MULTI-OBJECT TRACKING BY VIRTUAL NODES ADDED MIN-COST NETWORK FLOW

Peixin Liu *Xiaofeng Li* *Haoyang Feng* *Zhizhong Fu*

School of Communication and Information Engineering
University of Electronic Science and Technology of China
Chengdu, China 611731

ABSTRACT

In this paper, we propose a Virtual Missing-detection and Occlusion Model (VMOM) for the cost flow network approaches in multiple object tracking (MOT) systems. The VMOM generates virtual nodes which represent occlusions and missing detections. Regular nodes which indicate tracklets together with virtual nodes are used to establish a cost flow network. A Counter Embedded Iterative Shortest Paths (CEISP) algorithm is also proposed to solve the new network flow optimization efficiently. Finally, the tracker with our scheme is tested on the open Multiple Object Tracking Benchmark. Experimental results show the advantages of our approach in main performance indicators comparing with many excellent trackers.

Index Terms— object tracking, tracklet, occlusion, virtual node, network flow

1. INTRODUCTION

With the progress of object detection techniques [1], [2], multiple object tracking (MOT) developed rapidly. Tracking by detection methods based on network flow attract much attention in the MOT area. Zhang et al. [3] described the MOT problem as a network flow model originally. Stauffer et al. [4] defined tracklets as short single-identity tracks and associated them by Hungarian algorithm. Butt et al. [5] linked detections into three-frames tracklets. Then the tracklets were connected into final trajectories by a min-cost network flow algorithm. They also incorporated track smoothness constraints in their model. Dehghan et al. [6] proposed a new Target Identity-aware Network Flow (TINF) model to learn the best locations of all targets. A node in the TINF encoded the probability of a target identity, and Lagrangian relaxation was used to obtain results efficiently. Chari et al. [7] embedded pairwise costs into the min-cost network flow and transformed the original object function to an NP-hard integer programming problem. They adopted a convex relaxation solution to reach a suboptimal output. Wang et al. [8] mapped tracklets into nodes and proposed an online target-specific metric learning method for long term tracking.

This work was supported by NSFC 61671126.

To handle difficult scenes, complicate models and specific constraints are often used. Occlusions of targets and missing of detections are two major problems in MOT tasks. To overcome them, we propose the VMOM . In our approach, detections are firstly linked into tracklets with various lengths. According to the VMOM, Occlusions and missing-detections are estimated. Then they are transformed into virtual nodes. Regular nodes of tracklets and virtual nodes from VMOM are used to construct a cost flow network. Finally, a Counter Embedded Iterative Shortest Paths (CEISP) algorithm is proposed to minimize the cost of the network and output final trajectories. The new approach is experimented on the open MOT datasets with iterative windows strategy. Its performance is compared with advanced methods and good improvement is achieved.

The main contributions of this paper include: 1) We propose a new VMOM to generate hypotheses of occlusions and missing detections, and add virtual nodes of them to the cost flow network. In this way, the final trajectories are improved by taking occlusions and missing detections into account effectively; 2) A modified Shortest Path algorithm which uses a simple counter to handle the differences caused by nodes of tracklets. The modified solution, called CEISP algorithm helps to solve the min-cost network flow problem efficiently.

2. APPROACH

First, we generate tracklets and utilize them to built a cost flow network. To deal with missing detections and occlusions effectively, we propose the VMOM, which adds virtual nodes into the network. In the followings, we present this framework in detail.

2.1. Tracklet Generation

Let $Z = \{z_i^f\}$ be a set of object detections, where $z_i^f = (p_i^f, a_i^f, s_i^f)$ denotes the ith detection, f is the frame index of it. p, a and s indicate the position, appearance and size of the detection respectively. Similar to that in [9], the tracklets are generated by direct-link method. Link probability between two detections is defined as $P_{link}(z_j^{f+1}|z_i^f) = P_d P_a P_s$, where $P(y|x)$ is the probability of y under the condition of

x. The three terms in the right-hand-side of $P_{link}(z_j^{f+1}|z_i^f)$ are probabilities of affinities in displacement, appearance and size respectively.

For pedestrian tracking applications, the appearance affinity is defined as $P_a(z_j^{f+1}|z_i^f) = (k)^{-1}BC(z_j^{f+1}, z_i^f)$, where BC denotes the Bhattacharyya coefficient [10], k is a normalization factor. A three-parts color histogram is used which considers the head, body and legs parts of a pedestrian separately. For the size and displacement affinities, Gaussian functions with trained variances are used, for it is reasonable to assume that their detections are corrupted by additive Gaussian noises. Detections are linked into tracklets when the link probabilities reach maximum and beyonds certain threshold. Finally, the unlinked detections are regarded as tracklets with single detection. In this way, all detections are linked into reliable tracklets.

2.2. State Judgement

Let $T = \{t_i\}$ be the set of all tracklets. To judge the state of a tracklet t_k, we propose a forward and backward searching method as:

$$state(t_k) = [u(\sum_{x=1}^{M-f_e}\sum_{j=1}^{N_{e+x}} u(A_{e,j}^{f_e,f_e+x} - \varphi)), \\ u(\sum_{x=1}^{f_s-1}\sum_{j=1}^{N_{s-x}} u(A_{s,j}^{f_s,f_s-x} - \varphi))] \quad (1)$$

where $u(x)$ is the step function, A is the affinity in appearance and size, f is the frame number. The subscripts s and e indicate start and end of tracklet t_k. $A_{i,j}^{s,e}$ denotes the affinity of t_i^s and t_j^e. M denotes the total number of detections, N_s, N_e indicate the number of detections in frame f_s and f_e respectively. The state is expressed in a pair of bits. [0,1] indicates t_k is a start tracklet, [1,0] a terminal one, [1,1] a intermediate one and [0,0] a complete one.

2.3. Virtual Missing-detection and Occlusion Model

To improve tracking performance, we propose the VMOM which generates hypotheses of occlusions and missing detections. Then these hypotheses are transformed into virtual nodes which are added into cost flow network.

2.3.1. *Virtual Node for Occlusion*

Occlusion is based on a position relationship between detections of tracklets. Pedestrian walking can be treated as a linear motion during a few frames. Local linear regression is used to estimate the moving trends of tracklets.

For each tracklet t_i with state [0,1] or [1,1], the future locations are estimated by $\hat{x} = \hat{a} + \hat{b}\hat{f}$, where $\hat{a}$ and $\hat{b}$ are calculated by local linear regression. $\hat{f}$ is the frame offset. When the length of t_i is longer than five, the last five detections are used in the estimation and three future locations are obtained. Otherwise, except single-detection tracklets, all detections are used and two future locations are obtained. Then, a tracklet t_j is found which is the nearest one to the last detection of t_i and shares common frames with it. t_j is estimated in the same way. If the minimum distance between any detection of t_j and a future location of t_i at the same frame is less than the threshold ϕ, an Occlusion hypothesis is generated and a corresponding virtual node is added.

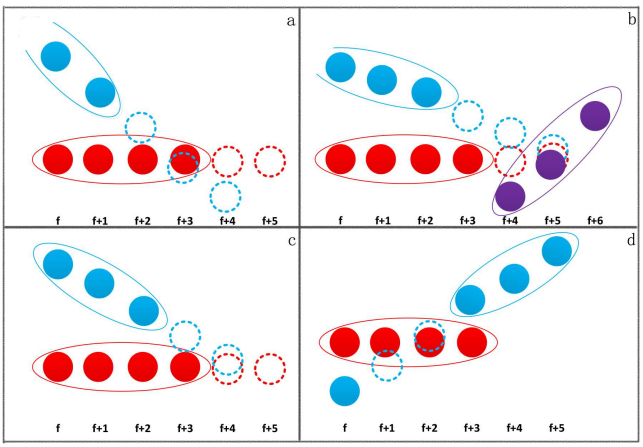

Fig. 1. Virtual nodes generating process

In Fig. 1(a), blue for t_i and red for t_j, the occlusion point is at frame offset 2. We add a virtual node v_i which shares the position and frame index with the detection in t_j, and takes size from the last detection of t_i. If the occlusion point is out of them and occluded by a third tracklet as shown in Fig. 1(b), two virtual nodes are added for t_i and t_j respectively.

2.3.2. *Virtual Node for Missing Detection*

Missing detections are estimated by occlusions dealing. As shown in Fig. 1(a), there exist an estimation at frame offset 2 between t_i and v_i, it indicates a missing detection. Monte Carlo method is used to recover the optimal position. As center of the estimation at offset 2, 10 samples which obey Gaussian distribution with zero mean and trained variance are generated. We get the size from last detection of t_i and extract appearance features of these 11 competitors. Winner is the final position. A virtual node for missing detection is generated and it only associates with t_i and v_i.

In Fig. 1(b), missing detection virtual nodes at 3 for t_i and 4 for t_j are generated. Fig. 1(c) shows a complex scene. We need to decide the situation of occlusion. Then missing virtual node is generated at frame offset 3 for t_i when t_i is occluded by t_j. Especially, as shown in Fig. 1(d), t_i is a single detection. We need to find a assistant tracklet t_k whose first detection is in the neighborhood of t_j and appearance is similar to t_i. Then the occlusion point is computed by them

and missing position is at frame offset 1. Finally, as virtual nodes are generated from regular nodes with states [0,1] and [1,1], the states of virtual nodes are set to [1,1].

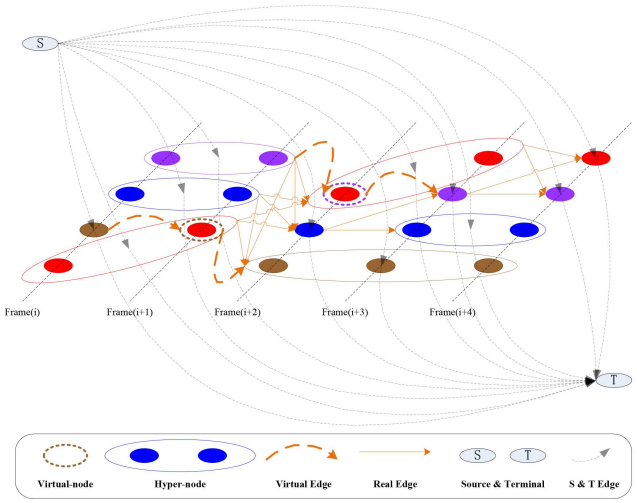

Fig. 2. Cost-flow network with virtual nodes

3. FORMULATION

In this section, virtual nodes generated by the VMOM and regualr nodes of tracklets are formulated into a min-cost network flows optimization problem as shown in Fig. 2. The notations used in our approach are shown in Table 1.

Notation	Description
-S, T	source and sink
-$R = \{r_1, r_2, ..., r_l\}$	regular nodes
-$V = \{v_1, v_2, ..., v_m\}$	virtual nodes
-$N = R \cup V = \{n_1, n_2, ..., n_{m+l}\}$	all nodes
-$E_{sr} = \{e_{sr_1}, e_{sr_2}, ...e_{sr_n}\}$	edges: S to r_i
-$E_{rt} = \{e_{r_1t}, e_{r_2t}, ...e_{r_nt}\}$	edges: r_j to T
-$E_{vt} = \{e_{v_1t}, e_{v_2t}, ...e_{v_mt}\}$	edges: v_j to T
-$E_{rr} = \{e_{r_ir_j}\}$	edges: r_i to r_j
-$E_{rv} = \{e_{r_iv_j}\}$	edges: r_i to v_i
-$E_{vr} = \{e_{v_jr_i}\}$	edges: v_i to r_j
-$E_{vv} = \{e_{v_iv_j}\}$	edges: v_i to v_j
-$p_{sr}, p_{rt}, p_{vt}, p_{rr}, p_{rv}, p_{vr}, p_{vv}$	probabilities of edges
-$L_k = \{S, n_{k_1}, ..., n_{k_q}, T\}$	kth path
-$L = \{L_k\}, k \in \{1, 2, ..., K\}$	set of all paths.

Table 1. Notations

Specifically, p_{sr}, p_{rt} and p_{vt} depend on states of regular nodes and virtual nodes. A parameter λ set to 0.75 is used for this purpose. For a regular node r_i, if it is a start node, $p_{sr} = \lambda$. Otherwise, $p_{sr} = 1 - \lambda$. Similar to p_{sr}, p_{rt} and p_{vt} can be achieved. p_{rr} is got by the similarity between two regular nodes. An e_{rv} is the regular-to-virtual edge and its probability p_{rv} is a trained value 0.75. Differ from p_{rv}, p_{vr} for the virtual-to-regular edge e_{vr} is achieved by the affinity between virtual node and regular node. A p_{vv} is a trained value set to 0.75. Finally, the objective function is given by:

$$\begin{aligned} \underset{L}{argmin} \sum_{L_i \in L} C(R, V, S, T) \\ s.t. \sum_{L_k \in L} \delta(n_i, L_k) \leqslant 1 \end{aligned} \tag{2}$$

where, $C(R, V, S, T) = \sum C(s, r) + \sum C(r, t) + \sum C(v, t) + \sum C(r, r) + \sum C(r, v) + \sum C(v, r) + \sum C(v, v)$. And $C(s, r) = -log(p_{sr})$, $C(r, t) = -log(p_{rt})$, $C(v, t) = -log(p_{vt})$, $C(r, r) = -log(p_{rr})$, $C(v, r) = -log(p_{vr})$, $C(r, v) = -log(p_{rv})$, $C(v, v) = -log(p_{vv})$ respectively. The condition forces that each node only belongs to one specific trajectory.

4. ALGORITHM

To solve the objective function (2), shortest paths is computed from S to T iteratively until there is no path between them, as the problem is a standard form of single source and sink network flows with non-negative costs. Capacity on each edge is 1. The Dijkstra's algorithm is applied to find a shortest path with computing complexity in $O(NlongN)$, making the overall complexity be $O(KNlongN)$. Nodes of tracklets which include several detections reduce the complexity and make Dijkstra's algorithm faster.

However, nodes with various starting frames make the objective function unable to be solved directly. We propose a new algorithm called CEISP algorithm. A counter is used to count the frame numbers between two nodes and the count is multiplied by the cost of each edge to eliminate the difference in gaps. The procedure of CEISP is shown in Algorithm 1.

Algorithm 1 CEISP

Input: video sequence and information of detections.
Output: Shortest paths $L = \{L_k\}, k \in \{1, 2, ..., K\}$.
Initialization:
- generate regular nodes set $R = \{r_1, r_2, ...r_l\}$
- generate virtual nodes set $V = \{v_1, v_2, ...v_m\}$
- judge states of R and V
- produce edges and compute their probabilities
- handle counters of edges
- build the graph $G(V, R, E, C)$
- initialize the residual $G_r = G(V, R, E, C)$

while there exists a path L_k from S to T in the residual network G_r
1. Find the minimum cost path L_k from S to T in G_r
2. Update the $G_r = G_r - L_k$

end

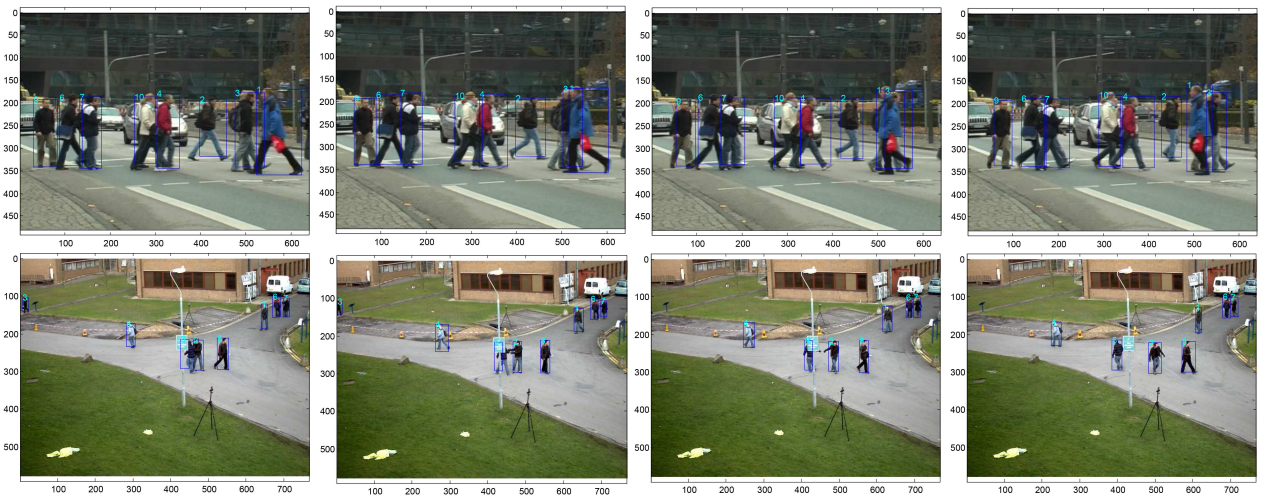

Fig. 3. Tracking with our method on sequences TUD-Crossing (first row, frame 42, 44, 46 and 48) and PETS-S2L1 (second row, frame 171, 173, 175 and 178).

5. EXPERIMENT

For fair comparisons, we register with the Multiple Object Tracking Benchmark [11] and evaluate our tracker on it. Performance of our method is compared with advanced trackers which are public on the Benchmark. Results of comparison are shown in Table 2 and 3. As we can see our tracker has the best MOTA performance. Our ID switches (IDs) are very larger, but the sum of the FN, FP and IDs of is the least.

Refer to our high IDs, there are two aspects to discuss. Firstly, quality of detections affects tracking performance largely. Detections given by [11] contain lots of false alarms and missing targets. In order to avoid preprocessing as much as possible, our tracker only eliminate the obvious bad detections which have too large or too small sizes. Secondly, the IDs of PET09-S2L2 and AVG-TownCentre sequences account for 48% of our total IDs in the 2D dataset of MOT 2015, and the IDs of MOT16-03 sequence accounts for 67% of our total in the dataset of MOT2016. These facts imply that our method is capable of working in most circumstances except for some specific ones. Our average tracker speeds are $43.9Hz$ and $37.4Hz$ respectively, except for MOT16-03.

With virtual nodes, when pedestrians pass by each other, frame and position of occlusion are often given precisely by our tracker. The complete trajectories of pedestrians can be generated quite well because the occlusion points are recovered. As shown in the first row of Fig. 3, the TUD-Crossing tracking results in frame 42, 44, 46 and 48, target 3 is occluded by 1. The tracker recovers the trajectory of target 3 correctly. If there exist barriers, targets will disappear several frames. For these situations, with the aid of virtual nodes, the tracker is able to recover trajectories too. For example, in the second row of Fig. 3, a street lamp at the center of scene covers pedestrian target 5. Our method tracks him accurately.

Methods	MOTA	MOTP	FP	FN	IDs
Ours	25.8	70.9	6316	37798	1493
JPM [12]	23.8	68.2	6373	40084	365
CEM [13]	19.3	70.7	14180	34591	813
SMOT [14]	18.2	71.2	8780	40310	1148
GSR [15]	15.8	69.4	7597	43633	514
DP[16]	14.5	70.8	13171	34814	4537

Table 2. Results for MOT2015

Methods	MOTA	MOTP	FP	FN	IDs
Ours	33.6	74.8	8743	109231	3149
CEM [13]	33.2	75.8	6837	114322	642
DP [16]	32.2	76.4	1123	121579	972
GPHD [17]	30.5	75.4	5169	120970	539
SMOT [14]	29.7	75.2	17426	107552	3108
JPM [12]	26.2	76.3	3689	130549	365

Table 3. Results for MOT2016

6. CONCLUSION

In this paper, we propose a VMOM to deal with occlusions and missing detections. Virtual nodes added by VMOM together with tracklets are used to build the cost flow network. A CEISP algorithm is proposed to handle the problem of differences in edges caused by nodes of tracklets. The tracker runs with iterative windows and experiments are carried out on the open Benchmark. Our scheme has advantages over many advanced trackers in main performance indicators. High ID switching in some occasions is under further study to improve the proposed scheme.

7. REFERENCES

[1] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, Sept. 2010.

[2] P. Dollr, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1532–1545, Aug. 2014.

[3] Li Zhang, Yuan Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2008, pp. 1–8.

[4] C. Stauffer, "Estimating tracking sources and sinks," in *Proc. Conf. Computer Vision and Pattern Recognition Workshop*, June 2003, vol. 4, p. 35.

[5] A. A. Butt and R. T. Collins, "Multiple target tracking using frame triplets," in *Asian Conference on Computer Vision*, 2012, pp. 163–176.

[6] A. Dehghan, Y. Tian, P. H. S. Torr, and M. Shah, "Target identity-aware network flow for online multiple target tracking," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 1146–1154.

[7] V. Chari, S. Lacoste-Julien, I. Laptev, and J. Sivic, "On pairwise costs for network flow multi-object tracking," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 5537–5545.

[8] B. Wang, G. Wang, K. L. Chan, and L. Wang, "Tracklet association with online target-specific metric learning," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2014, pp. 1234–1241.

[9] Chang Huang, Bo Wu, and Ramakant Nevatia, "Robust object tracking by hierarchical association of detection responses," in *Computer Vision - ECCV 2008, European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings*, 2008, pp. 788–801.

[10] Changjiang Yang, Ramani Duraiswami, and Larry Davis, "Efficient mean-shift tracking via a new similarity measure," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, pp. 176–183.

[11] Laura Leal-Taixé, Anton Milan, Ian Reid, Stefan Roth, and Konrad Schindler, "Motchallenge 2015: Towards a benchmark for multi-target tracking," *arXiv:1504.01942 [cs]*, vol. abs/1504.01942, 2015.

[12] S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid, "Joint probabilistic data association revisited," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, Dec. 2015, pp. 3047–3055.

[13] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 1, pp. 58–72, Jan. 2014.

[14] C. Dicle, O. I. Camps, and M. Sznaier, "The way they move: Tracking multiple targets with similar appearance," in *Proc. IEEE Int. Conf. Computer Vision*, Dec. 2013, pp. 2304–2311.

[15] L. Fagot-Bouquet, R. Audigier, Y. Dhome, and F. Lerasle, "Online multi-person tracking based on global sparse collaborative representations," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, Sept. 2015, pp. 2414–2418.

[16] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," in *Proc. CVPR 2011*, June 2011, pp. 1201–1208.

[17] Y. M. Song and M. Jeon, "Online multiple object tracking with the hierarchically adopted GM-PHD filter using motion and appearance," in *Proc. IEEE Int. Conf. Consumer Electronics-Asia (ICCE-Asia)*, Oct. 2016, pp. 1–4.