.07737

[15] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. 2012. Tracking-learning-detection. *IEEE transactions on pattern analysis and machine intelligence* 34, 7 (2012), 1409–1422.

[16] Margret Keuper, Siyu Tang, Yu Zhongjie, Bjoern Andres, Thomas Brox, and Bernt Schiele. 2016. A multi-cut formulation for joint segmentation and tracking of multiple objects. *arXiv preprint arXiv:1607.06317* (2016).

[17] Sohaib Khan and Mubarak Shah. 2003. Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25, 10 (2003), 1355–1360.

[18] Chanho Kim, Fuxin Li, Arridhana Ciptadi, and James M Rehg. 2015. Multiple hypothesis tracking revisited. In *Proceedings of the IEEE International Conference on Computer Vision*. 4696–4704.

[19] Quoc Cuong Le, Donatello Conte, and Moncef Hidane. 2018. Online Multiple View Tracking: Targets Association Across Cameras. In *6th Workshop on Activity Monitoring by Multiple Distributed Sensing (AMMDS 2018)*.

[20] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler. 2015. MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking. *arXiv:1504.01942 [cs]* (April 2015). http://arxiv.org/abs/1504.01942 arXiv: 1504.01942.

[21] Ivana Mikic, Simone Santini, and Ramesh Jain. 1998. Video processing and integration from multiple cameras. In *Proceedings of the 1998 Image Understanding Workshop, Morgan-Kaufman, San Francisco*, Vol. 6.

[22] Roman Pflugfelder and Horst Bischof. 2010. Localization and trajectory reconstruction in surveillance cameras with nonoverlapping views. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 4 (2010), 709–721.

[23] Hamed Pirsiavash, Deva Ramanan, and Charless C Fowlkes. 2011. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR 2011*. IEEE, 1201–1208.

[24] Vladimir Reilly, Haroon Idrees, and Mubarak Shah. 2010. Detection and tracking of large number of targets in wide area surveillance. In *European conference on computer vision*. Springer, 186–199.

[25] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. 2016. Deepmatching: Hierarchical deformable dense matching. *International Journal of Computer Vision* 120, 3 (2016), 300–323.

[26] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. 2016. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision*. Springer, 17–35.

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

[28] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. 2017. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. *arXiv preprint arXiv:1701.01909* 4, 5 (2017), 6.

[29] Hiroaki Sakoe and Seibi Chiba. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing* 26, 1 (1978), 43–49.

[30] Hiroaki Sakoe, Seibi Chiba, A Waibel, and KF Lee. 1990. Dynamic programming algorithm optimization for spoken word recognition. *Readings in speech recognition* 159 (1990), 224.

[31] Aswin C Sankaranarayanan, Ashok Veeraraghavan, and Rama Chellappa. 2008. Object detection, tracking and recognition for multiple smart cameras. *Proc. IEEE* 96, 10 (2008), 1606–1624.

[32] Guang Shu, Afshin Dehghan, Omar Oreifej, Emily Hand, and Mubarak Shah. 2012. Part-based multiple-person tracking with partial occlusion handling. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1815–1821.

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

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

[35] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. 2017. Multiple people tracking by lifted multicut and person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3539–3548.

[36] Yonatan Tariku Tesfaye, Eyasu Zemene, Andrea Prati, Marcello Pelillo, and Mubarak Shah. 2017. Multi-target tracking in multiple non-overlapping cameras using constrained dominant sets. *arXiv preprint arXiv:1706.06196* (2017).

[37] Xiaogang Wang. 2013. Intelligent multi-camera video surveillance: A review. *Pattern recognition letters* 34, 1 (2013), 3–19.

[38] Christian Wojek, Bernt Schiele, and Pietro Perona. 2009. Pedestrian detection: A benchmark. In *in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. Citeseer.

[39] Yu Xiang, Alexandre Alahi, and Silvio Savarese. 2015. Learning to track: Online multi-object tracking by decision making. In *2015 IEEE international conference on computer vision (ICCV)*. IEEE, 4705–4713.

[40] Yihong Xu, Yutong Ban, Xavier Alameda-Pineda, and Radu Horaud. 2019. DeepMOT: A Differentiable Framework for Training Multiple Object Trackers. *arXiv preprint arXiv:1906.06618* (2019).

[41] Amir Roshan Zamir, Afshin Dehghan, and Mubarak Shah. 2012. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *Computer Vision–ECCV 2012*. Springer, 343–356.

[42] Li Zhang, Yuan Li, and Ramakant Nevatia. 2008. Global data association for multi-object tracking using network flows. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 1–8.

[43] Ji Zhu, Hua Yang, Nian Liu, Minyoung Kim, Wenjun Zhang, and Ming-Hsuan Yang. 2018. Online multi-object tracking with dual matching attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 366–382.