LETTER

A Propagation Method for Multi Object Tracklet Repair

Nii L. SOWAH[†a)], **Qingbo WU**[†], **Fanman MENG**[†], **Liangzhi TANG**[†], **Yinan LIU**[†], *Nonmembers*, *and* **Linfeng XU**[†b)], *Member*

SUMMARY In this paper, we improve upon the accuracy of existing tracklet generation methods by repairing tracklets based on their quality evaluation and detection propagation. Starting from object detections, we generate tracklets using three existing methods. Then we perform co-tracklet quality evaluation to score each tracklet and filtered out good tracklet based on their scores. A detection propagation method is designed to transfer the detections in the good tracklets to the bad ones so as to repair bad tracklets. The tracklet quality evaluation in our method is implemented by intra-tracklet detection consistency and inter-tracklet detection completeness. Two propagation methods; global propagation and local propagation are defined to achieve more accurate tracklet propagation. We demonstrate the effectiveness of the proposed method on the MOT 15 dataset

key words: *tracklet, quality evaluation, propagation, repair, detection*

1. Introduction

Surveillance cameras have become very common devices seen daily, due to current security demands. This has increased the demand for highly reliable multiple object tracking (MOT) algorithms. The task of tracking multiple objects can be referred to as finding the locations of all objects in a video, together with their identities. To solve this task, many existing MOT algorithms follow the tracking-by-detection approach [1]–[5]. In this approach, object detections are first obtained frame-by-frame for the whole duration of the video, and the detections are associated in the temporal domain.

Tracking-by-detection methods are categorized into online and batch methods. Online methods [6], [7] can be applied to real-time applications, since they build trajectories sequentially based on the frame-by-frame association using the previous detections up to the present frame. However, such methods tend to produce fragmented trajectories and to drift under occlusion and detection errors, due to the difficulty in handling inaccurate detections. Batch methods [1], [5], [8], [9] on the other hand, build tracklets (short trajectories) from nearby detections and merge them using different optimization methods to generate the final trajectories. A few observations have been made in existing tracklet generation methods.

Firstly, it is hard to generate good tracklets for all detections in all segments of the video. Existing methods have incomplete tracklets or missing tracklets in some segments of videos. This is due to the fact that in some segments, the number of detections per frame is not consistent due to missed detections. Secondly, it is worth noting that there are good tracklets that are generated by the existing methods. Different algorithms have good tracklets in different segments of the video. Based on the above observations, we can see that good tracking results can be achieved if we repair the bad tracklets, whilst maintaining the good ones.

Inspired by the observations stated and the work of [10], we propose a new tracklet repair method to improve multi-object tracking. The proposed method consists of two steps: tracklet quality evaluation and tracklet propagation. The first term is to rank the tracklets from good to bad. We evaluate each tracklet based on its detection consistency and tracklet completeness, using our defined metrics. In the tracklet propagation stage, we transfer the good tracklets to the bad ones. Unlike the existing methods which design a new data association model, this paper proposed a new idea to generate good tracklets by repairing bad tracklets based

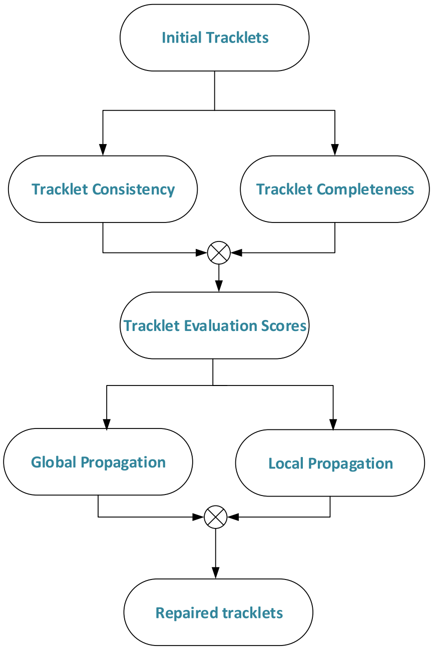

Fig. 1 Flowchart of the proposed method.

Manuscript received February 7, 2018.
Manuscript revised May 11, 2018.
Manuscript publicized May 29, 2018.

[†] The authors are with the University of Electronic Science and Technology of China, 611731, Chengdu, China.
a) E-mail: longdon001@yahoo.co.uk (Corresponding author)
b) E-mail: lfxu@uestc.edu.cn (Corresponding author)

DOI: 10.1587/transinf.2018EDL8029

on quality evaluation and tracklet propagation.

Our contributions include: (1) A tracklet quality evaluation method is proposed to score tracklets. We propose a new tracklet consistency and tracklet completeness evaluation method to distinguish bad tracklets from the good tracklets. (2) We propose a tracklet and detection propagation method to improve the bad tracklets. The global tracklet propagation and local detection propagation methods are proposed. The flowchart of the proposed method is illustrated in Fig. 1. In the first step, initial tracklets are generated from existing methods, and they are scored based on the consistency and completeness criteria. In the final stage we utilize good tracklets to repair the bad ones, using the global and local propagation schemes proposed in this paper

2. Tracklet Quality Evaluation

In this section, we evaluate the quality of a given tracklet in a segment of the video. We divide a video sequence into N segments of F frames each. A tracklet is said to be good if it contains the accurate number of detections and it also contains detections of only one object. Good tracklets will have lower scores, while bad tracklets will have higher scores. We evaluate each tracklet in each segment by two criteria. The first is to evaluate the detection consistency within each tracklet across frame in the segment. The second evaluates the completeness property of the tracklet.

2.1 Tracklet Consistency

Since each tracklet contains all or some detections of the same object, it is reasonable to assume that a high similarity can be observed among these detections. We evaluate the detection consistency based on the following cues.

Feature Similarity Cue. We introduce this term to evaluate how similar the features of the detections in a tracklet are. Detections belonging to the same object will have similar features. Given a tracklet t, we denote the detections in this tracklet as $D = \{D_i, ..., D_z\}$, where z is the number of detections in tracklet t. The appearance features of D_i is given by $H = \{H_i, ..., H_z\}$ where H_i is the feature of D_i. We obtain the similarity matrix M by calculating the distance of each feature pair (H_i, H_j).

$$M(i, j) = \chi^2(H_i, H_j) \tag{1}$$

$$= \sum_v \frac{(H_i(v) - H_j(v))^2}{H_i(v) + H_j(v)} \tag{2}$$

where χ^2 is the Chi square distance between the two features. For the i-th detection, we can find it's feature difference with the other detections by

$$u^1_{con}(i) = \sum_j M(i, j), i = 1, ...z. \tag{3}$$

A good detection has a smaller value of u^1_{con}, while a bad detection has larger value.

Gradient Difference Cue. We find that detections of the same object have a small magnitude of image gradient within a tracklet. Hence, we utilize this property to find the gradient difference between two detections given by

$$G_{D_i} = \frac{1}{N} \sum_{x,y \in D_i} \sqrt{(G_x^2 + G_y^2)} \tag{4}$$

$$u^2_{con}(i) = \eta|G_{D_i} + G_{D_j}| - |G_{D_i} - G_{D_j}| \tag{5}$$

where G_{D_i} is the average gradient of detection D_i in the horizontal and vertical directions, G_x and G_y. $u^2_{con}(i)$ is the gradient difference between D_i and D_j. η is a constant which was given a value of 0.5 in all experiments.

2.2 Tracklet Completeness

We investigate two cues to determine tracklet completeness.

Bounding Box Overlap Cue. Based on the assumption of uniform motion, we evaluate the bounding box overlap between neighboring detections in a tracklet. If the neighboring detections in a tracklet contain the bounding boxes of the same object, there should be some overlap between the detections. We define the bounding box overlap cue as

$$u^1_{com} = Ov(A_{D_i}, A_{D_j}) \tag{6}$$

$$Ov(A_{D_i}, A_{D_j}) = \frac{A_{D_i} \cap A_{D_j}}{A_{D_i} \cup A_{D_j}} \tag{7}$$

where $Ov(A_{D_i}, A_{D_j})$ is the bounding box overlap of A_{D_i}, A_{D_j}, the areas of the bounding boxes of detections D_i and D_j.

Tracklet Length Cue. Due to different tracklet generation methods, tracklet sizes for a specific object within a segment vary with different algorithms. Also, due to objects entering and leaving the field of view, tracklet sizes keep changing. We express this cue as

$$u^2_{com} = \alpha * (F - L_{t_i}) \tag{8}$$

where F is the number of frames in each tracklet(segment), L_{t_i} is the number of detections in the tracklet and $\alpha = 3$ is a constant. The smaller the tracklet length cue, u^2_{com}, the more complete the tracklet length is.

2.3 Total Tracklet Cost

We define a total four cues to evaluate tracklets in our method. For a good tracklet a small score will be obtained for all the cues. We evaluate each tracklet in a segment by a combination of these cues to determine the total score of the tracklet by

$$u = \frac{u_{com}}{u_{con}} * \lambda \tag{9}$$

where $\lambda = 0.1$ is a scaling factor, and

$$u_{con} = u^1_{con} + u^2_{con} \tag{10}$$

$$u_{com} = u^1_{com} + u^2_{com} \quad (11)$$

where u_{con} and u_{com} are the score of consistency and completeness respectively.

3. Tracklet Propagation

After evaluating the tracklets in a segment of the video, we can obtain the best tracklet among the three tracklet generation methods. The next step is to transfer the detections from the best tracklet to the other tracklets. We perform global propagation to transfer entire tracklets to a segment, and also perform local propagation to transfer lost or good detections to a bad tracklet.

3.1 Global Propagation

The global propagation is done at the segment level. Given the median number of detections per frame in a segment of the video, we determine the number of tracklets expected in that segment, regardless of the tracklet generation method. We intend to propagate global information by considering each segment of the video. The presence of false positive and false negative detections result in an unequal number of detections per group of neighboring frames. Given a segment N of the video with F frames, we first generate tracklets using methods [1], [4], [5]. Given the tracklets from the three algorithms, t_{A_i}, t_{B_i} and t_{C_i}, we seek to recover lost tracklets which have the similar identity in the compared methods. Since the tracklets are not labeled, we can only find the similar tracklets by the similarity distance between tracklets in each segment. We cluster all the tracklets in each segment using k-means clustering, given as

$$J(t_{ABC_i}) = \arg\max_{S} \sum_{i=1}^{k} \sum_{t \in S_i} ||t - \mu_i||^2 \quad (12)$$

where $J(t_{ABC_i})$ is the cluster of all tracklets in segment N from the three algorithms, k is the number of clusters and S_i are the clusters. From Eq. (12) we choose the centroid of each cluster as the unique tracklet for the segment. Next we cluster each tracklet algorithm with the unique segment tracklets to obtain the final global propagation.

3.2 Local Propagation

The local propagation is done at the tracklet level. Given three tracklets that have the same identity t_{A_i}, t_{B_i} and t_{C_i}, our goal is to find the good tracklet and transfer detections from it to correct the bad tracklets. We use the tracklet score in Sect. 2.3 to find the good tracklet, given by

$$u^G = min(u_{A_i}, u_{B_i}, u_{C_i}) \quad (13)$$

where u_{A_i}, u_{B_i} and u_{C_i} are the costs of the tracklets, and u^G is the cost of the good tracklet. We define a threshold, K_t, as the minimum difference between the cost of the good tracklet, t^G and a bad tracklet, t^B. If the difference is less than or equal to the threshold, we assume the tracklets are both good and we don't propagate detections. However, if the difference is greater than the threshold, we proceed with detection propagation from the good tracklet to the bad tracklet. We define the start and end detections in the good and bad tracklet as $t^G_{D_s}, t^G_{D_e}, t^B_{D_s}, t^B_{D_e}$ respectively. We obtain the length-repaired tracklet t^{rep}_L, given by

$$t^{rep}_L = t^B + (t^G_{D_s} - t^B_{D_s}) + (t^G_{D_e} - t^B_{D_e}) \quad (14)$$

and the overlap-repaired tracklet

$$t^{rep}_{Ov} = \begin{cases} t^G_{D_i} & \text{if } Ov(t^G_{D_i}, t^B_{D_i}) < 0.6, \\ 0 & \text{otherwise.} \end{cases} \quad (15)$$

where $Ov(t^G_{D_i}, t^B_{D_i})$ is the bounding box overlap between detection $t^G_{D_i}$ in the good tracklet and its corresponding detection $t^B_{D_i}$ in the bad tracklet. we obtain the final repaired tracklet t_r by

$$t_r = t^{rep}_L + t^{rep}_{Ov} \quad (16)$$

4. Experiments

To validate the proposed algorithm, experiments were performed on 3 sequences from the MOT15 dataset, PETS-S2L1, ADL-Rundle-8 and Venice-2. These sequences have varying degrees of occlusion and video complexity. Each video sequence was divided into segments of 10 frames each as the length of a tracklet. The ADL-Rundle-8 sequence is shot from a moving camera and whilst the other sequences are shot from a still camera. Detection confidence was not used in tracklet generation. We propose two metrics to verify our repaired tracklets. We propose mean tracklet loss per segment(TLPS) and tracklet overlap precision(TOP). The tracklet loss per segment is given as

$$TLPS = \frac{1}{Tseg} \sum_{j=1}^{Tseg} (N_{gt} - N_z), \quad (17)$$

where $Tseg$ is the total number of segments in the sequence, N_{gt} is the number of groundtruth tracklets per segment and N_z is the number of repaired or initial generated tracklets per segment. The tracklet overlap precision is given by

$$TOP = \frac{1}{Tseg} \sum_{j=1}^{Tseg} (Co_{gt} - Co_z), \quad (18)$$

where Co_{gt} is the groundtruth coordinates per segment and Co_z is the initial or repaired tracklet coordinates. A lower value of $TLPS$ is desired, whilst a higher value of TOP is desired. In Table 1 and Table 2, our repaired tracklets outperform the initial tracklets of the other methods, however the $TLPS_{Ini}$ of ACCV [4] and the TOP_{Ini} of GMCP [5] are higher. As can be seen in Table 3, our method outperforms all the the other methods. Constant parameters were obtained by experimental iteration, and the same values were

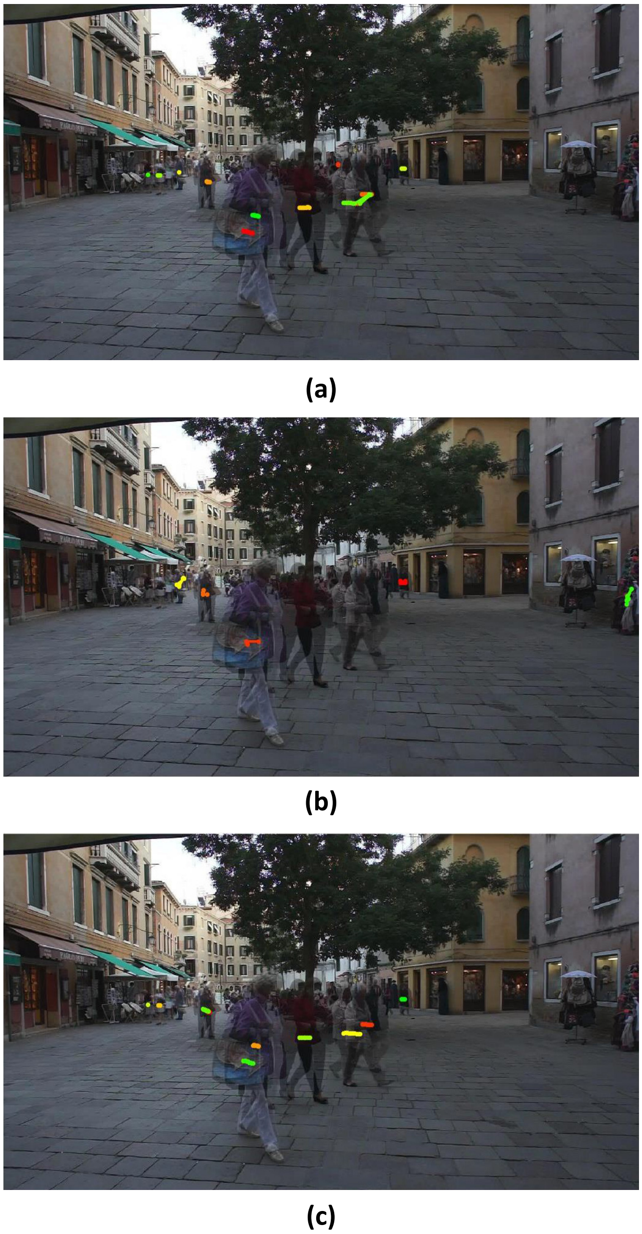

Fig. 2 The results of the proposed tracklet repair on a segment from the Venice-2 sequence. (a) Groundtruth tracklets. (b) Initial tracklets from GMCP. (c) Repaired tracklets. Colors on images represent tracklets.

Table 1 Quantitative results on Venice-2

Method	$TLPS_{Ini}$	$TLPS_{Rep}$	TOP_{Ini}	TOP_{Rep}
GMMCP [1]	4.285	**3.321**	0.329	**0.362**
ACCV [4]	**0.685**	0.708	0.114	**0.125**
GMCP [5]	0.0108	**0.0006**	**0.992**	0.123

used for all the experiments reported in this paper, as shown in Tables 1, 2 and 3. Figure 1 shows the tracklet results on a segment of the Venice-2 sequence. It can be seen that our proposed method is able to recover lost tracklets.

Table 2 Quantitative results on ADL-Rundle-8

Method	$TLPS_{Ini}$	$TLPS_{Rep}$	TOP_{Ini}	TOP_{Rep}
GMMCP [1]	0.897	**0.432**	0.033	**0.059**
ACCV [4]	**0.559**	0.813	0.101	**0.117**
GMCP [5]	0.017	**0.008**	**0.843**	0.536

Table 3 Quantitative results on TUD-Stadmitte

Method	$TLPS_{Ini}$	$TLPS_{Rep}$	TOP_{Ini}	TOP_{Rep}
GMMCP [1]	0.766	**0.575**	0.183	**0.218**
ACCV [4]	0.797	**0.434**	0.270	**0.319**
GMCP [5]	1.000	**0.529**	10.809	**15.568**

5. Conclusion

This paper presents a multi object tracklet repair method. We propose a tracklet evaluation and propagation method to improve initial tracklets of state-of-the-art methods. We propose metrics for measuring tracklet loss and precision. Based on the results, we hope to utilize the repaired tracklets to perform data association in future works.

References

[1] A. Dehghan, S.M. Assari, and M. Shah, “Gmmcp tracker: Globally optimal generalized maximum multi-clique problem for multiple object tracking,” Proc. CVPR, pp.4091–4099, 2015.

[2] S. Schulter, P. Vernaza, W. Choi, and M. Chandraker, “Deep network flow for multi-object tracking,” arXiv preprint arXiv:1706.08482, 2017.

[3] T. Kutschbach, E. Bochinski, V. Eiselein, and T. Sikora, “Sequential sensor fusion combining probability hypothesis density and kernelized correlation filters for multi-object tracking in video data,” 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp.1–5, IEEE, 2017.

[4] E. Ristani and C. Tomasi, “Tracking multiple people online and in real time,” Asian Conference on Computer Vision, pp.444–459, Springer, 2014.

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

[6] A. Milan, S.H. Rezatofighi, A.R. Dick, I.D. Reid, and K. Schindler, “Online multi-target tracking using recurrent neural networks,” AAAI, pp.4225–4232, 2017.

[7] J.H. Yoon, C.R. Lee, M.H. Yang, and K.J. Yoon, “Online multi-object tracking via structural constraint event aggregation,” Proc. IEEE Conference on computer vision and pattern recognition, pp.1392–1400, 2016.

[8] R. Yu, I. Cheng, B. Zhu, S. Bedmutha, and A. Basu, “Adaptive resolution optimization and tracklet reliability assessment for efficient multi-object tracking,” IEEE Trans. Circuits Syst. Video Technol., vol.28, no.7, 2017.

[9] N.L. Sowah, Q. Wu, F. Meng, W. Bo, and K.N. Ngan, “Strongly connected component multi-object tracking,” 2016 2nd IEEE International Conference on Computer and Communications (ICCC), pp.396–400, IEEE, 2016.

[10] H. Li, F. Meng, B. Luo, and S. Zhu, “Repairing bad co-segmentation using its quality evaluation and segment propagation,” IEEE Trans. Image Process., vol.23, no.8, pp.3545–3559, Aug. 2014.