Advances in Intelligent Systems Research (AISR), volume 155
2018 International Conference on Advanced Control, Automation and Artificial Intelligence (ACAAI 2018)

Heuristic Maximum Clique Based Identity Switches Awareness for Tracking

Chao Li[1,2], Di Cui[1,*], Hao Sheng[1,2], Jiahui Chen[1,2] and Yang Zhang[1,2]
[1] State Key Laboratory of Software Development Environment, School of Computer Science and Engineering, Beihang University, Beijing 100191, P.R.China
[2] Shenzhen Key Laboratory of Data Vitalization, Research Institute in Shenzhen, Beihang University, Shenzhen, P.R. China

***Abstract*— Multi-target tracking is a popular research topic in the field of computer vision. Most recent works based on tracking-by-detection paradigm have trouble when dealing with tracking errors (i.e. identity switches) that are caused by overlapping of different pedestrians. In this work, we propose a novel method which is capable to perceive the mismatched tracklets by using an Identity Switch Awareness Algorithm based on iterative maximum clique searching. Besides, we present an optimal method to accelerate the maximum clique searching process to alleviate the cost of computing. Finally, we introduce this algorithm into a network flow tracking method and test the validity of it on five challenging sequences. The results show obvious improvement on MOTA and IDs.**

Keywords—Multi-target tracking; Identity Switches; Maximum clique problem

I. INTRODUCTION

Multi-target tracking has received much attention in recent years due to its wide range of applications such as video surveillance [1] and human-computer interaction [2]. The task of multi-target tracking is to recover the spatio-temporal trajectories of pedestrians. Despite of recent progress in the field , tracking remains a challenging problem since occlusions and false detections problems are still unsolved.

In recent years, tracking-by-detection [3] benefited from the considerable improvements in object detection and became one of the most popular paradigms in multi-target tracking framework. This paradigm can be accomplished in two steps: firstly, an object detector extract target detections from each frame, then a tracker links these detections into multi-frame trajectories using data-association method. Data-association is an essential part for tracking since it evaluates similarity between different detections which helps recover detections to tracklets. However, considering the compute cost, most trackers conduct data-association only in a limited time-window, thus errors may occur because it is hard to distinguish from each other without enough context information when two or more pedestrians are close and have high appearance similarity in image space. Once an error occurs, it may propagates and accumulates during the tracking process and has bad effect on tracking results. The errors we mentioned here refers to identity switches (IDs), which is divided into two categories: identity switches within a tracklet and identity switches between tracklets. As shown in Figure 1, we name the former IDs as inner IDs, meaning that one tracklet includes more than one pedestrian. Correspondingly, we name the latter IDs as outer IDs, indicating one pedestrian is allocated to more than one tracklets. In this paper, we concentrate on the study of reducing the inner IDs.

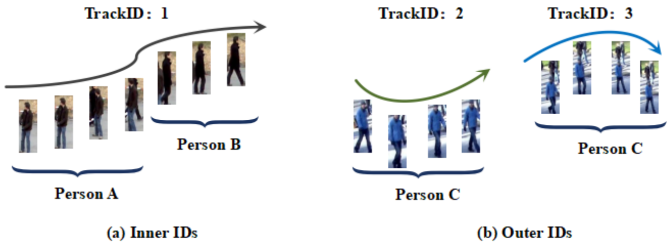

FIGURE I. TWO CATEGORIES OF ID SWITCH

Current tracking methods try a lot to decrease IDs. Approaches proposed in [4] and [5] consideredsocial behavior and motion model to eliminate ambiguity of detections. Moreover, global information is taken into account in approach [6] and IDs decrease significantly. However, although these methods can reduce potential linking errors in the tracking process, they cannot handle existing errors.

In this paper, we propose the identity switches awareness algorithm to distinguish different persons in one tracklet. We model this objective as a clustering problem and solve it using iterative maximum clique searching in two types of graphs, one is a node graph built on the similarity between pedestrian detections, while the other is a clique graph constructed based on the correlation between cliques. Besides, we optimize the backtracking method to accelerate the maximum clique searching process. Finally, we integrate IDs awareness information with network flow based tracking method. We verified our method on five sequences and achieve obvious improvement, especially in inner IDs.

The main contribution of this paper are:

•this paper defines a IDs awareness model to perceive IDs within each tracklet;

• a novel method to incorporate IDs information into multi-target tracking process;

• an efficient algorithm that optimizes the backtracking to accelerate the maximum clique searching process.

II. IDENTITY SWITCH AWARENESS ALGORITHM

In tracking process, each tracklet contains some detections and therefore can be regarded as a set with several nodes. We reformulate IDs awareness as detection clustering problem: when there are more than one cluster in a tracklet, it means IDs occurs. Here we solve the problem using maximum clique based iterative clustering method to perceive IDs.

A Iterative Clustering for Identity Switch Awareness

We propose an iterative clustering algorithm to cluster nodes into clusters. The whole process is described in Algorithm1. The iterative procedure will ultimately converge when clusters keep stable, which is the output of this algorithm.

As shown in Table.1, the input of Algorithm1 is one tracklet T containing some detections, while the output is a set of cliques, the number of which means the number of pedestrians in T. The algorithm includes two main parts: firstly, before we enter into iterative clustering stage, it is essential to build a graph based on all the vertices (getDetection() in line.3) as well as the connectivity between them. In our cases, there are two types of vertices, node and clique; thus we construct graphs based on two kinds of correlations, including correlation between nodes and correlation between cliques, which we explain in Ⅱ.B). Eventually, we accomplish ID switches awareness by iterative clustering based on a maximum cliques searching method we name as effective backtracking(EBT, which we fully discuss in Ⅱ.C);

In addition, there are two terminate events we set for circulations in Algorithm1. On the one hand, for the internal loop(the while loop begin at line.6), the symbol of end is all nodes in the graph has been went through. On the other hand, for the external loop(the while loop start from line.5) the ending flag is the size of graph is steady, meaning there will not be any clique or node merge with others.

B Graph Construction

There are two kinds of graphs in the process of iterative clustering, so we utilize three different ways to judge the correlation between nodes in two graphs. In the initial graph, we regard all the detections in one tracklet as a set of nodes, and edges(connectivity) between every two nodes are judged by their correlation. After that stage when cliques (which are identical to sub-tracklets) are generated, we consider all the cliques as nodes of the graph, thus the criterion of connectivity is based on correlations between every two cliques.

1) Construction of node graph

We regard each tracklet T as a set of detections D = $\{d_1,d_2,...,d_n\}$ and choose appearance model of each detection as feature $D_{ft} = \{app_1,app_2,...,app_n\}$, since the appearance model is definitively the most salient feature used in all current multi-target tracking algorithm. However, instead of using the traditional appearance model such as HOG and Optical Flow, here we adopt a new model trained from convolution neural network, which can better handle pedestrians' appearance as well as behavior change in longer-distance.Then for each tracklet, a similarity matrix S_{n-n} is generated as follow, where n is the number of detections in tracklet they belong to.

$$S_{n-n}(d_i, d_j) = \left\| app_i - app_j \right\|_2 \qquad d_i, d_j \, □ \, D. \tag{1}$$

Eventually, we define an undirected graph G = (V, E), where V = {v1, v2,....,vn} is equal to the set of detections D, E = (e_{ij}) is an n×n matrix representing connectivity among nodes. Since the connectivity of two nodes can be either connect or disconnect, eij is notated as follow:

$$e_{ij} = \begin{cases} 1 & i \neq j \text{ and } S_{n-n}(v_i, v_j) > thre_1 \\ 0 & i = j \text{ or } S_{n-n}(v_i, v_j) \leq thre_1 \end{cases}, \tag{2}$$

where $thre_1$ is a threshold that can be chosen according to the distribution of S_{n-n}, and the self similarity is set as 0 because this kind of correlation is not under consideration in our method.

TABLE I. ALGORITHM1 ITERATIVE CLUSTERING

```
Input: input one tracklet T = {d1, d2, ..., dn}

Output: output the clique set C = {c1, c2, ..., cn}
1:  C←∅, V←∅, E←∅
2:  if T ≠ ∅ then
3:      V ← getDetection(T)
4:          Build Node Graph G = (V, E)
5:      while true do
6:          while V ≠ ∅ do
7:              Max_Clique = EBT(G)
8:              C = C + Max_Clique
9:              V = V - Max_Clique
10:             Update(E)
11:             Build Node Graph G = (V, E)
12:         V ← getCliques(C)
13:         Update(E)
14:         Build Clique Graph G = (V, E)
15:         if G not changes in this iteration then
16:             break
17: return clique set C
```

2) Construction of clique graph

After the first step of our method, a set of cliques C= $\{c_1, c_2...c_n\}$has been generated. For we need to build new graphs to continue the circulation, correlation between cliques must be determined in this stage. And therefore for every two cliques c_i, c_j , we calculate their similarity S_{c-c} based on their feature F_c

$$S_{c-c}(c_i, c_j) = \| F_{c_i} - F_{c_j} \|_2 \ . \tag{3}$$

We consider the average appearance of detection F_{a_i} in each sub-tracklet as one of its features and the correlation on appearance between two cliques c_i, c_j, is defined as

$$\text{Co\_appearance}_{\{c_i,c_j\}} = \| \text{F}_{a_i} - \text{F}_{a_j} \|_2 \,. \tag{4}$$

Moreover, another significant feature we calculate from sub-tracklet Ti is the velocity v.We extract two position coordinates, P_s(start point) and P_e(end point), from the sub-tracklet at two vital time point Tp_{i-e} and Tp_{i-l}, then we compute velocity of Ti as follow:

$$\text{velocity}_{T_i} = \begin{cases} \dfrac{p_e - p_s}{\text{Tp}_{T_{i-l}} - Tp_{T_{i-e}}} & Tp_{T_{i-l}} \neq Tp_{T_{i-e}} \\ 0 & Tp_{T_{i-l}} = Tp_{T_{i-e}} \end{cases} , \tag{5}$$

where $Tp_{i-l} \neq Tp_{i-e}$ means sub-tracklet Ti contains more than one detection, otherwise its velocity is set as 0. We take the direction of sub-tracklet into account and the correlation of two sub-tracklets (T_i, T_j) on velocity can be measured as

$$\text{Co}\_velocity_{T_i,T_j} = \frac{|\,\text{velocity}_i - \text{velocity}_j\,|}{\beta + \cos(T_i, T_j)}, \tag{6}$$

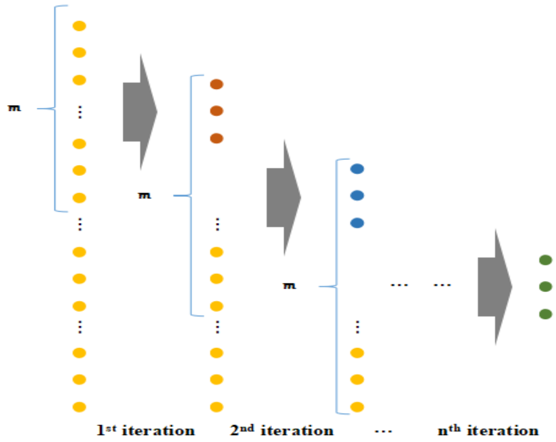

FIGURE II. EFFECTIVE BACKTRACKING

where $\cos(T_i, T_j)$ is the cosine angle between two direction vectors of two sub-tracklets. The constant β here ensure the result value is positive and avoid having 0 in the denominator.

With all these correlations, we define the overall correlation information among the sub-tracklets as

$$\text{S}_{\text{Ti,Tj}} = \text{Co\_velocity}_{\text{Ti,Tj}} + \text{Co\_appearance}_{\text{Ti,Tj}}. \tag{7}$$

For graph construction, we define edges of clique graph $E_c = (s_{ij})$. The value s_{ij} is defined as

$$s_{ij} = \begin{cases} 1 & \text{i} \neq \text{j and } \text{S}_{\text{c-c}}(v_i, v_j) > thre_2 \\ 0 & \text{i} = \text{j or } \text{S}_{\text{c-c}}(v_i, v_j) \leq thre_2 \end{cases}. \tag{8}$$

C Effective Backtracking

When the graph is built, the maximum clique problem is defined the same way as in [7]. Some researchers solved this problem using backtracking method explained in [8] and obtained acceptable results. However, although the pruning operation in Backtracking reduced the search space, it still takes exponential complexity. Further more, some tracklets with long time span might be unsolvable with this method. Therefore we design a effective backtracking method in our algorithm.

As is shown in Figure 2, for a tracklet T whose length exceeds a threshold m, we regard it as a nodes set V and search the maximum clique partly: At first, we pick up a set V_s containing m nodes from V as input and search a maximum clique in it. Subsequently we fuse the maximum clique into one node, and put it together with nodes left in V ,as well as nodes in V_s but fail to be selected to the maximum clique. Then we can repeat this process until all nodes have been gone through. In this way we divide long tracklet into short ones and search maximum clique step by step, which reduce the time complexity from $O(n{\bullet}2^n)$ to $O(n)$. Thus we can complete the algorithm in polynomial time.

III. TRACKLETS REFINEMENT ALGORITHM

Tracklets linking is a essential step in tracking algorithm to link short tracklet into long ones, making the trajectories more integrated. Nevertheless, since tracklets linking results usually rely on the quality of appearance and motion model of each tracklets, tracklets with IDs have inaccurate features and are prone to error in linking process, causing errors accumulation. We use the IDs awareness algorithm to alleviate the problem.

A Problem Formulation

Here we apply network flow method to link tracklets correspond to the same person, during which we regard our clustering results as an extra information and utilize them to improve IDs.

Firstly we create initial tracklets $T = \{T_1, T_2, \ldots, T_m\}$ using method described in [5], then we break tracklets that contain different persons into different clusters, and replace the tracklets set T by clusters set

$$\text{C} = \{\text{C}_1, \text{C}_2 \ldots \text{C}_n\}, \tag{9}$$

where C_i is the clustering results, and this replacement enhance reliability of primary tracklets. Then we incorporate flags $f_{ij} = \{0,1\}$ to representing the connectivity of tracklets T_i and T_j. And flags are subject to the constraints

$$\sum_j f_{ij} = 1 \qquad \sum_i f_{ij} = 1, \tag{10}$$

which enforce each tracklet must follow and be followed by at most one other tracklet.

Next in the process of linking tracklets, we use the linking cost $C_{t,a,v}$ described in [9] and incorporate the IAA (i.e.IDs Awareness Algorithm) cost between tracklets as

$$C_{IAA\{C_i,C_j\}} = \begin{cases} 0 & \text{if } \ IAA(C_i \cup C_j) = 1 \\ \lambda \bullet IAA(C_i \cup C_j) & \text{else} \end{cases}, \quad (11)$$

where $IAA(C_i \cup C_j)$ means the result of IAA(i.e. number of cliques) if we link C_i and C_j as the input tracklet. And we consider the result 1 as the symbol of {Ci,Cj} belong to same person. Otherwise we consider they are different person and their combination should be depressed. Eventually this network flow problem can be formulated as

$$T^* = arg\ \min \sum_{ij} f_{ij} (C_{t,a,v\{C_i,C_j\}} + C_{IAA\{C_i,C_j\}}). \quad (12)$$

Once the costs have been computed,the final tracking solution can be solved optimally by min-cost flow algorithm.

B Tracklets Correction and Linking

As shown in Alg.2, we apply tracklets correction and linking algorithm iteratively with the increment of time gap ΔT. The terminate condition is the convergence, meaning the linking between all the tracklet pairs belong to different person are depressed.

IV. EXPERIMENT

We evaluate our method on five public datasets, which are commonly used by previous MOT works. We compare our tracklets linking model with the same method without IAA, to show the single effect of IAA process.

Parameters: The thresholds thre1 and thre2 in (2) and (7) are set respectively to 0.6, 0.55. β in (6) is set to 1.25 and λ in (11) equals 0.2 in our experiment. The max time gap T_{max} and δ are respectively 30s and 5s in Algorithm2.

TABLE II. ALGORITHM2 TRACKLETS REFINEMENT

```
Input: $T = \{T_1, T_2 ... T_n\}$
Initial: $\Delta T = 0$, $T_{max}$
while true do
    if $\Delta T < T_{max}$
    $\Delta T \leftarrow \Delta T + \delta$
        $C_i \leftarrow IAA(T_i)$   $T_i$ □T
        $C \leftarrow C + C_i$
        calculate $C_{IAA\{Ci,Cj\}}$   $C_i$, $C_j$ □C
        $T' = \arg\min \sum f_{ij} \bullet Cost$
        if $T = C = T'$ do
        break
        else
       $T \leftarrow T'$
return Tracklet set T
```

Evaluation metrics: In our experiment,we use CLEAR MOT to measure tracking result [10].The multiple object tracking accuracy (MOTA), recall (Rcll) and the number of identity switching (IDs) are used. We also design an indicator, inner IDs, to highlight the validity of our method.

Figure 3 shows difference between iterative results of baseline method (the blue line) and that of our method(the red line) on PETS2009-S2L1 dataset. These two method converge in about 10 iterations. From these figures we find that the IAA can handle IDs problem effectively. Besides, the better performance on MOTA indicating that the improvement on IDs problem is beneficial to the whole tracking process.

As shown in Table 3, these four indicators get better when IAA is added to the baseline method. Specifically, figures for TUD-Crossing and TUD-Campus improves significantly (MOTA increase by 7.5% and 7.1% respectively), that is because in these datasets mismatched tracklets account for a considerable proportion of initial tracklets and are corrected in the following stage. In terms of recall, our approach has a noticeable advantage, proving the tracklets in our tracking results are more integrated owing to the decrease of errors. Furthermore, the slump on inner IDs illustrates that our method has almost achieved the expected goal.

V. CONCLUSIONS

In this paper, we designed an identity switches awareness algorithm which is capable of perceiving tracklets with errors and then incorporate identity switches information into network flow tracking framework. The algorithm is based on maximum clique searching and we build two types of graph during the iterative clustering process. In addition, to accelerate the maximum clique finding process, we present an optimal method to decrease the time cost. At last, we show that this method can be used to handle IDs problem by testing on five challenging sequences and achieve good performance.

ACKNOWLEDGMENT

This study is partially supported by the National Key R&D Program of China(No.2017YFC0806500), the National

TABLE III. TRACKING RESULT

Dataset	Algorithm	Rcll	IDs	MOTA	Inner IDs
PETS2009 -S2L1	Baseline	96.3	33	87.8	17
	Baseline + IAA	**97.0**	**18**	**90.0**	**2**
KITTI-17	Baseline	63.1	5	59.4	4
	Baseline + IAA	**66.8**	**2**	**63.7**	**1**
TUD-Crossing	Baseline	73.7	25	63.7	13
	Baseline + IAA	**80.2**	**12**	**70.8**	**0**
TUD-Campus	Baseline	51.0	8	47.9	5
	Baseline + IAA	**57.7**	**3**	**55.4**	**0**
TUD-Stadtmitte	Baseline	72.3	10	64.1	6
	Baseline + IAA	**75.4**	**5**	**67**.9	**1**

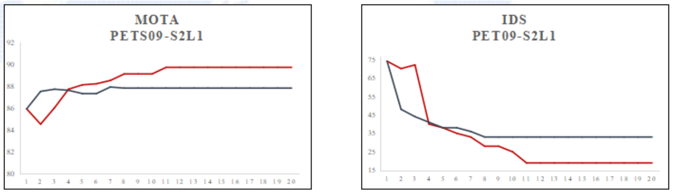

FIGURE III. ITERATIVE RESULTS ON PETS2009-S2L1

Natural Science Foundation of China(No.61370122), the Macao Science and Technology Development Fund (No.138/2016/A3), the Program of Introducing Talents of Discipline to Universities and the Open Fund of the State Key Laboratory of Software Development Environment under grant SKLSDE-2017ZX-09, the Project of Experimental Verification of the Basic Commonness and Key Technical Standards of the Industrial Internet network architecture. Thank you for the support from HAWKEYE Group.

REFERENCES

[1] Yilmaz A, Javed O, Shah M. Object tracking: A survey[J]. ACM Computing Surveys, 2006, 38(4).

[2] Jixu CHEN, Qiang JI. A Probabilistic Approach to Online Eye Gaze Tracking Without Explicit Personal Calibration[J]. IEEE Transactions on Image Processing, 2015, 24(3): 1076-1086.

[3] Andriluka M, Roth S, Schiele B, and Schiele B. People-tracking-by-detection and people-detection-by-tracking[C]. computer vision and pattern recognition, 2008: 1-8.

[4] Xiaojing CHEN, Zhen QIN, Le AN, and Bhanu B. An Online Learned Elementary Grouping Model for Multi-target Tracking[C]. computer vision and pattern recognition, 2014: 1242-1249.

[5] Mclaughlin N, Rincon J M, Miller P Cl. Enhancing Linear Programming with Motion Modeling for Multi-target Tracking[C]. workshop on applications of computer vision, 2015: 71-77.

[6] Tesfaye Y T, Zemene E, Pelillo M, and Prati A. Multi-object tracking using dominant sets[J]. Iet Computer Vision, 2016, 10(4): 289-298.

[7] Qinghua WU, Jinkao HAO. A review on algorithms for maximum clique problems[J]. European Journal of Operational Research, 2015, 242(3): 693-709.

[8] Tarjan R E. Depth-First Search and Linear Graph Algorithms[J]. SIAM Journal on Computing, 1972, 1(2): 146-160.

[9] Zhen QIN, Shelton C R. Improving multi-target tracking via social grouping[C]. computer vision and pattern recognition, 2012: 1972-1978.

[10] Lealtaixe L, Milan A, Reid I D, et al. MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking[J]. arXiv: Computer Vision and Pattern Recognition, 2015.