Multi-camera Tracklet association and fusion using ensemble of visual and geometric cues

Kanishka Nithin, François Bremond

Author affiliation

Multi-camera Tracklet association and fusion using ensemble of visual and geometric cues

Kanishka Nithin, Francois Bremond (Research Director -STARS team), INRIA Sophia Antipolis,
2004 Route des Lucioles -BP93 06902 Sophia Antipolis Cedex - France

Multi-camera Tracklet association and fusion using ensemble of visual and geometric cues

First Author
Institution1
firstauthor@i1.org

Second Author
Institution2
secondauthor@i2.org

Abstract

Data association and fusion is pivot for object tracking in multi-camera network. We present a novel framework for solving online multi-object tracking in partially overlapping multi-camera network by modelling tracklet association as combinatorial optimization problem hypothesized on ensemble of cues such as appearance, motion and geometry information. Our method learns discriminant weight as a measure of consistency and discriminancy of feature patterns to make ensemble feature selection and combination between local and global tracking information. Our approach contributes uniquely in the way tracklet selection, association and fusion is done. Once multi-view correspondences are established using planar homography, Dynamic Time Warping algorithm is used to make tracklet selection for which similarity has to be calculated i.e overlapping tracklets and subtracklets. Then trajectory similarities are computed for these selective tracklets and subtracklets using ensemble of appearance and motion cues weighted by online learnt discriminative function. Later on, we tackle the association problem by building a k-partite graph and association rules to match all the pair-wise trackets. Finally, from outcome of hungarian algorithm, the associated trajectories are later fused. Fusion is done based on calculated individual tracklet reliability criteria. Experimental results demonstrate our system achieve performance that significantly improve the state of the art on PETS 2009.

1. Introduction

Inspite of number of solutions, object tracking across multiple camera network is still considered most challenging and unsolved computer vision problem, mainly due to placement of cameras, multi-camera calibration, fuzzy data association, fusion and person re-identification across partially or non-overlapping network of cameras. Multi-camera systems help in obtaining more visual information about a same scene, thereby helps in overcoming deficits of single camera object tracking such as inter and intra-occlusions, inferior visibility in crowed scenes,object re-entry, abrupt movement of object, similar appearance and complex interactions among objects in dense environment. Such systems are also useful in rectifying incorrect and fragmented trajectories from individual cameras by associating and fusing collective information thereby substantially improve higher vision tasks such as activity recognition, animation, surveillance etc.

In this paper, we are trying to improve object tracking efficiency through real time multi-camera data association, fusion based on geometry and visual cues. We present a multi-camera tracking approach that associates and performs late fusion of trajectories in a centralized manner from distributed cameras.

Data from each camera in network is gathered to a central node by projecting the trajectories of people to the camera with the most inclusive view through a planar homography technique. Association and fusion is performed based on weighted combination of local and global features such as geometry,appearance and motion(Sec. 5). Association is modeled as a complete K-partite graph, K corresponds to number of cameras in network. Since we use complete K-partite graph, we have the optimal solution in real time. Whereas methods that model association as complex multivariate optimization such as [13], upon scaling, face the problem of being stuck at local minima, may provide sub-optimal solutions and aren't real time realizable. Fusion is performed using reliability based adaptive weighting method. Where the weights are derived from reliability attribute of each tracker from [6]. This enables correct and consistent trajectory after fusion even if the individual trajectories have inherent noises,occlusion and false positives.

Our method has following advantages: 1.Has centralized measures that counter for noisy measurements belonging to same objects overtime by distributed local trackers. 2 We integrated measures during fusion to account both short and long occlusions. 3.Our method doesnt involve costly optimization,metrics and data-gathering(fusion) strategy, thus significantly influencing on the scalability of net-

work and effective real-time applicability of multi-camera tracker. 4.It is modeled around real world context, giving importance to spatio-temporal pairwise relationship amongst tracklets in a bunch of frames(in our case, its 10frames/camera). 5 Our cost function allows us to model multilevel relationship amongst tracklets efficiently.

The reminder of the paper is divided into the following sections:In section 2,we review some significant previous works and how our method differs from them. In section 3,we discuss about multi-camera synchronization and multi-view geometry used in our approach. Next in section 4 we discuss how we formulate trajectory association problem. Followed by section 5 which describes calculation of trajectory similarity metrics. Trajectory fusion is introduced in section 6 , experimental results are presented in section 7 and finally section 8 concludes the paper.

2. Related work

In recent years, there has been variety of approaches proposed for data association in multi-object tracking. Comparatively multi-camera data association and tracking have seen less number of methods published in recent years. Mostly, multi-object temporal data association methods are extended to multi-camera data association setting. On general basis approaches can be outlined based on

1. Fusion time - either early fusion[9] or late fusion [20],
2. The search space - greedy i.e. temporally local or global optimization with longer temporal stride [21], [18].

Approach [13] extends the work of [2] to jointly model multi-camera reconstruction and global temporal data association using MAP. They use single global min cost flow graph for tracking across multiple cameras with a good number of heuristically determined parameters, the complexity increases with more capacities and more risk of not finding a optimal solution if the graph has negative cost cycle of infinite capacity. J. Berclaz *et al.* [2] having detection based on probablity occupancy map, also uses flow graph based method for solving both mono and multi-camera setup within an restricted and predetermined area of interest.

Murray Evans *et al.* [10] uses early fusion strategy for detection inspired from [**?**] and extends it for multi-camera tracking and estimating object size in multi-camera environment. Their approach leverages multiview information into early stage (detection) of pipeline to remove ghosts. Since the synergy map they use for ghost suppression also suppresses existing objects in previous frame thus they cannot perform tracking by associating detections moment to moment, multivariate optimization is performed on object size together with probable location of object in next frame. As optimizing an objective function that combines both object size estimate and tracking information is complex, the solution may be sub-optimal and wouldn't be real time realizable. By nature of their ghost suppression method which involves intricate assumptions over line from camera to object, it makes it difficult to track objects in cluttered or crowded environment.

Anjum, N. and Cavallaro, A. in [1] have presented an unsupervised inter-camera trajectory correspondence algorithm, for the association step, they propose a hybrid approach: project the trajectories from each camera view to the ground-plane in order to find associations among trajectories, and then, make image-plane re-projections of the matched trajectories. These methods rely entirely on goodness of homography, smallest margin of error in calibration gets added up during initial projections and re-projections. Thus are susceptible to introduce errors that end up being association errors. Sheikh, Y.A. and Shah, M. in [19] have proposed a target association algorithm that addressed the problem of associating trajectories across multiple moving airborne cameras with a constraint that at least one object is seen simultaneously between every pair of cameras for at least five frames. Since this method uses object centroid as feature points to recovert the homography and later use RANSAC to find out best subset of such points to find correspondence, it works well when in sparse environment, but in dense environments it may fail. Their approach assumes all the object to be tracked are on the common ground well alligned with all the cameras present in network.

In [5] Chang,T.H. and Gong,S. present a multi-camera system based on Bayesian Networks used to match objects in multiple camera views. For mono-camera tracking they use bayesian networks to enable a full set of possible matching assignments between consecutive frames (based on motion continuity and apparent colour). When the status of segmented blobs change suddenly or the matching becomes ambiguous, the system performs multi-camera cooperative tracking to match subjects across cameras. The system employs a Bayesian modality fusion to match over several subjects such as epipolar geometry, homography, landmark modality, apparent height, and apparent colour. Drawback here is the features are weighted higher based on how recent the evidence is.

Proposed Methodology

Our approach works based on the assumption that cameras in the network are synchronized, calibrated and homography between every pair of cameras are established. If the cameras aren't synchronized, we use linear regression to find correspondence between frames.

3. Trajectory Association

The association problem is related with the need of establishing correspondences between pair wise similar trajectories that come from different cameras.

For simplicity purpose, we experiment using two cameras, the association or correspondence may be modelled as a bi-partite (K-partite, K=2) matching problem in which each set has trajectories that belong to one camera. For each camera C^l and C^r, a set of trajectories S_l and S_r is defined.

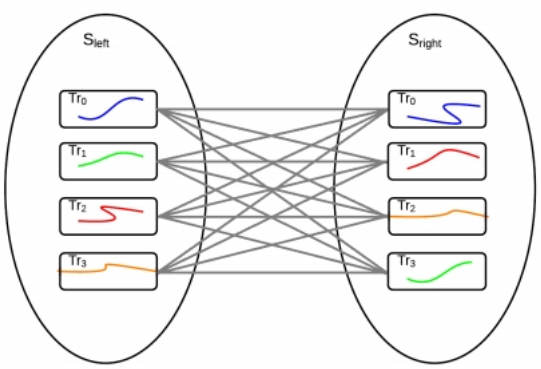

Figure 1. Observable tracklets as Bi-partite graph. Each set contains trajectories (nodes) coming from each camera. The edges represents hypothetic correspondences between trajectories.

A bi-partite graph is a graph in which the vertex set V can be divided into two disjoint subsets S_{left} and S_{right} such that every edge $e \in E$ has one end point in S_{left} and the other end point in S_{right} . Each Object being tracked is denoted as TO_i is the resulting observation (i.e. a track point) of the multi-target tracking algorithm presented by [6]. The physical objects have been synchronized in terms of frame number F and they have 2D space coordinates (x, y) Thus,

$$TO_t = (F, (x, y))_t \ .$$

Let TO_i represents the i^{th} tracked object that belongs to the trajectory $Tr_j^{C^k}$ observed in the camera C^k where $k = l, r$. Thus, each trajectory is composed by a time sequence of 3d points of physical objects:

$$Tr_i^{C^k} = \{TO_0^i, TO_1^i, TO_t^i, ..., TO_{ni}^i\}. \quad (1)$$

Where n_j is the length of above trajectory Consequently, each camera C^k has a set of N and M trajectories belonging to sets S_l and S_r respectively.

$$S_l = \{Tr_0^{C^l}, Tr_1^{C^l}, Tr_2^{C^l}, ...Tr_N^{C^l}\} \quad (2)$$

$$S_r = \{Tr_0^{C^r}, Tr_1^{C^r}, Tr_2^{C^r}, ...Tr_M^{C^r}\} \quad (3)$$

Once the bi-partite graph is built we need to compute overlapping trajectories across cameras and the camera pair-wise trajectory similarities. To perform this task we use spatio-temporal and appearance based trajectory features, assuming that two trajectories viewed from different cameras have to be similar both in time and space.

We abstract the trajectory association problem across multiple cameras as follows: Each trajectory $Tr_j^{C^k}$ is a node of the bi-partite graph that belongs to the set S_k linked with the camera C^k. A hypothesized association between two trajectories is represented by an edge in the bi-partite graph, as is shown in Figure 1. The goal is to find the best match in the graph.

3.1. Time Overlapping Trajectories

For each hypothetical association we first filter and remove the associations of trajectories that do not overlap in time. In the case of time overlapping trajectories we take the intersecting time interval between them, that is, the lower, and the highest time value between both trajectories to get a new time interval in which both trajectories are contained. In the example of the figure 3 we have two trajectories $T_{r_i}^{C_l} \in S_l$ with $0 < j_1 < N_1$ and $T_{r_j}^{C_r} \in S_r$ with $0 < j_2 < N_2$, the resulting overlapping time interval is $\Delta t = [Tr^{C^l}(t_0), Tr^{C^r}(t_f)]$. In order to apply dynamic time warping, we have to have trajectories of the same size to compared frame by frame. The gaps or missing points (due to miss detections or occlusions) are completed with local linear interpolation for the mentioned time interval Δt.

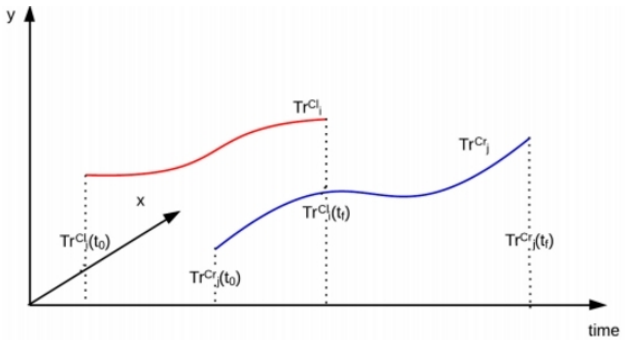

Figure 2. The time interval for each trajectory is shown. In this step we find the most intersecting time interval for both trajectories

3.2. Linear Interpolation and smoothing

Object detection is not perfect due to occlusions, visibility, density of crowd, placement of camera etc, thus, a linear interpolation is applied in order to reach a more complete trajectory. We assume that a person follows uniform linear motion between the previous and next frame. Based on that a linear interpolation is performed in order to correct miss detections of time length equal to predetermined number of frame. To perform this correction, For example,

for one frame miss detections, we estimate the position of the person in the current frame follows:

$$Tr_i^{C^k}(t) = \frac{Tr_i^{C^k}(t-1) - Tr_i^{C^k}(t+1)}{2} \quad (4)$$

Where $Tr_i^{C^k}(t)$ is position of tracked object at time t, $Tr_i^{C^k}(t-1)$ is position of tracked object at time $t-1$, $Tr_i^{C^k}(t+1)$ is position of tracked object at time $t+1$and C^k is the camera number.

The 2D space of the trajectories that belongs to the left camera are projected to 2D space of right camera in order to compare it and find similar trajectories. During this task some noise can arise. Thus, in order to deal with this noise we smooth the trajectory for better results. At this time, we are almost ready to compute the trajectory similarity. However, the common tracklets between both trajectories needs to be found.

3.3. Find Tracklets in Common

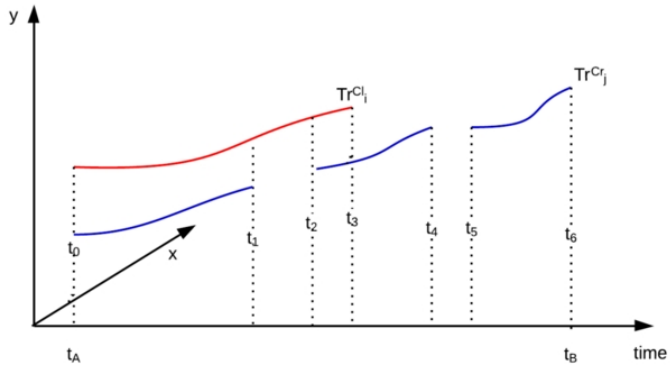

Figure 3. Shows 2 sample tracklets in common between two cameras in the time interval $[t_A, t_B]$.

Figure 3 shows a graphic interpretation of two overlapping time trajectories from the bipartite graph. The two trajectories have common tracklets in the subintervals $[t_0, t_1], [t_2, t_3] \subset [t_A, t_B]$ belonging to trajectories $Tr_i^{C^l}, Tr_j^{C^r}$, two tracklets in the subintervals $[t_3, t_4], [t_5, t_6] \subset [t_A, t_B]$ belonging to $Tr_j^{C^r}$. And finally, one tracklet $[t_1, t_2] \subset [t_A, t_B]$ belonging to $Tr_i^{C^l}$

Later on, a trajectory similarity algorithm is applied for every pair of tracklets in common among both trajectories. It is important to note that now the tracklets contains no empty positions, have the same length, and has been synchronized.

4. Trajectory Similarity Calculation

The comparison of two temporal sequences (e.g. trajectory) and their similarity measurement is a multidimensional sequence data problem that has been studied in many research fields such as data mining, motion tracking, and time series analysis [7]. There are several trajectory similarity measurements in the state of the art. Two similarity models draw our attention: Longest Common Subsequence described by Bergroth et al. [3], and Dynamic Time Warping introduced by Kassidas et al.[14] as they are the most successful ones and widely used. Amongst these we choose the later as it offers enhanced robustness, particularly being sensible to noisy data. As our goal is to associate trajectories we need a global measurement for trajectories comparison. Which is being done using Dynamic Time Warping (henceforth DTW).

DTW has some constraints that need to be addressed:

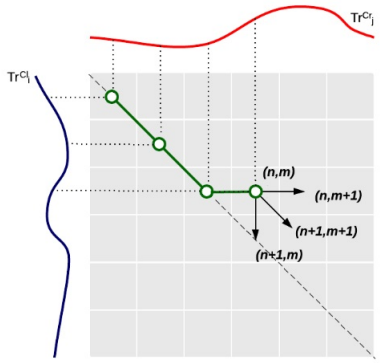

Figure 4. Optimal wrapping path is shown as green.

- *Trajectories of equal length.*
- *Time Synchronized trajectories.*

As a first step in DTW, is to place the trajectories in a grid in order to compares them, and initialize every element as ∞ (represent ∞ distance). Each element of the grid represent the Euclidean distance which is the alignment between two trajectories points $Tr_i^{C^l}(t_i), Tr_j^{C^r}(t_j) \forall t_i \in [0...n_i], \forall t_j \in [0...n_j]$

Many paths connecting the beginning and the ending point of the grid can be constructed. The goal of Dynamic Time Warping is to find the optimal path that minimize the global accumulative distance between both trajectories:

$$D(Tr_i^{C^l}, Tr_j^{C^r}) = min[\sum_{n_i,n_j=1}^{N} d(Tr_i^{C^l}(n_i), Tr_j^{C^r}(n_j))] \quad (5)$$

As shown in Figure 4, applying this method each grid point (n_i, n_j) now represents the minimum accumulated distance from the beginning to the current point.

We can appreciate in Figure 5 that the tracklets are very similar from frame 65 to 82, but after seem like they start to be unequal. The further close the optimal path wanders

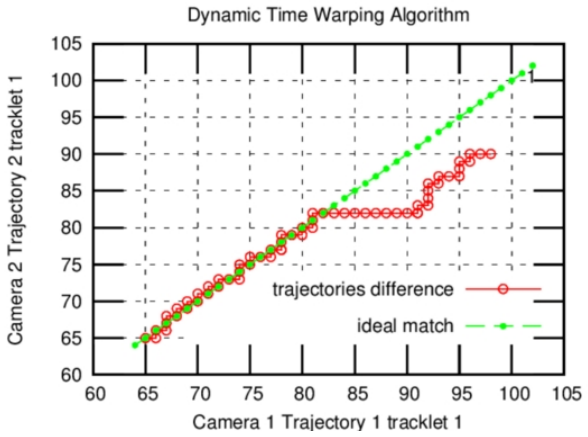

Figure 5. DTW results for tracklet 1 of two trajectories comparison. In X and Y the frames are shown. The optimal path is represented in green, and the DTW result is shown in red.

around the diagonal, the more the two sequences match together. It is important to mention that for long trajectories this algorithm is very expensive in terms of memory. For this reason we adapted the algorithm to reuse a fixed matrix that is emptied and re-filled every time we reach the maximum matrix size.

4.1. Ensemble feature combination using online learnt discriminative weights

It is a known fact that feature combinations capture more underlying semantics than single feature patterns. But using less influential patterns may not improve the performance of a tracker mainly due to limited discriminability of the feature. Trajectory similarity is calculated as a two stage approach (Local and Global). An ensemble of local and global features is used instead of determining similarity score. The algorithm learns weights online based on the consistency and maximum discriminability of the feature patterns.

Local tracklet similarity. At local stage, importance is given to local frame to frame geometric information. From DTW results, we derive Euclidean Distance metric for each set of trajectories.

$$EDM = D(Tr_i^{C^l}, Tr_j^{C^r}) \tag{6}$$

Global tracklet similarity. At global stage, information pertaining to overall appearance of the object is taken into account for determining the similarity between tracklets. Feature patterns used for determining a overall appearance score is discussed in section 5.2. A global matching score quantified from such features represent global tracklet similarity.

Each element of trajectory association cost matrix represents weighted sum of Euclidean distance and Global Matching Score(GMS) between two trajectories $Tr_i^{C^l}, Tr_j^{C^r}$. An entry in association cost matrix can be defined as follows:

$$D'(Tr_i^{C^l}, Tr_j^{C^r}) = (1 - \omega_m(Tr_i)).EDM(Tr_i^{C^l}, Tr_j^{C^r}) + \omega_m(Tr_i).GMS(Tr_i^{C^l}, Tr_j^{C^r}) \tag{7}$$

where GMS is defined in section 4.2 and EDM is given in equation 6.

Where ω_m is appearance descriptor weight learned to specify if more importance should be given to appearance cues or geometric cue. i.e online discriminant weighting to decide a tradeoff between local information extracted from frames or global appearance information from tracklets. The learnt weight helps in better feature selection and combination to enhance inter-tracklet discrimination and also cope-up with intra-tracklet variations. In this approach both local geometric and global appearance feature patterns compliment each other and is impactfull in situations where the dataset involves significant appearance changes across object pose, illumination, viewing angle and different camera parameters.

As shown in equation 11, ω_m is calculated as minimum between consistency and discriminative power of appearance based global tracklet features. To calculate consistency and discriminative power of tracklet features across cameras, we need to color calibrate the cameras for accounting color distortion between them. Therefore as a preprocessing measure before validating discriminability, we perform histogram specification and histogram matching i.e considering camera 1 as reference camera, we project and transform the histogram of camera 2 onto histogram of camera 1. Level of color distortion after matching is validated by comparing the tranformed histogram and reference histogram using correlation.

Discriminative power of the GMS features of trackets are calculated as a adopted fisher score. Fisher Score is a quantitative measure popularly used in statistics usually for numerically solving maximum likelihood problems. Fisher score gained popularity in computer vision for its ability to compare one feature subset with another in order to find most discriminating set of features [12]. [8] used fisher score to online select most discriminative set of tracking features. Since ours is a multi-camera setup, we need to adopt this fisher score to avoid certain undesirable scenarios from affecting the final discriminant score. Constraints we lay on fisher score are

- In a multicamera tracking problem, the discriminating power of tracklet features should be measured across cameras and not intra-camera. Thus in equation 8, instead of calculating the mean over all trackets over both cameras, we calculate mean only on the camera with candidate matching tracklets.

- Using weighted feature mean and variance based on online discriminative descriptor weights obtained while calculating GMS. This enables most discriminating features to influence fisher score.

$$FS(Tr_i, Tr_j) = \frac{\sum_{f=1}^{F} w_f(\mu_{fi} - \mu_f^{c^j})^2}{\sum_{f=1}^{F} w_f(\rho_{fi}^2)} \tag{8}$$

where f corresponds to any feature considered in GMS, μ_{fi} and ρ_{fi} are the mean and the variance of the f-th feature of i-th tracklet respectively, F is the number of features considered, w_f is descriptor similarity weight of f-th feature used in GMS and $\mu_f^{c^j}$ is the mean of f-th feature of over all tracklets belonging to complimentary pair of camera C^j.

Similarly, Overall consistency score on entire tracklet is calculated as square root of sum of weighted consistency score of individual features over a tracklet.

$$SD'(Tr_i) = \sqrt{\frac{n_1.\rho_{f1} + n_2.\rho_{f2} + ...n_F.\rho_{fF}}{n_1 + n_2 + ... + n_F}} \tag{9}$$

Where n_F is total number of F^{th} feature instances, ρ_{fF} is individual consistency score of F^{th} feature.

An individual consistency score is obtained for each feature in GMS metric over the entire tracklet (Tr_i). For a feature f, consistency score can be calculated as follows.

$$\rho_{fi} = \sqrt{\frac{\sum_{t=1}^{n_i} (f(TO_t^i) - \overline{f(TO^i)})^2}{n_i}} \tag{10}$$

Where $\underline{f(TO_t^i)}$ is i^{th} feature extracted from tracked object TO_t, $\overline{f(TO_i)}$ is i^{th} feature mean and n_i is total number of i^th feature instances.

Appearance descriptor discriminant weight ω_m is

$$\omega_m(Tr_i) = min(SD'(Tr_i), FS(Tr_i, Tr_j)) \tag{11}$$

4.2. Global Matching Score (GMS)

Appearance based cues have played a vital role in tracklet association rule mining. Given a set of appearance cues, how do we select a high quality ones for effective discrimination from other candidates for tracklet association? This is answered to an extent in mono-camera tracklet reliability descriptor work by [17].
We select set of cues inspired from their work and extend it to suit our approach. We use k=6 cues for our work. Namely,

2D shape ratio (k=1) and 2D area (k=2)

Shape ratio and area of an object are obtained from respective bounding boxes, within a temporal window, they are immune to lighting and contrast changes. Thus they are one of good cues to use.

Color histogram (k=3) and Dominant Color (k=4)

It is basically a normalized RGB color histogram of pixels inside bounding box of moving object. Dominant color descriptor is used to take into consideration only important color of object.

Motion descriptor (k=5)

Depending on the context, constant velocity model or Brownian model is used to describe motion represented by Gaussian distribution. It is useful when objects have similar appearance.

Color covariance descriptor (k=6)

Color covariance descriptor is a covariance matrix that characterizes the appearance of regions in image and is invariant to size and identical shifting of color values. There by resisting to illumination changes

To ensure reliable tracklet association,[17] intelligently weights the discriminative appearance and motion model descriptors and generates a global matching score (GMS) . The global matching score of tracklet Tr^i with each tracklet in its matching candidate list (represented by Tr^j) is

$$GMS = \frac{\sum_{k=1}^{6} w_k^{ij}.DS_k(Tr_i^{C^l}, Tr_j^{c^r})}{\sum_{k=1}^{6} w_k^{ij}} \tag{12}$$

Where w_k^{ij} are corresponding weights of each feature descriptors $DS_k(Tr_i^{C^l}, Tr_j^{c^r})$ calculated online by modeling them directly propotional to descriptor similarity of a tracklet with its matching candidate and inversely propotional to descriptor similarity of other overlapping tracklets.

If Tr_i, Tr_j are matching candidates, Tr_i, Tr_p are other overlapping tracklets, Their discriminative descriptor weight is calculated as

$$w_k^{i,j} = \alpha^{[DS_k(Tr_i, Tr_j) - \tilde{X}(DS_k(Tr_i, Tr_p)) - 1]} \tag{13}$$

Where $\alpha = 10$ determined experimentally. The discriminative weight for motion cue alone is calculated as

$$w_6^{i,j} = 0.5 - 0.5max(w_k^{i,j})\ k = 1...5 \tag{14}$$

Now the bi-partite graph is complete and the weight W_{ij} of each edge $e \in E$ in $G = (V; E)$ is $D(Tr_i^{C^l}, Tr_i^{C^r})$ given by equation 7.

4.3. Hungarian Algorithm

The task at hand is finding the maximum matching of G . Formally, *maximum matching is defined as a matching with the largest possible number of edges; it is globally optimal*. In other words, the goal is to find an optimal assignment: the one that minimizes total cost of a matrix. To find the maximum matching in G. we apply the Hungarian Algorithm defined by Kuhn [15] given the cost matrix built with the W_{ij} values. The Hungarian method is a combinatorial optimization algorithm that solves the assignment problem in polynomial time $\mathcal{O}(n^3)$, where n is number of nodes or vertexes V of the bi-partite graph G. After apply the Hungarian Algorithm to the matrix we get the maximum matching as shown in figure 6.

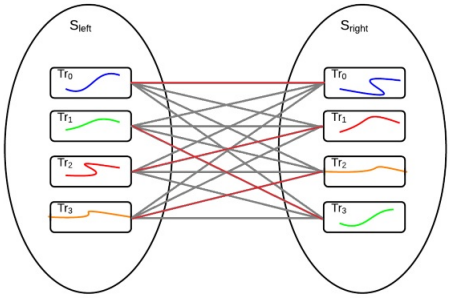

Figure 6. Associations of each trajectories after hungarian algorithm.

5. Trajectory Fusion

Once association is done, the next step is to fuse a pair of corresponding trajectories. In order to perform this task, a merged trajectory with the information coming from both views is built. To fuse two trajectories coming from two different cameras at a time t, e.g. $Tr_i \in S_{left}$ with $0 < i < N$ and $Tr_j \in S_{right}$ with $0 < j < M$ into a global one $Tr_{Gi,Gj}$ we apply an adaptive weighting method as follows:

$$Tr_{Gi,Gj}(t) = \begin{cases} w_1 Tr_i^{Cl}(t) + w_2 Tr_j^{Cr}(t) \; if \; Tr_i^{Cl}(t), Tr_j^{Cr}(t) \\ \qquad \text{overlap over time t} \\ Tr_i^{Cl}(t) \; if \; \text{only} \, Tr_i^{Cl}(t) \text{exists at time t} \\ Tr_j^{Cr}(t) \; if \; \text{only} \, Tr_j^{Cr}(t) \text{exists at time t} \end{cases} \quad (15)$$

As we defined in Eq.(1), each trajectory is composed by a set of tracked Objects. [6] defined a method to quantify the reliability of the trajectory of each interest point by considering the coherence of the Frame-to-Frame (F2F) distance, the direction, and the HOG similarity of the points belonging to a same trajectory. Thus, as each physical object has a reliability attribute R with values $[0, 1]$ the weighed function is defined in terms of its R value as follows:

$$w_1 = \frac{R_{PO_n}}{R_{PO_n} + R_{PO_m}} \quad w_2 = \frac{R_{PO_m}}{R_{PO_n} + R_{PO_m}} \quad (16)$$

The key idea behind this weighted approach is that resulting trajectory(merged) is in between the other trajectories and will be close to the trajectory having higher reliability values.

The fused trajectory is not smooth. In order to get a better and smoothed one, we apply a simple moving average technique (also called moving mean). The first element of the moving average is obtained by taking the average of the last N elements of the trajectory. Then the subset is modified by shifting it forward; excluding the first number of the trajectory and including the next trajectory point. This process is repeated over the entire trajectory. As result of this process we obtain a smoothed trajectory without noise.

6. Evaluation

Our approach is evaluated on the publicly available PETS2009 dataset [11]. PETS2009 is a challenging dataset due to its low FPS and inter-object occlusions . We choose View1,View3 in S2.L1 scenario to evaluate. There is one static occlusions in View1 namely a pole with display board , View3 has 2 static occlusions namely a pole with display board and a tree occupying significant area in right side of video. There is substantial color tone variation between the views, making it hard for color based cues. For this reason most of the methods avoid this combination of view. For multi-camera evaluation, we use ground truth based on a common plane ie world referential of View1.

Unfortunately, the lack of common metrics for measuring the performance of multi-camera multi-object detection makes result comparison even more difficult.
But for evaluating our work, we use the following metrics:
1.CLEAR [4] metrics namely Multiple Object Tracking Accuracy(MOTA) and Multiple Object Tracking Precision(MOTP).
2. Identity Switches(IDS), Track Fragments(FM), Mostly Tracked(MT), Partly Tracked(PT) and Mostly Lost(ML) from [16]

6.1. Experimental settings

//say what are the parameters to tweak ,tune and how they are obtained, their impact/how they affect on our results

Table 1. Comparison of our method with previous multicamera state of art on PETS2009 dataset

Scenario	Method	Camera ID	MOTA(%)	MOTP(%)	MT(%)	PT(%)	ML(%)	FM	IDS
PETS 2009 S2.L1	Berclaz et al.9	1,3,5,6,8	82	56	-	-	-	-	-
	Leal-Taixe et al.25b	1,5	76	60	-	-	-	-	-
	Leal-Taixe et al. 25b	1,5,6	71.4	53.4	-	-	-	-	-
	Murray Evans et al. 10	2 Cameras	63	55	-	-	-	-	-
	Martin Hofmann et al. 7	1,5	99.4	82.9	100	0	0	1	1
	Martin Hofmann et al. 7	1,5,7	99.4	83.0	100	0	0	1	2
	Our Approach(RGB)	1,3	76.33	65.28	92.59	0.035	0.714	-	2

Table 1 summarizes past few results on multi-camera PETS2009 dataset. Unlike other methods which use heavy computation for best results as a trade off over real-time performance, our objective was to make the algorithm more real time and online with buffer frames at the same time making minimal sacrifice on the accuracy. This is achieved as our method uses network flow simplex method which is both effective and simple. We use buffer frame size = 20 frames in a temporal sliding window pattern to be able to perform association and fusion online

7. Conclusion

We introduced a multi object tracking association and fusion across multi-camera network. In order to accomplish this, we built a multi-camera framework making use of mono-camera tracking. The trajectory similarity is computed by using Dynamic Time Warping approach and online learning of discriminative weights belonging to appearance and geometric cues respectively. Afterwards, the association relies on maximum bipartite graph matching performed by Hungarian algorithm Finally, the fusion between associated trajectories is performed by an adaptive weighted method based on reliability score of individual tracklets. We evaluated the multi-camera approach and compare its results against the state of art on public dataset. Our approach outperforms considerably some existing multi-camera tracking and comparable to the state of art on benchmark dataset, as has been shown in Final results and has a good occlusion management.

References

[1] N. Anjum and A. Cavallaro. Trajectory association and fusion across partially overlapping cameras. In *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*, pages 201–206. IEEE, 2009.

[2] J. Berclaz, F. Fleuret, and P. Fua. Multiple object tracking using flow linear programming. In *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on*, pages 1–8. IEEE, 2009.

[3] L. Bergroth, H. Hakonen, and T. Raita. A survey of longest common subsequence algorithms. In *String Processing and Information Retrieval, 2000. SPIRE 2000. Proceedings. Seventh International Symposium on*, pages 39–48. IEEE, 2000.

[4] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *Journal on Image and Video Processing*, 2008:1, 2008.

[5] T.-H. Chang and S. Gong. Tracking multiple people with a multi-camera system. In *Multi-Object Tracking, 2001. Proceedings. 2001 IEEE Workshop on*, pages 19–26. IEEE, 2001.

[6] D. P. Chau, F. Bremond, and M. Thonnat. A multi-feature tracking algorithm enabling adaptation to context variations. In *Imaging for Crime Detection and Prevention 2011 (ICDP 2011), 4th International Conference on*, pages 1–6. IET, 2011.

[7] P. Chen, J. Gu, D. Zhu, and F. Shao. A dynamic time warping based algorithm for trajectory matching in lbs. *International Journal of Database Theory and Application*, 6(3):39–48, 2013.

[8] R. T. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1631–1643, 2005.

[9] M. Evans, L. Li, and J. Ferryman. Suppression of detection ghosts in homography based pedestrian detection. In *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, pages 31–36. IEEE, 2012.

[10] M. Evans, C. J. Osborne, and J. Ferryman. Multicamera object detection and tracking with object size estimation. In *Advanced Video and Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on*, pages 177–182. IEEE, 2013.

[11] J. Ferryman and A. Ellis. Pets2010: Dataset and challenge. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pages 143–150. IEEE, 2010.

[12] Q. Gu, Z. Li, and J. Han. Generalized fisher score for feature selection. *arXiv preprint arXiv:1202.3725*, 2012.

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

[14] A. Kassidas, J. F. MacGregor, and P. A. Taylor. Synchronization of batch trajectories using dynamic time warping. *AIChE Journal*, 44(4):864–875, 1998.

[15] H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[16] C.-H. Kuo and R. Nevatia. How does person identity recognition help multi-person tracking? In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1217–1224. IEEE, 2011.

[17] T. L. A. Nguyen, D. P. Chau, and F. Bremond. Robust global tracker based on an online estimation of tracklet descriptor reliability. 2015.

[18] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1201–1208. IEEE, 2011.

[19] Y. A. Sheikh and M. Shah. Trajectory association across multiple airborne cameras. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):361–367, 2008.

[20] M. Taj and A. Cavallaro. Distributed and decentralized multicamera tracking. *Signal Processing Magazine, IEEE*, 28(3):46–58, 2011.

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