Efficient Multi-Target Tracking via Discovering Dense Subgraphs

Behzad Bozorgtabar[a], Roland Goecke[a,b]

[a] *Vision & Sensing, HCT Research Centre, University of Canberra*
[b] *IHCC, CECS, Australian National University*

Abstract

In this paper, we cast multi-target tracking as a dense subgraph discovering problem on the undirected relation graph of all given target hypotheses. We aim to extract multiple clusters (dense subgraphs), in which each cluster contains a set of hypotheses of one particular target. In the presence of occlusion or similar moving targets or when there is no reliable evidence for the target's presence, each target trajectory is expected to be fragmented into multiple *tracklets*. The proposed tracking framework can efficiently link such fragmented target trajectories to build a longer trajectory specifying the true states of the target. In particular, a discriminative scheme is devised via learning the targets' appearance models. Moreover, the smoothness characteristic of the target trajectory is utilised by suggesting a smoothness tracklet affinity model to increase the power of the proposed tracker to produce persistent target trajectories revealing different targets' moving paths. The performance of the proposed approach has been extensively evaluated on challenging public datasets and also in the context of team sports (e.g. *soccer*, *AFL*), where team players tend to exhibit quick and unpredictable movements. Systematic experimental results conducted on a large set of sequences show that the proposed approach performs better than the state-of-the-art trackers, in particular, when dealing with occlusion and fragmented target trajectory.

Keywords: Multi-target tracking, dense subgraphs, relation affinity graph, discriminative target appearance model

1. Introduction

Multi-target tracking in real world scenarios is a crucial problem for many computer vision tasks. Some of its potential applications include anomaly detection, visual surveillance, human-computer interaction and sports analysis.

Generally speaking, a multi-target tracking framework takes a set of target detections in each frame from any pre-trained detector as its input and aims to recover the trajectories of all targets as well as maintaining their consistent identities. However, the problem and its difficulty depend on conflicting challenges, such as occlusion, drastically varying illumination and viewpoint for the target

during the tracking, and therefore its trajectory is expected to be fragmented into multiple tracklets across video sequence. In addition, recovering the true target trajectories is an ambiguous problem when several visually similar targets move closely together. Apart from these challenges, the detector cannot be assumed to be accurate and may produce false negatives (missed targets) or extra detections. In such scenarios, revealing each individual's identity results in an extreme complexity in the number of tracks and measurements, and thus quickly becomes infeasible.

Recently, *tracking-by-detection* methods [1, 2] have attracted much attention due to the promising improvements on object detection. They usually seek proper association between the target hypotheses by constructing their mutual similarities (affinities) based on multiple cues, such as *motion similarity*.

In the proposed framework, multi-target tracking is formulated as a dense subgraph extraction problem, in which each recovered dense subgraph specifies the trajectory of the one particular target during the tracking. Here, the higher order correspondences between the observations (targets' hypotheses across video sequence) are considered. With respect to the literature, this approach has the following advantages:

1. An iterative strategy is devised to partition the spatio-temporal graph of target hypotheses into different subgraphs. Each reveals a different target trajectory. At each iteration, the most coherent vertices of the constructed spatio-temporal graph in a short time period of the video will be added to the existing subgraphs (tracklets) until reaching the densest subgraphs, which are more likely to be the true trajectories of the targets. Since the proposed method works on small graph partitions (in the few video frames), it is much more efficient in terms of computational expense.
2. In the presence of occlusion of the target, whether caused by another target or visual obstacles where the target trajectories are only partially observable due to a failure of the pre-trained detector, a discriminative scheme is devised via learning the targets' appearance models. In addition, target hypotheses are sparse in time (*false negative*), which prevents obtaining good target motion estimates. Therefore, the smoothness characteristic of the target trajectory is utilised by suggesting a smoothness tracklet affinity model to increase the power of the proposed tracker to associate the fragmented target tracklets to produce persistent target trajectories revealing different targets' moving paths (see Fig. 1).

2. Related Work

Most of the recent approaches to tracking which pursue a *tracking-by-detection* strategy, comprises two steps: (i) obtaining a set of independent target candidates in each frame (*detection*) and (ii) assigning the target identities to these candidates (*data association*). Although there are some related work [3], which address both problems (object detection and data association) via proposing

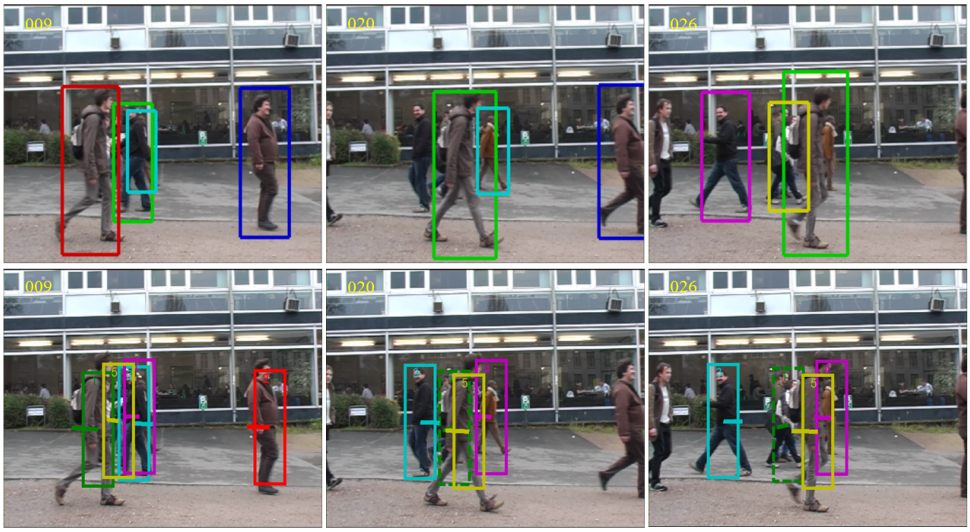


Figure 1: Some tracking results from a '*TUD-Campus*' video sequence without (*first row*) and with (*second row*) considering higher order relationships between different detections across the temporal domain. The proposed tracker (*second row*) is able to handle mutual occlusion of a walking person caused by another similar person, while the target is assigned a new ID (highlighted with different colour) in the first row obtained by locally constrained-temporal-domain methods, which associate targets in a limited time period of the sequence.

these two steps in a single framework, but the majority of state-of-the-art methods considered these two problems separately and the priority is usually given to the data association design. Therefore, the related work in this tracking step is extensively described here:

2.1. Local Data Association

These methods consider correspondence between the target observations (detections' responses) in the temporally local frames which yields the polynomial complexity. The best example of such approaches are Bi-partite matching methods [4, 5]. Brendel et al. [5] presented an approach to formulate data association problem as finding the *maximum-weight independent set* (MWIS) of the graph of detections' responses. Shu *et al.* [4] suggested part-based model to obtain human part detections, which is robust against partial occlusion and appearance changes. Some methods devise a greedy strategy for the data association by solving a series of Bi-partite assignment problems to assign an evolving set of target trajectories to the target hypothesises in each video frame. Shafique and Shah [6] proposed a *k-partite* complete graph where the connections in the graph are not limited to only two consecutive frames. Bae and Yoon [7] proposed an online multi-object tracking system with a new data association technique considering the track existence probability. Although this class of methods is computationally efficient, they fail in tracking visually similar moving targets or when there exists heavy occlusion or pose variations, due to their limited-temporal-locality strategies.

2.2. Global Data Association

Unlike local association based methods, recently, another category of data association based techniques (namely *global association*) have been proposed which are by far, the most explored strategy for multi-target tracking. The global association approaches operate on the *batch* of frames and sometimes the whole temporal sequence at once [8, 9]. These approaches formulate the higher order relationships between different target observations (detection responses).

Multi-object tracking has been recently formulated as a *network flow* problem where a set of object tracks are resolved simultaneously by solving min-cost flow techniques. Different approaches to minimum cost flow have been presented recently. Zhang *et al.* [10] proposed a global optimal solution for the network flow optimisation using *push-relabel* algorithm. Pirsiavash *et al.* [11] utilise the same graph as in [10], and use a fast greedy shortest path algorithm for the tracking problem. Berclaz *et al.* [12] suggested a globally tracking by relying on the *k-shortest paths* (KSP) algorithm to solve the flow problem. To handle the difficulties in object localisation caused by occlusion and clutter in the video, recently Chari *et al.* [13] proposed to add pairwise costs to the min-cost network flow framework where a convex relaxation strategy with an efficient *rounding heuristic* is designed to solve the problem. The main drawback of these approaches is that, they do not consider acceleration information for computing trajectories which is required for trajectories' smoothness in spatio-temporal domain.

Work by Milan *et al.* [14, 15] proposed multi-target tracking in crowded scenarios by incorporating mutual exclusion between the targets at both: *trajectory estimation* where any two trajectories should remain spatially separated as well as the *data association* step, in which each trajectory should be assigned at most one detections per frame. However for this purpose, they employed an already-performed tracking from [11] as an input to their approach. Hence these work should rather be considered as trajectory-refinement procedure.

Recently, *graph partitioning* based methods have been proposed for multi-target tracking problem. Here, given a set of target hypotheses produced by a generic object detector, the aim is to build a graph of all target hypotheses and to devise a strategy to associate them across the time. Kumar et al. [16] formulated multiple object tracking as a graph partitioning problem, in which the sum of weighted edges connecting vertices with the same label must be maximised. Then, a *Conditional Random Field (CRF)* is defined and optimised using an efficient combination of *message passing* and *move making* algorithms.

The work of Zamir *et al.* [9] and Wen *et al.* [17] are most relevant to the proposed approach. Zamir *et al.* [9] proposed a sequential Generalized Minimum Clique Problem (GMCP) where the computed cliques on a graph of detection responses specify the target trajectories. For handling the occlusion, hypothetical nodes were devised for the missing targets assuming constant velocity in a short period of time. In addition, similar to the proposed framework, the input video sequence is divided into a number of segments and the final trajectory of each person obtained by stitching together the obtained target tracklets in each segment.

However, the proposed approach differs in two aspects. First, unlike their work that focus on one target at a time, rather than dealing with all targets simultaneously, here, all targets are considered jointly in the presented framework. Second, the proposed framework is more workable in cases where there are fragmented trajectories and the targets only are observable in a short temporal span. We extend the dense subgraph extraction algorithm [18] to formulate our proposed tracking framework in an efficient way in which the number of initialisations of the algorithm has been significantly reduced due to the number of constructed initial target tracklets while not sacrificing for the tracker's accuracy. In addition, we propose a new graph density by defining several affinity models based on different cues between the tracklets which promises a high probability of obtaining all significant dense subgraphs (potential tracklets). Similar to the proposed contribution, Wen *et al.* [17] formulated the multi-target tracking as a hierarchical dense neighbourhood searching problem on the multiple relation affinity hypergraphs. However, they consider each target tracklet (graph node) as an initialisation for the optimisation, which increases the results complexity. In contrast, the proposed framework starts with the initial tracklets (*less reliable* tracklets) and iteratively grows and condenses the tracklets to obtain more reliable tracklets, which are more likely to be true trajectories of the targets.

2.3. Contribution

Briefly, the contributions of this paper are summarised as below:

1. In this paper, multi-target tracking is formulated as an efficient subgraph extraction problem, which considers all the relations, e.g. motion similarity between any pair of target tracklets apart from their closeness in the temporal domain. Different from *constrained-temporal-locality* methods, which are not well suited to consider the target observations outside of the temporal neighbourhood they are focused on, this approach processes much longer temporal segments to link and cluster target hypotheses jointly across space and time. An iterative two-step tracking scheme is proposed as shown in Fig. 2. It operates on the initial target tracklets (subgraphs). First in the *trajectory growth* step, the most correlated vertices to the current subgraph are chosen and added to grow the target tracklets. Further, in each subgraph, the uncorrelated vertices, which are more likely to be false positive or belong to the occluded targets, are penalised, allowing for subgraph *condensation*. Therefore, multiple dense subgraphs[1] robustly revealed. The proposed framework yields a better formulation of the *data association* problem, which is shown to lead to superior results.
2. In addition, to handle false positive detections and the difficulties caused by the presence of interacting targets which confuse association between the target hypotheses, a target-specific metric learning model is proposed to obtain effective appearance cues for reliable association between different tracklets. For this purpose, a *discriminative* projection space is learned effectively where the appearance features of tracklets are projected onto a low-dimensional subspace to lead to more distinguishable target representations, while keeping the computational complexity low. Moreover, to address the specific problems caused by unreliable detection responses or occlusion, the target trajectories' smoothness is exploited via proposing the tracklet smoothness affinity model to take advantage of point trajectories. Meanwhile, in order to penalise an abrupt gap between the tracklets belonging to the same target, motion similarities between the tracklets are utilised to enforce merging them through (e.g. occlusion).

3. Tracking via Dense Subgraph Extraction (Proposed Framework)

Given a set of target object hypotheses (target states) O for the image sequences , where the state of target object j at frame t is given as $O_t^j = \left\{p_t^j, s_t^j, v_t^j\right\}$ if an object j appears at frame t ($b_t^j = 1$) and p_t^j, s_t^j and v_t^j are the position, scale and velocity, respectively[2]. The aim is to find the most likely set $\mathcal{T} = \{\tau_1, \tau_2, \cdots, \tau_N\}$ of the target object tracklets. The tracklet τ_i of the i^{th} target object is defined as a set of states up to frame t, and is represented as

[1] Points to those subgraphs where the affinities between their vertices are large.

[2] The presence of the j^{th} target object is denoted via the binary function b_t^j

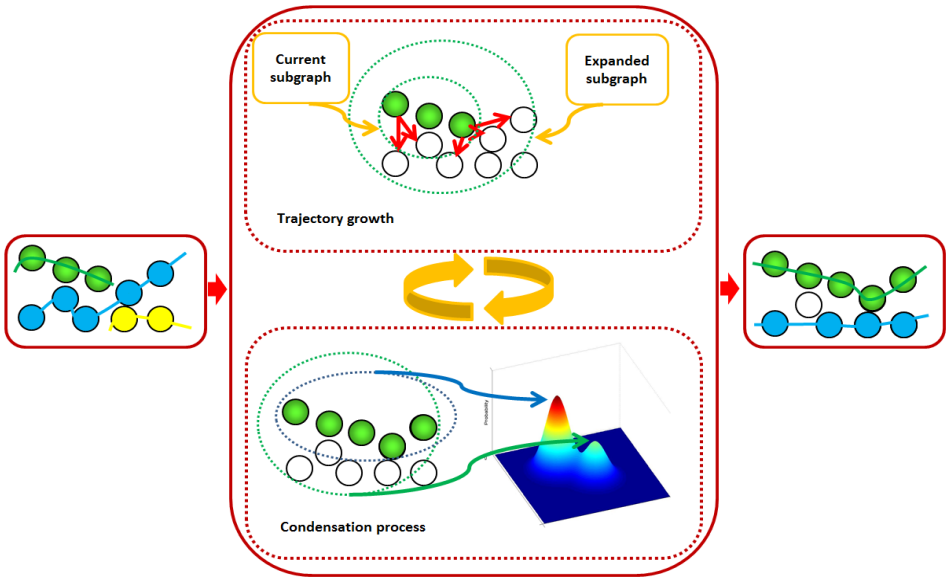


Figure 2: An illustration of the proposed iterative algorithm. It starts from the initial tracklets obtained by the initialisation and adds the most coherent vertices of the graph based on the *average affinity* between each vertex and the current subgraph. In continue, all vertices in the current video segment are considered and the uncorrelated ones are excluded. The circles represent different spatio-temporal graph vertices and their colour indicates different target IDs.

$\tau_i = \left\{ O^i_{t^i_s:t^i_e} \mid b^i_{t^i_s:t^i_e} = 1, 1 \leq t^i_s \leq t^i_e \leq t \right\}$, which actually is a temporal sequence of i^{th} target detection responses starting at time stamp t^i_s and ending at the end frame t^i_e. Building the spatio-temporal hypergraph $G = (V, E)$ for the current video segment, whose vertices $V = \{v_1, \cdots, v_m\}$ correspond to target tracklets and the edges E include more than just two vertices as $E \subset \overbrace{V \times \cdots \times V}^{l}$[3], the goal is to partition the spatio-temporal graph into dense subgraphs such that each subgraph corresponds to the trajectory of one person. The term '*dense*' refers to the most correlated set of vertices of the spatio-temporal graph, which is more likely to be a consistent target trajectory. The notations used in this paper are presented in Table 1.

In the proposed approach, the tracking task is formulated as an *iterative* algorithm starting from the initial tracklets obtained by the initialisation and then adding the most similar graph vertices based on the *average affinity* between each vertex and the current tracklet. In continue, considering relationships between all target tracklets within the current video segment, the less coherent tracklets are excluded, which are more likely to be fragmented or belong to the different targets. These two steps are iterated until convergence is reached, when the final target tracklets are revealed from the constructed relation affinity graph among the tracklets. For this purpose, the input video is divided into a few temporal segments and a set of target tracklets is produced within each segment using the proposed iterative method. Then, the tracklets obtained in all segments are stitched together to obtain the final trajectory of each target by carrying out the same framework (dense subgraph extraction) again for the entire video time span (see Fig. 3).

To evaluate the similarity of any two tracklets, *local* cues such as dense motion trajectories and *global* cues such as appearance features, as well as *intermediate* ones such as motion smoothness are considered. The proposed method matches the tracklets of one target across the full video duration, while incorporating the remaining targets using the $\overbrace{m \times \cdots \times m}^{l}$ symmetric affinity array A, exhibits the mutual similarities between the tracklets. Mathematically, $A(\tau_i, \tau_j)$ is used to denote the probability of τ_i and τ_j belonging to the same target, which is computed as follows:

$$A(\tau_i, \tau_j) = \Lambda_S(\tau_i, \tau_j) \Lambda_M(\tau_i, \tau_j) \Lambda_A(\tau_i, \tau_j) \tag{1}$$

where $\Lambda_S(\tau_i, \tau_j)$, $\Lambda_M(\tau_i, \tau_j)$ and $\Lambda_A(\tau_i, \tau_j)$ are the tracklet *smoothness* affinity, tracklet *motion* affinity and the tracklet *appearance affinity*, respectively (see Section 4). Considering the spatio-temporal constraint, if the two tracklets have overlap, they should not correspond to the same target. Thus, their mutual affinity[4] is set to zero as $A(\tau_i, \tau_j) = 0$. The multiplicative formulation for the

[3] l is the number of graph vertices involved in each hyperedge.
[4] The affinity value lies in $[0, 1]$.

Table 1: The table of notations used in this paper.

Notation symbol	Explanation
N	Number of possible target tracklets
O	Set of target object hypotheses (detection responses)
b_t^j	Binary function indicates the j^{th} target object presence at frame t
O_t^j	State of target object j at frame t
p_t^j	Position of target object j at frame t
s_t^j	Scale of target object j at frame t
v_t^j	Velocity of target object j at frame t
$\mathcal{T}$	Set of all target tracklets
t_s^i	Starting time index for the i^{th} target tracklet
t_e^i	Ending time index for the i^{th} target tracklet
i	A node of spatio-temporal graph, *i.e.* a tracklet index
τ_i	Tracklet of the i^{th} target object
$E \subset \overbrace{V \times \cdots \times V}^{l}$	Graph hyperedges include l dense vertices
m	Number of graph vertices in the spatio-temporal graph
l	Number of vertices involving in each graph hyperedge
A	Symmetric graph affinity array
Λ_S	Tracklet *smoothness* affinity
Λ_M	Tracklet *motion* affinity
Λ_A	Tracklet *appearance* affinity
V_S	A vertex set with any subset S of all vertices I
G_S	A subgraph with the vertex set as V_S
y	Probabilistic cluster of vertices (indicator vector of the graph vertices)
Δ	The standard simplex
F	A simplex projection
$\mathcal{G}$	A set of all subgraphs
n	Number of vertices in the subgraph
$\phi(y)$	Compactness of the probabilistic cluster
y_S	The probabilistic indicator vector of the subgraph G_S
A_S	The affinity matrix of the subgraph G_S
(t_1, t_2)	The indices of frames closest in time for the two tracklets
$Tr(\tau_i, \tau_j)$	All common point trajectories overlapping tracklets (τ_i and τ_j)
tr_o	A sequence of space-time point trajectory (the o^{th} point trajectory)
D_i^t	Detection response at frame t for the tracklet τ_i
loc	Euclidean (relative) distance function
P_j^{head}	The position of the *head* of tracklet j
P_i^{tail}	The position of the *tail* of tracklet i
Δt	The frame gap between two tracklets
v_j^B	The backward velocity of the tracklet j
v_i^F	The forward velocity of the tracklet i
r	Number of initial target tracklets
f_i, $\hat{f}_i$	The original and projected feature vector belonging to the i^{th} target tracklet, respectively
c_i	The i^{th} tracklet label (identity)
M	The parameter of the Mahalanobis distance function
L	The linear transformation for the tracklet features
T	Number of collected training features from the tracklets

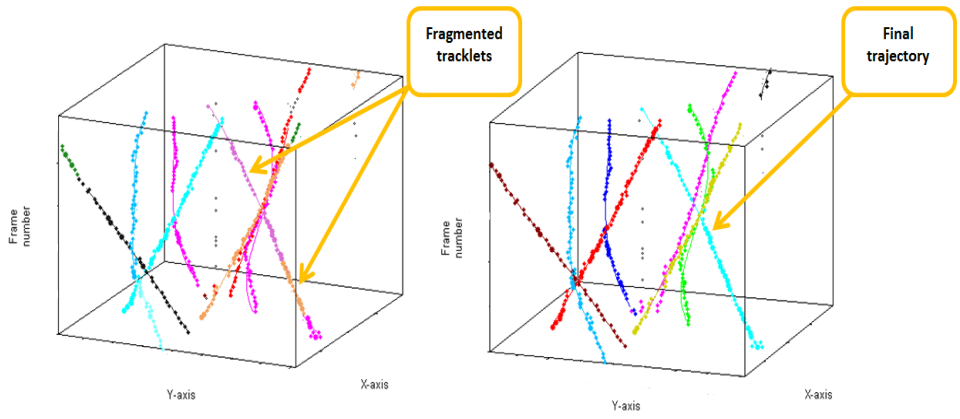


Figure 3: An illustration of the final trajectory produced by linking the fragmented tracklets (initialised tracklets) over the temporal segments of the '*PETS (S2.L1)*' video sequences.

trackelts' affinity array is more effective than the alternative additive scheme in the proposed tracking framework. For example, the appearance affinity value Λ_A gives higher scores to the distinguishable tracklets, which are discriminated easily from the background or other targets. Thus, it can be considered as an importance weight for other two tracklet affinity models. Moreover, the appearance affinity value of the interacting targets, when they are closely moving and/or partially occluding each other, is close to 1 and it has no effect on the total tracklet affinity function.

3.1. Preliminaries

Considering the index set of all graph vertices $I = \{1, \cdots, m\}$, taking any subset $S \subseteq I$ into account, a subgraph G_S of G with the vertex set $V_S = \{v_i \mid i \in S\}$ is denoted. Given a relation affinity graph on the vertices, the aim is to produce the dense subgraphs (partitions of the graph vertices) that agree as much as possible to constitute the high average affinity in the form of the following quadratic optimisation problem:[5]

$$\max_{y} f(y) = y^T A y, \quad s.t. \quad y \in \Delta \tag{2}$$

where y represents a set of vertices by a probabilistic subgraph (target tracklet), which is a unit vector in the space of standard *simplex* as $\Delta = \{y \in \mathcal{R}^m : y \geq 0, \sum_{i=1}^{m} |y_i| = 1\}$.

To this end, considering a set of all subgraphs $\mathcal{G}$ in the graph G and a *simplex projection* as $F : \mathcal{G} \rightarrow \Delta$, a mapping vector over the subgraph G_S is defined as

[5] y is relaxed into the continuous space Δ.

Algorithm : Retrieving the optimal dense subgraphs

Input : The initial clusters (subgraphs) $\{C_{y^*}\}$ represented by the local maximisers y^* and the corresponding affinity array A, set $S = \emptyset$ and $f = -\infty$

1: Arrange the values of y^* in descending order according to their probabilities to indicate clusters (subgraphs);
2: **for** $i = 1$ **to** r **do**
3: Build the set $\tilde{y}$ closets to y^* to approximate the dense subgraphs and set its elements sequentially based on the number of vertices in $S \leftarrow S \cup \{y^*(i)\}$, where $\tilde{y}_k = \frac{1}{i}$ for all $k \in S$; otherwise set to zero;
4: If $\tilde{y}^T A \tilde{y} > f$, then set f to the new average affinity $\tilde{y}^T A \tilde{y}$; otherwise, break;
5: **end for**

Output: The optimal points $\tilde{y} \in \Delta_{\mathcal{G}}$ representing the unique set of the optimal clusters $\{C_{\tilde{y}}\}$

Figure 4: An algorithm for retrieving the optimal dense subgraphs.

$F(G_S) = y$ such that $y_i = \frac{1}{n}$ if the i^{th} vertex is included in a probabilistic subgraph G_S and $v_i = 0$ otherwise. n is the number of vertices in the subgraph G_S and $F(\mathcal{G})$ is denoted as $\Delta_{\mathcal{G}}$ where $\Delta_{\mathcal{G}} = \left\{y \in \Delta \mid \exists n > 0 \quad \forall i = 1, \cdots, m \quad y_i \in \left\{0, \frac{1}{n}\right\}\right\}$. The indices of all nonzero elements of $y \in \Delta$ comprise its compactness, represented as $\phi(y) = \{i \mid y_i \neq 0\}$. Here, if the local maximizer y^* of Eq. (2) belongs to the space of simplex $\Delta_{\mathcal{G}}$, then the optimal dense subgraph is recovered as $G_{\phi(y^*)}$; Otherwise, a *greedy* strategy is proposed to approximate the optimal solution via considering the subgraph's dense neighbourhood (see Algorithm in Fig. 4). The suggested iterative algorithm consists of two iterative steps, including *condensation* and *trajectory growth*, as follows:

Condensation Step.. Given the initialised tracklets (subgraphs), the well-known algorithm, *replicator dynamic* [19] is utilised to efficiently solve the optimisation problem of Eq. (2) on the small subgraph $G_S \subseteq G$ as:

$$(y_S)_i(t+1) = (y_S)_i(t) \frac{(A_S y_S(t))_i}{y_S(t)^T A_S y_S(t)} \quad i \in S \tag{3}$$

where y_S and A_S are the probabilistic indicator vector and the affinity matrix of the subgraph G_S, respectively. Here, the aim is to condense the current subgraph to form a denser subgraph. However, the probabilistic cluster of vertices y_S^* obtained by Eq. (3) may not indicate an optimal solution y^* of the graph G in the current video segment.

Trajectory Growth.. In this step, the aim is to add the most relevant dense neighbours of the current subgraph based on the affinity value between the candidate dense neighbour vertex and the current subgraph. In fact, this affinity

value indicates the *confidence score* of the neighbourhood vertices to measure their compactness with respect to other vertices within the current cluster (subgraph).

Theorem.. According to Liu *et al.* [18], the *reward* for those neighbourhood vertices not associated to the current subgraph G_S, is not bigger than $f(y^*)$ in Eq. (2). Therefore, the corresponding target tracklet is grown into its similar dense neighbours with the larger confidence scores to the current tracklet (subgraph) such as: $\{k \mid Score(y^*, v_k) > f(y^*), k \in I\}$. Thus, the aim is to find the update vector Δy in which $y^{new} = y^{old} + \Delta y$. The update vector Δy was efficiently computed as in [18] (For more details, please see [18]). By this way, the *densest* subgraphs are gradually revealed from the initial subgraphs. First, the initial subgraphs merge to a more compact subgraph, and then expand to its neighbours. These two steps iterate until the final target tracklets are discovered in each video segment.

4. Graph Affinity

4.1. Tracklet Smoothness

The target trajectories should be continuous and smooth in the spatio-temporal domain. For example, when a significant part of the target trajectory is occluded due to severe occlusion or extreme target object deformations, it is difficult to judge whether the tracklets (partial trajectories) extracted by the tracker within the video segment correspond to the unique target trajectory or should be considered as the fragmented tracklets, which belong to other targets. Therefore, reliable tracklets are needed, which are smooth in the spatio-temporal domain and more likely being associated with the same target. Here, similar to [20], *dense point* trajectories are utilised, which can be extended to those frames without any detection responses and properly distinguish targets under heavy occlusion caused by the background or other interacting targets. The goal is to find the common dense point trajectories overlapping between the set of bounding boxes of the detections within the tracklets $Tr(\tau_i, \tau_j)$ to build their mutual affinities $\Lambda_S(\tau_i, \tau_j)$ as shown in Fig. 5. Thus, the smooth affinity between their frames closest in time (t_1, t_2) is computed as:

$$\Lambda_S(\tau_i, \tau_j) = exp\left(-\sum_{o \in Tr(\tau_i, \tau_j)} \frac{\left|loc\left(tr_o, D_i^{t_1}\right) - loc\left(tr_o, D_j^{t_2}\right)\right|^2}{\sigma_l^2}\right) \quad (4)$$

where D_i^t is the detection response at frame t for the tracklet τ_i, $loc(tr_o, D_i^t)$ specifies the *Euclidean distance* between the dense (point) trajectory tr_o and the centre of detection bounding box D_i at frame t. The parameter σ_l^2 controls the sensitivity of the smoothness affinity value to the abrupt target motion change[6].

[6]Tracklet smoothness affinity value is set to $\sigma_l^2 = 4$, experimentally.

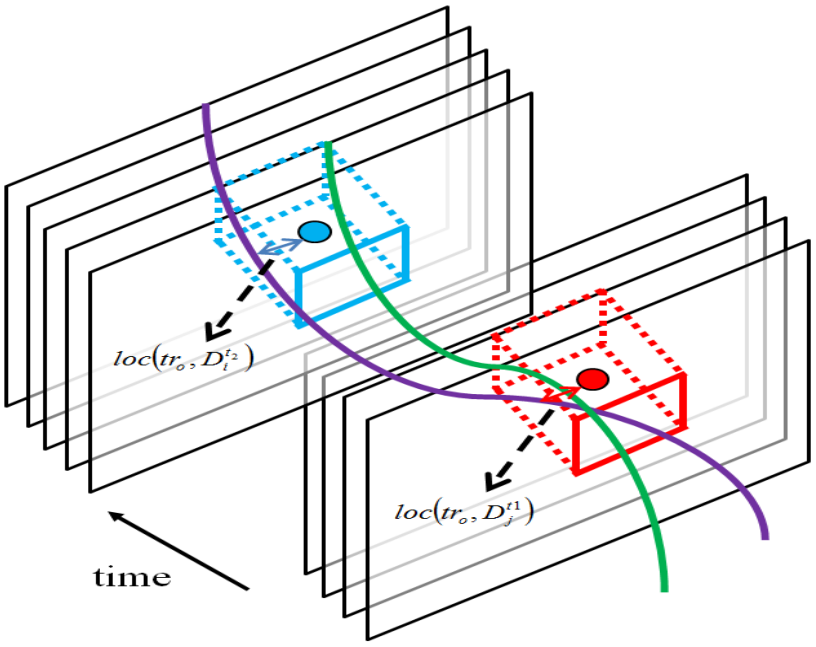


Figure 5: Illustration of point (dense) trajectories linking detection responses within the tracklets.

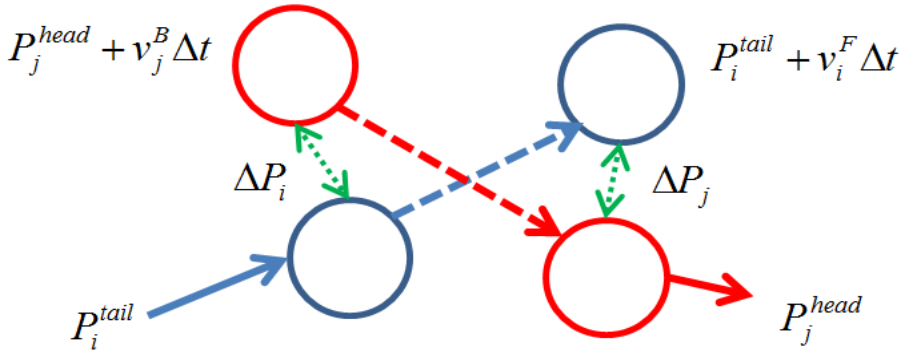


Figure 6: The tracklets motion model.

The affinity score for the two tracklets is higher when their common dense trajectories are able to keep up with their detection responses. $A_S(\tau_i, \tau_j) = 0$, if there are no common dense point trajectories between two tracklets, denoting the absence of effective information to measure their mutual affinities. Thus, the proposed affinity links those sparse target detections through their gaps, which are more likely consistent in the spatio-temporal domain.

4.2. Motion Affinity

The motion similarity between any two tracklets $\Lambda_M(\tau_i, \tau_j)$ is computed similar to [1], which is based on the relative distance between the predicted positions with the linear motion model and the real positions. Considering positions of the two tracklets P_j^{head} and P_i^{tail} with the frame gap Δt between them, the relative distances $\Delta P_i = \left(P_j^{head} + v_j^B \Delta t\right) - P_i^{tail}$ and $\Delta P_j = \left(P_i^{tail} + v_i^F \Delta t\right) - P_j^{head}$ are considered, in which the forward velocity v_i^F is evaluated from the head to the tail of tracklet i, while the backward velocity v_j^B is evaluated from the tail to the head of tracklet j (see Fig. 6)[7]. Finally, the motion affinity is measured with the estimated forward and backward motions of tracklets as:

$$\Lambda_M(\tau_i, \tau_j) = N(\Delta P_i, \Sigma)\, N(\Delta P_j, \Sigma) \tag{5}$$

[7]The positions and velocities along with x and y coordinates are predicted and updated in the Kalman filtering process.

where $N(\cdot, \Sigma)$ is a zero-mean Gaussian function[8]. Then, consistency along a target trajectory is considered by assuming linear motion between the closest detected bounding boxes on either side of the temporal axis.

4.3. Discriminative Appearance Affinity

The appearance model of the target object is a critical part for tracking. A reliable appearance model needs to establish a good balance between being adaptive, to account for appearance change caused by (e.g. pose changes), and being moderate in the case of partial occlusion, where we need to keep track of the target after being lost during tracking. Here, the goal is to learn the appearance model for the target object, which effectively makes it distinguishable not only from the background but also from other interacting targets. In order to learn the discriminative appearance models, a *metric learning* problem is defined to project high-dimensional tracklet features onto the learned low dimensional subspace in a way that each target tracklet is distinguishable from other tracklets.

Unlike the previous approaches [7], in which *ensemble* learning was utilised to combine a number of weak classifiers via boosting to learn a decision boundary for distinguishing each target object from the background or other target object, a computationally efficient metric function is designed to compute the appearance affinities between the tracklets. First, the training samples (high confidence detection responses) are collected from the initial tracklets. For each target candidate (detection response), an appearance descriptor obtained by concatenating feature histograms of the *HSV* colour and the *Histogram of Oriented Gradient* (*HOG*) to take advantage of both *colour* and *shape* information.

Given the training samples $\{(f_i), c_i\}_{i=1}^{T}$, where f_i is the feature vector belonging to the i^{th} tracklet and c_i is its corresponding tracklet label, learning the target appearance model is formulated as a distance metric function, which can enhance discrimination between feature vectors by ensuring that distances are smaller when features are extracted within a tracklet of the same target, and larger otherwise. The distance function is modelled using the *Mahalanobis* distance as follows:

$$d(f_i, f_j) = (f_i - f_j)^T M (f_i - f_j) \tag{6}$$

where M is the parameter of the Mahalanobis distance, which can be efficiently learned using an adaptation of the *Large Margin Nearest Neighbour* (*LMNN*) framework [21]. In fact, by solving the parameter M, as a decomposition problem $M = LL^T$, a mapping L is learned to transform feature vectors into the new space as $\hat{f} = L.f$. To compute the affinity score between the two tracklets, the relative distance between their learned discriminative projection is calculated

[8]The covariance for forward and backward motions is set to $\Sigma = diag\left[25^2 \quad 75^2\right]$ and fixed for all experiments.

as:

$$\Lambda_A(\tau_i, \tau_j) = \frac{L^T f(\tau_i) . L^T f(\tau_j)}{\|L^T f(\tau_i)\| . \|L^T f(\tau_j)\|} \quad (7)$$

4.4. Initial Tracklets

In order to obtain the initial target tracklets, a *heuristic* approach is used to create a set of short tracklets as the input of the proposed approach. For each detection at time t, if it belongs to the set of non-collision detection responses of one target, it should correspond to the closest detection among the detection responses of the next video frame, while the second closest detection would correspond to a different target. Thus, for each detection, its *closest* and *second* closest detection for the next frame are considered and the ratio of their distance[9] is measured. A ratio smaller than a threshold[10] means that both detections are more likely representing the same target.

5. Experiments and Results

In this paper, the performance of the proposed tracking framework is evaluated with several state-of-the-art trackers, such as globally tracking on discrete grid (KSP) [12], GMCP tracker [9], GMMCP-tracker [8], linear programming multiple people tracker [22], continuous energy minimisation based tracker [14], DCO tracker [15], globally greedy tracker [11], H^2T tracker [17], identity-aware network flow tracker [3], pairwise network flow tracker[13], [23] and online multi-object tracker proposed by Bae and Yoon [7] on the standard dataset. The performance of the proposed tracker on the sports videos is also examined. Table 2 shows a quantitative comparison of the proposed system to previous methods. For a fair comparison, all results are computed using the code provided by the authors and the same pre-trained *detector* and ground truth are used. Fig. 7 shows some tracking results of the proposed method. In the majority of cases, the proposed framework outperforms state-of-the-art methods in terms of the mentioned performance measures (Section 5.2).

5.1. Parameters Setting

The parameters of the proposed tracking algorithm are fixed in all experiments. Some parameters are found empirically, and most parameters fixed for all datasets. The parameters do not affect the overall performance of the proposed tracker much. For the appearance model, the HSV colour histogram with 192 bins is utilised and to capture the shape information, the HOG feature is implemented by setting the cell size to be 8 and concatenate these two feature

[9] Position and object size features are used.

[10] The threshold is empirically set to 0.25 for a less crowded scene (less than 8 $persons/frame$) and 0.6 for a high density crowded scene (greater than 10 $persons/frame$).

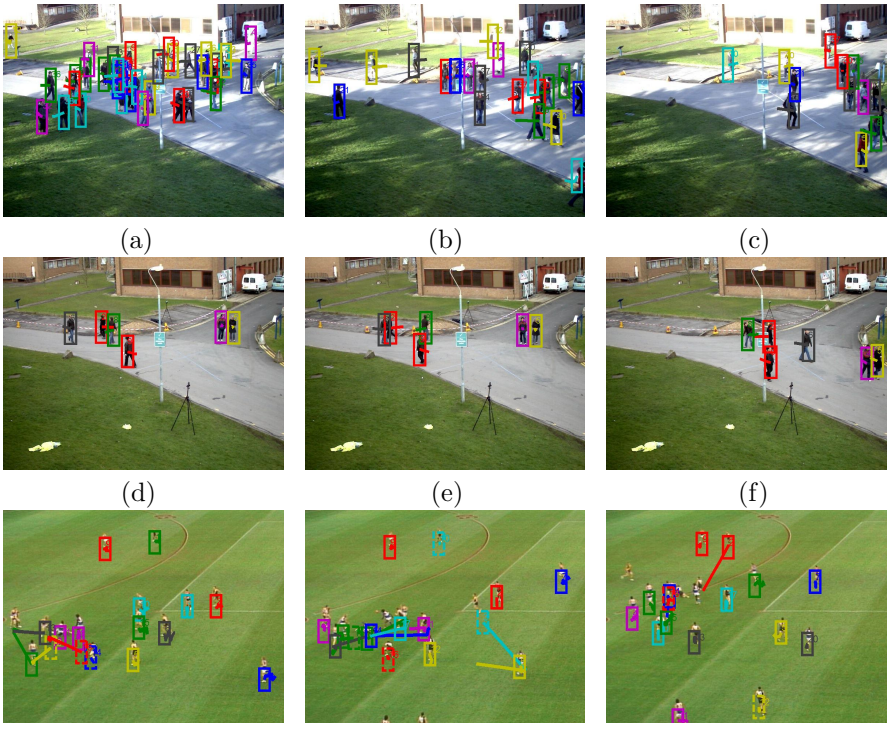


Figure 7: Sample tracking results of the proposed tracker in sequences (a-c) '*PETS (S2.L2)*' dataset, frames #53, #145 and #284 from left to right, (d-f) '*PETS (S2.L1)*' dataset, frames #298, #304 and #360 (left to right) and (g-h) '*AFL*' dataset, frames #46, #78 and #139 (left to right), respectively. Different colours represent different targets. Lines represent target trajectories over 5 frames.

elements into one vector. Every $8 - 12$ frames are combined to generate one segment of the video.

5.2. Performance Measure

Quantitative evaluation of multiple target tracking is a difficult task due to different assignment strategies, annotation and metric descriptions. In this paper, different performance measures are utilised, including the *CLEAR MOT* metrics [24] and metrics from [25, 26]:

- **MOTA** $(\uparrow) = 1 - \frac{\sum_t m_t + fp_t + mm_t}{\sum_t g_t}$: combines three error types: missed targets m_t, false positive fp_t and the identity switches or mismatches mm_t, respectively at frame t.
- **MOTP** $(\uparrow)$: Measures the localisation error of the tracker computed by intersection areas between bounding boxes of tracking results and ground truth over union areas of the bounding boxes.
- **FP** $(\downarrow)$: The false positives.
- **IDS** $(\downarrow)$: The number of identity switches.
- **GT:** The number of targets in the ground truth.
- **MT** $(\uparrow)$: The number of mostly tracked targets, which their trajectories are successfully tracked for more than 80% of their groundtruth time span.
- **ML** $(\downarrow)$: The number of mostly lost targets, which the trajectories are only tracked in less than 20% of their groundtruth time span.
- **FG** $(\downarrow)$: The number of fragmented tracklets.
- **REC** $(\uparrow)$: Recall, the ratio of the correctly matched detections over the total number of detections in ground truth.
- **PRE** $(\uparrow)$: Precision, the ratio of the correctly matched detections over the total number of output detections.
- **FAF** $(\downarrow)$: The number of false alarms per frame.

Here, the symbol $\uparrow$ means that higher scores indicate better results, and $\downarrow$ means that lower scores specify better performances.

5.3. Datasets results

We evaluate the performance of the proposed approach on sports video sequences.

- **Sports Videos.** Two new sports video datasets, namely *Women Soccer* and *Field Hockey*, are introduced. The former is collected via a *panning* camera (1444×806 pixels) and the latter contains high resolution frames (1132×638 pixels) from four views (only one view is used in our experiments). There exist different tracking difficulties such as *mutual occlusion* between the soccer players in *Women Soccer* dataset. In addition, the challenging sport video sequence (*AFL*) is utilised, in which there are complex interactions among a crowd of players. For the main baseline, SMOT [27] is used, which is specifically proposed for tracking visually similar players in sports videos by relying on motion similarity and utilising a generalised linear assignment to recover long-term tracks. In these videos, SMOT [27] exhibits a good performance in the absence of detection noise. However, it fails in recovering the true trajectories of the targets in the realistic scenarios. In DCO [15], since the optimisation reaches only a moderate local minimum, therefore, it produces many fragmented target tracklets leading to a high number of interrupted trajectories (*FG*).

  The proposed approach is also examined on challenging publicly available video sequences.

- **PETS 2009.** The *PETS 2009* dataset[28] is captured from multiple cameras, and only sequences (*PETS S2.L1* and *PETS S2.L2*) captured from *view-1* are used with frame size of 768×576 pixels in the experiments. *PETS S2-L1* includes 795 frames showing up to 8 similar walking pedestrians. The dataset's difficulty stems from the presence of *nonlinear motion*. *PETS S2.L2* is even more difficult than *PETS S2.L1*, due to its crowded density of the pedestrians and frequent mutual occlusions between them. The proposed system is capable to reach the *lowest ID* changes especially for the targets leaving and re-entering the scene. In majority of the used metrics, the proposed tracker performs better.

- **TUD.** *TUD-Crossing* and *TUD-Campus* [29] are two short sequences with frequent occlusions of the side-view pedestrians, containing 201 and 71 frames, respectively. Compared to other methods such as [15], which requires an explicit parameter estimation step, the proposed framework is much faster with fewer parameters. The results of the proposed tracker are averaged over Campus and Crossing sequences.

- **Parking Lot.** This sequence includes $1,000$ frames of a largely crowded scene with up to 14 pedestrians walking in parallel. It includes missed detections and parallel motion with similar appearances. The proposed tracker achieves the lowest number of *ID switches* as a result of correctly linking the fragmented trajectories. Furthermore, with the apt trajectories recovering strategy, the quantitative result outperforms others in terms of *recall* and *overall accuracy*.

- **TownCentre.** The HD dataset, namely *TownCentre* sequence of size 1920×1080 pixels is captured from a busy town, where there is varying

number of people who are walking[11]. In addition, some people become partially occluded during this video.

5.4. Evaluation of the main system parts

To verify the effectiveness of the main parts of the proposed approach, two different experiments are devised to compare the performance of the system with different constraints:

- **Exp. 1:** Eliminate the learned appearance models of the targets.
- **Exp. 2:** Eliminate the trajectory smoothness part in the proposed relation affinity graph.

The average results of evaluation metrics are provided for the datasets include *PETS.L1*, *PETS.L2*, *AFL* and *Women soccer* sequences, respectively in Table 3. From the accuracy results, the effectiveness of each different part of the proposed tracking framework is examined.

In *Exp. 1*, the proposed discriminative target appearance models are replaced by the appearance model of using the *Bhattacharyya* distance of multi-cue feature histograms. From the observations, the *MOTP* and *MOTA* measures have been declined noticeably since the discrimination power is reduced. The tracking results of the proposed tracker *without* considering the trajectory smoothness part (*Exp. 2*) is demonstrated in Fig. 8. As anticipated, the overall system improves tracking framework performance for the two metrics, while the other experiments (*Exp. 1*)-(*Exp. 2*) are still comparable with the benchmark.

Moreover, the proposed tracker's sensitivity to the chosen parameters including Σ and σ is examined for the different sequences and the obtained results are shown in Table 4 and Table 5, respectively. Since the size of tracked target will be changed from frame to frame, the standard deviation σ for the target size noises has been set to 2 experimentally to handle the size variations. This scale has be determined for a target height of 160 pixels and automatically adjusted by the estimated height of tracked object. With increasing target' size deviation σ and target tracklets' forward/backward motion covariance Σ, the tracking performance will slightly decrease due to the more pairwise relationships between incorrect target tracklets in the defined tracklet affinity model. However, these changes are quite small and do not affect the proposed tracker's performance.

5.5. Reliability of the Tracklets

To demonstrate the effectiveness of the designed tracklet affinity measurements, which fit into the iterative subgraph extraction framework, the following criteria are considered:

[11] In average, 16 people are walking simultaneously.

Table 2: Performance comparison between the proposed tracker and other state-of-the-art methods. The same detectors and ground truths are used for the trackers on the different datasets. Tracking results that are not available are labelled by '−'. The best results is shown in **Bold**.

Dataset	Method	MOTP (%)	MOTA (%)	GT	ML	MT	IDS	FP	FG	REC	PRE	FAF
	Proposed method	**81.3%**	**93.4%**	23	**0**	**22**	**0**	**34**	**3**	**99.2%**	**98.7%**	0.09
	[14]	80.2%	91.6%	23	1	21	11	59	6	92.4%	98.4%	**0.07**
PETS	[15]	74.3%	83.6%	19	0	18	22	–	–	96.9%	94.1%	0.36
(S2.L1)	[12]	72.0%	80.3%	23	2	17	28	126	22	83.8%	96.3%	0.16
	[9]	69.0%	90.3%	–	–	–	8	–	–	96.4%	69.4%	–
	[11]	74.3%	77.4%	23	1	14	57	93	62	81.2%	97.2%	0.12
	[17]	72.9%	92.7%	23	0	22	5	62	10	94.4%	98.4%	0.08
	[7]	69.7%	80.3%	23	0	22	3	–	2	99.0%	90.2%	0.18
	Proposed method	63.4%	**68.2%**	74	**2**	**34**	**12**	**185**	**25**	**75.6%**	**95.7%**	0.58
	[14]	59.4%	56.9%	74	12	28	99	622	73	65.5%	89.8%	1.43
PETS	[15]	59.8%	46.0%	74	8	25	126	–	105	–	–	–
(S2.L2)	[12]	60.9%	24.2%	74	40	7	22	193	38	26.8%	92.1%	**0.44**
	[11]	**64.1%**	45.0%	74	17	7	137	199	216	49.0%	95.4%	0.46
	[17]	52.7%	62.1%	74	3	27	125	640	175	71.2%	90.3%	1.47
	[7]	63.4%	63.89%	74	–	–	28	–	51	68.2%	88.83%	1.68
	Proposed method	**78.0%**	**81.1%**	13	1	10	0	45	**5**	**77.3%**	**90.4%**	1.02
TUD	[30]	**69.0%**	78.8%	–	–	–	2	–	–	–	–	–
	[31]	75.0%	78.0%	–	–	6	1	–	8	–	–	–
	Proposed method	**83.2%**	**92.1%**	14	0	11	12	45	**15**	**92.3%**	98.1%	**0.14**
Parking	[32]	76.5%	73.1%	14	0	11	83	253	70	86.8%	89.4%	1.01
Lot	[9]	74.1%	90.4%	14	–	–	–	–	–	85.3%	98.2%	–
	[11]	75.3%	65.7%	14	1	1	52	**39**	60	69.4%	97.8%	0.16
	[17]	81.9%	88.4%	14	0	11	21	**39**	23	90.8%	**98.3%**	0.16
	[3]	69.3%	90.7%	14	0	11	**3**	–	–	–	–	–
	Proposed method	**76.5%**	**66.8%**	16	**1**	**14**	**6**	35	**15**	**81.0%**	**89.0%**	0.84
	[27]	60.8%	16.7%	16	3	2	14	–	38	52.0%	59.8%	–
AFL	[15]	63.3%	29.7%	16	2	3	97	–	93	56.3%	70.9%	–
	[33]	63.6%	41.4%	16	2	7	22	–	39	65.8%	73.2%	–
	Proposed method	**94.7%**	**92.3%**	9	**0**	**9**	**0**	5	**3**	**98.4%**	**99.1%**	**0.1**
Women	[27]	65.9%	46.5%	9	2	6	4	10	22	59.0%	65.7%	0.85
Soccer	[15]	63.8%	40.3%	9	3	4	6	23	31	55.5%	64.9%	0.9
	[33]	67.0%	44.3%	9	2	4	6	19	19	64.6%	72.1%	0.73
	Proposed method	**77.6%**	**79.4%**	16	5	**88**	**39**	**22**	**19**	**83.3%**	**85.5%**	**0.45**
	[9]	71.9%	75.9%	16	–	–	–	–	–	–	–	–
Town	[23]	70.7%	63.4%	16	–	–	–	–	–	–	–	–
Centre	[22]	71.5%	67.3%	16	–	–	–	–	–	–	–	–
	[8]	66.3%	77.3%	16	**4**	86	68	–	–	–	–	–
	Proposed method	**86.7%**	**90.2%**	13	**1**	**11**	**3**	**10**	**7**	**82.9%**	**87.4%**	**0.32**
	[9]	71.0%	88.3%	13	5	6	10	–	22	69.7%	81.9%	–
Field	[15]	75.3%	88.5%	13	4	7	9	–	18	71.1%	83.6%	–
Hockey	[3]	65.1%	89.2%	13	2	10	10	–	15	75.7%	72.3%	–
	[13]	78.2%	87.5%	13	2	9	7	–	11	74.4%	70.0%	–

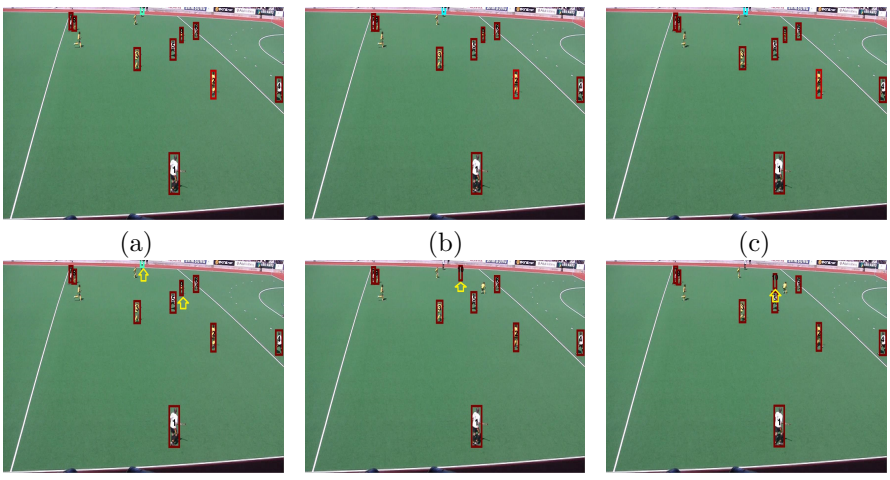


Figure 8: Tracking results from the '*Field Hockey*' dataset. (a-c) Frames #121, #122 and #123 from left to right *with* considering trajectory smoothness part and (d-f) the same frames without considering trajectory smoothness part, respectively. An identity (ID) switch occurs between the targets 9 and 10, when trajectory smoothness part is not taken into account.

Table 3: Quantitative evaluation results for the main system parts.

Datasets	Brief description	MOTP (%)	MOTA (%)
AFL and Women Soccer	Milan *et al.* [15]	63.5%	35.0%
	Dicle *et al.* [27]	63.3%	31.6%
	Exp. 1	79.5%	51.2%
	Exp. 2	82.3%	65.0%
	Overall system	85.6%	79.5%
PETS	Milan *et al.* [15]	67.0%	64.8%
	Hamed *et al.* [11]	69.2%	61.2%
	Exp. 1	68.4%	70.1%
	Exp. 2	70.1%	74.5%
	Overall system	72.3%	80.8%

Exp. 1– System without the learned appearance models.
Exp. 2– System without the trajectory smoothness part.

Table 4: Sensitivity test results for the motion covariance.

Datasets	Motion Covariance	MOTP (%)	MOTA (%)
PETS (S2.L2)	$\Sigma = diag\begin{bmatrix}15^2 & 65^2\end{bmatrix}$	63.7%	68.7%
	$\Sigma = diag\begin{bmatrix}20^2 & 70^2\end{bmatrix}$	63.5%	68.5%
	$\Sigma = diag\begin{bmatrix}20^2 & 75^2\end{bmatrix}$	63.5%	68.4%
	$\Sigma = diag\begin{bmatrix}25^2 & 75^2\end{bmatrix}$	63.4%	68.2%
	$\Sigma = diag\begin{bmatrix}25^2 & 80^2\end{bmatrix}$	62.9%	67.8%
	$\Sigma = diag\begin{bmatrix}35^2 & 85^2\end{bmatrix}$	62.1%	67.3%
	$\Sigma = diag\begin{bmatrix}45^2 & 95^2\end{bmatrix}$	61.7%	66.9%
TUD	$\Sigma = diag\begin{bmatrix}15^2 & 65^2\end{bmatrix}$	78.4%	81.3%
	$\Sigma = diag\begin{bmatrix}20^2 & 70^2\end{bmatrix}$	78.2%	81.2%
	$\Sigma = diag\begin{bmatrix}20^2 & 75^2\end{bmatrix}$	78.1%	81.2%
	$\Sigma = diag\begin{bmatrix}25^2 & 75^2\end{bmatrix}$	78.0%	81.1%
	$\Sigma = diag\begin{bmatrix}25^2 & 80^2\end{bmatrix}$	77.7%	80.6%
	$\Sigma = diag\begin{bmatrix}35^2 & 85^2\end{bmatrix}$	77.3%	80.1%
	$\Sigma = diag\begin{bmatrix}45^2 & 95^2\end{bmatrix}$	76.9%	79.7%

Table 5: Sensitivity test results for the variance σ.

Datasets	Variance σ	MOTP (%)	MOTA (%)
PETS (S2.L2)	$\sigma_l^2 = 3$	63.4%	68.4%
	$\sigma_l^2 = 4$	63.4%	68.2%
	$\sigma_l^2 = 6$	63.2%	67.9%
	$\sigma_l^2 = 8$	63.1%	67.4%
	$\sigma_l^2 = 10$	62.6%	67.2%
	$\sigma_l^2 = 12$	62.5%	67.1%
TUD	$\sigma_l^2 = 3$	78.1%	81.2%
	$\sigma_l^2 = 4$	78.0%	81.0%
	$\sigma_l^2 = 6$	77.8%	80.7%
	$\sigma_l^2 = 8$	77.5%	80.4%
	$\sigma_l^2 = 10$	77.3%	80.3%
	$\sigma_l^2 = 12$	77.1%	80.1%

- (C_1) **Tracklet-Target Hypothesis Similarity**: Higher similarity between a target tracklet and the corresponding target hypothesis in each video frame represents the reliability of the tracklet.
- (C_2) **Robustness to Occlusions**: The target tracklets should be robust to occlusion and missed detections.
- (C_3) **Tracklet Cardinality**: A long tracklet is more likely to be a true trajectory of the target object since it contains more statistical target information.

Therefore, the tracklet reliability (confidence score) can be formulated based on the above requirements according to Eq. (8), which can be explicated as how well the obtained target tracklet by the proposed framework, matches the real trajectory of the target object during different scenarios (e.g. occlusion, appearance changes):

$$Rel\left(\tau_i\right) = \frac{Length\left(\tau_i\right) - w}{Length\left(\tau_i\right)} . A\left(\tau_i, O_k^i\right) \quad , k \in \left[t_1^i : t_2^i\right] \tag{8}$$

where w is the number of the frames the target i is lost or confused due to the occlusion (target-target occlusion). $A\left(\tau_i, O_k^i\right)$ is the affinity score (Eq. (1)) between the target hypothesis and the corresponding tracklet. The obtained confidence score lies in [0 1]. The reliability of different female soccer player trajectories from the *Women Soccer* dataset, is illustrated in Fig. 9. The darker the colour of the bounding box is, the more reliable the specific target tracklet has been for the frame sequences.

5.6. Video Segment Size

In sports videos, such as the *Field Hockey* dataset, the players exhibit complex interactions and frequent occlusions occur within the group of players. Each of these difficulties violates the assumption of smooth movement for a relatively long period of time. Therefore, the video segment size should be chosen in a way that this assumption is not violated. As it shown in Fig. 10 , the tracking *accuracy* and *precision* for the *Field Hockey* dataset increased slightly by a rise

Figure 9: Some tracking results from the '*Women Soccer*' sequence. The darker the colour is, the more reliable the specific target tracklet has been.

in the number of segment's frames from 6 to 10, then decreased to less than 88% *MOTP* for 20 frames in a video segment.

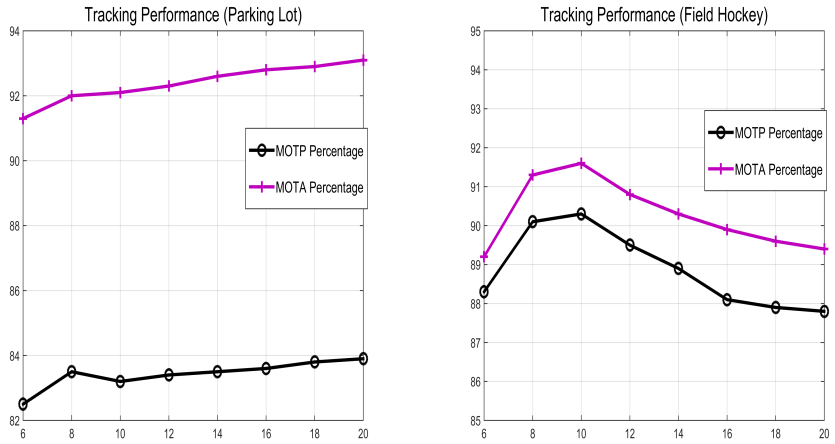
In contrast, for some other video sequences, where interactions among the group of targets may be relatively simple, such as parallel walking of the pedestrians in the *Parking Lot* dataset, with increasing the number of frames per segment, the performance of the tracker will improve constantly from 82.5% MOTP to 83.8% (*black* curve in Fig. 10) and from 91.3% MOTP to 92.9% (*pink* curve in Fig. 10). This is due to having more statistics on *colour* and *appearance* features of the targets for the proposed tracking framework to link target tracklets jointly across space and time, which results in more robust associations over the sequence. However, in general, the tracking performance remained steady at about $8-12$ frames per segment and is not very sensitive to the video segment size.

5.7. Run Time and Convergence

The proposed framework was implemented using a single 3.1 GHz core without any parallel programming. In the proposed iterative algorithm, the two steps (*trajectory growth* and *condensation step*) iterate till reaching the optimal solutions (*dense subgraphs*).

In the trajectory growth step, for each initial subgraph, the number of nearby vertices can be handled and the correlated vertices will always be added to the current subgraph. In the condensation step, some irrelevant vertices will be discarded, and only a very dense cluster of vertices will remain. Both the run time and convergence of the proposed algorithm depend on the density of the spatio-temporal graph of vertices[12] in the different videos. However, the proposed system performs on small subgraphs and thus is fast in run time. The

[12]The number of targets per frame.



(a) (b)

Figure 10: Comparison of tracking performance (*MOTP* and *MOTA* metrics) for different sizes of the video segment on (a) the 'Parking Lot' and (b) the 'Field Hockey' sequences, respectively.

main computational cost lies in the *condensation* phase. Considering the number of vertices in the largest subgraph as N and the number of iterations for the *replicator* equation Eq. (3) is t_r, then the time complexity of the condensation step is $O(Nt_r)$. The total time complexity is $O(Nt_rt_s)$, where t_s is the number of iterations for both steps. The computation expense of the proposed tracking system is not significantly increased by the number of the temporal segments. The reason is that there is a trade-off between the number of frames in each temporal segment (batch size) and its time processing. The smaller the number of frames in each segment, the faster the *condensation* step is obtained. In the experiments, for the less crowded scenes, such as *PETS S2.L1* (8 persons/frame), the processing time is ~ 0.05 s/frame, which is 20x faster than the best result of Milan *et al.* [15]. For crowded sequences, such as *AFL1*, the run time is ~ 0.09 s/frame. Moreover, in comparison with the best average run time results of Dehghan *et al.* [8] for a batch of 50 frames of *Parking Lot* sequence ($1.57s$), the proposed tracker is 10x faster.

6. Conclusion

In this paper, a multi-target tracking framework is formulated as a dense subgraph discovering problem where the aim is to iteratively recover dense clusters of target hypotheses, each belongs to the different targets. Unlike the previous approaches, which rely on computationally expensive optimisation algorithms without any guarantee to obtain the optimal solutions, in this paper, a general framework is proposed, which can automatically find the optimal target

tracklets. Both local and global cues are exploited to model the relationships between the tracklets. To increase the discriminative ability of the suggested system, the distinguishable appearance based models are learned for the targets, which would be beneficial for the performance of the proposed approach. In addition, the effectiveness of each tracking main part is evaluated. The experimental results on the challenging tracking datasets have shown the improved performance of the proposed approach, compared to other state-of art tracking systems.

References

[1] B. Yang, R. Nevatia, An online learned crf model for multi-target tracking, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 2034–2041.

[2] J. Liu, P. Carr, R. T. Collins, Y. Liu, Tracking sports players with context-conditioned motion models, in: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2013, pp. 1830–1837.

[3] A. Dehghan, Y. Tian, P. H. Torr, M. Shah, Target identity-aware network flow for online multiple target tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1146–1154.

[4] G. Shu, A. Dehghan, O. Oreifej, E. Hand, M. Shah, Part-based multiple-person tracking with partial occlusion handling, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 1815–1821.

[5] W. Brendel, M. Amer, S. Todorovic, Multiobject tracking as maximum weight independent set, in: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2011, pp. 1273–1280.

[6] K. Shafique, M. Shah, A noniterative greedy algorithm for multiframe point correspondence, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (1) (2005) 51–65.

[7] S.-H. Bae, K.-J. Yoon, Robust online multi-object tracking with data association and track management, IEEE Transactions on Image Processing 23 (7) (2014) 2820–2833.

[8] A. Dehghan, S. M. Assari, M. Shah, GMMCP Tracker: Globally Optimal Generalized Maximum Multi Clique Problem for Multiple Object Tracking, in: CVPR, Vol. 1, 2015, p. 2.

[9] A. R. Zamir, A. Dehghan, M. Shah, GMCP-tracker: Global multi-object tracking using generalized minimum clique graphs, in: Computer Vision–ECCV 2012, Springer, 2012, pp. 343–356.

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

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

[12] J. Berclaz, F. Fleuret, E. Turetken, P. Fua, Multiple object tracking using k-shortest paths optimization, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (9) (2011) 1806–1819.

[13] V. Chari, S. Lacoste-Julien, I. Laptev, J. Sivic, On pairwise costs for network flow multi-object tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5537–5545.

[14] A. Milan, K. Schindler, S. Roth, Continuous Energy Minimization for Multi-Target Tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (1) (2014) 58–72.

[15] A. Milan, K. Schindler, S. Roth, Detection-and trajectory-level exclusion in multiple object tracking, in: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2013, pp. 3682–3689.

[16] R. Kumar, G. Charpiat, M. Thonnat, Multiple Object Tracking by Efficient Graph Partitioning, in: Computer Vision – ACCV 2014 – 12th Asian Conference on Computer Vision, Lecture Notes in Computer Science, Vol. 9006, 2015, pp. 445–460.

[17] L. Wen, W. Li, J. Yan, Z. Lei, D. Yi, S. Z. Li, Multiple target tracking based on undirected hierarchical relation hypergraph, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2014, pp. 1282–1289.

[18] H. Liu, L. J. Latecki, S. Yan, Fast detection of dense subgraphs with iterative shrinking and expansion, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (9) (2013) 2131–2142.

[19] J. W. Weibull, Evolutionary game theory, MIT press, 1997.

[20] K. Fragkiadaki, W. Zhang, G. Zhang, J. Shi, Two-granularity tracking: Mediating trajectory and detection graphs for tracking under occlusions, in: Computer Vision–ECCV 2012, Springer, 2012, pp. 552–565.

[21] K. Q. Weinberger, L. K. Saul, Distance metric learning for large margin nearest neighbor classification, The Journal of Machine Learning Research 10 (2009) 207–244.

[22] L. Leal-Taixé, G. Pons-Moll, B. Rosenhahn, Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker, in: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), IEEE, 2011, pp. 120–127.

[23] S. Pellegrini, A. Ess, L. Van Gool, Improving data association by joint modeling of pedestrian trajectories and groupings, in: Computer Vision–ECCV 2010, Springer, 2010, pp. 452–465.

[24] K. Bernardin, R. Stiefelhagen, Evaluating multiple object tracking performance: the CLEAR MOT metrics, EURASIP Journal on Image and Video Processing 2008 (2008) 246309.

[25] C.-H. Kuo, C. Huang, R. Nevatia, Multi-target tracking by on-line learned discriminative appearance models, in: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 685–692.

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

[27] C. Dicle, O. I. Camps, M. Sznaier, The way they move: Tracking multiple targets with similar appearance, in: 2013 IEEE International Conference on Computer Vision (ICCV), IEEE, 2013, pp. 2304–2311.

[28] J. Ferryman, A. Shahrokni, An overview of the PETS 2009 challenge, in: Proceedings of the Eleventh IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, IEEE, 2009.

[29] A. Andriyenko, K. Schindler, Multi-target tracking by continuous energy minimization, in: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2011, pp. 1265–1272.

[30] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, L. Van Gool, Online multiperson tracking-by-detection from a single, uncalibrated camera, Pattern Analysis and Machine Intelligence, IEEE Transactions on 33 (9) (2011) 1820–1833.

[31] A. V. Segal, I. Reid, Latent data association: Bayesian model selection for multi-target tracking, in: Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE, 2013, pp. 2904–2911.

[32] A. Andriyenko, K. Schindler, S. Roth, Discrete-continuous optimization for multi-target tracking, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 1926–1933.

[33] A. Milan, R. Gade, A. Dick, T. B. Moeslund, I. Reid, Improving Global Multi-target Tracking with Local Updates, in: Computer Vision - ECCV 2014 Workshops, Lecture Notes in Computer Science, Vol. 8927, Springer, 2015, pp. 174–190.