IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 12, 2019, accepted June 28, 2019, date of publication July 2, 2019, date of current version July 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2926416

Fast Online Multi-Pedestrian Tracking via Integrating Motion Model and Deep Appearance Model

MIAO HE[1,2,3,4,5], HAIBO LUO[1,2,4,5], BIN HUI[1,2,4,5], AND ZHENG CHANG[1,2,4,5]

[1]Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China
[2]Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110016, China
[3]Research Institute, University of Chinese Academy of Sciences, Beijing 100049, China
[4]Key Laboratory of Opto-Electronic Information Processing, Chinese Academy of Science, Shenyang 110016, China
[5]The Key Lab of Image Understanding and Computer Vision, Shenyang 110016, China

Corresponding author: Miao He (hemiao@sia.cn)

ABSTRACT In recent years, multi-object tracking has attracted more and more attention, both in academia and engineering, but most of the recent works do not pay attention to the speed of the algorithm and only pursue the accuracy. In this paper, we propose an online multi-pedestrian tracking algorithm, taking into account both the accuracy and the speed. First, the motion models of the targets are established by the Kalman filter. At the same time, the appearance models of the targets are extracted by the convolutional neural network. Moreover, a data association algorithm is proposed, which integrates the motion information, including scale, intersection-over-union, and distance, and the appearance information, including the current appearance model and the long-term appearance model. With the data association algorithm, the matching between detections and tracklets is realized, and the goal of tracking by detection is achieved. We compare the proposed algorithm with other algorithms on the MOT15 benchmark and the MOT16 benchmark. The experiment results show that the algorithm has high accuracy and good real-time performance.

INDEX TERMS Online, pedestrian detection, multi-object tracking, re-identifying, Kalman filter, data association.

I. INTRODUCTION

Multi-pedestrian tracking is a key technology in the field of image processing. It has important applications in many fields, such as public security, intelligent transportation, video surveillance and robot vision [1]. With the increasing demand of these applications in recent years, multi-pedestrian tracking has attracted more and more researchers.

Compared with single target tracking task, multi-target tracking task is more comprehensive and more complex. Challenges including occlusion, deformation, motion blur, crowded scenario, fast motion, illumination variation, scale variation and other challenging aspects in single target tracking will appear simultaneously [2]. In addition, multi-target tracking also needs to face other complex problems, such as the initialization and termination of the track, the interaction of a large number of similar targets and so on [2]. Therefore, multi-target tracking is still a challenging task in the field of image processing [3].

Thanks to the development of deep learning technology [4]–[8], the pedestrian detection accuracy based on convolutional neural network has reached an unprecedented level [9]. Therefore, the methods of tracking-by-detection are also becoming popular. Compared with detection-free tracking [10], [11], tracking-by-detection does not need to initialize the target manually, and is better at coping with the emergence of new targets and the disappearance of old targets [2]. In this kind of method, the targets of each frame are firstly detected by a detector, and then, these detections between frames are connected into tracklets by a data association algorithm.

In tracking-by-detection algorithms, data association directly affects the final accuracy of the algorithm. Recently, many batch based offline data association methods have been proposed [12]–[14]. These methods take into account all the frames of the whole image sequence and have more

The associate editor coordinating the review of this manuscript and approving it for publication was Marco Martalo.

advantages in tracking accuracy. However, these algorithms have great limitations in practical applications, and are more used in video post-processing [15]. In such areas as intelligent transportation, intelligent security and machine intelligence, only online tracking can meet the needs of the applications. At the same time, these applications also have certain requirements for the speed of tracking algorithms.

In order to meet these requirements, we propose an online tracking-by-detection method that takes account of accuracy and speed. In the first frame of the image sequence, the detection results are used to initialize Kalman filters in order to establish the motion models of the targets. At the same time, the appearance features of the detections are extracted by convolutional neural network in order to establish the appearance models. These models are used to initialize the tracklets. Starting from the second frame, the detection results are matched with the tracklets by their motion models and appearance models. The matched detection results are used to update the models of tracklets. Unmatched detection results enter new tracklet initialization program. If a initializing tracklet is matched with detections in consecutive multiple frames, the tracklet is regarded as tracking a new target and initialized successfully. Unmatched tracklets enter the disappearance process. If a tracklet fails to match a detection in successive multiple frames, it is assumed that the target tracked by the tracklet disappears from the field of vision and the tracking of the tracklet is terminated. We test and evaluate the proposed algorithm on two challenging datasets, 2D MOT 2015 [16] and MOT16 [17]. The speed of the algorithm is superior, reaching real-time on the 2D MOT2015 dataset. At the same time, the algorithm obtains the precision that is comparable with other top-10 algorithms on this dataset.

The rest of this paper is organized as follows. The progress and related work in the field of multi-target tracking will be introduced in Sec.2. Sec.3 will introduce the principles and details of the proposed algorithm. In Sec.4, we will test and analyze the algorithm, and compare it with other algorithms in detail. The conclusion of this paper and the future work will be discussed in Sec.5.

II. RELATED WORK

In a video sequence $V = \{I_1, I_2, \ldots, I_T\}$ with a length of T, multi-object tracking can be treated as a multi-variable estimation problem. The i-th target observation at time t is represented as O_t^i. The observation of a total of M_t targets at a certain time t is expressed as $O_t = \left\{O_t^1, O_t^2, \ldots, O_t^{M_t}\right\}$. $O_{1:T}$ then represents the set of observations for all targets in the entire video sequence. S_t^i indicates the state of the target i at time t. The target appeared in the video sequence is $S = \left\{S^1, S^2, \ldots, S^M\right\}$, and M is the total number of targets. The aim of multi-target tracking is to find all the corresponding observations for any S^m in $O_{1:T}$ (e.g. $O^m = \left\{O_3^5, O_4^2, \ldots, O_{15}^7\right\}$, and modify the observations to obtain the state of the target, such as $S^m = \left\{S_3^5, S_4^2, \ldots, S_{15}^7\right\}$.

Targets in multi-target tracking tasks can be pedestrians [18], vehicles on the road [19], [20], players on the soccer field [21], groups of animals [22], or even different parts of a single target [23]. In this paper, we mainly focus on pedestrian tracking for the following reasons: first, pedestrian is a typical non-rigid object, which is an ideal example of multi-object tracking; second, pedestrian targets are relatively rich in scenes, the problems encountered are more comprehensive and complex; third, pedestrian tracking are closer to human daily life, and have great application value.

According to the different initialization modes, multi-target tracking algorithms can be divided into two types: DBT (Detection-Based Tracking) and DFT (Detection-Free Tracking). DBT, or tracking-by-detection, is a more popular tracking framework [12], [13], [24]–[26], which can automatically find new targets without manual annotation of the first frame of the target. The DFT algorithm does not depend on the detector, and can track any type of target, so it also has high application value [2].

Multi target tracking algorithms can also be divided into two modes: online tracking and off-line tracking. The difference is whether the future frames are used when processing the current frame. In online tracking methods, only the current frame and several previous frames are used to process the tracking [10], [11], [27], whereas the observation results of the future frames are needed to be obtained in advance in off-line tracking methods [28]–[30]. In recent years, applications, including pedestrian flow analysis [31], automatic driving technology [32], automatic traffic management [33] and intelligent public security [34], require real-time tracking to analysis results and make decisions. For these applications, offline tracking cannot meet the requirements. Compared with offline tracking, online tracking can be used not only for offline post-processing tasks, but also for those tasks that require results processed in real time, so it has wider application prospects. For this reason, this paper focuses on online multi-target tracking algorithm.

Appearance models are widely used in the field of multi-target tracking. D. Mitzel et al. use color histogram to build appearance models for real-time multi-target tracking [35]. T. Yu et al. proposed a multi-target tracking method based on hog features to construct part-based person representations [36]. The regional covariance matrix is also used as the appearance model for multi-target tracking by Kuo *et al.* [12]. In recent years, convolutional neural networks have been used to extract visual features as appearance models because of the obvious advantages of deep learning methods [37]. An appearance model based on deep learning feature extraction method is proposed by Ma *et al.* [38], which trains the feature extraction ability of the convolutional network on a recognition data set and improves the effect of multi-target tracking.

In the field of multi-target tracking, the motion model plays a same important role as the appearance model. Since the motion of the target in the image is usually gentle, the estimation of the moving trend of the target can predict the position

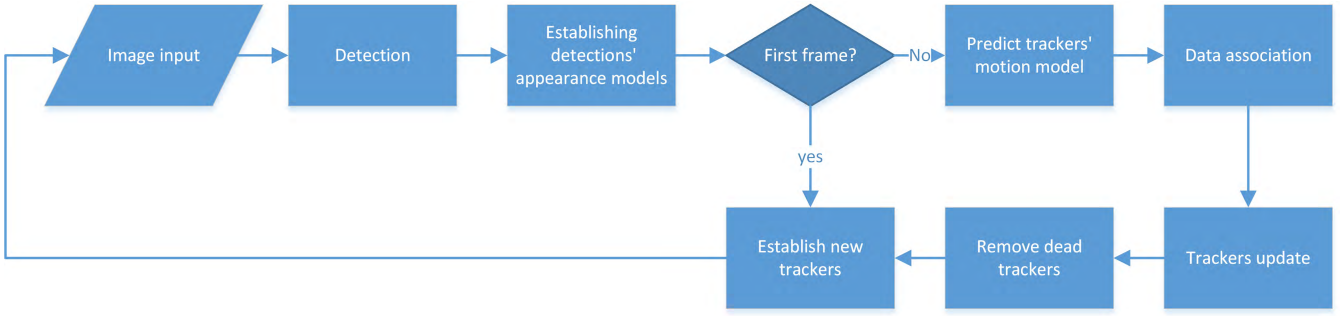

FIGURE 1. Flow chart of the tracking algorithm.

of the target in the next frame, so as to reduce the search area and even directly obtain the tracking results. Linear motion model is the most commonly used motion model in the field of multi-target tracking. A. Bewley et al. proposes a fast online multi-target tracking algorithm based on Kalman filter which establish a linear uniform motion model, and achieved good effect [39]. At the same time, there are also multi-target tracking algorithms which use non-linear models to model and predict the motions of the targets. B. Yang et al. build a non-linear motion map to better explain direction changes, making the connection of short tracklets of the target more robust [40].

In tracking-by-detection algorithms, a data association algorithm is needed to match the observation results with the tracklets. The data association algorithm is a deterministic optimization algorithm to find the maximum posteriori solution for multi-target tracking. By casting data association as a bipartite graph matching problem, one can use such algorithms as greedy bipartite assignment algorithm or Hungarian algorithm to obtain the solution. B. Wu et al. proposes a multi-target tracking method based on greedy bipartite assignment algorithm [41]. Hungary algorithm is introduced by A. A. Perera et al. as data association algorithm for multi-target tracking [42].

III. METHODOLOGY

We propose an on-line tracking-by-detection method, which consists of the following five steps. Firstly, detect pedestrians by convolutional network detector. Secondly, build appearance models of detections by re-identify network. Next, predict motion state of tracklets by Kalman filter. Then, associate detections and tracklets using data association algorithm with motion models and appearance models. Finally, complete the multi-pedestrian tracking task with tracklet update strategy. The overall algorithm flow is shown in Figure 1.

A. DETECTOR

For tracking-by-detection methods, the target detecting effect has a direct impact on tracking accuracy. In this paper, we use high resolution YoloV3 [43] detector to detect pedestrians. For deep learning based detectors, including Yolo and SSD, the input resolution of the network has an important impact on the accuracy of the model. We train our detector based on pictures with person in MSCOCO dataset.

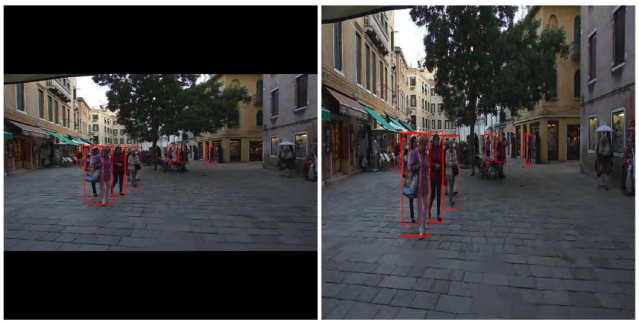

FIGURE 2. The results of the same image resized to 608*608 resolution by letterbox (left) and bilinear interpolation (right) respectively.

Instead of using the letterbox method, which is the same as the author of Yolo, we chose bilinear interpolation to resize the image. Letterbox method can ensure that the input image does not deform, but the resolution of the input of the network has a certain waste. While letterbox method and bilinear interpolation method are used to resize a single image, the sizes of a same object in the two resized images are different. The target is larger in the image obtained from bilinear interpolation than in the image obtained by letterbox, as is shown in Figure 2. In an image with a resolution of 1920*1080, for a target with a height of 60 pixels, after letterbox and bilinear interpolation, the heights of the target in the new image are 19 and 33.8, respectively. Therefore, using bilinear interpolation method to resize the image as network input, the ability of network to detect small targets will be improved, which is more suitable for multi-pedestrian tracking application scenarios. To validate this idea, we do experiment on the MSCOCO dataset. We train the detector with 64115 images which have person labels in the train dataset and test on 2693 images with person labels in the verification dataset. The experiment results show that with bilinear interpolation instead of letterbox, the detection mAP (mean average precision) of all pedestrian targets changes from 72.64% to 72.60%, almost unchanged. For targets whose height is less than a quarter of the image height, the mAP is increased from 56.14% to 57.78%, which proves the effectiveness of

FIGURE 3. Detection results with NMS (a) and Soft-NMS (b).

the method in improving the pedestrian detection effect of small targets.

In the detection results of the detector, there are usually multiple anchors corresponding to the same target, so it is necessary to remove redundant region proposals. Non maximum suppression (NMS) is the most common way to remove redundant proposals. However, in the multi-pedestrian tracking scenario, the non maximum suppression method will greatly increase the miss rate because of the relatively dense pedestrians. Therefore, unlike Yolo authors, we choose soft non maximum suppression(Soft-NMS) [44] method to remove candidate regions. Instead of eliminating the proposal directly, Soft-NMS will reduce the confidence of the proposal whose intersection over union with some other proposal is greater than the threshold and with lower confidence. This process can be expressed by the following functions,

$$s_i = \begin{cases} s_i, & IOU(d_m, d_i) < N_i \\ s_i(1 - IOU(d_m, d_i)), & IOU(d_m, d_i) \geq N_i \end{cases} \quad (1)$$

where d_i is a detection result with score s_i, d_m is another detection result which has higher score than d_i, N_i represent the threshold of Soft-NMS.

As is shown in Figure 3, with the help of Soft-NMS, the problem of missing detection in dense targets has got a certain degree of reduction, and the redundant region proposals can still be removed.

B. MOTION MODEL

The motion model used in the proposed algorithm is a linear motion model based on Kalman filter. Each target is modeled by its position, velocity, aspect ratio and scale. The model is expressed as $x = [u, v, s, r, u', v', s']^T$,where (u, v) represents the pixel location of the target center, s represents the scale of the target, and r represents the aspect ratio of the target. (u', v') represents the speed of the target in horizontal and vertical directions, s' represents the changing rate of the scale of the target. Here we assume that the aspect ratio of a target is constant, so there is no r' in the state variables.

In each frame, the state of the target is predicted firstly. When the predicted result matches the observed result of a detector successfully in the data association algorithm, the state of the target is the optimal estimation updated by the observed result with Kalman filter. If the predicted result does not have a corresponding detection result, the Kalman filter prediction result is used to update the state of the target. The Kalman filter model uses uniform linear motion model, so at this time in the tracking result, the target moves uniformly and in a straight line, the scale of the target changes uniformly according to the trend, and the aspect ratio of the target remains unchanged, which is consistent with the intuitive feeling of human towards the movement of an occluded target.

C. APPEARANCE MODEL

The appearance model is built by convolutional neural network and trained by re-identifying datasets. The overall structure of the network is shown in Figure 4. As the main body of the network, we use the pruned VGG network [45]. The forward propagation speed of the network is five times faster than that of the original VGG network, and the increase of top-5 error on the ImageNet is only 1.7%, which meets the need of our pursuit of speed without affecting the accuracy.

The input resolution of the network is 128×64. Because of the obvious differences in color and shape between the upper and lower parts of the pedestrian, the upper and lower parts of the input image are divided into two inputs in the input layer. The network is trained with a triplet loss function, so in the training process, each input of the network is three images, the first two are a pair of positive samples, and the third one is a negative sample. The three images are divided into six parts through the input layer, which is then input into the pruned VGG network. In the network, the parameters of the three branches of the upper-body feature extracting network are shared, and the parameters of the three branches of the lower-body feature extracting network are shared. The six parts of the input are merged to get three feature vectors corresponding to the three samples. Finally, the triplet loss function is used to adjust the parameters. The following expression shows the triplet loss function,

$$L_{tri} = \sum_{i=1}^{N} (D_{a,p} - D_{a,n} + margin) \quad (2)$$

where $D_{a,p}$ is the distance between the anchor sample and the positive sample, $D_{a,n}$ is the distance between the anchor sample and the negative sample, *margin* is the distance used to separate the positive pair form the negative.

Triplet loss was first proposed in Google's FaceNet [46]. Its main idea is to reduce the intra-cluster distance and increase the between-cluster distance in the feature space, so as to achieve the clustering effect. In the field of re-identification, this loss function is proved to be significantly better than softmax loss function. However, triplet loss has a high probability of selecting very similar positive samples and very dissimilar negative samples for training, which leads to weak generalization ability and error-prone classification of similar samples. This problem has a certain impact on tracking accuracy in multi-pedestrian tracking. Therefore, we introduce hard sample mining [47]. While training the network, 18 pedestrians are randomly selected from the training dataset, then four images from each pedestrian are

FIGURE 4. Overall structure of the network for appearance model.

randomly selected and finally 72 images are obtained. In these 72 images, each image is treated as a training sample together with the farthest positive sample and the nearest negative sample in the current feature space. The final loss function can be expressed as:

$$L_{tri} = \sum_{p=1}^{p}\sum_{i=1}^{I}(\max_{t=1,2,\ldots,I} D(y_p^i, y_p^t) - \max_{\substack{n=1,2,\ldots,p \\ m=1,2,\ldots,I \\ n\neq p}} D(y_p^i, y_n^m) + margin) \quad (3)$$

where y_p^i is the feature of the i-th image of the p-th pedestrian. In each frame, the appearance model of each detection is first established. If the detection and a tracklet match successfully in the data association algorithm, the model of the tracklet is updated with the appearance model of the detection. In order to take account of the influence from the confidence of the detection, we use the following formula.

$$model_t = \frac{model_{t-1} + confidence * model_{in}}{1 + confidence} \quad (4)$$

where $model_t$ is the feature of a tracklet in the t-th frame, *confidence* is the confidence of the detection which is matched successfully with the tracklet.

Through this update method, the model can obtain short-term memory. However, in the process of occlusion, occlusion will appear in the bounding box of the detection, so using the detection appearance model to update the tracklet in these frames will pollute the appearance model of the tracklet. Therefore, two tracklet appearance models, a long-term one and a short-term one, are needed to be established. The long-term model is conducive to target matching after occlusion, and the short-term model is conducive to target matching during occlusion. In order to achieve such a purpose, a queue of appearance models is established for each tracklet. The queue stores 25 models at most, and if the queue is full, the oldest model is deleted when a new model enters. When matching detections and tracklets, the oldest tracklet appearance models and the latest one are selected, and the Euclidean distances between them and the detection appearance model are calculated. The smaller one in the two Euclidean distances is selected as the appearance model distance between the tracklet and the detection.

D. DATA ASSOCIATION ALGORITHM

From the motion model and appearance model, the obtained data includes bounding boxes and appearance feature vectors of detections, predicted bounding box and long-term and

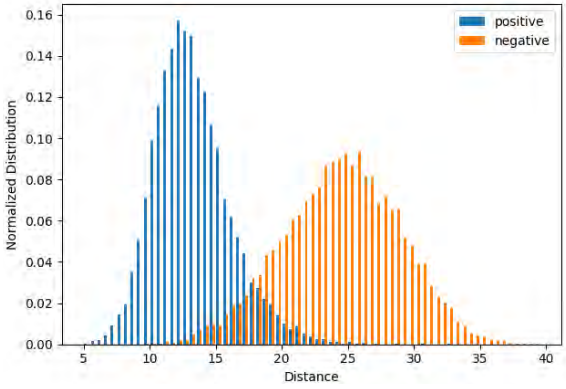

FIGURE 5. Statistical results of appearance model distance tested by 50000 pairs of positive and 50000 pairs of negative samples. The model is trained on market 1501 dataset and tested on the re-identifying dataset extracted from MOT15 training set.

short-term feature vectors of tracklets. Based on the data obtained, the data association algorithm is used to match tracklets and detections. The cost used by the association algorithm includes two parts: constraint and distance.

Constraints include Intersection over Union (IoU) constraint, appearance model constraint and scale change constraint. In the tracking phase, we require the IoU of the detection box and the tracking box to be greater than zero. At the same time, the distance between the appearance models should be less than a certain threshold T_1, otherwise the tracklet and detection are unmatched. If the appearance model distance is smaller than another smaller threshold T_2, the detection and the tracklet should be determined to match each other. Here, the thresholds are derived from a large number of positive sample pairs and negative sample pairs by statistical methods, as is shown in Figure 5. T_1 is the maximum distance for these positive sample teams, and T_2 is the minimum distance for these negative sample pairs. This method is more reliable than directly classify a pair of models with a threshold.

In addition, the scale difference is required to be less than a threshold, which is used to deal with the situation that the detection and tracklet are close enough to each other on the image, making the IoU greater than zero, but the actual distance is large. Instead of using the area ratio of the tracklet to the detection, the following formula is used to calculate the scale change to cope with partial occlusion of the detection.

$$D_{scale} = min(\frac{max(w_1, w_2)}{min(w_1, w_2)}, \frac{max(h_1, h_2)}{min(h_1, h_2)}) - 1 \tag{5}$$

where (w_1, h_1) and (w_2, h_2) are respectively the length and width of the bounding boxes of the detection and tracklet to be matched. As shown in Figure 6, compared with the area, this method can better represent the scale difference between the targets. Among the three constraints, appearance model constraint has the highest priority, followed by scale constraint, and IoU constraint is the lowest.

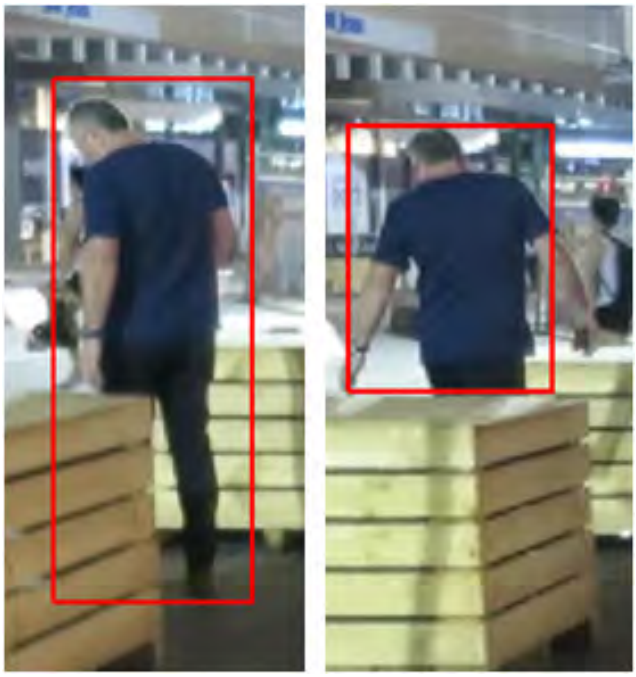

FIGURE 6. Detection results for identical targets on two graphs. The area ratio is about 2:1, whereas D_scale is close to 0.

Distance includes standardized pixel distance and appearance model distance. The standardized pixel distance is a standardized distance based on the pixel distance between detection and tracklet and their scales. The formula is as follows.

$$D_{sp} = \sqrt{(\frac{x_1 - x_2}{min(w_1, w_2)})^2 + (\frac{y_1 - y_2}{min(h_1, h_2)})^2} \tag{6}$$

This distance can, to a certain extent, indicate the actual distance between detections and tracklets. Compared with Euclidean distance, this distance is robust to the scale of the targets on the image. The calculation method of model distance is shown in Seq.3.3. For the tracklet whose target is lost in several frames, the reliability of its motion model decreases as the time of target loss increases. Therefore, when the standardized pixel distance and the model distance are fused, the weighting coefficients of the two distances vary with the time of target loss. The specific method is shown in the following expressions.

$$lamb = \begin{cases} \frac{t_{loss}}{5} & t_{loss} \leq 5 \\ 1 & t_{loss} > 5 \end{cases} \tag{7}$$

$$D = D_{app} * (1 + lamb) + D_{sp} * (1 - lamb) \tag{8}$$

We choose Hungary algorithm as matching algorithm. First, we quantify the satisfaction of constraints. After quantization, the quantization cost between detections and tracklets satisfying the constraints is 0. If the constraints are not satisfied, the quantization cost is given a large value, which makes it difficult for them to match each other in

TABLE 1. Quantization results of algorithms on MOT2015 dataset. The green color indicate the best performing tracker in online trackers and batch trackers on each metric.

	MOTA	IDF1	MT	ML	FP	FN	ID Sw.	MOTP	ID Precision	ID Recall	processing
NOMT	55.5	59.1	39.0%	25.8%	5594	21322	427	76.6	69.3	51.5	batch
TSML_CDE	49.1	52.1	30.4%	26.4%	5204	25460	637	74.3	64.9	43.5	batch
DMT	44.5	49.2	34.7%	22.1%	8088	25335	684	72.9	58.9	42.3	batch
CDA_DDAL	51.3	54.1	36.3%	22.2%	7110	22271	544	74.2	62.9	47.4	online
MDP_SubCNN	47.5	55.7	30.0%	18.6%	8631	22969	628	74.2	64.2	49.2	online
SORT	33.4	40.4	11.7%	30.9%	7318	32615	1001	72.1	54.5	32.1	online
NSH(proposed)	52.2	57.2	34.4%	16.1%	7464	21353	578	74.9	65.6	50.8	online

Hungarian algorithm. When the model distance between the detection and the tracklet is less than the threshold T_2, a large negative value is given to the quantization cost, which ensures that they match each other in Hungarian algorithm. The priority between constraints is reflected in the magnitude of the corresponding quantization cost. The cost matrix used in Hungary matching is the sum of the quantified constraint matrix and the distance matrix.

Finally, in the matched results, pairs of tracklets and detections which do not satisfy all the constraints and whose appearance model distance is not less than T_2 are found, and the matchings between these pairs are disconnected.

E. TRACKING STRATEGY

After the matching results are obtained, the motion model and appearance model of a tracklet are updated with the methods in Seq.3.2 and Seq.3.3. In the proposed algorithm, a tracklet has three processes: initialization process, tracking process and target disappearance process. After updating, tracklets in the tracking process output the states to the tracking results.

The unmatched detections are new tracking target candidates. New tracklets are set up for these candidates and then enter the initialization process. During initialization, the state of the tracklets do not output as a tracking result. If a tracklet matches detection boxes in several consecutive frames, the tracklet is initialized successfully and enter tracking process, and the state of the tracklet begins to output to the tracking result. The number of the consecutive frames should not be too large to prevent the tracklet from being unable to output tracking results for a long time after the target appears. Through experiments, we choose to use three consecutive frames to filter the initialization of the tracklet. With this method, the influence of false positives of the detector on tracking results can be reduced.

For an unmatched tracklet, if the tracklet has not matched a detection for three consecutive frames, it enters the target disappearance process and its state is no longer output to the tracking results. If the tracklet have not matched to any detection for several consecutive frames, it is considered that the target corresponding to the tracklet disappears from the field of view and the tracklet is deleted. If the number of the consecutive frames is too small, the target cannot be tracked after a short period of occlusion, and if the number of the consecutive frames is too large, the ID switches will be increased. Through experiments, we choose to use 25 consecutive frames to filter the deletion of the tracklet. If the tracklet successfully matches a detection within 25 frames, the tracklet restores to the tracking process and its state is output to the tracking result again.

IV. EXPERIMENTS

A. DATASETS

The performance of the algorithm is evaluated by using 2D MOT 2015 dataset and MOT 2016 dataset.

MOT2015 dataset consists of 22 sequences, including 16 popular sequences from KITTI, ETH, PETS and TUD datasets and 6 new challenging video sequences, half for training and half for testing. In terms of the motion of camera, the viewpoint and the weather, the sequences are very different. The annotations for testing data are not released for fairness of evaluation results. Test results need to be submitted to benchmark website to evaluate the effectiveness of the algorithm.

MOT2016 dataset consists of 14 sequences, most of which are new sequences that are more challenging. Sequences are also very different from each other. Compared with the mot2015 dataset, the mean crowd density of the mot2016 dataset is three times higher. The annotations for testing data are not released either.

Datasets rely mainly on MOTA to quantitatively evaluate the performance of the tracking algorithms. The MOTA is a widely accepted measure for evaluating the performance of a tracker, which is shown as following,

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDSW_t)}{\sum_t GT_t} \tag{9}$$

where t denotes the frame index, GT denotes the number of ground truth objects, FN denotes the number of false negatives, FP denotes the number of false positives, $IDSW$ denotes the number of ID switches. At the same time, MOTP, ID F1 score(IDF1), mostly tracked targets(MT), mostly lost targets(ML), and FRAG were also considered. MOTP is multiple object tracking precision, which measures the difference between true positive and ground truth. FRAG indicates the total number of trajectory interruptions.

B. TRACKING RESULTS COMPARED WITH OTHER ALGORITHMS ON MOT2015

1) PERFORMANCE

First, we tested the performance of the algorithm on the mot2015 dataset. In order to verify the effectiveness of

TABLE 2. Detail quantization results of algorithms on sequences in 2DMOT2015. The green color indicate the best performing tracker on each metric and the yellow color indicate the second best one.

sequence	Method	MOTA	IDF1	MOTP	FAF	MT	ML	FP	FN	ID SW	Frag	processing
ETH-Jelmoli	NOMT	51.0	63.5	78.9	1.1	40.0%	37.8%	469	766	8	20	batch
	TSML_CDE	46.6	58.6	74.6	0.8	26.7%	28.9%	345	985	25	36	batch
	DMT	48.2	62.8	75.0	1.7	60.0%	11.1%	758	529	26	74	batch
	CDA_DDAL	47.5	62.0	77.3	1.3	40.0%	35.6%	556	765	12	24	online
	MDP_SubCNN	48.2	65.7	77.3	1.1	35.6%	22.2%	492	814	9	37	online
	SORT	39.0	52.9	74.1	1.0	20.0%	28.9%	439	1071	38	71	online
	NSH(proposed)	62.0	74.0	77.8	0.3	33.3%	31.1%	131	821	12	39	online
ETH-Linthescher	NOMT	66.0	70.3	79.6	0.5	41.1%	35.5%	468	2423	41	56	batch
	TSML_CDE	53.3	56.7	74.8	0.2	14.7%	48.7%	224	3856	89	90	batch
	DMT	60.5	60.2	76.4	1.2	43.7%	24.4%	1425	1963	138	205	batch
	CDA_DDAL	60.0	58.4	77.8	0.9	34.0%	32.5%	1030	2475	69	115	online
	MDP_SubCNN	63.9	67.1	77.1	0.4	24.4%	31.0%	495	2657	70	143	online
	SORT	52.2	54.5	73.8	0.3	14.7%	40.6%	397	3725	144	193	online
	NSH(proposed)	65.6	64.9	76.7	0.6	34.5%	22.8%	718	2274	76	176	online
ETH-Crossing	NOMT	60.8	70.7	82.8	0.2	26.9%	50.0%	38	347	8	10	batch
	TSML_CDE	56.7	59.3	78.6	0.0	15.4%	38.5%	6	418	10	9	batch
	DMT	59.0	54.7	81.5	0.8	30.8%	30.8%	169	221	21	24	batch
	CDA_DDAL	61.4	61.7	79.3	0.5	26.9%	30.8%	113	265	9	17	online
	MDP_SubCNN	63.8	76.5	79.5	0.3	19.2%	26.9%	64	293	6	21	online
	SORT	55.4	49.7	80.3	0.3	15.4%	38.5%	58	368	21	23	online
	NSH(proposed)	64.4	73.0	79.2	0.3	26.9%	26.9%	76	269	12	27	online
KITTI-16	NOMT	47.7	69.5	71.2	1.7	41.2%	5.9%	351	528	10	27	batch
	TSML_CDE	40.7	63.3	68.6	1.6	23.5%	5.9%	336	658	15	69	batch
	DMT	44.7	60.5	69.3	1.1	23.5%	0.0%	232	690	19	43	batch
	CDA_DDAL	50.4	65.2	71.0	1.3	35.3%	5.9%	262	564	18	50	online
	MDP_SubCNN	50.0	66.6	70.3	1.3	35.3%	5.9%	262	566	22	47	online
	SORT	34.6	42.8	70.1	0.7	11.8%	5.9%	144	938	30	60	online
	NSH(proposed)	51.3	70.8	73.1	1.3	29.4%	5.9%	268	542	18	40	online
Venice-1	NOMT	44.8	50.7	76.8	0.2	29.4%	47.1%	88	2428	5	5	batch
	TSML_CDE	31.1	39.9	72.9	0.2	11.8%	47.1%	70	3064	8	17	batch
	DMT	32.4	38.0	70.9	1.2	29.4%	41.2%	527	2538	18	44	batch
	CDA_DDAL	51.3	51.9	72.8	0.8	29.4%	23.5%	367	1842	15	60	online
	MDP_SubCNN	42.6	48.4	76.0	1.6	35.3%	23.5%	729	1867	21	49	online
	SORT	24.7	24.4	69.7	1.1	11.8%	47.1%	485	2888	62	112	online
	NSH(proposed)	46.5	58.0	75.6	1.0	29.4%	17.6%	432	2001	9	69	online

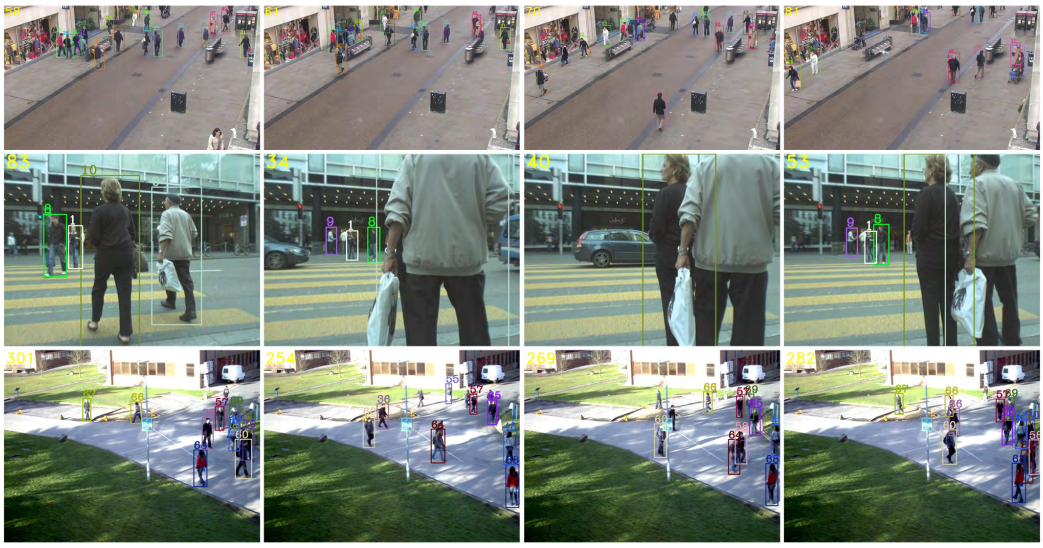

FIGURE 7. Tracking results of the proposed algorithm on MOT2015 dataset.

the algorithm, we select other popular trackers, including NOMT [15], CDA_DDAL [48], TSML_CDE [49], MDP_SubCNN [27], DMT [29] and SORT [39], to compare with the proposed algorithm. The detailed comparison results are shown in Table 1 and Table 2. It can be seen that our algorithm achieves high accuracy as an online algorithm, and can compete with most of the offline algorithms.

TABLE 3. Quantization results of algorithms on MOT2016 dataset. The algorithms with * use the same detector. The green color indicate the best performing tracker in online trackers and batch trackers on each metric.

	MOTA	MOTP	IDF1	MT	ML	FP	FN	ID Sw.	processing
KDNT*	68.2	79.4	60.0	41.0%	19.0%	11479	45605	993	batch
NOMT	62.2	79.6	62.6	32.5%	31.1%	5119	63352	406	batch
MCMOT_HDM	62.4	78.3	51.6	31.5%	24.2%	9855	57257	1394	batch
SORT*	59.8	79.6	53.8	25.4%	22.7%	8698	63245	1423	online
DeepSORT*	61.4	79.1	62.2	32.8%	18.2%	12852	56668	781	online
RAR16wVGG	63.0	78.8	63.8	39.9%	22.1%	13663	53248	482	online
NSH(proposed)*	63.9	78.5	61.5	34.3%	17.7%	9829	55000	913	online

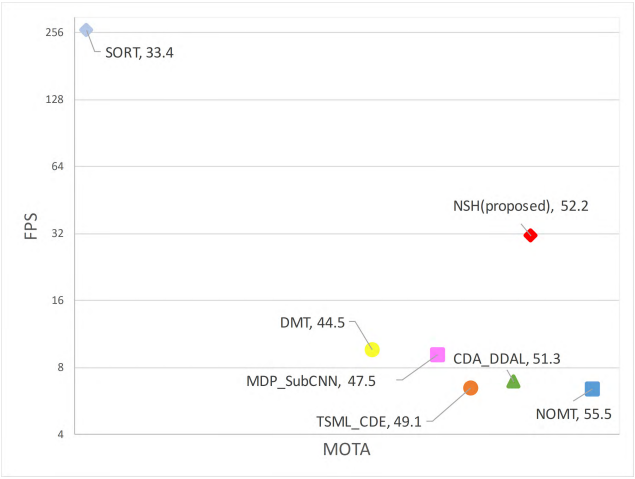

FIGURE 8. The performance comparison of the proposed algorithm and other baseline algorithms on the MOT2015 benchmark.

MOT 2015 dataset only track targets with height greater than 59 pixels, so targets far from the camera are not considered, which is quite similar to the requirements of most applications. Using YOLO detector to detect targets under such condition has achieved good detection results, and also has advantages in detection speed. The visual tracking results of the proposed algorithm is shown on Figure 7.

2) RUNTIME

Most of the multi-target tracking algorithms improve the tracking accuracy by sacrificing the speed of the algorithm. In offline applications, real-time performance is not the key factor. However, in the application of security monitoring, intelligent transportation and intelligent robots, real-time performance is a very important requirement for the algorithm. In Figure 8, we compare both the speed and the accuracy of some popular algorithms published on MOT Challenge with the proposed algorithm. Our algorithm is neither the fastest nor the most accurate, but the advantages of the proposed algorithm in the balance of speed and accuracy can be seen from the figure. On the premise of reaching real-time, our algorithm achieves similar performance with the most accurate algorithm. The proposed algorithm relies mainly on the Tensorflow framework and runs on GTX Titan (Maxwell) GPU. In this dataset, the proposed algorithm runs at 31.3 fps for tracking, which makes it the fastest one in the top-10 algorithms published on the dataset website.

C. TRACKING RESULTS COMPARED WITH OTHER ALGORITHMS ON MOT2016

In order to verify the generalization ability of the proposed algorithm, we also evaluate the effectiveness of the algorithm

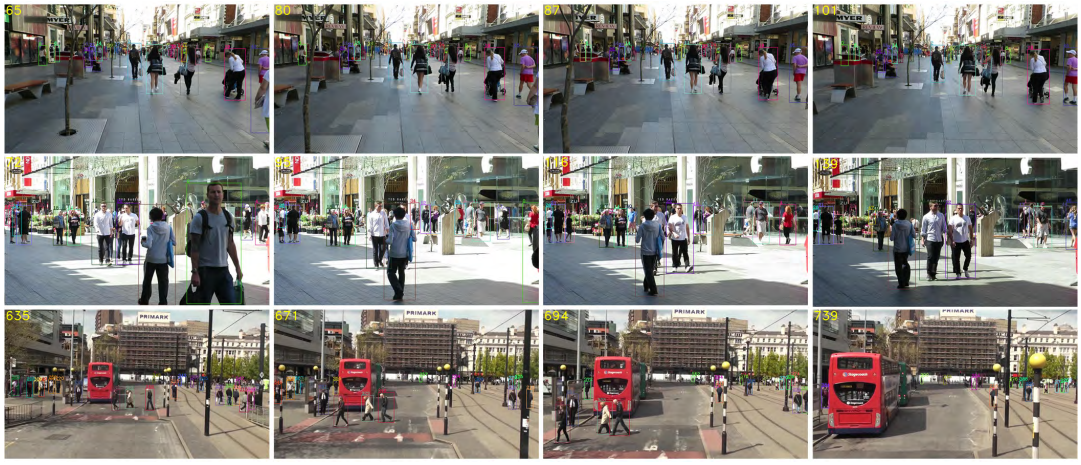

FIGURE 9. Tracking results of the proposed algorithm on MOT2016 dataset.

TABLE 4. The comparison between the proposed algorithm and other algorithms on the MOT16 dataset. The algorithms with * use the same detector. The green color indicate the best performing tracker on each metric and the yellow color indicate the second best one.

sequence	Method	MOTA	IDF1	MOTP	MT	ML	FP	FN	ID SW	Frag	processing
MOT16-01	KDNT*	51.7	62.7	78.1	56.5%	4.3%	1148	1897	42	46	batch
	NOMT	54.1	55.9	75.9	39.1%	26.1%	198	2724	12	16	batch
	MCMOT_HDM	54.7	47.8	76.8	43.5%	21.7%	349	2503	42	43	batch
	SORT*	52.9	45.9	78.6	43.5%	8.7%	758	2206	50	68	online
	DeepSORT*	46.3	57.6	78.2	47.8%	4.3%	1344	2061	29	64	online
	RAR16wVGG	59.5	58.5	78.0	52.2%	8.7%	545	2020	22	44	online
	NSH(proposed)*	48.9	54.0	78.2	47.8%	4.3%	1216	2013	38	66	online
MOT16-03	KDNT*	81.4	64.8	79.4	75.7%	3.4%	5631	13539	267	253	batch
	NOMT	72.7	68.9	80.6	47.3%	11.5%	1061	27319	113	175	batch
	MCMOT_HDM	73.2	55.4	78.4	50.0%	8.8%	4507	23060	487	472	batch
	SORT*	69.1	58.3	79.8	47.3%	10.1%	4436	27442	435	662	online
	DeepSORT*	70.9	68.8	79.5	50.7%	9.5%	5354	24913	151	795	online
	RAR16wVGG	71.3	68.4	78.7	60.8%	9.5%	8700	21129	134	579	online
	NSH(proposed)*	74.8	67.0	78.7	48.0%	9.5%	1812	24173	311	1056	online
MOT16-06	KDNT*	63.6	59.5	80.8	43.9%	19.5%	695	3410	98	124	batch
	NOMT	61.3	66.9	78.6	40.7%	33.0%	726	3680	64	88	batch
	MCMOT_HDM	57.7	53.0	79.7	35.7%	30.8%	553	4143	183	162	batch
	SORT*	57.4	57.8	80.0	28.1%	22.2%	440	4295	181	210	online
	DeepSORT*	60.8	58.6	79.3	40.3%	17.6%	979	3425	118	227	online
	RAR16wVGG	66.0	73.1	80.6	50.2%	21.3%	948	2932	37	83	online
	NSH(proposed)*	60.1	62.3	78.8	39.8%	18.1%	1133	3360	113	204	online
MOT16-07	KDNT*	58.4	54.4	80.8	40.7%	7.4%	805	5887	99	113	batch
	NOMT	49.9	50.6	79.4	24.1%	24.1%	752	7365	61	75	batch
	MCMOT_HDM	54.4	44.3	78.6	22.2%	13.0%	619	6678	144	128	batch
	SORT*	56.2	45.9	79.5	24.1%	9.3%	557	6418	170	213	online
	DeepSORT*	57.9	54.1	79.0	35.2%	5.6%	950	5816	101	203	online
	RAR16wVGG	59.3	55.1	78.9	31.5%	7.4%	607	5960	79	130	online
	NSH(proposed)*	58.7	54.8	78.9	37.0%	5.6%	1012	5635	102	196	online
MOT16-08	KDNT*	39.6	45.3	81.4	22.2%	33.3%	566	9448	96	112	batch
	NOMT	42.9	40.0	80.0	22.2%	30.2%	675	8801	83	91	batch
	MCMOT_HDM	41.9	34.9	78.8	19.0%	15.9%	1378	8132	216	210	batch
	SORT*	35.4	36.5	81.0	15.9%	31.7%	823	9801	179	222	online
	DeepSORT*	37.1	41.6	79.8	19.0%	25.4%	1279	9094	155	255	online
	RAR16wVGG	36.3	40.3	81.0	22.2%	33.3%	689	9883	80	116	online
	NSH(proposed)*	37.9	43.4	79.6	19.0%	20.6%	1311	8948	128	240	online
MOT16-12	KDNT*	48.2	57.5	80.1	20.9%	39.5%	457	3795	46	53	batch
	NOMT	50.3	60.3	80.0	31.4%	39.5%	497	3606	22	27	batch
	MCMOT_HDM	42.3	53.1	80.5	24.4%	33.7%	1165	3559	62	53	batch
	SORT*	45.3	54.7	79.9	18.6%	39.5%	478	4000	58	81	online
	DeepSORT*	44.4	58.2	79.3	22.1%	31.4%	950	3625	38	96	online
	RAR16wVGG	46.6	60.2	79.7	24.4%	41.9%	655	3743	30	44	online
	NSH(proposed)*	43.4	56.9	79.5	22.1%	33.7%	1063	3585	47	76	online
MOT16-14	KDNT*	45.4	45.9	75.8	21.3%	22.0%	2177	7629	285	392	batch
	NOMT	39.8	53.3	73.0	14.6%	45.1%	1210	9857	51	170	batch
	MCMOT_HDM	42.0	48.2	74.7	18.9%	31.7%	1284	9182	260	250	batch
	SORT*	42.4	46.4	76.7	7.3%	28.7%	1206	9083	350	379	online
	DeepSORT*	46.3	52.5	75.5	14.6%	23.2%	1996	7734	189	368	online
	RAR16wVGG	50.2	57.4	76.2	23.2%	26.8%	1519	7581	100	255	online
	NSH(proposed)*	47.3	54.4	75.7	23.8%	20.7%	2282	7286	174	337	online

on the MOT2016 dataset. Because the MOT2016 dataset annotates all pedestrian targets at any scale and distance, the detection recall of Yolo detector is not high enough under such conditions. So we select the detector provided by KDNT [50] to detect the targets in this dataset. We choose other tracking algorithms, including KDNT, NOMT, MCMOT_HDM [51], SORT, DeepSORT [52] and RAR16wVGG [53], to compare with the proposed algorithm. Table 3 and Table 4 shows the comparisons of the proposed algorithm with other algorithms in the entire dataset and each sequence of the dataset respectively. It can be seen that the algorithm has also achieve competitive accuracy with offline algorithms on the dataset. The tracking result of the algorithm is shown in Figure 9.

D. DISCUSSION

Our algorithm establishes the motion model by Kalman filter, associates the prediction result of the tracklet with the detection result by distance and scale, and obtains the preliminary estimation of the tracking result. Due to the reasonable setting of tracking strategy, it is robust to the errors of detectors in some frames. Thus, this method achieves good results in short-term tracking. Because of the low computational complexity of Kalman filter, the motion model has high real-time performance, and 178.8 fps can be achieved in 2D MOT 2015 dataset. However, in long-term tracking, the motion model is no longer reliable because of the frequent occlusion. So we add the appearance model based on re-identifying network to improve the tracking results. By fusing appearance model distance and standardized pixel distance, the robustness of tracking to occlusion problem is further improved, which makes the ID switch of the algorithm reduce from 1372 to 578, and the algorithm achieves high results in IDF1, ID Precision and ID Recall. At the same time, in order to reduce the impact of adding appearance model on the real-time performance of the algorithm, channel pruning method is used to reduce the computational complexity of appearance feature extraction network, so that the algorithm maintains a high processing speed.

V. CONCLUSION

In this paper, a fast online multi-pedestrian tracking method based on integrating motion model and deep appearance model is proposed. The method establishes motion model with Kalman filter and appearance model with deep re-identification network. Data association algorithm based on Hungarian algorithm is used to integrate motion model and long-term and short-term appearance models, then match the tracklets and detections to complete the tracking task. The experiment results show the effectiveness and robustness of the proposed algorithm, and prove that the algorithm has a good balance in accuracy and speed. Although the algorithm has achieved good results, there are still some shortcomings that need further improvement. Since the computational complexity of the algorithm concentrates mainly on the establishment of pedestrian re-identification appearance model, our future work will focus on the research of more lightweight, fast and high-precision appearance model to further improve the accuracy and speed of the tracking algorithm.

REFERENCES

[1] L. Liang-Qun, Z. Xi-Yang, L. Zong-Xiang, and X. Wei-Xin, "Fuzzy logic approach to visual multi-object tracking," *Neurocomputing*, vol. 281, pp. 139–151, Mar. 2018.

[2] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, X. Zhao, and T.-K. Kim, "Multiple object tracking: A literature review," 2014, *arXiv:1409.7618*. [Online]. Available: https://arxiv.org/abs/1409.7618

[3] J. Chen, H. Sheng, Y. Zhang, and Z. Xiong, "Enhancing detection model for multiple hypothesis tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 2143–2152.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[8] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[9] C. Zhou and J. Yuan, "Bi-box regression for pedestrian detection and occlusion estimation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2018, pp. 138–154.

[10] L. Zhang and L. van der Maaten, "Preserving structure in model-free tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 756–769, Apr. 2014.

[11] L. Zhang and L. van der Maaten, "Structure preserving object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1838–1845.

[12] C.-H. Kuo, C. Huang, and R. Nevatia, "Multi-target tracking by on-line learned discriminative appearance models," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 685–692.

[13] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 58–72, Jan. 2014.

[14] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[15] W. Choi, "Near-online multi-target tracking with aggregated local flow descriptor," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3029–3037.

[16] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "MOTChallenge 2015: Towards a benchmark for multi-target tracking," 2015, *arXiv:1504.01942*. [Online]. Available: https://arxiv.org/abs/1504.01942

[17] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," 2016, *arXiv:1603.00831*. [Online]. Available: https://arxiv.org/abs/1603.00831

[18] W. Tian, M. Lauer, and L. Chen, "Online multi-object tracking using joint domain information in traffic scenarios," *IEEE Trans. Intell. Transp. Syst.*, to be published.

[19] S. Sharma, J. A. Ansari, J. K. Murthy, and K. M. Krishna, "Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 3508–3515.

[20] S. Scheidegger, J. Benjaminsson, E. Rosenberg, A. Krishnan, and K. Granström, "Mono-camera 3D multi-object tracking using deep learning detections and PMBM filtering," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 433–440.

[21] N. L. Baisa and A. Wallace, "Development of a n-type GM-PHD filter for multiple target, multiple type visual tracking," *J. Vis. Commun. Image Represent.*, vol. 59, pp. 257–271, Feb. 2019.

[22] W. Luo, T.-K. Kim, B. Stenger, X. Zhao, and R. Cipolla, "Bi-label propagation for generic multiple object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1290–1297.

[23] J. Gao, T. Zhang, X. Yang, and C. Xu, "P2T: Part-to-target tracking via deep regression learning," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 3074–3086, Jun. 2018.

[24] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua, "Multiple object tracking using k-shortest paths optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1806–1819, Sep. 2011.

[25] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Robust tracking-by-detection using a detector confidence particle filter," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 1515–1522.

[26] A. Ess, B. Leibe, K. Schindler, and L. van Gool, "Robust multiperson tracking from a mobile platform," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 10, pp. 1831–1846, Oct. 2009.

[27] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4705–4713.

[28] Z. Qin and C. R. Shelton, "Improving multi-target tracking via social grouping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1972–1978.

[29] H.-U. Kim and C.-S. Kim, "CDT: Cooperative detection and tracking for tracing multiple objects in video sequences," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 851–867.
[30] J. F. Henriques, R. Caseiro, and J. Batista, "Globally optimal solution to multi-object tracking with merged measurements," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2470–2477.
[31] N. K. Verma, R. Dev, S. Maurya, N. K. Dhar, and P. Agrawal, "People counting with overhead camera using fuzzy-based detector," in *Computational Intelligence: Theories, Applications and Future Directions*, vol. 1. Singapore: Springer, 2019, pp. 589–601.
[32] M. Dimitrievski, P. Veelaert, and W. Philips, "Behavioral pedestrian tracking using a camera and LiDAR sensors on a moving vehicle," *Sensors*, vol. 19, no. 2, p. 391, 2019.
[33] H. Yang and S. Qu, "Real-time vehicle detection and counting in complex traffic scenes using background subtraction model with low-rank decomposition," *IET Intell. Transp. Syst.*, vol. 12, no. 1, pp. 75–85, Nov. 2017.
[34] W. Aitfares, A. Kobbane, and A. Kriouile, "Suspicious behavior detection of people by monitoring camera," in *Proc. 5th Int. Conf. Multimedia Comput. Syst. (ICMCS)*, Sep./Oct. 2016, pp. 113–117.
[35] D. Mitzel and B. Leibe, "Real-time multi-person tracking with detector assisted structure propagation," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Nov. 2011, pp. 974–981.
[36] T. Yu, Y. Wu, N. O. Krahnstoever, and P. H. Tu, "Distributed data association and filtering for multiple target tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[37] J. Zhang and P. Zhou, "Integrating low-resolution surveillance camera and smartphone inertial sensors for indoor positioning," in *Proc. IEEE/ION Position, Location Navigat. Symp. (PLANS)*, Apr. 2018, pp. 410–416.
[38] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3074–3082.
[39] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3464–3468.
[40] B. Yang and R. Nevatia, "Multi-target tracking by online learning of non-linear motion patterns and robust appearance models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1918–1925.
[41] B. Wu and R. Nevatia, "Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors," *Int. J. Comput. Vis.*, vol. 75, no. 2, pp. 247–266, Nov. 2007.
[42] A. G. A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and W. Hu, "Multi-object tracking through simultaneous long occlusions and split-merge conditions," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2006, pp. 666–673.
[43] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: https://arxiv.org/abs/1804.02767
[44] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS—Improving object detection with one line of code," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5562–5570.
[45] Y. He, X. Zhang, and J. Sun, "Channel pruning for accelerating very deep neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1398–1406.
[46] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.
[47] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," 2017, *arXiv:1703.07737*. [Online]. Available: https://arxiv.org/abs/1703.07737
[48] S.-H. Bae and K.-J. Yoon, "Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 595–610, Mar. 2018.
[49] B. Wang, G. Wang, K. L. Chan, and L. Wang, "Tracklet association by online target-specific metric learning and coherent dynamics estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 3, pp. 589–602, Mar. 2016.
[50] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan, "POI: Multiple object tracking with high performance detection and appearance feature," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 36–42.
[51] B. Lee, E. Erdenee, S. Jin, M. Y. Nam, Y. G. Jung, and P. K. Rhee, "Multi-class multi-object tracking using changing point detection," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 68–83.
[52] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3645–3649.
[53] K. Fang, Y. Xiang, X. Li, and S. Savarese, "Recurrent autoregressive networks for online multi-object tracking," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 466–475.

MIAO HE received the B.S. degree from the College of Electronic Information and Optical Engineering, Nankai University, China, in 2015. He is currently pursuing the Ph.D. degree with the Shenyang Institute of Automation, Chinese Academy of Sciences. His current research interests include target detection, semantic segmentation, multi-target tracking, and pedestrian re-identifying.

HAIBO LUO received the B.S. degree in electronic engineering from the Harbin Institute of Technology, in 1990, and the Ph.D. degree in pattern recognition and intelligent systems from the Shenyang Institute of Automation, Chinese Academy of Sciences, in 2009, where he is currently a Professor. His research interests include real-time image processing, automatic target recognition, and polarization imaging.

BIN HUI received the B.S. degree from Northeastern University, China, in 1996, and the M.S. degree from the Shenyang Institute of Automation, Chinese Academy of Sciences, China, in 2009, where he is currently a Professor. His research interests include real-time image processing, pattern recognition, and polarization imaging.

ZHENG CHANG received the B.S. degree in automation from the Huazhong University of Science and Technology, China, in 1999. He is currently a Professor with the Shenyang Institute of Automation, Chinese Academy of Sciences. His research interests include pattern recognition, photoelectric imaging, and image processing.