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 – UESTC

Introduction

With the progress of object detection techniques, Multiple Object Tracking (MOT) developed rapidly. Tracking by detection methods based on network flow attract much attention in the MOT area.

Let $Z=\{z_i^f\}$ be a set of object detections, where z_i^f denotes the ith detection, f is the frame index. And let $L=\{l_k\}$ be a set of the trajectories. Tracking by detection MOT problem can be formulated as equation (1).

$$L^* = \arg\max_L P(L\mid Z)eq.(1)$$

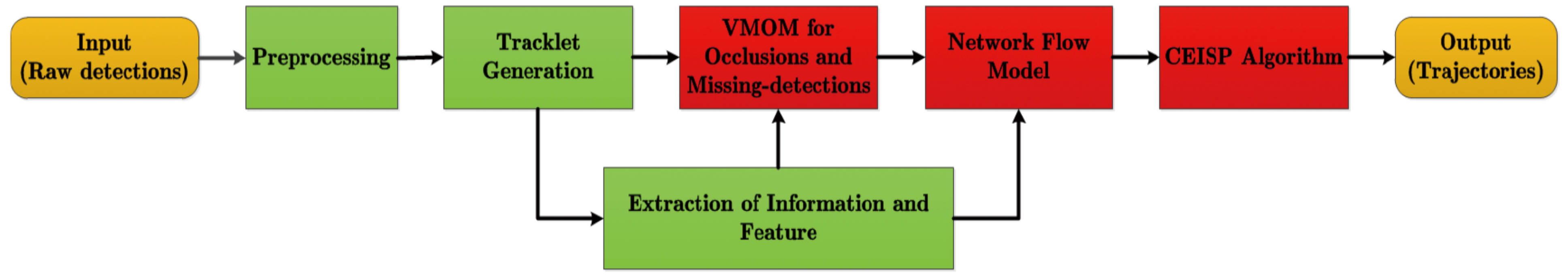

Fig. 1. Tracking system

Fig.1 presents our tracking system.

1. Raw detections as inputs are preprocessed firstly. In this process, abnormal bounding boxes will be deleted, which are too large or small and appear in unreasonable positions.
2. Preprocessed detections are linked into tracklets in the Tracklet Generation module.
3. We extract information and feature from the tracklets set.
4. Occlusions and missing-detections are analyzed in the Virtual Missing-detection and Occlusion Model (VMOM) which we proposed. Then the model is embedded in a network.
5. We develop a efficient Counter Embedded Iterative Shortest Paths (CEISP) algorithm which is suitable for our tracking model to get final trajectories.

Approach

Tracklet Generation

$Z=\{z_i^f\}$ is treated as preprocessed detections set. Tracklets are generated by direct-link method. Link probability between two detections is based on distance, size and appearance. Bhattacharyya coefficient and Gaussian function are used to get the similarity.

State Judgment

$T=\{t_k\}$ is the set of tracklets generated by last step. A forward-backward searching method (2) is proposed to judge state for each t_k. $u(x)$ is the step function, $A_{i,j}^{s,e}$ denotes the affinity between t_i^s and t_j^e. [0,1] indicates start, [1,0] a terminal one, [1,1] a intermediate one and [0,0] a complete one.

$$state(t_k)=[u((\sum_{x=1}^{f_s-1}\sum_{j=1}^{N_{s-x}}u(A_{s,j}^{f_s,f_s-x}-\varphi))-\theta),u((\sum_{x=1}^{M-f_e}\sum_{j=1}^{N_{e+x}}u(A_{e,j}^{f_e,f_e+x}-\varphi))-\theta)]eq.(2)$$

Virtual Node for Occlusion

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

Virtual Node for Missing Detection

Searching of missing detections is estimated by occlusions dealing. Monte Carlo method is used to estimate the optimal positions.

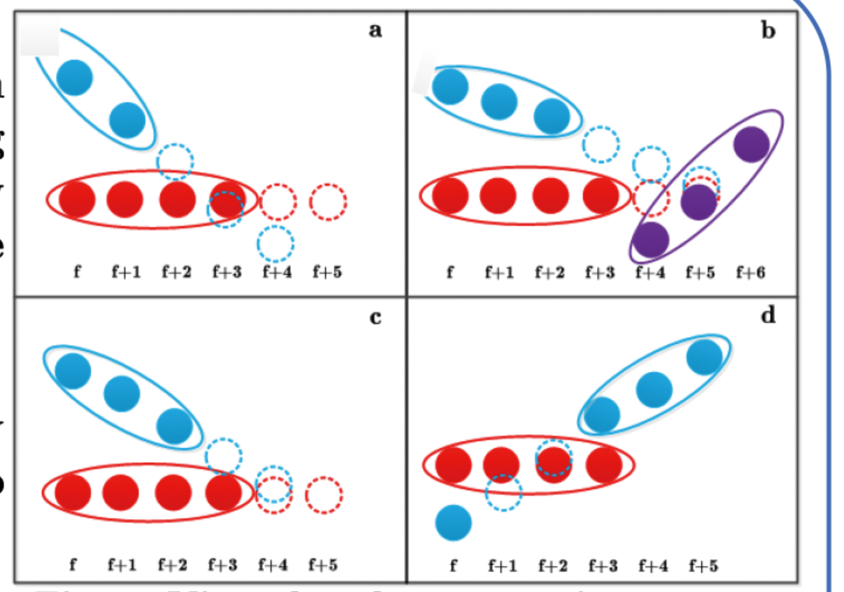

Fig. 2. Virtual nodes generating process

Formulation of Network Flow

$$L^* = \arg\min_L \sum_{L_i\in L} C(R,V,S,T) \quad s.t.\sum_{L_i\in L}\delta(n_i,L_k)\le 1 eq.(3)$$

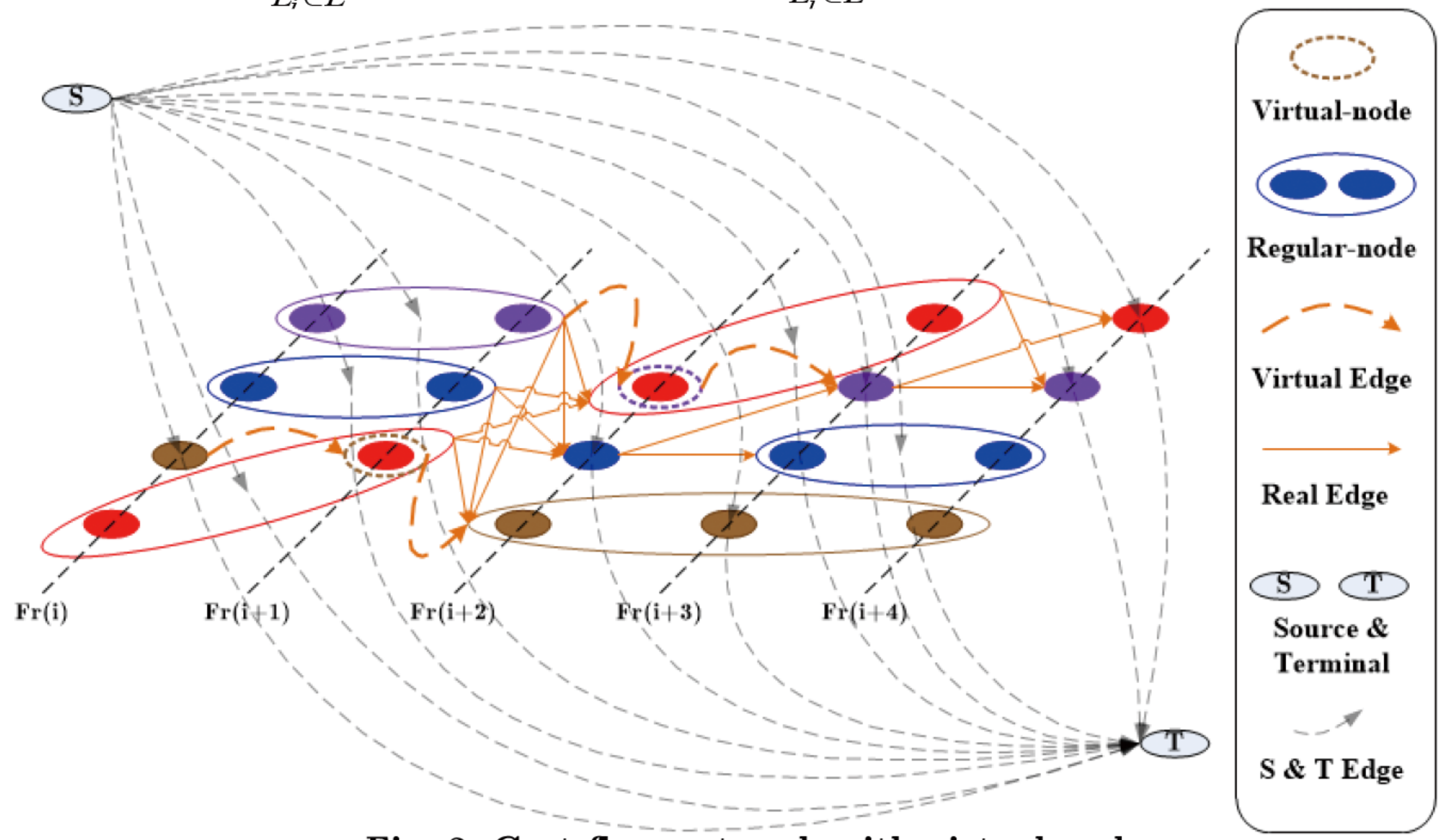

Fig. 3. Cost-flow network with virtual nodes

Algorithm CEISP

Input: Video sequence and information of detections.
Output: Shortest paths $L=\{L_k\}, k\in\{1,2,\ldots,K\}$.
Initialization:
- generate regular nodes set $R=\{r_1,r_2,\ldots,r_l\}$
- generate virtual nodes set $V=\{v_1,v_2,\ldots,v_m\}$
- 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 network $G_r=G(V,R,E,C)$

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

Experiments and Conclusion

We utilize the open Multiple Object Tracking Benchmark to evaluate our method. MOT-2015 and MOT-2016 are the test datasets. Table 1 presents tracking performance. Average tracking speeds are 43.9Hz and 37.4Hz respectively.

Seq	Rcll	Prcn	FAR	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP
TUD-Crossing	69.1	97.2	0.11	13	4	8	1	22	340	33	51	64.2	72.1
PETS09-S2L2	51.5	81.9	2.52	42	2	37	3	1100	4673	454	606	35.4	69.2
ETH-Jelmoli	53.9	75.2	1.02	45	7	23	15	450	1169	43	110	34.5	72.5
ETH-Linthescher	24.3	86.0	0.30	197	4	59	134	354	6764	171	217	18.4	73.0
ETH-Crossing	28.1	91.3	0.12	26	2	11	13	27	721	26	21	22.8	73.8
AVG-TownCentre	36.1	76.8	1.73	226	9	117	100	778	4566	270	356	21.5	69.2
ADL-Rundle-1	26.9	78.2	1.39	32	3	10	19	696	6804	122	126	18.1	73.7
ADL-Rundle-3	41.1	89.1	0.81	44	4	26	14	509	5988	115	135	35.0	73.1
KITTI-16	47.9	80.1	0.97	17	0	15	2	202	887	46	98	33.3	72.1
KITTI-19	42.4	62.7	1.27	62	4	44	14	1349	3076	117	374	15.0	65.7
Venice-1	38.4	67.9	1.84	17	1	10	6	829	2810	96	146	18.1	70.6
MOT16-01	22.6	91.5	0.30	23	1	5	14	134	4948	51	79	19.7	72.2
MOT16-03	47.1	90.5	3.43	148	11	100	37	5151	55307	2112	1601	40.2	74.8
MOT16-06	51.2	89.8	0.56	221	29	86	106	672	5630	212	258	43.5	72.5
MOT16-07	17.1	87.2	0.82	54	2	12	40	410	11218	236	215	13.7	75.2
MOT16-08	33.0	85.8	1.46	63	4	32	27	915	11218	136	215	26.1	78.5
MOT16-12	38.8	92.4	0.30	86	8	27	54	266	5075	126	107	34.1	76.6
MOT16-14	26.8	80.6	1.59	164	3	62	99	1195	13524	262	352	18.9	73.4

Table 1. Tracking performance

Conclusion: Local occlusions can be processed in our method as shown in Fig. 4. Virtual nodes recover the occlusions accurately. They also bring some mistakes. From these tracking results as shown in Table 1, our tracker can not track objects in these scenes very accurately. Tracklet with single detection makes some confusion during the occlusions processing period, which are presented by the indicators such as FP, FN and IDs. Our tracker needs further research to be improved.

Fig. 4. Samples of tracking results