Unbalanced Optimal Transport in Multi-Camera Tracking Applications

Quoc Cuong Le, Donatello Conte, Moncef Hidane

Unbalanced Optimal Transport in Multi-Camera Tracking Applications

Quoc Cuong LE[1], Donatello CONTE[1], and Moncef HIDANE[2]

[1] Université de Tours
Laboratoire d'Informatique Fondamentale et Appliquée de Tours, EA - 6300
quoccuong.le@etu.univ-tours.fr, donatello.conte@univ-tours.fr
[2] INSA Centre Val de Loire
Laboratoire d'Informatique Fondamentale et Appliquée de Tours, EA - 6300
moncef.hidane@insa-cvl.fr

Abstract. Multi-view multi-object tracking algorithms are expected to resolve multi-object tracking persistent issues within a single camera. However, the inconsistency of camera videos in most of the surveillance systems obstructs the ability of re-identifying and jointly tracking targets through different views. As a crucial task in multi-camera tracking, assigning targets from one view to another is considered as an assignment problem. This paper is presenting an alternative approach based on Unbalanced Optimal Transport for the unbalanced assignment problem. On each view, targets' position and appearance are projected on a learned metric space, and then an Unbalanced Optimal Transport algorithm is applied to find the optimal assignment of targets between pairs of views. The experiments on common multi-camera databases show the superiority of our proposal to the heuristic approach on MOT metrics.

Keywords: Multi-object Tracking · Multi-view Tracking · Unbalanced Optimal Transport.

1 Introduction

Multiple Object Tracking (MOT) is still one of the most challenging and vital problems in computer vision. Therein, the goal is to determine the position and identity of a variable number of targets throughout video frames. In the past recent years, the rise of deep learning approaches [17] has led to an increase in the performance, robustness and reliability of *Single* Object Tracking (SOT) algorithms. Implementing these SOT trackers to track multiple objects simultaneously, however, appears challenging for many typical reasons, such as initialization step at every frame, interactions between targets causing frequent mutual occlusions and identity switches.

A popular approach to track multiple objects is to adopt the *tracking-by-detections* paradigm ([1, 40]), relying on detections at every frame. This approach has been reinforced in the recent years since many powerful object detection algorithms ([16, 29, 23]), e.g., Faster R-CNN, Mask R-CNN, YOLO, SSD, have

emerged and even outperformed humans in the past recent years. In principle, the detection-based MOT methods directly link together detections which belong to targets, on the entire videos, in order to form their final trajectories. *Tracking-by-detection* MOT methods lead to *data association algorithms*, often formulated as a global optimization problems in which a graph-based representation of detections with edges weighted by a distance (or similarity) is adopted. The distance between detections mostly includes Euclidean distance, time delay, and appearance affinity [40, 2, 35, 32]. The online/offline distinguishes between methods that solely use results from previous detections, and the ones that consider the whole (or batch) time-sequence in order to compute data associations.

Both SOT-based and *online* association-based methods require an efficient way to control the state of targets in order to prevent missing tracks, occlusions, and identity switches. For SOT-based methods, Markov Decision Processes (MDP) were adopted in the papers [38, 42] to tackle this issue. This approach has been extended to an overlapping multiple camera setting in [21] and has shown capability in allowing the individual cameras to recapture/re-identify their lost targets.

Within multiple camera systems, Multi-Camera Multi-Object Tracking (MCMOT) or Multi-Target Multi-Camera Tracking (MTMCT) problem is frequently formulated as an assignment problem or, in many cases, the re-identification/ recognition problems as the object of interest is mainly human or transport vehicle. Since data association approaches are extendable in multi-camera cases, most of MCMOT algorithms of the state-of-the-art are mainly derived from single-camera association-based MOT approaches. Indeed, the role of multi-camera tracking is to link detections or tracklets *across* cameras in the network. Generally, associating targets between two views is formulated as assignment problem, or bi-graph matching problem, which is originally resolved by Hungarian or Munkres algorithms. This is not the case of MOT because of the varying target number. As a solution to this issue, the modified version of the Munkres algorithm is used with virtual targets.

In our case study, we address the target association problem between different views within an overlapping camera system for online multi-camera applications. The target matching is well defined by the unbalanced assignment problem, in which the number of targets in one view is not equal to those in another view. In this paper, we propose a novel assignment approach formulated as an unbalanced optimal transport problem for multi-view tracking applications. Our second contribution is to develop a deep distance learning framework for Optimal Transport. Our third contribution is to adopt the target association between two cameras within multiple camera systems. Our multi-camera tracking framework is functional with mere pairs of cameras, which is called as "dual-camera" approach in the multi-camera tracking problem (Fig. 1). Our approach helps elevate the all-camera condition, renders it more flexible to any number of cameras inside the camera system, and essentially adapts well with our proposed assignment problem at the early of this paper.

The structure of this paper is as follows: we first mention the works related to ours; secondly, we describe the formulation of our dual-camera target association problem as unbalanced optimal transport; then introduce our proposed distance learning method of Optimal Transport based on a deep neural network; and finally, we present our experimental results showing the advantage of our method.

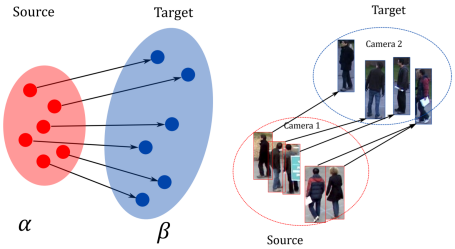

Fig. 1. (a) Assignment between two distributions as Optimal Transport problem (b) Target assignment across cameras in multiple cameras tracking application

2 Related Works

2.1 Single view multi-object tracking

Since handling multiple different targets is the main work of MOT algorithms, the tracking-by-detection paradigm has evolved as their major approach. This is especially true as the result of the advent of high-performing category detectors. Tracking-by-detection approaches can be sorted into the following two groups.

Offline approaches Following the tracking-by-detection paradigm [40, 2], graph optimization problems are formulated to create links between detections of targets, in successive frames, then the chain of detections through frames determines the full trajectory of a target. These methods have become popular because they simplified the classic issues mentioned above, such as trackers management, interaction, initialization, and update. The data association problem is formulated via a graph whose nodes represent the detections/features and whose edges are weighted by the distance (or similarity) between detections. 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 [41], and relying on criteria such as Graph Clique [40, 36], Graph Multicut [18, 31], Network Flow [41, 27, 2, 7], Maximum Weight Independent Set [5, 20, 8]

Online approaches To fulfill the need for immediate tracking results in many applications, numerous papers proposed online tracking methods [1, 27, 32, 11, 42]. 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 [32, 28, 42], 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 first to current frames of videos (i.e., causal system), or several methods can be considered as "near-online" methods such as [8, 31], as their offline optimization process applies on a window of frames at the time in videos, which causes the delay on tracking results. Alternatively, the tracking-by-detection strategy and multiple SOT algorithms are combined to benefit from the SOT trackers, and the ability to recover lost targets of data association approaches in [38, 42], classified as SOT-based approaches.

2.2 Multi-view multi-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 cover the observation of the objects fully. 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 made the hypothesis that the exact position of each camera is already known, and camera calibration has been done before applying the tracking process. In the tracking phase, the trackers implemented on different cameras usually pool their results with 3-D coordination via projection from the image plane to ground plane in the real world [24, 33, 26]. This allows combining the different results, and in particular, reconnecting detections/tracklets to missing targets. Meanwhile, K-shortest path (KSP) [15] only uses detections from all cameras to first detect targets' positions on the ground via a POM (Probabilistic Occupancy Map), then perform tracking later.

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 [35, 31, 18], graph cliques [40, 10], network flow [41, 27, 7]. Meanwhile, the other data association methods including bipartite matching [1, 32] and independent set [5, 20] 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 [36, 19] generalize multi-camera tracking into two main steps: MOT on every single view, then linking the trajectories across cameras. Unfortunately, none of those mentioned methods perform online. Recently, Le et al. [21] introduced

an online multi-camera tracking based on data association on each processing frame. In the next section, we introduce our dual-camera tracking approach in a multi-camera setting based on unbalanced optimal transport to handle hard occlusions and prevents identity switches. Our strategy is to assign targets from one to another view with the help of Deep Neural Nets. In literature, there are several approaches that have the same initiative to combine deep neural nets and optimization methods on which gradients are backpropagated such as [39, 4]. However, to the best of our knowledge, our paper is the first one applying this strategy in MOT with the multiple overlapping cameras.

3 Proposed Method

3.1 Targets Association Across Cameras as an Unbalanced Optimal Transport Problem

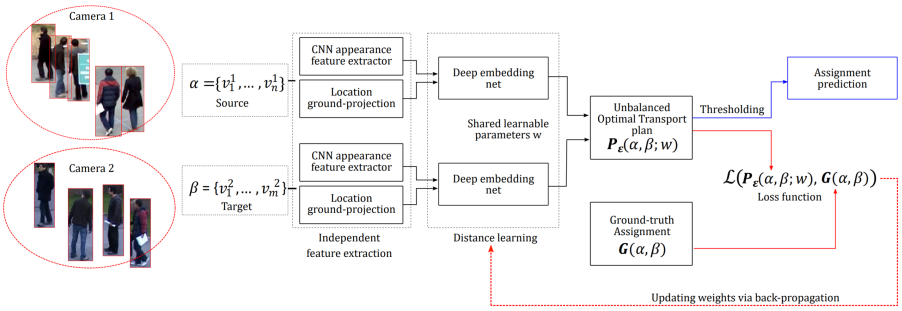

Fig. 2. The pipeline of our distance learning framework. The red arrow indicates the direction during training process, meanwhile the blue lines for testing.

This section describes in detail our approach to solving target association across cameras via optimal transport. Within a frame-synchronized, overlapping camera network, associating targets between different cameras emerges as the main issue for collaborative tracking.

Let us start by considering the case of a network consisting of two cameras C_1 and C_2. At a given *frame index* F, we have $\{v^1_{1,F}, \dots, v^1_{n,F}\}$ targets detected in Camera C_1 and $\{v^2_{1,F}, \dots, v^2_{m,F}\}$ targets detected in Camera C_2. In order to simplify our notations, we drop the F subscript in sequel. In general, $n \neq m$, since some targets can be seen by only one camera. This can happen either because a given target occupies a position that does not belong the common field of view or, more crucially, because of occlusion.

Each detection v_i^k, $k \in \{1, 2\}$ is characterized by an *feature vector* generally consisting of an *appearance vector*, extracted from the bounding box provided

by the detector, and the target's position. The current practice leverages the capability of recent deep convolutional neural networks to extract useful appearance features from the bounding boxes provided by the detectors, e.g. by using VGG [34] or ResNet [17] as a 'backbone'. For the target's position, it is necessary to define a common coordinate system, where the position of targets can be converted into the same measure unit. For pedestrian tracking, this issue is usually resolved by projecting the target's feet point on the image into the ground plane via the homography matrix of the camera. This is the solution we retain in our current setting.

These feature vectors allow to define a cost matrix $\mathbf{C} \in \mathbb{R}^{n \times m}$, whose entry $\mathbf{C}_{i,j}$ defines the cost of associating target v_i^1 to target v_j^2. The matrix C allows, in turn, to formulate the problem of target association between Cameras C_1 and C_2, at frame index F, as a, possibly unbalanced, *assignment problem*. These problems amount to solving integer linear programs, using either combinatorial algorithms such as the Hungarian or the auction algorithm, or, ignoring the integer constraints, continuous linear programming.

For associating targets across *different* cameras, the definition of an appropriate cost matrix poses two serious problems. The first one is related to the definition of appearance features. These features should incorporate some kind of invariance with respect to the different cameras, that is, the appearance feature of the same target computed through two different cameras should be close. This invariance is not necessarily enforced when using popular convolutional networks such as VGG or ResNet. The second issue is related to the combination of the appearance features and the position, the appearance features being generally in the range $[0, 1]$, while the position extending to the whole field of view.

In order to solve the issues raised by the two previous problems, we propose to adopt a learning-based approach, where *the appearance features and their combination with the target's position* are learned from a set of examples. The training data in this case are generated from the training sequences of the datasets we consider. More precisely, we extract from each training video frame the provided bounding boxes and the corresponding ground-truth assignments. With this training set at hand, we aim at *end-to-end gradient-based learning*, that is, the empirical loss that we minimize for learning should be related to the assignment task we consider, and implemented by (automatically computed) gradient descent.

Using combinatorial algorithms such as the Hungarian or auction algorithms rules out the possibility of using automatic differentiation engines for performing gradient descent. Furthermore, even when ignoring integer constraints, linear programming solvers can hardly be differentiated, since their solutions are not unique. To deal with this problem, we follow a recent line of works [9] by considering the natural relaxation of the assignment, namely the optimal transport problem and its entropic regularization [25].

In our formulation of assignment problem of targets in two views via Optimal Transport, the sets of targets $v_i^{1^n}$ in one view is being matched with those $v_i^{2^m}$

in other view. Those two sets represents two empirical distributions: one *Source* and one *Target*, supported on a feature space $\mathcal{X}$.

$$\alpha := \sum_{i=1}^{n} \mathbf{a}_i \delta_{x_i}, \quad \beta := \sum_{j=1}^{m} \mathbf{b}_j \delta_{y_j}, \tag{1}$$

where δ_x is the Dirac at $x \in \mathcal{X}$ and $\mathbf{a}_i$ and $\mathbf{b}_j$ are the corresponding weights. In our setting, we will consider uniform discrete measures, which is, all the components of a weight vector are equal.

The optimal transport between *source* ans *target* is represented by an *optimal transport plan* $\mathbf{P}$, which minimizes the following transportation cost:

$$\mathbf{L}_{\mathbf{C}}(\mathbf{a}, \mathbf{b}) := \min_{\mathbf{P} \in \mathbf{U}(\mathbf{a}, \mathbf{b})} \langle \mathbf{C}, \mathbf{P} \rangle = \min_{\mathbf{P} \in \mathbf{U}(\mathbf{a}, \mathbf{b})} \sum_{i,j} \mathbf{C}_{ij} \mathbf{P}_{ij}, \tag{2}$$

where $\mathbf{C}$ is the ground cost matrix, whose elements $\mathbf{C}_{ij}$ are the pairwise distance between the Dirac δ_{x_i} of the source measure α and those δ_{y_j} of the target measure β, and $\mathbf{U}(\mathbf{a}, \mathbf{b})$ is a coupling from the *source* $\mathbf{a}$ to the *target* $\mathbf{b}$. The feasible couplings are defined by a set of coupling matrices $\{\mathbf{P} \in \mathbb{R}_+^{n \times m}\}$, where $\mathbf{P}_{ij}$ depicts the amount of mass flowing from x_i toward y_j, under the mass preservation constraint.

$$\mathbf{U}(\mathbf{a}, \mathbf{b}) = \{\mathbf{P} \in \mathbb{R}_+^{n \times m} : \mathbf{P}\mathbb{1}_m = \mathbf{a} \,\text{and}\, \mathbf{P}^T \mathbb{1}_n = \mathbf{b}\}. \tag{3}$$

The Optimal Transport (OT) problem with entropic regularization has a dual form following [13]:

$$\min_{\mathbf{P} > 0} \max_{(\mathbf{f}, \mathbf{g}) \in \mathbb{R}^n \times \mathbb{R}^m} \langle \mathbf{C}, \mathbf{P} \rangle - \varepsilon \mathbf{H}(\mathbf{P}) + \langle \mathbf{a} - \mathbf{P}\mathbb{1}_m, \mathbf{f} \rangle + \langle \mathbf{b} - \mathbf{P}^T \mathbb{1}_n, \mathbf{g} \rangle \tag{4}$$

where the set of admissible dual variables (called *potentials*) ($\mathbf{f} \in \mathbb{R}^n, \mathbf{g} \in \mathbb{R}^m$). The optimal transport plan solved via the dual problem (4) has a closed-form [13]:

$$\mathbf{P}^\star = \pi = \exp\left(\frac{1}{\varepsilon}\left(\mathbf{f} \oplus \mathbf{g} - \mathbf{C}\right)\right).(\alpha \otimes \beta), \tag{5}$$

where $\mathbf{f} \oplus \mathbf{g}$ is denoted as a sum matrix of 2 vectors f and g whose cell $\{\mathbf{f} \oplus \mathbf{g}\}_{ij}$ is equal to $\mathbf{f}_i + \mathbf{g}_j$, in the same manner, $\alpha \otimes \beta$ is also denoted as a product matrix of 2 vectors α and β whose cell $\{\alpha \otimes \beta\}_{ij}$ is equal to $\alpha_i \beta_j$.

3.2 Ground cost learning for UOT-based targets association across cameras

This section describes our proposed deep distance learning framework, which helps to compute an appropriate distance for the Optimal Transport problem between targets of one camera and those of another. More concretely, on each camera, the appearance feature of each target is extracted from its image patch via a deep convolutional neural network, e.g. VGG [34], ResNet [17]. Meanwhile,

its position x is determined by projecting the target's feet point on the image into the ground plane via the homography matrix of the camera. Both the appearance and location feature vectors are the input of a deep neural network whose output is an embedding in a feature space $\mathcal{X}$. The collection of all points mapped from all targets of a camera via the deep neural net generates a distribution. As discussed in the previous section, given two distributions originated from the targets of a pair of cameras, target association in a pair of two cameras is an Unbalanced Optimal Transport from a distribution, called *source*, to the other one, called *target*. Therefore, the Optimal Transport plan is followed by a thresholding step, to obtain a binary matrix as the association matrix of targets between the two cameras. Fig. 2 displays the pipeline of our distance learning framework, which aims to learn ground cost between targets across cameras so that the optimal transport plan approximates the ground-truth assignment.

Because our deep-learning-based method is a supervised learning approach, it is required a training data with labels. Our training data are directly generated from the training sequences of a dataset. Precisely, for each frame of videos, every pair of cameras gives a single assignment as the label of a sample, while the data of the sample is extracted from the ground-truth bounding boxes via the deep extracting feature net. In the case of N cameras in the network, the combination of camera possible pairs is $N(N-1)/2$, which is also the number of samples generated in each frame instant.

In our formulation, for each target i, given $\Phi_i \in \mathbb{R}^{2048}$ (i.e., output of ResNet50 backbone [17]) and $x_i \in \mathbb{R}^2$ (i.e., target coordinate on ground), the embedding function f_w, via our deep neural network (see Fig. 3), projects the appearance feature and location of target i into the feature space $\mathcal{X}$,

$$f_w : (\Phi, x) \rightarrow \mathcal{X},$$

where w is the parameters of the deep neural net. As a result of unbalanced optimal transport, the transport plan shows the mass flows from point i of *source* to point j of *target*. Based on the properties of optimal transport [25], any pair of close points distributions *source* and *target* results in a significant mass flow on its transport plan compared with others. Fig. 4 (a) is an optimal transport plan in which i^{th} row represents the mass of the i^{th} *source* point being transferred to all *target*. Since only consistent mass transfers from one point on *source* to a unique point in *target* is sought, the optimal transport plan between *source* and *target* is expected to be well "sparse", which means that the matching can be deduced straightforwardly (see Fig. 4 (b)) by thresholding the optimal transport plan. We then can obtain the assignment from *source* to *target*.

In terms of optimization, the dissimilarity between the optimal transport plan $\mathbf{P}_\varepsilon(\alpha, \beta)$ and the ground-truth assignment $\mathbf{G}(\alpha, \beta)$ is measured by a loss function $\mathcal{L}$. The learnable parameters w of our neural net is then determined via a minimization problem:

$$w = \arg\min_w \mathcal{L}\left(\mathbf{P}_\varepsilon\left(.; w\right), \mathbf{G}(.)\right) \tag{6}$$

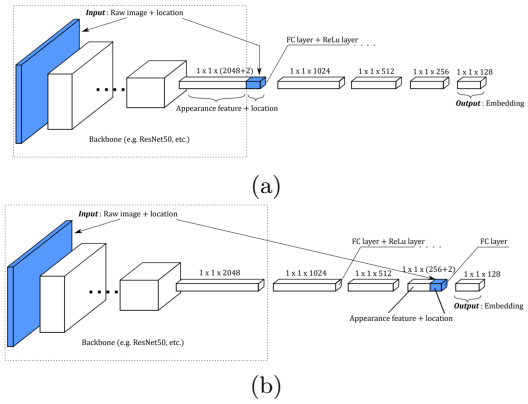

Fig. 3. Proposed distance learning neural net. The neural net consists of a CNN backbone (e.g. ResNet50 in our case), which extract appearance features from raw image, and a series of Fully Connected (FC) layers with ReLU layers as activations. Model (a) with locations at the bottom of the deep distance network, meanwhile, model (b) with locations at the second last FC layer.

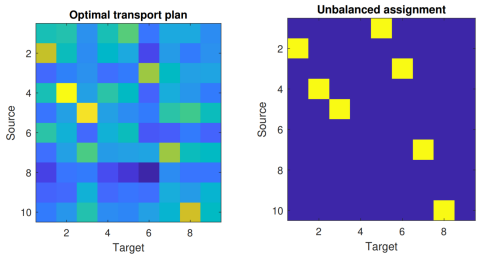

Fig. 4. Comparison between optimal transport plan (a) and assignment matrix (b)

The loss functions in our framework are formulated as following. Given two sets of targets $\left\{v_i^{k_1}\right\}_n$ and $\left\{v_j^{k_2}\right\}_m$, each belongs to a single camera, the assignment task is to find the correspondence of common targets in the pair of cameras k_1 and k_2 while excluding the targets which can be seen in only one view. Given $\mathbf{P}_\varepsilon \in \mathbb{R}_+^{n \times m}$ the transport plan and $\mathbf{G} \in \{0,1\}^{n \times m}$ the ground-truth assignment, we propose our loss which is delivered from the dual problem of regularized optimal transport problem (4). Therefore, the first order condition to reach the optimal solution [25] yields to:

$$\log(\mathbf{P}_{ij}) = \frac{\mathbf{f}_i + \mathbf{g}_j - \mathbf{C}_{ij}}{\varepsilon} \tag{7}$$

In the training phase of our experiments, by default, both total masses of the measure α and β are equal 1. The constraint of mass conservation in the Bal-

anced Optimal Transport problem leads to the sum of all elements of the optimal transport plan $\mathbf{P}$ smaller or equal 1. The equality happens in the case of the balanced optimal transport, and the inequality for the Unbalanced Optimal Transport case. As a result, the ground-truth assignment $\mathbf{G}$ needs to be normalized to keep the assignment matrix and optimal transport plan comparable. Hence, from the expression of the transport plan (7), our loss is formulated as follows:

$$\mathcal{L}\left(\mathbf{P}_{\varepsilon}, \mathbf{G}\right) = \sum_{i,j} \left| \log(\mathbf{G}'_{ij}) - \max\left(\frac{\mathbf{f}_i + \mathbf{g}_j - \mathbf{C}_{ij}}{\varepsilon}, \log\sigma\right)\right| \tag{8}$$

where the normalized assignment coupling $\mathbf{G}_{i,j}$ is defined as

$$\mathbf{G}'_{ij} = \frac{\mathbf{G}_{ij}}{\sum_{i,j}\mathbf{G}_{ij} + \gamma} + \sigma \geq \sigma \tag{9}$$

with σ is a tiny threshold value, and γ is a normalization constant. This threshold value is added to avoid the logarithm of zero value in the loss function (8) and to set a margin for any near-zero transport, which does not contribute to the distance loss if its value is extremely low.

Additionally, the parameters of our neural net w are updated iteratively via minimizing the loss $\mathcal{L}$. The derivation of the loss function to the net parameters $\partial\mathcal{L}/\partial w$ is computed via back-propagation, which occurs after every optimal transport of a *source-target* pair from two cameras.

4 Experimental Results

4.1 Implementations

In our implementations, we build two versions of distance learning neural nets in order to compute the source-target distance in the Optimal Transport:

(a) The appearance feature of targets obtained from the backbone of ResNet50, in addition to their location, is considered as the inputs of our distance learning network. Our deep network is a series of Full Connected Layers with ReLU layer on the top of each. The outputs of FC layers are respectively 1024, 512, 256 and 128 (see Fig. 3 (a)).
(b) The second model is modified from the original one, but instead of using locations in the first layer, it is concatenated with the second last output layer. The intuition behind is to emphasize the location feature of targets, because, in tracking applications, positions of targets are crucial to the performance of tracking algorithm (see Fig. 3 (b)).

In the deployment phase, numerous experiments with different configurations are conducted within the framework of multiple camera tracking of the paper [21]. Therefore, the target assignment or matching is applied on only two cameras, collaborative tracking in our multiple camera approach occurs on pairs

of cameras, but one camera can reach all others through the whole tracking process. Precisely, at each frame, each camera consecutively pairs with all other cameras, then within each pair of cameras, an optimal transport plan $\mathbf{C}$ is computed in order to link targets from one camera to the other in the pair. As mentioned in the section 3.1, the value of each cell C_{ij} of the optimal transport plan implies how likely element i of *source* set is matched to element j of *target* set based on the amount of mass being transferred, named *OT value*. Therefore, each missing target on one view is associated with its corresponding target on each other view by an OT value obtained from its optimal transport plan. Hence, the tracking result of the missing target is replaced by the "tracked" target with the *highest* value among its correspondences on all other cameras.

The Optimal Transport algorithm we used in this paper is a public Optimal Transport library [14] on Python with GPU parallelization support, named KeOps-GeomLoss[3]. The parameters of the unbalanced optimal transport problem (4) are set as follows:$p = 2$; "blur" $= 0.5 \rightarrow \varepsilon = blur^p$; "reach" $= 0.1 \rightarrow \tau_1 = \tau_2 = \tau = reach^p$; $D_\varphi = KL$: soft Kulback–Leibler divergence.

Meanwhile, the other parameters in the expression (9) are adjusted for $\sigma = 10^{-8}$ and $\gamma = 10^{-4}$. The value of threshold to convert transport plan to assignment matrix is set equal to 10^{-3}, which is greater than σ in order to reduce the sensibility during testing phrase. The detailed implementation of our deep distance learning method will be available publicly on our project page.

4.2 Benchmarking Performance

This section shows our experimental results verifying the efficiency of the multi-camera MOT algorithm with various appearance features. As a performance evaluation for MOT algorithms, the benchmark MotChallenge [22] has been released with two datasets (MOT15 and MOT16), which contain many single-view video sequences recorded by static or dynamic cameras, and the evaluation metrics of CLEAR MOT [3] and ID measure [30] are used. Additionally, the MotChallenge also provides multiple video sequences, but most of them are not from multiple camera aspect, which requires overlapping zones, synchronization, calibration. Therefore, these datasets, unfortunately, unfit to this case study that focuses on using multiple overlapping views to tackle the targets missing by occlusions. As the multi-camera method aims to improve identity robustness in single views, we will emphasize ID scores in the sequel.

Datasets. In our experiments we used the well-known PETS2009 [12] 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 one main view (from the camera 1) and 4 close-up views (from the cameras 5, 6, 7, and 8) are used. Besides of the sequence "PETS09-S2L1" with 7 cameras, the sequence "PETS09-S2L2" is also available with only 3 cameras. The scenario of surveillance is to track an influx of people moving

[3] https://www.kernel-operations.io/geomloss/

on the roads with different speed, and this makes it far more crowded that "PETS09-S2L1". Since the lack of cameras in this sequence, we set up dual-camera tracking experiments on View 1 and View 2. View 3 is excluded, due to its small impact on the sequence and the absence of ground-truth data as well. On the other hand, the EPFL dataset provides multiple indoor and outdoor video sequences, recording pedestrians by 4 different cameras. Due to the similarity between sequence scenarios, only the sequence "Terrace1" is selected for the experiments. In terms of camera topology, only about $15 - 20\%$ of the observable zones are covered by all cameras in our tracking sequences.

Detection. In all tracking-by-detection approaches, the detector plays an important role in tracking performance. Detections in video frames are generated by the public high-accuracy detectors such as OpenPose [6] and R-CNN [16].

Evaluation metric. To validate the efficiency of our various settings on the multi-camera MOT approach, we adopt the CLEAR MOT metrics 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]. In comparison between MOT scores and ID-measures, all multiple camera approaches slightly improves both MOTA and MOTP scores, regarding to ID-measures. Because the CLEAR MOT metric does not focus on re-identification ability of tracking algorithms [30], while ID-measure scores does. In other words, the significant improvement can be seen on IDF1 and IDP score. As another important indicator for tracking performance in CLEAR MOT metric, IDs score (i.e. identity switches) relates more to ID preservation, which is essential in multiple camera tracking. Therefore, in the following analysis, we measure the impact of methods based on IDF1, IDP and IDs scores rather than MOTA and MOTP.

4.3 Performance analysis

The results shown in the following tables are the average values of all views. Concretely, the overall tracking results of the PETS sequence can be seen in the Table 1, Table 2 and Table 3. Each score column has either a $\uparrow$ or a $\downarrow$ indicating whether better corresponds to higher or lower, respectively. The red color indicates the best score and the blue for the second best.

Primarily, our multi-camera tracking method aims to address hard occlusion problems. It leads to an important reduction of identity switches and a significant improvement of ID measures in comparison with the single-camera method. In the sequence "PETS09-S2L1", the targets have their complex movements and mutual interactions inside the overlapped area of the tracking scene. All the methods using the target trajectory as the features of the affinity measure show the better scores in all categories, in comparison with the approaches, which do not consider historical position record of targets (i.e., trajectories), but only the instant measure including image patch and position of targets. In detail, the method with full camera collaboration (All-cam) [21] shows off its superiority.

Method + Feature	**IDF1↑**	**IDP ↑**	**IDs↓**	**MOTA↑**	**MOTP↑**
Single cam [38]+∅	57.49	62.24	333	68.44	68.83
All cam [21]+path	72.8	78.53	98	73.26	70.69
KSP [15]+∅	21.51	18.16	812	-29.63	64.27
Dual-cam [21]+path	67.96	72.72	126	73.4	70.65
Dual-cam UOT+DL (a)	68.15	73.41	153	73.16	70.81
Dual-cam UOT+DL (b)	66.71	71.73	174	72.19	70.65
Dual-cam UOT+pos	66.04	70.66	163	72.76	70.60

Table 1. Scores on "PETS09-S2L1" multi-camera sequence.

Meanwhile, our Unbalanced Optimal Transport approach (UOT) is less robust, but still significantly improves tracking scores compared to single-camera approach. Notwithstanding, in the tests with the sequence "EPFL/terrace1", the tracking scene composes 8 identities moving mainly around a relatively small area covered by a smaller camera number, which makes the scene more crowded and targets hardly seen by all cameras. Consequently, the original approach [21] failed to improve tracking results, because, with a smaller camera amount, it is obviously less probable that there are more than 2 or 3 cameras observed the same target at the same time. The results in Tab. 2 show that all other approaches with dual-camera mode perform significantly better than the original ones. The next remark is that in the scenario where there are only short trajectories that can be seen, the trajectory feature is less reliable. In other words, shorter trajectories, less effective the original approach is. Hence, in Tab. 2, our distance learning method based on Optimal Transport outweighs the conventional approaches which only use position or trajectory of target as input feature for affinity measure.

Method + Feature	**IDF1↑**	**IDP ↑**	**IDs↓**	**MOTA↑**	**MOTP↑**
Single cam [38]+∅	21.88	25.66	388	55.98	72.53
All cam [21]+path	21.32	25.05	461	54.14	72.47
KSP [15]+∅	25.85	23.51	695	19.57	62.26
Dual-cam [21]+path	23.23	26.72	382	56.71	72.43
Dual-cam UOT+DL (a)	24.36	31.40	305	46.86	72.91
Dual-cam UOT+DL (b)	25.15	28.88	385	56.93	72.63
Dual-cam UOT+pos	22.00	25.32	381	56.40	72.48

Table 2. Scores on "terrace1" multi-camera sequence.

Secondly, the KSP method performs poorly on the sequence PETS09-S2L1, but gives a greater score on EPFL/terrace1. We can explain that KSP method was developed on the EPFL Multiple View Pedestrian Dataset. In fact, they assume that the targets being observed by cameras system does not leave the scene during their presence. In other words, the targets have to finish their complete trajectories before leaving the scene. The out/in positions of targets is also fixed on the scene, so we can see the actors walking in and out at the

same place. Under these conditions, they came up with the K-shortest path problem where K, which is the number of targets in the tracking videos. On EPFL/terrace1, the algorithm has found 8 paths, which exactly corresponds to 8 targets in the video that help them get the highest IDF1 score. Back to the sequence PETS09, the algorithm cannot deal with the targets that usually went out of and returned to the scene. It only found the longest paths and ignored the targets which appeared in a short period of time and regularly got confused by other targets at the boundary. Moreover, in this database, there is no constrain on where people will appear and disappear on the scene. Apparently, this leads to 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 a grid, called Probabilistic Occupancy Map (POM), the discrete unit size directly affects the accuracy of the tracker. Unfortunately, increasing the size of POM required more iterations to make sure the occupancy map converged correctly.

Method + Feature	**IDF1↑**	**IDP ↑**	**IDs↓**	**MOTA↑**	**MOTP↑**
Single cam [38]+∅	53.46	55.53	321	63.66	75.14
Dual-cam [21]+path	55.63	57.67	327	63.79	75.16
Dual-cam UOT+DL (a)	53.16	55.76	310	62.62	75.16
Dual-cam UOT+DL (b)	53.75	55.83	329	63.51	75.16
Dual-cam UOT+pos	57.30	59.38	312	63.74	74.98

Table 3. Scores on "PETS09-S2L2" dual-camera sequence.

Finally, on the tests with dual-camera sequence "PETS09-S2L2", we excluded the methods which require more than 2 cameras to be operational, including all camera [21] and KSP [2]. As single object trackers can generate long trajectories for targets, trajectories are still an important feature to distinguish targets that we can see in Table 3. The approach [21] with dual-camera only using trajectory as target features archived the second-best result on ID-measures and the best on MOT-scores. Meanwhile, our UOT dual-cam approach based on position only obtained the best scores on ID-measures and the second-best on MOT-scores. Unfortunately, two of our UOT methods using distance learning could not outperform others in this sequence.

5 Conclusion

In this paper, we proposed a novel unbalanced assignment method based on optimal transport to address the target assignment problem between two cameras in an online multi-camera tracking application. A deep metric learning method is introduced with an efficient metric loss function. Our experiments showed the effectiveness of our approach to the multiple camera tracking systems.

References

1. Andriluka, M., Roth, S., Schiele, B.: People-tracking-by-detection and people-detection-by-tracking. In: 2008 IEEE Conference on CVPR. pp. 1–8. IEEE (2008)
2. Berclaz, J., Fleuret, F., Turetken, E., Fua, P.: Multiple object tracking using k-shortest paths optimization. IEEE Trans. on PAMI **33**(9), 1806–1819 (2011)
3. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: the clear mot metrics. Journal on Image and Video Processing **2008**, 1 (2008)
4. Brachmann, E., Rother, C.: Neural-guided ransac: Learning where to sample model hypotheses. In: ICCV. pp. 4322–4331 (2019)
5. Brendel, W., Amer, M., Todorovic, S.: Multiobject tracking as maximum weight independent set. In: CVPR 2011. pp. 1273–1280. IEEE (2011)
6. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: CVPR (2017)
7. Chari, V., Lacoste-Julien, S., Laptev, I., Sivic, J.: On pairwise costs for network flow multi-object tracking. In: CVPR. pp. 5537–5545 (2015)
8. Choi, W.: Near-online multi-target tracking with aggregated local flow descriptor. In: ICCV. pp. 3029–3037 (2015)
9. Cuturi, M., Teboul, O., Vert, J.P.: Differentiable ranks and sorting using optimal transport. arXiv preprint arXiv:1905.11885 (2019)
10. Dehghan, A., Modiri Assari, S., Shah, M.: Gmmcp tracker: Globally optimal generalized maximum multi clique problem for multiple object tracking. In: CVPR. pp. 4091–4099 (2015)
11. Fagot-Bouquet, L., Audigier, R., Dhome, Y., Lerasle, F.: Improving multi-frame data association with sparse representations for robust near-online multi-object tracking. In: ECCV. pp. 774–790. Springer (2016)
12. Ferryman, J., Shahrokni, A.: Pets2009: Dataset and challenge. In: PETS-Winter. pp. 1–6. IEEE (2009)
13. Feydy, J., Séjourné, T., Vialard, F.X., Amari, S.I., Trouvé, A., Peyré, G.: Interpolating between optimal transport and mmd using sinkhorn divergences. arXiv preprint arXiv:1810.08278 (2018)
14. Feydy, J., Séjourné, T., Vialard, F.X., Amari, S.i., Trouve, A., Peyré, G.: Interpolating between optimal transport and mmd using sinkhorn divergences. In: The 22nd Int. Conf. on Artificial Intelligence and Statistics. pp. 2681–2690 (2019)
15. Fleuret, F., Berclaz, J., Lengagne, R., Fua, P.: Multicamera people tracking with a probabilistic occupancy map. IEEE Trans. on PAMI **30**(2), 267–282 (2008)
16. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proc. of the IEEE int. Conf. on Computer Vision. pp. 2961–2969 (2017)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
18. Keuper, M., Tang, S., Zhongjie, Y., Andres, B., Brox, T., Schiele, B.: A multi-cut formulation for joint segmentation and tracking of multiple objects. arXiv preprint arXiv:1607.06317 (2016)
19. Khan, S., Shah, M.: Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. IEEE Trans. on PAMI **25**(10), 1355–1360 (2003)
20. Kim, C., Li, F., Ciptadi, A., Rehg, J.M.: Multiple hypothesis tracking revisited. In: ICCV. pp. 4696–4704 (2015)
21. Le, Q.C., Conte, D., Hidane, M.: Online multiple view tracking: Targets association across cameras. In: 6th Workshop on AMMDS (2018)

22. Leal-Taixé, L., Milan, A., Reid, I., Roth, S., Schindler, K.: MOTChallenge 2015: Towards a benchmark for multi-target tracking. arXiv:1504.01942 [cs] (Apr 2015)
23. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: ECCV. pp. 21–37. Springer (2016)
24. Mikic, I., Santini, S., Jain, R.: Video processing and integration from multiple cameras. In: Proce. of the 1998 Image Understanding Workshop. vol. 6 (1998)
25. Peyré, G., Cuturi, M., et al.: Computational optimal transport. Foundations and Trends® in Machine Learning **11**(5-6), 355–607 (2019)
26. Pflugfelder, R., Bischof, H.: Localization and trajectory reconstruction in surveillance cameras with nonoverlapping views. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(4), 709–721 (2010)
27. Pirsiavash, H., Ramanan, D., Fowlkes, C.C.: Globally-optimal greedy algorithms for tracking a variable number of objects. In: CVPR. pp. 1201–1208 (2011)
28. Reilly, V., Idrees, H., Shah, M.: Detection and tracking of large number of targets in wide area surveillance. In: ECCV. pp. 186–199. Springer (2010)
29. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
30. Ristani, E., Solera, F., Zou, R., Cucchiara, R., Tomasi, C.: Performance measures and a data set for multi-target, multi-camera tracking. In: European Conference on Computer Vision. pp. 17–35. Springer (2016)
31. Ristani, E., Tomasi, C.: Tracking multiple people online and in real time. In: ACCV. pp. 444–459. Springer (2014)
32. Sadeghian, A., Alahi, A., Savarese, S.: Tracking the untrackable: Learning to track multiple cues with long-term dependencies. arXiv:1701.01909 **4**(5), 6 (2017)
33. Sankaranarayanan, A.C., Veeraraghavan, A., Chellappa, R.: Object detection, tracking and recognition for multiple smart cameras. Proceedings of the IEEE **96**(10), 1606–1624 (2008)
34. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
35. Tang, S., Andres, B., Andriluka, M., Schiele, B.: Subgraph decomposition for multi-target tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5033–5041 (2015)
36. Tesfaye, Y.T., Zemene, E., Prati, A., Pelillo, M., Shah, M.: Multi-target tracking in multiple non-overlapping cameras using constrained dominant sets. arXiv preprint arXiv:1706.06196 (2017)
37. Wang, X.: Intelligent multi-camera video surveillance: A review. Pattern recognition letters **34**(1), 3–19 (2013)
38. Xiang, Y., Alahi, A., Savarese, S.: Learning to track: Online multi-object tracking by decision making. In: 2015 IEEE international conference on computer vision (ICCV). pp. 4705–4713. No. EPFL-CONF-230283, IEEE (2015)
39. Xu, Y., Osep, A., Ban, Y., Horaud, R., Leal-Taixé, L., Alameda-Pineda, X.: How to train your deep multi-object tracker. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6787–6796 (2020)
40. Zamir, A.R., Dehghan, A., Shah, M.: Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In: ECCV 2012, pp. 343–356 (2012)
41. Zhang, L., Li, Y., Nevatia, R.: Global data association for multi-object tracking using network flows. In: CVPR, 2008. pp. 1–8 (2008)
42. Zhu, J., Yang, H., Liu, N., Kim, M., Zhang, W., Yang, M.H.: Online multi-object tracking with dual matching attention networks. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 366–382 (2018)