Multi-Object Tracking of Pedestrian Driven by Context

Thi Lan Anh Nguyen, Francois Bremond, Jana Trojanova

Multi-Object Tracking of Pedestrian Driven by Context

Nguyen Thi Lan Anh Francois Bremond
INRIA Sophia Antipolis
2004 Route des Lucioles -BP93 Sophia Antipolis Cedex, 06902 - France
{thi-lan-anh.nguyen|francois.bremond}@inria.fr

Jana Trojanova
Honeywell Prague Lab, V Parku 2326/18, 148 00 Praha 4, Czech Republic
jana.trojanova@honeywell.com

Abstract

The characteristics like density of objects, their contrast with respect to surrounding background, their occlusion level and many more describe the context of the scene. The variation of the context represents ambiguous task to be solved by tracker. In this paper we present a new long term tracking framework boosted by context around each tracklet. The framework works by first learning the database of optimal tracker parameters for various context offline. During the testing, the context surrounding each tracklet is extracted and match against database to select best tracker parameters. The tracker parameters are tuned for each tracklet in the scene to highlight its discrimination with respect to surrounding context rather than tuning the parameters for whole scene. The proposed framework is trained on 9 public video sequences and tested on 3 unseen sets. It outperforms the state-of-art pedestrian trackers in scenarios of motion changes, appearance changes and occlusion of objects.

1. Introduction

Multi-object tracking (MOT) is essential to many applications in computer vision. As so many trackers have been proposed in past one would expect the tracking task as solved. It is true for scenarios containing solid background with low number of objects and few interactions. However scenarios with appearance changes due to pose variation, abrupt motion changes and occlusion still represent a big challenge.

The object occlusion is one of the most difficult aspects of pedestrians tracking. Several MOT frameworks have been proposed in the past to solve this issue. These frameworks consist of two parts the object detection followed by the data-association based tracking. The data association is found across the batch of frames which enable the ability to better deal with noisy object detection such as missed/false detection and occlusion. Depending on the length of the batch there are two types of association local and global.

Very popular method for local data association is the bipartite matching. The exact solution can be found via Hungarian algorithm [11, 2]. These methods are computationally inexpensive, but can deal only with short term occlusion. An example of global method is extension of the bipartite matching into network flow [16, 3]. Given the objects detections at each frame, the direct acyclic graph is formed and the solution is found through minimum-cost flow algorithm. The algorithms reduce trajectory fragments and improves trajectory consistency but lack of robustness to identity switches of close or intersecting trajectories. To overcome the ID switches, the paper in [15] proposes global data association using a model which is close to the real world tracking scenario. It incorporates both motion and appearance features into generalized minimum clique graph. They form a k-partite graph, where all the pairwise relationships between detections in the video is considered. The track of a person forms a clique and MOT is formulated as constraint maximum weight clique problem. In order to globally optimize the tracks, entire sequence must be provided beforehand. The weights to balance motion and appearance feature are set manually. The algorithm overcome the identity switch for intersecting trajectories however if the appearances of pedestrians walking in same direction are similar, the ID-switch remains.

Another set of methods for MOT is online parameter adaptation [14, 4]. They tune the tracking parameters based on the context information. While methods mention above uses one appearance and/or one motion feature for the whole video. The online methods typically use set of features. These features are weighted for new frame based on context information accumulated up to the present mo-

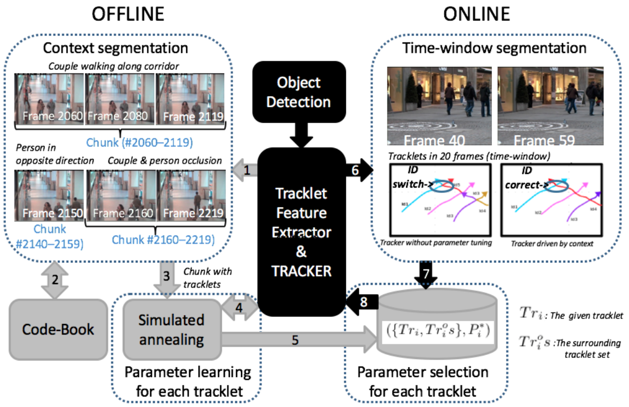
OFFLINE
ONLINE
Context segmentation
Couple walking along corridor
Frame 2060
Frame 2080
Frame 2119
Chunk (#2060–2119)
Person in opposite direction
Couple & person occlusion
Frame 2150
Frame 2160
Frame 2219
Chunk #2140–2159)
Chunk #2160–2219)
Object Detection
Tracklet Feature Extractor & TRACKER
Time-window segmentation
Frame 40
Frame 59
Tracklets in 20 frames (time-window)
ID switch->
ID correct->
Tracker without parameter tuning
Tracker driven by context
1
2
3
4
5
6
7
8
Chunk with tracklets
Code-Book
Simulated annealing
Parameter learning for each tracklet
$(\{Tr_i, Tr_i^o s\}, P_i^*)$
Parameter selection for each tracklet
Tr_i: The given tracklet
$Tr_i^o s$:The surrounding tracklet set

Figure 1. Our proposed framework.

ment. The [14] runs multiple trackers at time. The tracker interaction is conducted based on the transition probability matrix (TPM). The TPM updates are computed by investigating each tracker's reliability. The single tracker is responsible for one feature. Therefore, this method concentrates rather on trackers integration than feature combination. It has strong limitations on self-adaptability to scene variations characterized by more than one feature (appearance versus motion). Also ruining multiple trackers introduce high computational load and restrict the usage of the method in real time. The [4] learns the parameters for the scene context offline. In online phase the tracking parameters are selected from database based on the current context of the scene. These parameters are applied to all objects in the scene. Such a concept assumes discriminative appearance and trajectories among individuals, which is not always the case in real scenarios.

The limitation of the data association methods is identity switching when occlusion appears. The online parameter adaptation methods try to overcome the issue by tuning the parameters for the video context. However they ignores the individuality of the objects and use same set of features to all objects. In this paper, we try to overcome these limitations by proposing a new long term tracking framework. It combines short data association and online parameter tuning for individual tracklets. This framework has several dominant contributions as follow:

- We introduce new long term tracking framework which combines short data association and the online parameter tuning for individual tracklets. In contrast to previous method that used same setting for all tracklets (section 3).

- We show that large number of parameters can be efficiently tuned via multiple simulated annealing. Whereas previous method could tune only the limited number of parameters and fix the rest to be able to do exhaustive search (section 3.2).

- We define the surrounding context around each tracklet (section 2.2) and similarity metric among tracklets allowing us to match learned context with unseen video set(section 3.3).

2. The Proposed Framework

Figure 1 illustrates proposed framework. It highlights the all steps done in the offline and online phases. The objective of the offline phase is to learn a database of tracklets with optimal tracker parameters for various tracklet surrounding contexts. The database is then used in online phase for parameter selection for each tracklet based on its surrounding context in the scene.

2.1. Definitions

Video segment context characterizes the relationship of the object in the scene. We follow the definition of the video segment context from [6]. The video segment context is viewed as set of six code-books described for six video context features: density of mobile objects, their occlusion level, their contrast with regard to the surrounding background, their contrast variance, their 2D area and their 2D area variance.

Tracklet Tr_i is defined as a chain of tracked objects called node N_i in consecutive frames $< m, n >$ where i represents the ID of object and N represents the object bounding-box.

$$Tr_i = \{N_i^m, N_i^{m+1}, ..., N_i^{n-1}, N_i^n\} \tag{1}$$

Detection anomaly correction

Detection anomaly is defined as a node belonging to tracklet whose features are not consistent compared to other nodes(For example, the distance of 2 bounding-boxes in 2 consecutive frames is larger than threshold or object color change remarkably in 2 consecutive frames). To make tracklet more reliable, this node should be removed and we use the linear interpolation method to fill it.

Tracklet feature and tracklet feature similarities

Tracklet features F_i are extracted from features of nodes N_i belonging to tracklet. The feature pool F_i of each tracklet Tr_i is divided into 2 feature pools $F_i = \{F_i^O, F_i^{OE}\}$:

- F_i^O (individual features) represents the pool of features that are computed using only the data of the tracklet. F_i^O includes 6 features including 2D Shape ratio, 2D Area, Color histogram, Dominant color, Color Covariance and motion model. These features and their similarity computation are defined in [11].

- F_i^{OE} (surrounding features) represents the pool of features that are computed based on the interaction of tracklet to surrounding environment defined by outer bounding box (OBB). The size of OBB is determined by 1.3 times of object bounding-box size on width and height dimensions. Any tracklet appears in the OBB is considered to interact with tracklet Tr_i and these features are computed. Three features are defined in the OBB:

- Occlusion: The occluded level of given tracklet Tr_i caused by other surrounding tracklets. The value is in the range $< 0, 1 >$, 0 is non-occluded and 1 is full-occluded.

- Mobile object density: The number of other tracklets inside the outer bounding-box of given tracklet Tr_i.

- Contrast: is defined as the color histogram difference between the bounding-box and the outer bounding-box.

Tracklet feature $F_i^k \in F_i$ is represented by the weighted mean $\mu(F_i^k)$ and the weighted standard deviation $\sigma(F_i^k)$ over time t which are computed as follow:

$$\mu(F_i^k) = \frac{\sum_{t=m}^{n} w(t) * F_i^k(t)}{\sum_{t=m}^{n} w(t)} \tag{2}$$

$$\sigma(F_i^k) = \sqrt{\frac{\sum_{t=m}^{n} w(t) * (F_i^k(t) - \mu(F_i^k))^2}{\sum_{t=m}^{n} w(t)}} \tag{3}$$

where w is the weight function which is used to decrease the impact of the interpolated features while relying on the directly extracted features from object detection. The weight function is defined by:

$$w(t) = \begin{cases} w_I & \text{if } F_i^k(t) \text{ is interpolated} \\ w_E & \text{if } F_i^k(t) \text{ is directly extracted} \end{cases}$$

w_E and w_I satisfy:

$$\begin{cases} Nb_E * w_E + Nb_I * w_I = 1 \\ w_I = \alpha * w_E \end{cases}$$

f_p^t stands for object feature p at time t, α is a coefficient which determines the reliability of interpolated features compared to directly extracted features. Supposed that Nb_I and Nb_E are numbers of interpolated nodes and real tracked nodes, correspondingly, α is determined by $\alpha = \frac{Nb_I}{Nb_I + Nb_E}$.

2.2. Tracklet representation

Surrounding tracklet set $Tr_i^c s$ is figured out as a set of surrounding tracklets Tr_i^c which are inside the outer bounding-box of tracklet Tr_i.

As shown in Figure 2, the tracklet represented by red bounding-box in (a) is discriminated to its surrounding tracklet by color feature while the tracklet represented by red bounding-box in (b) is discriminated to its surrounding tracklet by movement. Then, they should be in different contexts. In order to characterize a tracklet in particular context, we define the representation of tracklet Tr_i is the combination of the features' information of Tr_i and $Tr_i^c s$. Based on the tracklet representation, the discriminate tracklet features are selected to characterize tracklet.

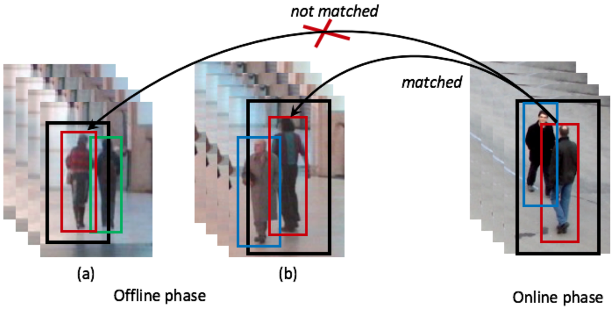

Figure 2. Tracklet representation $\{Tr_i, Tr_i^c s\}$ and tracklet representation matching. Tracklet Tr_i is presented by color "red" and fully covered by the outer bounding-box "black". The other colors are presented for surrounding tracklets.

The representation of tracklet is deployed in detail as follow:

$$\{Tr_i, Tr_i^c s\} = \{F_i, [F_i^c]_M\} \tag{4}$$

where M is the size of surrounding tracklet set $Tr_i^c s$ and F_i^c is feature pool of each surrounding tracklet $Tr_i^c \in Tr_i^c s$.

2.3. The framework flows

The framework flows are shown in Figure 1.

In **offline phase**, firstly, video is segmented by context. In particular, the video is split into segments of fixed size. Each segment is processed with object detection algorithm [10] and tracker [5](*flow_1*). All tracklets extracted for each video segment are assigned with context via codebook model(*flow_2*). If two or more consecutive segments have same context, they are merged to form a video chunk. Next step is the optimal parameter learning. The video chunks (video segments with same context) and its tracklets are passed to simulated annealing (*flow_3*). In this step, the representation of each tracklet is determined. The optimal parameters P_i^* for each tracklet representation are learned (*flow_5*) based on the performance evaluation of tracker (*flow_4*) against the ground truth information. Finally, tracket representation combined with its optimal parameter set is stored in database. The learned data is formalized as follow: $(\{Tr_i, Tr_i^c s\}, P_i^*)$.

More details on optimization of parameters P_i^* is provided in section 3.2.

In **online phase** the same object detector and tracker are applied on fixed size time-window(in our case is 20 frames) to extract tracklets (*flow_6*). The extracted tracklets are matched against the learned database (*flow_7*). Tracker parameters are tuned for each extracted tracklet by parameters of closest tracklet match. The distance of one tracklet with a learned tracklet via tracklet representation is provided in section 3.3. Finally, tracker with parameters tuned for each tracklet is applied on the current time-window (*flow_8*) and then tracklets are updated.

3. Tracker parameters tuning

3.1. Hypothesis

In order to select the best tracker parameters for each tracklet, the proposed approach relies on a hypothesis that if representations of two tracklets are close enough, the learned optimal tracking parameter values of one tracklet could be applied effectively for the other one. The hypothesis is formalized as follow:

$$\begin{aligned}&\text{If } (\|\{Tr_j, Tr_j^c s\} - \{Tr_i, Tr_i^c s\}\| < \epsilon_1)\\ &\quad\text{and } (Q(\Im(\{Tr_i, Tr_i^c s\}, P_i^*), GT) > \theta)\\ &\quad\Rightarrow Q(\Im(\{Tr_j, Tr_j^c s\}, P_i^*), GT) > \theta + \epsilon_2\end{aligned} \tag{5}$$

where $\|\{Tr_j, Tr_j^c s\} - \{Tr_i, Tr_i^c s\}\|$ is the tracklet representation distance (provided in section 3.3) of two tracklets Tr_i and Tr_j, Q is the tracking performance of tracking algorithm $\Im$, GT stands for tracking ground-truth and P_i^* is the optimal parameter set of tracklet Tr_i. In this work, we use the Mostly-Track(MT) metric (detailed in experiment part) and the tracking time metric in [9] to evaluate the tracking performance Q.

The purpose of hypothesis is that if the representation $\{Tr_j, Tr_j^c s\}$ of extracted tracklet Tr_j in online phase is matched against any record in the database $\{Tr_i, Tr_i^c s\}$, the tracker could gain the optimal performance for the extracted tracklet when applying the according learned parameter set P_i^*. Besides that, this hypothesis is also applied in training phase. If tracklet Tr_j's representation is closed enough to previous learned tracklet Tr_i, they could use the same optimal parameters. The reliability of the hypothesis will be validated in the experiment part.

3.2. Offine tracking parameter learning

We have a training video segmented by context and now we want to learn the best parameters for each tracklet and save it in database. For learning, we are using simulated annealing (SA). This methods helps in cases where exhaustive search is impossible. SA is an alternative to gradient descent that can stuck in local optimization. The method is meta-heuristic and approximate the global optimum in large searching space.

Simulated annealing based optimizer model The tracking parameters are tuned based on the tracker performance which is evaluated against the ground truth information. Therefore, the objective function is defined by finding the optimal tracker parameter set to maximize the tracking performance $Q(\Im(\{Tr_i, Tr_i^c s\}, P_i)$. Then, the objective function is determined:

$$P_i^* = \arg\max_{P_i} Q(\Im(\{Tr_i, Tr_i^c s\}, P_i), GT) \tag{6}$$

We apply the multiple-SA method to find the optimal parameter setting. Running multiple optimizers in parallel increases the searching speed. The starting points are selected by dividing the searching space into subsets and selecting the middle point of each subset. Therefore, the best performance of optimizers will approximate more accurately the global optimized values. Learned parameter values according to the optimizer getting the highest performance are accepted as the optimal tracker parameter set.

3.3. Online parameter tuning

In the testing phase, tracker is firstly applied for each the video segmented by time-window to extract tracklets and their surrounding contexts. Then, the extracted tracklet is matched to closest learned tracklet to get the optimal parameters. The tracklet representation distance are computed to compare two tracklets. Finally, the tracker with tuned parameters is applied for each tracklet on current time-window and tracklets are updated.

Tracklet representation distance In order to match two tracklets, we focus on the discriminative features of tracklet. Two tracklets need to have similar surrounding features. Therefore, the tracklet representation distance $\|\{Tr_j, Tr_j^c s\} - \{Tr_i, Tr_i^c s\}\|$ shown in Equation 5 is computed relying on the distance of individual feature discriminative level and the similarity of surrounding features between these tracklets compared to their corresponding surrounding tracklets. This distance is formalized as follow:

$$\begin{aligned}&\|\{Tr_j, Tr_j^c s\} - \{Tr_i, Tr_i^c s\}\|\\ &\simeq \beta \times \|Disc^{F_j^O}(Tr_j, Tr_j^c s) - Disc^{F_i^O}(Tr_i, Tr_i^c s)\|\\ &+ (1-\beta) \times Simi(F_j^{OE}, F_i^{OE})\end{aligned} \tag{7}$$

where $Disc^{F_i^O}(Tr_i, Tr_i^c s)$ and $Disc^{F_j^O}(Tr_j, Tr_j^c s)$ are the discriminative levels of tracklets Tr_i and Tr_j with its surrounding tracklets, respectively. $Simi(F_j^{OE}, F_i^{OE})$ is surrounding feature similarity of Tr_i and Tr_j. The weight β shows the reliability of discrimination of feature pool F^O over the similarity of F^{OE}. We set β values by 0.7. Define that $p \in \{i, j\}$, N is the size of F_p^O, $Disc^{F_p^O}(Tr_p, Tr_p^c s)$

and $Simi(F_j^{OE}, F_i^{OE})$ in equation 7 are deployed by:

$$Disc^{F_p^O}(Tr_p, Tr_p^c s) = \frac{\sum_{k=1}^{N} \omega_i^k \times Disc^k(Tr_p, Tr_p^c s)}{\sum_{k=1}^{N} \omega_i^k} \tag{8}$$

$$Simi(F_j^{OE}, F_i^{OE}) = \frac{\sum_{k=N+1}^{N+3} \gamma^k \times Simi(F_j^k, F_i^k)}{\sum_{k=1}^{N} \omega_i^k} \tag{9}$$

$$Disc^k(Tr_p, Tr_p^c s) = 1 - \tilde{X}(Simi(F_p^k, (F_p^c)^k)) \tag{10}$$

By equation 8, the discriminative level $Disc^{F_p^O}(Tr_p, Tr_p^c s)$ is computed by the weighted average of all tracklet individual features' discrimination $Disc^k(Tr_p, Tr_p^c s)$ of tracklet Tr_p to its surrounding tracklet set $Tr_p^c s$. Tracklet feature's discrimination $Disc^k(Tr_p, Tr_p^c s)$, shown in equation 10, is inversely proportional to the median $\tilde{X}$ of their feature similarities. The surrounding feature weights γ^k are set by values 0.5, 0.2, 0.3 to occlusion, mobile object density and contrast features, respectively. The assignation means that occlusion context is the most focused on and need to be fit as much as possible to learned tracklet. The way to compute tracklet feature similarities is provided in section 2.1.

Since the reliability of tracklet features is influenced by context, the individual feature weights ω need to be set and tuned along change in context. Therefore, tracker parameters P^* defined in section 2.3 in this approach represents for features weight ω.

4. Experiments

In this part, we evaluate the performance of the proposed framework. The baseline tracker [5] is extended in [6] by tracker parameter tuning for the whole context. We further extended methods from [5, 6] in the framework by parameter tuned for each tracklets. All mentioned methods are compared against five state of the art methods.

4.1. Training phases

The proposed approach is trained on nine video sequences: six videos from CAVIAR dataset[1] and three from ETISEO dataset[2]. The videos are selected such that they represent a variety of tracking contextual information (e.g..low/high density of object in the scene, strong/weak object contrast). The offline training phase requires the ground-truth of object tracking as input. From the hypothesis shown in equation 5, some tracklets are close to each others then we use only representative tracklet. Therefore, 284 tracklet representations are learned after training 780 samples. This learned tracklet database is used as reference to automatically tune parameters for tracklets in online phase.

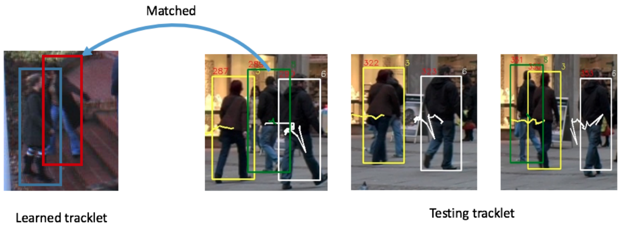

Figure 3. TUD-Stadtmitte dataset: Tracklet ID_8 represented by color "green" matches to closest tracklet in learned dataset to recover mis-detection caused by occlusion.

4.2. Testing phases

The proposed framework is evaluated on 3 video sequences in 2 public datasets (PETs2009 and TUD). For all these videos, the observed scenes are different from the ones of training videos. Upon concatenating the video over 20 frames, the tracker parameter tuner in proposed approach adapts the tracklet feature weights to the change of its surrounding context based on learned database.

Metrics The performance of the tracker is compared with respect to two metrics defined in [7]: Multi-object tracking precision($MOTP$) and multi-object tracking accuracy($MOTA$). In addition, some other metrics are proposed to use. Let GT be the number of trajectories in ground-truth of the testing video. MT(Mostly Track) shows the ratio of mostly tracked trajectories, ML (Mostly Lost) represents the ratio of mostly lost trajectories and PT (Partially Track) is the ratio of partially tracked trajectories($PT = GT - MT - ML$).

PETs 2009 dataset The sequence S2_L1, camera view 1, is selected for testing because this sequence is used for evaluation in several state of the art trackers. It consists of 794 frames with 21 mobile objects with different degrees of inter-person and person-object occlusion.

As visualized in figure 2, we choose two learned tracklet representations from CAVIAR dataset and testing tracklet from sequence S2˙L1, view 1. The color of testing tracklet represented by red bounding-box has not much different to its surrounding tracklet (represented by blue bounding-box) but they move with inverse direction. Therefore, the motion feature is important to discriminate this tracklet to others. Linked to explanation in section 2.2, tracklet context (b) also focuses on motion to discriminate tracklets. Therefore, based on the tracklet representation distance, the tracklet in online phase is more closed to context (b) than (a) and tracker uses the optimal tracker parameters tuned for tracklet context (b) to control feature weights for tracklet in online phase.

TUD datasets The second test is conducted with the TUD dataset(including TUD-Stadtmitte and TUD-Crossing sequences). Both of these sequences are quite short, with

[1] homepages.inf.ef.ac.uk/rbf/CAVIAR/

[2] www-sop.inria.fr/orion/ETISEO/

Dataset	Method	MOTA	MOTP	GT	MT	PT	ML
PETS2009 - S2L1˙View1	Shitrit *et al.* [12]	0.81	0.58	21	–	–	–
	Bae **et al.**-*global association* [2]	0.73	0.69	23	100	0	0.0
	Chau *et al.* [5]	0.62	0.63	21	–	–	–
	Chau [6]([5] + parameter tuning for whole video context)	0.85	0.71	21	–	–	–
	Ours ([5] + Proposed approach)	0.86	0.73	21	76.2	14.3	9.5
TUD-Stadtmitte	Andriyenko *et al.* [1]	0.62	0.63	9	60.0	20.0	10.0
	Milan *et al.* [8]	0.71	0.65	9	70.0	20.0	0.0
	Chau *et al.* [5]	0.45	0.62	10	60.0	40.0	0.0
	Chau [6]([5] + parameter tuning for whole video context)	–	–	10	70.0	10.0	20.0
	Ours ([5] + Proposed approach)	0.47	0.65	10	70.0	30.0	0.0
TUD-Crossing	Tang *et al.* [13]	–	–	11	53.8	38.4	7.8
	Chau *et al.* [5]	0.69	0.65	11	46.2	53.8	0.0
	Ours ([5] + Proposed approach)	0.72	0.67	11	53.8	46.2	0.0

Table 1. Tracking performance. The best values are printed in red.

more or less than 200 frames, but they contain challenges for trackers due to heavy and frequent object occlusions. Figure 3 shows a snapshot of the tracking performance of the proposed algorithm. Testing tracklet with low light intensity, its appearance color is not discriminative with surrounding tracklets but it moves in different direction to others. It is closest to learned tracklet represented by color ”red”, therefore, the parameters tuned for this tracklet have values: 0.512 for motion feature while 0.215 for color histogram and 0.193 for color covariance. Thanks to tuned parameters, the testing tracklet’s mis-detections are recovered correctly.

Comparison is shown in Table 1 over three testing video sequences. In most cases, our proposed approach has equal or better results compared to state-of-the-art trackers, the baseline tracker [5] as well as parameter tuning method [6].

In particular, in PETS2009 dataset with S2L1-View1 sequence, the proposed approach performance has higher result than state of the art trackers [12, 2] and parameter tuning method for whole context [6] which uses the same baseline tracker [5] over MOTA and MOTP metrics. Especially, thanks to the proposed parameter tuning method, the baseline tracker [5] is improved significantly, from 0.62 to 0.86 for MOTA value and from 0.63 to 0.73 for MOTP value. Metrics MT and ML which are evaluated by percentage of ground-truth objects whose trajectories are covered by tracking output (at least 80% for MT and less than 20% for ML). Comparing our tracker to tracker [2] in case of global association method, two methods use different ground-truth. In particular, when objects leave the scene and come back, our ground-truth labels these objects as the same but other considers these objects as different. Furthermore, the object detector we use loses three objects. Then comparing by trajectory information between these trackers is not reasonable.

On both TUD sequences (TUD-Stadtmitte and TUD-Crossing), our approach does not lose any object. The metric MT are also on the top compared to other referenced methods from state of the art. We have lower performance compared to tracker [1, 8] on the sequence TUD-Stadmitte in MOTA metric because this metric relies remarkably on the overlap between grounth-truth boundingbox and detector output. In the case we would have a better detector, our performance should be significantly improved and comparable with other one.

5. Conclusions and future works

This paper proposes a new framework which online tunes tracker parameters to adapt tracker to video context variation. It tunes parameters for each tracklet instead of for the whole video to ensure that tuned parameters characterize the context around each tracklet. A new way to represent tracklet’s surrounding context is proposed to highlight its discrimination to other tracklets. Moreover, this framework can tune any number of tracking parameter and could be flexibly applied for other trackers with different tracker parameter set. The experimental results show the significant performance improvement of our approach compared to tracker using static parameter values, some parameter tuners as well as some state of the art trackers over three public benchmark datasets. The more characterized tracklets are learned, the higher accuracy the multi-object tracker reaches. In the future work, to reduce the reference time to find the best learned tracklet in a huge learned database, we will propose a method to index learned tracklets.

Acknowledgement

The research leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Union’s Seventh Framework Programme FP7/2007-2013/ under REA grant agreement n°[324359]. This work is also supported by the Provence-Alpes-Cote d’Azur Region (PACA).

References

[1] A. Andriyenko and K. Schindler. Multi-target tracking by continuous energy minimization. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1265–1272, June 2011.

[2] S. H. Bae and K. J. Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1218–1225, June 2014.

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

[4] D. P. Chau, J. Badie, F. Bremond, and M. Thonnat. Online Tracking Parameter Adaptation based on Evaluation. In *IEEE International Conference on Advanced Video and Signal-based Surveillance*, Krakow, Poland, Aug. 2013.

[5] D. P. Chau, F. Brmond, and M. Thonnat. A multi-feature tracking algorithm enabling adaptation to context variations. *CoRR*, abs/1112.1200, 2011.

[6] D. P. Chau, M. Thonnat, F. Bremond, and E. Corvee. Online Parameter Tuning for Object Tracking Algorithms. *Image and Vision Computing*, 32(4):287–302, Feb. 2014.

[7] C.-H. Kuo and R. Nevatia. How does person identity recognition help multi-person tracking? In *CVPR*, pages 1217–1224. IEEE Computer Society, 2011.

[8] A. Milan, K. Schindler, and S. Roth. Multi-target tracking by discrete-continuous energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2015.

[9] A.-T. Nghiem, F. Bremond, and M. T. adn V Valentin. Etiseo, performance evaluation for video surveillance system, 2007. AVSS.

[10] A. T. Nghiem, F. Bremond, and M. Thonnat. Controlling background subtraction algorithms for robust object detection. In *Crime Detection and Prevention (ICDP 2009), 3rd International Conference on*, pages 1–6, Dec 2009.

[11] T. L. A. Nguyen, D. P. Chau, and F. Bremond. Robust global tracker based on an online estimation of tracklet descriptor reliability. In *Advanced Video and Signal Based Surveillance (AVSS), 2015 12th IEEE International Conference on*, pages 1–6, Aug 2015.

[12] H. B. Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Tracking multiple people under global appearance constraints. In *2011 International Conference on Computer Vision*, pages 137–144, Nov 2011.

[13] S. Tang, M. Andriluka, A. Milan, K. Schindler, S. Roth, and B. Schiele. Learning people detectors for tracking in crowded scenes. In *2013 IEEE International Conference on Computer Vision*, pages 1049–1056, Dec 2013.

[14] J. H. Yoon, D. Y. Kim, and K.-J. Yoon. Visual tracking via adaptive tracker selection with multiple features. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part IV*, ECCV'12, pages 28–41, Berlin, Heidelberg, 2012. Springer-Verlag.

[15] A. R. Zamir, A. Dehghan, and M. Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part II*, ECCV'12, pages 343–356, Berlin, Heidelberg, 2012. Springer-Verlag.

[16] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.